Cisco Meeting Server

Cisco Meeting Server 2.5, 2.6, and 2.7
Customization Guidelines

December 03, 2019
# Contents

Change History .................................................................................................................. 5

1 Introduction ..................................................................................................................... 6
   1.1 What’s new or changed in Cisco Meeting Server 2.4 that affects customization ...... 7
   1.2 What can be branded? ............................................................................................... 7
   1.3 Web Server Requirements ....................................................................................... 7
   1.4 Using this guide ........................................................................................................ 8

1 Introduction ..................................................................................................................... 9
   1.5 What’s new or changed in Cisco Meeting Server 2.5 that affects customization ...10
   1.6 What are the ways to manage branding files? ........................................................... 10
   1.7 What can be branded? ............................................................................................... 11
   1.8 Web Server Requirements ....................................................................................... 11
   1.9 Using this guide ........................................................................................................ 12

2 WebRTC App customization .............................................................................................. 13
   2.1 File Properties and Names ....................................................................................... 14
      2.1.1 Sign-in page background image and logo ......................................................... 15
      2.1.2 sign_in_settings.json parameters ................................................................ 15
      2.1.3 Other settings in sign_in_settings.json ........................................................... 15
   2.2 Overview of customization procedure (web server hosted branding) .................... 16

3 Call customization ............................................................................................................ 17
   3.1 IVR Message customization ..................................................................................... 17
      3.1.1 IVR Messages to be customized ..................................................................... 17
      3.1.2 Recording format for IVR messages .............................................................. 18
      3.1.3 IVR background image properties .................................................................. 18
      3.1.4 Overview of customization procedure (web server hosted branding) .......... 18
   3.2 SIP/Lync Call Message Customization .................................................................... 19
      3.2.1 Audio messages to be customized ................................................................. 19
      3.2.2 Recording format for audio messages ............................................................ 22
      3.2.3 Background image properties ....................................................................... 23
      3.2.4 Overview of customization procedure (web server hosted branding) .......... 23
      3.2.5 Missing and invalid files ............................................................................... 24

4 Invitation text customization ............................................................................................ 26
   4.1 Creating the invitation_template.txt ...................................................................... 26
4.1.1 Permitted conditional statements and placeholders ...................................26
4.1.2 Example invitation template .......................................................................27
4.2 Overview of customization procedure (web server hosted branding) ...............28

5 Implementing locally hosted branding ................................................................30
5.1 WebRTC app customizations ..........................................................................30
5.2 IVR Message, SIP/Lync Call Message and Invitation Text customization ..........31
5.3 Testing customized invitation_template.txt ....................................................32
5.4 Removing locally hosted branding files ..........................................................32

6 Detailed Customization Procedure Using Postman ............................................33
6.1 Setting up the web server ..................................................................................33
6.2 Using the API for branding ...............................................................................36
   6.2.1 Installing Chrome Postman ..........................................................................36
   6.2.2 Applying branding to WebRTC apps ............................................................39
   6.2.3 Applying branding to SIP calls and/or customized invitations ....................40
   6.2.4 Applying branding to an IVR .......................................................................42
6.3 Using the API for branding ...............................................................................45
   6.3.1 Installing RESTer .......................................................................................45
   6.3.2 Add/Confirm SSL Exception .........................................................................46
   6.3.3 Generate a Basic Auth header ......................................................................46
   6.3.4 Applying branding to WebRTC apps ............................................................50
   6.3.5 Applying branding to SIP calls and/or customized invitations ....................54
   6.3.6 Applying branding to an IVR .......................................................................56

7 Detailed customization procedure using RESTer ...............................................60
7.1 Setting up the web server ..................................................................................60
7.2 Using the API for branding ...............................................................................63
   7.2.1 Installing RESTer .......................................................................................63
   7.2.2 Add/Confirm SSL Exception .........................................................................65
   7.2.3 Generate a Basic Auth header ......................................................................65
   7.2.4 Applying branding to WebRTC apps ............................................................68
   7.2.5 Applying branding to SIP calls and/or customized invitations ....................72
   7.2.6 Applying branding to an IVR .......................................................................75

8 Switching between branding methods ...............................................................78
8.1 Changing from web server (or default) branding to locally hosted branding ....78
8.2 Changing from locally hosted to web server branding .....................................78
8.3 Mixing locally hosted and web server customization ........................................79
8.4 Limitations of using locally hosted branding ..................................................79

Appendix A Using Other Popular API Tools ..........................................................80
   A.1 Using Firefox Poster ....................................................................................80
      A.1.1 Applying branding to SIP calls using Firefox Poster ............................82
   A.2 Using Firefox Poster ....................................................................................85
      A.2.1 Applying branding to SIP calls using Firefox Poster ............................88
   A.3 Using Chrome Advanced Rest Client ..........................................................91
      A.3.1 Applying branding to SIP calls using Chrome Advanced Rest Client ....94

Cisco Legal Information .........................................................................................97
Cisco Trademark ......................................................................................................98

Figures:

   Figure 1: Overview of guides covering the Meeting Server ...............................6
   Figure 2: Overview of guides covering the Meeting Server ...............................9
   Figure 3: WebRTC app assets ............................................................................13
   Figure 4: Creating zip file for WebRTC app assets ...........................................14
   Figure 5: Example contents of sign_in_settings.json .....................................15
## Change History

<table>
<thead>
<tr>
<th>Date</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>December 03, 2019</td>
<td>Edits to a few audio prompt descriptions.</td>
</tr>
<tr>
<td>May 10, 2019</td>
<td>Applies to 2.5 and later. API tools information updated.</td>
</tr>
<tr>
<td>December 13, 2018</td>
<td>New version of guide for 2.5, from this version you have the option of hosting one set of branding files locally on Meeting Server. The time delay to apply new branding settings has been reduced to a maximum of 1 hour whichever method is chosen (local or remote).</td>
</tr>
<tr>
<td>December 05, 2018</td>
<td><a href="#">Launch link information in example invitation template updated.</a></td>
</tr>
<tr>
<td>September 21, 2018</td>
<td>New version of guide for 2.4, branding license no longer required.</td>
</tr>
</tbody>
</table>
1 Introduction

The Cisco Meeting Server software can be hosted on specific servers based on Cisco Unified Computing Server (UCS) technology as well as on the X-Series hardware, or on a specification-based VM server. Cisco Meeting Server is referred to as the Meeting Server throughout this document.

This document covers the features for customizing release 2.6 of the Meeting Server software. These features allow modification of the voice prompts, background images and logos, and the text shown in invitations.

It follows on from the appropriate Deployment Guide (see Figure 1).

Figure 1: Overview of guides covering the Meeting Server
1.1 What’s new or changed in Cisco Meeting Server 2.4 that affects customization

A license key is no longer required to apply single or multiple brandings, see Section 1.2

1.2 What can be branded?

Some aspects of the participant experience of meetings hosted on Meeting Servers can be branded, they include:

- the WebRTC app sign in background image, sign in logo, text below sign in logo and the text on the browser tab,
- IVR messages,
- SIP and Lync participant’s splash screen images and all audio prompts/messages,
- some text on the meeting invitation.

From version 2.4, no license is required to apply single or multiple brands to these customizable features. If you apply a single brand with only a single set of resources specified (one WebRTC app sign–in page, one set of voice prompts, one invitation text), then these resources are used for all spaces, IVRs and Web Bridges in the deployment. Multiple brandings allow different resources to be used for different spaces, IVRs and Web Bridges. Resources can be assigned at the system, tenant, space or IVR level using the API.

brandings, or use the Web Admin interface if you only wish to apply a single brand to the IVR messages, SIP/Lync call messages or meeting invites. The Customization Guidelines describes the elements that can be branded and explains how to apply the branding files. [pastacey: remove as webadmin can't be used for this anymore]

Note: The WebRTC app sign–in page cannot be rebranded using the Web Admin interface, nor can multiple brandings be applied via the Web Admin Interface.

1.3 Web Server Requirements

All the customizations described in this document require a directory on a web server on which .wav, jpg, png or archive (e.g. zip) files can be stored. The web server must be reachable by the Call Bridge, and must not require the Call Bridge to perform any form of HTTP authentication.

For more information on setting up the web server, see Section 6.1.

The import occurs when the Call Bridge first needs to use the customized files.

Note: If you require multiple brandings, for example one per tenant, then you need to deploy a separate Web Bridge for each branding. Each can be a standalone Web Bridge on a VM server.
1.4 Using this guide

Chapter 2 covers branding the WebRTC app. It details the elements that can be branded on the WebRTC app and explains how to customize the app with your branding.

Chapter 3 covers branding the images and messages that participants see and hear when dialing into an IVR or dialing directly into a space using a SIP endpoint or Lync.

Chapter 4 describes customizing the text shown in invitations which is sent to participants inviting them to join a call or space.

Chapter 6 provides a step by step procedure on customizing calls using the Chrome Postman application.

As an alternative to Chrome Postman, Appendix A details the steps on customizing calls using Firefox Poster and Chrome Advanced Rest Client.
1 Introduction

The Cisco Meeting Server software can be hosted on specific servers based on Cisco Unified Computing Server (UCS) technology as well as on the X-Series hardware, or on a specification-based VM server. Cisco Meeting Server is referred to as the Meeting Server throughout this document.

This document covers the features for customizing release 2.6 of the Meeting Server software. These features allow modification of the voice prompts, background images and logos, and the text shown in invitations.

It follows on from the appropriate Deployment Guide (see Figure 2).

Figure 2: Overview of guides covering the Meeting Server
1.5 What’s new or changed in Cisco Meeting Server 2.5 that affects customization

From this version, you have the option of either using a web server to store branding files or storing one set of branding files locally on the Meeting Server.

Prior to version 2.5, using branding files for the Meeting Server required you to configure a separate web server to hold the branding files (voice prompts and lobby screen branding assets). From version 2.5 one set of branding files can be held locally on the Meeting Server. These locally hosted branding files are available to the Call Bridge and Web Bridge once the Meeting Server is operational, removing the risk of delays in applying customization due to problems with the web server. The images and audio prompts replace the equivalent files built into the Meeting Server software; during start up, these branding files are detected and used instead of the default files. Locally hosted branding files are overridden by any remote branding from a web server.

You can change these locally hosted files simply by uploading a newer version of the files and restarting the Call Bridge and Web Bridge. If you remove the locally hosted files, the Meeting Server will revert to using the built-in (US English) branding files after the Call Bridge and Web Bridge have been restarted, providing a web server has not been set up to provide the branding files.

From version 2.5, if you do not manually restart the Call Bridge, the time delay to switch over to new branding settings has been reduced to a maximum of 1 hour whichever method is chosen (local or remote); for older versions of Meeting Server, the delay could be as much as 24 hours. If you restart the Call Bridge, changes take immediate effect.

See Chapter 5 for the steps to follow to implement locally hosted branding.

Note: To use multiple sets of branding files, you still need to use an external web server.

From version 2.4, a license key is no longer required to apply single or multiple brandings, see Section 1.7

1.6 What are the ways to manage branding files?

From version 2.5, you have the following options for managing branding files:

- Store one set of branding files locally on the Meeting Server.
- Store branding files on a web server, see Web Server Requirements.

Note: We recommend NOT mixing locally hosted and web server customization. For more information, see Switching between branding methods.
1.7 What can be branded?

If you are using a web server to store your branding files, then the list below can be branded. If you have chosen locally hosted branding, see Section 8.4 for limitations which apply.

Some aspects of the participant experience of meetings hosted on Meeting Servers can be branded, they include:

- the WebRTC app sign in background image, sign in logo, text below sign in logo and the text on the browser tab,
- IVR messages,
- SIP and Lync participant’s splash screen images and all audio prompts/messages,
- some text on the meeting invitation.

From version 2.4, no license is required to apply single or multiple brands to these customizable features. If you apply a single brand with only a single set of resources specified (one WebRTC app sign-in page, one set of voice prompts, one invitation text), then these resources are used for all spaces, IVRs and Web Bridges in the deployment. Multiple brandings allow different resources to be used for different spaces, IVRs and Web Bridges. Resources can be assigned at the system, tenant, space or IVR level using the API.

brandings, or use the Web Admin interface if you only wish to apply a single brand to the IVR messages, SIP/Lync call messages or meeting invites. The Customization Guidelines describes the elements that can be branded and explains how to apply the branding files. [pastacey: remove as webadmin can’t be used for this anymore]

**Note:** The WebRTC app sign-in page cannot be rebranded using the Web Admin interface, nor can multiple brandings be applied via the Web Admin Interface.

1.8 Web Server Requirements

If you intend to use remote branding, you will require a directory on a web server on which .wav, .jpg, .png or archive (e.g. zip) files can be stored. The web server must be reachable by the Call Bridge, and must not require the Call Bridge to perform any form of HTTP authentication. For more information on setting up the web server, see Section 7.1.

The import occurs when the Call Bridge first needs to use the customized files.

**Note:** If you require multiple brandings, for example one per tenant, then you need to deploy a separate Web Bridge for each branding. Each can be a standalone Web Bridge on a VM server.
1.9 Using this guide

Chapter 2 covers branding the WebRTC app. It details the elements that can be branded on the WebRTC app and explains how to customize the app with your branding.

Chapter 3 covers branding the images and messages that participants see and hear when dialing into an IVR or dialing directly into a space using a SIP endpoint or Lync.

Chapter 4 describes customizing the text shown in invitations which is sent to participants inviting them to join a call or space.

Chapter 5 describes the procedure to implementing customization from locally hosted branding files.

Chapter 7 provides a step-by-step procedure example on customizing calls using the RESTer API tool.

Chapter 8 provides instructions to switch from one method of branding to another and also limitations

Note: All information applies to web server branding and locally hosted branding unless stated otherwise.

As an alternative to Chrome Postman, Appendix A details the steps on customizing calls using Firefox Poster and Chrome Advanced Rest Client.
2 WebRTC App customization

Use the API to customize these elements of the WebRTC app:

- sign-in background image,
- sign-in dialog box — icon displayed,
- sign-in dialog box — text below icon
- text on browser tab

Figure 3: WebRTC app assets

For web server hosted branding, the branding files are held within an archive (zip) file stored on the web server. The location of this resourceArchive is set via the API (a POST method to the "/webBridges" node or a PUT to a "/webBridges/<web bridge id>").

The Call Bridge retrieves the archive file from the web server and pushes it to the Web Bridge to be served. In the event of a failure (for example, if the configured URI can’t be reached) an alarm is displayed in the Web Admin Interface and on the API "/system/alarms" node, but users can still log in using the WebRTC app.

For locally hosted branding information on where and how to locate the zipped branding files, see Chapter 5.

Without a branding license you can only control the background image and logo on the WebRTC landing page of a single Web Bridge, and this must be done via the Web Admin Interface Configuration > General page.
2.1 File Properties and Names

The branding files must be placed together in an archive file such as a zip file. The total file size of the zipped file must be less than 1 MB, and the maximum decompressed size of any one file is 512 KB. For web server branding, the zip file is retrieved by the Call Bridge and then used by the Web Bridge to brand the WebRTC app. For locally hosted branding, the zip file is retrieved locally by the Web Bridge to brand the WebRTC app.

When you zip the files, do not zip the folder containing the branded files. If this is done, this will create an extra layer of folder (zipped file -> folder -> branded files). Instead, highlight the branded files and right-click to zip them (or open a zip application and zip the files together). This will create a zipped file with the branded files without creating an extra layer of folder (e.g. zipped file -> branded files).

For example, in Figure 4, WebRTC_client.zip folder contains the branded files so when configuring for Web Bridge customization for remote branding, you can use the path http://<webserver address>/Branding/webRTC_client.zip.

Figure 4: Creating zip file for WebRTC app assets

Table 1: WebRTC app assets

<table>
<thead>
<tr>
<th>Use</th>
<th>Filename to use (filenames are case sensitive)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign in page background image</td>
<td>sign_in_background.jpg</td>
</tr>
<tr>
<td>Sign in page logo</td>
<td>sign_in_logo.png</td>
</tr>
<tr>
<td>Text below sign in logo and text on browser tab</td>
<td>sign_in_settings.json</td>
</tr>
</tbody>
</table>
2.1.1 Sign-in page background image and logo

The background image must be in .jpg format and the logo file must be in .png format. Transparency in .png files is supported, and recommended, for the logo.

Maximum size for the background image is 1920 pixels wide and 1200 pixels high, and less than 500 kB in size. It will be scaled isotropically and then cropped at either the right or bottom to fit the browser window.

The recommended resolution for the logo image is 254 pixels wide by 64 pixels high; these dimensions fill the space to the edges of the Sign in box horizontally. This file must be less than 250 kB in size.

2.1.2 sign_in_settings.json parameters

Table 2: sign_in_settings.json parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>browserTabLabel</td>
<td>Text on browser tab. Recommend 20 characters or less.</td>
</tr>
<tr>
<td>panelLabel</td>
<td>Text below icon on sign in dialog box. Recommend 20 characters or less.</td>
</tr>
<tr>
<td>allowClient</td>
<td>Controls whether the option to “Launch desktop application” is displayed on the screen that displays different ways to join a meeting. Default is True.</td>
</tr>
<tr>
<td>allowWebRTC</td>
<td>Controls whether the option to &quot;Use this computer&quot; is displayed on the same screen. Default is True.</td>
</tr>
</tbody>
</table>

Figure 5: Example contents of sign_in_settings.json

```
{
    "browserTabLabel" : "Cisco Meeting App",
    "panelLabel" : "Signing on",
    "allowClient" : "true",
    "allowWebRTC" : "true"
}
```

Note: The sign_in_settings.json file must use straight quotes. If your system uses smart quotes by default, you must change the setting.

Note: If you are testing your branding, you need to refresh the page twice (or use CTRL+F5 on Google Chrome) as the browser does not actually re-fetch images on a single refresh.

2.1.3 Other settings in sign_in_settings.json

Other settings in sign_in_settings.json are shown in the table below:
Table 3: Other parameters in sign_in_settings.json

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>supportFirefox</td>
<td>No effect</td>
</tr>
<tr>
<td>supportSafari</td>
<td>No effect</td>
</tr>
<tr>
<td>supportInternetExplorer</td>
<td>No effect</td>
</tr>
<tr>
<td>supportWebkitWebrtc</td>
<td>No effect</td>
</tr>
<tr>
<td>supportFirefoxWebrtc</td>
<td>No effect</td>
</tr>
<tr>
<td>showDialInformation</td>
<td>No effect</td>
</tr>
<tr>
<td>showFooter</td>
<td>No effect</td>
</tr>
</tbody>
</table>

2.2 Overview of customization procedure (web server hosted branding)

These steps give an overview of the customization procedure, for a detailed procedure see Chapter 7.

1. Create a zip archive file containing these files:
   - sign_in_background.jpg
   - sign_in_logo.png
   - sign_in_settings.json

2. On the web server create a directory for the customization files and place the zip file in it.

3. Using the API, create a webBridge node specifying the resourceArchive field and URL on the web server where the zip file is held. (See Chapter 7.2.4 for details). The Meeting Server uses these details to retrieve the zip file from the web server.

Note: You can only apply one archive file per Web Bridge.

Note: If you specify a port value in the path, this will be used in place of the default port values of :80 for http and :443 for https. If you do not specify a port value then the default value will be used. If you specify a port value make sure that the web server used in step 2 is listening on this port.
3 Call customization

There are two types of call customization:

- Messages heard and image seen when dialing into an IVR
- Messages heard and images seen when dialing directly as a SIP (including Lync) call into a space on a Meeting Server

**Note:** The use of API objects `/ivrBrandingProfiles` and `/callBrandingProfiles` requires a branding license.

### 3.1 IVR Message customization

The messages heard when calling an IVR can be customized via the API by using the `ivrBrandingProfile`. This profile can be applied at the system level, on a per-tenant basis or for individual IVRs.

#### 3.1.1 IVR Messages to be customized

The following table lists all the IVR messages required for and the associated filenames to use for the recordings.

**Table 4: IVR messages for customization**

<table>
<thead>
<tr>
<th>Text of message</th>
<th>Filename to use (filenames are case sensitive)</th>
<th>Played when ……</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please enter the call ID, followed by the '#' (pound) key.</td>
<td><code>ivr_id_entry.wav</code></td>
<td>dialling via IVR to enter a specific space</td>
</tr>
<tr>
<td>Unable to recognize that call ID. Please try again.</td>
<td><code>ivr_id_incorrect_try_again.wav</code></td>
<td>the incorrect call ID is entered to join the space</td>
</tr>
<tr>
<td>Please try again: this is your last attempt.</td>
<td><code>ivr_id_incorrect_final_attempt.wav</code></td>
<td>two incorrect pins/call ID's have been entered to join the space</td>
</tr>
<tr>
<td>Unable to recognize that call ID. Goodbye.</td>
<td><code>ivr_id_incorrect_goodbye.wav</code></td>
<td>entering three incorrect call ID's to join the space</td>
</tr>
<tr>
<td>Welcome to a Cisco meeting.</td>
<td><code>ivr_welcome.wav</code></td>
<td>joining a space</td>
</tr>
<tr>
<td>Unable to connect you. Goodbye.</td>
<td><code>ivr_timeout.wav</code></td>
<td>after dialling via IVR and not entering the call ID, the call times out</td>
</tr>
</tbody>
</table>
3.1.2 Recording format for IVR messages

Use audio files created with Audacity and saved as a single track PCM16, at 8, 16, 22.05, 32, 44.1 or 48 kHz sample rate, 16 bits per sample and mono. The file size for each recording must be less than 400kB.

**Note:** We have not tested other audio tools and, if used, they may cause problems. [pastacey: remove as we’ve not mentioned ANY audio tools up to this point so makes no sense]

**Note:** It is not necessary to include any additional periods of silence at the start or end of these prompts.

3.1.3 IVR background image properties

The background image file must be .jpg format, less than 500kB in size and a maximum of 1920 pixels wide by 1200 pixels. Images will be centered, scaled and padded with black to preserve their aspect ratio. Progressive JPEG is not supported.

**Table 5:** IVR background image for customization

<table>
<thead>
<tr>
<th>Image use</th>
<th>Filename to use (filenames are case sensitive)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IVR background image</td>
<td>ivr_background.jpg</td>
</tr>
</tbody>
</table>

3.1.4 Overview of customization procedure (web server hosted branding)

The following steps provide an overview of the customization procedure, for a detailed procedure refer to Chapter 7.

**Note:** If using web server hosted branding, ensure that your web server is configured correctly with the proper MIME type for the WAV and JPEG file extensions (e.g. audio/wav and image/jpeg MIME type).

1. On the web server create a directory for the IVR customization files and place in it all the files listed in Section 3.1.1 and Section 3.1.3. Do not zip these files

   Cisco have created an archive with their audio and video files, the zip files are downloadable from the Customization subsection on this web page. The files can be individually used when a mix of custom and default files are required. The necessary files must be copied into the directory. If any files are missing then no alternative or default file will be used in their place.

2. Using the API, create an ivrBrandingProfile specifying the resourceLocation field and URL on the web server where the customized ivr files are held. (See Section 7.2.6 for details). The Meeting Server uses these details to retrieve the files from the web server.

3. Use this ivrBrandingProfile to customize your calls on:
a. a system level (that is, as part of the global systems profile)
b. a per-tenant basis (as part of the tenant definition)
c. a per-IVR basis (as part of an IVR definition)

**Note:** When IVR branding is applied at more than one level, then the lowest level ivrBrandingProfile defined is the one that it used in a call leg. Only the ivrBrandingProfile applied at the system level is used for single branding. See the Using Profiles section in the API Reference.

### 3.2 SIP/Lync Call Message Customization

This section describes how to customize the in-call experience when dialing into a Meeting Server space from a SIP or Lync endpoint.

For web server hosted branding, the files are retained on the Call Bridge while they are in continuous use for one or more calls. If you change the background image or the audio messages, then the new images and audio files will not be utilized until any calls utilizing call branding have completed, and the files have been re-fetched.

#### 3.2.1 Audio messages to be customized

The following list is all the audio messages required for and the associated filenames to use for the recordings.
<table>
<thead>
<tr>
<th>Text of message</th>
<th>Filename to use (filenames are case sensitive)</th>
<th>Repeats for audio calls</th>
<th>Played when ......</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome to a Cisco meeting</td>
<td>welcome.wav</td>
<td>No</td>
<td>joining a call</td>
</tr>
<tr>
<td>I haven’t been able to connect you. Goodbye.</td>
<td>timeout.wav</td>
<td>No</td>
<td>after dialling via an IVR and not entering the call id, the call times out</td>
</tr>
<tr>
<td>Hello. You are invited to a Cisco call.</td>
<td>call_outgoing_welcome.wav</td>
<td>No</td>
<td>when confirmation is set to True in the API, “call_” prompts are played from a call that isn’t associated with a cospace</td>
</tr>
<tr>
<td>Press ‘1’ to join the call.</td>
<td>call_join_confirmation.wav</td>
<td>No</td>
<td>after the &quot;Hello. You are invited to a Cisco call.&quot; prompt is played</td>
</tr>
<tr>
<td>You are joining the call now.</td>
<td>call_join.wav</td>
<td>No</td>
<td>after pressing ‘1’ to join the call</td>
</tr>
<tr>
<td>Hello. You are invited to a Cisco meeting.</td>
<td>cospace_outgoing_welcome.wav</td>
<td>No</td>
<td>when confirmation is set to True in the API, “cospace_” prompts are played in configured meetings (e.g. cospace meetings)</td>
</tr>
<tr>
<td>Press ‘1’ to enter the meeting.</td>
<td>cospace_join_confirmation.wav</td>
<td>No</td>
<td>after the &quot;Hello. You are invited to a Cisco meeting.&quot; prompt is played</td>
</tr>
<tr>
<td>You are entering the meeting now.</td>
<td>cospace_join.wav</td>
<td>No</td>
<td>after dialling the URI of a space from a SIP endpoint, or after dialing the IVR and entering the call ID of the space (plays after PIN if space has a PIN)</td>
</tr>
<tr>
<td>You have been disconnected from the meeting</td>
<td>disconnected.wav</td>
<td>No</td>
<td>the participant has been disconnected from the meeting.</td>
</tr>
<tr>
<td>This meeting is being recorded</td>
<td>meeting_recorded.wav</td>
<td>No</td>
<td>recording starts or when joining a call that is being recorded</td>
</tr>
<tr>
<td>This meeting is no longer being recorded</td>
<td>meeting_recording Ended.wav</td>
<td>No</td>
<td>recording ends</td>
</tr>
<tr>
<td>This meeting is being streamed</td>
<td>meeting_streamed.wav</td>
<td>No</td>
<td>streaming starts (from version 2.1)</td>
</tr>
<tr>
<td>Text of message</td>
<td>Filename to use (filenames are case sensitive)</td>
<td>Repeats for audio calls</td>
<td>Played when ......</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------</td>
<td>-------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>This meeting is no longer being streamed</td>
<td>meeting_streaming Ended.wav</td>
<td>No</td>
<td>streaming stops (from version 2.1)</td>
</tr>
<tr>
<td>Your meeting has ended</td>
<td>meeting Ended.wav</td>
<td>No</td>
<td>the meeting has ended.</td>
</tr>
<tr>
<td>You are the only participant</td>
<td>only_participant.wav</td>
<td>Yes</td>
<td>there is only one participant in the meeting.</td>
</tr>
<tr>
<td>a joining tone</td>
<td>participant_join.wav</td>
<td>No</td>
<td>a new participant joins the space.</td>
</tr>
<tr>
<td>a leaving tone</td>
<td>participant_leave.wav</td>
<td>No</td>
<td>a participant leaves the space.</td>
</tr>
<tr>
<td>Please enter the PIN, followed by the '#' (pound) key. If you have a PIN then please enter it, followed by the '#' (pound) key, otherwise, please press the '#' (pound) key</td>
<td>passcode_entry.wav</td>
<td>No</td>
<td>a PIN is required to enter the space</td>
</tr>
<tr>
<td>Please enter the PIN, followed by the '#' (pound) key. If you have a PIN then please enter it, followed by the '#' (pound) key, otherwise, please wait</td>
<td>passcode_or_blank_required_entry.wav</td>
<td>No</td>
<td>a PIN is required for the host to enter the space as host, but guests only need to use the # (pound) key (from version 2.1)</td>
</tr>
<tr>
<td>That PIN isn’t correct. Please try again.</td>
<td>passcode_incorrect_try_again.wav</td>
<td>No</td>
<td>the incorrect PIN is entered to join the space</td>
</tr>
</tbody>
</table>
### 3.2.2 Recording format for audio messages

If you plan to record your own audio messages, then save each prompt as type “WAV (Microsoft) signed 16 bit PCM”. Audio files can be converted into the correct format using Audacity. The project rate should be one of 8000, 16000, 22050, 32000, 44100, 48000 Hz.

If creating files using another application, the output must be:

- single track PCM format
- 16 bits per sample
- Mono
- 8, 16, 22.05, 32, 44.1 or 48 kHz sample rate

For recordings that repeat, the file size must be less than 1000kB. This is sufficient for 32 seconds when using a 16 kHz sample rate. The file size for all other recording must be less than 400kB.

**Note:** We have not tested other audio tools and, if used, they may cause problems.

**Note:** It is not necessary to include any additional periods of silence at the start or end of the prompts that do not repeat. For those that repeat, silence can be used to create a gap between the instances.
3.2.3 Background image properties

The background file must be in .jpg format, less than 500kB in size and a maximum of 1920 pixels wide and 1200 pixels high. Images will be centered, scaled and padded to preserve their aspect ratio. Progressive JPEG is not supported.

Table 7: SIP call background image for customization

<table>
<thead>
<tr>
<th>Image use</th>
<th>Filename to use (filenames are case sensitive)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call logo</td>
<td>background.jpg</td>
</tr>
<tr>
<td>Screen can be shown when a PIN needs to be entered.</td>
<td>passcode_background.jpg</td>
</tr>
<tr>
<td>Screen can be shown when a PIN is required for the host to enter the space as host, but guests only need to use the # (pound) key (from version 2.1)</td>
<td>passcode_or_blank_required_background.jpg</td>
</tr>
<tr>
<td>Screen can be shown when a PIN is required to enter the coSpace as host, but guests join after a short timeout (from version 2.1)</td>
<td>passcode_or_blank_timeout_background.jpg</td>
</tr>
<tr>
<td>Screen can be shown when awaiting activation, (waiting for host to join, meeting needs to be unlocked etc).</td>
<td>deactivated_background.jpg</td>
</tr>
</tbody>
</table>

Note: The behaviour and choice of background image for /callBrandingProfiles does not affect the /ivrBrandingProfiles

Note: In locally hosted branding, only background.jpg will be used for the call background and ivr background images; passcode_background.jpg, passcode_or_blank_required_background.jpg, passcode_or_blank_timeout_background.jpg, deactivated_background.jpg and ivr_background.jpg are ignored.

3.2.4 Overview of customization procedure (web server hosted branding)

The following steps provide an overview of the customization procedure, for a detailed procedure example, see Chapter 7. For steps on implementing locally hosted branding, see Chapter 5.

Note: If using web server hosted branding, ensure that your web server is configured correctly with the proper MIME type for the WAV and JPEG file extensions (e.g. audio/wav and image/jpeg MIME type).

1. On the web server create a directory for the customization files and place in it all the files listed in Section 3.2.1 and Section 3.2.3. Do not zip these files.
Cisco have created an archive with their audio and video files, the zip files are downloadable from the Customization subsection on this web page. The files can be individually used when a mix of custom and default files are required. The necessary files must be copied into the directory. If any files are missing then no alternative or default file will be used in their place.

2. Using the API, create a callBrandingProfile specifying the resourceLocation field and URL on the webserver where the customized files are held. (See Section 7.2.5 for details). The Meeting Server uses these details to retrieve the files from the web server.

**Note:** If no resourceLocation is specified, then by default the Cisco logo will be used.

3. Use this callBrandingProfile to customize your SIP/Lync calls on:
   a. a system level (that is, as part of the global systems profile)
   b. a per-tenant basis (as part of the tenant definition)
   c. a per-space basis (as part of the space definition)

**Note:** When call branding is applied at more than one level, then the lowest level callBrandingProfile defined is the one that it used in a call leg. Only the callBrandingProfile applied at the system level is used for single branding. See the Using Profiles section in the API Reference.

### 3.2.5 Missing and invalid files

Files missing from the web server and those with an invalid format are ignored unless the alternative or default file is available.

**Table 8: Alternative and default files**

<table>
<thead>
<tr>
<th>Missing or invalid file</th>
<th>Alternative or default file</th>
</tr>
</thead>
<tbody>
<tr>
<td>background.jpg</td>
<td>No alternative file, screen remains black.</td>
</tr>
<tr>
<td>passcode_background.jpg</td>
<td>background.jpg *</td>
</tr>
<tr>
<td>passcode_or_blank_required_background.jpg</td>
<td>background.jpg *</td>
</tr>
<tr>
<td>passcode_or_blank_timeout_background.jpg</td>
<td>background.jpg *</td>
</tr>
<tr>
<td>deactivated_background.jpg</td>
<td>background.jpg *</td>
</tr>
<tr>
<td>welcome.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>timeout.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>call_join_confirmation.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>call_join.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>Missing or invalid file</td>
<td>Alternative or default file</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>call_outgoing_welcome.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>cospace_join_confirmation.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>cospace_join.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>cospace_outgoing_welcome.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>disconnected.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>meeting_recorded.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>meeting_recording_ended.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>meeting_streamed.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>meeting_streaming_ended.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>meeting_ended.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>onlyParticipant.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>participant_join.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>participant_leave.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>passcode_entry.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>passcode_or_blank_required_entry.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>passcode_or_blank_timeout_entry.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>passcode_incorrect_try_again.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>passcode_incorrect_final_attempt.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>passcode_incorrect_goodbye.wav</td>
<td>No alternative or default file **</td>
</tr>
<tr>
<td>waiting_for_host.wav</td>
<td>No alternative or default file **</td>
</tr>
</tbody>
</table>

**Note:** * You do not need to load multiple copies of background.jpg, the single file will be used automatically to replace the missing images.

**Note:** ** If this audio file is missing or format is invalid, then nothing will be played.
4 Invitation text customization

Customization of the invitation text is done via the use of a template which the Cisco Meeting App requests from the Web Bridge. The templates are cached by the app so any changes to a template will not immediately appear in the app.

4.1 Creating the invitation_template.txt

Create an invitation_template.txt file in UTF-8 using the information in Section 4.1.1. An example that you can copy and customize is shown in Section 4.1.2.

4.1.1 Permitted conditional statements and placeholders

The template can contain both conditional statements and placeholders. This allows a single template to be used for multiple spaces and gives a consistent feel to the invitations.

The placeholders that are currently defined (each starts and ends with a % character) are listed in the table below:

<table>
<thead>
<tr>
<th>Placeholder</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>%name%</td>
<td>Conference name</td>
</tr>
<tr>
<td>%uri%</td>
<td>Dial-in URI of the conference</td>
</tr>
<tr>
<td>%numeric_id%</td>
<td>Numeric ID of the conference</td>
</tr>
<tr>
<td>%hyperlink%</td>
<td>Direct hyperlink to the conference on the web bridge</td>
</tr>
<tr>
<td>%passcode%</td>
<td>Numeric PIN for the conference</td>
</tr>
<tr>
<td>%</td>
<td>Always replaced with %</td>
</tr>
</tbody>
</table>

**Note:** that any % within the text will be interpreted as the start of a placeholder. To insert a single ‘%’ in the text use the placeholder %%

<table>
<thead>
<tr>
<th>%launch_link%</th>
<th>Direct hyperlink to launch the conference in desktop or iOS Cisco Meeting App.</th>
</tr>
</thead>
</table>

**Note:** If you are using Microsoft Outlook, add a "url:" prefix to the link so it can be recognised as hyperlink in Microsoft Outlook/email clients.

The following conditional statements are supported, see table below. They can be nested if required.
Table 10: Conditional statements in invitation template

<table>
<thead>
<tr>
<th>Conditional statement</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>#if condition</td>
<td>If the condition is true then include the lines until an else or endif statement. The condition section of this if statement takes the form of one of the placeholders. For example, “#if name”</td>
</tr>
<tr>
<td>#else</td>
<td>If the condition in the previous if statement was false then include the lines until an endif statement</td>
</tr>
<tr>
<td>#endif</td>
<td></td>
</tr>
</tbody>
</table>

4.1.2 Example invitation template

Note:
- There is a 10000 byte limit on the size of the invitation_template.txt file.
- All invitation templates must be provided in UTF-8 format.
- Extended ASCII characters are not supported.
- Terminate the invitation_template.txt file with a carriage return.
- Omitting the carriage return at the end of the file will result in the file not working.

Use the example below and customize with your specific values in the placeholders. Save it as invitation_template.txt.

```plaintext
#if name
You're invited to %name%
#else
You're invited to my Cisco space
#endif

#if hyperlink
   Click to join: %hyperlink%
#else
#if numeric_id
   Click to join: https://join.example.com
   Call ID: %numeric_id%
#endif
#endif

#if hyperlink
Click to launch using Cisco Meeting App: %launch_link%
#else
Launch link not available
```
#endif

#if uri
   Or call in:
   - Video system, Jabber or Lync: %uri%
#endif

#if numeric_id
   Phone Access: Call the regional access number, then enter %numeric_id%
   US Toll Free: (800)-555-1234
   UK Toll Free: 0800-800-8000
#endif

#if passcode
   Passcode: %passcode%
#endif

Note: %launch_link% placeholder should be included in the #if_hyperlink condition so it is included in the template if hyperlinks are enabled, and excluded if hyperlinks are disabled.

4.2 Overview of customization procedure (web server hosted branding)

The following steps provide an overview of the customization procedure, for a detailed procedure refer to Chapter 7.

Note: All invitation templates must be provided in UTF-8 format. Extended ASCII characters are not supported

1. Create your invitation_template.txt file
2. On the web server create a directory for the customized invitation file and place the file invitation_template.txt into it.
3. Using the API, create a callBrandingProfile specifying the invitationTemplate field and URL on the webserver where the template is held, you need to specify the full filename including the path to the file, for example: http://192.0.2.0/branding/invitation_template.txt. (See Section 7 for details). The Meeting Server uses these details to retrieve the template from the webserver.
4. Use this callBrandingProfile to customize invitations at:
   - a system level (that is, as part of the global systems profile)
   - a per-tenant basis (as part of the tenant definition)
   - a per-space basis (as part of the space definition)
Note: When the multi brands feature key is present and call branding is applied at more than one level, then the lowest level callBrandingProfile defined is the one that is used in a call leg. If the single brand feature key is present then only the system level callBrandingProfile is used. See the Using Profiles section in the API Reference.
5 Implementing locally hosted branding

If you are using locally hosted branding files, the following steps provide an overview of the customization procedure.

**Note:** If you are changing from using a web server to hosting the files locally then follow the guidance in Section 8.1 before following the steps below.

**Note:** For limitations in using locally hosted branding, see Limitations of using locally hosted branding.

5.1 WebRTC app customizations

The branding files for the WebRTC app are held within an archive (zip) file, from version 2.5 this zip file can be locally hosted on the Meeting Server. If you are changing from using a web server to hosting the files locally then follow the guidance in Section 8.1 before following the steps below.

The following steps provide an overview of the customization procedure, for a detailed procedure refer to the Customization Guidelines.

**Note:** The commands in the following steps are for console/terminal environments (i.e. command prompt or terminal) and not for SFTP clients such as WinSCP.

1. Create a zip archive file containing these files:
   - sign_in_settings.json
   - sign_in_logo.png
   - sign_in_background.jpg

   **Note:** This zip file must be named `web_branding.zip`, it cannot have a different filename.

2. For each Meeting Server with an enabled Web Bridge which will locally host this zip archive:
   a. Connect your SFTP client to the IP address of the MMP.
   b. Log in using the credentials of the MMP admin user.
   c. Upload the zip file `web_branding.zip`. For example:
      ```
      PUT web_branding.zip
      ```
   d. Connect your SSH client to the IP address of the MMP.
   e. Log in using the credentials of the MMP admin user.
f. Restart the Web Bridge
   
   `webbridge restart`
   
   The new branding will be picked up after the restart. The Web Bridge retrieves the locally hosted branding file for the WebRTC app, rather than relying on the Call Bridge to pass the file.

5.2 IVR Message, SIP/Lync Call Message and Invitation Text customization

To locally host the IVR messages, SIP/Lync call messages and invitation text you need to create a Call Bridge branding zip file.

The following steps provide an overview of the customization procedure, for a detailed procedure refer to the Customization Guidelines.

1. Create the call branding zip file, this file must be named `call_branding.zip` to ensure it is processed correctly.

   a. Create a single folder with the files listed in Chapter 3 of the Customization Guidelines, these are the same files that are used if a web server is deployed.

   b. Create a single folder with the files listed in Chapter 3, these are the same files that are used if a web server is deployed.

   **Note:** In locally hosted branding, only `background.jpg` will be used for the call background and ivr background images; `passcode_background.jpg`, `passcode_or_blank_required_background.jpg`, `passcode_or_blank_timeout_background.jpg`, `deactivated_background.jpg` and `ivr_background.jpg` are ignored.

   c. Add the file `invitation_template.txt` containing the invitation text to the folder, as described in Chapter 4 of the guidelines.

   d. Add the file `invitation_template.txt` containing the invitation text to the folder, as described in Chapter 1.

   **Note:** For this call branding zip file, you must use the filename `invitation_template.txt` even if you are already using a different filename on a web server.

   e. Zip up the files in the folder, all files should be at the top level of the zip file (no folders nested in the zip file), the filename must be `call_branding.zip`.

2. Install the IVR, call and invitation customization on every Call Bridge. For each Meeting Server:

   a. Connect your SFTP client to the IP address of the MMP.

   b. Log in using the credentials of the MMP admin user.
c. Upload the zip file `call_branding.zip`. For example:
   ```bash
   PUT call_branding.zip
   ```
d. Connect your SSH client to the IP address of the MMP.
e. Log in using the credentials of the MMP admin user.
f. Restart the Call Bridge
   ```bash
   callbridge restart
   ```

The new branding will be picked up after the restart.

### 5.3 Testing customized `invitation_template.txt`

The invitation template is delivered from the Meeting Server to Cisco Meeting App clients and cached locally, so after customization on the Meeting Server there may be a delay before clients begin to use the new text. Logging the client out and in again should fetch the new version immediately, but any clients which stay logged in will not see the new text until their cache times out.

From version 2.5, this delay has been reduced to at most 1 hour. For clients which have cached text from a Meeting Server running an older version, the delay could be as much as 24 hours in the worst case.

### 5.4 Removing locally hosted branding files

Follow these steps for each Meeting Server hosting local branding files.

1. Connect your SFTP client to the IP address of the MMP.
2. Log in using the credentials of the MMP admin user.
3. Remove the locally hosted branding files from the Web Bridge
   ```bash
   RM web_branding.zip
   ```
4. Remove the locally hosted branding files from the Call Bridge
   ```bash
   RM call_branding.zip
   ```
5. Connect your SSH client to the IP address of the MMP.
6. Log in using the credentials of the MMP admin user.
7. Restart the Web Bridge
   ```bash
   webbridge restart
   ```
8. Restart the Call Bridge
   ```bash
   callbridge restart
   ```
6 Detailed Customization Procedure Using Postman

This section provides a step by step example of call branding using Microsoft Windows. It assumes that Microsoft Internet Information Services (IIS) Manager is installed with the latest patches on the web server that will host your call branding files.

6.1 Setting up the web server

1. On the web server create a directory structure under the default IIS Manager location c:\inetpub\wwwroot to hold the branding/custom files. The example below shows a Branding folder with four subfolders: Call_customization, Invitation_template, ivr_customization, webRTC_client.

2. Open IIS Manager and expand the Default Web Site. The newly created folders should show within the list.
**Note:** If the newly created folders do not show within the list or your folders are outside of the standard IIS Manager location, you will need to create a new Virtual Directory. Within IIS, navigate to Default Web Site, right click and select Virtual Directory. Provide an Alias for this folder and browse for the physical path where the files will reside.

3. Place all the files into the folders, for the appropriate to the level of branding you intend to do.
   a. For WebRTC app customization these will be the zip files from **Section 2.2**.

   ![WebRTC_client_folder](image)

   **Note:** in the above screenshot the WebRTC_client customization files are zipped together by location.

   b. Forivr customization these will be the files from **Section 3.1**

   ![ivr_customization_folder](image)

   **Note:** the files for ivr customization are not zipped together.
c. For call customization these will be the files from Section 3.

<table>
<thead>
<tr>
<th>Name</th>
<th>Date modified</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>background</td>
<td>11/08/2015 15:01</td>
<td>JPEG image</td>
</tr>
<tr>
<td>call_join.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>call_join_confirmation.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>call_outgoing_welcome.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>cospace_join.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>cospace_join_confirmation.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>cospace_outgoing_welcome.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>disconnected.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>meeting-ended.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>only-participant.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>passcode_entry.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>passcodeIncorrect_final_attempt.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>passcodeIncorrect_goodbye.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>passcodeIncorrect_try_again.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>timeout.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>waiting_for_host.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>welcome.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
</tbody>
</table>

Note: the files for call customization are not zipped together.

d. For the invitation template, this will be the file you created from Chapter 4.
4. Verify the web service is working. Open a browser and enter the full URL for the background image file used to customize the WebRTC app, including the file name background.jpg to see if the file opens. For example:

6.2 Using the API for branding

The remainder of this chapter covers using the Chrome tool Postman, to brand the WebRTC app, IVR, SIP calls and apply custom invitations. Appendix A details the steps to applying call customization to SIP calls using Firefox Poster, and the Chrome Advanced Rest Client.

6.2.1 Installing Chrome Postman

5. Install Postman into Chrome (if not already installed) by opening the Apps button in the upper left corner. If Postman is already installed go to step 10.

6. Click on Web Store to open the Chrome App Store
7. Enter Postman in the Search field and press Enter.

8. Install Postman – REST Client by clicking on the + FREE button to the right.

9. Click Add to continue with the installation.

Once installed, Postman – REST Client will appear on the Apps page.
10. Open Postman by clicking on the Icon from within the Apps page of Chrome

11. Select the Basic Auth tab and enter the Username and Password for the API of the Meeting Server. Click Refresh headers. Enter the URL of the Meeting Server and press SEND

A 200 OK Response should be returned.

**Note:** if you have difficulty logging in, then try to log into the Web Admin interface, accept any certificate warnings and add them as a permanent exception. (If you do not do this and there is a certificate warning, Poster will not work.)

12. To verify API connectivity is working, perform a GET operation on the /system/status node of your server.
If it fails, verify that the login is correct and that you can access the Web URL from this PC.

Once connectivity is verified, in addition to WebRTC app branding, you can apply the branded/customized files to SIP calls (see Section 6.2.3), invitations (see Section 6.2.3) and IVRs (see Section 6.2.4) at a system level, per tenant level or per space or ivr level.

6.2.2 Applying branding to WebRTC apps

Branding files for WebRTC apps are applied to a webBridge node. You can only configure one Web Bridge on a Meeting Server. If you need different WebRTC app branding for different tenants, then you need to set up a Web Bridge for each branded WebRTC app.

13. **POST** to the webBridges node the url parameter and the address for the Call Bridge to use to reach this Web Bridge. Press **Send** to submit the data.

**Note:** Select the **x-www-form-urlencoded** format.

A 200 OK Response should be returned.

14. Change the request type back to **GET**, and press **Send**. The response should look like this:
15. **PUT** to the `webBridges` node the `resourceArchive` parameter and the `url` where the WebRTC app branding files are held on your web server.

```
PUT /api/v1/webBridges/a9eaa623-837c-435e-97dc-266471677829

form-data x-www-form-urlencoded raw
resourceArchive https://10.10.10.12/Branding/webRTC
Key Value

Send Preview Add to collection
```

A 200OK Response should be returned.

16. To verify the data has been set correctly, simply change the request type to `GET` using the same `URL` as the previous `PUT`. Press Send. The response should look like this:

```
GET /api/v1/webBridges/a9eaa623-837c-435e-97dc-266471677829

Send Preview Add to collection
```

Your WebRTC app branded files should now be used for WebRTC app calls to the Web Bridge, make a test call to check that the branded files are applied.

### 6.2.3 Applying branding to SIP calls and/or customized invitations

Branding files for SIP calls and customized invitation templates are applied to the `callBrandingProfiles` node.

17. **POST** to the `callBrandingProfiles` node the parameter appropriate to the type of branding (`resourceLocation`/`invitationTemplate`) and the `urls` where the branding files are held on your
web server. Be sure to specify the full filename including the path to the file for the invitationTemplate URL only, for example: http://192.0.2.0/branding/invitation_template.txt. You can also use https. Set the request type to POST and press Send to submit the data.

**Note:** If you have created subfolders under /Branding then you need to POST the url of each subfolder to callBrandingProfiles.

A 200OK Response should be returned.

18. Change the request type back to GET, and press Send. The response should look like this:

19. Make a note of the ID number listed next to “callBrandingProfile id=”, in this example it is: 5cf83076-ee90-4e08-9d50-22d634d93c65

20. Assign this callBrandingProfile ID to the branding level required:
   - the system level (that is, as part of the global systems profile),
   - a per-tenant basis (as part of the tenant definition),
   - a per-space basis (as part of the space definition).
The example below applies the branding to a previously created tenant:

a. change the request type to PUT

b. enter the /tenant/<tenant id> in this example e46eb768-2d87-41f1-ba9b-742bb471ca61

c. enter the callBrandingProfile parameter and the id from step “Make a note of the ID number listed next to “callBrandingProfile id=”, in this example it is: 5cf83076-ee90-4e08-9d50-22d634d93c65” on the previous page

d. press Send, and you should get another 200OK Response

21. To verify the data has been set correctly, simply change the request type to GET using the same URL as the previous PUT. Press Send.

22. Your Call Branding files should now be used for calls for that tenant. Place a call from a SIP endpoint to a space setup for the tenant. Make sure the call displays the background picture, and that the voice prompt is audible.

6.2.4 Applying branding to an IVR

Branding files for IVR are applied to the ivrBrandingProfiles node.
23. POST to ivrBrandingProfiles node the resourceLocation parameter and the url where the ivr branding files are held on your web server. Be sure that you set the request type to POST. Press Send to submit the data.

**Note:** If you have created subfolders under /Branding then you need to POST the url of the ivr subfolder to ivrBrandingProfiles.

24. Change the request type back to GET, and press Send. The response should look like this:

25. Make a note of the ID number listed next to “ivrBrandingProfile id=“, in this example it is: 9d4fb901-28e8-43c3-95e5-58d1ccce781e

26. Assign this ivrBrandingProfile ID to the branding level required:
   - the system level (that is, as part of the global systems profile),
   - a per-ivr basis (as part of the ivr definition).
   - enter the ivrBrandingProfile field and from step 25

The example below applies the ivr branding to a previously created ivr (using POST to /ivrs followed by a GET to /ivrs to retrieve the ivr id):
a. change the request type to PUT

b. enter /ivr/<ivr id> in this example the ivr id = 0e045363-1275-46ad-b131-f546541fdd11

c. enter the ivrBrandingProfile field and <ivrBrandingProfile id> from step 25

![Image showing a PUT request with PUT as the method, PUT as the URL, and a form-data request body with Key: ivrBrandingProfile Value: 9d4fb0128e8-43c3-95e5-58d1c0ca78]

27. Press Send, and you should get another 200OK Response.

28. To verify the data has been set correctly, simply change the request type to GET using the same URL as the previous PUT. Press Send.

![Image showing a GET request with GET as the method, GET as the URL, and a form-data request body with Key: ivrBrandingProfile Value: 9d4fb0128e8-43c3-95e5-58d1c0ca78]

29. Your IVR Branding files should now be used for calls to the specified ivr, make test calls to the ivr to check that the branded ivr files are applied.
6.3 Using the API for branding

The remainder of this chapter covers using the RESTer API tool to brand the WebRTC app, IVR, SIP calls and apply custom invitations. Appendix A details the steps to applying call customization to SIP calls using the Chrome Advanced Rest Client.

6.3.1 Installing RESTer

1. Open Google Chrome as your browser and in the top left-hand corner of the browser window select the Apps icon. Or click: https://chrome.google.com/webstore/category/extensions in a Chrome browser window to skip to step 3.

2. Click on Web Store to open the Chrome App Store and select Extensions.

3. Enter RESTer in the Search field and press Enter, then locate the RESTer extension in the results and click Add to Chrome.
4. Click **Add extension** to continue with the installation.

5. Once installed, a confirmation dialog displays and the RESTer icon appears in the menu bar.

6.3.2 Add/Confirm SSL Exception

RESTer will follow the same SSL certificate verification and exception rules that your browser is configured for. If the URL for your webadmin site will not pass certificate validation or does not have an exception saved, RESTer will reject all requests. To check or add an exception:

1. In a Chrome window, open the URL to your Meeting Server webadmin interface, making sure you use HTTPS and the Port number (if listening on non-standard port). Example: https://cms.lab:445
2. If Chrome loads the Meeting Server web page without error you are OK to proceed.
   or
3. If Chrome shows a Privacy Error, click **Advanced** and then click the link to proceed to the address.

6.3.3 Generate a Basic Auth header

Your requests will need to authenticate to the server. RESTer can generate and save an authorization header so this step does not have to be repeated for each command.
1. If not already open, click on the RESTer icon in the browser menu bar to open the tool.

2. Click the **Authorization** tab, then click **Basic**.

3. Enter an appropriate set of credentials for your server in the pop-up that displays, and click **Save**.

4. The tab will update to show the **Authorization** tab is now using the saved credential.
## Detailed Customization Procedure Using Postman

<table>
<thead>
<tr>
<th>Method</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td><a href="https://cms.lab:445/api/v1/callLegProfiles">https://cms.lab:445/api/v1/callLegProfiles</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HEADERS</th>
<th>BODY</th>
<th>AUTHORIZATION</th>
<th>VARIABLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use existing tokens</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>admin</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Custom</td>
<td>Never expires</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generate new token</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Custom</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[is this red text section covered already by steps 1-3 above “Generate a basic Auth Header”?]

---

**Collection / Title**

<table>
<thead>
<tr>
<th>Method</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HEADERS</th>
<th>BODY</th>
<th>AUTHORIZATION</th>
<th>VARIABLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Value</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
To verify API connectivity is working, perform a GET operation on the /system/status node of your server.
If it fails, verify that the login is correct and that you can access the Web URL from this PC.

Once connectivity is verified, in addition to WebRTC app branding, you can apply the branded/customized files to SIP calls (see Section 6.2.3), invitations (see Section 6.2.3) and IVRs (see Section 6.2.4) at a system level, per tenant level or per space or ivr level.

[Need all steps below updated with RESTer screen shots]

6.3.4 Applying branding to WebRTC apps

Branding files for WebRTC apps are applied to a webBridge node. You can only configure one Web Bridge on an Meeting Server. If you need different WebRTC app branding for different tenants, then you need to set up a Web Bridge for each branded WebRTC app.
13. **POST** to the webBridges node the url parameter and the address for the Call Bridge to use to reach this Web Bridge. Press **Send** to submit the data.

**Note:** Select the x-www-form-urlencoded format. [pastacey - can’t locate on RESTer??]

select Body and then form template in rester

A 200 OK Response should be returned.

![Collection / Title](image)

14. Change the request type back to **GET**, and press **Send**. The response should look like this:
15. PUT to the webBridges node the resourceArchive parameter and the url where the WebRTC app branding files are held on your web server. A 200 OK Response should be returned.
16. To verify the data has been set correctly, simply change the request type to GET using the same URL as the previous PUT. Press Send. The response should look like this:

Your WebRTC app branded files should now be used for WebRTC app calls to the Web Bridge, make a test call to check that the branded files are applied.
6.3.5 Applying branding to SIP calls and/or customized invitations

Branding files for SIP calls and customized invitation templates are applied to the callBrandingProfiles node.

17. POST to the callBrandingProfiles node the parameter appropriate to the type of branding (resourceLocation/invitationTemplate) and the urls where the branding files are held on your web server. Be sure to specify the full filename including the path to the file for the invitationTemplate URL only, for example: https://192.0.2.23/branding/invitation_template.txt. You can also use http. Set the request type to POST and press Send to submit the data.

**Note:** If you have created subfolders under /Branding then you need to POST the url of each subfolder to callBrandingProfiles.

A 200 OK Response should be returned.

18. Change the request type back to GET and press Send. The response should look like this:
19. Make a note of the ID number listed next to "callBrandingProfile id=" in this example it is: 5cf83076-ee90-4e08-9d50-22d634d93c65

20. Assign this callBrandingProfile ID to the branding level required:
   - the system level (that is, as part of the global systems profile),
   - a per-tenant basis (as part of the tenant definition),
   - a per-space basis (as part of the space definition).

   The example below applies the branding to a previously created tenant:
   a. change the request type to PUT
   b. enter the /tenant/<tenant id> in this example e46eb768-2d87-41f1-ba9b-742bb471ca61
   c. enter the callBrandingProfile parameter and the id from step "Make a note of the ID number listed next to "callBrandingProfile id=" in this example it is: 5cf83076-ee90-4e08-9d50-22d634d93c65" on page 41
   d. press Send, and you should get another 200OK Response

21. To verify the data has been set correctly, simply change the request type to GET using the same URL as the previous PUT. Press Send.
22. Your Call Branding files should now be used for calls for that tenant. Place a call from a SIP endpoint to a space setup for the tenant. Make sure the call displays the background picture, and that the voice prompt is audible.

### 6.3.6 Applying branding to an IVR

Branding files for IVR are applied to the ivrBrandingProfiles node.

23. POST to ivrBrandingProfiles node the resourceLocation parameter and the url where the ivr branding files are held on your web server. Be sure that you set the request type to POST. Press Send to submit the data.

**Note:** If you have created subfolders under /Branding then you need to POST the url of the ivr subfolder to ivrBrandingProfiles.
A 200OK Response should be returned.

24. Change the request type back to GET, and press Send. The response should look like this:
25. Make a note of the ID number listed next to “ivrBrandingProfile id=”, in this example it is:
9d4fb901-28e8-43c3-95e5-58d1ccee781e

26. Assign this ivrBrandingProfile ID to the branding level required:
   - the system level (that is, as part of the global systems profile),
   - a per-ivr basis (as part of the ivr definition).
   - enter the ivrBrandingProfile field and from step 25

The example below applies the ivr branding to a previously created ivr (using POST to /ivrs followed by a GET to /ivrs to retrieve the ivr id):
   a. change the request type to PUT
   b. enter /ivrs/<ivr id> in this example the ivr id = 0e045363-1275-46ad-b131-f546541fdd11
c. enter the ivrBrandingProfile field and <ivrBrandingProfile id> from step 25

27. Press Send, and you should get another 200 OK Response.

28. To verify the data has been set correctly, simply change the request type to GET using the same URL as the previous PUT. Press Send.

29. Your IVR Branding files should now be used for calls to the specified ivr, make test calls to the ivr to check that the branded ivr files are applied.
7 Detailed customization procedure using RESTer

This section provides a step-by-step example of call branding using Microsoft Windows in a web server hosted branding scenario. It assumes that Microsoft Internet Information Services (IIS) Manager is installed with the latest patches on the web server that will host your call branding files.

7.1 Setting up the web server

1. On the web server create a directory structure under the default IIS Manager location `c:\inetpub\wwwroot` to hold the branding/custom files. The example below shows a Branding folder with four subfolders: Call_customization, Invitation_template, ivr_customization, webRTC_client.

   ![Directory structure](image)

2. Open IIS Manager and expand the Default Web Site. The newly created folders should show within the list.

   ![IIS Manager screenshot](image)
Note: If the newly created folders do not show within the list or your folders are outside of the standard IIS Manager location, you will need to create a new Virtual Directory. Within IIS, navigate to Default Web Site, right click and select Virtual Directory. Provide an Alias for this folder and browse for the physical path where the files will reside.

3. Place all the files into the folders, for the appropriate to the level of branding you intend to do.
   a. For WebRTC app customization these will be the zip files from Section 2.2.

   ![WebRTC_client_folder]

   Note: in the above screenshot the WebRTC_client customization files are zipped together by location.

   b. For ivr customization these will be the files from Section 3.1

   ![ivr_customization_folder]

   Note: the files for ivr customization are not zipped together.
c. For call customization these will be the files from Section 3.

<table>
<thead>
<tr>
<th>Name</th>
<th>Date modified</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>background</td>
<td>11/08/2015 15:01</td>
<td>JPEG image</td>
</tr>
<tr>
<td>call_join.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>call_join_confirmation.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>call_outgoing_welcome.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>cospace_join.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>cospace_join_confirmation.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>cospace_outgoing_welcome.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>disconnected.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>meetingEnded.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>onlyParticipant.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>passcode_entry.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>passcode_incorrect_final_attempt.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>passcode_incorrect_goodbye.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>passcode_incorrect_try_again.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>timeout.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>waiting_for_host.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
<tr>
<td>welcome.wav</td>
<td>11/08/2015 15:01</td>
<td>WAV File</td>
</tr>
</tbody>
</table>

Note: the files for call customization are not zipped together.

d. For the invitation template, this will be the file you created from Chapter 4.

4. Verify the web service is working. Open a browser and enter the full URL for the background image file used to customize the WebRTC app, including the file name background.jpg to see
if the file opens. For example: [pastacey: I've "removed" this step as Steve K says: "example is broken in that step 4 it's assuming you are still using the web branding via webadmin which is no longer available. These steps don't make sense as is because of that"

7.2 Using the API for branding

The remainder of this chapter shows an example using the RESTer API tool to brand the WebRTC app, IVR, SIP calls and apply custom invitations.

For more information on using the API, see the API Reference Guide.

7.2.1 Installing RESTer

1. Open Google Chrome as your browser and in the top left-hand corner of the browser window select the Apps icon. Or click: https://chrome.google.com/webstore/category/extensions in a Chrome browser window to skip to step 3.

2. Click on Web Store to open the Chrome App Store and select Extensions.
3. Enter RESTer in the Search field and press Enter, then locate the RESTer extension in the results and click Add to Chrome.

4. Click Add extension to continue with the installation.

5. Once installed, a confirmation dialog displays and the RESTer icon appears in the menu bar.
7.2.2 Add/Confirm SSL Exception

RESTer will follow the same SSL certificate verification and exception rules that your browser is configured for. If the URL for your webadmin site will not pass certificate validation or does not have an exception saved, RESTer will reject all requests. To check or add an exception:

1. In a Chrome window, open the URL to your Meeting Server webadmin interface, making sure you use HTTPS and the Port number (if listening on non-standard port). Example: https://cms.lab:445

2. If Chrome loads the Meeting Server web page without error you are OK to proceed.

or

3. If Chrome shows a Privacy Error, click Advanced and then click the link to proceed to the address.

7.2.3 Generate a Basic Auth header

Your requests will need to authenticate to the server. RESTer can generate and save an authorization header so this step does not have to be repeated for each command.

1. If not already open, click on the RESTer icon in the browser menu bar to open the tool.

2. Click the Authorization tab, then click Basic.
3. Enter an appropriate set of credentials for your server in the pop-up that displays, and click **Save**.

4. The tab will update to show the **Authorization** tab is now using the saved credential.

[is this red text section covered already by steps 1-3 above “Generate a basic Auth Header”?]
Open RESTer by clicking on the icon in the menu bar.

4. Enter the URL of the Meeting Server and press SEND. A 200 OK Response should be returned.

5. To verify API connectivity is working, perform a GET operation on the /system/status node of your server.
If it fails, verify that the login is correct and that you can access the Web URL from this PC.

Once connectivity is verified, in addition to WebRTC app branding, you can apply the branded/customized files to SIP calls (see Section 7.2.5), invitations (see Section 7.2.5) and IVRs (see Section 7.2.6) at a system level, per tenant level or per space orivr level.

### 7.2.4 Applying branding to WebRTC apps

Branding files for WebRTC apps are applied to a webBridge node. You can only configure one Web Bridge on a Meeting Server. If you need different WebRTC app branding for different tenants, then you need to set up a Web Bridge for each branded WebRTC app.

6. **POST** to the webBridges node the url parameter and the address for the Call Bridge to use to reach this Web Bridge under the Body tab. Click the 3 dots symbol and select Form from the
drop-down list. Press **Send** to submit the data.

**Note:** The request must include a header for the correct content type. To do this, click the Headers Tab and enter a new header as **Content-Type: application/x-www-form-urlencoded** – click on Name and Value to select from drop-downs. Alternatively, RESTer will detect what you are trying to do and offer to set the header for you for future requests. Click the highlighted warning banner **Set Content Type** and RESTer automatically adds the required Content Header to your request.

A **200 OK** Response should be returned.

7. Change the request type back to **GET**, and press **Send**. The response should look like this:
8. PUT to the webBridges node the resourceArchive parameter and the url where the WebRTC app branding files are held on your web server. A 200 OK Response should be returned.
9. To verify the data has been set correctly, simply change the request type to GET using the same URL as the previous PUT. Press Send. The response should look like this:

```
 Method | URL
---------|------------------

 HEADERS | BODY | AUTHORIZATION | VARIABLES
---------|------|---------------|-------------
 Name | resourceArchive | Value | https://192.0.2.23/Branding/webRTC

 Response | 200 OK
 Date: Wed, 08 May 2010 13:45:17 GMT
 Server: Apache
 Cache-Control: max-age=0, no-cache, no-store, must-revalidate
 Pragma: no-cache
 Content-Type: text/xml

<?xml version="1.0"?>
<webrtc><resourceArchive id="f98f5880-85eb-42a7-ba6d841706e"/>
<url>https://192.0.2.23/Branding/webRTC</url>
<resolveSyncConferenceId>false</resolveSyncConferenceId>
<isIdentityModeSecure>true</isIdentityModeSecure>
</webrtc>
```

Your WebRTC app branded files should now be used for WebRTC app calls to the Web Bridge, make a test call to check that the branded files are applied.
7.2.5 Applying branding to SIP calls and/or customized invitations

Branding files for SIP calls and customized invitation templates are applied to the callBrandingProfiles node.

17. POST to the callBrandingProfiles node the parameter appropriate to the type of branding (resourceLocation/invitationTemplate) and the urls where the branding files are held on your web server. Be sure to specify the full filename including the path to the file for the invitationTemplate URL only, for example: https://192.0.2.23/branding/invitation_template.txt. You can also use http. Set the request type to POST and press Send to submit the data.

**Note:** If you have created subfolders under /Branding then you need to POST the url of each subfolder to callBrandingProfiles.

<table>
<thead>
<tr>
<th>Method</th>
<th>URL</th>
<th>Headers</th>
<th>BODY</th>
<th>AUTHORIZATION</th>
<th>VARIABLES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>InvitationTemplate</td>
<td>InvitationTemplate</td>
<td>Text</td>
<td><a href="https://192.0.2.23/Branding/webRTC/invitation_template.txt">https://192.0.2.23/Branding/webRTC/invitation_template.txt</a></td>
</tr>
</tbody>
</table>

Response 200 OK

A 200 OK Response should be returned.

17. Change the request type back to GET and press Send. The response should look like this:
18. Make a note of the ID number listed next to “callBrandingProfile id=", in this example it is: 82ee5aa6-19d7-4e08-4923-ab7a20af252f

19. Assign this callBrandingProfile ID to the branding level required:
   - the system level (that is, as part of the global systems profile),
   - a per-tenant basis (as part of the tenant definition),
   - a per-space basis (as part of the space definition).

The example below applies the branding to a previously created tenant:
   a. change the request type to PUT
   b. enter the /tenant/<tenant id> in this example 0839b3ed-13ad-437f-b75e-976d20f62615
   c. enter the callBrandingProfile parameter and the id from step "Make a note of the ID number listed next to “callBrandingProfile id=", in this example it is: 82ee5aa6-19d7-4e08-4923-ab7a20af252f" above
d. press Send, and you should get another 200 OK Response

20. To verify the data has been set correctly, simply change the request type to GET using the same URL as the previous PUT. Press Send.
21. Your Call Branding files should now be used for calls for that tenant. Place a call from a SIP endpoint to a space setup for the tenant. Make sure the call displays the background picture, and that the voice prompt is audible.

7.2.6 Applying branding to an IVR

Branding files for IVR are applied to the ivrBrandingProfiles node. You can only configure one Web Bridge on a Meeting Server. If you need different IVR branding for different tenants, then you need to set up a Web Bridge for each branded IVR profile.

23. POST to ivrBrandingProfiles node the resourceLocation parameter and the url where the ivr branding files are held on your web server. Make sure the request type is set to POST. Press Send to submit the data.

**Note:** If you have created subfolders under /Branding then you need to POST the url of the ivr subfolder to ivrBrandingProfiles.

A 200 OK Response should be returned.

24. Change the request type back to GET, and press Send. The response should look like this:
25. Make a note of the ID number listed next to “ivrBrandingProfile id=”, in this example it is: 9d4fb901-28e8-43c3-95e5-58d1ccce781e

26. Assign this ivrBrandingProfile ID to the branding level required:
   - the system level (that is, as part of the global systems profile),
   - a per-ivr basis (as part of the ivr definition).
   - enter the ivrBrandingProfile field and from step 25

The example below applies the ivr branding to a previously created ivr (using POST to /ivrs followed by a GET to /ivrs to retrieve the ivr id):
   a. change the request type to PUT
   b. enter /ivrs/<ivr id> in this example the ivr id = 0e045363-1275-46ad-b131-f546541fdd11
c. enter the ivrBrandingProfile field and <ivrBrandingProfile id> from step 25

27. Press Send, and you should get another 200 OK Response.

28. To verify the data has been set correctly, simply change the request type to GET using the same URL as the previous PUT. Press Send.

29. Your IVR Branding files should now be used for calls to the specified ivr, make test calls to the ivr to check that the branded ivr files are applied.
8 Switching between branding methods

You can change from one method to another, however we recommend NOT mixing the two methods. For more information, refer to the steps below:

8.1 Changing from web server (or default) branding to locally hosted branding

If changing from web server (or default) branding to locally hosted branding, follow these recommendations:

- for every Call Bridge ensure:
  - the resourceLocation parameter for the IVR messages is not set; use a PUT method to set the resourceLocation parameter as blank on /ivrBrandingProfile/<ivr branding profile id>,
  - the resourceLocation parameter for the call messages is not set; use a PUT method to set the resourceLocation parameter as blank on /callBrandingProfile/<call branding profile id>,
  - the invitationTemplate parameter for the call messages is not set; use a PUT method to set the invitationTemplate parameter as blank on /callBrandingProfile/<call branding profile id>.

- for every Call Bridge which has a configured Web Bridge, make sure the resourceArchive field is not set in that configuration; use the PUT method to set the resourceArchive parameter as blank on API object /webBridges/<web bridge id>.

- stop using web server URLs in scripts and configurations of call leg profiles,
- configure call_branding.zip files on every Call Bridge, as described above,
- configure web_branding.zip files on every Web Bridge, as described above.

8.2 Changing from locally hosted to web server branding

If changing from locally hosted to web server branding, then follow these recommendations:

- remove locally hosted call_branding.zip files from every Call Bridge as described above,
- remove locally hosted web_branding.zip files from every Web Bridge as described above,
- change all your scripts and configurations to use http://mywebserver/... as documented in the Customization Guidelines.
8.3 Mixing locally hosted and web server customization

If you install branding zip files on your Meeting Servers, but also deploy a web server and use it to serve branding resource files, then note the following:

For IVR, call and invitation customization:

- customization using the web server will override the locally hosted files,
- leaving the API fields blank or unset will cause the locally hosted files to be used.

For WebRTC customization:

- customization using the web server will override the locally hosted files
- it is possible to configure the same Web Bridge in the configuration of more than one Call Bridge. In this case, a configured resource archive from a Call Bridge on another Meeting Server may override the locally hosted branding file for a Web Bridge. Because this might be unexpected, we recommend NOT mixing the two configurations.

8.4 Limitations of using locally hosted branding

- Only one background image file, background.jpg, is used in locally hosted branding, other image files will be ignored.
- If you want different image backgrounds in different situations, for example during pass code entry or IVR, the only way is to use a web server for customization as described in the Customization Guidelines.
- To use multiple sets of branding files, you still need to use an external web server.

Note: If files are too large, missing or otherwise invalid then they will be treated in the same way as their web server equivalents and will not be used. There will be no attempt to fall back to default resources. Any missing audio prompts are simply not played, and an invalid or omitted background.jpg file is replaced with a solid black background. Refer to the Customization Guidelines for file size limitations.
Appendix A  Using Other Popular API Tools

This appendix covers using Firefox Poster and the Chrome Advanced Rest Client, to create and apply a callBrandingProfile.

A.1 Using Firefox Poster

Follow steps 1 to 4 in Chapter 6 before proceeding with these steps.

Install Poster into Firefox (if not already installed) by opening the Menu button in the upper right corner and choosing Add-ons. If Poster is already installed go to step “Once enabled, access Poster through Tools > Poster” on the next page.

5. Enter Poster in the Search field and press Enter.
6. Click Install and follow directions to restart Firefox once complete.

7. Enable the Menu Bar by right clicking just to the right of the + sign that opens a new tab and selecting Menu Bar.

8. Once enabled, access Poster through Tools > Poster

9. To verify that API connectivity is working, enter the URL of the Meeting Server, and the Username and Password for the API. Press GET.

A 200OK Response should be returned with the Status Data, for example:
If the GET fails, verify that:

- the login is correct,
- you can access the Web URL from this PC
- Firefox trusts the certificate assigned to Web Admin, or make a Permanent exception by adding this Trust to the browser’s Trust store. Do this by opening a standard browser session to Web Admin using Firefox, and accept the warnings and add the exception.

A.1.1 Applying branding to SIP calls using Firefox Poster

This example only covers branding SIP calls at a system level using Firefox Poster. To configure the other types of branding, review Chapter 6 and adapt the steps to be suitable with Firefox Poster.

11. Once connectivity is verified, define the Call Branding Profile as below defining the URL where the Branding files are located.

   For example: resourceLocation=https<url of branding files>
12. Press POST to submit the data, you should get a 200OK Response like this:

13. Make a note of the Location field in this response and record the ID number shown, it is the long number after the /callBrandingProfiles/ text. In the example above, the ID is eea60a42-4435-4c1f-99b7-6999e884abf4.

14. Assign this ID to the Call Branding Profile to be used:
a. enter the /system/profiles URL

b. enter the id from step 13 into the callBrandingProfile. For example:
   callBrandingProfile= eea60a42-4435-4c1f-99b7-6999e884abf4

c. change the request type to PUT

You should receive another 200OK Response like this:

15. To verify the data has been set correctly, simply press GET using the same URL as the previous PUT and it should return the settings as such:
16. Your Branding files should now be used for calls. Place a call from a SIP endpoint to a space. Make sure the call displays the background picture, and that the voice prompt is audible.

This appendix covers using Firefox Poster and the Chrome Advanced Rest Client, to create and apply a callBrandingProfile.

A.2 Using Firefox Poster

Follow steps 1 to 4 in Chapter 6 before proceeding with these steps.

Install Poster into Firefox (if not already installed) by opening the Menu button in the upper right corner and choosing Add-ons. If Poster is already installed go to step “Once enabled, access Poster through Tools > Poster” on page 81.
5. Enter Poster in the Search field and press Enter.

6. Click Install and follow directions to restart Firefox once complete.

7. Enable the Menu Bar by right clicking just to the right of the + sign that opens a new tab and selecting Menu Bar.

8. Once enabled, access Poster through Tools > Poster
9. To verify that API connectivity is working, enter the URL of the Meeting Server, and the Username and Password for the API. Press GET.

A 200OK Response should be returned with the Status Data, for example:
If the GET fails, verify that:

- the login is correct,
- you can access the Web URL from this PC,
- Firefox trusts the certificate assigned to Web Admin, or make a Permanent exception by adding this Trust to the browser’s Trust store. Do this by opening a standard browser session to Web Admin using Firefox, and accept the warnings and add the exception.

A.2.1 Applying branding to SIP calls using Firefox Poster

This example only covers branding SIP calls at a system level using Firefox Poster. To configure the other types of branding, review Chapter 6 and adapt the steps to be suitable with Firefox Poster.

11. Once connectivity is verified, define the Call Branding Profile as below defining the URL where the Branding files are located.

For example: resourceLocation=https<url of branding files>
12. Press POST to submit the data, you should get a 200OK Response like this:

13. Make a note of the Location field in this response and record the ID number shown, it is the long number after the /callBrandingProfiles/ text. In the example above, the ID is eea60a42-4435-4c1f-99b7-6999e884abf4.

14. Assign this ID to the Call Branding Profile to be used:
a. enter the /system/profiles URL
b. enter the id from step 13 into the callBrandingProfile. For example:
callBrandingProfile= eea60a42-4435-4c1f-99b7-6999e884abf4
c. change the request type to PUT

You should receive another 200OK Response like this:

15. To verify the data has been set correctly, simply press GET using the same URL as the previous PUT and it should return the settings as such:
16. Your Branding files should now be used for calls. Place a call from a SIP endpoint to a space. Make sure the call displays the background picture, and that the voice prompt is audible.

A.3 Using Chrome Advanced Rest Client

Follow steps 1 to in Chapter 6 before proceeding with these steps.

5. Install the Advanced REST Client into Chrome (if not already installed) by opening the Apps button in the upper left corner. If Advanced REST Client is already installed go to step

6. Click on Web Store to open the Chrome App Store
7. Enter Advanced REST Client in the Search field and press Enter. Install Advanced REST Client by clicking on the + FREE button to the right.

8. Click Add to continue with the installation.

9. Once installed, the Advanced REST Client will appear on the Apps page.
10. Open the Advanced REST Client by clicking on the icon from within the Apps page of Chrome. To verify API connectivity is working, enter the URL of the Meeting Server and press SEND.

11. Enter the API login when prompted and press Log In.

A 200OK Response should be returned with the Status Data, for example:
If it fails, verify that the login is correct and that you can access the Web URL from this PC.

A.3.1 Applying branding to SIP calls using Chrome Advanced Rest Client

This example only covers branding SIP call

12. Once connectivity is verified, define the Call Branding Profile by specifying the URL where the Branding files are located. Make sure that you set the request type to POST. Press Send to submit the data.

A 200OK Response should be returned, for example:
13. Change the request type back to GET, and press Send. The response should look like this:

14. Make a note of the ID number listed next to “callBrandingProfile id=”, in this example it is 59ba0728-8650-4936-9cb4-2c818dfba821
15. Assign this ID to the Call Branding Profile to be used (see note below)
   a. enter the /system/profiles URL
   b. enter the id from step "Make a note of the ID number listed next to “callBrandingProfile id="; in this example it is 59ba728-8650-4936-9cb4-2c818dfba821" on the previous page into the callBrandingProfile field
   c. change the request type to PUT

You should receive another 200OK Response like this:

16. To verify it has been set correctly, simply change the request type to GET using the same URL as the previous PUT. Press Send.

17. That is it, your Branding files should now be used for calls. Place a call from a SIP endpoint to a space. Make sure the call displays the background picture, and that the voice prompt is audible.
Cisco Legal Information

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCurate BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB’s public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS” WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOSS PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

© 2016–2019 Cisco Systems, Inc. All rights reserved.
Cisco Trademark

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)