プラットフォーム

カスタム埋め込みタブ

Clients			
ウィンドウ	MAC	iPhone および iPad	Android
はい	はい	はい	はい

導入			
On-Premises	Webex Messenger	チームメッセージングモード	VDI 版ソフトフォン
はい	はい	はい	はい

カスタム埋め込みタブは、クライアントインターフェイスでHTMLコンテンツを表示します。Cisco Jabber のカスタム埋め込みタブ定義を作成します。これで、プログラムに従ってカスタムタブを現在のクライアントテーマに調整できるようになりました。

（注）

- Jabber 埋め込みブラウザでは、SSO が有効になっている Web ページからのポップアップのある cookie の共有はサポートされていません。ポップアップウィンドウのコンテンツをロードできない場合があります。
- エンタープライズ内の任意の Web サービス (アクセスの必要がある Jabber クライアントをリモート処理する) の HTTP サーバ許可リスト（ホワイトリスト）を設定します。詳細については、『*Mobile and Remote Access via Cisco Expressway Deployment Guide*』を参照してください。

カスタム埋め込みタブの定義

カスタム組み込みタブは jabber-config.xml ファイルを使用してか設定できます。次の XML スニペットに、カスタム タブ定義の構造を示します。

```
<jabber-plugin-config>
 <browser-plugin>
  <page refresh="" preload="" internal="">
   <tooltip></tooltip>
   <icon></icon>
   <url></url>
  </page>
 </browser-plugin>
</jabber-plugin-config>
```

Windows 版 Cisco Jabber は、カスタム埋め込みタブのコンテンツを表示する Chromium 埋め込みフレームワークを使用します。

Mac 版 Cisco Jabber は Safari WebKit のレンダリング エンジンを使用して埋め込みタブのコンテンツを表示します。

次の表は、カスタム埋め込みタブ定義のパラメータについて説明します。

パラメータ	説明
browser-plugin	カスタム埋め込みタブのすべての定義が含まれます。 値にはすべてのカスタム タブ定義が含まれます。
ページ	1 つのカスタム埋め込みタブ定義が含まれます。
refresh	コンテンツが更新されるときを制御します。 • true: ユーザがタブを選択するたびに、コンテンツが更新されます。 • false（デフォルト）：ユーザがクライアントを再起動するか、またはサインインしたときに、コンテンツが更新されます。 このパラメータはオプションであり、ページ要素の属性です。
preload	コンテンツが読み込まれるときを制御します。 • true：クライアントが開始したときにコンテンツがロードされます。 • false（デフォルト）：ユーザがタブを選択したときにコンテンツがロードされます。 このパラメータはオプションであり、ページ要素の属性です。
tooltip	カスタム埋め込みタブのマウスを置いたときに表示されるテキストを定義します。 値は Unicode 文字の文字列です。

パラメータ	説明
icon	タブのアイコンを指定します。ローカル アイコンまたはホステッド アイコンを指定できます。 • ローカル アイコン：次のように URL を指定します。`file://file_path/icon_name` • ホステッドアイコン：次のようにURLを指定します。`http://path/icon_name` 次のように、ローカルおよびホストされたアイコンを、*% jabber theme%* 変数を使用して現在の Jabber のテーマに一致するように変更することもできます。`http://path/Icon_name_%JabberTheme%. jpg` Jabber クライアントは、アイコンを要求するときに*%Jabber* テーマ*%*を解釈します。値は次のいずれかになります。 • デフォルト: デフォルトの Jabber のテーマ • dark: Jabber の 「ダーク」テーマ • distinct: Jabber の 「ハイコントラスト」テーマ • highcontrast: Windows のハイコントラストテーマ URLに*% jaburl%*が含まれていない場合 、そのテーマによってアイコンが変更されることはありません。各アイコンのファイル名にテーマ名を含めることができます。カスタムアイコンのダウンロードに失敗した場合、または選択したテーマのアイコンが存在しない場合は、Jabber クライアントがデフォルトの画像を使用します。 .JPG、.PNG、および .GIF 形式など、クライアントブラウザがレンダリング可能であれば、どのアイコンでも使用できます。 このパラメータはオプションです。アイコンを指定しない場合、クライアントはHTMLページからファビコンを読み込みます。ファビコンがない場合、クライアントはデフォルト アイコンを読み込みます。

パラメータ	説明
url	埋め込みタブのコンテンツが格納される URL を指定します。 クライアントは、ブラウザのレンダリングエンジンを使用して埋め込みタブのコンテンツを表示します。そのため、ブラウザがサポートするコンテンツであれば、どのコンテンツでも指定できます。 Mac 版 Cisco Jabber の場合、URL 要素に HTTP または HTTPS が含まれていなければなりません。 このパラメータは必須です。 （注） ターゲットの Web ページで Windows 統合認証が必要な場合、Jabber はユーザにサインイン資格情報を要求するメッセージをデフォルトで表示します。このメッセージが表示されないようにするには、Windows レジストリで認証サーバのホワイトリストを設定します。 次の場所にホワイトリストされた URL を追加します: • `HKEY_LOCAL_MACHINE \SOFTWARE\Policies\Google\Chrome\AuthServerWhitelist` • `HKEY_LOCAL_MACHINE \SOFTWARE\Policies\Google\Chrome\AuthNegotiateDelegateWhitelist` たとえば、`AuthServerWhitelist` リストと`AuthNegotiateDelegateWhitelist`ポリシーを * example.com、* `foobar.com`、`* baz` に設定するとします。' example.com '、' foobar.com '、または ' baz ' のいずれかで終わる Url が許可リストに含まれています。'*' プレフィックスがない場合、URL は完全に一致する必要があります。
internal	Web ページが、ネットワークの内部ページまたは外部ページであることを指定します。 • True (デフォルト): Web ページはネットワークの内部ページです。 • false: Web ページはネットワークの外部ページです。

ユーザのカスタム タブ

クライアントユーザインターフェイスを使用して、ユーザに、カスタム埋め込みタブに対するタブ名および URL を指定することを許可します。ユーザは、カスタム埋め込みタブの他のパラメータを設定できません。

ユーザが自分のタブをカスタマイズできるようにするには、AllowUserCustomTabsを**`true`**に設定します。

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

（注） AllowUserCustomTabsの既定値は**true**です。

カスタム アイコン

最適な結果を得るには、カスタム アイコンは次のガイドラインに準拠する必要があります。

- 寸法：20 X 20 ピクセル
- 透明な背景
- PNG ファイル形式

特定のテーマに合わせて、さまざまなバージョンのアイコンを含めることができます。詳細については、icon パラメータの説明を参照してください。

カスタム タブからのチャットとコール

プロトコル ハンドラを使用し、カスタム埋め込みタブからチャットやコールを開始できます。カスタム埋め込みタブが HTML ページであることを確認してください。

チャットを開始するには、`XMPP:` または `IM:` プロトコル ハンドラを使用します。
音声通話やビデオ通話を開始するには、`TEL:` プロトコル ハンドラを使用します。

UserID トークン

`${UserID}` トークンを url パラメータの値の一部として指定できます。ユーザがサインインすると、クライアントにより、`${UserID}` トークンがログインしたユーザのユーザ名に置き換えられます。

ヒント また、クエリー文字列に `${UserID}` トークンを指定することもできます（例：`www.cisco.com/mywebapp.op?url=${UserID}`）。

次の例は、`${UserID}` トークンの使用方法を示しています。

1. カスタム埋め込みタブで次の内容を指定します。

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

2. Mary Smith がサインインします。Mary のユーザ名は msmith です。
3. クライアントにより、`${UserID}` トークンが次のようにMaryのユーザ名に置き換えられます。

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

JavaScript 通知

カスタム埋め込みタブに JavaScript 通知を実装できます。ここでは、JavaScript 通知用にクライアントが提供するメソッドについて説明します。また、通知のテストに使用できる JavaScript フォームの例についても説明します。非同期サーバ コールに対する JavaScript 通知の実装方法と他のカスタム実装に関する説明は、このマニュアルでは取り扱いません。詳細については、該当する JavaScript のマニュアルを参照してください。

通知メソッド

クライアントには、JavaScript 通知の次のメソッドを提供するインターフェイスが含まれています。

- SetNotificationBadge：クライアントから JavaScript でこのメソッドを呼び出します。このメソッドの文字列の値は、次のいずれかになります。
  - 空：空の値にすると、既存の通知バッジすべてが削除されます。
  - 1 ~ 999 の数字
  - 2 桁の英数字の組み合わせ（例：A1）
- onPageSelected()：ユーザがカスタム組み込みタブを選択すると、クライアントはこのメソッドを呼び出します。
- onPageDeselected()：ユーザが別のタブを選択すると、クライアントはこのメソッドを呼び出します。

（注） iPhone および iPad 版 Jabber には適用されません

- onHubResized()：ユーザがクライアント ハブ ウィンドウをサイズ変更または移動すると、クライアントはこのメソッドを呼び出します。
- onHubActivated()：クライアント ハブ ウィンドウがアクティブになると、クライアントはこのメソッドを呼び出します。
- onHubDeActivated()：クライアント ハブ ウィンドウが非アクティブになると、クライアントはこのメソッドを呼び出します。

カスタム タブでのプレゼンスの登録

次の JavaScript メソッドを使用して、連絡先のプレゼンスを登録し、クライアントからプレゼンスの更新情報を受信することができます。

- SubscribePresence()：このメソッドにユーザの IM アドレスを使用してストリング値を指定します。

- OnPresenceStateChanged：このメソッドを使用すると、ユーザは連絡先のプレゼンスに関してクライアントから更新情報を取得できます。文字列として次の値のいずれか1つを指定できます。
  - IM アドレス
  - 基本プレゼンス（Available（対応可）、Away（不在）、Offline（オフライン）、Do Not Disturb（着信拒否））
  - 高度なプレゼンス（会議中、コール中、カスタム プレゼンス状態）

（注）
- [連絡先] リストに登録されていないユーザのサブスクリプションは、68 分後に期限切れになります。サブスクリプションの有効期限が切れた後、そのユーザのプレゼンスデータを表示するには、再度サブスクライブする必要があります。
- iPhone および iPad 版 Jabber は、OnPresenceStateChanged のみサポートしています。

Custom タブでのロケール情報の取得

次の JavaScript メソッドを使用すると、連絡先の現在のロケール情報をクライアントから取得できます。

- GetUserLocale()：このメソッドを使用して、クライアントにロケール情報を要求できます。
- OnLocaleInfoAvailable：このメソッドを使用して、クライアントからロケール情報を取得できます。クライアントのロケール情報を含む文字列値を使用できます。

（注）
iPhone および iPad 版 Jabber は、OnLocaleInfoAvailable のみサポートしています。

クライアントテーマに合わせてカスタムタブを調整する

次の JavaScript メソッドを使用して、現在のクライアントテーマを返すことができます。

- QueryCurrentTheme (): このメソッドを使用すると、現在の Jabber のテーマを取得できます。カスタムタブのページで、この方法を次のように使用します。
  `window.external.QueryCurrentTheme()` 。
- Onchanges Echanged (テーマ): Jabber でテーマが変更された場合、このメソッドは受動的に新しいテーマを受け取ります。

テーマでは、次の値を使用できます。

- デフォルト: デフォルトの Jabber のテーマ
- dark: Jabber の 「ダーク」テーマ

- distinct: Jabber の 「ハイコントラスト」テーマ
- highcontrast: Windows のハイコントラストテーマ

JavaScript の例

1 ～ 999 の数字を入力できるフォームを表示するために JavaScript を使用する HTML の例を示します。

```
<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>
```

この例の JavaScript フォームをテストするには、前述の例を HTML ページにコピーしてから、そのページをカスタム埋め込みタブとして指定します。

カスタム タブでのコール イベントの表示

カスタム タブでコール イベントを表示するために次の JavaScript 関数を使用できます。

OnTelephonyConversationStateChanged：テレフォニー サービスの API は、クライアントがカスタム埋め込みタブでコール イベントを表示できるようにします。カスタム タブに `OnTelephonyConversationStateChanged` JavaScript 関数を実装できます。クライアントはテレフォニーのカンバセーションの状態が変化するたびに、この関数を呼び出します。この関数は、クライアントがコールイベントを取得するために解析する JSON 文字列を受け入れます。

次のスニペットは、コール イベントを保持する JSON を示します。

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

JSON の各参加者オブジェクトには、次のプロパティを含めることができます。

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

次に、この関数のカスタム埋め込みタブでの実装例を示します。この例では、state と acceptanceState プロパティの値を取得し、これらの値をカスタム タブに表示します。

```
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);
      }
    }
```

次に、この関数の考えられるすべてのフィールドでの実装例を示します。

```
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);
      }
    }
```

カスタム埋め込みタブの例

次に、1 つの埋め込みタブを含む設定ファイルの例を示します。

```
<?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>
```

Chromebook Android 版 Cisco Jabber の設定

Clients			
ウィンドウ	MAC	iPhone および iPad	Android
—	—	—	あり

導入			
On-Premises	Webex Messenger	チームメッセージングモード	VDI 版ソフトフォン
はい	はい	はい	—

Chromebook Android 版 Cisco Jabber の設定チェックリスト

タスク	詳細
デバイスモデルと OS がサポートされていることを確認する	サポートされている *Android OS* 要件と *Chromebook* モデルを参照してください
企業のネットワークに MRA 設定を追加する	「企業のネットワークでの*MRA*設定の追加」を参照してください。
Chromebook ユーザをタブ (TAB) デバイスタイプとして設定する	タブ *(TAB)* デバイスタイプとしての *Chromebook* ユーザの設定を参照してください。
必要なポートを開いたままにして、ユーザが Chromebook のすべての Cisco Jabber サービスにアクセスできるようにします。	ポートの設定を参照

サポートされている Android OS 要件および Chromebook モデル

Chromebook に Chrome OS v53 以降が搭載されている必要があります。Android 版 Cisco Jabber は、Google Play ストアからダウンロードすることができます。

サポートされている Chromebook モデル:

- HP Chromebook 13 G1 ノートブック PC
- Google Chromebook Pixel
- Samsung Chromebook Pro

企業のネットワークに MRA 設定を追加する

企業およびモバイルおよび Remote Access (MRA) ネットワークから接続している間は、Cisco Jabber を使用します。コールサービスを使用するには、Cisco Jabber が MRA ネットワークを使用してログインしている必要があります。

ユーザーが企業ネットワーク内で動作しているときに MRA ネットワークに接続するには、内部ドメインネームサーバ (DNS) を "_collab-edge._tls.<domain>.com" SRV レコードで設定します。DNS の完全な詳細については、「サービスの発見*Cisco Jabber* プランニングガイド *12.1*を参照してください。

Chromebook ユーザをタブ (TAB) デバイスタイプとして設定する

Chromebook ユーザをタブ (TAB) デバイスタイプとして設定します。ユーザ向けソフトフォンサービス設定方法の詳細については、*Cisco Jabber* のオンプレミス展開ガイドのソフトフォンの設定を参照してください。

ポートの設定

これらのポートが Chromebook で Cisco Jabber サービスにアクセスできることを確認してください:

目的	プロトコル	オンプレミス ネットワーク (送信元)	Expressway-E (宛先)
XMPP (IM & P)	TCP	>=1024	5222
HTTP プロキシ (UDS)	TCP	>=1024	8443
Media	UDP	>=1024	36002 ～ 59999
SIP シグナリング	TLS	>=1024	5061

制限事項

ビデオコール中に、ユーザが別のアプリに切り替えた場合、ビデオは停止します。

Cisco Jabber モバイルアプリのプロモーション

Clients			
ウィンドウ	MAC	iPhone および iPad	Android
対応	—	—	—

導入			
On-Premises	Webex Messenger	チームメッセージングモード	VDI 版ソフトフォン
はい	はい	はい	はい

Windows 版 Cisco Jabber のユーザの通知を有効にして、モバイルアプリ 版 (Android および iOS) Cisco Jabberの使用を促進することができます。通知をクリックすると、[**設定 (** Settings)] ページがユーザに移動し、Google Play または iTunes ストアからアプリをダウンロードするかを選択できるようになります。これらの通知をコントロールするように、新しいパラメータ EnablePromoteMobileが追加されています。この機能はデフォルトで無効に設定されています。

このパラメータ設定の詳細情報については、『*Parameter Reference Guide for Cisco Jabber*』を参照してください。