Configure the Clients

Introduction to Client Configuration

Cisco Jabber can retrieve configuration settings from the following sources:

- Service Profiles — You can configure some client settings in UC service profiles on Cisco Unified Communications Manager release 9 and later. When users launch the client, it discovers the Cisco Unified Communications Manager home cluster using a DNS SRV record and automatically retrieves the configuration from the UC service profile.

  Applies to on-premises deployments only.

- Phone Configuration — You can set some client settings in the phone configuration on Cisco Unified Communications Manager release 9 and later. The client retrieves the settings from the phone configuration in addition to the configuration in the UC service profile.

Applies to on-premises deployments only.

- Cisco Unified Communications Manager IM and Presence Service — You can enable instant messaging and presence capabilities and configure certain settings such as presence subscription requests.

  In the **Advanced settings** window, if you select either **Cisco IM & Presence** or **Cisco Communications Manager 8.x**, the client retrieves UC services from Cisco Unified Presence or Cisco Unified Communications Manager IM and Presence Service. The client does not use service profiles or SSO discovery.

  Applies to on-premises deployments only.

- Client Configuration Files — You can create XML files that contain configuration parameters. You then host the XML files on a TFTP server. When users sign in, the client retrieves the XML file from the TFTP server and applies the configuration.

  Applies to on-premises and cloud-based deployments.

- Cisco Webex Administration Tool — You can configure some client settings with the Cisco Webex Administration Tool.

  Applies to cloud-based deployments only.

Configure Service Profiles

You can configure some client settings in UC service profiles on Cisco Unified Communications Manager version 9 and later.

Important

- Cisco Jabber only retrieves configuration from service profiles on Cisco Unified Communications Manager if the client gets the `_cisco-uds` SRV record from a DNS query.

  In a hybrid environment, if the CAS URL lookup is successful Cisco Jabber retrieves the configurations from Cisco WebEx Messenger service and the `_cisco-uds` SRV record is ignored.

- In an environment with multiple Cisco Unified Communications Manager clusters, you can configure the Intercluster Lookup Service (ILS). ILS enables the client to find the user's home cluster and discover services.

  If you do not configure ILS, then you must manually configure remote cluster information, similar to the EMCC remote cluster set up. For more information on Remote Cluster Configuration, see the *Cisco Unified Communications Manager Features and Services Guide*.

Related Topics

Remote Cluster Configuration on Cisco Unified Communications Manager 10.0

Set Parameters on Service Profile

The client can retrieve UC service configuration and other settings from service profiles.

Parameters in Service Profiles

Learn which configuration parameters you can set in service profiles. Review the corresponding parameters in the client configuration file.

IM and Presence Service Profile

The following table lists the configuration parameters you can set in the IM and Presence Service profile:

Parameter	Description
Product type	Provides the source of authentication to Cisco Jabber and has the following values: • Unified CM (IM and Presence Service) — Cisco Unified Communications Manager IM and Presence Service is the authenticator. • WebEx (IM and Presence Service) — The Cisco WebEx Messenger service is the authenticator. **Note** As of this release, the client issues an HTTP query in addition to the query for SRV records. The HTTP query allows the client to determine if it should authenticate to the Cisco WebEx Messenger service. As a result of the HTTP query, the client connects to the Cisco WebEx Messenger service in cloud-based deployments before getting the `_cisco-uds` SRV record. Setting the value of the **Product type** field to **WebEx** may have no practical effect if the WebEx service has already been discovered by a CAS lookup. • Not set — If the service profile does not contain an IM and presence service configuration, the authenticator is Cisco Unified Communications Manager.

Parameter	Description
Primary server	Specifies the address of your primary presence server. • On-Premises Deployments — You should specify the fully qualified domain name (FQDN) of Cisco Unified Communications Manager IM and Presence Service. • Cloud-Based Deployments — The client uses the following URL as default when you select **WebEx** as the value for the **Product type** parameter: `https://loginp.webexconnect.com/cas/auth.do` This default URL overrides any value that you set.

Voicemail Profile

The following table lists the configuration parameters you can set in the voicemail profile:

Parameter	Description
Voicemail server	Specifies connection settings for the voicemail server.
Credentials source for voicemail service	Specifies that the client uses the credentials for the instant messaging and presence or conferencing service to authenticate with the voicemail service. Ensure that the credentials source that you set match the user's voicemail credentials. If you set a value for this parameter, users cannot specify their voicemail service credentials in the client user interface.

Conferencing Profile

The following table lists the configuration parameters you can set in the conferencing profile:

Conferencing Service Configuration	Description
Conferencing server	Specifies connection settings for the conferencing server.
Credentials source for web conference service	Specifies that the client uses the credentials for the instant messaging and presence or voicemail service to authenticate with the conferencing service. Ensure that the credentials source that you set match the user's conferencing credentials.

Directory Profile

See the *Client Configuration for Directory Integration* chapter for information about configuring directory integration in a service profile.

CTI Profile

The following table lists the configuration parameters you can set in the CTI profile:

CTI Service Configuration	Description
CTI server	Specifies connection settings for the CTI server.

Add Cisco Unified Communications Manager Services

Add Cisco Unified Communications Manager services to specify the address, ports, protocols, and other settings for services such as IM and Presence Service, voicemail, conferencing, and directory.

Procedure

Step 1 Open the **Cisco Unified CM Administration** interface.

Step 2 Select **User Management** > **User Settings** > **UC Service**.

The **Find and List UC Services** window opens.

Step 3 Select **Add New**.

The **UC Service Configuration** window opens.

Step 4 Select the UC service type you want to add and then select **Next**.

Step 5 Configure the UC service as appropriate and then select **Save**.

What to do next

Add your UC services to service profiles.

Create Service Profiles

After you add and configure Cisco Unified Communications Manager services, you add them to a service profile. You can apply additional configuration in the service profile.

Procedure

Step 1 Open the **Cisco Unified CM Administration** interface.

Step 2 Select **User Management** > **User Settings** > **Service Profile**.

The **Find and List UC Services** window opens.

Step 3 Select **Add New**.

The **Service Profile Configuration** window opens.

Step 4 Enter a name for the service profile in the **Name** field.

Step 5 Select **Make this the default service profile for the system** if you want the service profile to be the default for the cluster.

Note On Cisco Unified Communications Manager release 9.x only, users who have only instant messaging capabilities (IM only) must use the default service profile. For this reason, you should set the service profile as the default if you plan to apply the service profile to IM only users.

Step 6 Add your UC services, apply any additional configuration, and then select **Save**.

What to do next

Apply service profiles to end user configuration.

Apply Service Profiles

After you add UC services and create a service profile, you apply the service profile to users. When users sign in to Cisco Jabber, the client can then retrieve the service profile for that user from Cisco Unified Communications Manager.

Procedure

Step 1 Open the **Cisco Unified CM Administration** interface.

Step 2 Select **User Management** > **End User**.

The **Find and List Users** window opens.

Step 3 Enter the appropriate search criteria to find existing users and then select a user from the list.

The **End User Configuration** window opens.

Step 4 Locate the **Service Settings** section.

Step 5 Select a service profile to apply to the user from the **UC Service Profile** drop-down list.

Important **Cisco Unified Communications Manager release 9.x only:** If the user has only IIM and Presence Service capabilities (IM only), you must select **Use Default**. For IM only users, Cisco Unified Communications Manager release 9.x always applies the default service profile regardless of what you select from the **UC Service Profile** drop-down list.

Step 6 Apply any other configuration as appropriate and then select **Save**.

Associate Users with Devices

On Cisco Unified Communications Manager version 9.x only, when the client attempts to retrieve the service profile for the user, it first gets the device configuration file from Cisco Unified Communications Manager. The client can then use the device configuration to get the service profile that you applied to the user.

For example, you provision Adam McKenzie with a CSF device named `CSFAKenzi`. The client retrieves `CSFAKenzi.cnf.xml` from Cisco Unified Communications Manager when Adam signs in. The client then looks for the following in `CSFAKenzi.cnf.xml`:

```
<userId  serviceProfileFile="identifier.cnf.xml">amckenzi</userId>
```

For this reason, if you are using Cisco Unified Communications Manager version 9.x, you should do the following to ensure that the client can successfully retrieve the service profiles that you apply to users:

- Associate users with devices.
- Set the **User Owner ID** field in the device configuration to the appropriate user. The client will retrieve the Default Service Profile if this value is not set.

Note A CSF should not be associated to multiple users if you intend to use different service profiles for these users.

Procedure

Step 1 Associate users with devices.

a) Open the **Unified CM Administration** interface.
b) Select **User Management** > **End User**.
c) Find and select the appropriate user.
The **End User Configuration** window opens.
d) Select **Device Association** in the **Device Information** section.
e) Associate the user with devices as appropriate.
f) Return to the **End User Configuration** window and then select **Save**.

Step 2 Set the **User Owner ID** field in the device configuration.

a) Select **Device** > **Phone**.
b) Find and select the appropriate device.
The **Phone Configuration** window opens.
c) Locate the **Device Information** section.
d) Select **User** as the value for the **Owner** field.
e) Select the appropriate user ID from the **Owner User ID** field.
f) Select **Save**.

Set Parameters on Phone Configuration for Desktop Clients

The client can retrieve configuration settings in the phone configuration from the following locations on Cisco Unified Communications Manager:

Enterprise Phone Configuration

Applies to the entire cluster.

Note For users with only IM and Presence Service capabilities (IM only), you must set phone configuration parameters in the **Enterprise Phone Configuration** window.

Common Phone Profile Configuration

Applies to groups of devices and takes priority over the cluster configuration.

Cisco Unified Client Services Framework (CSF) Phone Configuration

Applies to individual CSF devices and takes priority over the group configuration.

Parameters in Phone Configuration

The following table lists the configuration parameters you can set in the **Product Specific Configuration Layout** section of the phone configuration and maps corresponding parameters from the client configuration file:

Desktop Client Settings Configuration	Description
Video Calling	Enables or disables video capabilities. **Enabled (default)** Users can send and receive video calls. **Disabled** Users cannot send or receive video calls. **Restriction** This parameter is available only on the CSF device configuration.
File Types to Block in File Transfer	Restricts users from transferring specific file types. Set a file extension as the value, for example, `.exe`. Use a semicolon to delimit multiple values, for example, `.exe;.msi;.rar;.zip`
Automatically Start in Phone Control	Sets the phone type for users when the client starts for the first time. Users can change their phone type after the initial start. The client then saves the user preference and uses it for subsequent starts. **Enabled** Use the desk phone device for calls. **Disabled (default)** Use the software phone (CSF) device for calls.
Jabber For Windows Software Update Server URL	Specifies the URL to the XML file that holds client update information. The client uses this URL to retrieve the XML file from your web server. In hybrid cloud-based deployments, you should use the Cisco WebexAdministration Tool to configure automatic updates.
Problem Report Server URL	Specifies the URL for the custom script that allows users to submit problem reports.

Set Parameters on Phone Configuration for Mobile Clients

The client can retrieve configuration settings in the phone configuration from the following locations on Cisco Unified Communications Manager:

- Cisco Dual Mode for iPhone (TCT) Configuration — Applies to individual TCT devices and takes priority over the group configuration.
- Cisco Jabber for Tablet (TAB) Configuration — Applies to individual TAB devices and takes priority over the group configuration.

Parameters in Phone Configuration

The following table lists the configuration parameters you can set in the **Product Specific Configuration Layout** section of the phone configuration and maps corresponding parameters from the client configuration file:

Parameter	Description
On-Demand VPN URL	URL for initiating on-demand VPN. **Note** Applicable for iOS only.
Preset Wi-fi Networks	Enter the SSIDs for Wi-Fi networks (SSIDs) approved by your organization. Separate SSIDs with a forward slash (/). Devices do not connect to secure connect if connected to one of the entered Wi-Fi networks.
Default Ringtone	Sets the default ringtone to **Normal** or **Loud**.
Video Capabilities	Enables or disables video capabilities. • Enabled (default) — Users can send and receive video calls. • Disabled — Users cannot send or receive video calls.
Dial via Office **Note** TCT and BOT devices only.	Enables or disables Dial via Office. • Enabled — Users can dial via office. • Disabled (default) — Users cannot dial via office.

Create and Host Client Configuration Files

In on-premises and hybrid cloud-based deployments you can create client configuration files and host them on the Cisco Unified Communications Manager TFTP service.

In cloud-based deployments, you should configure the client with the Cisco WebEx Administration Tool. However, you can optionally set up a TFTP server to configure the client with settings that are not available in Cisco WebEx Administration Tool.

Important In most environments, the client does not require any configuration to connect to services. You should create a configuration file only if you require custom content such as:

- Automatic updates
- Problem reporting
- User policies and options

Important You must create a global configuration file to set up:

- Directory integration for on-premises deployments.
- Voicemail service credentials for hybrid-cloud deployments.

Client Configuration Files

Before you deploy configuration files, review the differences between global and group configuration files. To successfully deploy configuration files you should also review the requirements for configuration files such as supported encoding.

Review details about configuration files and understand requirements such as supported encoding.

Global Configuration Files

Global configuration files apply to all users. The client downloads the global configuration file from your TFTP server during the login sequence.

The default name for the global configuration file is `jabber-config.xml`.

Group Configuration Files

Note

- Group configuration files are supported on Cisco Jabber for Windows and on Cisco Jabber for mobile devices.
- Group configuration files apply to subsets of users. Group configuration files take priority over global configuration files.

Group Configuration File Names

You specify the name of the group configuration files in the **Cisco Support Field** on the CSF, BOT, TCT, or TAB device configuration in Cisco Unified Communications Manager.

If you remove the name of the group configuration file in the CSF device configuration on Cisco Unified Communications Manager, the client detects the change, prompts the users to sign out, and loads the global configuration file. You can remove the name of the group configuration file in the CSF, BOT, TCT, or TAB

device configuration by deleting the entire `configurationFile=group_configuration_file_name.xml` string or by deleting the group configuration filename from the string.

Configuration File Requirements

- Configuration filenames are case sensitive. Use lowercase letters in the filename to prevent errors and to ensure the client can retrieve the file from the TFTP server.
- You must use utf-8 encoding for the configuration files.
- The client cannot read configuration files that do not have a valid XML structure. Ensure you check the structure of your configuration file for closing elements and that elements are nested correctly.
- Your XML can contain only valid XML character entity references. For example, use `&` instead of `&`. If your XML contains invalid characters, the client cannot parse the configuration file.

Tip Open your configuration file in Microsoft Internet Explorer to see if any characters or entities are not valid.

If Internet Explorer displays the entire XML structure, your configuration file does not contain invalid characters or entities.

If Internet Explorer displays only part of the XML structure, your configuration file most likely contains invalid characters or entities.

Specify Your TFTP Server Address

The client gets configuration files from a TFTP server. The first step in configuring the client is to specify your TFTP server address so the client can access your configuration file.

Attention If Cisco Jabber gets the `_cisco-uds` SRV record from a DNS query, it can automatically locate the user's home cluster. As a result, the client can also locate the Cisco Unified Communications Manager TFTP service.

You do not need to specify your TFTP server address if you deploy the `_cisco-uds` SRV record.

Specify Your TFTP Server on Cisco Unified Presence

If you are using Cisco Unified Communications Manager release 8.x, complete the steps to specify the address of your TFTP server on Cisco Unified Presence. If you are using Cisco Unified Communications Manager release 9.x, then you do not need to follow the steps below.

Procedure

Step 1 Open the **Cisco Unified Presence Administration** interface.

Step 2 Select **Application** > **Cisco Jabber** > **Settings**.

Note In some versions of Cisco Unified Presence, this path is as follows: **Application** > **Cisco Unified Personal Communicator** > **Settings**.

The **Cisco Jabber Settings** window opens.

Step 3 Locate the fields to specify TFTP servers in one of the following sections, depending on your version of Cisco Unified Presence:

- **Cisco Jabber Security Settings**
- **CUPC Global Settings**

Step 4 Specify the IP address of your primary and backup TFTP servers in the following fields:

- **Primary TFTP Server**
- **Backup TFTP Server**
- **Backup TFTP Server**

Note Ensure that you enter the fully qualified domain name (FQDN) or IP address for the TFTP servers rather than a host name.

Step 5 Select **Save**.

Specify Your TFTP Server on Cisco Unified Communications Manager IM and Presence Service

If you are using Cisco Unified Communications Manager release 9.x, then you do not need to follow the steps below.

Procedure

Step 1 Open the **Cisco Unified CM IM and Presence Administration** interface.

Step 2 Select **Application** > **Legacy Clients** > **Settings**.

The **Legacy Client Settings** window opens.

Step 3 Locate the **Legacy Client Security Settings** section.

Step 4 Specify the IP address of your primary and backup TFTP servers in the following fields:

- **Primary TFTP Server**
- **Backup TFTP Server**
- **Backup TFTP Server**

Step 5 Select **Save**.

Specify TFTP Servers in Phone Mode

If you deploy the client in phone mode you can provide the address of the TFTP server as follows:

- Users manually enter the TFTP server address when they start the client.

- You specify the TFTP server address during installation with the TFTP argument.
- You specify the TFTP server address in the Microsoft Windows registry.

Specify TFTP Servers with the Cisco WebEx Administration Tool

If the client connects to the Cisco WebEx Messenger service, you specify your TFTP server address with the Cisco WebEx Administrator Tool.

Procedure

Step 1 Open the Cisco WebEx Administrator Tool.

Step 2 Select the **Configuration** tab.

Step 3 Select **Unified Communications** in the **Additional Services** section.
The **Unified Communications** window opens.

Step 4 Select the **Clusters** tab.

Step 5 Select the appropriate cluster from the list.
The **Edit Cluster** window opens.

Step 6 Select **Advanced Server Settings** in the **Cisco Unified Communications Manager Server Settings** section.

Step 7 Specify the IP address of your primary TFTP server in the **TFTP Server** field.

Step 8 Specify the IP address of your backup TFTP servers in the **Backup Server #1** and **Backup Server #2** fields.

Step 9 Select **Save**.
The **Edit Cluster** window closes.

Step 10 Select **Save** in the **Unified Communications** window.

Create Global Configurations

The client downloads the global configuration file from your TFTP server during the login sequence. Configure the client for all users in your deployment.

Before you begin

If the structure of your configuration file is not valid, the client cannot read the values you set. Review the XML samples in this chapter for more information.

Procedure

Step 1 Create a file named `jabber-config.xml` with any text editor.

- Use lowercase letters in the filename.
- Use UTF-8 encoding.

Step 2 Define the required configuration parameters in `jabber-config.xml`.

Step 3 Host the group configuration file on your TFTP server.

If your environment has multiple TFTP servers, ensure that the configuration file is the same on all TFTP servers.

Create Group Configurations

Group configuration files apply to subsets of users and are supported on Cisco Jabber for desktop (CSF devices) and on Cisco Jabber for mobile devices. Group configuration files take priority over global configuration files.

If you provision users with CSF devices, specify the group configuration filenames in the **Cisco Support Field** field on the device configuration. If users do not have CSF devices, set a unique configuration filename for each group during installation with the TFTP_FILE_NAME argument.

Before you begin

- If you have Cisco Unified Communications Manager 8.6, the **Cisco Support Field** field does not exist. Download the ciscocm.addcsfsupportfield.cop COP file from the Cisco Jabber administration package to your file system and deploy to Cisco Unified Communications Manager. For more information about deploying COP files, see the Cisco Unified Communications Manager documentation.

  The COP file adds the **Cisco Support Field** field to CSF devices in the **Desktop Client Settings** section on the **Phone Configuration** window.

- If the structure of your configuration file is not valid, the client cannot read the values you set. Review the XML samples in this chapter for more information.

Procedure

Step 1 Create an XML group configuration file with any text editor.

The group configuration file can have any appropriate name; for example, `jabber-groupa-config.xml`.

Step 2 Define the required configuration parameters in the group configuration file.

Step 3 Add the group configuration file to applicable CSF devices.

a) Open the **Cisco Unified CM Administration** interface.
b) Select **Device** > **Phone**.
c) Find and select the appropriate CSF device to which the group configuration applies.
d) In the **Phone Configuration** window, navigate to **Product Specific Configuration Layout** > **Desktop Client Settings**.
e) In the **Cisco Support Field** field, enter `configurationfile=`*`group_configuration_file_name.xml`*. For example, enter `configurationfile=groupa-config.xml`.

Note If you host the group configuration file on your TFTP server in a location other than the default directory, you must specify the path and the filename; for example, `configurationfile=/customFolder/groupa-config.xml`.

Do not add more than one group configuration file. The client uses only the first group configuration in the **Cisco Support Field** field.

f) Select **Save**.

Step 4 Host the group configuration file on your TFTP server.

Host Configuration Files

You can host configuration files on any TFTP server. However, Cisco recommends hosting configuration files on the Cisco Unified Communications Manager TFTP server, which is the same as that where the device configuration file resides.

Procedure

Step 1 Open the **Cisco Unified OS Administration** interface on Cisco Unified Communications Manager.

Step 2 Select **Software Upgrades** > **TFTP File Management**.

Step 3 Select **Upload File**.

Step 4 Select **Browse** in the **Upload File** section.

Step 5 Select the configuration file on the file system.

Step 6 Do not specify a value in the **Directory** text box in the **Upload File** section.

You should leave an empty value in the **Directory** text box so that the configuration file resides in the default directory of the TFTP server.

Step 7 Select **Upload File**.

Restart Your TFTP Server

You must restart your TFTP server before the client can access the configuration files.

Procedure

Step 1 Open the **Cisco Unified Serviceability** interface on Cisco Unified Communications Manager.

Step 2 Select **Tools** > **Control Center - Feature Services**.

Step 3 Select **Cisco Tftp** from the **CM Services** section.

Step 4 Select **Restart**.

A window displays to prompt you to confirm the restart.

Step 5 Select **OK**.

The **Cisco Tftp Service Restart Operation was Successful** status displays.

Step 6 Select **Refresh** to ensure the **Cisco Tftp** service starts successfully.

What to do next

To verify that the configuration file is available on your TFTP server, open the configuration file in any browser. Typically, you can access the global configuration file at the following URL:
`http://tftp_server_address:6970/jabber-config.xml`

Configuration File Structure

You create client configuration files in an XML format that contains the following elements

XML Declaration

The configuration file must conform to XML standards and contain the following declaration:

```
<?xml version="1.0" encoding="utf-8"?>
```

Root Element

The root element config, contains all group elements. You must also add the version attribute to the root element as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<config version="1.0">
</config>
```

Group Elements

Group elements contain configuration parameters and values. You must nest group elements within the root element.

Group Elements and Parameters

The following table describes the group elements you can specify in a client configuration file:

Element	Description
Client	Contains configuration parameters for the client.
Directory	Contains configuration parameters for directory integration.
Options	Contains configuration parameters for user options.
Phone	Contains configuration parameters for phone services.
Policies	Contains configuration parameters for policies.
Presence	Contains configuration parameters for presence options.
Voicemail	Contains configuration parameters for the voicemail service.

XML Structure

The following snippet shows the XML structure of a client configuration file:

```
<Client>
  <parameter>value</parameter>
</Client>
<Directory>
  <parameter>value</parameter>
</Directory>
<Options>
  <parameter>value</parameter>
</Options>
<Phone>
  <parameter>value</parameter>
</Phone>
<Policies>
  <parameter>value</parameter>
</Policies>
<Presence>
  <parameter>value</parameter>
</Presence>
<Voicemail>
  <parameter>value</parameter>
</Voicemail>
```

Summary of Configuration Parameters

The following table lists all the parameters you can include in the client configuration:

Parameter	Group Element
PrtLogServerUrl	Client
UpdateUrl	Client
jabber-plugin-config	Client
Forgot_Password_URL	Client
Persistent_Chat_Enabled	Client
Mention_P2Pchat	Client
Mention_GroupChat	Client
Mention_PersistentChat	Client
spell_check_enabled	Client
Disable_IM_History	Client
Set_Status_Away_On_Inactive	Options
Set_Status_Inactive_Timeout	Options
Set_Status_Away_On_Lock_OS	Options
StartCallWithVideo	Options
Start_Client_On_Start_OS	Options
AllowUserCustomTabs	Options
ShowContactPictures	Options

Parameter	Group Element
ShowOfflineContacts	Options
DockedWindowVisible	Options
DockedWindowPosition	Options
Callhistory_Expire_Days	Options
DeviceAuthenticationPrimaryServer	Phone
DeviceAuthenticationBackupServer	Phone
TftpServer1	Phone
TftpServer2	Phone
CtiServer1	Phone
CtiServer2	Phone
useCUCMGroupForCti	Phone
CcmcipServer1	Phone
CcmcipServer2	Phone
Meeting_Server_Address	Phone
Meeting_Server_Address_Backup	Phone
Meeting_Server_Address_Backup2	Phone
EnableDSCPPacketMarking	Phone
EnableVideo	Policies
InitialPhoneSelection	Policies
UserDefinedRemoteDestinations	Policies
enableLocalAddressBookSearch	Policies
EnableAccessoriesManager	Policies
BlockAccessoriesManagerPlugins	Policies
ForceFontSmoothing	Policies
Screen_Capture_Enabled	Policies
File_Transfer_Enabled	Policies
Disallowed_File_Transfer_Types	Policies
EnableBFCPVideoDesktopShare	Policies
Meetings_Enabled	Policies
Telephony_Enabled	Policies
Voicemail_Enabled	Policies
EnableTelProtocolHandler	Policies

Parameter	Group Element
EnableIMProtocolHandler	Policies
EnableSIPProtocolHandler	Policies
EnableSaveChatToFile	Policies
EnableSIPURIDialling	Policies
DirectoryURI BDIDirectoryURI	Policies
ForceC2XDirectoryResolution	Policies
ServiceDiscoveryExcludedServices	Policies
VoiceServicesDomain	Policies
ctiwindowbehaviour	Policies
EnableCallPickup	Policies
EnableGroupCallPickup	Policies
EnableOtherGroupPickup	Policies
EnableHuntGroup	Policies
PreventDeclineOnHuntCall	Policies
TelemetryEnabled	Policies
TelemetryCustomerID	Policies
TelemetryEnabledOverCellularData	Policies
EnableTelProtocolPopupWindow CiscoTelProtocolPermissionEnabled	Policies
EnableP2PDesktopShare	Policies
Customize_Phone_Server	Policies
Customize_Voicemail_Server	Policies
ServicesDomainSsoEmailPrompt	Policies
EnableForensicsContactData	Policies
LoginResource	Presence
PresenceServerAddress	Presence
PresenceServerURL	Presence
VoiceMailService_UseCredentialsFrom	Voicemail
VoicemailPrimaryServer	Voicemail

Related Topics

Group Elements and Parameters, on page 16
Client Parameters, on page 20

Example Configuration

The following is an example of a configuration file used in an on-premises deployment for all clients:

```
<?xml version="1.0" encoding="utf-8"?>
<config version="1.0">
 <Client>
  <PrtLogServerUrl>http://server_name:port/path/prt_script.php</PrtLogServerUrl>
  <jabber-plugin-config>
   <browser-plugin>
    <page refresh="true" preload="true">
     <tooltip>Cisco</tooltip>
     <icon>http://www.cisco.com/web/fw/i/logo.gif</icon>
     <url>www.cisco.com</url>
    </page>
   </browser-plugin>
  </jabber-plugin-config>
 </Client>
  <Options>
    <Set_Status_Inactive_Timeout>20</Set_Status_Inactive_Timeout>
    <StartCallWithVideo>false</StartCallWithVideo>
  </Options>
  <Policies>
    <Disallowed_File_Transfer_Types>.exe;.msi</Disallowed_File_Transfer_Types>
  </Policies>
  <Directory>
  <BDIPresenceDomain>example.com</BDIPresenceDomain>
    <BDIPrimaryServerName>dir.example.com</BDIPrimaryServerName>
    <BDISearchBase1>ou=staff,dc=example,dc=com</BDISearchBase1>
    <BDIConnectionUsername>ad_jabber_access@example.com</BDIConnectionUsername>
    <BDIConnectionPassword>jabber</BDIConnectionPassword>
    <BDIPhotoUriSubstitutionEnabled>True</BDIPhotoUriSubstitutionEnabled>
    <BDIPhotoUriSubstitutionToken>sAMAccountName</BDIPhotoUriSubstitutionToken>
    <BDIPhotoUriWithToken>http://example.com/photo/sAMAccountName.jpg
    </BDIPhotoUriWithToken>
  </Directory>
</config>
```

Client Parameters

The following table describes the parameters you can specify within the Client element:

Parameter	Value	Description
PrtLogServerUrl	URL	Specifies the custom script for submitting problem reports.

Parameter	Value	Description
UpdateUrl	URL	Specifies the URL to the automatic updates XML definition file on your HTTP server. The client uses this URL to retrieve the update XML file. In hybrid cloud-based deployments, you should use the Cisco WebEx Administration Tool to configure automatic updates.
jabber-plugin-config	Plug-in definition	Contains plug-in definitions such as custom embedded tabs that display HTML content.
Forgot_Password_URL	URL	Specifies the URL of your web page for users to reset or retrieve forgotten passwords. In hybrid cloud-based deployments, you should use the Cisco WebEx Administration Tool to direct users to the web page to reset or retrieve forgotten passwords.
Persistent_Chat_Enabled	true false	Specifies whether the Persistent Chat feature is available in the client. **true** If the value is set to true, the Persistent Chat interface is shown in the client. **false (default)** The default value is assumed if the setting is not present in the configuration file.
Mention_P2Pchat	true false	Specifies whether mentions are enabled in person to person chat. **true (default)** Enables mentions in person to person chat. **false** Disables mentions in person to person chat.
Mention_GroupChat	true false	Specifies whether mentions are enabled in group chat. **true (default)** Enables mentions in group chat. **false** Disables mentions in group chat.
Mention_PersistentChat	true false	Specifies whether mentions are enabled in persistent chat. **true (default)** Enables mentions in persistent chat. **false** Disables mentions in persistent chat.

Parameter	Value	Description
spell_check_enabled	true false	Specifies whether spell check is enabled in the client. Spell check supports autocorrect, allows users to select the correct word from a list of suggestions, and add the word to a dictionary. Spell check has the following limitations: • Cisco Jabber for Windows supports spell check in Windows 8. • Spell check uses the built-in functionality of the operating system language. If the Cisco Jabber for Windows client language is different to the operating system language then spell check uses the operating system language • The shortcut keys for undo and redo (Ctrl-Z and Ctrl-Y) do not work when spell check is enabled. **true** Spell check is enabled. **false (default)** Spell check is disabled.
Disable_IM_History	true false	Specifies whether to retain chat history after participants close the chat window. **Note** This parameter is not available for IM-only deployments. **true** Do not retain chat history after participants close the chat window. **false (default)** Retain chat history: • After participants close the chat window. • Until the participants sign out. If the participants re-open the chat window, the last 99 messages show. Message archiving should be disabled on the server.

Parameter	Value	Description
CachePasswordMobile	true false	Specifies whether the password is remembered or not on the client side. **true (default)** The password is prefilled and **Automatic sign-in** is shown. Users can allow the client to cache their password. This option allows users to automatically sign in when the client starts. **false** After the client successfully registers to the Cisco Unified Communications Manager, the password field is empty and **Automatic sign-in** is not shown. Users cannot allow the client to cache their password. Users must enter their password each time the client starts. **Note** The client displays **Automatic sign-in** on first sign-in, or if the user clears the application data.

Options Parameters

The following table describes the parameters you can specify within the Options element:

Parameter	Value	Description
Set_Status_Away_On_Inactive	true false	Specifies if the availability status changes to **Away** when users are inactive. **true (default)** Availability status changes to **Away** when users are inactive. **false** Availability status does not change to **Away** when users are inactive.
Set_Status_Inactive_Timeout	Number of minutes	Sets the amount of time, in minutes, before the availability status changes to **Away** if users are inactive. The default value is 15.

Parameter	Value	Description
Set_Status_Away_On_Lock_OS	true false	Specifies if the availability status changes to **Away** when users lock their operating systems. **true (default)** Availability status changes to **Away** when users lock their operating systems. **false** Availability status does not change to **Away** when users lock their operating systems.
StartCallWithVideo	true false	Specifies how calls start when users place calls. Calls can start with audio only or audio and video. **true (default)** Calls always start with audio and video. **false** Calls always start with audio only. **Important** Server settings take priority over this parameter in the client configuration file. However, if users change the default option in the client user interface, that setting takes priority over both the server and client configurations. Configure this setting on the server as follows: **Cisco Unified Presence** 1. Open the **Cisco Unified Presence Administration** interface. 2. Select **Application** > **Cisco Jabber** > **Settings**. 3. Select or clear the **Always begin calls with video muted** parameter and then select **Save**. **Cisco Unified Communications Manager version 9.x and higher** 1. Open the **Cisco Unified CM Administration** interface. 2. Select **System** > **Enterprise Parameters**. 3. Set a value for the **Never Start Call with Video** parameter and then select **Save**.

Parameter	Value	Description
Start_Client_On_Start_OS	true false	Specifies if the client starts automatically when the operating system starts. **true** The client starts automatically. **false (default)** The client does not start automatically.
AllowUserCustomTabs	true false	Specifies if users can create their own custom embedded tabs. **true (default)** Users can create custom embedded tabs. **false** Users cannot create custom embedded tabs. **Note** This parameter affects only custom embedded tabs that users create. • If you allow users to create custom embedded tabs, they cannot modify or remove the tabs that you define in the client configuration. • If you do not allow users to create custom embedded tabs, the tabs that you define are still available to users.
ShowContactPictures	true false	Specifies if contact pictures display in the contact list. **true (default)** Contact pictures display in the contact list. **false** Contact pictures do not display in the contact list.
ShowOfflineContacts	true false	Specifies if offline contacts display in the contact list. **true (default)** Offline contacts display in the contact list. **false** Offline contacts do not display in the contact list.

Parameter	Value	Description
DockedWindowVisible	TRUE FALSE	Specifies if the docked window displays when the client starts. **true (default)** The docked window displays when the client starts. **false** The docked window does not display when the client starts.
DockedWindowPosition	TopCenter TopLeft TopRight	Sets the position of the docked window on the user's screen. **TopCenter (default)** The position of the docked window is at the top center of the screen. **TopLeft** The position of the docked window is at the top left of the screen. **TopRight** The position of the docked window is at the top right of the screen.
Callhistory_Expire_Days	Number of days	Sets the number of days before the call history is deleted. If the value is 0 or not specified in the configuration file the call history is not deleted until the count exceeds the maximum number of stored calls.

Phone Parameters

The following table describes the parameters you can specify within the Phone element:

Parameter	Value	Description
DeviceAuthenticationPrimaryServer	Hostname IP address FQDN	Specifies the address of the primary instance of Cisco Unified Communications Manager to which users authenticate in phone mode deployments. Set one of the following as the value: • Hostname (*hostname*) • IP address (*123.45.254.1*) • FQDN (*hostname.domain.com*) This parameter can only be used in Cisco Jabber 9.6 and 9.7.

Parameter	Value	Description
DeviceAuthenticationBackupServer	Hostname IP address FQDN	Specifies the address of the backup instance of Cisco Unified Communications Manager to which users authenticate in phone mode deployments. Set one of the following as the value: • Hostname (*hostname*) • IP address (*123.45.254.1*) • FQDN (*hostname.domain.com*) This parameter can only be used in Cisco Jabber 9.6 and 9.7
TftpServer1	Hostname IP address FQDN	Specifies the address of the primary Cisco Unified Communications Manager TFTP service where device configuration files reside. Set one of the following as the value: • Hostname (*hostname*) • IP address (*123.45.254.1*) • FQDN (*hostname.domain.com*) You should set this parameter in the client configuration only if: • You deploy the client in phone mode. • The TFTP server address for the device configuration is different to the TFTP server address for the client configuration. During installation, you should set the address of the TFTP server where the client configuration file resides with the following argument: TFTP.
TftpServer2	Hostname IP address FQDN	Specifies the address of the secondary Cisco Unified Communications Manager TFTP service. This parameter is optional.
CtiServer1	Hostname IP address FQDN	Specifies the address of the primary CTI server. You should specify a CTI server address in the client configuration if users have desk phone devices.
CtiServer2	Hostname IP address FQDN	Specifies the address of the secondary CTI server. This parameter is optional.

Parameter	Value	Description
useCUCMGroupForCti	true false	Specifies if the Cisco Unified CM Group handles load balancing for CTI servers. Set one of the following values: **true** The Cisco Unified CM Group handles CTI load balancing. You should set this value in phone mode deployments only. In full UC mode, the presence server automatically handles CTI load balancing. **false (default)** The Cisco Unified CM Group does not handle CTI load balancing.
CcmcipServer1	Hostname IP address FQDN	Specifies the address of the primary CCMCIP server. This parameter is required: • Only if the address of your CCMCIP server is not the same as the TFTP server address. If the address of the CCMCIP server is the same as the TFTP server address, the client can use the TFTP server address to connect to the CCMCIP server. • In deployments with Cisco Unified Communications Manager version 8. In deployments with Cisco Unified Communications Manager version 9 and higher, the client can discover the CCMCIP server if you provision the `_cisco-uds` SRV record. Cisco Unified Communications Manager release 9.x and earlier—If you enable Cisco Extension Mobility, the `Cisco Extension Mobility` service must be activated on the Cisco Unified Communications Manager nodes that are used for CCMCIP. For information about Cisco Extension Mobility, see the *Feature and Services* guide for your Cisco Unified Communications Manager release.
CcmcipServer2	Hostname IP address FQDN	Specifies the address of the secondary CCMCIP server. This parameter is optional.

Parameter	Value	Description
Meeting_Server_Address	Cisco WebEx meetings site URL	Specifies the primary Cisco WebEx meeting site URL for users. The client populates this meeting site in the user's host account on the **Options** window. Users can enter their credentials to set up the host account and access their Cisco WebEx meetings, if the meeting site requires credentials. The client populates the meeting site in the user's host account on the **Preferences** > **Meetings** window. Users can enter their credentials to set up the host account and access their meetings site, if the Cisco WebEx meeting site requires credentials. **Important** If you specify an invalid meeting site, users cannot add, or edit, any meetings sites in the client user interface. This parameter is optional.
Meeting_Server_Address_Backup	Cisco WebEx meetings site URL	Specifies the secondary Cisco WebEx meeting site URL for users. This parameter is optional.
Meeting_Server_Address_Backup2	Cisco WebEx meetings site URL	Specifies the tertiary Cisco WebEx meeting site URL for users. This parameter is optional.
EnableDSCPPacketMarking	true false	Specifies if DSCP marking is applied to the packets: **true (default)** DSCP marking is enabled and the checkbox in the client is not shown. **false** DSCP marking is not made to packets and the checkbox in the client is not shown.

Related Topics

TFTP Server Address

Policies Parameters

Policies parameters let you control specific client functionality.

On-Premises Policies

The following table describes the parameters you can specify within the Policies element in on-premises deployments:

Parameter	Value	Description
Screen_Capture_Enabled	true false	Specifies if users can take screen captures. **true (default)** Users can take screen captures. **false** Users cannot take screen captures.
File_Transfer_Enabled	true false	Specifies if users can transfer files to each other. **true (default)** Users can transfer files to each other. **false** Users cannot transfer files to each other.
Disallowed_File_Transfer_Types	File extension	Restricts users from transferring specific file types. Set file extensions as the value, for example, `.exe`. Use a semicolon to delimit multiple file extensions, for example, `.exe;.msi;.rar;.zip`.
Customize_Phone_Server	true false	Allows users to change their phone server settings in the client in on-premises deployments. Do not set this parameter to true if you are deploying SAML SSO, as changing phone server settings could interfere with SSO working properly. **true** Users can change their phone server settings. **false (default)** Users cannot change their phone server settings.
Customize_Voicemail_Server	true false	Allows users to change their voicemail server settings in the client in on-premises deployments. Do not set this parameter to true if you are deploying SAML SSO, as changing voicemail server settings could interfere with SSO working properly. **true** Users can change their voicemail server settings. **false (default)** Users cannot change their voicemail server settings.

Related Topics

Common Policies, on page 31

Cisco WebEx Policies, on page 41

Common Policies

The following table describes the parameters you can specify within the Policies element in both on-premises deployments and hybrid cloud-based deployments:

Parameter	Value	Description
EnableVideo	true false	Enables or disables video capabilities. **true (default)** Users can make and receive video calls. **false** Users cannot make or receive video calls.
InitialPhoneSelection	deskphone softphone	Sets the phone type for users when the client starts for the first time. Users can change their phone type after the initial start. The client then saves the user preference and uses it for subsequent starts. **deskphone** Use the desk phone device for calls. **softphone (default)** Use the software phone (CSF) device for calls. The client selects devices in the following order: 1. Software phone devices 2. Desk phone devices If you do not provision users with software phone devices, the client automatically selects desk phone devices.
UserDefinedRemoteDestinations	true false	Lets users add, edit, and delete remote destinations through the client interface. Use this parameter to change the default behavior when you provision Extend and Connect capabilities. By default, if a user's device list contains only a CTI remote device, the client does not let that user add, edit, or delete remote destinations. This occurs to prevent users from modifying dedicated remote devices that you assign. However, if the user's device list contains a software device or a desk phone device, the client lets users add, edit, and delete remote destinations. **true** Users can add, edit, and delete remote destinations. **false (default)** Users cannot add, edit, and delete remote destinations.

Parameter	Value	Description
enableLocalAddressBookSearch	true false	Lets users search for and add local Microsoft Outlook contacts to their contact lists. **true (default)** Users can search for and add local contacts to their contact lists. **false** Users cannot search for or add local contacts to their contact lists.
EnableAccessoriesManager	true false	Enables the accessories API in the client. This API lets accessory vendors create plugins to enable call management functionality for devices such as headsets. **true (default)** Enable the accessories API. **false** Disable the accessories API.

BlockAccessoriesManagerPlugins	Plugin library	Disables specific Accessories Manager plugins from third party vendors such as Jabra or Logitech. You should set the name of the plugin DLL file as the value. Use a comma to separate multiple values, for example, on Microsoft Windows:
		`<BlockAccessoriesManagerPlugins>`
		`JabraJabberPlugin.dll,lucpcisco.dll`
		`</BlockAccessoriesManagerPlugins>`
		There is no default value.

ForceFontSmoothing	true false	Specifies if the client applies anti-aliasing to smooth text. **true (default)** The client applies anti-aliasing to text. **false** The operating system applies anti-aliasing to text.

Parameter	Value	Description
EnableBFCPVideoDesktopShare	true false	Enables BFCP video desktop sharing capabilities. **true (default)** Enables BFCP video desktop sharing on the client. **false** Disables BFCP video desktop sharing. **Note** BFCP video desktop sharing is enabled on the server as follows: • On Cisco Unified Communications Manager version 8.x and lower, you must select the **Allow Presentation Sharing using BFCP** checkbox. • On Cisco Unified Communications Manager version 9.x and higher, BFCP video desktop sharing is enabled by default.
Meetings_Enabled	true false	Enables meetings capabilities in the client. Works in conjunction with the CalendarIntegrationType parameter. **true (default)** Enables meetings capabilities, allowing you to create meetings and get reminders to join meetings. **false** Disables meetings capabilities and user interface.
CalendarIntegrationType	0 1	This parameter works in conjunction with the Meetings_Enabled parameter. **0** Disables calendar integration in the Meetings tab of the client user interface. If you disable this parameter, the Meetings tab in the client is empty, but the Meetings tab remains on the hub window. **1** Enables calendar integration in the Meetings tab of the client user interface.

Parameter	Value	Description
Telephony_Enabled	true false	Enables audio and video capabilities and user interface in the client. **true (default)** Enables audio and video capabilities and user interface. **false** Disables audio and video capabilities and user interface. If you are upgrading to this release, and your client is enabled for IM-only mode, then you must set this parameter to false. If you do not set this parameter in IM-only mode deployments, then users may see disabled telephony capabilities on their user interface.
Voicemail_Enabled	true false	Enables voicemail capabilities and user interface in the client. **true (default)** Enables voicemail capabilities and user interface. **false** Disables voicemail capabilities and user interface.
EnableTelProtocolHandler	true false	Specifies if the client registers as the protocol handler for the `tel:` URI. **true (default)** The client registers as the protocol handler for the `tel:` URI. **false** The client does not register as the protocol handler for the `tel:` URI.
EnableIMProtocolHandler	true false	Specifies if the client registers as the protocol handler for the `IM:` URI or `XMPP:` URI. **true (default)** The client registers as the protocol handler for the `IM:` URI or `XMPP:` URI. **false** The client does not register as the protocol handler for the `IM:` URI or `XMPP:` URI.

Parameter	Value	Description
EnableSIPProtocolHandler	true false	Specifies if the client registers as the protocol handler for the `SIP:` URI. **true (default)** The client registers as the protocol handler for the `SIP:` URI. **false** The client does not register as the protocol handler for the `SIP:` URI.
EnableSaveChatToFile	true false	Allows users to save their chats to the file system as HTML. **true (default)** Users can save their chats to file. **false** Users cannot save their chats to file.
EnableSIPURIDialling	true false	Enables URI dialing with Cisco Jabber and allows users to make calls with URIs. **true** Users can make calls with URIs. **false (default)** Users cannot make calls with URIs.

Parameter	Value	Description
DirectoryURI BDIDirectoryURI	Directory attribute	Specifies the directory attribute that holds the SIP URI for users. **On-Premises Deployments** Set one of the following as the value: • mail • msRTCSIP-PrimaryUserAddress **Cloud-Based Deployments** Set one of the following as the value: • mail • imaddress • workphone • homephone • mobilephone The mail attribute is used by default. **Important** The value you specify must match the directory URI setting for users in Cisco Unified Communications Manager or the Cisco WebEx Administration Tool.
ForceC2XDirectoryResolution	true false	Specifies if the client queries the directory to resolve contact information when users perform click-to-x actions. **true (default)** The client queries the directory when users perform click-to-x actions. **false** The client does not query the directory for click-to-x actions. **Note** This parameter does not take effect when users connect to the corporate network through Expressway for Mobile and Remote Access. In this case, UDS provides contact resolution and the client cannot query the directory.

Parameter	Value	Description

ServiceDiscoveryExcludedServices	WEBEX
	CUCM
	CUP
		Specifies whether to exclude certain services from Service Discovery.
		WEBEX
		When you set this value, the client:
		• Does not perform CAS lookup
		• Looks for `_cisco-uds`, `_cuplogin`, and `_collab-edge`
		CUCM
		When you set this value, the client:
		• Does not look for `_cisco_uds`
		• Looks for `_cuplogin` and `_collab-edge`
		CUP
		When you set this value, the client:
		• Does not look for `_cuplogin`
		• Looks for `_cisco-uds_collab-edge`
		You can specify multiple, comma-separated values to exclude multiple services. For example:
		`<ServiceDiscoveryExcludedServices>`
		`WEBEX,CUCM`
		`</ServiceDiscoveryExcludedServices>`

VoiceServicesDomain	FQDN	Specifies the Fully Qualified Domain Name that represents the DNS domain where the DNS SRV records for *_collab-edge* and *_cisco-uds* are configured. **Example:** Given the following DNS SRV records: • *_collab-edge.*_tls.voice.example.com • *_cisco-uds.*_tcp.voice.example.com The *VoiceServicesDomain* value would be *voice.example.com*.

Parameter	Value	Description
ctiwindowbehaviour	OnVideo OnCall Never	Specifies the behavior of the conversation window when the user has answered a call in deskphone control mode (CTI mode). **OnVideo** Conversation window is only displayed for video calls. This option is not supported on Cisco Jabber for Mac. **OnCall (default)** Conversation window is always displayed when a call is answered. **Never** Conversation window is never displayed when a call is answered.
EnableCallPickup	true false	Specifies if a user can pickup a call in their call pickup group. **true** Enables call pickup. **false (default)** Disables call pickup.
EnableGroupCallPickup	true false	Specifies if a user can pickup incoming calls in another call pickup group, by entering the call pickup group number. **true** Enables group call pickup. **false (default)** Disables group call pickup.
EnableOtherGroupPickup	true false	Specifies if a user can pickup an incoming call in a group that is associated with their own call pickup group. **true** Enables other group call pickup **false (default)** Disables other group call pickup
EnableHuntGroup	true false	Specifies if a user can log into a hunt group. **true** Users can log into their hunt group. **false (default)** Users cannot log into their hunt group.

Parameter	Value	Description
PreventDeclineOnHuntCall	true false	Specifies if the Decline button is displayed for an incoming call in a hunt group. **true** Decline button is not displayed for an incoming call in a hunt group. **false (default)** Decline button is displayed for an incoming call in a hunt group.
TelemetryEnabled	true false	Specifies whether analytics data will be gathered. **true (default)** Analytics data will be gathered. **false** Analytics data will not be gathered.
TelemetryCustomerID	String	Specifies the source of analytic information. This can be a string that explicitly identifies an individual customer or a string that identifies a common source without identifying the customer. Cisco recommends using a Global Unique Identifier (GUID) generating utility to generate a 36 character unique identifier or to use a reverse domain name. The following utilities are available for generating a GUID: • Mac OS X - uuidgen • Linux - uuidgen • Microsoft Windows - [guid]::NewGuid().ToString() or (from cmd.exe) powershell -command "[guid]::NewGuid().ToString()" • Online - guid.us This identifier should be globally unique regardless of the method used to create the GUID.

Parameter	Value	Description
TelemetryEnabledOverCellularData	true false	Specifies whether analytics data will be sent over Wi-Fi only. **true (default)** Analytics data will be sent over Wi-Fi and mobile data connections. **false** Analytics data will be sent over Wi-Fi connections only. This parameter is optional.
EnableTelProtocolPopupWindow CiscoTelProtocolPermissionEnabled	true false	Specifies whether the pop-up window is enabled or disabled which asks users to confirm if they want to make a call after they click on a ciscotel:uri enabled number. **true (default)** Pop-up window is enabled and users are asked to confirm that they want to place the call. **false** Pop-up window is disabled and the call is made without requesting confirmation first. This may cause accidental or unwanted calls. **Note** The CiscoTelProtocolPermissionEnabled parameter replaces the EnableTelProtocolPopupWindow parameter. Both parameters are supported in the client, however the pop-up window is disabled if either parameter is set to false.
ServicesDomainSsoEmailPrompt	ON OFF	Specifies whether the user is shown the email prompt for the purposes of determining their home cluster. **ON** The prompt is shown. **OFF (default)** The prompt is not shown.
EnableP2PDesktopShare	true false	Allows users to share their screen if not on a call. **true (Default)** Allows users to share their screens. **false** Users cannot do peer to peer screen sharing.

Parameter	Value	Description
EnableForensicsContactData	true false	Specifies whether users' Contacts folder is collected by the Problem Reporting Tool (PRT) when reporting a problem that is related to their contacts. **true (default)** Contacts folder is collected by the PRT tool. **false** Contacts folder is not collected by the PRT tool.

Related Topics

On-Premises Policies, on page 30
Cisco WebEx Policies, on page 41

Cisco WebEx Policies

If you use the Cisco WebEx Messenger service for instant messaging and presence capabilities, you can set policies for the client through the Cisco WebEx Administration Tool. See *Using policy actions available in Cisco WebEx* for a list of available policies and descriptions.

Related Topics

On-Premises Policies, on page 30
Common Policies, on page 31
Using policy actions available in Cisco WebEx

Presence Parameters

The following table describes the parameters you can specify within the Presence element:

Parameter	Value	Description
LoginResource	multiResource wbxconnect	Controls user log in to multiple client instances. **multiResource (default)** Users can log in to multiple instances of the client at the same time. **wbxconnect** Users can log in to one instance of the client at a time. The client appends the `wbxconnect` suffix to the user's JID. Users cannot log in to any other Cisco Jabber client that uses the `wbxconnect` suffix.

Parameter	Value	Description
PresenceServerAddress	Hostname IP address FQDN	Specifies the address of a presence server for on-premises deployments. Set one of the following as the value: • Hostname (*hostname*) • IP address (*123.45.254.1*) • FQDN (*hostname.domain.com*)
PresenceServerURL	CAS URL	Specifies the Central Authentication Service (CAS) URL for the Cisco WebEx Messenger service. The following is an example of a URL you can set as the value: `https://loginp.webexconnect.com/cas/sso/ex_org/orgadmin.app`
CalendarWebExMeetingPresence	true false	Enables users' presence to change to "In a WebEx meeting" even if they do not join the WebEx session link but the meeting is in their Microsoft Outlook calendar. **true** Users' presence changes to "In a WebEx meeting" even if they do not join the WebEx session link. **false (default)** Users must join the WebEx session link for their presence to change to "In a WebEx meeting". Otherwise, their presence remains "Available", even if the meeting is in their Microsoft Outlook calendar.

Service Credentials Parameters

You can specify service credentials parameters so that users do not need to authenticate with certain services.

Voicemail Service Credentials

You can specify the following parameter to configure voicemail service credentials within the Voicemail element:

Parameter	Value	Description
VoiceMailService_UseCredentialsFrom	phone	Specifies that the client uses the phone service credentials to access voicemail services. Ensure the user's phone service credentials match their voicemail service credentials. If you set this configuration, users cannot specify voicemail service credentials in the client interface. This parameter is not set by default. You should set this parameter in the following deployments only: • Hybrid cloud-based deployments. • Phone mode deployments. In on-premises deployments, you should set the credentials source for voicemail services on the presence server.

The following is an example of the voicemail service credentials parameter:

```
<?xml version="1.0" encoding="utf-8"?>
<config version="1.0">
  <Voicemail>
    <VoicemailService_UseCredentialsFrom>phone</VoicemailService_UseCredentialsFrom>
  </Voicemail>
</config>
```

Voicemail Parameters

The following table describe the voicemail service configuration parameters you can specify within the Voicemail element:

Key	Value	Description
VoicemailPrimaryServer	Hostname IP address FQDN	Specifies the address of your voicemail server. Set one of the following as the value: • Hostname (*hostname*) • IP address (*123.45.254.1*) • FQDN (*hostname.domain.com*)

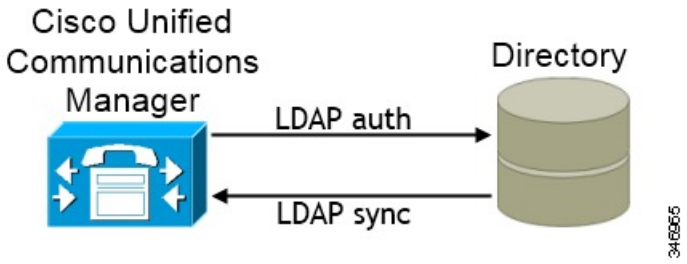
Related Topics

Service Credentials Parameters, on page 42

Set Up Directory Synchronization and Authentication

When you set up an on-premises deployment, you should configure Cisco Unified Communications Manager to do both of the following:

- Synchronize with the directory server.
- Authenticate with the directory server.



Synchronizing with the directory server replicates contact data from your directory to Cisco Unified Communications Manager.

Enabling authentication with the directory server lets Cisco Unified Communications Manager proxy authentication from the client to the directory server. In this way, users authenticate with the directory server, not with Cisco Unified Communications Manager or a presence server.

Related Topics

Configuring Cisco Unified Communications Manager Directory Integration

Synchronize with the Directory Server

Directory server synchronization ensures that contact data in your directory server is replicated to Cisco Unified Communications Manager.

Enable Synchronization

To ensure that contact data in your directory server is replicated to Cisco Unified Communications Manager, you must synchronize with the directory server. Before you can synchronize with the directory server, you must enable synchronization.

Procedure

Step 1 Open the **Cisco Unified CM Administration** interface.

Step 2 Select **System** > **LDAP** > **LDAP System**.

The **LDAP System Configuration** window opens.

Step 3 Locate the **LDAP System Information** section.

Step 4 Select **Enable Synchronizing from LDAP Server**.

Step 5 Select the type of directory server from which you are synchronizing data from the **LDAP Server Type** drop-down list.

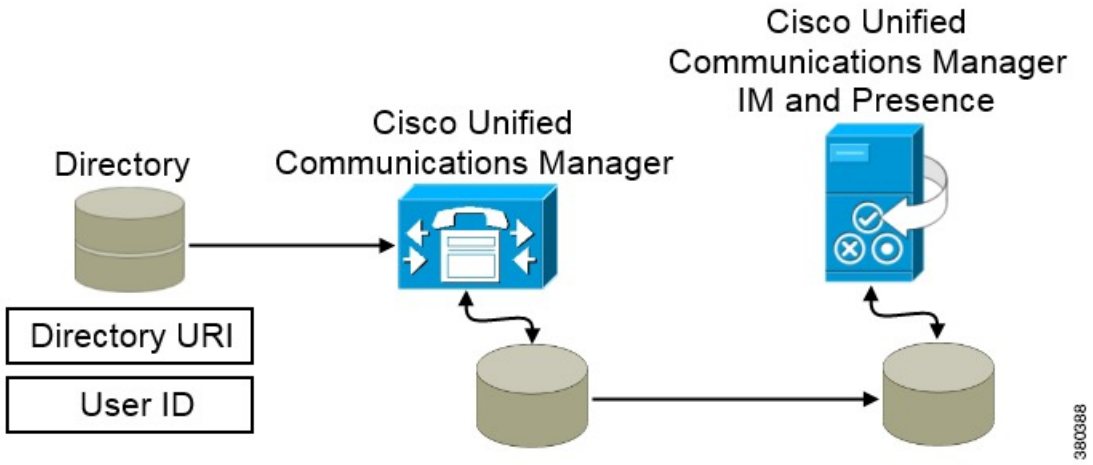
What to do next

Specify an LDAP attribute for the user ID.

Populate User ID and Directory URI

When you synchronize your LDAP directory server with Cisco Unified Communications Manager, you can populate the end user configuration tables in both the Cisco Unified Communications Manager and the Cisco Unified Communications Manager IM and Presence Service databases with attributes that contain values for the following:

- User ID — You must specify a value for the user ID on Cisco Unified Communications Manager. This value is required for the default IM address scheme and for users to sign in. The default value is `sAMAccountName`.
- Directory URI — You should specify a value for the directory URI if you plan to:
  - Enable URI dialing in Cisco Jabber.



When Cisco Unified Communications Manager synchronizes with the directory source, it retrieves the values for the directory URI and user ID and populates them in the end user configuration table in the Cisco Unified Communications Manager database.

The Cisco Unified Communications Manager database then synchronizes with the Cisco Unified Communications Manager IM and Presence Service database. As a result, the values for the directory URI and user ID are populated in the end user configuration table in the Cisco Unified Communications Manager IM and Presence Service database.

Specify an LDAP Attribute for the User ID

When you synchronize from your directory source to Cisco Unified Communications Manager, you can populate the user ID from an attribute in the directory. The default attribute that holds the user ID is `sAMAccountName`.

Procedure

Step 1 Locate the **LDAP Attribute for User ID** drop-down list on the **LDAP System Configuration** window.

Step 2 Specify an attribute for the user ID as appropriate and then select **Save**.

Important If the attribute for the user ID is other than `sAMAccountName` and you are using the default IM address scheme in Cisco Unified Communications Manager IM and Presence Service, you must specify the attribute as the value for the parameter in your client configuration file as follows:

The EDI parameter is `UserAccountName`.

```
<UserAccountName>attribute-name</UserAccountName>
```

The BDI parameter is `BDIUserAccountName`.

```
<BDIUserAccountName>attribute-name</BDIUserAccountName>
```

If you do not specify the attribute in your configuration, and the attribute is other than `sAMAccountName`, the client cannot resolve contacts in your directory. As a result, users do not get presence and cannot send or receive instant messages.

Related Topics

Specify an LDAP Attribute for the Directory URI, on page 46

Specify an LDAP Attribute for the Directory URI

On Cisco Unified Communications Manager release 9.0(1) and later, you can populate the directory URI from an attribute in the directory.

Before you begin

Enable Synchronization.

Procedure

Step 1 Select **System** > **LDAP** > **LDAP Directory**.

Step 2 Select the appropriate LDAP directory or select **Add New** to add an LDAP directory.

Step 3 Locate the **Standard User Fields To Be Synchronized** section.

Step 4 Select one of the following LDAP attributes from the **Directory URI** drop-down list:

- **msRTCSIP-primaryuseraddress**—This attribute is populated in the AD when Microsoft Lync or Microsoft OCS are used. This is the default attribute.
- **mail**

Step 5 Select **Save**.

Related Topics

Specify an LDAP Attribute for the User ID, on page 45

Perform Synchronization

After you add a directory server and specify the required parameters, you can synchronize Cisco Unified Communications Manager with the directory server.

Before you begin

If your environment includes a presence server, you should ensure the following feature service is activated and started before you synchronize with the directory server:

- Cisco Unified Presence — **Cisco UP Sync Agent**
- Cisco Unified Communications Manager IM and Presence Service — **Cisco Sync Agent**

This service keeps data synchronized between the presence server and Cisco Unified Communications Manager. When you perform the synchronization with your directory server, Cisco Unified Communications Manager then synchronizes the data with the presence server. However, the **Cisco Sync Agent** service must be activated and started.

Procedure

Step 1 Select **System** > **LDAP** > **LDAP Directory**.

Step 2 Select **Add New**.

The **LDAP Directory** window opens.

Step 3 Specify the required details on the **LDAP Directory** window.

See the Cisco Unified Communications Manager Administration Guide for more information about the values and formats you can specify.

Step 4 Create an LDAP Directory Synchronization Schedule to ensure that your information is synchronized regularly.

Step 5 Select **Save**.

Step 6 Select **Perform Full Sync Now**.

Note The amount of time it takes for the synchronization process to complete depends on the number of users that exist in your directory. If you synchronize a large directory with thousands of users, you should expect the process to take some time.

User data from your directory server is synchronized to the Cisco Unified Communications Manager database. Cisco Unified Communications Manager then synchronizes the user data to the presence server database.

Authenticate with the LDAP Server

Perform this procedure if you want to enable LDAP authentication so that end user passwords are authenticated against the password that is assigned in the company LDAP directory. LDAP authentication gives system administrators the ability to assign an end user a single password for all company applications. This configuration applies to end user passwords only and does not apply to end user PINs or application user passwords. When users sign in to the client, the presence service routes that authentication to Cisco Unified

Communications Manager. Cisco Unified Communications Manager then sends that authentication to the directory server.

Procedure

Step 1 Open the **Cisco Unified CM Administration** interface.

Step 2 Select **System** > **LDAP** > **LDAP Authentication**.

Step 3 Select **Use LDAP Authentication for End Users**.

Step 4 Specify LDAP credentials and a user search base as appropriate.

See the *Cisco Unified Communications Manager Administration Guide* for information about the fields on the **LDAP Authentication** window.

Step 5 Select **Save**.

Federation

Federation lets Cisco Jabber users communicate with users who are provisioned on different systems and who are using client applications other than Cisco Jabber.

Interdomain Federation

Interdomain federation enables Cisco Jabber users in an enterprise domain to share availability and send instant messages with users in another domain.

- Cisco Jabber users must manually enter contacts from another domain.
- Cisco Jabber supports federation with the following:
  - Microsoft Office Communications Server
  - Microsoft Lync
  - IBM Sametime
  - XMPP standard-based environments such as Google Talk

Note Expressway for Mobile and Remote Access doesn't enable XMPP Interdomain federation itself. Cisco Jabber clients connecting over Expressway for Mobile and Remote Access can use XMPP Interdomain federation if it has been enabled on Cisco Unified Communications Manager IM and Presence.

  - AOL Instant Messenger

You configure interdomain federation for Cisco Jabber on Cisco Unified Communications Manager IM and Presence Service. See the appropriate server documentation for more information.

Related Topics

Integration Guide for Configuring Cisco Unified Presence Release 8.6 for Interdomain Federation
Interdomain Federation for IM and Presence Service on Cisco Unified Communications Manager

Intradomain Federation

Intradomain federation enables users within the same domain to share availability and send instant messages between Cisco Unified Communications Manager IM and Presence Service and Microsoft Office Communications Server, Microsoft Live Communications Server, or another presence server.

Intradomain federation allows you to migrate users to Cisco Unified Communications Manager IM and Presence Service from a different presence server. For this reason, you configure intradomain federation for Cisco Jabber on the presence server. See the following for more information:

- Cisco Unified Presence: *Integration Guide for Configuring Partitioned Intradomain Federation for Cisco Unified Presence Release 8.6 and Microsoft LCS/OCS*
- Cisco Unified Communications Manager IM and Presence Service: *Partitioned Intradomain Federation for IM and Presence Service on Cisco Unified Communications Manager*

Configure Intradomain Federation for BDI or EDI

In addition to configuring intradomain federation on the presence server, you might need to specify some configuration settings in the Cisco Jabber configuration files.

To resolve contacts during contact search or retrieve contact information from your directory, Cisco Jabber requires the contact ID for each user. Cisco Unified Communications Manager IM & Presence server uses a specific format for resolving contact information that does not always match the format on other presence servers such as Microsoft Office Communications Server or Microsoft Live Communications Server.

The parameters that you use to configure intradomain federation depend on whether you use *Enhanced Directory Integration* (EDI) or *Basic Directory Integration* (BDI). EDI uses native Microsoft Windows APIs to retrieve contact data from the directory service and is only used by Cisco Jabber for Windows. For BDI, the client retrieves contact data from the directory service and is used by Cisco Jabber for Mac, Cisco Jabber for Android, and Cisco Jabber for iPhone and iPad.

Procedure

Step 1 Set the value of the relevant parameter to true:

- For BDI: BDIUseSipUriToResolveContacts
- For EDI: UseSIPURIToResolveContacts

Step 2 Specify an attribute that contains the Cisco Jabber contact ID that the client uses to retrieve contact information. The default value is `msRTCSIP-PrimaryUserAddress`, or you can specify another attribute in the relevant parameter:

- For BDI: BDISipUri
- For EDI: SipUri

Note When you deploy intradomain federation and the client connects with Expressway for Mobile and Remote Access from outside the firewall, contact search is supported only when the contact ID uses one of the following formats:

- sAMAccountName@domain
- UserPrincipleName (UPN)@domain
- EmailAddress@domain
- employeeNumber@domain
- phoneNumber@domain

Step 3 In the UriPrefix parameter, specify any prefix text that precedes each contact ID in the relevant SipUri parameter.

Example:

For example, you specify `msRTCSIP-PrimaryUserAddress` as the value of SipUri. In your directory the value of `msRTCSIP-PrimaryUserAddress` for each user has the following format: `sip:`*`username@domain`*.

- For BDI: BDIUriPrefix
- For EDI: UriPrefix

Example

The following XML snippet provides an example of the resulting configuration for BDI:

```
<Directory>
  <BDIUseSIPURIToResolveContacts>true</BDIUseSIPURIToResolveContacts>
  <BDISipUri>non-default-attribute</BDISipUri>
  <BDIUriPrefix>sip:</BDIUriPrefix>
</Directory>
```

The following XML snippet provides an example of the resulting configuration for EDI:

```
<Directory>
  <UseSIPURIToResolveContacts>true</UseSIPURIToResolveContacts>
  <SipUri>non-default-attribute</SipUri>
  <UriPrefix>sip:</UriPrefix>
</Directory>
```

Related Topics

Example of Intradomain Federation, on page 50

Example of Intradomain Federation

The following example shows how to create intradomain federation contacts using the following BDI or EDI parameters and example values:

For BDI: BDISipUri
For EDI: SipURI

Value: msRTCSIP-PrimaryUserAddress

For BDI: BDIUseSIPURIToResolveContacts
For EDI: UseSIPURIToResolveContacts

Value: true

For BDI: BDIUriPrefix
For EDI: UriPrefix

Value: sip

For the user Mary Smith, the directory contains **`sip:msmith@domain.com`** as the value of the msRTCSIP-PrimaryUserAddress attribute.

The following workflow describes how the client connects to your directory to resolve contact information for Mary Smith:

1. Your presence server passes `msmith@domain.com` to the client.
2. The client adds sip: to `msmith@domain.com` and then queries your directory.
3. `sip:msmith@domain.com` matches the value of the `msRTCSIP-PrimaryUserAddress` attribute.
4. The client retrieves contact information for Mary Smith.

When Cisco Jabber users search for Mary Smith, the client removes the sip: prefix from `sip:msmith@domain.com` to get her contact ID.

Related Topics

Configure Intradomain Federation for BDI or EDI, on page 49

Administer and Moderate Persistent Chat Rooms

Note

- Persistent Chat Rooms and their administration is for on-premises deployments only.
- Persistent Chat Rooms are not available for mobile clients.

You administer persistent chat rooms from the Jabber client by creating rooms, delegating their moderators, and specifying members. The node on which the room is created is created automatically, although you can override it and specify a specific node. Administrators and moderators are privileged users in Persistent Chat rooms. You can administer Persistent Chat rooms on any service node that you are an administrator for on Cisco Unified Communications Manager IM and Presence servers.

Administrator Capabilities

Administrators can perform the following tasks from the **All Rooms** tab of Persistent Chat in the client hub window:

- Create rooms. When you create a room, you automatically become the room administrator.

- Define and change up to 30 moderators for a chat room (who become *room owners*).
- Specify and change the room name.
- Define the maximum number of participants in a room. This number cannot be less than the number of participants already in a room.
- Add and remove room members.
- Block, remove, and revoke participants.
- Destroy rooms (which removes it from the server, but the history is not deleted).

Moderator Capabilities

Up to 30 moderators can be defined by an administrator for one Persistent Chat room. Moderators can perform the following tasks:

- Change the subject of a room.
- Edit members (which includes adding, removing, and banning them).

Room Creation

When creating a room, you can provide the following types of information:

- Room name (required, maximum 200 characters)
- Description
- Room type (public or restricted)

  After the room type has been defined, it cannot be changed by anyone.
- Specify whether to add the room to your **My Rooms** tab (off by default)
- Add up to 30 moderators (who must have a valid Jabber ID to moderate a room).
- Room password

After you create the room, you have the option to add members to the room immediately or at a later time. Refresh the **All Rooms** list in order to see your new room in the list of available rooms.

Problem Reporting

Applies to: Cisco Jabber for Windows

Setting up problem reporting enables users to send a summary of issues that they encounter with the client. There are two methods for submitting problem reports as follows:

- Users submit the problem report directly through the client interface.
- Users save the problem report locally and then upload it at a later time.

The client uses an HTTP POST method to submit problem reports. Create a custom script to accept the POST request and specify the URL of the script on your HTTP server as a configuration parameter. Because users

can save problem reports locally, you should also create an HTML page with a form to enable users to upload problem reports.

Before you begin

Complete the following steps to prepare your environment:

1. Install and configure an HTTP server.
2. Create a custom script to accept the HTTP POST request.
3. Create an HTML page that enables users to upload problem reports that are saved locally. Your HTML page should contain a form that accepts the problem report saved as a `.ZIP` archive and contains an action to post the problem report using your custom script.

The following is an example form that accepts problem reports:

```
<form name="uploadPrt" action="http://server_name.com/scripts/UploadPrt.php" method="post"
 enctype="multipart/form-data">
 <input type="file" name="zipFileName" id="zipFileName" /><br />
 <input type="submit" name="submitBtn" id="submitBtn" value="Upload File" />
</form>
```

Procedure

Step 1 Host your custom script on your HTTP server.

Step 2 Specify the URL of your script as the value of the PrtLogServerUrl parameter in your configuration file.

Configure Automatic Updates

Applies to: Cisco Jabber for Windows, Cisco Jabber for Mac

To enable automatic updates, you create an XML file that contains the information for the most recent version, including the URL of the installation package on the HTTP server. The client retrieves the XML file when users sign in, resume their computer from sleep mode, or perform a manual update request from the **Help** menu.

Note If you use the Cisco WebEx Messenger service for instant messaging and presence capabilities, you should use the Cisco WebEx Administration Tool to configure automatic updates.

XML File Structure

XML files for automatic updates have the following structure:

```
<JabberUpdate>
          <App name="JabberWin">
               <LatestBuildNum>12345</LatestBuildNum>
               <LatestVersion>10.5.x</LatestVersion>
               <Mandatory>true</Mandatory>
               <Message>
                  <![CDATA[<b>This new version of Cisco Jabber lets you do the
                  following:</b><ul><li>Feature 1</li><li>Feature 2</li></ul>For
                  more information click <a target="_blank"
```

```
                href="http://cisco.com/go/jabber">here</a>.]]>
              </Message>
              <DownloadURL>http://http_server_name/CiscoJabberSetup.msi</DownloadURL>

        </App>
      </JabberUpdate>
```

Example XML File 1

The following is example XML file for automatic updates:

```
<JabberUpdate>
<App name="JabberWin">
  <LatestBuildNum>12345</LatestBuildNum>
  <LatestVersion>9.x</LatestVersion>
  <Message><![CDATA[<b>This new version of Cisco Jabber lets you do the
following:</b><ul><li>Feature 1</li><li>Feature 2</li></ul>For
  more information click <a target="_blank"
href="http://cisco.com/go/jabber">here</a>.]]></Message>
  <DownloadURL>http://http_server_name/CiscoJabberSetup.msi</DownloadURL>
</App>
</JabberUpdate>
```

Example XML File 2

The following is an example XML file for automatic updates for both Cisco Jabber for Windows and Cisco Jabber for Mac:

```
<JabberUpdate>
 <App name="JabberMac">
  <LatestBuildNum>12345</LatestBuildNum>
  <LatestVersion>9.6.1</LatestVersion>
  <Message><![CDATA[<b>This new version of Cisco Jabber lets you do the
following:</b><ul><li>Feature 1</li><li>Feature 2</li>
  </ul>For more information click <a target="_blank"
href="http://cisco.com/go/jabber">here</a>.]]>
  </Message>

<DownloadURL>http://http_server_name/Cisco-Jabber-Mac-9.6.1-12345-MrbCdd.zip</DownloadURL>

 </App>
 <App name="JabberWin">
  <LatestBuildNum>12345</LatestBuildNum>
  <LatestVersion>9.0</LatestVersion>
  <Message><![CDATA[<b>This new version of Cisco Jabber lets you do the
following:</b><ul><li>Feature 1</li><li>Feature 2
  </li></ul>For more information click <a target="_blank"
href="http://cisco.com/go/jabber">here</a>.]]>
  </Message>
  <DownloadURL>http://http_server_name/CiscoJabberSetup.msi
  </DownloadURL>
</App>
</JabberUpdate>
```

Before you begin

- Install and configure an HTTP server to host the XML file and installation package.
- Ensure users have permission to install software updates on their workstations.

  Microsoft Windows stops update installations if users do not have administrative rights on their workstations. You must be logged in with administrative rights to complete installation.

Procedure

Step 1 Host the update installation program on your HTTP server.

Step 2 Create an update XML file with any text editor.

Step 3 Specify values in the XML as follows:

- `name`—Specify the following ID as the value of the `name` attribute for the `App` element:
  - JabberWin—The update applies to Cisco Jabber for Windows.
  - JabberMac—The update applies to Cisco Jabber for Mac.
- `LatestBuildNum`—Build number of the update.
- `LatestVersion`—Version number of the update.
- `Mandatory`—(Windows clients only) True or False. Determines whether users must upgrade their client version when prompted.
- `Message`—HTML in the following format:

  ```
  <![CDATA[your_html]]>
  ```

- `DownloadURL`—URL of the installation package on your HTTP server.

  For Cisco Jabber for Mac the URL file must be in the following format:

  ```
  Cisco-Jabber-Mac-version-size-dsaSignature.zip
  ```

Step 4 Save and close your update XML file.

Step 5 Host your update XML file on your HTTP server.

Step 6 Specify the URL of your update XML file as the value of the UpdateUrl parameter in your configuration file.

Custom Embedded Tabs

Custom embedded tabs display HTML content in the client interface. Learn how to create custom embedded tab definitions for Cisco Jabber.

Note The Jabber embedded browser does not support cookie sharing with pop-ups from SSO enabled webpages. The content on the pop-up window may fail to load.

Custom Embedded Tab Definitions

The custom embedded tab can only be configured using the `jabber-config.xml` file. The following XML snippet shows the structure for custom tab definitions:

```
<jabber-plugin-config>
 <browser-plugin>
  <page refresh="" preload="">
```

```
   <tooltip></tooltip>
   <icon></icon>
   <url></url>
  </page>
 </browser-plugin>
</jabber-plugin-config>
```

Cisco Jabber for Windows supports Internet Explorer version 9 or earlier. The client uses Internet Explorer in version 9 mode if a later version is on the workstation.

The following table describes the parameters for custom embedded tab definitions:

Parameter	Description
browser-plugin	Contains all definitions for custom embedded tabs. The value includes all custom tab definitions.
page	Contains one custom embedded tab definition.
refresh	Controls when the content refreshes. • true — Content refreshes each time users select the tab. • false (default) — Content refreshes when users restart the client or sign in. This parameter is optional and is an attribute of the page element.
preload	Controls when the content loads. • true — Content loads when the client starts. • false (default) — Content loads when users select the tab. This parameter is optional and is an attribute of the page element.
tooltip	Defines hover text for the custom embedded tab. This parameter is optional. If you do not specify the hover text, the client will use *Custom tab*. The value is string of unicode characters.
icon	Specifies an icon for the tab. You can specify a local or hosted icon as follows: • Local icon—Specify the URL as follows: `file://file_path/icon_name` • Hosted icon—Specify the URL as follows: `http://path/icon_name` You can use any icon that the client browser can render, including .JPG, .PNG, and .GIF formats. This parameter is optional. If you do not specify an icon, the client loads the favicon from the HTML page. If no favicon is available, the client loads the default icon.

Parameter	Description
url	Specifies the URL where the content for the embedded tab resides. The client uses the browser rendering engine to display the content of the embedded tab. For this reason, you can specify any content that the browser supports. This parameter is required.

User Custom Tabs

Users can create their own custom embedded tabs through the client user interface.

You must enable users to create custom embedded tabs. Set true as the value for the AllowUserCustomTabs parameter in your configuration file as follows:

```
<Options>
  <AllowUserCustomTabs>true</AllowUserCustomTabs>
</Options>
```

Note User custom embedded tabs are set to true by default.

Custom Icons

To achieve optimal results, your custom icon should conform to the following guidelines:

- Dimensions: 20 x 20 pixels
- Transparent background
- PNG file format

Chats and Calls from Custom Tabs

You can use protocol handlers to start chats and calls from custom embedded tabs. Make sure the custom embedded tab is an HTML page.

Use the `XMPP:` or `IM:` protocol handler to start chats.

Use the `TEL:` protocol handler to start audio and video calls.

Related Topics

Protocol Handlers

UserID Tokens

You can specify the `${UserID}` token as part of the value for the url parameter. When users sign in, the client replaces the `${UserID}` token with the username of the logged in user.

Tip You can also specify the `${UserID}` token in query strings; for example, `www.cisco.com/mywebapp.op?url=${UserID}`.

The following is an example of how you can use the `${UserID}` token:

1. You specify the following in your custom embedded tab:

   ```
   <url>www.cisco.com/${UserID}/profile</url>
   ```

2. Mary Smith signs in. Her username is msmith.
3. The client replaces the `${UserID}` token with Mary's username as follows:

   ```
   <url>www.cisco.com/msmith/profile</url>
   ```

JavaScript Notifications

You can implement JavaScript notifications in custom embedded tabs. This topic describes the methods the client provides for JavaScript notifications. This topic also gives you an example JavaScript form that you can use to test notifications. It is beyond the scope of this documentation to describe how to implement JavaScript notifications for asynchronous server calls and other custom implementations. You should refer to the appropriate JavaScript documentation for more information.

Notification Methods

The client includes an interface that exposes the following methods for JavaScript notifications:

- SetNotificationBadge — You call this method from the client in your JavaScript. This method takes a string value that can have any of the following values:
  - Empty — An empty value removes any existing notification badge.
  - A number from 1 to 999
  - Two digit alphanumeric combinations, for example, A1
- onPageSelected() — The client invokes this method when users select the custom embedded tab.
- onPageDeselected() — The client invokes this method when users select another tab.

Note Not applicable for Jabber for iPhone and iPad

- onHubResized() — The client invokes this method when users resize or move the client hub window.
- onHubActivated() — The client invokes this method when the client hub windows is activated.
- onHubDeActivated() — The client invokes this method when the client hub window is deactivated.

Subscribe to Presence in Custom Tabs

You can use the following JavaScript functions to subscribe to the presence of a contact and receive presence updates from the client:

- SubscribePresence() — Specify a string value using the IM address of a user for this method.
- OnPresenceStateChanged — This method enables users to receive updates from the client on the presence of a contact. You can specify one of the following values as the string:
  - IM address
  - Basic presence (Available, Away, Offline, Do Not Disturb)
  - Rich presence (In a meeting, On a call, or a custom presence state)

Note

- If you subscribe to the presence of a person who is not on your contact list (also called *temporary presence subscription*), the subscription expires after 68 minutes. After the subscription expires, you must re-subscribe to the person's presence in order to continue to receive presence updates.
- Jabber for iPad and iPhone only supports OnPresenceStateChanged.

Get Locale Information in Custom Tabs

You can use the following JavaScript functions to retrieve the current locale information of a contact from the client:

- GetUserLocale() — This method enables users to request locale information from the client.
- OnLocaleInfoAvailable — This method enables users to receive locale information from client. You can use a string value that contains the client locale information.

Note Jabber for iPad and iPhone only supports OnLocaleInfoAvailable.

Example JavaScript

The following code is an example of an HTML page that uses JavaScript to display a form into which you can input a number from 1 to 999:

```
<html>
        <head>
                <script type="text/javascript">
                                function OnPresenceStateChanged(jid, basicPresence,
localizedPresence)
                                {
                                        var cell = document.getElementById(jid);
                                        cell.innerText = basicPresence.concat(",
",localizedPresence);
                                }

                                function GetUserLocale()
                                {
```

```
                                        window.external.GetUserLocale();
                                }

                                function SubscribePresence()
                                {

window.external.SubscribePresence('johndoe@example.com');
                                }

                                function OnLocaleInfoAvailable(currentLocale)
                                {
                                      var cell = document.getElementById("JabberLocale");

                                        cell.innerText = currentLocale;
                                }

                                function onHubActivated()
                                {
                                        var cell = document.getElementById("hubActive");
                                        cell.innerText = "TRUE";
                                }

                                function onHubDeActivated()
                                {
                                        var cell = document.getElementById("hubActive");
                                        cell.innerText = "FALSE";
                                }

                                function onHubResized()
                                {
                                        alert("Hub Resized or Moved");
                                }

                                function OnLoadMethods()
                                {
                                        SubscribePresence();
                                        GetUserLocale();
                                }
                        </script>
                </head>

                <body onload="OnLoadMethods()">
                        <table>
                                <tr>
                                        <td>John Doe</td>
                                        <td id="johndoe@example.com">unknown</td>
                                </tr>
                        </table>
                        <table>
                                <tr>
                                        <td>Jabber Locale: </td>
                                        <td id="JabberLocale">Null</td>
                                </tr>
                                <tr>
                                        <td>Hub Activated: </td>
                                        <td id="hubActive">---</td>
                                </tr>
                    </table>
                </body>
```

```
</html>
```

To test this example JavaScript form, copy the preceding example into an HTML page and then specify that page as a custom embedded tab.

Show Call Events in Custom Tabs

You can use the following JavaScript function to show call events in a custom tab:

OnTelephonyConversationStateChanged — An API in the telephony service enables the client to show call events in a custom embedded tab. Custom tabs can implement the `OnTelephonyConversationStateChanged` JavaScript function. The client calls this function every time a telephony conversation state changes. The function accepts a JSON string that the client parses to get call events.

The following snippet shows the JSON that holds the call events:

```
{
      "conversationId": string,
      "acceptanceState": "Pending" | "Accepted| | "Rejected",
      "state": "Started" | "Ending" | "Ended",
      "callType": "Missed" | "Placed" | "Received" | "Passive" | "Unknown",
      "remoteParticipants": [{participant1}, {participant2}, …, {participantN}],
      "localParticipant": {
      }
}
```

Each participant object in the JSON can have the following properties:

```
{
      "voiceMediaDisplayName": "<displayName>",
      "voiceMediaNumber": "<phoneNumber>",
      "translatedNumber": "<phoneNumber>",
      "voiceMediaPhoneType": "Business" | "Home" | "Mobile" | "Other" | "Unknown",
      "voiceMediaState": "Active" | "Inactive" | "Pending" | "Passive" | "Unknown",
}
```

The following is an example implementation of this function in a custom embedded tab. This example gets the values for the `state` and `acceptanceState` properties and shows them in the custom tab.

```
function OnTelephonyConversationStateChanged(json) {
      console.log("OnTelephonyConversationStateChanged");
      try {
        var conversation = JSON.parse(json);
        console.log("conversation id=" + conversation.conversationId);
        console.log("conversation state=" + conversation.state);
        console.log("conversation acceptanceState=" + conversation.acceptanceState);
        console.log("conversation callType=" + conversation.callType);
      }
      catch(e) {
        console.log("cannot parse conversation:" + e.message);
      }
    }
```

The following is an example implementation of this function with all possible fields:

```
function OnTelephonyConversationStateChanged(json) {
      console.log("OnTelephonyConversationStateChanged");
      try {
        var conversation = JSON.parse(json);
        console.log("conversation state=" + conversation.state);
```

```
        console.log("conversation acceptanceState=" + conversation.acceptanceState);
        console.log("conversation callType=" + conversation.callType);
        for (var i=0; i<conversation.remoteParticipants.length; i++) {
          console.log("conversation remoteParticipants[" + i + "]=");
          console.log("voiceMediaDisplayName=" +
conversation.remoteParticipants[i].voiceMediaDisplayName);
          console.log("voiceMediaNumber=" +
conversation.remoteParticipants[i].voiceMediaNumber);
          console.log("translatedNumber=" +
conversation.remoteParticipants[i].translatedNumber);
          console.log("voiceMediaPhoneType=" +
conversation.remoteParticipants[i].voiceMediaPhoneType);
          console.log("voiceMediaState=" +
conversation.remoteParticipants[i].voiceMediaState);
        }
        console.log("conversation localParticipant=");
        console.log("  voiceMediaDisplayName=" +
conversation.localParticipant.voiceMediaDisplayName);
       console.log("  voiceMediaNumber=" + conversation.localParticipant.voiceMediaNumber);

       console.log("  translatedNumber=" + conversation.localParticipant.translatedNumber);

        console.log("  voiceMediaPhoneType=" +
conversation.localParticipant.voiceMediaPhoneType);
        console.log("  voiceMediaState=" + conversation.localParticipant.voiceMediaState);
      }
      catch(e) {
        console.log("cannot parse conversation:" + e.message);
      }
    }
```

Custom Embedded Tab Example

The following is an example of a configuration file with one embedded tab:

```
<?xml version="1.0" encoding="utf-8"?>
<config version="1.0">
 <Client>
  <jabber-plugin-config>
   <browser-plugin>
     <page refresh ="true" preload="true">
     <tooltip>Cisco</tooltip>
     <icon>https://www.cisco.com/web/fw/i/logo.gif</icon>
     <url>https://www.cisco.com</url>
    </page>
   </browser-plugin>
  </jabber-plugin-config>
 </Client>
</config>
```

Custom Emoticons

Applies to: Cisco Jabber for Windows

You can add custom emoticons to Cisco Jabber for Windows by creating emoticon definitions in an XML file and saving it to the file system.

Note To achieve optimal results, your custom emoticons should conform to the following guidelines:

- Dimensions: 17 x 17 pixels
- Transparent background
- PNG file format
- RGB colors

Procedure

Step 1 Create a file named `emoticonDefs.xml` with any text editor.

Step 2 Specify the emoticon definitions as appropriate in `emoticonDefs.xml`.

See *Emoticon Definitions* for more information on the structure and available parameters for `emoticonDefs.xml`.

Step 3 Save and close `emoticonDefs.xml`.

Step 4 Save `emoticonDefs.xml` in the appropriate directory on the file system.

Cisco Jabber for Windows loads emoticon definitions from the following directories on the file system.

- The directory can differ depending on your operating system
  - For 32-bit operating systems:
    - `Program Files\Cisco Systems\Cisco Jabber\Emoticons`
    - `Program Files\Cisco Systems\Cisco Jabber\CustomEmoticons`
  - For 64-bit operating systems:
    - `Program Files(x86)\Cisco Systems\Cisco Jabber\Emoticons`
    - `Program Files(x86)\Cisco Systems\Cisco Jabber\CustomEmoticons`

  The `Emoticons` folder contains the default emoticons for Cisco Jabber for Windows and the default `emoticonDefs.xml`.

  The `CustomEmoticons` folder does not exist by default. Administrators can create this folder to contain custom emoticon definitions to include in organizational deployments.

  Emoticons that you define in the `CustomEmoticons` folder take precedence over emoticon definitions in the default `Emoticons` folder.

- *`%USERPROFILE%`*`\AppData\Roaming\Cisco\Unified Communications\Jabber\CSF\CustomEmoticons`

  This folder contains custom emoticon definitions for individual instances of Cisco Jabber for Windows.

  Emoticons that you define in this directory take precedence over emoticon definitions in the `CustomEmoticons` folder in the installation directory.

Step 5 Restart Cisco Jabber for Windows.

Cisco Jabber for Windows loads the custom emoticon definitions in `emoticonDefs.xml`.

Remember Custom emoticon definitions are available to users only if they are defined locally in `emoticonDefs.xml`. If you send custom emoticons to users who do not have the same emoticon definitions, those users receive the default keys, not the icons; for example:

1. User A defines a custom emoticon in `emoticonDefs.xml`.

   The custom emoticon definition exists only on User A's local file system.
2. User A sends that custom emoticon to User B.
3. User B receives only the default key for the custom emoticon. User B does not receive the icon.

Emoticon Definitions

Cisco Jabber for Windows loads emoticon definitions from `emoticonDefs.xml`.

The following XML snippet shows the basic structure for the emoticon definitions file:

```
<emoticons>
 <emoticon defaultKey="" image="" text="" order="" hidden="">
  <alt></alt>
 </emoticon>
</emoticons>
```

The following table describes the elements and attributes for defining custom emoticons:

Element or attribute	Description
emoticons	This element contains all emoticon definitions.
emoticon	This element contains the definition of an emoticon.
defaultKey	This attribute defines the default key combination that renders the emoticon. Specify any key combination as the value. This attribute is required. defaultKey is an attribute of the emoticon element.
image	This attribute specifies the filename of the emoticon image. Specify the filename of the emoticon as the value. The emoticon image must exist in the same directory as `emoticonDefs.xml`. This attribute is required. Cisco Jabber for Windows supports any icon that Internet Explorer can render, including `.jpeg`, `.png`, and `.gif`. image is an attribute of the emoticon element.

Element or attribute	Description
text	This attribute defines the descriptive text that displays in the **Insert emoticon** dialog box. Specify any string of unicode characters. This attribute is optional. text is an attribute of the emoticon element.
order	This attribute defines the order in which emoticons display in the **Insert emoticon** dialog box. Specify an ordinal number beginning from 1 as the value. order is an attribute of the emoticon element. This attribute is required. However, if the value of hidden is **`true`** this parameter does not take effect.
hidden	This attribute specifies whether the emoticon displays in the **Insert emoticon** dialog box. Specify one of the following as the value: **true** Specifies the emoticon does not display in the **Insert emoticon** dialog box. Users must enter the key combination to render the emoticon. **false** Specifies the emoticon displays in the **Insert emoticon** dialog box. Users can select the emoticon from the **Insert emoticon** dialog box or enter the key combination to render the emoticon. This is the default value. This attribute is optional. hidden is an attribute of the emoticon element.
alt	This element enables you to map key combinations to emoticons. Specify any key combination as the value. For example, if the value of defaultKey is **`:)`**, you can specify **`:-)`** as the value of alt so that both key combinations render the same emoticon. This element is optional.

Remember The default emoticons definitions file contains the following key combinations that enable users to request calls from other users:

- :callme
- :telephone

These key combinations send the callme emoticon, or communicon. Users who receive this emoticon can click the icon to initiate an audio call. You should include these key combinations in any custom emoticons definition file to enable the callme emoticon.

Emoticon Definition Example

```
<emoticons>
 <emoticon defaultKey=":)" image="Emoticons_Smiling.png" text="Smile" order="1">
  <alt>:-)</alt>
  <alt>^_^</alt>
 </emoticon>
 <emoticon defaultKey=":(" image="Emoticons_Frowning.png" text="Frown" order="2">
  <alt>:-(</alt>
 </emoticon>
</emoticons>
```