

Configure DTMF Sequences in Cisco Meeting Server Spaces

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Configure](#)

[1. Create a dtmfProfile object and apply it globally](#)

[2. Fine tune permissions to perform the DTMF actions](#)

[Allow Actions in a Specific Space Only](#)

[Allow Only Users With a Password to Perform the Actions](#)

[Related Information](#)

Introduction

This document describes the steps to configure DTMF sequences to allow users to perform actions on Cisco Meeting Server (CMS) spaces.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco Meeting Server
- DTMF

Components Used

The information in this document is based on these software and hardware versions:

- Cisco Meeting Server running software version 3.8

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

It is possible to define custom DTMF sequences in CMS for space participants to use to invoke actions that affect the space and its participants. For example, a DTMF sequence can be used by a participant to toggle their own audio mute, to start or stop streaming the meeting, or to end the meeting.

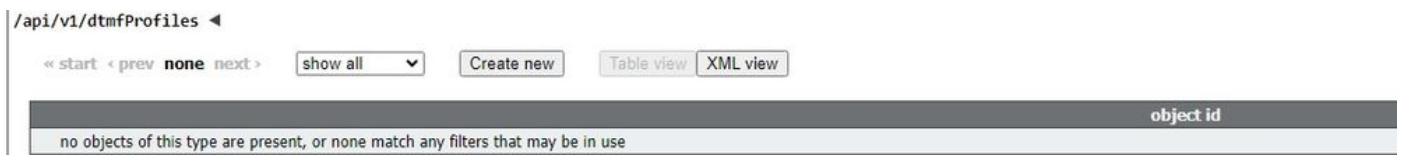
A **dtmfProfile** defines the DTMF sequences to be used to invoke each specific action available. By assigning a **dtmfProfile** to **system/profiles** via the CMS API, the DTMF sequences become available globally. Restrictions for the DTMF actions can be fine tuned by creating **callLegProfiles** with different permissions for each of the actions, and by applying them at different levels of the API.

Configure

1. Create a dtmfProfile object and apply it globally

To create the **dtmfProfile** object:

1. Log in to the CMS web admin page.
2. Navigate to **Configuration > API**.
3. Navigate to the **/api/v1/dtmfProfiles** section and expand it.
4. Click **Create new**.



5. Define the DTMF sequences to be used to perform the required action(s). In this example, DTMF sequences are defined for the **toggleMuteSelfAudio** and **endCall** actions.

/api/v1/dtmfProfiles

muteSelfAudio	<input type="checkbox"/>	<input type="text"/>
unmuteSelfAudio	<input type="checkbox"/>	<input type="text"/>
toggleMuteSelfAudio	<input checked="" type="checkbox"/>	11*
muteAllExceptSelfAudio	<input type="checkbox"/>	<input type="text"/>
unmuteAllExceptSelfAudio	<input type="checkbox"/>	<input type="text"/>
endCall	<input checked="" type="checkbox"/>	99*
nextLayout	<input type="checkbox"/>	<input type="text"/>
previousLayout	<input type="checkbox"/>	<input type="text"/>
lockCall	<input type="checkbox"/>	<input type="text"/>
unlockCall	<input type="checkbox"/>	<input type="text"/>
startRecording	<input type="checkbox"/>	<input type="text"/>
stopRecording	<input type="checkbox"/>	<input type="text"/>
startStreaming	<input type="checkbox"/>	<input type="text"/>
stopStreaming	<input type="checkbox"/>	<input type="text"/>
allowAllMuteSelf	<input type="checkbox"/>	<input type="text"/>
cancelAllowAllMuteSelf	<input type="checkbox"/>	<input type="text"/>
allowAllPresentationContribution	<input type="checkbox"/>	<input type="text"/>
cancelAllowAllPresentationContribution	<input type="checkbox"/>	<input type="text"/>
muteAllNewAudio	<input type="checkbox"/>	<input type="text"/>
unmuteAllNewAudio	<input type="checkbox"/>	<input type="text"/>
defaultMuteAllNewAudio	<input type="checkbox"/>	<input type="text"/>
muteAllNewAndAllExceptSelfAudio	<input type="checkbox"/>	<input type="text"/>
unmuteAllNewAndAllExceptSelfAudio	<input type="checkbox"/>	<input type="text"/>
getTotalParticipantCount	<input type="checkbox"/>	<input type="text"/>

6. Click **Create**.

To apply the **dtmfProfile** globally:

1. Navigate to **Configuration > API**.
2. Navigate to the **/api/v1/system/profiles** section and expand it.
3. Click **View or edit**.

/api/v1/system/profiles ◀

Object configuration

no data fields present for this object

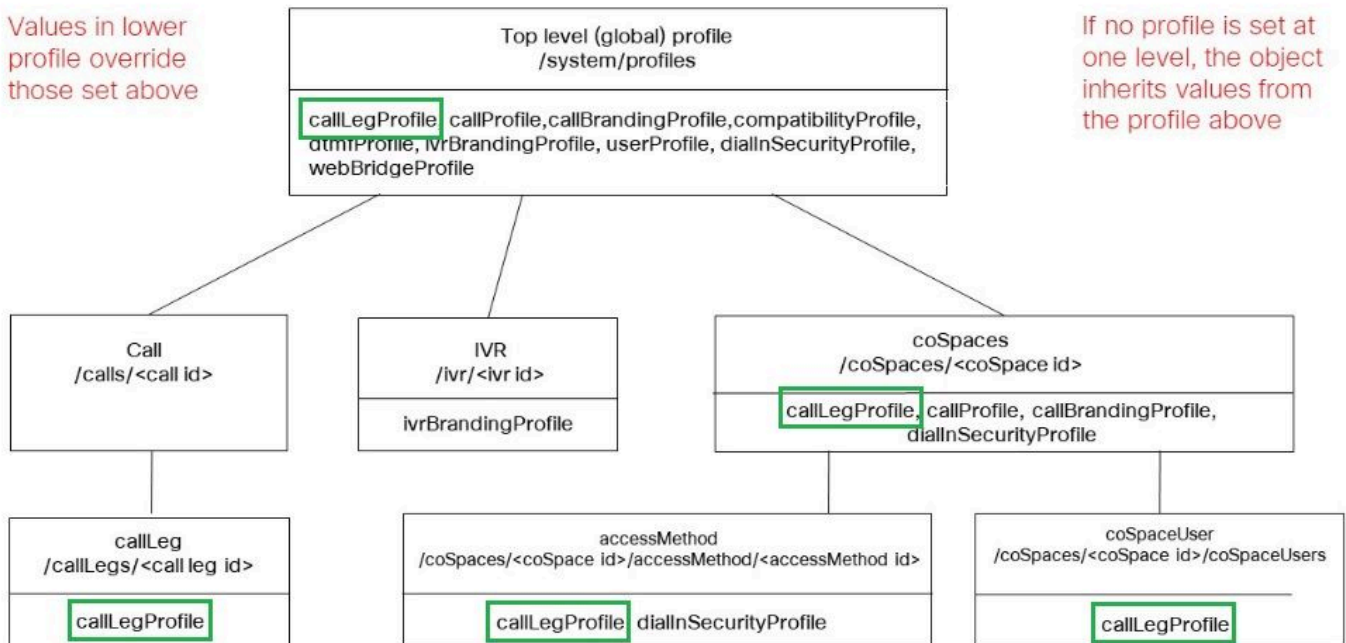
- On the **dtmfProfile** setting, click **Choose** and select the previously created object.
- Click **Modify**.

/api/v1/system/profiles

callLegProfile	<input type="checkbox"/>	<input type="text"/>	Choose
callProfile	<input type="checkbox"/>	<input type="text"/>	Choose
dtmfProfile	<input checked="" type="checkbox"/>	85e509af-bab5-4ff4-bbe1-94cb99ac80cf	Choose
userProfile	<input type="checkbox"/>	<input type="text"/>	Choose
ivrBrandingProfile	<input type="checkbox"/>	<input type="text"/>	Choose
callBrandingProfile	<input type="checkbox"/>	<input type="text"/>	Choose
compatibilityProfile	<input type="checkbox"/>	<input type="text"/>	Choose
dialInSecurityProfile	<input type="checkbox"/>	<input type="text"/>	Choose
lobbyProfile	<input type="checkbox"/>	<input type="text"/>	Choose
webBridgeProfile	<input type="checkbox"/>	<input type="text"/>	Choose

2. Fine tune permissions to perform the DTMF actions

Granular control of permissions to execute the DTMF action can be achieved by configuring **callLegProfiles**, which can be applied at different levels in the API. This diagram illustrates all possible levels:



The level at which the **callLegProfile** is applied determines its scope. Profiles assigned at lower levels override those set above. These examples illustrate how this principle can be utilized to allow invoking DTMF actions to only certain spaces or users.

Allow Actions in a Specific Space Only

A **callLegProfile** negating permissions to execute the DTMF actions can be created and applied at the **/system/profiles** level. Then, another **callLegProfile** allowing the actions can be created and applied at the **coSpace** level, therefore limiting the scope of the permissions to a specific **coSpace** (or a set of **coSpaces**, if applied to several).

1. Navigate to **Configuration > API**.
2. Navigate to the **/api/v1/callLegProfiles** section and expand it.
3. Click **Create new**.

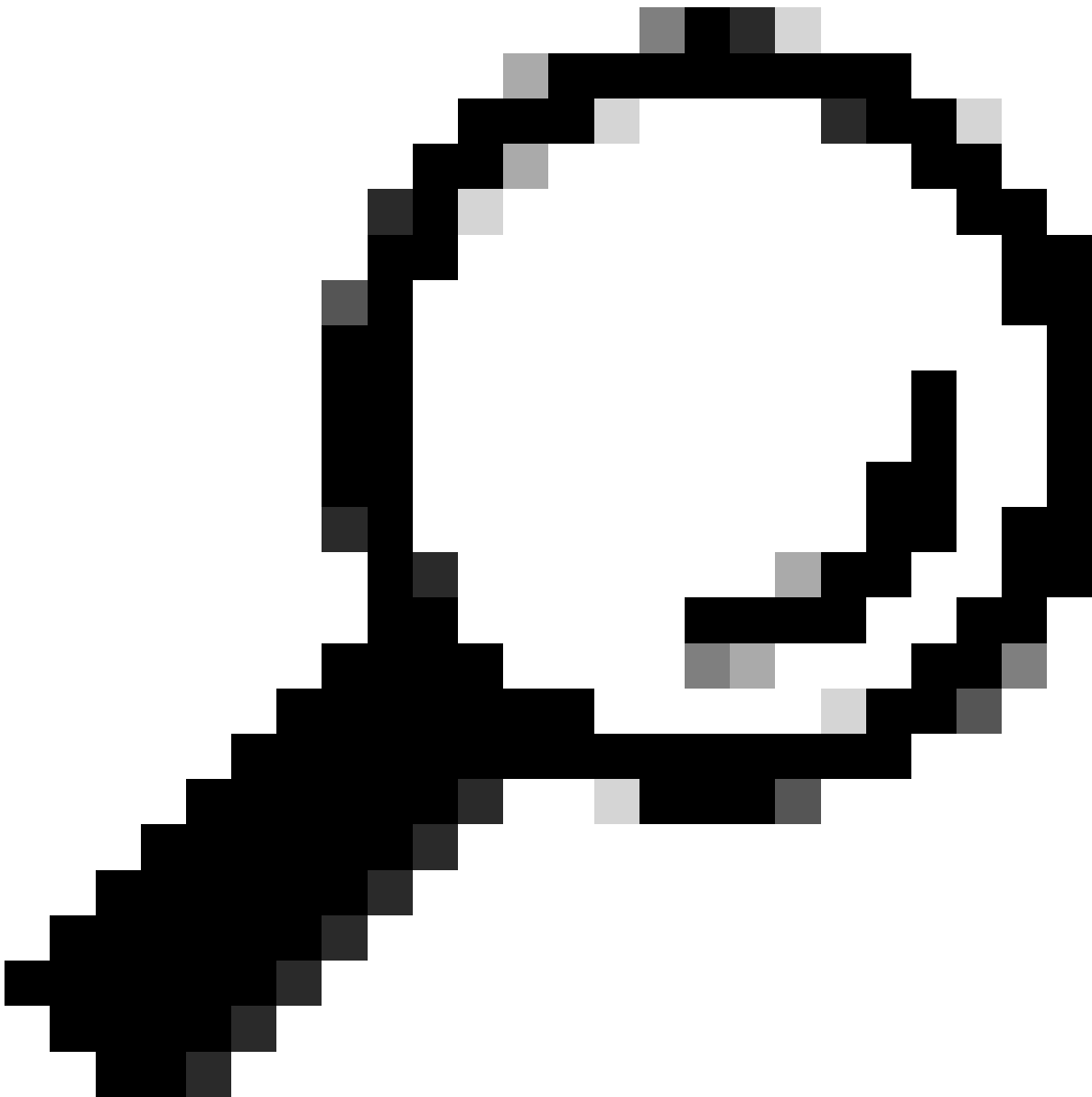


4. Navigate to the actions that need to be disallowed, and set them to false.
5. Click **Create**.

In this example a **callLegProfile** has been created to disallow ending the meeting by setting **endCallAllowed** to **false**:

/api/v1/callLegProfiles

needsActivation	<input type="checkbox"/>	<unset> ▼
defaultLayout	<input type="checkbox"/>	<unset> ▼
participantLabels	<input type="checkbox"/>	<unset> ▼
presentationDisplayMode	<input type="checkbox"/>	<unset> ▼
presentationContributionAllowed	<input type="checkbox"/>	<unset> ▼
presentationViewingAllowed	<input type="checkbox"/>	<unset> ▼
endCallAllowed	<input checked="" type="checkbox"/>	false ▼
disconnectOthersAllowed	<input type="checkbox"/>	<unset> ▼



Tip: If a suitable **callLegProfile** already exists, it can be modified instead of creating a new one.

6. Assign it to **system/profile** to globally disallow the action:

/api/v1/system/profiles

callLegProfile	<input checked="" type="checkbox"/>	<input type="text" value="e3aad583-491f-4c32-8d77-aa479388fc89"/>	Choose
callProfile	<input type="checkbox"/>	<input type="text"/>	Choose

7. Create a new **callLegProfile**, this time to allow the action. In this example a **callLegProfile** has been created with **endCallAllowed** set to **true**:

/api/v1/callLegProfiles

needsActivation	<input type="checkbox"/>	<unset> ▼
defaultLayout	<input type="checkbox"/>	<unset> ▼
participantLabels	<input type="checkbox"/>	<unset> ▼
presentationDisplayMode	<input type="checkbox"/>	<unset> ▼
presentationContributionAllowed	<input type="checkbox"/>	<unset> ▼
presentationViewingAllowed	<input type="checkbox"/>	<unset> ▼
endCallAllowed	<input checked="" type="checkbox"/>	true ▼
disconnectOthersAllowed	<input type="checkbox"/>	<unset> ▼

8. Navigate to **Configuration > API > /api/v1/coSpaces** and expand it.

9. Find the **coSpace** that you want to assign it to, and under **callLegProfile**, choose the one you created to allow the actions.

/api/v1/coSpaces/01d4b658-ccf3-48d2-b079-34c81c1e9791

userProvisionedCoSpace	<input type="checkbox"/>		GUID (none available)
name	<input type="checkbox"/>	Test space	- present
uri	<input type="checkbox"/>	2000	(URI user part) - present
secondaryUri	<input type="checkbox"/>		(URI user part)
callId	<input type="checkbox"/>	2000	- present
cdrTag	<input type="checkbox"/>		
passcode	<input type="checkbox"/>		
defaultLayout	<input type="checkbox"/>	<unset> ▼	
tenant	<input type="checkbox"/>		Choose
callLegProfile	<input checked="" type="checkbox"/>	ed1a5d12-796b-4bb4-b81e-f50812548d3b	Choose
callProfile	<input type="checkbox"/>		Choose

Since this **callLegProfile** is applied at a lower API level, it overrides the **callLegProfile** previously applied to **/system/profiles**, resulting in the DTMF actions being invocable from this particular coSpace only.

Allow Only Users With a Password to Perform the Actions

The scope of the permissions can be narrowed down to a group of users requiring special rights, like video operators. An **accessMethod** can be created, with its own directory number, to access a **coSpace** with a specific **callLegProfile** in effect that allows the DTMF actions.

1. Navigate to **Configuration > API > /api/v1/coSpaces** and expand it.

2. From the list, choose the **coSpace** you want to create the **accessMethod** for.

3. from the **Related objects** list, click the **/api/v1/coSpaces/<coSpace ID>/accessMethods** link:

Related objects: </api/v1/coSpaces>

</api/v1/coSpaces/56e3aa90-93e4-4189-9882-0aa02d60d59c/accessMethods>

</api/v1/coSpaces/56e3aa90-93e4-4189-9882-0aa02d60d59c/coSpaceUsers>

4. On the **uri** field, enter a number for privileged users to dial into this space.
5. Create a **passcode**. When privileged users dial the **accessMethod** number, they need to enter this code followed by the # sign to be allowed into the meeting.
6. Under **callLegProfile**, choose the one that allows ending the call.
7. Optionally, enter a **name** for the **accessMethod** to make it easily recognizable from within the API explorer.

In this example an **accessMethod** with number 3001 is created for video operators to use when joining the **coSpace** (the directory number 3000 has been assigned to the **coSpace**, regular users dial this number to join meetings on this space). It is protected by a password, and the **callLegProfile** that allows ending the call is assigned to it exclusively.

</api/v1/coSpaces/56e3aa90-93e4-4189-9882-0aa02d60d59c/accessMethods/a410d047-cd33-43d8-8db2-0c377d19351c>

uri	<input type="checkbox"/>	<input type="text" value="3001"/>	(URI user part) - present
callId	<input type="checkbox"/>	<input type="text" value="3001"/>	- present
passcode	<input type="checkbox"/>	<input type="text" value="1234"/>	- present
name	<input type="checkbox"/>	<input type="text" value="Video operator"/>	- present
callLegProfile	<input type="checkbox"/>	<input type="text" value="ed1a5d12-796b-4bb4-b81e-f50812548d3b"/> <input type="button" value="Choose"/>	- present
secret	<input type="checkbox"/>	<input type="text" value="djJv.8wVpDbteNCNDqskFQ"/>	- present

8. Apply the **callLegProfile** that disallows the actions is globally by assigning it to **system/profiles**.

In this example, there is no need to assign a **callLegProfile** to the **coSpace** itself. It inherits the globally applied profile, and therefore users that join the meeting by dialing the **coSpace** number (3000) do not have permission to end the call via DTMF.

</api/v1/coSpaces/56e3aa90-93e4-4189-9882-0aa02d60d59c>

userProvisionedCoSpace	<input type="checkbox"/>	<input type="text"/>	GUID (none available)
name	<input type="checkbox"/>	<input type="text" value="Test space 2"/>	- present
uri	<input type="checkbox"/>	<input type="text" value="3000"/>	(URI user part) - present
secondaryUri	<input type="checkbox"/>	<input type="text"/>	(URI user part)
callId	<input type="checkbox"/>	<input type="text" value="3000"/>	- present
cdrTag	<input type="checkbox"/>	<input type="text"/>	
passcode	<input type="checkbox"/>	<input type="text"/>	
defaultLayout	<input type="checkbox"/>	<input type="text" value="<unset>"/>	
tenant	<input type="checkbox"/>	<input type="text"/> <input type="button" value="Choose"/>	
callLegProfile	<input type="checkbox"/>	<input type="text"/> <input type="button" value="Choose"/>	
callProfile	<input type="checkbox"/>	<input type="text"/> <input type="button" value="Choose"/>	

As a result of this configuration, video operators can join the meeting by dialing the **accessMethod** number (3001) instead of the **coSpace** number (3000). Their call legs use the **callLegProfile** that allows the action to be applied, therefore only they can use the defined DTMF sequence to end the call.

Related Information

