



Cisco WebDialer API Reference Guide

This guide describes the Simple Object Access Protocol (SOAP) and HTML over HTTP interfaces that are used to develop customized directory search applications for Cisco WebDialer Version 1.0(3), and contains the following sections:

- [Audience, page 1](#)
- [Definitions, page 2](#)
- [Overview, page 2](#)
- [Call Flows, page 4](#)
- [Interfaces, page 8](#)
- [Cisco WebDialer WSDL, page 16](#)
- [Sample JavaScript, page 18](#)
- [Related Documentation, page 19](#)

Audience

This guide is intended for software developers who are using Cisco WebDialer to develop client applications compatible with the Cisco CallManager IP Telephony system.

This guide assumes that the developer is familiar with the following technologies and languages:

- Simple Object Access Protocol (SOAP)
- Extensible Markup Language (XML)
- Hypertext Transfer Protocol (HTTP)
- Hypertext Markup Language (HTML)
- Web Service Definition Language (WSDL)

The developer must also have an understanding of Cisco CallManager and its applications. Documents on Cisco CallManager and other related technologies are listed in the [Related Documentation](#) section.

Definitions

Cisco WebDialer Server	A Cisco CallManager server that hosts the Cisco WebDialer application
Cisco WebDialer Application	The software that is installed on a Cisco CallManager server. It enables the click-to-dial functionality by creating hyperlinked telephone numbers in a company directory. This functionality allows users to make calls from the web page by clicking on the telephone number of the person they are trying to call. The Cisco WebDialer application consists of two Java servlets, the WebDialer servlet and the Redirector servlet.
WebDialer Servlet	A Java servlet that allows Cisco CallManager users in a specific cluster to make and end calls, as well as access their phone and line configuration.
Redirector Servlet	A Java servlet that finds the Cisco CallManager cluster for a request made by a Cisco WebDialer user. It redirects that request to the specific Cisco WebDialer server located in that user's Cisco CallManager cluster.

Overview

Cisco WebDialer, which is installed on a Cisco CallManager server and used in conjunction with Cisco CallManager, allows Cisco IP Phone users to make calls from web and desktop applications. For example, Cisco WebDialer uses hyperlinked telephone numbers in a company directory to allow users to make calls from a web page by clicking on the telephone number of the person that they are trying to call.

The two main components of Cisco WebDialer are the Webdialer and Redirector servlets.

Webdialer Servlet

The WebDialer servlet is a Java servlet that allows Cisco CallManager users in a specific cluster to make and end calls, as well as to access their phone and line configuration.

Cisco WebDialer applications interact with the WebDialer servlet through two interfaces:

- The SOAP over HTTP interface—This interface that is based on the Simple Object Access Protocol (SOAP) gets used to develop desktop applications such as Microsoft Outlook Add-in and SameTime Client Plug-in. Developers can use the `isClusterUserSoap` interface to design multicluster applications that require functionality similar to a Redirector servlet.

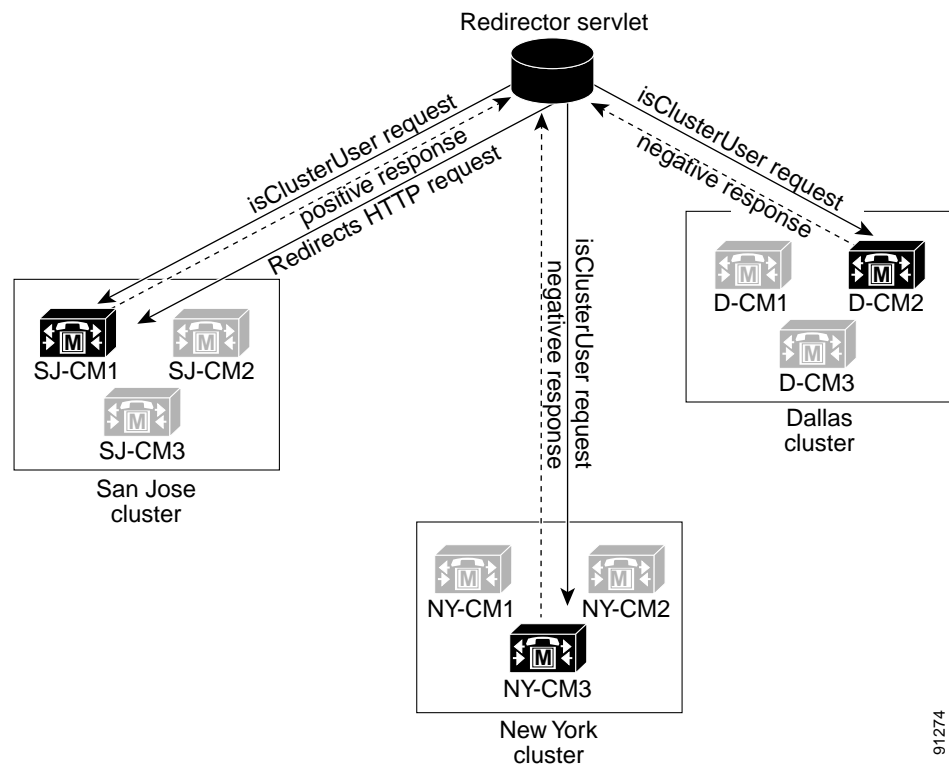
- HTML over HTTP interface—This interface that is based on the HTTP protocol gets used to develop web-based applications such as the Cisco CallManager directory search page (directory.asp). Developers who use this interface can use the Redirector servlet for designing multicenter applications.

Redirector Servlet

The Redirector servlet, a Java based Tomcat servlet finds the Cisco CallManager cluster for a request made by a Cisco WebDialer user. It redirects that request to the specific Cisco WebDialer server that is located in that user's Cisco CallManager cluster. Availability of the Redirector servlet occurs only for multicenter applications and only for applications that are developed by using HTML over HTTP interfaces.

Figure 1 illustrates how a Redirector servlet redirects a call in a multicenter environment.

Figure 1 Multiple Clusters



91274

Example of Cisco WebDialer Using the Redirector Servlet

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

The system administrator configures the Webdialer servlets on any of the Cisco CallManager servers by entering the IP address of that specific Cisco CallManager server in the *wdservers* service parameter.

For information on configuring Webdialer and Redirector servlets, refer to the *Cisco WebDialer* chapter in the *Cisco CallManager Features and Services Guide, Release 4.0(1)*.

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

1. The Cisco CallManager server sends an initial *makeCall* HTTP request to the Redirector servlet.
2. If this request is received for the first time, the Redirector servlet reads the Cisco WebDialer server cookie and finds it empty.

For a repeat request, the Redirector servlet reads the IP address of the Cisco WebDialer server that previously serviced the client and sends a *isClusterUser* HTTP request only to that server.

3. The Redirector servlet sends back a response asking for information, which results in the authentication dialog box opening for the user.
4. The user enters the Cisco CallManager 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* HTTP request to each Cisco WebDialer server that has been configured by the system administrator.

[Figure 1](#) illustrates how this request is sent to the Webdialer servlets that have been configured for SJ-CM1, D-CM2, and NY-CM3. Depending on the geographical location of the calling party, the WebDialer servlet from the cluster representing that location responds positively to the Redirector servlet. The remaining Webdialer servlets that were contacted return a negative response. The WebDialer 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's browser for future use.

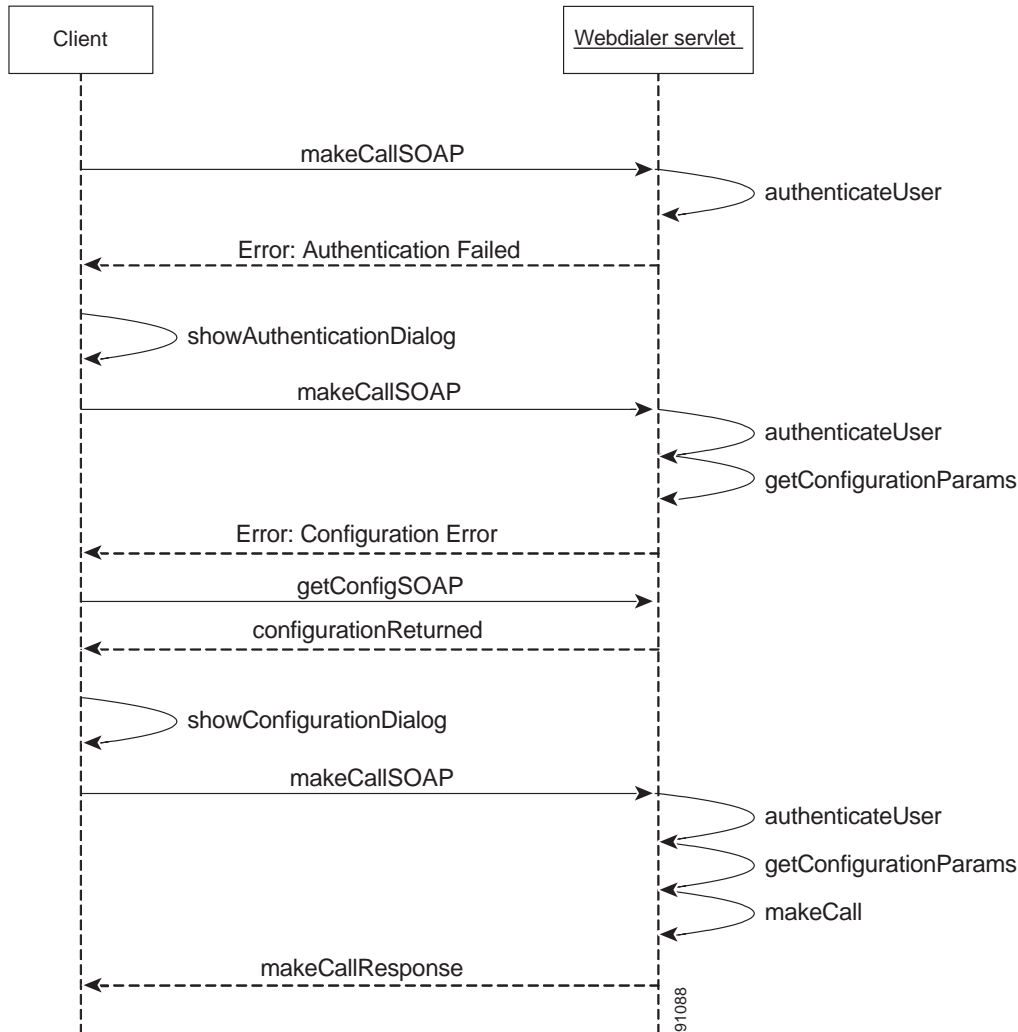
Call Flows

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

Desktop-based Client Application Call Flow

[Figure 2](#) shows the call flow for an outgoing call from a client application such as Microsoft Outlook Plug-in to a WebDialer servlet. 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 2 Cisco WebDialer Call Flow for a Client-based Application



If the user makes a call for the first time:

1. The client sends a makeCallSoap request to the configured WebDialer servlet.
2. The WebDialer servlet attempts to authenticate the user. [Figure 2](#) shows an authentication failure because the authentication information is incomplete or does not exist.
3. The WebDialer servlet sends an authentication failure response to the client application.
4. The client application displays a dialog box on the computer screen of the user asking for the user ID and password. The user enters this information and clicks the **submit** button. The user ID and password is now stored for future invocations of the application.
5. The application sends a repeat SOAP request to the WebDialer servlet. The request contains credential information on the user.
6. The WebDialer servlet authenticates the user.
7. The WebDialer servlet reads any missing configuration information in the request.
8. The WebDialer servlet returns a configuration error message to the client application.

9. The client application sends a getConfigSoap request to the WebDialer servlet.
10. The WebDialer servlet responds with the user's configuration information stored in the directory.
11. The client application displays a configuration dialog box on the user's computer screen asking the user to select or update the configuration. The user enters the information and clicks the **submit** button. The user's configuration information is now stored for future invocations of the application.
12. The client resends the makeCallSoap request to the WebDialer servlet. This request contains the user's configuration information.
13. The WebDialer servlet authenticates the user and dials the telephone number using the information contained in the makeCallSoap request. It responds back to the client with a success or failure message.

**Note**

The call flow goes directly to step twelve:

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

Browser-based Application Call Flow

Figure 3 shows the call flow for an HTTP based browser application such as a directory search page, personal address book, or the Cisco CallManager directory search page (directory.asp) which is bundled with Cisco CallManager.

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.

If the user makes a call for the first time:

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

**Note**

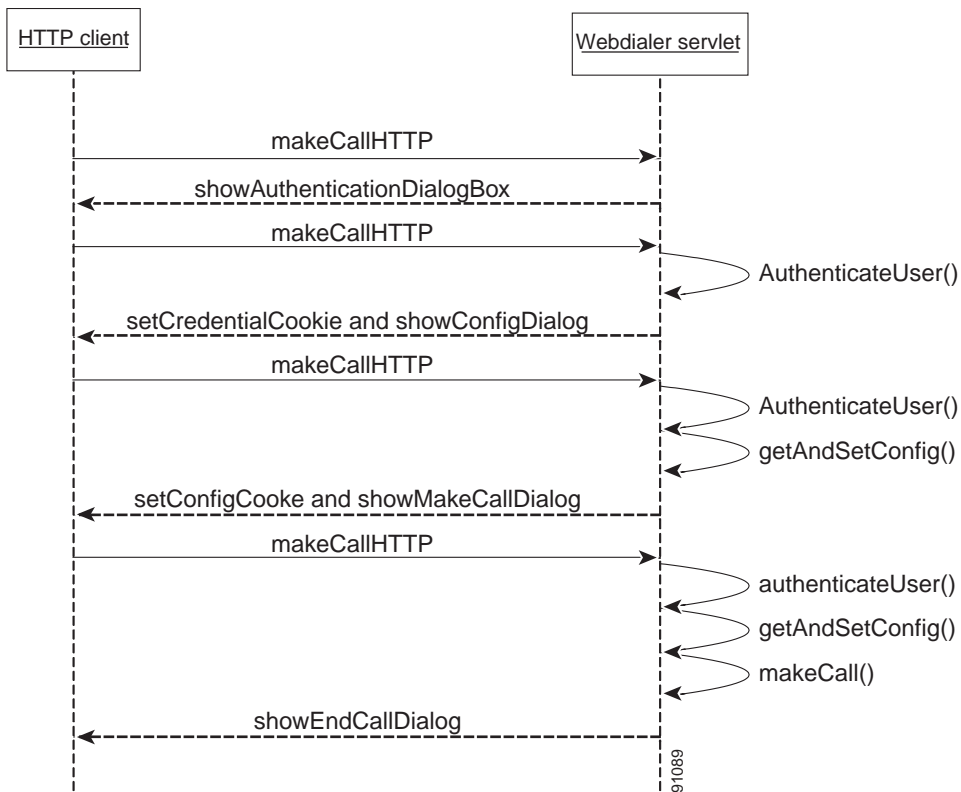
Authentication is successful if the user's credentials are sent with the request, and the call flow goes directly to number seven.

3. The WebDialer 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 makeCallHTTP request containing the user's credentials to the WebDialer servlet.
6. The WebDialer servlet authenticates the user.
7. The WebDialer servlet reads the configuration information in the cookie which is sent with the request.
8. Assuming that the request is made for the first time, the servlet sends a response containing a cookie to the client's browser. The cookie containing the client's credentials is stored on the client's browser. The client's credentials are 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's browser sends a makeCall HTTP request to the WebDialer servlet. The request contains a cookie with the credential and configuration information in parameter form.
11. The WebDialer servlet uses the credentials to authenticate the user and saves the configuration information in its memory.
12. The WebDialer servlet sends a makeCall confirmation dialog to the client's browser with the configuration information stored in a cookie. The cookie is stored on the client's browser for future invocations.
13. The Make Call dialog box appears on the user's computer screen. The user clicks the **Dial** button which sends another makeCall HTTP request to the WebDialer servlet.
14. The WebDialer servlet authenticates the user 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's computer screen and stays there for the amount of time configured in the service parameters.

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

Figure 3 Cisco WebDialer Call Flow for a Browser-based Application



Interfaces

Cisco WebDialer applications interact with the WebDialer servlet through two interfaces:

- The SOAP over HTTP interface— This interface is based on the Simple Object Access Protocol (SOAP) and is used to develop desktop applications such as Microsoft Outlook Add-in and SameTime Client Plug-in. Developers can use the `isClusterUserSoap` interface to design multi-cluster applications that require functionality similar to a Redirector servlet.
- HTML over HTTP interface— This interface is based on the HTTP protocol and is used to develop web-based applications such as the Cisco CallManager directory search page (`directory.asp`). Developers using this interface can use the Redirector servlet for designing multi-cluster applications.

SOAP over HTTP Interface

To access the SOAP interfaces for Cisco WebDialer, use the Cisco WebDialer Web Service Definition Language (WSDL) in the [“Cisco WebDialer WSDL” section on page 16](#).

makeCallSoap

The `makeCallSoap` interface is accessed by initiating a SOAP request to the URL `http://ipaddress/wdsoap/servlet/rpcrouter` for the service name `WebdialerSoap`. In this URL, `ipaddress` is the IP address of the Cisco CallManager server. This SOAP interface is used by plug-in clients such as Microsoft Outlook Plug-in.

Parameters

Table 1 *makeCallSoap Parameters*

Parameter	Mandatory	Description	Data Type	Range of Values	Default Value
Destination	Mandatory	Standard canonical form. For example +1 408 5551212 or extensions such as 2222.	String	None	None
Credential	Mandatory	The user ID or password of the user or proxy user. For more information on creating a proxy user, refer to the <i>Cisco WebDialer</i> chapter in the <i>Cisco CallManager Features and Services Guide, Release 4.0(1)</i> .	Refer to the credential data type in the “Cisco WebDialer WSDL” section on page 16 .	None	None
Profile	Mandatory	The profile that is used to make a call. An example of a typical profile is a calling device such as an IP phone or line.	Refer to the profile data type in the “Cisco WebDialer WSDL” section on page 16 .	None	None

Results

Refer to the “[Cisco WebDialer WSDL](#)” section on page 16 for return values and their data type.

Table 2 *makeCallSoap Results*

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 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	
5	responseCode	Integer	No device is configured for the user, or, there are missing parameters in the request.	The application initiates a getConfigSOAP request and displays the selected device and line to the user.
	responseDescription	String	No device is configured for the user, or, there are missing parameters in the request.	
6	responseCode	Integer	Service is temporarily unavailable.	Displays the appropriate error dialog with an option to try again.
	responseDescription	String	Service is 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	

Table 2 *makeCallSoap Results (continued)*

Error Code	Name	Type	Description	Action by application
9	responseCode	Integer	Service overloaded	Displays the appropriate error dialog with an option to try again.
	responseDescription	String	Service overloaded	

endCallSoap

The endCallSoap interface is accessed by initiating a SOAP request to the URL `http://ipaddress/wdsoap/servlet/rpcrouter` for the service name *WebdialerSoap*. In this URL, ipaddress is the IP address of the Cisco CallManager server on which Cisco WebDialer is configured.

This SOAP interface is used by plug-in based clients such as Microsoft Outlook Add-in.

Parameters

Table 3 *endCallSoap Parameters*

Parameter	Mandatory	Description	Data Type	Range of Values	Default Value
Credential	Mandatory	The user ID or password of the user or proxy user. For information on creating a proxy user, refer to the <i>Cisco WebDialer</i> chapter in the <i>Cisco CallManager Features and Services Guide, Release 4.0(1)</i> .	Refer to the credential data type in the “Cisco WebDialer WSDL” section on page 16.	None	None
Profile	Mandatory	The profile that is used to make a call. An example of a typical profile is a calling device such as an IP phone or line.	Refer to the profile data type in the “Cisco WebDialer WSDL” section on page 16.	None	None

Results

Refer to the [“Cisco WebDialer WSDL”](#) section on page 16 for return values and their data type.

Table 4 *endCallSoap Results*

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.

Table 4 *endCallSoap Results (continued)*

Error Code	Name	Type	Description	Action by application
	responseDescription	String	Call failure error	
2	responseCode	Integer	Authentication error	Displays the authentication dialog for the user to enter their 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	
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, there are missing parameters in the request.	The application initiates a getConfigSOAP request and displays the selected device and line to the user.
	responseDescription	String	No device is configured for the user, or, there are missing parameters in the request.	
6	responseCode	Integer	Service is temporarily unavailable.	Displays the appropriate error dialog with an option to try again.
	responseDescription	String	Service is 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	

getProfileSoap

The getProfileSoap interface is accessed by initiating a SOAP request to the URL `http://ipaddress/wdsoap/servlet/rpcrouter` for the service name *WebdialerSoap*. In this URL, ipaddress is the IP address of the Cisco CallManager server on which Cisco WebDialer is configured.

This SOAP interface is used by plug-in based clients such as Microsoft Outlook Add-in.

Parameters

Table 5 *getProfileSoap Parameters*

Parameter	Mandatory	Description	Data Type	Range of Values	Default Value
Credential	Mandatory	The user ID or password of the user or proxy user. For information on creating a proxy user, refer to the <i>Cisco WebDialer</i> chapter in the <i>Cisco CallManager Features and Services Guide, Release 4.0(1)</i> .	Refer to the credential data type in the “Cisco WebDialer WSDL” section on page 16.	None	None
UserID	Mandatory	The user ID for which the configuration is requested.	String	None	None

Results

Refer to the [“Cisco WebDialer WSDL”](#) section on page 16 for return values and their data type.

Table 6 *get ProfileSoap Results*

Error Code	Name	Type	Description	Action by plug-in application
0	responseCode	Integer	Returns an array of phones or lines on the phone associated with the user. Refer to the <i>Cisco WebDialer WSDL</i> for the <i>WDDeviceInfo</i> data type.	Displays a dialog box on the computer screen.
	responseDescription	String	Success	
	deviceInfoList	Array	Returns an array of the <i>WDDeviceInfo</i> data type.	
1	responseCode	Integer	No device is configured for the user.	Displays an appropriate error message.
	responseDescription	String	No device is 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	

Table 6 *get ProfileSoap Results (continued)*

Error Code	Name	Type	Description	Action by plug-in 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 is temporarily unavailable.	Displays the appropriate error dialog with an option to try again.
	responseDescription	String	Service is temporarily unavailable.	
9	responseCode	Integer	Service is overloaded.	Displays the appropriate error dialog with an option to try again.
	responseDescription	String	Service is overloaded.	

isClusterUserSoap

The isClusterUserSoap interface is accessed by initiating a SOAP request to the URL `http://ipaddress/wdsoap/servlet/rpcrouter` for the service name *WebdialerSoap*. In this URL, ipaddress is the IP address of the Cisco CallManager server on which Cisco WebDialer is configured.

This SOAP interface is used by plug-in based clients such as Microsoft Outlook Add-in. It is used for multi-cluster applications that require functionality, similar to a Redirector servlet, for redirecting calls to the various locations where Cisco WebDialer is installed on a network. The application uses this interface to locate and verify the Cisco WebDialer servicing the user, followed by makeCall, endCall or getProfile requests to that Cisco WebDialer.

Parameters

Table 7 *isClusterSoap Parameters*

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

Results

Refer to the [“Cisco WebDialer WSDL”](#) section on page 16 for return values and their data type.

Table 8 *isClusterSoap Results*

Name	Type	Description
result	Boolean	The result is true if the user is present in the directory of the cluster. The result is false if the user is not present in the directory of the cluster.

HTML over HTTP Interface

This section describes the HTML over HTTP interfaces.

makeCall

The makeCall interface is used in customized directory search applications. It is also used by the directory search page (directory.asp) that is bundled with Cisco CallManager. The makeCall interface is accessed by initiating an HTTP request to the URL `http://ipaddress/webdialer/Webdialer`. In this URL, ipaddress is the IP address of the Cisco CallManager server for which Cisco WebDialer is configured.

This interface is used by browser-based applications in which the browser accepts cookies. 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 JavaScript” section on page 18](#).

Parameters

Table 9 *makeCall Parameters*

Parameter	Mandatory	Description	Data Type	Range of Values	Default Value
destination	Mandatory	The destination number called by the application. This number is converted to a regular telephone number by applying the application dial rules. For information on dial rules, refer to the <i>Cisco WebDialer</i> chapter in the <i>Cisco CallManager Features and Services Guide, Release 4.0(1)</i> .	String	None	None

Results

Table 10 *makeCall Results*

Name	Description
result	Cisco WebDialer displays the appropriate dialog and its applicable success or error message. It displays an authentication dialog if there is no active session.

makeCallProxy

The makeCallProxy interface is accessed by initiating an HTTP request to the URL `http://ipaddress/webdialer/Webdialer?cmd=doMakeCallProxy`. This interface is used by browser-based applications in which the browser accepts cookies. If the cookies are disabled in a browser, the user profile exists for only the length of the session.

The makeCallProxy interface can be used by applications such as a personal address book, defined in the CCMUser pages at `http://cmserver/CCMUser`. The credential of the application is used, as a proxy, to make calls on behalf of users. Since these users have authenticated themselves before accessing the CCMUser page, they are not 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.

For a sample script that is used to enable directory search pages, go to the [“Sample JavaScript” section on page 18](#).

Parameters

Table 11 *makeCallProxy Parameters*

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 making a request on behalf of the user. For example, consider a Cisco CallManager 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 CCMUser pages. After this login, other pages do not require the user to log in again. For web page applications that are not integrated, appid is the same as userid.	String	None	None
pwd	Mandatory	The password of the appid.	String	None	None
destination	Mandatory	The number to be called. This number is converted to an E.164 number by the dial plan service.	String	None	None

Results

Table 12 *makeCallProxy Results*

Name	Description
result	Cisco WebDialer displays the appropriate dialog and its applicable success or error message.

Cisco WebDialer WSDL

The Web Service Definition Language (WSDL) for Cisco WebDialer mentioned below is based on the WSDL specification. This WSDL for Cisco WebDialer is also available on the Cisco WebDialer server installation at:

http://<WebDialer_server_name>/webdialer/wsd/WebdialerSoap.wsdl

Use this specific WSDL and the interfaces mentioned in this document to develop customized applications for Cisco WebDialer. For a list of references on Cisco CallManager, SOAP, and WSDL, refer to the [Related Documentation](#) section.

```
<wSDL:definitions xmlns:tns="urn:WebdialerSoap"
xmlns:soap="http://schemas.xmlsoap.org/wsd/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsd/http/"
xmlns:mime="http://schemas.xmlsoap.org/wsd/mime/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsd="http://schemas.xmlsoap.org/wsd/" xmlns="http://schemas.xmlsoap.org/wsd/"
targetNamespace="urn:WebdialerSoap" name="urn:WebdialerSoap">
  <wsd:types>
    <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:tns="urn:WebdialerSoap" targetNamespace="urn:WebdialerSoap">
      <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <xsd:complexType name="CallResponse">
        <xsd:sequence>
          <xsd:element name="responseCode" type="xsd:int" />
          <xsd:element name="description" nillable="true" type="xsd:string" />
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="Credential">
        <xsd:sequence>
          <xsd:element name="userID" nillable="true" type="xsd:string" />
          <xsd:element name="password" nillable="true" type="xsd:string" />
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="UserProfile">
        <xsd:sequence>
          <xsd:element name="user" nillable="true" type="xsd:string" />
          <xsd:element name="deviceName" nillable="true" type="xsd:string" />
          <xsd:element name="lineNumber" nillable="true" type="xsd:string" />
          <xsd:element name="supportEM" type="xsd:boolean" />
          <xsd:element name="locale" nillable="true" type="xsd:string" />
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="GetConfigResponse">
        <xsd:sequence>
          <xsd:element name="responseCode" type="xsd:int" />
          <xsd:element name="description" nillable="true" type="xsd:string" />
          <xsd:element name="deviceInfoList" nillable="true"
type="tns:ArrayOfWDDeviceInfo" />
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="WDDeviceInfo">
        <xsd:sequence>
          <xsd:element name="deviceName" nillable="true" type="xsd:string" />
          <xsd:element name="lines" nillable="true" type="tns:ArrayOfstring" />
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="ArrayOfWDDeviceInfo">
        <xsd:complexContent>
          <xsd:restriction base="soapenc:Array">
```

```

        <xsd:attribute ref="soapenc:arrayType"
wsdl:arrayType="tns:WDDeviceInfo[]" />
        </xsd:restriction>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ArrayOfstring">
    <xsd:complexContent>
        <xsd:restriction base="soapenc:Array">
            <xsd:attribute ref="soapenc:arrayType"
wsdl:arrayType="xsd:string[]" />
            </xsd:restriction>
        </xsd:complexContent>
    </xsd:complexType>
</xsd:schema>
</wsdl:types>
<wsdl:message name="makeCallSoap0In">
    <wsdl:part name="cred" type="tns:Credential" />
    <wsdl:part name="dest" type="xsd:string" />
    <wsdl:part name="prof" type="tns:UserProfile" />
</wsdl:message>
<wsdl:message name="makeCallSoap0Out">
    <wsdl:part name="Result" type="tns:CallResponse" />
</wsdl:message>
<wsdl:message name="endCallSoap1In">
    <wsdl:part name="cred" type="tns:Credential" />
    <wsdl:part name="prof" type="tns:UserProfile" />
</wsdl:message>
<wsdl:message name="endCallSoap1Out">
    <wsdl:part name="Result" type="tns:CallResponse" />
</wsdl:message>
<wsdl:message name="getProfileSoap2In">
    <wsdl:part name="cred" type="tns:Credential" />
    <wsdl:part name="userid" type="xsd:string" />
</wsdl:message>
<wsdl:message name="getProfileSoap2Out">
    <wsdl:part name="Result" type="tns:GetConfigResponse" />
</wsdl:message>
<wsdl:message name="isClusterUser3In">
    <wsdl:part name="userid" type="xsd:string" />
</wsdl:message>
<wsdl:message name="isClusterUser2Out">
    <wsdl:part name="Result" type="xsd:boolean" />
</wsdl:message>
<portType name="WebdialerSoapService">
    <wsdl:operation name="makeCallSoap">
        <wsdl:input message="tns:makeCallSoap0In" />
        <wsdl:output message="tns:makeCallSoap0Out" />
    </wsdl:operation>
    <wsdl:operation name="endCallSoap">
        <wsdl:input message="tns:endCallSoap1In" />
        <wsdl:output message="tns:endCallSoap1Out" />
    </wsdl:operation>
    <wsdl:operation name="getProfileSoap">
        <wsdl:input message="tns:getProfileSoap2In" />
        <wsdl:output message="tns:getProfileSoap2Out" />
    </wsdl:operation>
    <wsdl:operation name="isClusterUserSoap">
        <wsdl:input message="tns:isClusterUser3In" />
        <wsdl:output message="tns:isClusterUser2Out" />
    </wsdl:operation>
</portType>
<binding name="WebdialerSoapService" type="tns:WebdialerSoapService">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="makeCallSoap">

```

```

        <soap:operation soapAction="urn:makeCallSoap"/>
        <input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:WebdialerSoap"/>
        </input>
        <output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:WebdialerSoap"/>
        </output>
    </wsdl:operation>
    <wsdl:operation name="endCallSoap">
        <soap:operation soapAction="urn:endCallSoap"/>
        <input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:WebdialerSoap"/>
        </input>
        <output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:WebdialerSoap"/>
        </output>
    </wsdl:operation>
    <wsdl:operation name="getProfileSoap">
        <soap:operation soapAction="urn:getProfileSoap"/>
        <input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:WebdialerSoap"/>
        </input>
        <output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:WebdialerSoap"/>
        </output>
    </wsdl:operation>
    <wsdl:operation name="isClusterUserSoap">
        <soap:operation soapAction="urn:isClusterUserSoap"/>
        <input>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:WebdialerSoap"/>
        </input>
        <output>
            <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:WebdialerSoap"/>
        </output>
    </wsdl:operation>
</binding>
<service name="WebdialerSoap">
    <port name="WebdialerSoapService" binding="tns:WebdialerSoapService">
        <soap:address location="http://WebDialer_server_name/wdsoap/servlet/rpcrouter"/>
    </port>
</service>
</wsdl:definitions>

```

Sample JavaScript

This JavaScript is a sample script that is used to enable Cisco WebDialer from a directory search page.

Single Cluster Applications

This script is used for single cluster applications if all users are in only one cluster.

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

function launchWebDialerServlet( destination ) {
    url = 'http://<%=server_name%>/webdialer/Webdialer?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>&nbsp;</TD>
```

Multiple Cluster Applications

This script is used 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= 'http://<%=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>&nbsp;</TD>
```

Related Documentation

This section lists documents and URLs that provide information on Cisco CallManager, Cisco IP Phones, and the technologies that are required to develop applications with Cisco WebDialer.

Related Topics

- Cisco WebDialer Administration— *Cisco WebDialer* chapter in the *Cisco CallManager Features and Services Guide, Release 4.0(1)*.
- Service Parameters Configuration— *Cisco CallManager Administration Guide, Release 4.0(1)*.
- Cisco IP Phone Configuration— *Cisco CallManager Administration Guide, Release 4.0(1)*.
- Application Dial Rules Configuration— *Application Dial Rules* chapter in the *Cisco CallManager Administration Guide, Release 4.0(1)*.

Additional Cisco Documentation

- *Cisco CallManager Release 4.0(1)*—A suite of documents related to the installation and configuration of Cisco CallManager. Refer to the *Cisco CallManager Documentation Guide for Release 4.0(1)* for a list of documents on installing and configuring Cisco CallManager 4.0(1).
- *Cisco IP Phones and Services*—A suite of documents related to the installation and configuration of Cisco IP Phones.

- *Cisco DistributedDirector*—A suite of documents that are related to the installation and configuration of Cisco DistributedDirector.
- *Simple Object Access Protocol (SOAP) 1.1*
- *Web Service Definition Language (WSDL) 1.1*
- *SOAP Tutorial*
- *WSDL Tutorial*—Web Service Definition Language tutorial.
- <http://www.soapagent.com/>—An open SOAP directory with links to articles, tutorials, and white papers.