

Guidance for Smooth Upgrade from Cisco Meeting Server 2.9 to 3.0 (and Onwards)

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Important information about upgrades](#)

[Summary of things to consider](#)

[Licenses](#)

[Webbridge \(WebRTC and CMA client\)](#)

[Web GUI Changes](#)

[Recorders / Streamers](#)

[Cisco Expressway Considerations](#)

[CMS Edge](#)

[CMS \(Acano\) X-Series](#)

[SIP Edge](#)

[Further Information](#)

[Licensing - Check licenses before upgrade](#)

[Determine how many users are assigned a PMP license once you upgrade](#)

[Do you have enough SMP licenses?](#)

[Configure CMM](#)

[Configure Webbridge \(WebRTC And CMA client\)](#)

[Web app user space creation permissions](#)

[Chat Function](#)

[WebRTC point to point calls](#)

[Notable webBridge settings changes](#)

[External Access section removed from Web GUI](#)

[Recording or Streaming](#)

[Recorder](#)

[Streamer](#)

[Expressway Consideration](#)

[CMS Edge](#)

Introduction

This document describes the challenges of upgrading a Cisco Meeting Server deployment running version 2.9 (or earlier) to 3.0 (or later) and how to handle those for a smooth upgrade process.

Features removed: XMPP was removed (which affects WebRTC), trunks/load balancers, webbridge

Features changed: Recorders and streamers are now SIP, and webbridge is replaced with webbridge3

This document only covers topics you need to consider before upgrading. It does not cover all of the new features available in 3.X.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- CMS administration
- CMS upgrades
- Certificate creation and signing

Everything mentioned here is outlined in various documents. It is always advisable to read the product release notes and refer to our programming guides and deployment guides if you need any further clarification on features: [CMS Installation and Configuration Guides](#) and [CMS Product Release notes](#) .

Components Used

The information in this document is based on Cisco Meeting Server.

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

This document is intended as guidance in case you already have a CMS 2.9.x (or earlier) deployment, regardless if single combined, or resilient, and when you plan to upgrade to CMS 3.0. The information in this document pertains to all models of CMS.

 **Note:** X-series cannot be upgraded to CMS 3.0. You need to plan to replace your X-series servers as soon as possible.

Important information about upgrades

The only supported method for upgrading CMS is a stepped upgrade. At the time of writing this, CMS 3.5 has been released. If you are on CMS 2.9, you must upgrade in a stepped fashion (2.9 --> 3.0 --> 3.1 --> 3.2 --> 3.3 --> 3.4 --> 3.5 (Note upgrade process has changes as of CMS 3.5, so read the release notes carefully!!)

If you do not perform a stepped upgrade, and are experiencing unusual behavior, TAC could request you downgrade and start over.

Also, as of CMS 3.4, CMS MUST use Smart Licensing. You can not upgrade to CMS 3.4 or newer and still use traditional licenses. Do not upgrade to CMS 3.4 or newer unless you have set up Smart Licensing.

Summary of things to consider

Use these questions to navigate to the sections that pertain to your own situation. Each consideration refers to a hyperlink to more detailed description presented in this document.

Licenses

Do you have enough Personal MultiParty (PMP) / Shared MultiParty (SMP) licenses on your servers before the upgrade?

In 3.0 the PMP licenses are allocated, even if the user is not signed in. For example, if you have imported 10000 users through LDAP, but you have only 100 PMP licenses, this puts you out of compliance as soon as you upgrade to 3.0. For this use case, make sure to certainly check for tenants who have userProfile set and/or system/profiles to see if a userProfile with hasLicense with a value of true is set.

How to check the userProfile on API and see if you have hasLicense=true set (meaning PMP licensed users), is covered up in more detail in [this section](#).

Do you have PMP/SMP licenses in your current cms.lic file?

Due to a license behavior changes in 3.0 onwards, you must confirm if you have enough PMP/SMP licenses before performing the upgrade. This is described in more detail in [this section](#).

Do you have Cisco Meeting Manager (CMM) deployed?

CMS 3.0 requires CMM 3.0 due to changes in how licenses are handled. It is recommended to deploy CMM 2.9 before you perform an upgrade of your environment to 3.0 as you can check your 90 day report for license consumption for the past 90 days. This is described in more detail in [this section](#).

Do you have Smart Licensing?

CMS 3.0 requires CMM 3.0 due to changes in how licenses are handled. So if you are using Smart Licensing through CMM already, ensure you have PMP and SMP licenses associated to your cluster.

Webbridge (WebRTC and CMA client)

Do you use WebRTC in CMS 2.9?

Webbridge has changed in CMS 3.0 significantly. For guidance on the migration from webbridge2 to webbridge3 and the use of web app, the information is found in [this section](#).

Do your users use the CMA thick client?

As these clients are XMPP based, these clients can not be used anymore after the upgrade as the XMPP server has been removed. If this applies for your use case, you can find more information in [this section](#).

Do you use Chat in WebRTC?

The chat functionality is removed from web app in 3.0. In CMS 3.2, chat is re-introduced, but it is not persistent. You can find more information on this feature in [this section](#).

Do your users perform Point to Point calls from WebRTC to devices?

In CMS 3.0, a web app user can not dial directly to another device anymore. Now you must join a meeting space, and have the permission to add participants to the meeting to perform on the same action. You can find more information on this part in [this section](#).

Do your users create their own coSpaces from WebRTC?

In 3.0, in order for web app users to be able to create their own spaces from the client, a coSpaceTemplate needs to be created in API and assigned to the user. This can be manual or automatic during LDAP import. CanCreateCoSpaces is removed from UserProfile. You can find more information on this feature in [this section](#).

Web GUI Changes

Do you have webBridge settings configured in the web admin GUI?

The webBridge settings are removed from the GUI in 3.0, so you must configure the webbridges in the API and note what your current settings are in the GUI so you can configure the webBridgeProfiles in the API accordingly. You can find more information on this change in [this section](#).

Do you have External Settings configured in the web admin GUI?

The External Settings have been removed from the GUI in CMS 3.1. If you have either Webbridge URL or IVR configured in your CMS 3.0 or older web admin GUI (Configuration --> General --> External Settings), these have been removed from the web page and now need to be configured in the API. The previous settings prior to upgrading to 3.1 do NOT get added into API, and must be done manually. You can find more information on this change in [this section](#).

Recorders / Streamers

Do you currently use any CMS recorder(s) and/or streamer(s)?

The CMS recorder and streamer component are now SIP based instead of XMPP based. Therefore as the XMPP is being removed, these need to be tweaked after the upgrade. You can find more information on this change in [this section](#).

Cisco Expressway Considerations

What is your current Cisco Expressway version if you are using Expressway to proxy WebRTC?

CMS 3.0 requires Expressway 12.6 or newer. You can find more information on this WebRTC proxy feature in [this section](#).

CMS Edge

Do you currently have a CMS Edge in your environment?

CMS Edge is re-introduced on CMS 3.1 with higher scalability for external connections. You can find more information on this part in [this section](#).

CMS (Acano) X-Series

Do you currently have x-series servers in your environment?

These server can not be upgraded to CMS 3.0 and you must be looking at replacing these soon (move to a virtual machine or CMS appliance before upgrading to 3.0). You can find the end of life notice about these servers in [this link](#).

SIP Edge

Do you currently use SIP Edge in your environment?

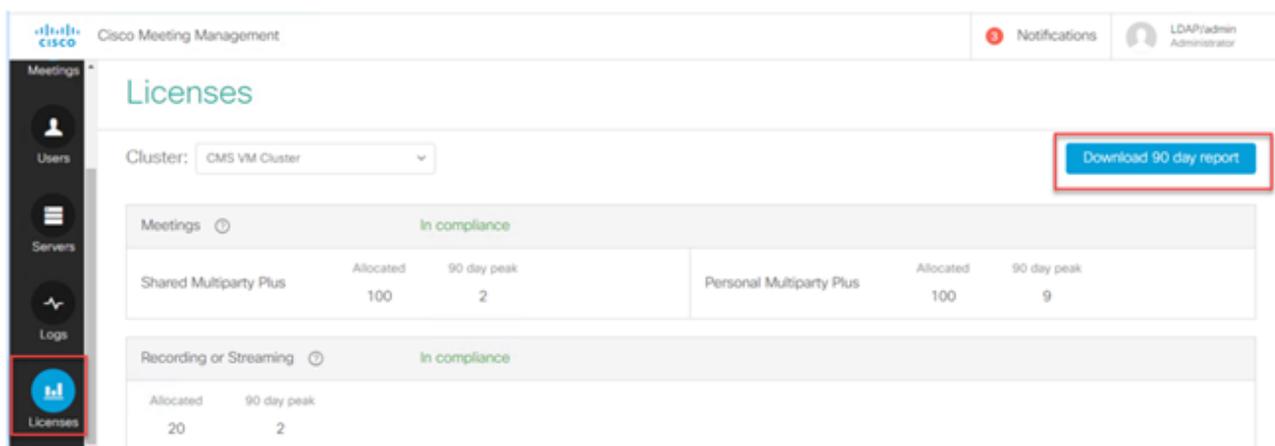
Sip Edge is fully deprecated as of CMS 3.0. You need to use Cisco Expressway to bring SIP calls into your CMS. Please contact your Cisco Account representative on how to get Expressways for your organization.

Further Information

Licensing - Check licenses before upgrade

Out of compliance license status is the most impactful issue when you upgrade to 3.0 or higher from a 2.x version. This section describes how to determine the amount of PMP/SMP licenses you need for a smooth upgrade.

Before you upgrade your deployment to 3.0, deploy CMM 2.9 and check the **90 day report** under **Licenses** tab to see if the license usage stayed under your current allocated license amount on the CMS nodes:



The screenshot displays the Cisco Meeting Management interface for the 'Licenses' tab. The cluster is set to 'CMS VM Cluster'. A 'Download 90 day report' button is highlighted with a red box. The 'Meetings' section is 'In compliance' and shows a table with columns for 'Shared Multiparty Plus' and 'Personal Multiparty Plus', each with 'Allocated' and '90 day peak' values. The 'Recording or Streaming' section is also 'In compliance' and shows a table with 'Allocated' and '90 day peak' values.

Shared Multiparty Plus		Personal Multiparty Plus	
Allocated	90 day peak	Allocated	90 day peak
100	2	100	9

Allocated	90 day peak
20	2

If you use Traditional licensing (cms.lic file is installed locally on your CMS nodes), check the CMS license file for the quantities of personal and shared licenses (100 / 100 as per the image here) on each of the CMS nodes (download through WinSCP from each callBridge node).

```

],
"issued_to": "Darren McKinnon - TAC",
"notes": "Darren McKinnon - TAC",
"features":
{
  "callbridge":
  {
    "expiry": "2100-Jan-03"
  },
  "webbridge3":
  {
    "expiry": "2100-Jan-03"
  },
  "customizations":
  {
    "expiry": "2100-Jan-03"
  },
  "recording":
  {
    "expiry": "2100-Jan-03",
    "limit": "10"
  },
  "personal":
  {
    "expiry": "2100-Jan-03",
    "limit": "100"
  },
  "shared":
  {
    "expiry": "2100-Jan-03",
    "limit": "100"
  },
  "streaming":
  {
    "expiry": "2100-Jan-03",
    "limit": "10"
  }
}

```

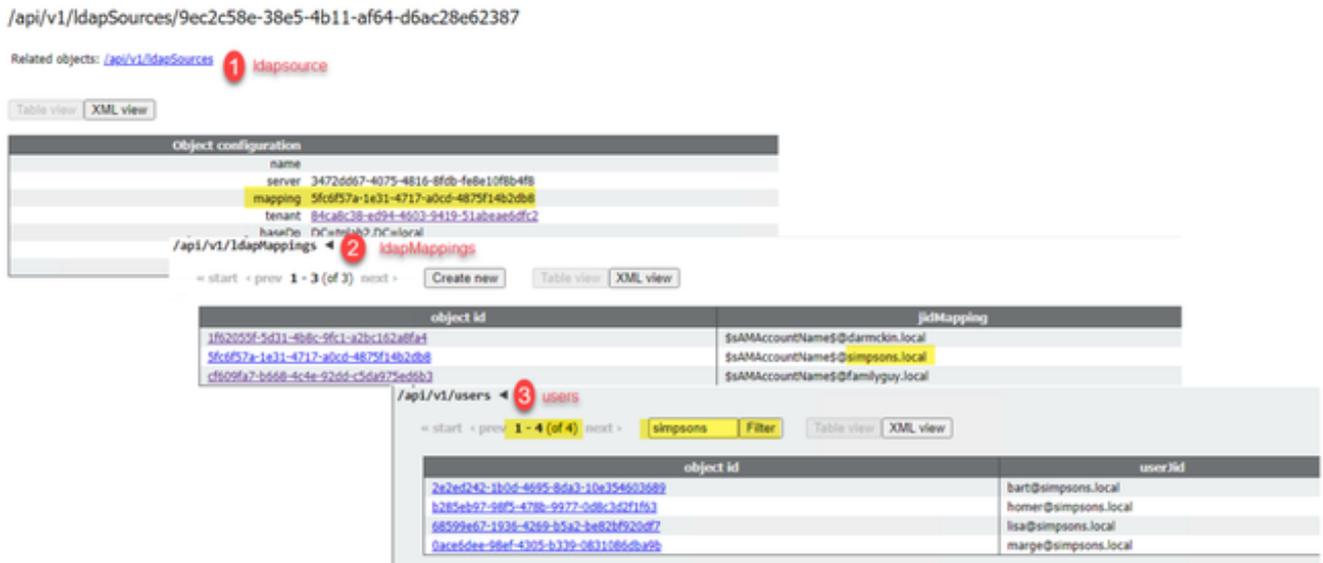
 : This can be more complex in other situations in which case you need to check this with the ActiveDirectory mappings and filters you created.

Step 1. Find the mapping ID from the ldapSource.

Step 2. Look up ldapMappings to find jidMapping.

Step 3. Search in api/v1/users for the domain used in the jidMapping.

Step 4. Add up the users found from each ldapSource. This is how many LDAP imported users need PMP licenses.



The screenshot shows a web interface for managing LDAP sources and mappings. The path is `/api/v1/ldapSources/9ec2c58e-38e5-4b11-af64-d6ac28e62387`. It shows a list of related objects, including `ldapSource` (1) and `ldapMappings` (2). The `ldapMappings` table shows a mapping ID `5f6f57a-1e31-4717-a0cd-4875f14b2db8` associated with the domain `simpsons.local`. The `users` table shows a list of users with their `object id` and `user/id` attributes, filtered by the domain `simpsons`.

2. System/Profiles

If a userProfile is set at the system/profiles level, and that userProfile has "hasLicense=true" then any user imported into CMS is be assigned a PMP license when the server is upgraded. If you imported 10,000 users but you only have 100 PMPs, this results in you being out of compliance when you upgrade to CMS 3.0, and can cause a 30 second on screen message to appear and audio prompt at the start of calls.

If the userProfile at the system level indicates users are to get a PMP, go to `api/v1/users` to see how many users there are in total:



The screenshot shows the `api/v1/users` endpoint with a filter set to `simpsons`. The table displays a list of users with their `object id` and `user/id` attributes. A red arrow points to the text "Will show total number of imported users" above the table.

If you had previously imported all users from your ldap, but realize now that you only need a certain subset from from that list, create a better filter in your ldapSource so it imports just the users you want to be assigned PMP licenses. Revise your filter on ldapSource and then perform a new LDAP sync in `api/v1/ldapsync`. This results in only your desired users being imported, and all others from this previous import removed.

 **Note:** If you do this correctly and the new import only removes unwanted users, remaining users coSpace CallIDs and secrets do not change, but if you do make a mistake, this can result in all callIDs and secrets changing. Make a backup of your database nodes before attempting this if you are concerned about this!

Do you have enough SMP licenses?

When you were looking at your daily peaks from the CMM 90 Day Report, do you have enough SMP licenses already to cover what your peak was? SMP licenses are used when the owner of the meeting has not been assigned a PMP license (either as coSpace owner / ad-hoc meeting / TMS scheduled meeting). If you are intentionally using SMP and have enough to cover your peak times, then this is all OK. If you check the 90 day peak for SMP and it is unclear why these are consumed, here are some things to check.

1. Ad hoc calls (as escalated from CUCM) use an SMP license if the device used to merge is not associated to a user who has been assigned a PMP license in CMS through the userProfile. CUCM provides the GUID of the user escalating the meeting. If that GUID corresponds to a Meeting Server imported LDAP user with an assigned PMP license, the license of that user is be used.
2. If a coSpace owner has not been assigned a PMP license, calls to those particular coSpaces use a SMP license.
3. If the meeting was scheduled in TMS version 15.6 or newer, the meeting owner is sent to CMS and if that user was not assigned a PMP license, that meeting uses a SMP license.

Configure CMM

As from CMS 3.0, CMM 3.0 is required for CMS to function properly. CMM is responsible for the licensing of CMS, so if you plan to upgrade CMS to 3.0, you must have a CMM server. It is recommended to deploy CMM 2.9 while you are on CMS 2.9 so you can check your license consumption before upgrading.

CMM checks all added callBridges for SMP and PMP licenses and the callBridge license. It uses the number that is the highest accross the various devices within the cluster.

For example, if CMS1 has 20 PMP and 10 SMP licenses, and CMS2 has 40 PMP and 5 SMP licenses in traditional licensing, the CMM reports that you have 40 PMP and 10 SMP licenses to use.

If you have more PMP licenses than imported users, you do not have any issues related to PMP (or SMP) licenses but if you check that 90 day peak and you find that you used more than available, you can still upgrade to CMS 3.0 and use the 90 day Trial License on CMM to sort things out with your licensing, or take action before the upgrade.

Configure Webbridge (WebRTC And CMA client)

CMS 3.0 removes the XMPP server component, and with that, removes webBridge and the ability to use the CMA thick client. WebBridge3 is what is used now to connect web app users (formerly referred to as WebRTC users) to meetings using the browser. When you upgrade to 3.0, you need to configure webbridge3.

 **Note:** CMA thick client does not work after upgrading to CMS 3.0!

This video does walk you through the process on how to create the webbridge 3 certificates.

<https://video.cisco.com/detail/video/6232772471001?autoStart=true&q=cms>

Prior to the upgrade to 3.0, customers must plan on how to configure Webbridge3. The most important steps are highlighted here.

1. You do need a key and cert chain for webbridge3. The old webbridge cert can be used if the cert contains all CMS server FQDNs or IP addresses as Subject Alternative Name (SAN)/ Common Name (CN) that are running webbridge3, and if either of the these are met:
 - a. Certificate has no Enhanced Key Usage (meaning it can be used either as Client or server).
 - b. Certificate has both Client and Server Authentication. HTTPs cert only really needs Server Authentication, while C2W certificate requires both server and client).
2. If you want to create a new certificate for the "**webbridge3 https**" cert, it is recommended to be publicly signed (to avoid certificate warnings on client when using web app). This same cert can be used for the "webbridge3 c2w cert", and the cert must have the FQDN of the webbridge servers in the SAN/CN.
3. CallBridges need to communicate with the new webbridge3 using a port that is configured in **webbridge3 c2w listen** command. This can be any available port, such as 449. Users need to be sure the callbridges can talk to webbridge3 on this port, and have any firewall changes made in advance, if necessary. It cannot be the same port used by "webbridge https" to listen on.

Before the CMS upgrade to 3.0, it is recommended to take a backup using 'backup snapshot <servername_date>', and then log into the webadmin page on your callbridge nodes to remove all XMPP Settings and Webbridge Settings. Then connect to the MMP on your servers, and perform these steps on all Core servers that have xmpp and webbridge over a SSH connection:

1. **xmpp disable**
2. **xmpp reset**
3. **xmpp certs none**
4. **xmpp domain none**
5. **webbridge disable**
6. **webbridge listen none**
7. **webbridge certs none**
8. **webbridge trust none**

Once you upgrade to 3.0, begin by configuring webbridge3 on all servers that previously ran webbridge. You must do this because there are already DNS records out there that point to these servers, so in that way you ensure that if a user gets sent to a webbridge3, it is prepared to handle the request.

Webbridge3 Configuration (all over SSH connection)

Step 1. Configure webbridge3 http listening port.

Webbridge3 https listen a:443

Step 2. Configure certificates for webbridge3 for browser connections. This is the certificate sent to browsers and needs to be signed by a public Certificate Authority (CA) and containing the FQDN used in the browser for the browser to trust the connection.

Webbridge3 https certs wb3.key wb3trust.cer (This needs to be a trust chain: make a trust cert that has end entity on top, followed by Intermediate CAs in order, finishing with RootCA).

```

-----BEGIN CERTIFICATE-----
Entity cert ← wb3/cb cert
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Intermediate cert
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
root cert
-----END CERTIFICATE-----
single carriage return at end

```

Step 3. Configure port to use to listen for callBridge to webbridge (c2w) connections. Since 443 is used for the webbridge3 https listen port, this config must be a different, available port like for example 449.

Webbridge3 c2w listen a:449

4. Configure certificates that webbridge sends to callbridge for the c2w trust

Webbridge3 c2w certs wb3.key wb3trust.cer

5. Configure the trust store WB3 uses to trust the callBridge certificate. This needs to be the same as the certificate used on the callbridge CA bundle (and must be a bundle of intermediate certificates on top, and Root CA at the end, followed by a single carriage return).

Webbridge3 c2w trust rootca.cer

6. Enable webbridge3

Webbridge3 enable

```
Usage:
webbridge3
webbridge3 restart
6 webbridge3 enable
webbridge3 disable
1 webbridge3 https listen <interface:port whitelist>
2 webbridge3 https certs <key-file> <crt-fullchain-file>
webbridge3 https certs none
webbridge3 http-redirect (enable [port]|disable)
3 webbridge3 c2w listen <interface:port whitelist>
4 webbridge3 c2w certs <key-file> <crt-fullchain-file>
webbridge3 c2w certs none
5 webbridge3 c2w trust <crt-bundle>
webbridge3 c2w trust none
webbridge3 options <space-separated options>
webbridge3 options none
webbridge3 status
```

CallBridge config changes (all over SSH connection)

Step 1. Configure the callBridge trust with the CA cert/bundle that signed the webbridge3 c2w certificate.

Callbridge trust c2w rootca.cer

Step 2. Restart the callBridge to get the new trust to take effect. This drops all calls on this particular callBridge so use this with caution.

Callbridge restart

API configuration for callBridges to connect to webBridge3

1. Create a new webBridge object using POST in the API and give it a URL value using FQDN and port configured on webbridge c2w interface white list (step 3 in the webbridge3 config)

c2w://webbridge.darmckin.local:449

At this point, Webbridge3 works again, and you can join spaces as guest or if you have previously imported users, they must be able to sign in.

Web app user space creation permissions

Are your users used to being able to create their own spaces in WebRTC? As of CMS 3.0, web app users cannot create their own coSpaces unless they have a cospace template assigned to them allowing for this.

Even with a coSpaceTemplate assigned, this does not create a space that others can dial into (no URI, no

Call ID or passcode), but if the coSpace has a callLegProfile with 'addParticipantAllowed', then they can dial out from the space.

In order to have dial strings that can be used to call into the new space, the coSpaceTemplate must have an accessMethodTemplate setup (see 2.9 release notes - https://www.cisco.com/c/dam/en/us/td/docs/conferencing/ciscoMeetingServer/Release_Notes/Version-2-9/Cisco-Meeting-Server-Release-Notes-2-9-6.pdf).

In the API, create coSpaceTemplate(s) and then create an accessMethodTemplate(s) and assign the coSpaceTemplate to the ldapUserCoSpaceTemplateSources or you can manually assign a coSpaceTemplate to a user in api/v1/users.

You can create and assign multiple CoSpaceTemplates and accessMethodsTemplates. See the CMS API guide for more information (<https://www.cisco.com/c/en/us/support/conferencing/meeting-server/products-programming-reference-guides-list.html>)

The screenshot displays the API configuration for a CoSpaceTemplate and its associated AccessMethodTemplate. The top section shows the CoSpaceTemplate configuration with the following details:

Object configuration	
name	First CoSpaceTemplate
callProfile	008e1aa7-0079-4d65-b6ae-fb218bd2e6b4
callLegProfile	ef583b0e-a6fe-49cf-bece-b557332a76bf
numAccessMethodTemplates	2

The bottom section shows the configuration for an AccessMethodTemplate, which is linked to the CoSpaceTemplate above. The configuration includes:

- name: First CoSpaceTemplate (present)
- description: (empty)
- callProfile: 008e1aa7-0079-4d65-b6ae-fb218bd2e6b4 (Choose - present)
- callLegProfile: ef583b0e-a6fe-49cf-bece-b557332a76bf (Choose - present)
- dialInSecurityProfile: (empty) (Choose)

A red arrow points from the URL of the AccessMethodTemplate in the top section to the configuration form in the bottom section.

CoSpaceTemplate (API config)

Name: Any name you want to give the coSpaceTemplate.

Description: Brief Description if desired.

callProfile: White callProfile do you want any Spaces created with this template to use? If not provided, it uses what is set at the system/profile level.

callegPrprofile: Which calllegProfile do you want any spaces created with this template to use? If not provided, it uses what is set at the system/profile level.

dialInSecurityProfile: Which dialInSecurityProfile do you want any spaces created with this template to use? If not provided, it uses what is set at the system/profile level.

AccessMethodTemplate (API config)

Name: Any name you want to give the coSpaceTemplate.

uriGenerator: The expression to be used to generate URI values for this access method template; the allowed set of characters are 'a' to 'z', 'A' to 'Z', '0' to '9', '.', '-', '_' and '\$'; if non empty it must contain exactly one '\$' character. Example of this is \$.space wich uses the name provided by user when creating the space and append ".space" to it. "Team Meeting" creates the url 'Team.Meeting.space@domain'.

callLegProfile: Which calllegProfile do you want any accessMethods created with this template to use? If not provided, it uses what is set CoSpaceTemplate level and if none there, use what is at the system/profile level.

generateUniqueCallId: Whether to generate a unique numeric ID for this access method which overrides the global one for the cospace.

dialInSecurityProfile: Which dialInSecurityProfile do you want any accessmethods created with this template to use? If not provided, it uses what is set CoSpaceTemplate level and if none there, use what is at the system/profile level.

Chat Function

CMS 3.0 removed the persistent chat function, but in CMS 3.2 the non-persistent chat within spaces returned. Chat is available to web app users and is not stored anywhere. Once CMS 3.2 is installed, web app users are by default able to message eachother during meetings. These messages are only available during the meeting, and only messages exchanged after joining are seen. You can not join late and scroll back to see previous messages.

WebRTC point to point calls

On CMS 2.9.x, WebRTC participants were able to dial from their client directly to other contacts. Starting in CMS 3.0, this is no longer possible. Now users must sign in and join a space. From there, if they have permission in the callLegProfile (**addParticipants** parameter set to True), they are able to add other contacts. This causes CMS to dial out to the participant and they meet on a space in CMS.

Notable webBridge settings changes

CMS 3.0 and 3.1 has removed or relocated some of the webbridge settings from the GUI and they need to be configured in the API to keep the consistent experience for users. On 3.x, use **api/v1/webBridges** and **api/v1/webBridgeProfiles**.

Check what you currently have configured so when you upgrade to 3.0, you can configure the webbridge and webbridge Profiles in API accordingly.

The image displays three screenshots of the CMS GUI, illustrating the changes in settings across different versions:

- CMS 2.9.x:** Shows 'Web bridge settings' (highlighted with a red box) including fields for 'Guest account client URI', 'Guest account JID domain' (tp1ab2.local), 'Guest access via ID and passcode' (secure: require passcode to be supplied with ID), 'Guest access via hyperlinks' (allowed), 'User sign in' (allowed), and 'Joining scheduled Lync conferences by ID' (not allowed). Below this is an 'IVR' section with 'IVR numeric ID' (7772) and 'Joining scheduled Lync conferences by ID' (not allowed). A red box highlights the 'External access' section with 'Web Bridge URI' (https://14.49.25.94) and 'IVR telephone number'. A 'Submit' button is at the bottom.
- CMS 3.0:** Shows 'Lync Edge settings' with 'Server address', 'Username', and 'Number of registrations' fields. Below is an 'IVR' section with 'IVR numeric ID' (7772) and 'Joining scheduled Lync conferences by ID' (not allowed). A red box highlights the 'External access' section with 'Web Bridge URI' (https://14.49.25.94) and 'IVR telephone number'. A 'Submit' button is at the bottom.
- CMS 3.1:** Shows 'Lync Edge settings' with 'Server address', 'Username', and 'Number of registrations' fields. Below is an 'IVR' section with 'IVR numeric ID' (7772) and 'Joining scheduled Lync conferences by ID' (not allowed). A 'Submit' button is at the bottom.

In 3.0, **Web bridge settings** were removed on the GUI, then in CMS 3.1, the **External access** fields have been removed as well.

Web bridge settings in GUI

- **Guest account client URI** - this was used by the callBridge to find the webBridge. If you had multiple webBridges in your deployment for WebRTC, this field must already be blank, and you must have unique URLs in api/v1/webbridges for each webBridge that the callBridge needs to connect to. Delete anything in this field and make sure you have the webBridges configured in the API.
- **Guest Account Jid Domain** - this is not used anymore in CMS 3.0 and you can delete this.
- **Guest Access Via ID and Passcode** - removed and not replaced in CMS 3.0.
- **Guest Access Via Hyper Links** - now configurable under webBridgeProfiles in API in setting "AllowSecrets".

The image shows two versions of the API form for creating web bridges. The top version, labeled 'CMS 2.9.x', includes fields for 'url', 'resourceArchive', 'tenant', 'tenantGroup', 'idEntryMode', 'allowWeblinkAccess', 'showSignIn', 'resolveCoSpaceCallIds', 'resolveLyncConferenceIds', 'callBridge', and 'callBridgeGroup'. The bottom version, labeled 'CMS 3.0', shows a simplified form with 'url', 'tenant', 'tenantGroup', 'callBridge', 'callBridgeGroup', and 'webBridgeProfile'. Both forms have a 'Create' button at the bottom.

Notice in CMS 3.0, several fields have been removed from /api/v1/webBridges.

- **resourceArchive** - now in webbridgeProfiles.
- **idEntryMode** - now deprecated.
- **allowWeblinkAccess** - now in webBridgeProfiles as allowSecrets.
- **showSignIn** - now in webBridgeProfiles as userPortalEnabled.
- **resolveCoSpaceCallIds**- now in webbridgeProfiles.
- **resolveLyncConferenceIDs** - now in webbridgeProfiles.

The image shows the API form for creating web bridge profiles, labeled 'CMS 3.0 onward'. It includes fields for 'name', 'resourceArchive', 'allowPasscodes', 'allowSecrets', 'userPortalEnabled', 'allowUnauthenticatedGuests', 'resolveCoSpaceCallIds', and 'resolveCoSpaceUris'. A 'Create' button is located at the bottom.

WebBridgeProfile

- **resourceArchive** - if you use custom backgrounds and your resource archive is stored on a web server, enter the URL here.
- **allowPasscodes** - if false, users do not have an option to join meetings as guests. They can only sign in or use a URL containing the space info and secret
- **allowSecrets** - If this is set to false, users cannot join spaces using a URL such as https://meet.company.com/meeting/040478?secret=gPDnucF8is4W1cS87_1.zw. Users need to use <https://meet.company.com> and enter the Call ID/Meeting ID/URI and PIN/Passcode if one is configured.

- **userPortalEnabled** - if this is set to false, the web app portal landing page does not show the sign in option. It only displays the fields for entering the Call ID/Meeting ID/URI and PIN/Passcode if one is configured.
- **allowUnauthenticatedGuests** - if set to False, guests cannot join any meetings - even with the full URL that contains the meeting ID and secret. When False, only users who can sign in can join a meetings. Example. User2 is trying to use the URL for User1's meeting. After entering the URL, User2 must sign in to continue to User1's meeting.
- **resolveCoSpaceCallIds** - if set to False, guests can only join meetings by entering the URI and PIN/Passcode if used. Call ID/Meeting ID/Numeric ID are not accepted.
- **resolveCoSpaceUris** - 3 possible settings: off, domainSuggestionDisabled, and domainSuggestionEnabled. Whether or not this webBridge accepts coSpace and coSpace accessMethod SIP URIs for the purpose of allowing visitors to join cospace meetings.

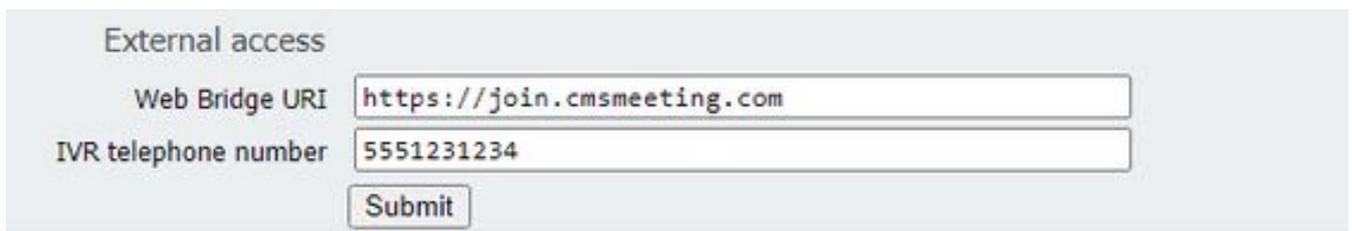
- When set to 'off' join by URI is disabled.

- When set to 'domainSuggestionDisabled' join by URI is enabled but the domain of the URI is not autocompleted or verified on webBridges using this webBridgeProfile.

- When set to 'domainSuggestionEnabled' join by URI is enabled and the domain of the URI can be autocompleted and verified on webBridges using this webBridgeProfile.

External Access section removed from Web GUI

In CMS 3.1, the External Access section has been removed from the web GUI. If you had these configured prior to the upgrade, then you need to reconfigure them in the API under webbridgeProfiles.



First, you need to create a webbridgeProfile as described in the previous section. Once you have created a webbridgeProfile, then you can create a IVR Number and/or Web Bridge URI via the links available in API under the newly created webBridgeProfile.



You can create up to 32 IVR numbers or 32 webbridgeAddresses per webBridgeProfile

Recording or Streaming

The recorder and streamer component on CMS 2.9.x and earlier were XMPP clients, and from CMS 3.0, they are SIP based. This now permits layouts for recordings and streaming to be changed using the default

layout in API. Also, now name labels are shown in the recording/streaming session. See CMS 3.0 release notes for more information about the recorder/streaming features -

https://www.cisco.com/c/dam/en/us/td/docs/conferencing/ciscoMeetingServer/Release_Notes/Version-3-0/Cisco-Meeting-Server-Release-Notes-3-0.pdf.

If you have recorder or streamer configured in 2.9.x, you need to reconfigure the settings in MMP and API so that these continue to work after the upgrade.

Before the CMS upgrade to 3.0, it is recommended to take a backup using 'backup snapshot <servername_date>', and then log into the webadmin page on your callbridge nodes to remove all XMPP Settings. Then connect to the MMP on your servers, and perform the these steps on all Core servers that have xmpp over a SSH connection:

1. **xmpp disable**
2. **xmpp reset**
3. **xmpp certs none**
4. **xmpp domain none**

Recorder

MMP

The Figures show an example of the configurations seen on CMS 2.9.1 when recorder was configured, and how it looks immediately after the upgrade to 3.0.

```
CMSRecorder> recorder
Enabled                : true
Interface whitelist    : a:443
Key file               : recorder.key
Certificate file       : recorder.cer
CA Bundle file        : rootca.cer
Trust bundle          : onecert.cer
NFS domain name       : 14.49.25.22
NFS directory         : E/Shares/Recordershare
Resolution            : 720p
CMSRecorder>

CMSRecorder> recorder
Enabled                : false
SIP interfaces        : none
SIP key file          : none
SIP certificate file   : none
SIP traffic trace     : Disabled
NFS domain name       : 14.49.25.22
NFS directory         : E/Shares/Recordershare
Resolution            : 720p
Call Limit            : none
CMSRecorder>
```

After the upgrade, you must reconfigure the recorder:

Step 1. Configure the SIP listening interface.

recorder sip listen a 5060 5061 (The interface and ports that SIP recorder is set up to listen on for TCP and TLS, respectfully. If you do not want to use TLS, you can use '**recorder sip listen a 5060 none**')

Step 2. Configure the certificates the recorder uses if you are using a TLS connection.

recorder sip certs <key-file> <crt-file> [crt-bundle] (Without these certificates, the tls service does not start on the recorder. The recorder uses the crt-bundle to verify the callBridge certificate.)

Step 3. Configure the call limit.

recorder limit <0-500|none> (Sets the limit for the number of simultaneous recordings the server can serve. This table is in our documentation and the recorder limit must align with the resources on the server.)

Table 6: Internal SIP recorder performance and resource usage

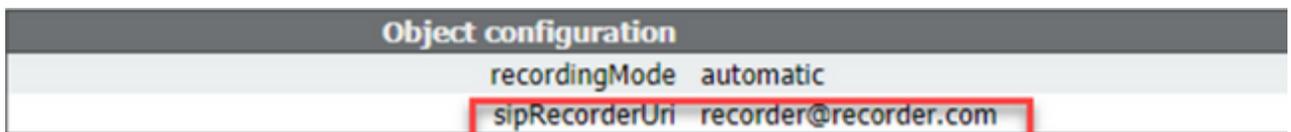
Recording Setting	Recordings per vCPU	RAM required per recording	Disk budget per hour	Maximum concurrent recording
720p	2	0.5GB	1GB	40
1080p	1	1GB	2GB	20
audio	16	100MB	150MB	100

Key point to note (applies to new internal recorder component only):

- Performance scales linearly adding vCPUs up to the number of host physical cores.

API

On `api/v1/callProfiles`, you need to configure the **sipRecorderUri**. This is the URI that the callBridge dials when it has to start a recording. The domain of this URI needs to be added to your outbound rules table and point to the recorder (or call control) as the SIP Proxy to Use.



This Figure shows a direct dial to the recorder component on the outbound rules found on **Configuration > Outbound Calls**.

Outbound calls

Filter:

	Domain	SIP proxy to use	Local contact domain	Local from domain	Trunk type	Behavior	Priority	Encryption
<input type="checkbox"/>	recorder.com	14.49.17.246:5061	Recorder	<use local contact domain>	Standard SIP	Continue	1	Encrypted
<input type="checkbox"/>	streamer.com	14.49.17.246:6001		<use local contact domain>	Standard SIP	Continue	1	Encrypted
<input type="checkbox"/>	recorder.com	14.49.17.246		<use local contact domain>	Standard SIP	Stop	0	Auto
<input type="checkbox"/>	streamer.com	14.49.17.246:6000	Streamer	<use local contact domain>	Standard SIP	Stop	0	Auto
					Standard SIP	Stop	0	Auto

This Figure shows a call to the recorder component via a call control (like for example Cisco Unified Communications Manager (CUCM) or Expressway).

Outbound calls

Filter

	Domain	SIP proxy to use	Local contact domain	Local from domain	Trunk type	Behavior	Priority	Encryption
<input type="checkbox"/>	recorder.com	14.49.17.229		<use local contact domain>	Standard SIP	Continue	1	Encrypted
<input type="checkbox"/>	streamer.com	14.49.17.229		<use local contact domain>	Standard SIP	Continue	1	Encrypted
<input type="checkbox"/>	recorder.com	14.49.17.252		<use local contact domain>	Standard SIP	Stop	0	Auto
<input type="checkbox"/>	streamer.com	14.49.17.252		<use local contact domain>	Standard SIP	Stop	0	Auto
					Standard SIP	Stop	0	Auto

CUCM (green arrow pointing to 14.49.17.229)

Expressway (red arrow pointing to 14.49.17.252)

Note: If you configured the recorder to use SIP TLS and if calls are failing, check your callBridge node in MMP to see if you have TLS SIP verification enabled. The MMP command is **'tls sip'**. Calls could fail because the recorder certificate is not trusted by the callBridge. You can test this by disabling this on the callBridge using **'tls sip verify disable'**.

Multiple recorders?

Configure each one as explained, and adjust your outbound rules accordingly. If you use a direct to recorder method, change the existing outbound to recorder rule to behavior "Continue" and add a new outbound rule beneath the previous one with the priority one less than the first one. When the first recorder has reached its call limit, it sends a 488 Unacceptable here back to callBridge, and the callBridge moves to the next rule.

If you want to load balance your recorders, use a call control and adjust your call control routing so it is able to place calls to multiple recorders.

Streamer

MMP

After the upgrade from 2.9.x to 3.0, you need to reconfigure streamer.

Step 1. Configure the SIP listening interface.

streamer sip listen a 6000 6001 (The interface and ports that SIP streamer is set up to listen on for TCP and TLS, respectfully. If you do not want to use TLS, you can use **'streamer sip listen a 6000 none'**)

Step 2. Configure the certificates the streamer uses if you are using a TLS connection.

streamer sip certs <key-file> <crt-file> [crt-bundle] (Without these certificates, the tls service does not start on the streamer. The streamer uses the crt-bundle to verify the callBridge certificate.)

Step 3. Configure the call limit

streamer limit <0-500|none> (Sets the limit for the number of simultaneous streams the server can serve. This table is in our documentation and the streamer limit must align with the resources on the server.)

Table 7: Internal SIP streamer recommended specifications

Number of vCPUs	RAM	Number of 720p streams	Number of 1080p streams	Number of audio-only streams
4	4GB	50	37	100
4	8GB	100	75	200
8	8GB	200	150	200

Key points to note (applies to both new internal recorder and streamer components):

- Number of vCPUs should not oversubscribe the number of physical cores.
- Maximum number of 720p streams supported is 200 regardless of adding more vCPUs
- Maximum number of 1080p streams supported is 150 regardless of adding more vCPUs.
- Maximum number of audio-only streams supported is 200 regardless of adding more vCPUs.

API

On `api/v1/callProfiles`, you need to configure the `sipStreamUri`. This is the URI that the callBridge dials when it has to start streaming. The domain of this URI needs to be added to your outbound rules table and point to the streamer (or call control) as the SIP Proxy to Use.

`/api/v1/callProfiles/a7f80cbd-5c0b-4888-b3cb-5109408a1dec`

Related objects: [/api/v1/callProfiles](#)

Table view XML view

Object configuration	
streamingMode	automatic
sipStreamerUri	stream@streamer.com

This Figure shows a direct dial to the streamer component on the outbound rules found on **Configuration > Outbound Calls**.

Outbound calls

Filter

	Domain	SIP proxy to use	Local contact domain	Local from domain	Trunk type	Behavior	Priority	Encryption
<input type="checkbox"/>	recorder.com	14.49.17.246:5061	Recorder	<use local contact domain>	Standard SIP	Continue	1	Encrypted
<input type="checkbox"/>	streamer.com	14.49.17.246:5001		<use local contact domain>	Standard SIP	Continue	1	Encrypted
<input type="checkbox"/>	recorder.com	14.49.17.246		<use local contact domain>	Standard SIP	Stop	0	Auto
<input type="checkbox"/>	streamer.com	14.49.17.246:6000	Streamer	<use local contact domain>	Standard SIP	Stop	0	Auto
					Standard SIP	Stop	0	Auto

This Figure shows a call to the recorder component via a call control (like for example Cisco Unified Communications Manager (CUCM) or Expressway).

Outbound calls

Filter Submit

	Domain	SIP proxy to use	Local contact domain	Local from domain	Trunk type	Behavior	Priority	Encryption
<input type="checkbox"/>	recorder.com	14.49.17.229		<use local contact domain>	Standard SIP	Continue	1	Encrypted
<input type="checkbox"/>	streamer.com	14.49.17.229		<use local contact domain>	Standard SIP	Continue	1	Encrypted
<input type="checkbox"/>	recorder.com	14.49.17.252		<use local contact domain>	Standard SIP	Stop	0	Auto
<input type="checkbox"/>	streamer.com	14.49.17.252		<use local contact domain>	Standard SIP	Stop	0	Auto
					Standard SIP	Stop	0	Auto

Annotations: A green arrow points from the 'SIP proxy to use' column to the 'Local contact domain' column. A red arrow points from the 'SIP proxy to use' column to the 'Local contact domain' column. A blue 'CUCM' label is placed above the first two rows. A red 'Expressway' label is placed above the last two rows.

Note: If you configured the streamer to use SIP TLS and if calls are failing, check your callBridge node in MMP to see if you have TLS SIP verification enabled. The MMP command is '**tls sip**'. Calls could fail because the streamer certificate is not trusted by the callBridge. You can test this by disabling this on the callBridge using '**tls sip verify disable**'.

Multiple Streamers?

Configure each one as explained, and adjust your outbound rules accordingly. If you use a direct to streamer method, change the existing outbound to recorder rule to behavior "Continue" and add a new outbound rule beneath the previous one with the priority one less than the first one. When the first streamer has reached its call limit, it sends a 488 Unacceptable here back to callBridge, and the callBridge moves to the next rule.

If you want to load balance your streamers, use a call control and adjust your call control routing so it is able to place calls to multiple streamers.

Expressway Consideration

If you use Cisco Expressway for Web Proxy, you must ensure your Expressway is running at least X12.6 before the CMS upgrade. This is required by CMS 3.0 for web proxy to work and to be supported.

The capacity for web app participants has increased over Expressways when used with CMS 3.0. For a large OVA Expressway, the expected capacity is 150 Full HD calls (1080p30) or 200 Other type calls (for example 720p30). You can increase this capacity by clustering Expressways, up to 6 nodes (where 4 is used for scaling and 2 for redundancy, so up to a maximum of 600 Full HD calls, or 800 Other type calls).

Table 3: Cisco Meeting Server web app call capacities – external calling

Setup	Call Type	CE1200 Platform	Large OVA Expressway
Cisco Expressway Pair (X12.6 or later)	Full HD	150	150
	Other	200	200

CMS Edge

CMS Edge is re-introduced in CMS 3.1 as that offers higher capacities than the Expressway for external web app sessions. There are two recommended configurations.

Small Edge Specs

4 GB RAM, 4 vCPUs, **1Gbps** network interface

This VM Edge spec has sufficient power to cover a single CMS1000 audio and video load capacity which is 48 x 1080p, 96 x 720p, 192 x 480p, and 1000 audio calls.

For the deployment it is recommended to have 1 small edge server per CMS1000 or 4 small edge servers per CMS2000.

Large Edge Specs

8 GB RAM, 16 vCPUs, **10Gbps** network interface

This VM Edge spec has sufficient power to cover a single CMS2000 audio and video capacity which is 350 x 1080p, 700 x 720p, 1000 x 480p, and 3000 x audio calls.

For the deployment it is recommended to have 1 large edge server per CMS2000, or per 4 CMS1000.

Type of Calls	1 x 4 vCPU VM call capacity	1 x 16 vCPU VM call capacity
Full HD calls, 1080p30 video	100	350
HD calls, 720p30 video	175	700
SD calls, 480p30 video	250	1000
Audio Calls (G.711)	850	3000