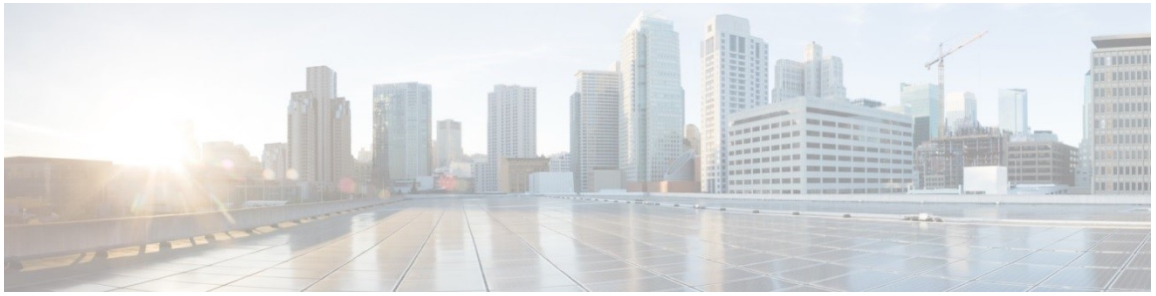
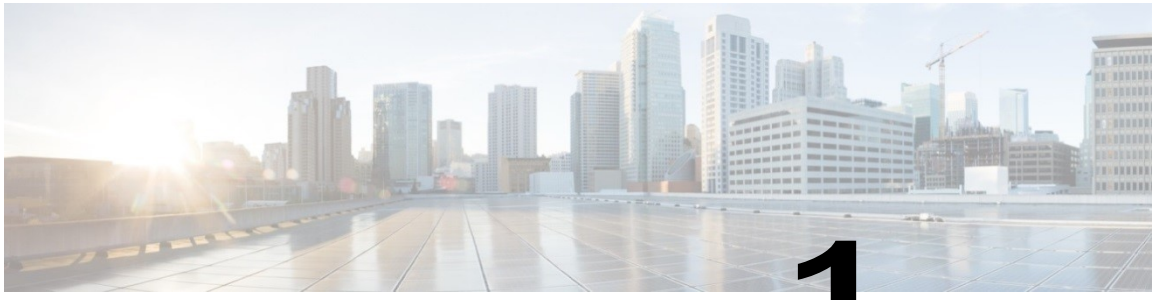
Getting Started Guide for Cisco Jabber SDK for Web, Voice and Video Workload Version 11.8.2, System Release 11.8

First Published: December 19, 2013
Last Modified: August 28, 2017

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883



CONTENTS



CHAPTER 1

Preface

The Cisco Jabber Voice and Video SDK (part of the Cisco Jabber SDK for Web) allows you to embed voice and video telephony into your web pages. Although the SDK is designed to hide as much of the underlying complexity as possible, this guide is designed to give you important background information that will help you to get started as well as to understand some of the more advanced features of the SDK.

- Cisco Jabber Voice and Video SDK, page 5

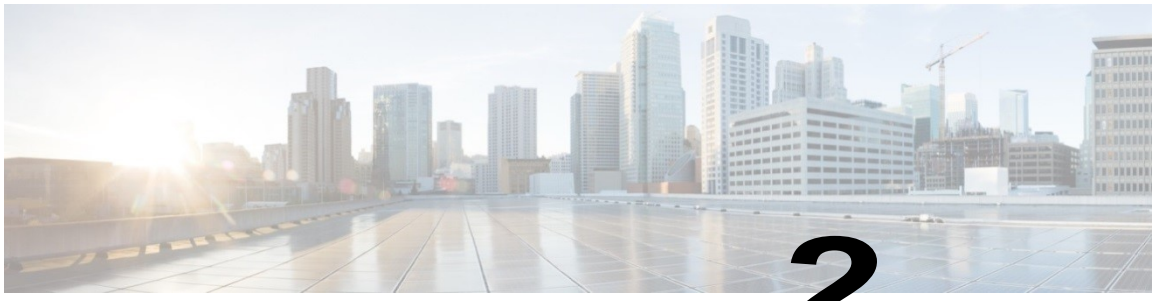
Cisco Jabber Voice and Video SDK

With the Cisco Jabber Voice and Video SDK, you can enable the following telephony features directly in the browser:

- Make and receive calls
- Transmit and receive HD video
- Transfer calls
- Create and manage conferences
- Hold and resume calls
- Mute/unmute calls
- Use DTMF
- Control your desk phone
- BFCP Screen Share
- Far end camera control (FECC)
- Extend and Connect
- Hunt Group
- Call Pickup

The Voice and Video SDK provides three phone operating modes:

- Desk phone mode: Control of a physical desk phone, including the ability to make calls, end the call, and use mid-call features
- Softphone mode: Softphone features, including desktop media termination and calling features.
- Extend and Connect mode: Enables calling features with any remote destination endpoints using Extend and Connect feature.



CHAPTER 2

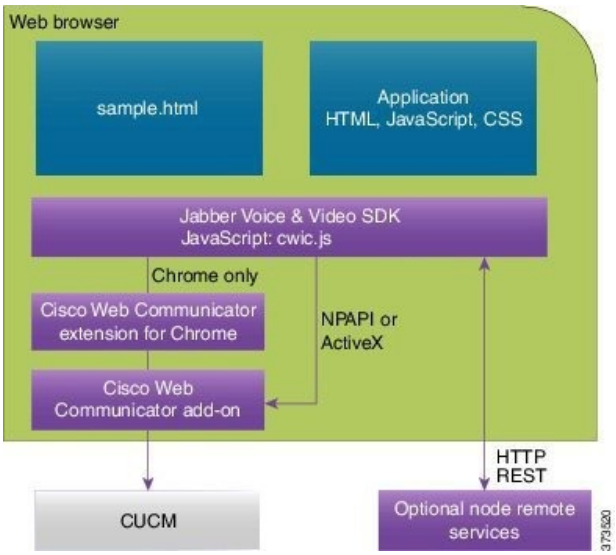
Cisco Jabber Voice and Video SDK Overview

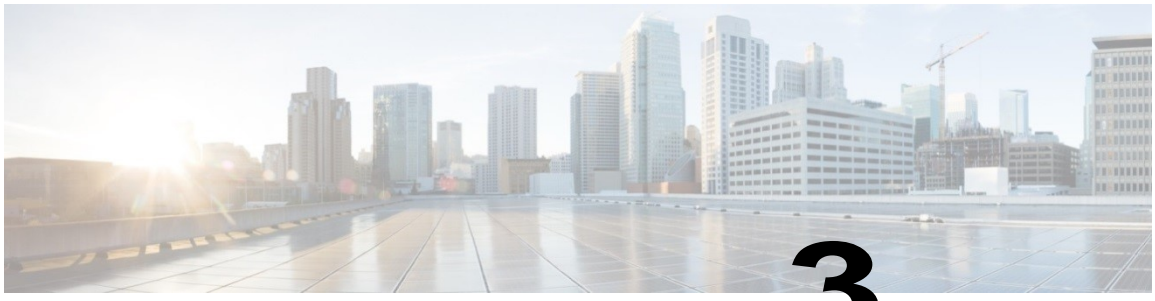
Architecture Overview

The SDK itself contains the following components:

- Cisco Web Communicator add-on that provides call control and media termination within the browser.
  - Available on Windows and Mac operating systems.
  - Compatible with Internet Explorer, Firefox, Chrome and Safari.
  - Support for *soft phone mode* (connects SIP with Unified Communications Manager, and uses the computer for audio), *desk phone mode* (connects CTI with Unified Communications Manager, and uses a physical phone for audio) and *Extend and Connect mode*.
- JavaScript API enables Cisco voice and video telephony from a web page.
- Optional server-side node that provides advanced features such as directory integration.

The following diagram shows the high-level architecture of the SDK.





CHAPTER 3

Environmental Requirements

The following sections describe the environmental requirements for Cisco Jabber SDK.

Supported Operating Systems and Browsers

Windows

Browser		Windows 7	Windows 8.1	Windows 10
		32-bit and 64-bit		
Chrome: Stable channel 55	32-bit	x	x	x
	64-bit	x	x	x
Firefox: Final release channel 50	32-bit	x	x	x
	64-bit			
Internet Explorer 11	32-bit	x	x	x
	64-bit	x	x	x

Important

- As Microsoft announced, after January 12, 2016, only the most recent version of Internet Explorer available for a supported operating system will receive technical support and security updates. Jabber Web SDK releases after 11.0 will follow this support policy. Enterprise Mode for Internet Explorer 11, released in April 2014, offers enhanced backward compatibility and enables you to run many legacy Web apps during your transition to modern Web standards.
- Web browsers using Jabber SDK on Windows 8 and later must run in Windows desktop mode. For IE, adding a *requiresActiveX* "META" tag to your page will prompt users to reload IE in the traditional desktop mode if they are using IE in the new Windows UI.
- Using the SDK and Protected Mode in IE is not supported.
- Using the SDK in IE legacy compatibility modes is not supported.
- Windows RT 8 and Windows RT 8.1 are not supported.

Mac OS X

Browser	Mac OS X 10.10 Yosemite	Mac OS X 10.11 El Capitan	Mac OS X 10.12 Sierra
Chrome: Stable channel 55	x	x	x
Firefox: Final release channel 50	x	x	x
Safari 9	x	x	x
Safari 10	x	x	x

We recommend that App Nap, a feature in Mac OS X, be disabled for all browsers supported by the Jabber SDK.

Additional Notes

The Cisco Jabber SDK does not support virtual desktop environments such as Citrix XenApp, Remote Desktop Protocol, and so on.

Ensure that your browser is not configured to prevent browser plug-ins from running. In Internet Explorer browser plug-ins are referred to as *ActiveX Controls*. Chrome and Firefox use the term *plug-in*.

Supported Processors

Minimum CPU instruction set is SSE3.
Minimum CPU speed and type:

- Mobile AMD Sempron Processor 3600+ at 2 GHz
- Intel Core 2 CPU T7400 at 2.16 GHz

Supported Codec Versions

Supported Audio Codecs:

- G.711 a-law and u-law
- G.722
- G.722.1 24k
- G.722.1 32k
- G.729a
- Opus codec is supported with CUCM 11.0

Supported Video Codecs:

- H.264/AVC

Supported CUCM and Expressway Versions

Cisco Unified Communications Manager 9.0 and later.
Cisco Expressway Series for Cisco Unified Communications Manager

- Cisco Expressway-E, version 8.1.1 or later
- Cisco Expressway-C, version 8.1.1 or later

Note The Cisco Jabber Voice and Video SDK does not support Cisco Unified Survivable Remote Site Telephony (SRST).

Cross-Origin Resource Sharing Support

In Cisco Unified Communications Manager 10.5, the Cross-Origin Resource Sharing (CORS) feature allows a web application to make an **XMLHttpRequest** to a domain other the domain that the JavaScript originated from. Without CORS, such cross-domain requests would be forbidden by web browsers.

Using Cisco Unified Communications Manager User Data Service, an application now has the ability to get and change its own user data, such as getting a phone book or directory number (DN), or it can add a user to its phone book.

Cisco Unified Communications Manager Administration gives administrators the ability to add or remove domains to the access list. A domain can have Read-Only or Full-Access privilege:

- A Read-Only domain can issue an **HTTP GET** or **HEAD** command to the Cisco Unified Communications Manager data.
- A Full-Access domain can issue an **HTTP GET**, **HEAD**, **PUT**, **POST**, or **DELETE** command to the Cisco Unified Communications Manager data.

The following procedure describes how to add a domain for CORS access:

Procedure

Step 1 Sign in to Cisco Unified Communications Manager Administration.
Step 2 Choose System > Cross-Origin Resource Sharing (CORS).
Step 3 Click Add New.

Sample code for a GET request

```
$.ajax({

          url: encodeURI(url.value),
          type: 'GET',
          contentType: 'application/json',
          dataType: 'json',
          xhrFields: {
              withCredentials: true
          },
          crossDomain: true,
          success: function (data) {
                    status.value = JSON.stringify(data);
          },
          error: function () {
                     status.value = 'request failed';
          }
});
```

Sample code for a POST request

```
$.ajax({

          url: encodeURI(url.value),
          type: POST,
          data: body.value
          contentType: 'application/json',
          dataType: 'json',
          xhrFields: {
              withCredentials: true
          },
          crossDomain: true,
          success: function (data) {
                    status.value = JSON.stringify(data);
          },
          error: function () {
                     status.value = 'request failed';
          }
});
```

Single Sign-On Requirements

- Cisco Unified Communications Manager release 10.5.0 or later
- Configured redirect URI on CUCM
- Provide HTML page at "redirect URI" location to get a token (if it's opened in popup window, popup blocker must be configured to allow this page)

Supported Identity Providers

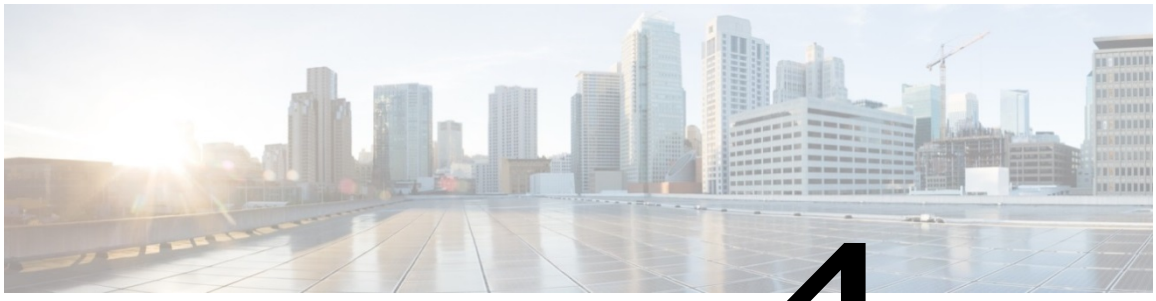
The IdP must be Security Assertion Markup Language (SAML) compliant.

- Ping Federate 6.10.0.4
- Microsoft Active Directory Federation Services (ADFS) 2.0
- Open Access Manager (OpenAM) 10.1

When you configure the IdP, the configured settings impact how you sign into the client. Some parameters, such as the type of cookie (persistent or session), or the authentication mechanism (Kerberos or Web form), determine how often you have to be authenticated.

SSO and Remote Access

- Cisco Unified Communications Manager release 10.5.2 or later.
- Cisco Expressway 8.5
- The Identity Provider used must have the same internal and external URL. If the URL is different, the user may be prompted to sign in again when changing from inside to outside the corporate firewall and vice versa.
- For remote access, AD users synchronized with CUCM must have email field populated.



CHAPTER 4

Getting Started

Naming Conventions

With Google's deprecation of the Netscape Plug-in Application Programming Interface (NPAPI) in Chrome, the architecture for the Jabber SDK to work with Chrome required a change from an NPAPI plug-in for Chrome to both a Chrome extension and a native-OS add-on to Chrome. These new components are in addition to the existing plug-ins for other supported browsers. Terminology changes were needed as follows:

- The term *add-on* is used for:
    - *CiscoWebCommunicatorAddOn* process that is installed.
- The term *plug-in* is used for:
    - *npCiscoWebCommunicator* dynamic library that is used as a browser plugin for Firefox, Internet Explorer and Safari.
- The existing plug-in for Internet Explorer, Firefox, and Safari is named the *Cisco Web Communicator plug-in*.
- The new components for Chrome support are the Cisco Web Communicator extension for Chrome and the Cisco Web Communicator add-on for Chrome. The Cisco Web Communicator add-on for Chrome is what Chrome calls a native messaging host.

Add Browser-Based Softphone

The Cisco Jabber SDK works with Cisco Unified Communications Manager software to provide voice and video functionality. You will either need access to your own Cisco Unified Communications Manager infrastructure, or you can check our testing overview for other options.
See the steps in the following sections to configure a browser-based softphone.

Download the Cisco Web Communicator Add-on

The Cisco Web Communicator add-on is required for signaling and media termination in the browser through `cwic.js`. It contains the:

- Cisco Web Communicator plug-in (Netscape Plug-in Application Programming Interface [NPAPI] for Firefox and Safari; ActiveX for Internet Explorer).
- Cisco Web Communicator add-on for Chrome.

To use the SDK in Chrome or Firefox (version 50 and above), the Cisco Web Communicator extension for Chrome/Firefox is also required. The extension is a bridge between `cwic.js` and the Cisco Web Communicator add-on.

Procedure

Step 1 Download and install the Cisco Web Communicator add-on.
If you have already purchased a user license for the Jabber SDK (meaning that you have purchased UCL, CUWL or other relevant licensing for the target users, along with the appropriate support and upgrade contract, for example, ESW and UCSS), download the add-on from the following location:
https://software.cisco.com/download/release.html?mdfid=283882159&softwareid=283995793

Step 2 If you want to use the SDK in Chrome, download and install the Cisco Web Communicator extension for Chrome from the Chrome Web Store: https://chrome.google.com/webstore/detail/cisco-web-communicator/ppbllmlcmhfnfflbkbinnhacecaankdh

Step 3 If you want to use SDK with Firefox version 50 and later, download and install the Cisco Web Communicator extension for Mozilla Firefox from Mozilla Add-on Store: https://addons.mozilla.org/en-US/firefox/addon/cisco-web-communicator/

Important Installation of Cisco Web Communicator Add-on requires administrative rights

Best Practices for Creating Applications for Chrome and Firefox (version 50 and above)

Obtaining the Add-on and Extension

The Chrome Jabber SDK implementation requires users to install both the Cisco Web Communicator add-on and the Cisco Web Communicator extension for Chrome or Firefox.

To make it as easy as possible for Chrome users to obtain the extension and the add-on, provide the following in your application:

- A link to the extension in the Chrome Web Store.
- A link to the extension in the Mozilla Add-on Store.
- A link to the location where you distribute the add-on.

We suggest that you set the links to open in a new window or tab so that users do not navigate away from your application.
For tutorial on how to best achieve this please refer to JSDK API documentation's tutorials.

Permissions Requested

The following tables describes the permissions requested by the Cisco Web Communicator extension:

Permission Requested	**Reason**
Access to your data on all websites	Allows the extension to inject, into every page, a content script that allows cwic.js to communicate with the extension.
Manage your apps, extensions, and themes	Allows the extension to determine the host name of a web page attempting to use cwic.js. This is the name you authorize when the add-on displays an Access Control Dialog (ACD).
Communicate with cooperating native applications	Necessary for the extension to communicate with the Cisco Web Communicator add-on.

Set Up Device Type in CUCM

This section explains how to set up the correct device type in Cisco Unified Communications Manager to enable you to use the Cisco Jabber SDK. In brief, it involves creating a CSF device with the name CSF<yourusername>, but follow these detailed steps to ensure proper functionality.

Procedure

Step 1 Sign in to Cisco Unified Communications Manager Administration.

Step 2 Choose **Device** > **Phone**.
The Find and List Phones window opens.

Step 3 Click **Add New**.
The Add a New Phone window opens.

Step 4 From the **Phone Type** drop-down list, select **Cisco Unified Client Services Framework**.

Step 5 Click **Next**.
The Phone Configuration window opens.

Step 6 In the **Device Information** section of the Phone Configuration window, set the following:

- Device Name—By default, the name should be of the form: ECP<username>. Example: ECPjohndoe. However, the SDK allows you to override the prefix to one of your own by passing in the devicePrefix field in the settings

object. You can also use your own custom scheme for associating CUCM device names with usernames by providing your own settings.predictDevice function.
    - The device name is not case sensitive.
    - The device name is created by placing the prefix in front of the username and then removing any characters which are not permitted. Symbols such as dots, hyphens, underscores must be stripped, as well as any accented characters or characters not in the Latin (English) alphabet.
    - Cisco Unified Communications Manager accepts a maximum length of 15 characters, so the generated name must be truncated to this length.
    - This may lead to some name clashes because names that are only unique in the thirteenth character and beyond become the same name when ECP is prepended and the total length is truncated to 15. Also, the names Joe.Bloggs and JoeBloggs both map to the same device name: ECPJoeBloggs. These ambiguities must be handled on a case-by-case basis, and may require that the user names be changed to make them unique.
    - Similarly, the user Frédéric will have a device name of ECPFrdric. Dropping the non-Latin characters can lead to further name clashes.
- Description—Enter a descriptive name, such as John Doe's Web Communicator.
- Device Pool—Set to the desired device pool.
- Phone Button Template—Set to **Standard Client Services Framework.**

Step 7 In the Protocol Specific Information section of the Phone Configuration window, set the following:

- Device Security Profile—Set to **Cisco Unified Client Services Framework - Standard SIP Non-Secure.**
- SIP Profile—Set to **Standard SIP Profile.**

Step 8 Click **Save.**

Step 9 Click **Apply Config** if this button is available (and confirm when prompted).

Step 10 To add a line for the Cisco Web Communicator device, click **Line [1] - Add a New DN** on the upper-left portion of the Phone Configuration window.
The Directory Number Configuration window opens.

Step 11 Type in a Directory Number.

Step 12 Click Save, and then click **Apply Config** if this button is available (and confirm when prompted).
You should see the name of the softphone device in the Associated Devices box on the Directory Number Configuration page.

Step 13 Scroll down to the bottom of the Directory Number Configuration page. Click **Associate End Users.**
The Find and List Users window opens.

Step 14 Use the search criteria to find the user you want to associate with the directory number, then check the box next to that user name and click **Add Selected.**
The Directory Number Configuration window should now show that the user is associated with the line. This information appears near the bottom of the window in the section called **User Associated With Line.**

Step 15 Click on the user name in the **User Associated with Line** section of the window.
The End User Configuration window opens.

Step 16 Scroll down to the Direct Number Associations section of the window and select the primary extension from the Primary Extension drop-down list. This is the directory number you configured in step 11.

Step 17 In the Permissions Information section at the bottom of the End User Configuration window, click Add to **User Group.**
The Find and List User Groups window opens.

Step 18 Use the search criteria to find Standard CCM End Users.

Step 19 Check the box next to **Standard CCM End Users**, then click **Add Selected**.
The Standard CCM End Users group should now appear in the Permissions Information section at the bottom of the End User Configuration window.

Step 20 Click **Save**.
The Cisco Web Communicator device is now configured in Cisco Unified Communications Manager.

Configure Device

Perform the following steps to configure your device using the SDK.

Procedure

Step 1 Open sample.html that comes with the SDK download.

Step 2 Enter your Cisco Unified Communications Manager username, password, and IP address in the fields provided.

Step 3 Select **SoftPhone** as the mode.

Step 4 Click **Login**.

Step 5 Select the device and then click **Connect**. The **Make Call** button will be enabled once your soft phone registers with Cisco Unified Communications Manager.

Step 6 Enter a number into the **Number** field and click **Make Call**.

Step 7 Finally, locate a colleague with whom you can make test calls and initiate some browser-to-browser video calls. Alternatively, you can make calls to existing Cisco hardware and software voice and video products.

What to Do Next

Deploy the cwic.js and ciscobase.js files to your web server and add a softphone to your existing web application.

Service Discovery and Sign-in Types

Having service discovery feature present, now two types of sign in are possible:

1. **Manual sign in** - server addresses (tftp, ccmcip, cti) must be submitted along with other standard sign in parameters.
2. **Discovery based sign in** – server addresses are discovered automatically.

From the 11.0 release of Cisco Jabber Web SDK, automatic discovery of users' home cluster is possible. Service Discovery is a CUCM feature, adding to CUCM 9, which uses UDS API's to determine a users' home cluster from a central server running UDS and from there be able to login and download the appropriate information required for connecting to desired telephony device.

A users' home cluster is discovered by querying a Central UDS server in a given deployment. To find a Central UDS server a DNS Service Record (SRV) called _cisco-uds is queried. Jabber SDK knows what SRV record to query based on user's domain, which is collected by:

a) Prompting user for an email address from which the domain is extracted and cached.
b) Automatically discovering User Principal Name (UPN) on Windows OS machines. If Jabber SDK on Windows platforms fails to discover the UPN it will prompt the user to enter their email address. Jabber SDK on Mac OS X will always prompt the user to enter their email address if it is not cached.

After discovering its home node, Jabber SDK needs to discover if SSO is enabled or not. It is internally done by querying a specific CUCM's API. Based on the result of this query, Jabber SDK will proceed with SSO or credentials based sign in.

In the case of manual sign in, all required input parameters are submitted in advance, whereas in discovery based sign in, required inputs are prompted and submitted through the sign in lifecycle and appropriate callbacks.

Note: To use Service Discovery feature, user's home cluster must be configured appropriately. Configuration of UC services is outside the scope of this document. For more information, please refer to the UCM maintenance and operations guide.

Implementing Sign-In

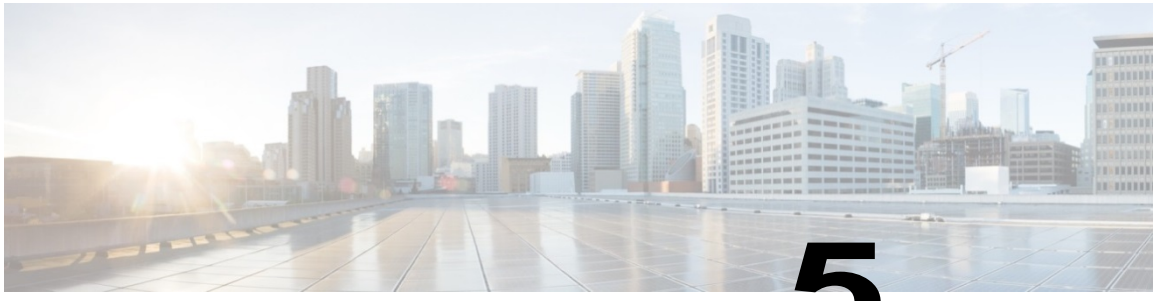
For tutorial on how to best achieve this please refer to JSDK API documentation's tutorials.

Video Capabilities

There are two ways to implement video rendering, depending on the browser type:

- In browser video window for browsers that support NPAPI/ActiveX (Firefox, Internet Explorer, Safari)
- Through native video window that is then used as an overlay over target HTML element, to crate seamless experience as if the video is rendered inside of browser page (Google Chrome and Firefox version 50 and above)

For tutorial on how to best use this feature please refer to JSDK API documentation's tutorials.



CHAPTER 5

Feature Notes

This chapter provides information on select features of the Voice and Video SDK.

Access Control Dialog Feature Overview

This feature is available for the Voice and Video SDK version 3.0.0 and later.

The SDK provides a GUI control, Access Control Dialog, for user notification and response when a website that has not been previously visited, or that is not in a whitelisted domain, tries to use the add-on. The dialog box is displayed in front of the browser. On some OS/browser combinations, the user may not be able to interact with the browser or interact with the webpage while the dialog box is open.

This feature is invoked when the add-on is initialized. It specifies the domain that is requesting permission and provides the user with the option to Deny, Allow or Always Allow access to a desktop phone, camera or microphone.

Always Allow selections are shared across all browsers on both Macintosh and Windows systems.

Access Control Dialog Box

Name	Description
User notification	"Allow the page to control your desk phone or to access your microphone and camera? Clicking Deny may limit application behavior during this session."
Always Allow	Website permitted to use the add-on during the current visit and for all future visits by the same user from the same client machine.
Allow	Website permitted to use the add-on during the current visit. Future visits to the same website trigger the modal dialog box.
Deny	Website not permitted to use the add-on during the current visit. Future visits to the same website trigger the modal dialog box.

Domain Whitelist Support

An administrator is able to add domains to the whitelist. The add-on install process does not modify existing whitelist entries.

Mac OS X.x

On OS *X.x*, an administrator adds the whitelisted domains to the add-on preference list:

com.cisco.CiscoWebCommunicator.plist

For example, to whitelist **jabberweb.cisco.com**, create the Key **AlwaysAllow_jabberweb.cisco.com** and set its Boolean value to **Yes**.

For user specific preferences:

- On Mac OS X 10.7, 10.8 and 10.9, add **com.cisco.CiscoWebCommunicator.plist** to the
- ~/Library/Preferences/folder. Overwriting this file may remove existing entries.
- On Mac OS X 10.10, run the following command:
- defaults import com.cisco.CiscoWebCommunicator
- */path_to_file*/com.cisco.CIscoWebCommunicator.plist

For system-wide preferences, add **com.cisco.CiscoWebCommunicator.plist** in the root HD/Library/Preferences/folder.

Windows

In order to whitelist a domain, create a DWORD registry value under the following keys with the specified parameter values.

User whitelist

Key: HKEY_CURRENT_USER\SOFTWARE\Cisco Systems, Inc.\Web Communicator\AlwaysAllow

Domain to whitelist: jabberweb.cisco.com

Boolean value: 1

Administrator whitelist

Key: HKEY_CURRENT_USER\SOFTWARE\Policies\Cisco Systems, Inc.\Web Communicator\AlwaysAllow

Domain to whitelist: jabberweb.cisco.com

Boolean value: 1

Note The Windows add-on merges the administrator and user-allowed domains by reading these two registry keys

As of 3.0.1:

Under the admin whitelist, administrators can use either key to push whitelisted domains out to users.
Administrator whitelist:
HKEY_CURRENT_USER\SOFTWARE\Policies\Cisco Systems, Inc.\Web Communicator\AlwaysAllow
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Cisco Systems, Inc.\Web Communicator\AlwaysAllow

Note Whitelisting is intended for managed environments where registry changes can be pushed out through some standard process.

Language Support

The Access Control Dialog box supports localization of the user notification message.

Languages

- Arabic (Saudi Arabia)
- Bulgarian (Bulgaria)
- Catalan (Catalan/Spain)
- Chinese (China)
- Chinese (Taiwan)
- Croatian (Croatia)
- Czech (Czech Republic)
- Danish (Denmark)
- Dutch (Netherlands)
- English (United States)
- Finnish (Finland)
- French (France)
- German (Germany)
- Greek (Greece)
- Hebrew (Israel)
- Hungarian (Hungary)
- Italian (Italy)
- Japanese (Japan)
- Korean (Korea Republic)
- Norwegian Bokmal (Norway)
- Polish (Poland)
- Portuguese (Brazil)
- Portuguese (Portugal)
- Romanian (Romania)
- Russian (Russia)
- Serbian (Serbia)
- Slovak (Slovakia)
- Slovenian (Slovenia)
- Spanish (Spain)
- Swedish (Sweden)
- Turkish (Turkey)
- Thai (Thailand)

Retain position of external video window

The position and size of the undocked external video window is retained between calls, browser restarts, different browsers or machine reboots. While the external video window is docked, the position and size of window are not saved. After undocking the window will retain its last position and size prior to docking.

The information about window position and size is kept in Registry on Windows OS and Preferences file on Mac OS.

Win: HKEY_CURRENT_USER\SOFTWARE\Cisco Systems, Inc.\Web Communicator\jsdkhostwinrect
Mac: jsdkhostwinrect@com.cisco.CiscoWebCommunicator

These values are deleted when the CiscoWebCommunicator Add-On is uninstalled .

Certificate Validation Feature Overview

Certificate Validation provides application developers with the ability to validate certificates used during user authentication against the certificate store on the client computer. Certificate Validation is mandatory for the Voice and Video SDK version 4.0.0 and later.

For tutorial on how to best use this feature please refer to JSDK API documentation's tutorials.

Windows

The self-signed certificate of the authentication server, or the root certificate of the Certificate Authority that signed the authentication server certificate, must be installed in the Current User or Machine account Trusted Root Certificate Authorities store. The Cisco Jabber Web SDK performs revocation list checking. If you are using a certificate authority, and the Certificate Authority policy requires CRL checking, it must be correctly configured in your network environment.

The Microsoft TechNet web site can provide more information about certificate management and usage in a Microsoft Windows environment. The Microsoft TechNet web site is available at http://technet.microsoft.com.

Mac OS X

The self-signed certificate of the authentication server, or the root certificate of the Certificate Authority that signed the authentication server certificate, must be installed in the user's Login or System store, the certificate must be marked Trusted for All Users, and marked as Always Trust for X.509 Basic Policy.

The Apple Support web site can provide more information about certificate management and usage in a Mac OS X environment. The Apple Support web site is available at http://www.apple.com/support/.

Single Sign On

Single Sign On (SSO) simplifies the login process and offers easier and consistent way to authenticate across different services. Instead of each service requiring a separate username and password, a single credential allows access to all enabled services.

This document only covers how to use SSO with Cisco Unified Communication; it does not cover SSO fundamentals or concepts. The Cisco implementation of SSO is straightforward and easy to quickly setup and use without you having to understand the fundamentals of SSO. However, if you would like to learn more about the fundamentals of SSO, we recommend that you research the following topics:

- Security Assertion Markup Language (SAML)
- Open Authentication (OAuth)
- Identity Provider (IdP)
- Assertion Consumer Service (ACS)

To use SSO feature, user's home cluster must be SSO enabled first. Configuration of UC services for SSO is outside the scope of this document. For more information, please refer to the UCM maintenance and operations guide.

Note: Jabber SDK SSO feature is supported with Cisco Unified Communications Manager v10.5+.

Understanding Redirects and Same-Origin Issues

For security reasons, web browsers prevent Javascript served by one origin from accessing data served by another origin (an origin is a combination of protocol, hostname, and port.) For example, Javascript on a page served from https://app.example.com cannot access the DOM of a popup window served from https://ucmanager.example.com, e.g. to retrieve a token.

The complete UC service SAML/OAuth SSO request sequence normally involves several HTTP interactions (302/Found redirects, Basic Auth challenge/response, scripted POST of a hidden HTLM form, etc.) which involve requests served from different 'origins': i.e. the UC server and the Identity Provider. By default, the final HTTP location (and token info) is served from the UC service host (this normally prevents Javascript access to the token from other browser windows).

In order to facilitate this process, UCM allow the administrator to configure one or more 3rd party 'redirect URLs' to which the SSO sequence can be redirected after successful sign in. The application provides this redirect URL as a parameter when it initiates the SSO sign in sequence. The redirect URL should point to a web page hosted on the application web server, which will typically include some Javascript which extracts the SSO 'access_token' data and provides it back to a parent browser window.

The application redirect URL is configured in UCM in the Enterprise Parameters area (System->Enterprise Parameters->SSO Configuration), via the Redirect URIs for Third Party SSO Client parameter (multiple comma-separated URLs are allowed):

Setup Jabber SDK to use SSO feature

For tutorial on how to best use this feature please refer to JSDK API documentation's tutorials.

Note: When user is successfully authenticated with IdP, IdP creates a cookie in the current browser to keep the user signed in across different services and sessions, so user is not prompted for credentials again during the lifetime of the cookie. Type and the lifetime of created cookie is dependent on IdP configuration, i.e. it could be configured to create session or permanent cookies. It is important to note that during the lifetime of created cookie only the user for which the cookie is created can sign in with SSO. This is in line with other web services supporting SSO. If you want to force the IdP to show the authentication UI again, cookie must be manually deleted from the cookie store.

Device Selection

The Jabber Web SDK provides means to choose from all available cameras, microphones, speakers, and other audio devices. This feature is available for the Voice and Video SDK version 4.0.0 and later.
In 11.0 release playout and ringer devices are separated, which means that different speaker devices could be used for conversation and ring. On Windows platforms, it is possible to set all available ringer devices as current ringer.

For tutorial on how to best use this feature please refer to JSDK API documentation's tutorials.

Volume control on Windows platforms

In 11.0 release, Jabber Web SDK provides possibility of multimedia devices volume control. It is possible to set different volume levels on playout, ringer and recording devices. This feature is available on Windows platform only.
For tutorial on how to best use this feature please refer to JSDK API documentation's tutorials.

Note: Between sessions, volume control settings are not saved per user basis, but on system level.

Custom ringtones on Windows platforms

In 11.7 release, Jabber Web SDK introduces Custom ringtones feature, which brings possibility of using custom .wav files as ringtones. SDK comes with twelve preset different ringtones, but it is possible to expand that set with user-chosen files. Those files should be put in installation_directory/Sound directory.
This feature is available on Windows only.

For tutorial on how to best use this feature please refer to JSDK API documentation's tutorials.

BFCP Screen Share

Binary Floor Control Protocol (BFCP) provides video desktop sharing capabilities for software phone devices, also known as CSF devices. Cisco Unified Communications Manager handles the BFCP packets that users transmit when using video desktop sharing capabilities. On Cisco Unified Communications Manager version 9.0(1) and later, BFCP presentation sharing is automatically enabled. For this reason, you do not need to perform any steps to enable video desktop sharing on CSF devices.

Extend and Connect

Extend and Connect is a feature that allows administrators to rapidly deploy UC Computer Telephony Integration (CTI) applications which interoperate with any endpoint. With Extend and Connect, users can leverage the benefits of UC applications from any location using any device. This feature also allows interoperability between newer UC solutions and legacy systems, so customers can migrate to newer UC solutions over time as existing hardware is deprecated.

Far End Camera Control (FECC)

In calls that support far-end camera control (FECC), user can adjust the far-end camera to give a better view during video calls. FECC is available to users if the endpoint that they are calling supports it. It can be configured whether users can access FECC-enabled endpoints in jabber-config.xml . Disabling the configuration parameter means that users are not provided with the ability to control far-end camera endpoints, even if the endpoint is capable. To disable FECC, set the *EnableFecc* parameter to false.
Limitations: FECC is only supported in point-to-point calls, but not in group calls or conferences where multiple video connections are connecting to the same bridge. FECC is only supported in Softphone mode

Hunt Group

A hunt pilot contains a hunt pilot number and an associated hunt list. Hunt pilots provide flexibility in network design. They work in conjunction with route filters and hunt lists to direct calls to specific devices and to include, exclude, or modify specific digit patterns. A hunt list contains a set of line groups in a specific order. A single line group can appear in multiple hunt lists. The group call pickup feature and directed call pickup feature do not work with hunt lists. A line group comprises a group of directory numbers in a specific order. The order controls the progress of the search for available directory numbers for incoming calls. Cisco Unified Communications Manager identifies a call that is to be routed through a defined hunt list, Cisco Unified Communications Manager finds the first available device on the basis of the order of the line groups that a hunt list defines. Cisco Unified Communications Manager 9.x and later allows configuring of automatic log out of a hunt member when there is no answer. For more information about hunt pilots,see the System Configuration Guide for Cisco Unified Communications Manager. To use this feature configuration parameter *EnableHuntGroup* should be added and set to true in jabber-config.xml.

Call Pickup

The Call Pickup feature allows users to answer calls that come in on a directory number other than their own. Directory numbers are assigned to call pickup groups and Cisco Unified Communications Manager automatically dials the appropriate call pickup group number. Users select Pickup to answer the call.
Group call pickup allows users to pick up incoming calls in another group. Users enter the group pickup number, select Pickup and Cisco Unified Communications Manager automatically dials the appropriate call pickup group number.
Other group pickup allows users to pick up incoming calls in a group that is associated with their group. When the user selects Other Pickup Cisco Unified Communications Manager automatically searches for the incoming call in the associated groups.
Directed call pickup allows users to pick up an incoming call on a directory number. Users enter the directory number, select Pickup and Cisco Unified Communications Manager connects the incoming call.
For more information about configuring call pickup, see the Feature Configuration Guide for Cisco Unified Communications Manager .
To use Call pickup feature configuration parameters *EnableCallPickup,EnableGroupCallPikcup and EnableOtherGroupPickup* should be added and set to true in jabber-config.xml

Node Component

The Cisco Jabber Voice and Video SDK node component—referred to as *node.csf*—is an optional, server-side component that brings the following advanced features to clients:

- Cisco Unified Communications Manager end-user authentication
- Phone device configuration retrieval
- Directory integration

node.csf exposes an HTTP REST API. The reference implementation is based on Node.js. node.csf is made up of independent modules. The reference implementation contains three modules:

- Hello:
  - Replies helloto clients
  - Used mainly for testing
- PhoneConfig:
  - Authenticates Cisco Unified Communications Manager end users
  - Retrieves list of users' phone devices
  - Downloads phone device configuration
- QuickContact—integrates with a directory to search for contacts

Install the Node Component

node.csf is based on the stable node.js version 0.8.*x* or later. node.csf currently works on Unix-based operating systems. It was tested on Linux and Mac OS X.

Procedure

Step 1 Install the latest stable node.js version on a Unix-based operating system.
Tip: Some Linux distributions have binary packages (`nodejs`) or the source code can be downloaded and built directly. Go to http://nodejs.org/#download

Step 2 To test node.js, run the command: node -v.
The version of `node.js` appears:

```
node -v
v0.10.5
```

Step 3 node.js has a package manager called node package manager (npm). node.csf is provided as a npm package. To install npm, go to https://www.npmjs.org/.

Step 4 To test that npm is installed successfully, run the command: `npm -v`.
The version of npm appears:

```
npm -v
1.0.105
```

Step 5 Installing the csf package requires the computer to be connected to the Internet temporarily so that extra dependencies can be obtained from the public npm repository. The `csf-x.y.x-rev.tgz` file is included in `CiscoJabberSDK.zip`. Download the zip file from the Jabber Web SDK Download and Docs page.

Step 6 The installation builds a native component automatically, which requires the gcc compiler to be available. To test whether the gcc compiler is installed, run the command: `gcc -v`.

Step 7 If the gcc compiler is not installed, refer to the OS documentation for information about installing compilers.
Tip: Some Linux distributions have build packages ready to be installed, such as the build-essential package on Ubuntu

Step 8 Install the csf package by running one of the followings commands:

```
$ npm install csf-3.0.0-61003.tgz
```

If the machine is behind a proxy, use the --proxyoption, for example:

```
$ npm --proxy http://myproxy.com:8080 install csf-3.0.0-61003.tgz
```

Step 9 To test whether node.csf is installed, do one of the following:

1. Run the command: `$ node node_modules/csf/csf.js.`
2. Open http://localhost:1789/helloin a web browser. node.csf replies.

Use a command-line tool such as cURL:

```
$ curl -X GET http://localhost:1789/hello
$ curl -X POST http://localhost:1789/hello -d "test"
```

This causes a server to start listening on port 1789 (default).

Node Component Configuration

node.csf can be configured using a text file that contains JSON. The main options are:

- Listening port (http or https)
- The execution environment:
    - *production* (recommended)
    - *beta* (dump exceptions with stack trace)
    - *development* (same as beta but with more logs)
- Some directories to be served statically (array of relative or absolute paths)

Cross-domain access Cross Origin Resource Sharing (CORS) is enabled by adding a corsproperty to the configuration file. The value is a JSON object passed to the connect-xcors module (empty object for default options).
node.csf also supports JSONP (JSON with padding), which provides another way of accessing cross-domain. This can be used with jQuery.ajax, for example, setting dataTypeto jsonp.

node.csf replies to a JSONP request by sending an object which can have either an errorproperty indicating a failure, or a dataproperty containing the requested data.

Secure http (HTTPS) can be enabled by adding a httpsproperty to the configuration file. The value is a JSON object with three properties: enabled(true or false), key, and certificate. A sample key and self-signed certificates are provided under the configdirectory. For more information, go to http://nodejs.org/api/ tls.html#tls_tls_ssl.
node.csf loads applicative modules specified in the modulesmap of the configuration file. Each module is configured using either a JSON object, which can be empty, or a string referring to an extra configuration file specific to the module (absolute path or relative to the main configuration file path).
The template of the configuration file for node.csf is included in the package, under node_modules/csf/config/csf.cfg.

```
{
  "port": 1789,
  "env": "production",
  "staticDirs": [],
  "cors": {},
  "https": {
    "enabled": false,
    "key": "node_modules/csf/config/csf-key.pem",
    "cert": "node_modules/csf/config/csf-cert-self.pem"
  },
  "modules": {
      "hello": {},
      "phoneconfig": {/*For more information, see Node Component Modules.*/},
      "quickcontact": {/*For more information, see Node Component Modules.*/}
  }
}
```

Related Topics

Node Component Modules, on page 30

Run the Node Component

Procedure

Step 1 To start node.csf with the default configuration, change to the installation directory and run the following command:

```
$ node node_modules/csf/csf.js
```

Step 2 To specify a configuration, enter a command similar to the following:

```
$ node node_modules/csf/csf.js myconfigfile.cfg
```

Step 3 To run node.csf in the background, use the **nohup** command. For example:

```
$ nohup node node_modules/csf/csf.js myconfigfile.cfg > node.out 2> node.err < /dev/null &
```

Tip: You can also use the Forever utility. For more information, go to http://blog.nodejitsu.com/ keep-a-nodejs-server-up-with-forever/

Node Component Modules

The following tables:

- Describe the modules in the node component.
- Provide examples of the configuration file for the node component modules.
-

Table 1: Node Component Modules

Module	Description	API Exposed

Hello	• Used mainly for testing.
	• Replies `hello` to any received request.
	• The module is always loaded, even if not listed in the configuration modules.
		`GET /hello`
		Replies Hellowith the current date and time
		`POST /hello`
		Replies with `Hello {post data}`

PhoneConfig	Interacts with Cisco Unified Communications Manager to provide phone configuration to the Cisco Jabber SDK	`GET /phoneconfig/devices?ccmcip={ccmcip}`
		• Authenticates (HTTP Basic) end user against Cisco Unified Communications Manager with address *{ccmcip}*
		Important If your Cisco Unified Communications Manager server uses a self-signed certificate, to use this module you must add the `rejectUnauthorized` parameter to the configuration file. This parameter allows connections to servers with untrusted certificates.
		• Returns a JSON array with all phone devices associated to the user. Each device is a JSON object with the properties:
		• name: device name (String)
		• model: device model (String)
		• description: device description (String)
		`GET /phoneconfig/device/user/{user}/client/cwc?tftp={tftp}`
		Returns the phone device configuration (XML) associated to user *{user}* and client type **cwc**. Configuration is a JSON object with the properties:
		• name: device name (String)
		• tftp: the successful TFTP address (String)
		• configuration: a string containing the XML `<device>...</device>`

Module	Description	API Exposed

QuickContact	• Integrates with a directory to provide contact information to clients.
	• Currently supports LDAP to search for a directory.
	• Uses ldapjs to connect to the directory For more information, go to http://ldapjs.org/.
	
	The QuickContact module must be configured with:
	• The LDAP server address
	• The LDAP principal credentials (read only is acceptable)
	• The search tree base
	• The attributes mapping
	
	The `photosrc` property can contain the *{screenName}* pattern to be replaced with the screenName value, for example: "`photosrc`": "`http://mydirectory.com/photo/{screenName}.jpg`"
		`GET /quickcontact/name/{name}?max={max}&photo={photo}`
		• Searches contacts whose screen, first, last, and display name starts with {name}.
		• *{max}* is the maximum number of contacts to return. *0* means unlimited (default), absolute max is 20.
		• If *{photo}* equals *yes*, each result includes a photosrc property.
		• Returns an array (possibly empty) of contact objects.
		
		`GET /quickcontact/email/{email}?max={max}&photo={photo}`
		• Searches contacts whose email address starts with *{email}*.
		• *{max}* is the maximum number of contacts to return. *0* means unlimited (default), absolute max is 20.
		• If *{photo}* equals *yes*, each result includes a `photosrc` property.
		• Returns an array (possibly empty) of contact objects.
		
		`GET /quickcontact/phoneNumber/{phoneNumber} ?cc={cc}&max={max}&photo={photo}`
		• Searches contacts whose phone number starts with *{phoneNumber}* (sequence of digits).
		• If *{cc}* equals *yes*, *{phoneNumber}* starts with a country code (false by default).
		• *{max}* is the maximum number of contacts to return. *0* means unlimited (default), absolute max is 20.
		• If *{photo}* equals *yes*, each result includes a photosrc property.
		• Returns an array (possibly empty) of contact objects.
		• *{phoneNumber}* should contain as many digits as possible to narrow the search down to fewer contacts.
		
		Note Search by phone number is not fully optimized yet and can result in large requests on the directory (LDAP) server.

Table 2: Node Component Modules Configuration File Examples

Module	Configuration file example

PhoneConfig	"phoneconfig": {
	"cucmCluster": {
	"firstCluster": {
	"primaryNode": "192.168.1.1",
	"nodes": ["192.168.1.1", "192.168.1.2"],
	"tftpNodes": ["192.168.1.1"]
	}
	},
	"cucmClusterDnRule": [
	["1234XXX","firstCluster"],
	["+12345678XXX","firstCluster"]
	]
	"CertificateAuthorization": {
	"rejectUnauthorized": false
	}
	
	},
	
	Example:
	
	GET /phoneconfig/cucmCluster/name/firstCluster
	{
	“primaryNode”: “192.168.1.1”,
	“nodes”: [“192.168.1.1”, “192.168.1.2”],
	“tftpNodes”: “192.168.1.1"
	}

Module	Configuration file example

QuickContact	"quickcontact": {
	"ldap": {
	"host": "192.168.1.1",
	"port": 1234,
	"credentials": {
	"user":"CN=cn,OU=ou,DC=dc,DC=dc2",
	"password":”12345"
	},
	"treebase": "OU=ou,OU=ou2,DC=dc,DC=dc2",
	"mapping": {
	"screenName":"screenName",
	"displayName": "display",
	"email": "mail",
	"phoneNumbers":{
	"work": "workNumber",
	"mobile":"mobileNumber"
	},
	"photosrc":
	"http://www.company.com/dir/photos/{screenName}.jpg",
	}
	}
	}
	
	Example:
	
	GET /quickcontact/name/jsmith?photo=yes
	{
	"screenName": "jsmith",
	"displayName": "John Smith",
	"email": "jsmith@company.com",
	"phoneNumbers": {
	"work": "+1 234 56 78",
	"mobile": "+9 876 54 32"
	},
	"photosrc": "http://www.company.com/dir/photo/jsmith.jpg"
	}

Directory Navigator Sample Application

The Directory Navigator is a single-page sample application that allows users to search a user directory from a web page. If your application is using the Cisco Jabber SDK node component for directory integration, the Directory Navigator allows you to add search and retrieval of directory profiles. The sample application illustrates the use of the node REST API, and can be used as an example to build on and incorporate into your web application.

The Directory Navigator:

- Is built on the Cisco Jabber SDK node component.
- Uses the quickcontact node module to search the directory and retrieve profiles.
- Can be used with the Chrome, Firefox, and Safari browsers. There are known display issues when the application is used with Internet Explorer browser.

The Directory Navigator user interface uses the Twitter Bootstrap library. The interface is fully responsive and adapts to different screen sizes for smartphones, tablets, and desktops.

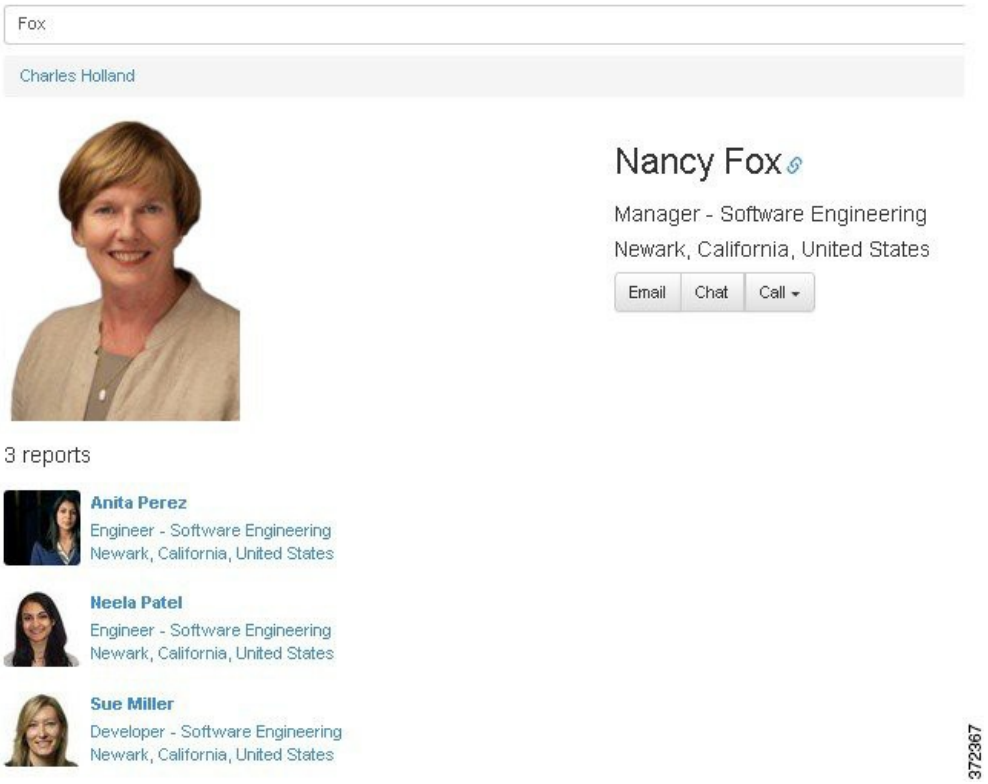
Users search the directory by entering a name in the search bar at the top of the application. As shown in the following figure, for each contact that matches the search criteria, the following information appears:

- Contact photo
- Display name
- Number of direct reports (in parentheses next the display name)
- Job title
- Location

Figure 1: Directory Navigator Search Results

When users click one of the search results, the contact profile appears, as shown in the following figure.

Figure 2: Directory Navigator Contact Profile



The contact profile includes the following additional information:

- Links to up to two levels of managers (the names of the managers appear above the contact photo)
- Buttons that allow users to:
    - Send an email.
    - Initiate a Cisco Jabber chat or call. Cisco Jabber must be installed locally.
- Links to the direct reports of the contact

Set Up Directory Navigator

The Directory Navigator queries the node quickcontact REST API to search contacts by name and retrieve contact information. If the serveSamples configuration parameter is set to true, the Directory Navigator is served automatically by the SDK node at `/quickcontact/samples/dirnavigator.html`.

```
{
    …
    "modules": {
        "quickcontact": {
            "serveSamples": true,
            …
        }
    }
}
```

The Directory Navigator expects the following contact keys mapped to LDAP attributes in the SDK node configuration file:

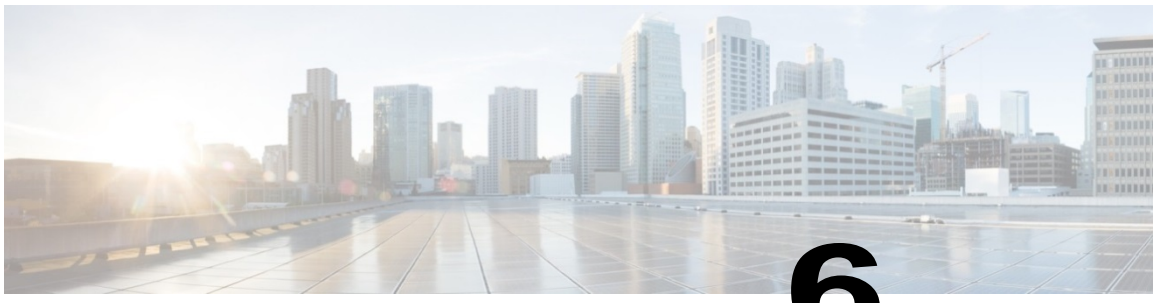
```
"mapping": {
    "screenName":"<ldap>",
    "displayName": "<ldap>",
    "email": "<ldap>",
    "phoneNumbers": {
         "work": "<ldap>",
         "mobile":"<ldap>"
    },
    "photosrc": "http://mydir.com/photos/{screenName}.jpg",
    "title": "<ldap>",
    "city": "<ldap>",
    "state": "<ldap>",
    "country": "<ldap>",
    "distinguishedName": "<ldap>",
    "manager": "<ldap>",
    "directReports": "<ldap>"
}
```

Note

- The `photosrc` attribute is an image source URL used for contact photos. The URL can contain a {`screenName`} token expanded to the actual screen name. The URL should be reachable from clients (browsers).
- The `phoneNumbers` value must contain an object whose keys are phone type and values are phone numbers. The application capitalizes phone types.
- The `distinguishedName` and `manager` values must contain an LDAP fully qualified distinguished name (FQDN).
- The `directReports` attribute must contain an array of FQDNs.

Contact result example:

```
"mapping": {
    "screenName":"nfox",
    "displayName": "Nancy Fox",
    "email": "nfox@example.com",
    "phoneNumbers":{
        "work": "+1 408 555 1514",
        "mobile":"+1 408 555 1515"
    },
    "photosrc": "http://mydir.com/photos/nfox.jpg",
    "title": "Manager - Software Engineering",
    "city": "Newark",
    "state": "CA",
    "country": "USA",
    "distinguishedName": "CN=nfox, DC=mycompany, DC=com",
    "manager": "CN=boss, DC=example, DC=com",
    "directReports": [
         "CN=aperez, DC=example, DC=com",
         "CN=npatel, DC=example, DC=com",
    ]}
```



CHAPTER 6

Troubleshooting

Debugging JavaScript

If you experience problems while using the Cisco Jabber SDK your first step is to open up the JavaScript console and other debugging tools in your browser.

Chrome

Chrome includes a good set of tools. Click the **Spanner** icon and choose **Tools** > **JavaScript Console**.

Firefox

Firefox includes a good set of tools. Choose **Tools** > **Web Developer** > **Web Console**.

Internet Explorer

Choose **Tools** > **Developer Tools** (or press F12). Click the **Script** tab to open the Console view in the right pane. The Console view displays any error messages from the Cisco Jabber SDK or other errors that may be causing problems.

Safari

By default, the developer tools in Safari are hidden. To activate the tools, choose **Preferences** > **Advanced** and check the option **Show Develop menu in menu bar**. This activates the Develop menu which contains several tools. To display the console, select **Show Error Console**.

JavaScript Events for Microsoft Windows Video Window

On Microsoft Windows, the video window object created by **cwic** does not generate HTML mouse and click events. A workaround is to attach the video window element to an invisible `<div>` (`style="display: none"`)and capture events from that element. Also on Microsoft Windows, HTML elements like buttons will not display on top of the video windows; the video windows always render on top. These limitations do not apply to video windows on Mac.

Disable App Nap

We recommend that App Nap, if the Mac OS X version being used has that feature, be disabled for all browsers supported by the Cisco Jabber Web SDK.

Procedure

Step 1 For Chrome and Firefox:

a) In Finder, right-click the browser's .appfile and click **Get Info**.

b) Check the **Prevent App Nap** check box.

Step 2 For Safari:

a) Open Terminal.

b) Run the following command:

```
defaults write -app Safari NSAppSleepDisabled -bool YES
```

Uninstall Cisco Web Communicator Add-on

Procedure

Do one of the following:

Option	Description

Mac	Run the following commands from a terminal window:
	`sudo rm -rf /Library/Internet\ Plug-Ins/CiscoWebCommunicator.plugin/ sudo pkgutil --forget com.cisco.pkg.CiscoWebCommunicator`
	
	For Mac OS X 10.10, after running the previous commands, run the following commands:
	`sudo rm –rf ~/Library/Application\\ Support/CiscoWebCommunicator`
	`defaults delete com.cisco.CiscoWebCommunicator`

Windows	1 From the **Control Panel**, click **Program and Features**. 2 Click **Cisco Web Communicator**. 3 Click **Uninstall**. 4 Follow the instructions on the screen.

Log Files

The add-on the SDK uses to enable telephony writes log files to your local file system. You can find the log files in the following locations:

- Windows:
  C:\Users\{username}\AppData\Local\Cisco\Unified Communications\Jabber Web SDK\Logs
- Mac OS X:
- /Users/{username}/Library/Logs/Jabber Web SDK/

While it is not really intended as a diagnostic tool for SDK users, softphone.log contains a wealth of detailed information and can be useful when you contact Cisco support.