Cisco Jabber for Windows and Mac Features

Call Pickup

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

The Call Pickup feature allows users to answer calls that come in on a directory number other than their own. Directory numbers are assigned to call pickup groups and Cisco Unified Communications Manager automatically dials the appropriate call pickup group number. Users select **Pickup** to answer the call.

Group call pickup allows users to pick up incoming calls in another group. Users enter the group pickup number, select **Pickup** and Cisco Unified Communications Manager automatically dials the appropriate call pickup group number.

Other group pickup allows users to pick up incoming calls in a group that is associated with their group. When the user selects **Other Pickup** Cisco Unified Communications Manager automatically searches for the incoming call in the associated groups.

Directed call pickup allows users to pick up an incoming call on a directory number. Users enter the directory number, select **Pickup** and Cisco Unified Communications Manager connects the incoming call.

For more information about configuring call pickup, see the *Feature Configuration Guide for Cisco Unified Communications Manager*.

Call pickup notifications

For multiple incoming calls, the notification displayed is *Call(s) available for pickup*. When the user answers a call, the user gets connected to the incoming call that has been ringing the longest.

Deskphone mode

In deskphone mode the following limitations apply:

- The Cisco Unified Communications Manager notification settings are not supported for the pickup group. The call pickup notification displayed is *CallerA->CallerB*.
- The Cisco Unified Communications Manager settings for audio and visual settings are not supported. The visual alerts are always displayed.

Shared line behavior

For users that have a deskphone and a CSF softphone with a shared line the following limitations apply:

- Attempt to pick up a call using the softphone when there is no call available, *No call available for PickUp* is displayed on the deskphone.
- Attempt to pick up a call using the deskphone when there is no call available, *No call available for PickUp* is displayed on the softphone.

User not a member of an associated group

For an incoming call to another pickup group where the user is not a member of an associated group:

- Directed call pickup can be used to pick up the incoming call.
- Group pickup does not work

Expected behavior using group call pickup and directed call pickup

The following are expected behaviors when using group call pickup and directed call pickup:

- Enter an invalid number
  - Softphone mode—The conversation window appears and the annunciator is heard immediately.
  - Deskphone mode—The conversation window, fast busy tone, or the annunciator followed by the fast busy tone, *Pickup failed* error message.
- Enter a valid number and no current call available to pick up
  - Softphone mode—Tone in headset, no conversation window appears and *No call available for pickup* error message.
  - Deskphone mode—No conversation window and *No call available for pickup* error message.
- Enter directory number of a phone in an associated group and no current call available to pick up
  - Softphone mode—Tone in headset, no conversation window appears and *No call available for pickup* error message.
  - Deskphone mode—No conversation window and *No call available for pickup* error message.
- Enter a directory number of a phone on the same Cisco Unified Communications Manager node and not in an associated group
  - Softphone mode—Conversation window appears and fast busy tone.
  - Deskphone mode—Conversation window appears, fast busy tone, and *Pickup failed* error message.

- Enter first digits of a valid group
  - Softphone mode—Tone in headset, conversation window appears, and after 15 seconds annunciator followed by the fast busy tone.
  - Deskphone mode—Conversation window appears, after 15 seconds annunciator, fast busy tone, and *Pickup failed* error message.

Call pickup using a deskphone that is not in a call pickup group

If a user attempts a call pickup from a deskphone that is not in a call pickup group, the conversation window appears for a moment. The user should not be configured to use the call pickup feature if they are not members of a call pickup group.

Original recipient information not available

When the Cisco Unified Communications Manager *Auto Call Pickup Enabled* setting is true, the recipient information is not available in the client when the call is picked up in softphone mode. If the setting is false, the recipient information is available.

Configure Call Pickup Group

Call pickup groups allow users to pick up incoming calls in their own group.

Procedure

Step 1 Open the **Cisco Unified Communication Manager** interface.

Step 2 Select **Call Routing** > **Call Pickup Group**
The **Find and List Call Pickup Groups** window opens.

Step 3 Select **Add New**
The **Call Pickup Group Configuration** window opens.

Step 4 Enter call pickup group information:

a) Specify a unique name for the call pickup group.
b) Specify a unique directory number for the call pickup group number.
c) Enter a description.
d) Select a partition.

Step 5 (Optional) Configure the audio or visual notification in the **Call Pickup Group Notification Settings** section.

a) Select the notification policy.
b) Specify the notification timer.

For further information on call pickup group notification settings see the call pickup topics in the relevant Cisco Unified Communications Manager documentation.

Step 6 Select **Save**.

What to Do Next

Assign a call pickup group to directory numbers.

Assign Directory Number

Assign a call pickup group to a directory number. Only directory numbers that are assigned to a call pickup group can use call pickup, group call pickup, other group pickup, and directed call pickup.

Before You Begin

Before you assign a call pickup group to a directory number, you must create the call pickup group.

Procedure

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

Step 2 Assign a call pickup group to a directory number using one of the following methods:

- Select **Call Routing** > **Directory Number**, find and select your directory number and in the Call Forward and Call Pickup Settings area select the call pickup group from the call pickup group drop down list.
- Select **Device** > **Phone**, find and select your phone and in the **Association Information** list choose the directory number to which the call pickup group will be assigned.

Step 3 To save the changes in the database, select **Save**.

Configure Other Call Pickup

Other Group Pickup allows users to pick up incoming calls in an associated group. Cisco Unified Communications Manager automatically searches for the incoming call in the associated groups to make the call connection when the user selects **Other Pickup**.

Before You Begin

Before you begin, configure call pickup groups.

Procedure

Step 1 Open the **Cisco Unified Communication Manager Administration** interface.

Step 2 Select **Call Routing** > **Call Pickup Group**
The **Find and List Call Pickup Groups** window opens.

Step 3 Select your call pickup group.
The **Call Pickup Group Configuration** window opens.

Step 4 In the **Associated Call Pickup Group Information** section, you can do the following:

- Find call pickup groups and add to current associated call pickup groups.

- Reorder associated call pickup groups or remove call pickup groups.

Step 5 Select **Save**.

Configure Directed Call Pickup

Directed call pickup allows you to pick up a incoming call directly. The user enters the directory number in the client and selects **Pickup**. Cisco Unified Communications Manager uses the associated group mechanism to control if the user can pick up an incoming call using Directed Call Pickup.

To enable directed call pickup, the associated groups of the user must contain the pickup group to which the directory number belongs.

When the user invokes the feature and enters a directory number to pick up an incoming call, the user connects to the call that is incoming to the specified phone whether or not the call is the longest incoming call in the call pickup group to which the directory number belongs.

Procedure

Step 1 Configure call pickup groups and add associated groups. The associated groups list can include up to 10 groups.
For more information, see topics related to defining a pickup group for Other Group Pickup.

Step 2 Enable the Auto Call Pickup Enabled service parameter to automatically answer calls for directed call pickups.
For more information, see topics related to configuring Auto Call Pickup.

Auto Call Pickup

You can automate call pickup, group pickup, other group pickup, and directed call pickup by enabling the Auto Call Pickup Enabled service parameter. When this parameter is enabled, Cisco Unified Communications Manager automatically connects users to the incoming call in their own pickup group, in another pickup group, or a pickup group that is associated with their own group after users select the appropriate pickup on the phone. This action requires only one keystroke.

Auto call pickup connects the user to an incoming call in the group of the user. When the user selects **Pickup** on the client, Cisco Unified Communications Manager locates the incoming call in the group and completes the call connection. If automation is not enabled, the user must select **Pickup** and answer the call, to make the call connection.

Auto group call pickup connects the user to an incoming call in another pickup group. The user enters the group number of another pickup group and selects **Pickup** on the client. Upon receiving the pickup group number, Cisco Unified Communications Manager completes the call connection. If auto group call pickup is not enabled, dial the group number of another pickup group, select **Pickup** on the client, and answer the call to make the connection.

Auto other group pickup connects the user to an incoming call in a group that is associated with the group of the user. The user selects **Other Pickup** on the client. Cisco Unified Communications Manager automatically searches for the incoming call in the associated groups in the sequence that the administrator enters in the **Call Pickup Group Configuration** window and completes the call connection after the call is found. If automation is not enabled, the user must select **Other Pickup**, and answer the call to make the call connection.

Auto directed call pickup connects the user to an incoming call in a group that is associated with the group of the user. The user enters the directory number of the ringing phone and selects **Pickup** on the client. Upon receiving the directory number, Cisco Unified Communications Manager completes the call connection. If auto directed call pickup is not enabled, the user must dial the directory number of the ringing phone, select **Pickup**, and answer the call that will now ring on the user phone to make the connection.

For more information about **Call Pickup**, see the *Feature Configuration Guide for Cisco Unified Communications Manager*.

Configure Auto Call Pickup

Procedure

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

Step 2 Select **System** > **Service Parameters**

Step 3 Select your server from the Server drop down list and then select the **Cisco Call Manager** service from the Service drop down list.

Step 4 In the **Clusterwide Parameters (Feature - Call Pickup)** section, select one of the following for **Auto Call Pickup Enabled**:

- true**—**The auto call pickup feature is enabled.
- false**—**The auto call pickup feature is not enabled. This is the default value.

Step 5 Select **Save**.

Persistent Chat Rooms

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

Configure Persistent Chat

Persistent chat must be enabled and configured on Cisco Unified Communications Manager IM and Presence Service before it can be used by the client.

Before You Begin

Persistent chat is only available on Cisco Unified Communications Manager IM and Presence Service 10.0 and later.

Refer to *Database Setup for IM and Presence Service on Cisco Unified Communications Manager* for your release for information on the database configuration necessary to support the persistent chat feature. Database configuration must be performed before continuing with this task.

Local chat message archiving must be enabled for persistent chat. Local chat message archiving is enabled on Cisco Unified Communications Manager IM and Presence Service using the **Allow clients to log instant message history** setting, for more information, see the *Enable Message Settings* topic.

Procedure

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

Step 2 Select **Messaging** > **Group Chat and Persistent Chat**.

Step 3 Select **Enable Persistent Chat**.

Step 4 Ensure the settings **How many users can be in a room at one time** and **How many hidden users can be in a room at one time** under the **Occupancy Settings** section contain the same, non-zero value.

Step 5 Configure the remaining settings as appropriate for your persistent chat deployment. We recommend the persistent chat settings in the following table.

Persistent Chat Setting	Recommended Value	Notes
System automatically manages primary group chat server aliases	Disabled	
Enable persistent chat	Enabled	
Archive all room joins and exits	Administrator Defined	This value is not currently used by for persistent chat.
Archive all room messages	Enabled	
Allow only group chat system administrators to create persistent chat rooms	Administrator Defined	Cisco recommends using the value Enabled unless Cisco Unified Personal Communicator is deployed in the enterprise environment.
Maximum number of persistent chat rooms allowed	Administrator Defined	
Number of connections to the database	Default Value	
Database connection heartbeat interval (seconds)	Default Value	
Timeout value for persistent chat rooms (minutes)	Default Value	
Maximum number of rooms allowed	Default Value	
Rooms are for members only by default	Disabled	

Persistent Chat Setting	Recommended Value	Notes
Room owners can change whether or not rooms are for members only	Enabled	Cisco Jabber requires this value to be Enabled.
Only moderators can invite people to members-only rooms	Enabled	Cisco Jabber requires this value to be Enabled.
Room owners can change whether or not only moderators can invite people to members-only rooms	Enabled	
Users can add themselves to rooms as members	Disabled	This value is not currently used by Cisco Jabber for persistent chat.
Room owners can change whether users can add themselves to rooms as members	Disabled	This value is not currently used by Cisco Jabber for persistent chat.
Members and administrators who are not in a room are still visible in the room	Enabled	Cisco Jabber requires this value to be Enabled.
Room owners can change whether members and administrators who are not in a room are still visible in the room	Enabled	
Rooms are backwards-compatible with older clients	Disabled	This value is not currently used by Cisco Jabber for persistent chat.
Room owners can change whether rooms are backwards-compatible with older clients	Disabled	This value is not currently used by Cisco Jabber for persistent chat.
Rooms are anonymous by default	Disabled	This value is not currently supported by Cisco Jabber for persistent chat. Cisco Jabber cannot join anonymous rooms.
Room owners can change whether or not rooms are anonymous	Disabled	This value is not currently supported by Cisco Jabber for persistent chat. Cisco Jabber cannot join anonymous rooms.
Lowest participation level a user can have to invite others to the room	Default Value	This value is not currently used by Cisco Jabber for persistent chat.
Room owners can change the lowest participation level a user can have to invite others to the room	Disabled	This value is not currently used by Cisco Jabber for persistent chat.
How many users can be in a room at one time	Administrator Defined	Cisco recommends using the default value.

Persistent Chat Setting	Recommended Value	Notes
How many hidden users can be in a room at one time	Administrator Defined	
Default maximum occupancy for a room	Default Value	
Room owners can change default maximum occupancy for a room	Default Value	
Lowest participation level a user can have to send a private message from within the room	Default Value	
Room owners can change the lowest participation level a user can have to send a private message from within the room	Default Value	
Lowest participation level a user can have to change a room's subject	Moderator	
Room owners can change the lowest participation level a user can have to change a room's subject	Disabled	
Remove all XHTML formatting from messages	Disabled	This value is not currently used by Cisco Jabber for persistent chat.
Room owners can change XHTML formatting setting	Disabled	This value is not currently used by Cisco Jabber for persistent chat.
Rooms are moderated by default	Disabled	This value is not currently used by Cisco Jabber for persistent chat.
Room owners can change whether rooms are moderated by default	Default Value	This value is not currently used by Cisco Jabber for persistent chat.
Maximum number of messages that can be retrieved from the archive	Default Value	
Number of messages in chat history displayed by default	Administrator Defined	Cisco recommends a value between 15 and 50. The **Number of messages in chat history displayed by default** setting does not apply retroactively to persistent chat rooms. Rooms created before the setting is changed will continue to use their originally configured value.

Persistent Chat Setting	Recommended Value	Notes
Room owners can change the number of messages displayed in chat history	Default Value	This value is not currently used by Cisco Jabber for persistent chat.

Note Persistent Chat rooms inherit their settings at the time of creation. Values changed after a room is created only apply to rooms created after the change has taken effect.

What to Do Next

Ensure you configure any client-specific parameters for persistent chat. For more information, see the Client parameters section of the latest Parameters Reference Guide for Cisco Jabber.

Enable file transfer in chat rooms. For more information, see Enable File Transfer and Screen Captures for Group Chats and Chat Rooms.

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.

Enable Persistent Chat Room Passwords

Persistent chat rooms that are password protected means that when users enter a room within a Jabber session, they must enter the password. Password protected rooms comply with the XEP-0045 specification from the XMPP Standards Foundation.

Procedure

Step 1 To set a password for a room, from the **Chat Rooms** tab on the hub window, select **All rooms** > **New room** > **Password**.

Step 2 To change the password for a room, open the chat room, click on **Edit Room**, select **Password**, then edit and save the password.

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.

Parameter	Description
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 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.
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.

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>
```

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.

Restore Chats on Login

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

This feature allows users to specify if open chat sessions are restored on next sign in. This only applies to 1:1 chats.

For desktop clients, this feature is configured using the RestoreChatOnLogin parameter. When the parameter is true the **Remember my open conversations** check box is selected on the **General** tab of the clients. The check box is not checked by default when users sign into Cisco Jabber for the first time.

For more information on parameters, see the *Parameter Reference Guide* for your release.