



Cisco Web Dialer API Reference Guide, Release 4.2(3)

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 Web Dialer Version 1.0(3). It contains the following sections:

- [Audience, page 2](#)
- [Definitions, page 2](#)
- [Overview, page 2](#)
- [Call Flows, page 5](#)
- [Interfaces, page 9](#)
- [Cisco Web Dialer WSDL, page 18](#)
- [Sample JavaScript, page 21](#)
- [Related Documentation, page 21](#)
- [Obtaining Documentation, page 22](#)
- [Documentation Feedback, page 23](#)
- [Cisco Product Security Overview, page 23](#)
- [Obtaining Technical Assistance, page 24](#)
- [Obtaining Additional Publications and Information, page 26](#)

Audience

This guide targets software developers who are using Cisco Web Dialer to develop client applications that are compatible with the Cisco Unified 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 understand Cisco Unified CallManager and its applications. The [“Related Documentation” section on page 21](#) lists documents about Cisco Unified CallManager and other related technologies.

Definitions

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

Overview

Cisco Web Dialer, which is installed on a Cisco Unified CallManager server and used with Cisco Unified CallManager, allows Cisco Unified IP Phone users to make calls from web and desktop applications. For example, Cisco Web Dialer 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 Web Dialer and Redirector servlets comprise the two main components of Cisco Web Dialer.

Web Dialer Servlet

The Web Dialer servlet, a Java servlet, allows Cisco Unified CallManager 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 HTTP—This interface, 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—This interface, based on the HTTP protocol, gets used to develop web-based applications such as the Cisco Unified CallManager directory search page (`directory.asp`). Developers who use this interface can use the Redirector servlet for designing multicluster applications.

**Note**

Beginning with Cisco Unified CallManager 5.0, the Web Dialer servlet will require that the client application use the HTTPS protocol. Cisco recommends using HTTPS when using the Web Dialer servlet to avoid future compatibility issues.

Redirector Servlet

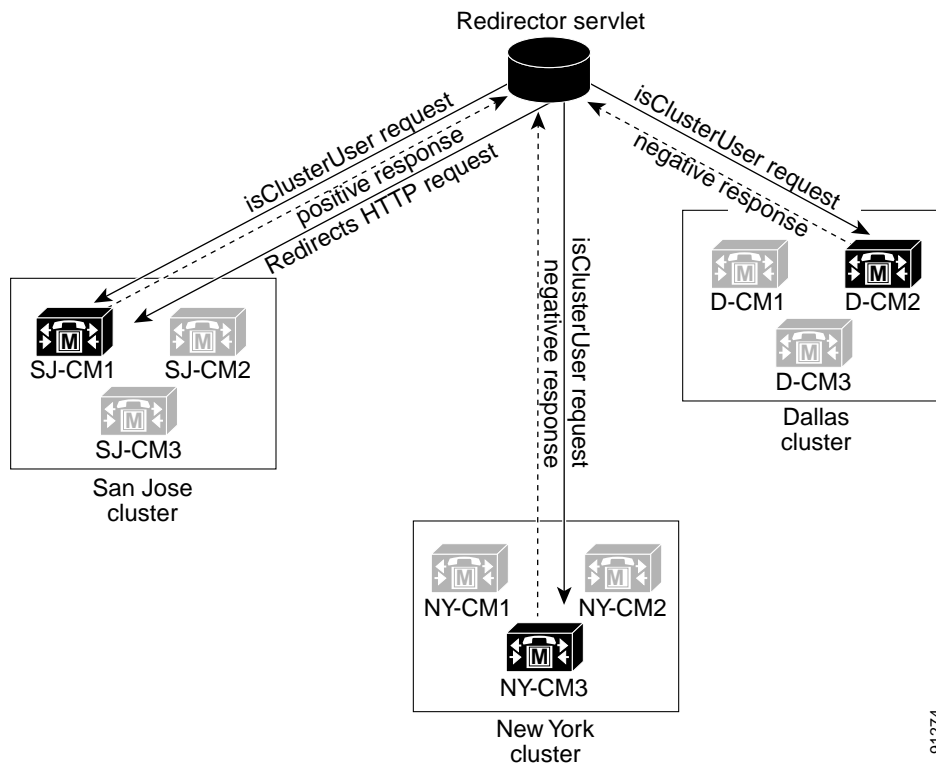
The Redirector servlet, a Java based Tomcat servlet, finds the Cisco Unified CallManager cluster for a request that a Cisco Web Dialer user makes. It redirects that request to the specific Cisco Web Dialer server that is located in the Cisco Unified CallManager cluster of that user. Availability of the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

**Note**

Beginning with Cisco Unified CallManager 5.0, the Redirector servlet will require the client application to use the HTTPS protocol. Cisco recommends using HTTPS when using the Redirector servlet to avoid future compatibility issues.

[Figure 1](#) illustrates how a Redirector servlet redirects a call in a multicluster environment.

Figure 1 Multiple Clusters



91274

Example of Cisco Web Dialer 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 Unified CallManager servers with Web Dialer servlets that have been configured for Cisco Unified CallManager servers SJ-CM1, D-CM2, and NY-CM3.

The system administrator configures the Web Dialer servlets on a Cisco Unified CallManager server by entering the IP address of that Cisco Unified CallManager server in the *wdservers* service parameter.

For information on configuring Web Dialer and Redirector servlets, refer to the *Cisco Web Dialer* chapter in the *Cisco Unified CallManager Features and Services Guide, Release 4.2(3)*.

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

1. The Cisco Unified 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 Web Dialer server cookie and finds it empty.
For a repeat request, the Redirector servlet reads the IP address of the Cisco Web Dialer 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 Unified 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 Web Dialer server that the system administrator has configured.

Figure 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 the cluster that represents that location 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.

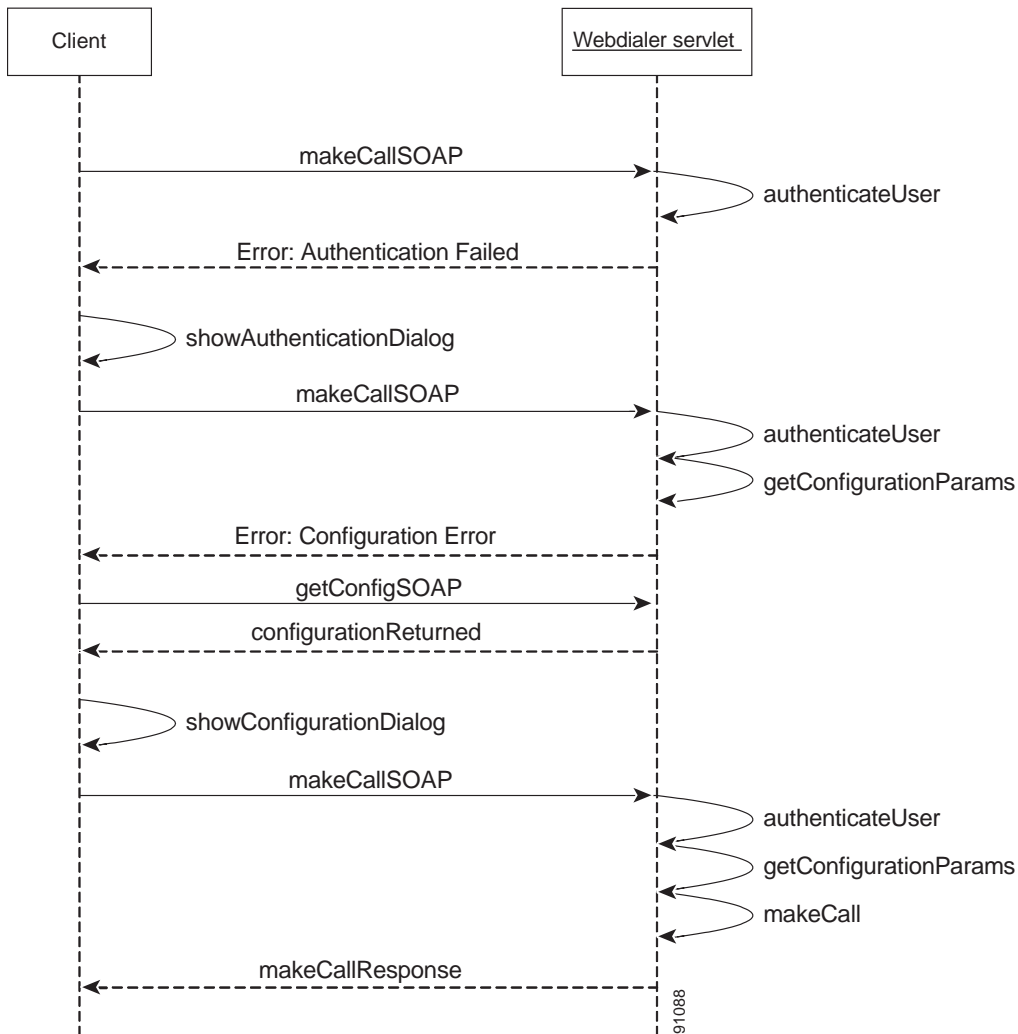
Call Flows

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

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 Web Dialer 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 Web Dialer Call Flow for a Client-based Application



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 2](#) shows an authentication failure 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 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 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 getConfigSoap 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 on the user computer screen asking the user to select or update the configuration. The user enters the information and clicks the **Submit** button. The user configuration information now gets stored for future invocations of the application.
12. The client re-sends 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 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 12:

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

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 Unified CallManager directory search page (directory.asp) which is bundled with Cisco Unified 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 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**

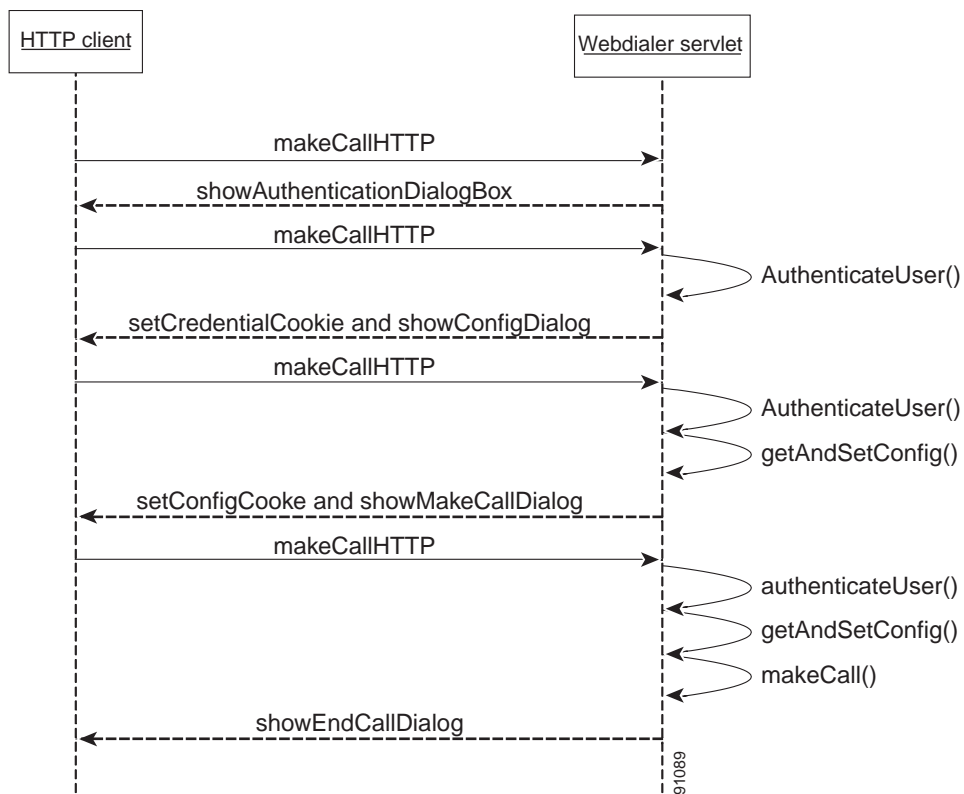
Authentication succeeds if the user credentials are sent with the request, and the call flow goes directly to Step 7.

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 makeCallHTTP 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 include 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 HTTP 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 stored in a cookie. The cookie gets stored on the client browser for future invocations.
13. The Make Call dialog box displays on the user's computer screen. The user clicks the **Dial** button, which sends another makeCall HTTP 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 timespan that is configured in the service parameters.

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

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



Interfaces

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

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



Note

The WebDialer servlet enforces a transaction call rate that varies by server class. Each WebDialer servlet running on an MCS-7825 or MCS-7835 server can support up to 2 calls per second. Each WebDialer servlet running on a MCS-7845 server can support up to 4 calls per second



Note

You must run the following files, normally located in `C:\Program Files\Cisco\Web Dialer`, to properly set the ENV variable and Java classes:
`installWDSservice.bat`
`installWDSOAP.bat`

SOAP over HTTP Interface

To access the SOAP interfaces for Cisco Web Dialer, use the Cisco Web Dialer Web Service Definition Language (WSDL) in the [“Cisco Web Dialer WSDL” section on page 18](#).

makeCallSoap

Clients access the `makeCallSoap` interface by initiating a SOAP request to the URL `http://ipaddress/wdsoap/servlet/rpcrouter` for the service name `WebdialerSoap`. In this URL, `ipaddress` specifies the IP address of the Cisco Unified CallManager server. Plug-in clients such as Microsoft Outlook Plug-in use this SOAP interface.

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 “Cisco Web Dialer” chapter in the <i>Cisco Unified CallManager Features and Services Guide, Release 4.2(3)</i> .	Refer to the credential data type in the “Cisco Web Dialer WSDL” section on page 18.	None	None
Profile	Mandatory	The profile that is used to make a call. A calling device such as an IP phone or line represents an example of a typical profile.	Refer to the profile data type in the “Cisco Web Dialer WSDL” section on page 18.	None	None

Results

Refer to the “Cisco Web Dialer WSDL” section on page 18 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	

Table 2 *makeCallSoap Results (continued)*

Error Code	Name	Type	Description	Action by application
5	responseCode	Integer	No device is configured for the user, or missing parameters exist 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 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 unreachable	Displays the appropriate error dialog that allows the user to edit the dialed number.
	responseDescription	String	Destination unreachable	
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	

endCallSoap

Clients access the endCallSoap interface by initiating a SOAP request to the URL <http://ipaddress/wdsoap/servlet/rpcrouter> for the service name *WebdialerSoap*. In this URL, ipaddress specifies the IP address of the Cisco Unified CallManager server on which Cisco Web Dialer is configured.

Plug-in based clients such as Microsoft Outlook Add-in use this SOAP interface.

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 “Cisco Web Dialer” chapter in the <i>Cisco Unified CallManager Features and Services Guide, Release 4.2(3)</i> .	Refer to the credential data type in the “ Cisco Web Dialer WSDL ” section on page 18.	None	None
Profile	Mandatory	The profile that is used to make a call. A calling device such as an IP phone or line represents and example of a typical.	Refer to the profile data type in the “ Cisco Web Dialer WSDL ” section on page 18.	None	None

Results

Refer to the “[Cisco Web Dialer WSDL](#)” section on page 18 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.
	responseDescription	String	Call failure error	
2	responseCode	Integer	Authentication error	Displays the authentication dialog for the 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	
4	responseCode	Integer	Directory error	Displays an appropriate directory error message.
	responseDescription	String	Directory error	

Table 4 *endCallSoap Results (continued)*

Error Code	Name	Type	Description	Action by application
5	responseCode	Integer	No device is configured for the user, or missing parameters exist 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 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 unreachable	Displays the appropriate error dialog that allows the user to edit the dialed number.
	responseDescription	String	Destination unreachable	
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

Clients access the getProfileSoap interface by initiating a SOAP request to the URL <http://ipaddress/wdsoap/servlet/rpcrouter> for the service name *WebdialerSoap*. In this URL, ipaddress specifies the IP address of the Cisco Unified CallManager server on which Cisco Web Dialer is configured.

Plug-in based clients such as Microsoft Outlook Add-in use this SOAP interface.

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 “Cisco Web Dialer” chapter in the <i>Cisco Unified CallManager Features and Services Guide, Release 4.2(3)</i> .	Refer to the credential data type in the “Cisco Web Dialer WSDL” section on page 18.	None	None
UserID	Mandatory	The user ID for which the configuration is requested.	String	None	None

Results

Refer to the “Cisco Web Dialer WSDL” section on page 18 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 that is associated with the user. Refer to the Cisco Web Dialer WSDL for the WDDeviceInfo data type.	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.	
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 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	

isClusterUserSoap

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

Plug-in based clients such as Microsoft Outlook Add-in use this SOAP interface. Multicenter applications that require functionality, similar to a Redirector servlet, use this interface to redirect calls to the various locations where Cisco Web Dialer is installed on a network. The application uses this interface to locate and verify the Cisco Web Dialer servicing the user, followed by `makeCall`, `endCall`, or `getProfile` requests to that Cisco Web Dialer.

Parameters

Table 7 *isClusterSoap Parameters*

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

Results

Refer to the [“Cisco Web Dialer WSDL”](#) section on page 18 for return values and their data type.

Table 8 *isClusterSoap Results*

Name	Type	Description
result	Boolean	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 of the cluster.

HTML over HTTP Interface

This section describes the HTML over HTTP interfaces.

makeCall

The makeCall interface gets used in customized directory search applications. The directory search page (directory.asp) that is bundled with Cisco Unified CallManager also uses this interface. The makeCall interface gets accessed by initiating an HTTP request to the URL `http://ipaddress/webdialer/Webdialer`. In this URL, ipaddress specifies the IP address of the Cisco Unified CallManager server for which Cisco 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 JavaScript” section on page 21](#).

Parameters

Table 9 *makeCall Parameters*

Parameter	Mandatory	Description	Data Type	Range of Values	Default Value
destination	Mandatory	Destination number called by the application. This number gets converted to a regular telephone number by applying the application dial rules. For information on dial rules, refer to the “Cisco Web Dialer” chapter in the <i>Cisco Unified CallManager Features and Services Guide, Release 4.2(3)</i> .	String	None	None

Results

Table 10 *makeCall Results*

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

makeCallProxy

The makeCallProxy interface gets accessed by initiating an HTTP request to the URL `http://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 for only the length of the session.

Applications such as a personal address book, defined in the CCMUser pages at `http://cmserver/CCMUser`, can use the makeCallProxy interface. The credential of the application gets used, as a proxy, to make calls on behalf of users. Because these users have authenticated themselves before accessing the CCMUser page, 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.

For a sample script that is used to enable directory search pages, go to the [“Sample JavaScript” section on page 21](#).

Parameters

Table 11 *makeCallProxy Parameters*

Parameter	Mandatory	Description	Data Type	Range of Values	Default Value
uid	Mandatory	User ID for which the request is made.	String	None	None
appid	Mandatory	Userid of the application that is making a request on behalf of the user. For example, consider a Cisco Unified CallManager personal address book where the application allows authentication proxy rights. The appid parameter gets 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 acts the same as userid.	String	None	None
pwd	Mandatory	Password of the appid	String	None	None
destination	Mandatory	Number to be called. The dial plan service converts this number to an E.164 number.	String	None	None

Results

Table 12 *makeCallProxy Results*

Name	Description
result	Cisco 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 following Web Service Definition Language (WSDL) for Cisco Web Dialer. You can find this WSDL for Cisco Web Dialer on the Cisco Web Dialer 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 Web Dialer. For a list of references on Cisco Unified 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">
  <wSDL: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:sequence>
      </xsd:complexType>
    </xsd:schema>
  </wSDL:types>
</wSDL:definitions>
```

```

        <xsd:element name="deviceInfoList" nillable="true"
type="tns:ArrayOfWDDDeviceInfo"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="WDDDeviceInfo">
    <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="ArrayOfWDDDeviceInfo">
    <xsd:complexContent>
        <xsd:restriction base="soapenc:Array">
            <xsd:attribute ref="soapenc:arrayType"
wsdl:arrayType="tns:WDDDeviceInfo[]" />
        </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>
</portType>

```

```

</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 sample JavaScript gets used to enable Cisco Web Dialer from a directory search page.

Single Cluster Applications

This script gets 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 gets 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 Unified CallManager, Cisco Unified IP Phones, and the technologies that are required to develop applications with Cisco Web Dialer.

Related Topics

- Cisco Web Dialer Administration— “Cisco Web Dialer” chapter in the *Cisco Unified CallManager Features and Services Guide, Release 4.2(3)*.
- Service Parameters Configuration— *Cisco Unified CallManager Administration Guide, Release 4.2(3)*.
- Cisco Unified IP Phone Configuration— *Cisco Unified CallManager Administration Guide, Release 4.2(3)*.

- Application Dial Rules Configuration—“Application Dial Rules” chapter in the *Cisco Unified CallManager Administration Guide, Release 4.2(3)*.

Additional Cisco Documentation

- *Cisco Unified CallManager Release 4.2(3)*—A suite of documents that relate to the installation and configuration of Cisco Unified CallManager. Refer to the *Cisco Unified CallManager Documentation Guide for Release 4.2(3)* for a list of documents on installing and configuring Cisco Unified CallManager 4.2(3).
- *Cisco Unified IP Phones and Services*—A suite of documents that relate to the installation and configuration of Cisco Unified IP Phones.
- *Cisco DistributedDirector*—A suite of documents that relate to the installation and configuration of Cisco DistributedDirector.

Other Related Documentation

- *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.

Obtaining Documentation

Cisco documentation and additional literature are available on Cisco.com. Cisco also provides several ways to obtain technical assistance and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation at this URL:

<http://www.cisco.com/univercd/home/home.htm>

You can access the Cisco website at this URL:

<http://www.cisco.com>

You can access international Cisco websites at this URL:

http://www.cisco.com/public/countries_languages.shtml

Documentation DVD

Cisco documentation and additional literature are available in a Documentation DVD package, which may have shipped with your product. The Documentation DVD is updated regularly and may be more current than printed documentation. The Documentation DVD package is available as a single unit.

Registered Cisco.com users (Cisco direct customers) can order a Cisco Documentation DVD (product number DOC-DOCDVD=) from the Ordering tool or Cisco Marketplace.

Cisco Ordering tool:

<http://www.cisco.com/en/US/partner/ordering/>

Cisco Marketplace:

<http://www.cisco.com/go/marketplace/>

Ordering Documentation

You can find instructions for ordering documentation at this URL:

http://www.cisco.com/univercd/cc/td/doc/es_inpk/pdi.htm

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Ordering tool:
<http://www.cisco.com/en/US/partner/ordering/>
- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, USA) at 408 526-7208 or, elsewhere in North America, by calling 1 800 553-NETS (6387).

Documentation Feedback

You can send comments about technical documentation to bug-doc@cisco.com.

You can submit comments by using the response card (if present) behind the front cover of your document or by writing to the following address:

Cisco Systems, Inc.
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Cisco Product Security Overview

Cisco provides a free online Security Vulnerability Policy portal at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

From this site, you can perform these tasks:

- Report security vulnerabilities in Cisco products.
- Obtain assistance with security incidents that involve Cisco products.
- Register to receive security information from Cisco.

A current list of security advisories and notices for Cisco products is available at this URL:

<http://www.cisco.com/go/psirt>

If you prefer to see advisories and notices as they are updated in real time, you can access a Product Security Incident Response Team Really Simple Syndication (PSIRT RSS) feed from this URL:

http://www.cisco.com/en/US/products/products_psirt_rss_feed.html

Reporting Security Problems in Cisco Products

Cisco is committed to delivering secure products. We test our products internally before we release them, and we strive to correct all vulnerabilities quickly. If you think that you might have identified a vulnerability in a Cisco product, contact PSIRT:

- Emergencies — security-alert@cisco.com
- Nonemergencies — psirt@cisco.com



Tip

We encourage you to use Pretty Good Privacy (PGP) or a compatible product to encrypt any sensitive information that you send to Cisco. PSIRT can work from encrypted information that is compatible with PGP versions 2.x through 8.x.

Never use a revoked or an expired encryption key. The correct public key to use in your correspondence with PSIRT is the one that has the most recent creation date in this public key server list:

<http://pgp.mit.edu:11371/pks/lookup?search=psirt%40cisco.com&op=index&exact=on>

In an emergency, you can also reach PSIRT by telephone:

- 1 877 228-7302
- 1 408 525-6532

Obtaining Technical Assistance

For all customers, partners, resellers, and distributors who hold valid Cisco service contracts, Cisco Technical Support provides 24-hour-a-day, award-winning technical assistance. The Cisco Technical Support Website on Cisco.com features extensive online support resources. In addition, Cisco Technical Assistance Center (TAC) engineers provide telephone support. If you do not hold a valid Cisco service contract, contact your reseller.

Cisco Technical Support Website

The Cisco Technical Support Website provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The website is available 24 hours a day, 365 days a year, at this URL:

<http://www.cisco.com/techsupport>

Access to all tools on the Cisco Technical Support Website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register at this URL:

<http://tools.cisco.com/RPF/register/register.do>

**Note**

Use the Cisco Product Identification (CPI) tool to locate your product serial number before submitting a web or phone request for service. You can access the CPI tool from the Cisco Technical Support Website by clicking the **Tools & Resources** link under Documentation & Tools. Choose **Cisco Product Identification Tool** from the Alphabetical Index drop-down list, or click the **Cisco Product Identification Tool** link under Alerts & RMAs. The CPI tool offers three search options: by product ID or model name; by tree view; or for certain products, by copying and pasting **show** command output. Search results show an illustration of your product with the serial number label location highlighted. Locate the serial number label on your product and record the information before placing a service call.

Submitting a Service Request

Using the online TAC Service Request Tool is the fastest way to open S3 and S4 service requests. (S3 and S4 service requests are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Service Request Tool provides recommended solutions. If your issue is not resolved using the recommended resources, your service request is assigned to a Cisco TAC engineer. The TAC Service Request Tool is located at this URL:

<http://www.cisco.com/techsupport/servicerequest>

For S1 or S2 service requests or if you do not have Internet access, contact the Cisco TAC by telephone. (S1 or S2 service requests are those in which your production network is down or severely degraded.) Cisco TAC engineers are assigned immediately to S1 and S2 service requests to help keep your business operations running smoothly.

To open a service request by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411 (Australia: 1 800 805 227)

EMEA: +32 2 704 55 55

USA: 1 800 553-2447

For a complete list of Cisco TAC contacts, go to this URL:

<http://www.cisco.com/techsupport/contacts>

Definitions of Service Request Severity

To ensure that all service requests are reported in a standard format, Cisco has established severity definitions.

Severity 1 (S1)—Your network is “down,” or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Severity 2 (S2)—Operation of an existing network is severely degraded, or significant aspects of your business operation are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Severity 3 (S3)—Operational performance of your network is impaired, but most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Severity 4 (S4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- Cisco Marketplace provides a variety of Cisco books, reference guides, and logo merchandise. Visit Cisco Marketplace, the company store, at this URL:

<http://www.cisco.com/go/marketplace/>

- *Cisco Press* publishes a wide range of general networking, training and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to Cisco Press at this URL:

<http://www.ciscopress.com>

- *Packet* magazine is the Cisco Systems technical user magazine for maximizing Internet and networking investments. Each quarter, Packet delivers coverage of the latest industry trends, technology breakthroughs, and Cisco products and solutions, as well as network deployment and troubleshooting tips, configuration examples, customer case studies, certification and training information, and links to scores of in-depth online resources. You can access Packet magazine at this URL:

<http://www.cisco.com/packet>

- *iQ Magazine* is the quarterly publication from Cisco Systems designed to help growing companies learn how they can use technology to increase revenue, streamline their business, and expand services. The publication identifies the challenges facing these companies and the technologies to help solve them, using real-world case studies and business strategies to help readers make sound technology investment decisions. You can access iQ Magazine at this URL:

<http://www.cisco.com/go/iqmagazine>

- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the Internet Protocol Journal at this URL:

<http://www.cisco.com/ipj>

- World-class networking training is available from Cisco. You can view current offerings at this URL:

<http://www.cisco.com/en/US/learning/index.html>

CCSP, CCVP, the Cisco Square Bridge logo, Follow Me Browsing, and StackWise are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn, and iQuick Study are service marks of Cisco Systems, Inc.; and Access Registrar, Aironet, BPX, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, FormShare, GigaDrive, GigaStack, HomeLink, Internet Quotient, IOS, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, LightStream, Linksys, MeetingPlace, MGX, the Networkers logo, Networking Academy, Network Registrar, *Packet*, PIX, Post-Routing, Pre-Routing, ProConnect, RateMUX, ScriptShare, SlideCast, SMARTnet, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0601R)

Copyright 2006 Cisco Systems Inc. All rights reserved.