CHAPTER 8

Cisco Web Dialer API Programming

This chapter describes the Simple Object Access Protocol (SOAP) and HTML over secure HTTP (HTTPS) interfaces that are used to develop customized click-to-dial applications for Cisco Web Dialer (Web Dialer) for Cisco Unified Communications Manager (Unified CM) and contains the following sections:

Overview

Web Dialer is a service that can be activated on a Unified CM subscriber to enable custom developed click-to-dial applications to issue MakeCall requests on behalf of a user. These applications can be server based, such as a click-to-dial enabled corporate directory, or desktop-based, such as an Outlook plug-in that lets users click to dial contacts.

The two main components of Web Dialer are the Web Dialer servlet and the Redirector servlet.

Table 8-1 explains some terms that are used in this chapter.

Table 8-1 Web Dialer Terms

Cisco Web Dialer Service	A server-based component of Cisco Unified Communications Manager that allows users to make calls from web and desktop applications.
Cisco Web Dialer Application	A customer or partner created application that calls SOAP or HTTP methods from the Web Dialer interface library.

Table 8-1 Web Dialer Terms (continued)

Web Dialer Servlet	A Java servlet that responds to SOAP or HTTP requests.
Redirector Servlet	A Java servlet that finds the home Unified Communications Manager cluster of a user and responds with one or more IP addresses of the Web Dialer enabled subscribers within the home cluster.

New and Changed Information

The following sections provide information on the changes in the Web Dialer APIs in Unified CM release 8.6(1) and the previous releases:

- New Information for Cisco Unified Communications Manager 8.6(1), page 8-2
- New and Changed Information in Previous Releases of Unified CM, page 8-2

For information about new, changed, or deprecated Web Dialer API methods from the interface library, see Chapter 9, "Cisco Web Dialer Operations By Release."

New Information for Cisco Unified Communications Manager 8.6(1)

There are no changes in the Web Dialer APIs in Unified CM release 8.6(1).

New and Changed Information in Previous Releases of Unified CM

The following sections provide the new and changed information in the older releases of Unified CM:

- New Information for Cisco Unified Communications Manager 8.5(1), page 8-2
- New Information for Cisco Unified Communications Manager 8.0(1), page 8-3
- New Information for Cisco Unified Communications Manager 7.1(2), page 8-3
- New Information for Cisco Unified Communications Manager 7.0, page 8-3
- New Information for Cisco Unified Communications Manager 6.0, page 8-3
- New Information for Cisco Unified Communications Manager 5.1, page 8-3

For information about new, changed, or deprecated Web Dialer API methods from the interface library, see Chapter 9, "Cisco Web Dialer Operations By Release."

New Information for Cisco Unified Communications Manager 8.5(1)

The following change was done in the Web Dialer APIs in Unified CM release 8.5(1):

- The Maximum Concurrent Call Requests was raised from six to eight. For more information, see Maximum Concurrent Call Requests, page 8-7

New Information for Cisco Unified Communications Manager 8.0(1)

A new service parameter, called Maximum Concurrent Call Requests, is added for modifying the throttle value of Web Dialer requests. This value was previously hard-coded. The throttle limits the number of CTI requests from Web Dialer. The minimum and maximum values for this throttle are one and six, and the recommended values for 7825 and 7845 servers are three and six respectively.

For more information, see Maximum Concurrent Call Requests, page 8-7.

New Information for Cisco Unified Communications Manager 7.1(2)

There are no changes in Web Dialer for Unified CM 7.1(2).

New Information for Cisco Unified Communications Manager 7.0

The following SOAP API methods have been added for Web Dialer in Unified CM 7.0:

- getProfileDetailSoap
- getPrimaryLine

New Information for Cisco Unified Communications Manager 6.0

Unified CM 6.0 includes the following change to Web Dialer:

- The getProfilSoap method returns only devices that are supported by Web Dialer. These devices are derived from those that are supported by Cisco JTAPI. Devices that are not supported by Cisco JTAPI are no longer returned. For additional information, refer to *Cisco Unified Communications Manager JTAPI Developers Guide* for release 6.0(1), which is available at this URL:

  http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/jtapi_dev/6_0_1/jtapi-dev.html

- Application Dial Rules support has been added for the SOAP API.

New Information for Cisco Unified Communications Manager 5.1

Unified CM 5.1 includes the following change to Web Dialer:

- Web Dialer and Redirector now require HTTPS.

  Developers should format Redirector and Web Dialer requests to use HTTPS. Unified CM requires the secured protocol to prevent unauthorized applications from reading user data.

Refer to *Cisco Unified CallManager Developers Guide for Release 5.0* for important changes to Web Dialer API programming in the 5.0 release.

Cisco Web Dialer Components

The following sections provide information about Web Dialer Components:

Cisco Web Dialer Servlet

The Web Dialer servlet, a Java servlet, allows Cisco Unified Communications Manager users in a specific cluster to make and end calls, as well as to access their phone and line configuration.

Cisco Web Dialer applications interact with the Web Dialer servlet through two interfaces:

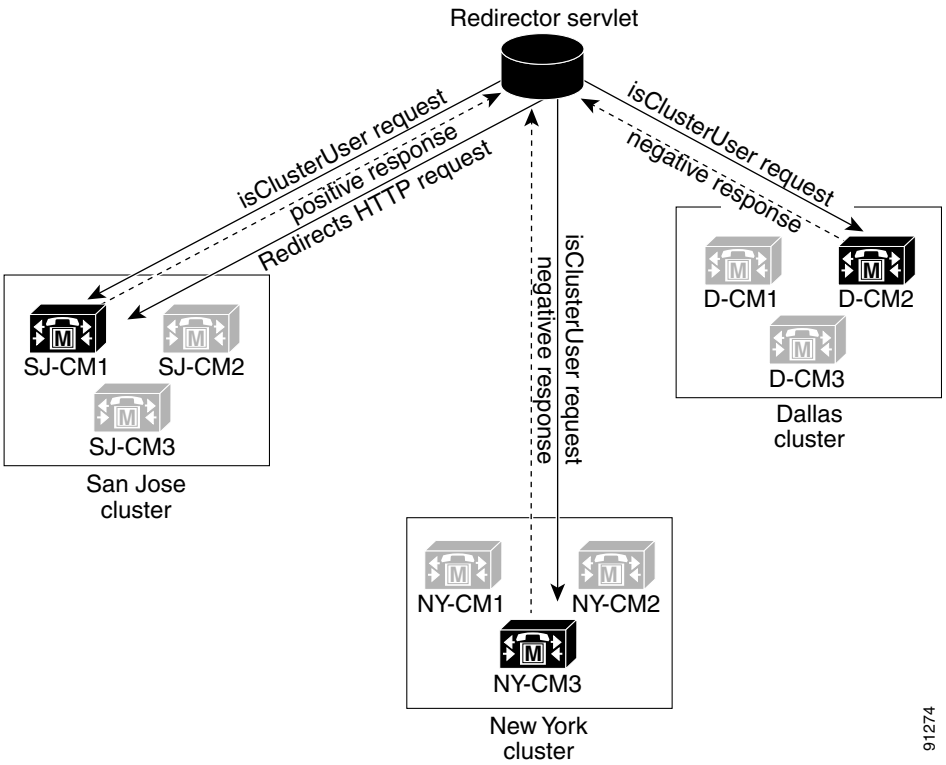
- SOAP over HTTPS—This interface, based on the Simple Object Access Protocol (SOAP), is used to develop desktop applications such as a Microsoft Outlook Plug-in or a SameTime Client Plug-in. Developers can use the isClusterUserSoap interface to design multicluster applications that require functionality similar to a Redirector servlet.
- HTML over HTTPS—This interface, based on the HTTPS protocol, is used to develop web-based applications such as the Unified CM directory search page (directory.asp). Developers who use this interface can use the Redirector servlet for designing multicluster applications.

Redirector Servlet

The Java-based Redirector servlet is responsible for distributing web (HTTP and HTTPS) MakeCall requests to the home Web Dialer server of a user. Redirector generally is used in a multi-cluster environments to instruct an application where to send MakeCall requests. When Redirector receives a MakeCall request, it sends the IsClusterUser broadcast message to all configured Web Dialer servers in the Enterprise. When Redirector receives a positive response, it forwards the request to the appropriate Web Dialer server. Redirector is available for HTTP and HTTPS applications only. SOAP-based applications are responsible for sending the MakeCall request to the home Web Dialer server of a user.

Figure 8-1 illustrates how a Redirector servlet redirects a call in a multicluster environment.

Figure 8-1 Multiple Clusters



Example of Web Dialer Using the Redirector Servlet

For example, consider three clusters, each one in a single city such as San Jose, Dallas, and New York. Each cluster contains three Cisco Unified Communications Manager servers with Web Dialer servlets that have been configured for Cisco Unified Communications Manager servers SJ-CM1, D-CM2, and NY-CM3.

The system administrator configures the Web Dialer servlets on any Cisco Unified Communications Manager server by entering the IP address of that specific Cisco Unified Communications Manager server in the List of WebDialers service parameter.

For information about configuring Web Dialer and Redirector servlets, refer to the "Web Dialer" chapter in the *Cisco Unified Communications Manager Features and Services Guide, Release* 5.0.

When a user who is located in San Jose clicks a telephone number in the corporate directory search page that is enabled by Web Dialer, the following actions happen:

1. The Cisco Unified Communications Manager server sends an initial *makeCall* HTTPS request to the Redirector servlet.
2. If this request is received for the first time, the Redirector servlet reads the Web Dialer server cookie and finds it empty.

   For a repeat request, the Redirector servlet reads the IP address of the Web Dialer server that previously serviced the client and sends a *isClusterUser* HTTPS request only to that server.
3. The Redirector servlet sends back a response that asks for information, which results in the authentication dialog box opening for the user.
4. The user enters the Cisco Unified Communications Manager user ID and password and clicks the **Submit** button.
5. The Redirector servlet reads only the user identification from this information and sends a *isClusterUser* HTTPS request to each Web Dialer server that the system administrator configured.

   Figure 8-1 illustrates how this request is sent to the Web Dialer servlets that have been configured for SJ-CM1, D-CM2, and NY-CM3. Depending on the geographical location of the calling party, the Web Dialer servlet from that cluster responds positively to the Redirector servlet. The remaining Web Dialer servlets that were contacted return a negative response. The Web Dialer servlet SJ-CM1 responds positively to the request because the calling party is located in San Jose (SJ-CM).

The Redirector servlet redirects the original request from the user to SJ-CM1 and sets a cookie on the user browser for future use.

Cisco Web Dialer Security Support

Web Dialer supports secure connections to CTI (TLS connection). For this feature, Web Dialer uses the security API that JTAPI provides. Refer to *Unified CM JTAPI Developers Guide* for the JTAPI API. Web Dialer uses the Application User, "WDSecureSysUser", for obtaining the CTI connection.

You must complete the following configuration before Web Dialer can be configured to open a CTI connection in secure mode.

Step 1 Activate the Cisco CTL Provider service in Cisco Unified Communications Manager Service Administration.

Step 2 Activate the Cisco Certificate Authority Proxy Function Service.

Step 3 Download the Cisco CTL Client from the Application plug-in and install it on any machine.

Step 4 Run the CTL Client, choose the option to "enable Cluster Security," and follow the instructions that display. This requires USB E-tokens.

Step 5 To verify that cluster security is enabled, go to Cisco Unified Communications Manager Administration and look at [System-> Enterprise Parameter configuration]. Look at the Security Parameters; the cluster security should be set to 1.

Step 6 In Cisco Unified Communications Manager Administration, from the User Management drop-down menu, select the Application User CAPF Profile option.

Step 7 Click **Add new InstanceID**.

Step 8 In the CAPF Profile configuration window, set up an InstanceID and CAPF profile for the InstanceID for the Application User WDSecureSysUser.

a. **InstanceID**: Enter the value of instance ID; for example, 001.

b. **Certificate Operation**: Select Install/Upgrade from the drop-down menu.

c. **Authentication Mode**: Select By Authorization String from the drop-down menu.

d. **Authorization String**: Enter the value of authorization string; for example, 12345.

e. **Key Size**: Select key size from drop-down menu; for example, 1024.

f. **Operation Completes By**: Enter the date and time in following format yyyy:mm:dd:hh:mn where yyyy=year, mm=month, dd=date, hh=hour, mn=minutes, such as 2006:07:30:12:30.

Note If this date and time has passed, the certificate update operation will fail.

g. Ignore the **Packet Capture Mode**, **Packet Capture Duration**, and **Certificate** fields.

h. **Certificate Status**: Select Operation pending from the drop-down menu.

If anything else is selected, the certificate update will fail.

Security Service Parameters

Web Dialer includes two mode-specific service parameters for CTI connection security.

- **CTI Manager Connection Security Flag**—This required service parameter indicates whether security for the Web Dialer service CTI Manager connection is enabled or disabled.

  If enabled (true), Cisco Web Dialer will open a secure connection to CTI Manager by using the Application CAPF profile that is configured for the instance ID (as configured in CTI Manager Connection Instance ID service parameter) for Application user WDSecureSysUser. The default value specifies false.

- **Application CAPF Profile Instance ID**: This service parameter specifies the Instance ID of the Application CAPF Profile for Application User WDSecureSysUser that this Web Dialer server will use to open a secure connection to CTI Manager. You must configure this parameter if the CTI Manager Connection Security Flag parameter is enabled (true).

- **Algorithm**:

  1. Read the service parameters.
  2. Get the node IP/name of the nodes where TFTP and CAPF are activated.
  3. For the instanceID (input in service parameters), if the Certificate Operation is 'Install/Upgrade' or 'Delete', delete the current certificates, if any.

4. If the Certificate Operation is not ‘Install/Upgrade’ or ‘Delete’, and a current certificate exists, use this certificate.
5. If no certificate is present, request one by using JTAPI API setSecurityPropertyForInstance; this will need username, instanceID, authCode, tftpServerName, tftpPort, capfServerName, capfPort, certPath, and securityFlag. This call will contact the TFTP server, download the certificate, contact the CAPF server, verify the CTL file, and request the client and server certificates.
6. If Step 5 is successful, set the following items on the ICCNProvider and call open().provider.setInstanceID(instanceID);provider.setTFTPServer(tftpServerName);provider.setCAPFServer(capfServerName);provider.setCertificatePath(certPath);provider.setSecurityOptions(securityFlag);
7. If Step 5 fails, throw initFailedException. You can see this in the Web Dialer traces.

Maximum Concurrent Call Requests

The maximum concurrent call request parameter specifies the maximum number of concurrent Web Dialer call requests per second supported by the Web Dialer service. This value was previously hard-coded. The minimum and maximum values for this throttle are one and eight, and the recommended values for 7825 and 7845 servers are three and six respectively. Version 8.6(1) increased the maximum from six to eight.

For example:

- MCS 7825H2 supports a maximum of two calls per second. Cisco recommends setting the MaxConcurrentCallRequests value to three to allow callers to disconnect the call as needed.
- MCS 7845H2 supports a maximum of four calls per second. Cisco recommends setting the MaxConcurrentCallRequests value to six to allow callers to disconnect the call as needed.

Enter a lower value if RTMT alerts, alarms, or performance counters suggest that the hardware associated with Web Dialer is over-utilized (for example, high CPU and/or memory usage). Enter a higher value to allow more simultaneous WebDialer call requests. Higher values can add more load to the CPU. The default value for this parameter is three.

Phone Support For Cisco Web Dialer

Web Dialer relies on Cisco JTAPI to place calls on the behalf of users. Table 8-2 provides information about CTI supported devices.

Table legend:

- —Supported
- —Not supported

Table 8-2 CTI Supported Device Matrix

Device/Phone Model	SCCP	SIP	Comments
Analog Phone			You can find information on the limitations of this device in *Cisco JTAPI Developer Guide for Cisco Unified CallManager 4.1(3)* .
Cisco 6901			

Table 8-2 CTI Supported Device Matrix (continued)

Device/Phone Model	SCCP	SIP	Comments
Cisco 6911	✓	✓	
Cisco 6921	✓	✓	
Cisco 6941	✓	✓	
Cisco 6961	✓	✓	
Cisco 7902	✓	✗	End of Software Maintenance Release 2007
Cisco 7905	✓	✗	End of Software Maintenance Release 2007
Cisco 7906	✓	✓	
Cisco 7910	✓	✗	End of Software Maintenance Release 2007
Cisco 7911	✓	✓	
Cisco 7912	✓	✗	End of Software Maintenance Release 2007
Cisco 7914 Sidecar	✓	✗	End of Software Maintenance Release 2010
Cisco 7915 Sidecar	✓	✓	
Cisco 7916 Sidecar	✓	✓	
Cisco CKEM Sidecar	✗	✓	
Cisco 7920	✓	✗	End of Software Maintenance Release 2008
Cisco 7921	✓	✗	
Cisco 7925 & 7925-EX	✓	✗	
Cisco 7931	✓	✗	CTI supported only if rollover is disabled. Starting withrelease 7.1 this device is supported when corresponding role is added to user.
Cisco 7935	✓	✗	End of Software Maintenance Release 2005
Cisco 7936	✓	✗	End of Software Maintenance Release 2011
Cisco 7937	✓	✗	
Cisco 7940	✓	✗	End of Software Maintenance Release 2011
Cisco 7941	✓	✓	
Cisco 7941G-GE	✓	✓	End of Software Maintenance Release 2009
Cisco 7942	✓	✓	
Cisco 7945	✓	✓	
Cisco 7960	✓	✗	End of Software Maintenance Release 2011

Table 8-2 CTI Supported Device Matrix (continued)

Device/Phone Model	SCCP	SIP	Comments
Cisco 7961	✓	✓	
Cisco 7961G-GE	✓	✓	End of Software Maintenance Release 2009
Cisco 7962	✓	✓	
Cisco 7965	✓	✓	
Cisco 7970	✓	✓	End of Software Maintenance Release 2009
Cisco 7971	✓	✓	End of Software Maintenance Release 2009
Cisco 7975	✓	✓	
Cisco 7985	✓	✗	End of Software Maintenance Release 2011
Cisco 8941	✓	✗	
Cisco 8945	✓	✗	
Cisco 8961	✗	✓	phoneSetDisplay() interface is not supported
Cisco 9951	✗	✓	phoneSetDisplay() interface is not supported
Cisco 9971	✗	✓	phoneSetDisplay() interface is not supported
Cisco ATA 186	✓	✗	You can find information on the limitations of this device in *Cisco JTAPI Developer Guide for Cisco Unified CallManager 4.1(3)* .
Cisco Cius	✗	✓	CTI support added in release 8.5(1) phoneSetDisplay() interface is not supported XSI interface is not supported. Silent Monitoring/Recording is not supported
Cisco IP Communicator	✓	✓	CTI support added in release 7.1(2)
Cisco Unified Personal Communicator	✗	✗	CTI support when running in desktop mode depends on physical device. CTI support added in release 7.5(1)
Cisco Unified Personal Communicator - Remote Desktop Control Mode	—	—	Refer to the device model under remote control to determine CTI support. Click-to-Answer requires device speakerphone support.
Cisco Unified Communicator Integration for Microsoft Office Communicator/Lync - Softphone Mode	✗	✓	CTI support added in release 8.5(2)

Table 8-2 CTI Supported Device Matrix (continued)

Device/Phone Model	SCCP	SIP	Comments
Cisco Unified Communicator Integration for Microsoft Office Communicator/Lync - Remote Desktop Control Mode	—	—	Refer to the device model under remote control to determine CTI support. Click-to-Answer requires device speakerphone support.
Cisco Web Communicator for Quad - Softphone Mode	—	—	Not a CTI supported device.
Cisco Web Communicator for Quad - Remote Desktop Control Mode	—	—	Refer to the device model under remote control to determine CTI support. Click-to-Answer requires device speakerphone support.
Cisco Unified Communications Integration for WebEx Connect - Softphone Mode	—	—	Not a CTI supported device.
Cisco Unified Communications Integration for WebEx Connect - Remote Desktop Control Mode	—	—	Refer to the device model under remote control to determine CTI support. Click-to-Answer requires device speakerphone support.
Cisco VGC Phone	✓	✗	
VG224	—	—	Not a CTI supported device.
VG248	✓	✗	You can find information on the limitations of this device in *Cisco JTAPI Developer Guide for Cisco Unified CallManager 4.1(3)* .
CTI Port	—	—	CTI supported virtual device that does not use SCCP or SIP
CTI Route Point	—	—	CTI supported virtual device that does not use SCCP or SIP
CTI Route Point (Pilot Point)	—	—	CTI supported virtual device that does not use SCCP or SIP
ISDN BRI Phone	—	—	Not a CTI supported device

Call Flows

The call flows in this section describe the flow of events for client and browser-based applications that use Web Dialer, which should help you design customized applications for Web Dialer.

Desktop-based Client Application Call Flow

Figure 8-2 shows the call flow for an outgoing call from a client application. The user clicks the **Dial** or **Make Call** button in the client application. If the user is making a call for the first time, the application does not have authentication or configuration information on the user.

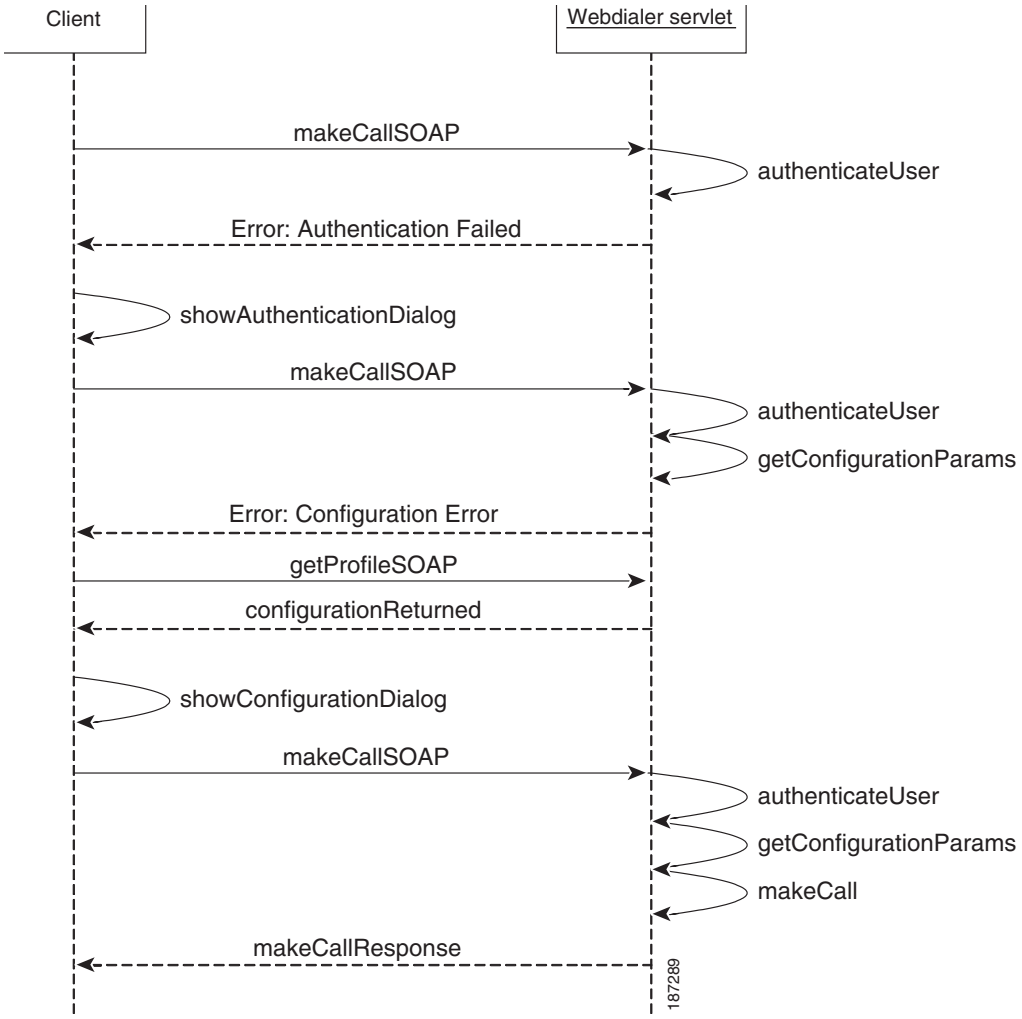
When the user makes a call for the first time,

1. The client sends a makeCallSoap request to the configured Web Dialer servlet.
2. The Web Dialer servlet attempts to authenticate the user. Figure 8-2 shows an authentication failure that occurred because the authentication information is incomplete or does not exist.
3. The Web Dialer servlet sends an authentication failure response to the client application.
4. The client application displays a dialog box that asks for the user ID and password. The user enters this information and clicks the **submit** button. The user ID and password get stored for future invocations of the application.
5. The application sends a repeat SOAP request to the Web Dialer servlet. The request contains credential information on the user.
6. The Web Dialer servlet authenticates the user.
7. The Web Dialer servlet reads any missing configuration information in the request.
8. The Web Dialer servlet returns a configuration error message to the client application.
9. The client application sends a getProfileSoap request to the Web Dialer servlet.
10. The Web Dialer servlet responds with the user configuration information that is stored in the directory.
11. The client application displays a configuration dialog box that asks the user to select or update the configuration. The user enters the information and clicks the **submit** button. The user configuration information gets stored for future invocations of the application.
12. The client resends the makeCallSoap request to the Web Dialer servlet. This request contains the user configuration information.
13. The Web Dialer servlet authenticates the user and dials the telephone number by using the information that the makeCallSoap request contains. It responds to the client with a success or failure message.

Note The call flow goes directly to step 12 in these situations:

- If the credential and configuration information is already stored when the application is installed.
- For all subsequent requests that the user makes.

Figure 8-2 Cisco Web Dialer Call Flow for a Client-Based Application

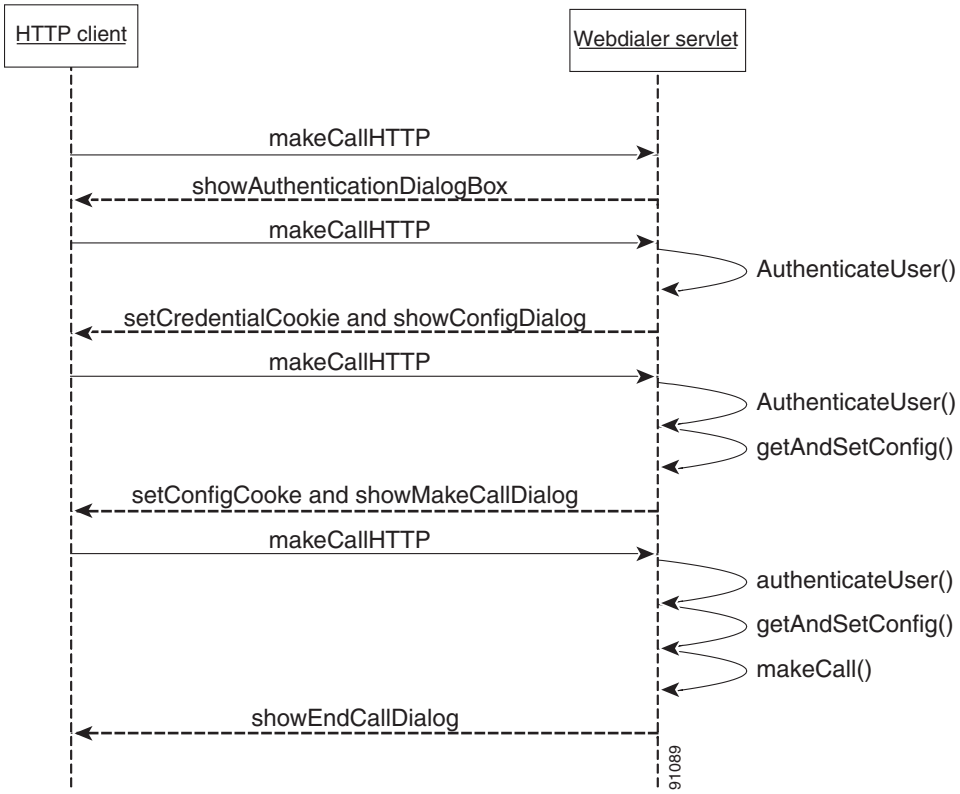


Browser-Based Application Call Flow

Figure 8-3 shows the call flow for an HTTP-based browser application such as a directory search page, personal address book, or the Cisco Unified Communications Manager directory search page (directory.asp).

The user clicks the **Dial** or **Make Call** button in the address book of the client application. If the user is making a call for the first time, the application does not have authentication or configuration information on the user.

Figure 8-3 Cisco Web Dialer Call Flow for a Browser-Based Application



When the user makes a call for the first time:

1. The client sends a makeCall HTTPS request to the configured Web Dialer servlet. The query string contains the number to be called.
2. The Web Dialer servlet authenticates the user. Authentication fails because the authentication information is incomplete or does not exist.

Note Authentication succeeds if the user credentials are sent with the request, and the call flow goes directly to step 7.

3. The Web Dialer servlet sends an authentication dialog to the client browser for user authentication.
4. The user enters the user ID and password and clicks the **Submit** button.
5. The client sends a makeCallHTTPS request that contains the user credentials to the Web Dialer servlet.
6. The Web Dialer servlet authenticates the user.
7. The Web Dialer servlet reads the configuration information in the cookie that is sent with the request.
8. Assuming that the request is made for the first time, the servlet sends a response that contains a cookie to the client browser. The cookie that contains the client credentials gets stored on the client browser. The client credentials comprise user ID, IP address, and the time of the request.
9. The user enters the updates in the configuration dialog box and clicks the **Submit** button.
10. The client browser sends a makeCall HTTPS request to the Web Dialer servlet. The request contains a cookie with the credential and configuration information in parameter form.

11. The Web Dialer servlet uses the credentials to authenticate the user and saves the configuration information in its memory.
12. The Web Dialer servlet sends a makeCall confirmation dialog to the client browser with the configuration information that is stored in a cookie. The cookie gets stored on the client browser for future invocations.
13. The Make Call dialog box appears on the user computer screen. The user clicks the **Dial** button, which sends another makeCall HTTPS request to the Web Dialer servlet.
14. The Web Dialer servlet authenticates the user by using the credentials in the cookie, retrieves the configuration information from the cookie, and makes the call.
15. The servlet responds by sending an endCall confirmation dialog to the user to end the call. The End Call dialog box appears on the user computer screen and stays there for the time interval that is configured in the service parameters.

For subsequent requests, the call flow starts at step 12 and ends at step 15.

Interfaces

Web Dialer applications interact with the Web Dialer servlet through two interfaces:

- SOAP over HTTPS—This interface, based on the Simple Object Access Protocol (SOAP), is used to develop desktop applications such as a Microsoft Outlook Plug-in and SameTime Client Plug-in. Developers can use the isClusterUserSoap interface to design multicluster applications that require functionality similar to a Redirector servlet.
- HTML over HTTPS—This interface, based on the HTTPS protocol, is used to develop web-based applications such as the Cisco Unified Communications Manager directory search page (directory.asp). Developers who are using this interface can use the Redirector servlet for designing multicluster applications.

SOAP Over HTTPS Interface

To access the SOAP interfaces for Web Dialer, use the Web Dialer Web Service Definition Language (WSDL) in the "Cisco Web Dialer WSDL" section on page 8-26.

makeCallSoap

You access the makeCallSoap interface by initiating a SOAP request to the URL https://<CUCM_Server>/webdialer/services/WebdialerSoapService70 where CUCM_Server specifies the IP address of the Cisco Unified Communications Manager server where Web Dialer is configured.

Parameter	Mandatory	Description	Data Type	Range Values	Default Value
Destination	Mandatory	Standard canonical form. For example, +1 408 5551212 or extensions such as 2222. The optional service parameter "Apply Application Dial Rules on SOAP Dial Request" is False by default; if this parameter is True, the destination gets transformed according to the dial rules.	String	None	None
Credential	Mandatory	The user ID or password of the user or proxy user. For more information on creating a proxy user, see the *Cisco Web Dialer* chapter in the *Cisco Unified Communications Manager Features and Services Guide, Release 5.0.*	Refer to the credential data type in the "Cisco Web Dialer WSDL" section on page 8-26.	None	None
Profile	Mandatory	The profile that is used to make a call. A typical profile is a calling device such as an IP phone or line. The line should be in the same format as returned by getProfileSoap—<number>;<partition>	Refer to the profile data type in the "Cisco Web Dialer WSDL" section on page 8-26.	None	None

Results

See the "Cisco Web Dialer WSDL" section on page 8-26 for return values and their data type.

Error Code	Name	Type	Description	Action by application
0	responseCode	Integer	Success	Displays a dialog box.
	responseDescription	String	Success	
1	responseCode	Integer	Call failure error	Displays a relevant error message.
	responseDescription	String	Call failure error	
2	responseCode	Integer	Authentication error	Displays the authentication dialog where the user enters ID and password information.
	responseDescription	String	User authentication error	

Error Code	Name	Type	Description	Action by application
3	responseCode	Integer	No authentication proxy rights	Void for user-based applications.
	responseDescription	String	No authentication proxy rights	
4	responseCode	Integer	Directory error	Displays an appropriate directory error message.
	responseDescription	String	Directory error	
5	responseCode	Integer	No device is configured for the user, or missing parameters exist in the request.	The application initiates a getProfileSoap request and displays the selected device and line to the user.
	responseDescription	String	No device is configured for the user, or missing parameters exist in the request.	
6	responseCode	Integer	Service temporarily unavailable	Displays the appropriate error dialog with an option to try again.
	responseDescription	String	Service temporarily unavailable	
7	responseCode	Integer	Destination cannot be reached.	Displays the appropriate error dialog that allows the user to edit the dialed number.
	responseDescription	String	Destination cannot be reached.	
8	responseCode	Integer	Service error	Displays the appropriate error dialog.
	responseDescription	String	Service error	
9	responseCode	Integer	Service overloaded	Displays the appropriate error dialog with an option to try again.
	responseDescription	String	Service overloaded	

This example shows a makeCallSoap request:

```
<?xml version="1.0" encoding="utf-8" ?>
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:WD70">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:makeCallSoap soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
         <in0 xsi:type="urn:Credential">
            <userID xsi:type="xsd:string">wd</userID>
            <password xsi:type="xsd:string">55555</password>
         </in0>
         <in1 xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">1234</in1>
         <in2 xsi:type="urn:UserProfile">
            <user xsi:type="xsd:string">wd</user>
```

```
            <deviceName xsi:type="xsd:string">SEP001558C8970F</deviceName>
            <lineNumber xsi:type="xsd:string">1234</lineNumber>
            <supportEM xsi:type="xsd:boolean">false</supportEM>
            <locale xsi:type="xsd:string">English</locale>
            <dontAutoClose xsi:type="xsd:boolean">false</dontAutoClose>
            <dontShowCallConf xsi:type="xsd:boolean">true</dontShowCallConf>
         </in2>
      </urn:makeCallSoap>
   </soapenv:Body>
</soapenv:Envelope>
```

endCallSoap

You access the endCallSoap interface by initiating a SOAP request to the URL https://<CUCM_Server>/webdialer/services/WebdialerSoapService70 where CUCM_Server specifies the IP address of the Cisco Unified Communications Manager server where Web Dialer is configured.

Parameter	Mandatory	Description	Data Type	Range Values	Default Value
Credential	Mandatory	The user ID or password of the user or proxy user. For information on creating a proxy user, see the *Cisco Web Dialer* chapter in the *Cisco Unified Communications Manager Features and Services Guide, Release 5.0*.	Refer to the credential data type in "Cisco Web Dialer WSDL" section on page 8-26.	None	None
Profile	Mandatory	The profile that is used to make a call. A typical profile is a calling device such as an IP phone or line.	Refer to the profile data type in the "Cisco Web Dialer WSDL" section on page 8-26.	None	None

See the "Cisco Web Dialer WSDL" section on page 8-26 for return values and their data type.

Error Code	Name	Type	Description	Action by application
0	responseCode	Integer	Success	Displays a dialog box on the computer screen.
	responseDescription	String	Success	
1	responseCode	Integer	Call failure error	Displays a relevant error message.
	responseDescription	String	Call failure error	
2	responseCode	Integer	Authentication error	Displays authentication dialog for user to enter user ID and password.
	responseDescription	String	User authentication error	
3	responseCode	Integer	No authentication proxy rights	Void for user-based applications.
	responseDescription	String	No authentication proxy rights	

Error Code	Name	Type	Description	Action by application
4	responseCode	Integer	Directory error	Displays an appropriate directory error message.
	responseDescription	String	Directory error	
5	responseCode	Integer	No device is configured for the user, or missing parameters exist in the request.	The Application initiates a getProfileSoap request and displays the selected device and line to the user.
	responseDescription	String	No device is configured for the user, or missing parameters exist in the request.	
6	responseCode	Integer	Service temporarily unavailable	Displays the appropriate error dialog with an option to try again.
	responseDescription	String	Service temporarily unavailable	
7	responseCode	Integer	Destination cannot be reached.	Displays the appropriate error dialog that allows the user to edit the dialed number.
	responseDescription	String	Destination cannot be reached.	
8	responseCode	Integer	Service error	Displays appropriate error dialog.
	responseDescription	String	Service error	
9	responseCode	Integer	Service overloaded	Displays the appropriate error dialog with an option to try again.
	responseDescription	String	Service overloaded	

This example shows an endCallSoap request:

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:WD70">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:endCallSoap soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
         <in0 xsi:type="urn:Credential">
            <userID xsi:type="xsd:string">wd</userID>
            <password xsi:type="xsd:string">55555</password>
         </in0>
         <in1 xsi:type="urn:UserProfile">
            <user xsi:type="xsd:string">wd</user>
            <deviceName xsi:type="xsd:string">SEP001558C8970F</deviceName>
            <lineNumber xsi:type="xsd:string">1234</lineNumber>
            <supportEM xsi:type="xsd:boolean">false</supportEM>
            <locale xsi:type="xsd:string">English</locale>
            <dontAutoClose xsi:type="xsd:boolean">false</dontAutoClose>
            <dontShowCallConf xsi:type="xsd:boolean">true</dontShowCallConf>
         </in1>
      </urn:endCallSoap>
   </soapenv:Body>
```

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

getProfileSoap

You access the getProfileSoap interface, which is used by plug-in based clients, by initiating a SOAP request to the URL https://<CUCM_Server>/webdialer/services/WebdialerSoapService70 where CUCM_Server specifies the IP address of the Cisco Unified Communications Manager server where Web Dialer is configured.

Parameter	Mandatory/ Optional	Description	Data Type	Value Range	Default Value
Credential	Mandatory	User ID or password of the user or proxy user. For information on creating a proxy user, see the *Cisco Web Dialer* chapter in *Cisco Unified Communications Manager Features and Services Guide, Release 5.0*.	Refer to the credential data type in the "Cisco Web Dialer WSDL" section on page 8-26.	None	None
UserID	Mandatory	The user ID for which the configuration is requested.	String	None	None

See the "Cisco Web Dialer WSDL" section on page 8-26 for return values and their data type.

Error Code	Name	Type	Description	Action by plug-in application
0	responseCode	Integer	Returns an array of phones or lines on the phone that is associated with the user. Refer to the Cisco Web Dialer WSDL for the WDDeviceInfo data type. **Note** getProfileSoap API does not return the Extension Mobility device profiles associated with the user.	Displays a dialog box on the computer screen.
	responseDescription	String	Success	
	deviceInfoList	Array	Returns an array of the the WDDeviceInfo data type	
1	responseCode	Integer	No device configured for the user	Displays an appropriate error message.
	responseDescription	String	No device configured for the user	

Error Code	Name	Type	Description	Action by plug-in application
2	responseCode	Integer	Authentication error	Displays the authentication dialog where the user enters ID and password information.
	responseDescription	String	User authentication error	
3	responseCode	Integer	No authentication proxy rights	Void for user-based applications.
	responseDescription	String	No authentication proxy rights	
4	responseCode	Integer	Directory error	Displays an appropriate directory error message.
	responseDescription	String	Directory error	
6	responseCode	Integer	Service temporarily unavailable	Displays the appropriate error dialog with an option to try again.
	responseDescription	String	Service temporarily unavailable	
9	responseCode	Integer	Service overloaded	Displays the appropriate error dialog with an option to try again.
	responseDescription	String	Service overloaded	

This example shows a getProfileSoap request used for debugging purposes (normally, the SOAP implementation layer would make this request):

```
<?xml version="1.0" ?>
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:WD70">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:getProfileSoap
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
         <in0 xsi:type="urn:Credential">
            <userID xsi:type="xsd:string">wd</userID>
            <password xsi:type="xsd:string">55555</password>
         </in0>
         <in1 xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">wd</in1>
      </urn:getProfileSoap>
   </soapenv:Body>
</soapenv:Envelope>
```

isClusterUserSoap

You access the isClusterUserSoap interface by initiating a SOAP request to the URL https://<CUCM_Server>/webdialer/services/WebdialerSoapService70 where CUCM_Server specifies the IP address of the Cisco Unified Communications Manager server where Web Dialer is configured.

Use this SOAP interface for multicluster applications that require functionality, similar to a Redirector servlet, for redirecting calls to the various locations where Web Dialer is installed on a network. The application uses this interface to locate and verify the Web Dialer that is servicing the user, followed by makeCall, endCall, or getProfile requests to that Web Dialer.

Parameter	Mandatory	Description	Data Type	Range of Values	Default Value
UserID	Mandatory	The user ID for which the the request is made.	String	None	None

See the "Cisco Web Dialer WSDL" section on page 8-26 for return values and their data type.

Name	Type	Description
result	Boolean	The result specifies True if the user is present in the directory of the cluster. The result specifies False if the user is not present in the directory.

This example shows an isClusterUserSoap request:

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:WD70">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:isClusterUserSoap
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
         <in0 xsi:type="soapenc:string"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">wd</in0>
      </urn:isClusterUserSoap>
   </soapenv:Body>
</soapenv:Envelope>
```

getProfileDetailSoap

You access the getProfileDetailSoap interface by initiating a SOAP request to the URL https://<CUCM_Server>/webdialer/services/WebdialerSoapService70 where CUCM_Server specifies the IP address of the Cisco Unified Communications Manager server where Web Dialer is configured.

Parameter	Mandatory	Description	Data Type	Range Values	Default Value
Credential	Mandatory	User ID or password of the user or proxy user. For information about creating a proxy user, refer to the "Cisco Web Dialer" chapter in *Cisco Unified Communications Manager Features and Services Guide, Release 5.0*.	See the credential data type in the "Cisco Web Dialer WSDL" section on page 8-26.	None	None

Results

See the "Cisco Web Dialer WSDL" section on page 8-26 for return values and their data type.

Error Code	Name	Type	Description	Action by application
0	responseCode	Integer	Returns an array of phones or lines on the phone that is associated with the user. Also returns Phone Description and the Phone type for each device. See the credential data type in the "Cisco Web Dialer WSDL" section on page 8-26.	Displays a dialog box.
	responseDescription	String	Success	
	deviceInfoListDetail	Array	Returns an array of the the WDDeviceInfoDetail data type	
1	responseCode	Integer	No device configured for the user	Displays an appropriate error message.
	responseDescription	String	No device configured for the user	
2	responseCode	Integer	Authentication error	Displays the authentication dialog where the user enters ID and password information.
	responseDescription	String	User authentication error	

Error Code	Name	Type	Description	Action by application
3	responseCode	Integer	No authentication proxy rights	Void for user-based applications.
	responseDescription	String	No authentication proxy rights	
4	responseCode	Integer	Directory error	Displays an appropriate directory error message.
	responseDescription	String	Directory error	
6	responseCode	Integer	Service temporarily unavailable	Displays the appropriate error dialog with an option to try again.
	responseDescription	String	Service temporarily unavailable	
9	responseCode	Integer	Service overloaded	Displays the appropriate error dialog with an option to try again.
	responseDescription	String	Service overloaded	

This example shows a getProfileDetailSoap request used for debugging purposes (normally, the SOAP implementation layer would make this request):

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:WD70">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:getProfileDetailSoap
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
         <in0 xsi:type="urn:Credential">
            <userID xsi:type="xsd:string">wd</userID>
            <password xsi:type="xsd:string">55555</password>
         </in0>
      </urn:getProfileDetailSoap>
   </soapenv:Body>
</soapenv:Envelope>
```

getPrimaryLine

You access the getPrimaryLine interface by initiating a SOAP request to the URL https://<CUCM_Server>/webdialer/services/WebdialerSoapService70 where CUCM_Server specifies the IP address of the Cisco Unified Communications Manager server where Web Dialer is configured.

Parameter	Mandatory	Description	Data Type	Range of Values	Default Value
UserID	Mandatory	The user ID for which the the request is made.	String	None	None

See the "Cisco Web Dialer WSDL" section on page 8-26 for return values and their data type.

Name	Type	Description
result	Boolean	The result returns the number that is configured by the Unified CM administrator as the primary line of the user.

This example shows a getPrimaryLine request:

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:WD70">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:getPrimaryLine
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
         <in0 xsi:type="urn:Credential">
            <userID xsi:type="xsd:string">wd</userID>
            <password xsi:type="xsd:string">55555</password>
         </in0>
      </urn:getPrimaryLine>
   </soapenv:Body>
</soapenv:Envelope>
```

HTML Over HTTPS Interfaces

This section describes the HTML over HTTPS interfaces.

Note If you are using the browser interface, then use the HTTP POST method to pass the parameters. This reduces the time delay that the Web Dialer takes to automatically convert GET parameters to POST.

makeCall

You use the makeCall interface in customized directory search applications. The Unified CM directory search page (directory.asp) also uses this interface. Access the makeCall interface by initiating an HTTPS request to the URL https://<ipaddress>/webdialer/Webdialer. In this URL, ipaddress specifies the IP address of the Cisco Unified Communications Manager server where Web Dialer is configured.

Browser-based applications in which the browser accepts cookies use this interface. The user profile exists only for the length of the session if the cookies are disabled in a browser. For a sample script that is used to enable directory search pages, go to the "Sample Code Snippet" section on page 8-30.

Parameter	Mandatory	Description	Data Type	Range of Values	Default Value
destination	Mandatory	Destination number called by the application. Number gets converted to a regular telephone number by applying the application dial rules. Refer to the *Cisco Web Dialer* chapter in the *Cisco Unified Communications Manager Features and Services Guide, Release 5.0*.	String	None	None

Name	Description
result	Web Dialer displays the appropriate dialog and its applicable success or error message. It displays an authentication dialog if no active session exists.

makeCallProxy

You access the makeCallProxy interface by initiating an HTTPS request to the URL https://ipaddress/webdialer/Webdialer?cmd=doMakeCallProxy. Browser-based applications in which the browser accepts cookies use this interface. If the cookies are disabled in a browser, the user profile exists only for the length of the session.

Applications such as a personal address book, defined in the Unified CMUser pages at https://cmserver/CMUser, can use the makeCallProxy interface. The credential of the application is used as a proxy to make calls on behalf of users. Because these users have authenticated themselves before accessing the Unified CMUser window, they do not get prompted again for their user ID and password. The application sends the user ID and password of the proxy user in the form of a query string in the request or as a parameter in the body of the POST message.

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

For a sample script that is used to enable directory search pages, go to the "Sample Code Snippet" section on page 8-30.

Parameter	Mandatory	Description	Data Type	Range of Values	Default Value
uid	Mandatory	The user ID for which the request is made	String	None	None
appid	Mandatory	The userid of the application that is making a request on behalf of the user. For example, consider a Unified CM personal address book where the application allows authentication proxy rights. The appid parameter is used when the user logs in once; for example, in the Unified CM User windows. After this login, other pages do not require the user to log in again. For web page applications that are not integrated, the appid matches the userid.	String	None	None
pwd	Mandatory	The password of the appid	String	None	None
destination	Mandatory	The number to be called. The dial plan service converts this number to an E.164 number.	String	None	None

Name	Description
result	Web Dialer displays the appropriate dialog and its applicable success or error message.

Cisco Web Dialer WSDL

The WSDL specification provides the basis for the Web Service Definition Language (WSDL) for Web Dialer. You can access the WSDL for Web Dialer on the Web Dialer server installation at

https://<CCM_Server>/webdialer/wsdl/wd70.wsdl

Use this specific WSDL and the interfaces that are mentioned in this document to develop customized applications for Web Dialer. For a list of references on Cisco Unified Communications Manager, SOAP, and WSDL, refer to the Related Documentation section.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="urn:WD70"
xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="urn:WD70"
xmlns:intf="urn:WD70" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!--WSDL created by Apache Axis version: 1.4 With AXIS-2250
Built on Apr 22, 2006 (06:55:48 PDT)-->
 <wsdl:types>
  <schema targetNamespace="urn:WD70" xmlns="http://www.w3.org/2001/XMLSchema">
   <import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
```

```
<complexType name="Credential">
 <sequence>
  <element name="userID" type="xsd:string"/>
  <element name="password" type="xsd:string"/>
 </sequence>
</complexType>
<complexType name="UserProfile">
 <sequence>
  <element name="user" type="xsd:string"/>
  <element name="deviceName" type="xsd:string"/>
  <element name="lineNumber" type="xsd:string"/>
  <element name="supportEM" type="xsd:boolean"/>
  <element name="locale" type="xsd:string"/>
  <element name="dontAutoClose" type="xsd:boolean"/>
  <element name="dontShowCallConf" type="xsd:boolean"/>
 </sequence>
</complexType>
<complexType name="CallResponse">
 <sequence>
  <element name="responseCode" type="xsd:int"/>
  <element name="responseDescription" type="xsd:string"/>
 </sequence>
</complexType>
<complexType name="WDDeviceInfo">
 <sequence>
  <element name="deviceName" type="xsd:string"/>
  <element name="lines" type="impl:ArrayOf_soapenc_string"/>
 </sequence>
</complexType>
<complexType name="ArrayOfWDDeviceInfo">
 <complexContent>
  <restriction base="soapenc:Array">
   <attribute ref="soapenc:arrayType" wsdl:arrayType="impl:WDDeviceInfo[]"/>
  </restriction>
 </complexContent>
</complexType>
<complexType name="GetConfigResponse">
 <sequence>
  <element name="description" type="xsd:string"/>
  <element name="deviceInfoList" type="impl:ArrayOfWDDeviceInfo"/>
  <element name="responseCode" type="xsd:int"/>
 </sequence>
</complexType>
<complexType name="ArrayOf_soapenc_string">
 <complexContent>
  <restriction base="soapenc:Array">
   <attribute ref="soapenc:arrayType" wsdl:arrayType="soapenc:string[]"/>
  </restriction>
 </complexContent>
</complexType>
<complexType name="WDDeviceInfoDetail">
 <sequence>
  <element name="deviceName" nillable="true" type="soapenc:string"/>
  <element name="lines" nillable="true" type="impl:ArrayOf_soapenc_string"/>
  <element name="phoneDesc" nillable="true" type="soapenc:string"/>
  <element name="phoneType" nillable="true" type="soapenc:string"/>
 </sequence>
</complexType>
<complexType name="ArrayOfWDDeviceInfoDetail">
 <complexContent>
  <restriction base="soapenc:Array">
   <attribute ref="soapenc:arrayType" wsdl:arrayType="impl:WDDeviceInfoDetail[]"/>
  </restriction>
 </complexContent>
```

```
    </complexType>
    <complexType name="ConfigResponseDetail">
     <sequence>
      <element name="description" nillable="true" type="soapenc:string"/>
      <element name="deviceInfoListDetail" nillable="true"
type="impl:ArrayOfWDDeviceInfoDetail"/>
      <element name="responseCode" type="xsd:int"/>
     </sequence>
    </complexType>
   </schema>
  </wsdl:types>
   <wsdl:message name="getProfileDetailSoapResponse">
      <wsdl:part name="getProfileDetailSoapReturn" type="impl:ConfigResponseDetail"/>
    </wsdl:message>
    <wsdl:message name="getPrimaryLineResponse">
      <wsdl:part name="getPrimaryLineReturn" type="soapenc:string"/>
    </wsdl:message>
    <wsdl:message name="getPrimaryLineRequest">
      <wsdl:part name="in0" type="impl:Credential"/>
    </wsdl:message>
    <wsdl:message name="getProfileDetailSoapRequest">
      <wsdl:part name="in0" type="impl:Credential"/>
    </wsdl:message>
    <wsdl:message name="getProfileSoapRequest">
      <wsdl:part name="in0" type="impl:Credential"/>
      <wsdl:part name="in1" type="soapenc:string"/>
    </wsdl:message>
    <wsdl:message name="getProfileSoapResponse">
      <wsdl:part name="getProfileSoapReturn" type="impl:GetConfigResponse"/>
    </wsdl:message>
    <wsdl:message name="endCallSoapResponse">
      <wsdl:part name="endCallSoapReturn" type="impl:CallResponse"/>
    </wsdl:message>
    <wsdl:message name="makeCallSoapResponse">
      <wsdl:part name="makeCallSoapReturn" type="impl:CallResponse"/>
    </wsdl:message>
    <wsdl:message name="isClusterUserSoapResponse">
      <wsdl:part name="isClusterUserSoapReturn" type="xsd:boolean"/>
    </wsdl:message>
    <wsdl:message name="makeCallSoapRequest">
      <wsdl:part name="in0" type="impl:Credential"/>
      <wsdl:part name="in1" type="soapenc:string"/>
      <wsdl:part name="in2" type="impl:UserProfile"/>
    </wsdl:message>
    <wsdl:message name="endCallSoapRequest">
      <wsdl:part name="in0" type="impl:Credential"/>
      <wsdl:part name="in1" type="impl:UserProfile"/>
    </wsdl:message>
    <wsdl:message name="isClusterUserSoapRequest">
      <wsdl:part name="in0" type="soapenc:string"/>
    </wsdl:message>
    <wsdl:portType name="WDSoapInterface">
      <wsdl:operation name="makeCallSoap" parameterOrder="in0 in1 in2">
        <wsdl:input message="impl:makeCallSoapRequest" name="makeCallSoapRequest"/>
        <wsdl:output message="impl:makeCallSoapResponse" name="makeCallSoapResponse"/>
      </wsdl:operation>
      <wsdl:operation name="endCallSoap" parameterOrder="in0 in1">
        <wsdl:input message="impl:endCallSoapRequest" name="endCallSoapRequest"/>
        <wsdl:output message="impl:endCallSoapResponse" name="endCallSoapResponse"/>
      </wsdl:operation>
      <wsdl:operation name="getProfileSoap" parameterOrder="in0 in1">
        <wsdl:input message="impl:getProfileSoapRequest" name="getProfileSoapRequest"/>
        <wsdl:output message="impl:getProfileSoapResponse"
name="getProfileSoapResponse"/>
```

```
      </wsdl:operation>
      <wsdl:operation name="isClusterUserSoap" parameterOrder="in0">
         <wsdl:input message="impl:isClusterUserSoapRequest"
name="isClusterUserSoapRequest"/>
         <wsdl:output message="impl:isClusterUserSoapResponse"
name="isClusterUserSoapResponse"/>
      </wsdl:operation>
      <wsdl:operation name="getProfileDetailSoap" parameterOrder="in0">
         <wsdl:input message="impl:getProfileDetailSoapRequest"
name="getProfileDetailSoapRequest"/>
         <wsdl:output message="impl:getProfileDetailSoapResponse"
name="getProfileDetailSoapResponse"/>
      </wsdl:operation>
      <wsdl:operation name="getPrimaryLine" parameterOrder="in0">
         <wsdl:input message="impl:getPrimaryLineRequest" name="getPrimaryLineRequest"/>
         <wsdl:output message="impl:getPrimaryLineResponse"
name="getPrimaryLineResponse"/>
      </wsdl:operation>
   </wsdl:portType>
   <wsdl:binding name="WebdialerSoapServiceSoapBinding" type="impl:WDSoapInterface">
      <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="makeCallSoap">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="makeCallSoapRequest">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WD70" use="encoded"/>
         </wsdl:input>
         <wsdl:output name="makeCallSoapResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WD70" use="encoded"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="endCallSoap">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="endCallSoapRequest">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WD70" use="encoded"/>
         </wsdl:input>
         <wsdl:output name="endCallSoapResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WD70" use="encoded"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="getProfileSoap">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="getProfileSoapRequest">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WD70" use="encoded"/>
         </wsdl:input>
         <wsdl:output name="getProfileSoapResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WD70" use="encoded"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="isClusterUserSoap">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="isClusterUserSoapRequest">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WD70" use="encoded"/>
         </wsdl:input>
         <wsdl:output name="isClusterUserSoapResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WD70" use="encoded"/>
         </wsdl:output>
```

```
      </wsdl:operation>
      <wsdl:operation name="getProfileDetailSoap">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="getProfileDetailSoapRequest">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WD70" use="encoded"/>
         </wsdl:input>
         <wsdl:output name="getProfileDetailSoapResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WD70" use="encoded"/>
         </wsdl:output>
      </wsdl:operation>
      <wsdl:operation name="getPrimaryLine">
         <wsdlsoap:operation soapAction=""/>
         <wsdl:input name="getPrimaryLineRequest">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WD70" use="encoded"/>
         </wsdl:input>
         <wsdl:output name="getPrimaryLineResponse">
            <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:WD70" use="encoded"/>
         </wsdl:output>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:service name="WDSoapInterfaceService">
      <wsdl:port binding="impl:WebdialerSoapServiceSoapBinding"
name="WebdialerSoapService">
         <wsdlsoap:address
location="https://localhost/webdialer/services/WebdialerSoapService70"/>
      </wsdl:port>
   </wsdl:service>
</wsdl:definitions>
```

Sample Code Snippet

This sample code snippet enables Web Dialer from a directory search page.

Single-Cluster Applications

Use this snippet for single-cluster applications if all users are in only one cluster.

```
f<FORM action="https://42.88.86.1/webdialer/Webdialer" method="post">
    <P>
    <INPUT type="hidden" name="destination" value="+666">
    <INPUT type="submit" value="Send">
    </P>
 </FORM>
```

Multiple-Cluster Applications

Use this snippet if all users are spread across different clusters.

```
function launchWebDialerWindow( url ) {
    webdialer=window.open( url, "webdialer", "status=no, width=420, height=300,
scrollbars=no, resizable=yes, toolbar=no" );
  }

  function launchWebDialerServlet( destination ) {
    url= 'https://<%=server_name%>/webdialer/Redirector?destination='+escape(destination);
    launchWebDialerWindow( url );
  }
```

```
!These functions can be called from the HTML page which has a hyperlink to the phone
number to be called. An example of it is
<TD><A href="javascript:launchWebDialerServlet( <%= userInfo.TelephoneNumber %> )"><%=
userInfo.TelephoneNumber %></A> </TD>
```