# Cisco Meeting Server

## Cisco Meeting Server Release 3.1.3

### Release Notes

June 01, 2021

# Contents

# What's changed

| Version | Change |
|---|---|
| June 1, 2021 | Third maintenance release (3.1.3).<br>Hashes updated.<br>See Resolved issues |
| April 28, 2021 | Minor edit to section 2.9.1. |
| March 15, 2021 | Updated the Short-term credentials for Cisco Meeting Server edge section with information about the feature being fully supported. |
| March 08, 2021 | Second maintenance release (3.1.2).<br>Hashes updated.<br>See resolved issues. |
| January 20, 2021 | Added 'Support for Vbrick API v2' in 'New Features' section |
| December 01, 2020 | VM and CMS2K hashes corrected. |
| November 30, 2020 | First release of version 3.1 |

# 1 Introduction

These release notes describe the new features, improvements and changes in 3.1 of the Cisco Meeting Server software.

The Cisco Meeting Server software can be hosted on:

- Cisco Meeting Server 2000, a UCS 5108 chassis with 8 B200 blades and the Meeting Server software pre-installed as the sole application.
- Cisco Meeting Server 1000, a Cisco UCS server preconfigured with VMware and the Cisco Meeting Server installed as a VM deployment.
- or on a specification-based VM server.

---

**Note:** Cisco Meeting Management 3.1 is required with Meeting Server 3.1. Meeting Management handles the product registration and interaction with your Smart Account (if set up) for Smart Licensing support.

---

Throughout the remainder of these release notes, the Cisco Meeting Server software is referred to as the Meeting Server.

If you are upgrading from a previous version, you are advised to take a configuration backup using the **`backup snapshot <filename>`** command, and save the backup safely on a different device. See the MMP Command Reference document for full details.

---

**Note about Microsoft RTVideo**: support for Microsoft RTVideo and consequently Lync 2010 on Windows and Lync 2011 on Mac OS, will be removed in a future version of the Meeting Server software. However, support for Skype for Business and Office 365 will continue.

---

## 1.1 Cisco Meeting Server platform maintenance

It is important that the platform that the Cisco Meeting Server software runs on is maintained and patched with the latest updates.

### 1.1.1 Cisco Meeting Server 1000 and other virtualized platforms

The Cisco Meeting Server software runs as a virtualized deployment on the following platforms:

- Cisco Meeting Server 1000
- specification-based VM platforms.

### 1.1.2 Cisco Meeting Server 2000

The Cisco Meeting Server 2000 is based on Cisco UCS technology running Cisco Meeting Server software as a physical deployment, not as a virtualized deployment.

CAUTION: Ensure the platform (UCS chassis and modules managed by UCS Manager) is up to date with the latest patches, follow the instructions in the Cisco UCS Manager Firmware Management Guide. Failure to maintain the platform may compromise the security of your Cisco Meeting Server.

### 1.1.3 Call capacities

Table 1 provides a comparison of the call capacities across the platforms hosting Cisco Meeting Server software version 3.1.

**Table 1: Call capacities across Meeting Server platforms**

| Type of calls | Cisco Meeting Server 1000 M4 | Cisco Meeting Server 1000 M5 | Cisco Meeting Server 2000 |
|---|---|---|---|
| Full HD calls<br>1080p60 video<br>720p30 content | 24 | 24 | 175 |
| Full HD calls<br>1080p30 video<br>1080p30/4K7 content | 24 | 24 | 175 |
| Full HD calls<br>1080p30 video<br>720p30 content | 48 | 48 | 350 |
| HD calls<br>720p30 video<br>720p5 content | 96 | 96 | 700 |
| SD calls<br>448p30 video<br>720p5 content | 192 | 192 | 1000 |
| Audio calls (G.711) | 1700 | 2200 | 3000 |

Table 2 below comes the call capacities for a single or cluster of Meeting Servers compared to load balancing calls within a Call Bridge Group.

Table 2: Meeting Server call capacity for clusters and Call Bridge groups

| Cisco Meeting Server platform | | Cisco Meeting Server 1000 M4 | Cisco Meeting Server 1000 M5 | Cisco Meeting Server 2000 |
|---|---|---|---|---|
| Individual Meeting Servers or Meeting Servers in a cluster (notes 1, 2, 3, and 4)<br>and<br>Meeting Servers in a Call Bridge Group | 1080p30<br>720p30<br>SD<br>Audio calls | 48<br>96<br>192<br>1700 | 48<br>96<br>192<br>2200 | 350<br>700<br>1000<br>3000 |
| | HD participants per conference per server | 96 | 96 | 450 |
| | web app call capacities (internal calling & external calling on CMS web edge):<br>Full HD<br>HD<br>SD<br>Audio calls | <br>48<br>96<br>192<br>500 | <br>48<br>96<br>192<br>500 | <br>350<br>700<br>1000<br>1000 |
| Meeting Servers in a Call Bridge Group | Call type supported | Inbound SIP<br>Outbound SIP | | |
| | Load limit | 96,000 | 96,000 | 700,000 |

Note 1: Maximum of 24 Call Bridge nodes per cluster; cluster designs of 8 or more nodes need to be approved by Cisco, contact Cisco Support for more information.

Note 2: Clustered Cisco Meeting Server 2000's without Call Bridge Groups configured, support integer multiples of maximum calls, for example integer multiples of 700 HD calls.

Note 3: Up to 16,800 HD concurrent calls per cluster (24 nodes x 700 HD calls) applies to SIP or web app calls.

Note 4: A maximum of 2600 participants per conference per cluster depending on the Meeting Servers platforms within the cluster.

Note 5: Table 2 assumes call rates up to 2.5 Mbps-720p5 content for video calls and G.711 for audio calls. Other codecs and higher content resolution/framerate will reduce capacity. When meetings span multiple call bridges, distribution links are automatically created and also count against a server's call count and capacity. Load limit numbers are for H.264 only.

Note 6: The call setup rate supported for the cluster is up to 40 calls per second for SIP calls and 20 calls per second for Cisco Meeting Server web app calls.

### 1.1.4 Cisco Meeting Server web app call capacities

This section details call capacities for deployments using Web Bridge 3 and web app for external and mixed calling. (For internal calling capacities, see Table 2.)

#### *1.1.4.1 Cisco Meeting Server web app call capacities – external calling*

Expressway (Large OVA or CE1200) is the recommended solution for deployments with small to medium web app scale requirements (i.e. 800 calls or less). However, for deployments that need larger web app scale, from version 3.1 we recommend Cisco Meeting Server web edge as the required solution which will scale up to SIP capacity (see Table 2).

For more information on using Cisco Meeting Server web edge solution, see Cisco Meeting Server 3.1 Release notes.

External calling is when clients use Cisco Meeting Server web edge, or Cisco Expressway as a reverse proxy and TURN server to reach the Web Bridge 3 and Call Bridge.

When using Expressway to proxy web app calls, the Expressway will impose maximum calls restrictions to your calls as shown in Table 3.

**Note: If you are deploying Web Bridge 3 and web app you must use Expressway version X12.6 or later, earlier Expressway versions are not supported by Web Bridge 3.**

**Table 3: Cisco Meeting Server web app call capacities – using Expressway for external calling**

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

The Expressway capacity can be increased by clustering the Expressway pairs. Expressway pairs clustering is possible up to 6 nodes (where 4 are used for scaling and 2 for redundancy), resulting in a total call capacity of four times the single pair capacity.

**Note:** The call setup rate for the Expressway cluster should not exceed 6 calls per second for Cisco Meeting Server web app calls.

#### *1.1.4.2 Cisco Meeting Server web app capacities – mixed (internal + external) calling*

Both standalone and clustered deployments can support combined internal and external call usage. When supporting a mix of internal and external participants the total web app capacity will follow Table 2 for Internal Calls and if using Cisco Meeting Server web edge solution for external calling. However, if using Expressway at the edge, the number of participants within the total that can connect from external is still bound by the limits in Table 3.

For example, a single standalone Meeting Server 2000 with a single Expressway pair supports a mix of 1000 audio-only web app calls but the number of participants that are external is limited to a maximum of 200 of the 1000 total.

## 1.2 Cisco Meeting Server web app Important information

If you are using Cisco Meeting Server web app (i.e. you have deployed Web Bridge 3), see Cisco Meeting Server web app Important Information for details on when features are released and issues resolved for the web app.

All information relevant to the web app is contained in this separate document and is not included in the Meeting Server release notes.

The Important Information guide describes the following:

- Any new or changed feature in the web app, and details of fixed issues and open issues associated with the web app with an indication of the version of Meeting Server where this feature/fix is available.
- Any upcoming changes in browsers affecting the web app, and the affected versions of the web app with recommended workarounds.

## 1.3 End of Software Maintenance

On release of Cisco Meeting Server software version 3.1, Cisco announces the time line for the end of software maintenance for the software in Table 4.

**Table 4: Time line for End of Software Maintenance for versions of Cisco Meeting Server**

| Cisco Meeting Server software version | End of Software Maintenance notice period |
|---|---|
| Cisco Meeting Server version 2.9.x | The last date that Cisco Engineering may release any final software maintenance releases or bug fixes for Cisco Meeting Server version 2.9.x is March 1, 2022. |

For more information on Cisco's End of Software Maintenance policy for Cisco Meeting Server click here.

# 2 New Features/Changes in version 3.1

Version 3.1 of the Meeting Server software, introduces the following new features and changes:

- Lobby/Welcome screen text allows the administrators to put the meeting title as over laid text on the welcome screen for SIP endpoints..
- New video enable/disable and audio mute/unmute mode so that administrators can now enable/disable video and/or mute/unmute audio for all participants.
- coSpace provisioning – extends the coSpace provisioning feature that allows the administrator to provision spaces for users based on an LDAP sync.
- RTMPS streaming support – extends the RTMP support in the internal SIP streamer application to RTMPS which allows streamer traffic to be encrypted.
- Serviceability enhancements:
  - improvements to ICE tracing.
  - packet capture improvements to increase the packet capture size when a server is under load to ensure that packet captures are as useful as possible.
  - packet capture now available on multiple interfaces.
  - improved TURN server logging.
- Cisco Meeting Server web app introduces many new features in 3.1. For a complete list of the web app features introduced in 3.1, see Cisco Meeting Server 3.1 web app Important Information. Web app features that require Meeting Server-side configuration are listed below:
  - changes to Cisco Meeting Server web app custom email invites
  - a user can now view call join information whilst in a call
  - web app user interface localization
  - single sign on
- Web Admin user interface changes – external access configuration options moved to webBridgeProfiles API.
- New edge solution using Cisco Meeting Server at the edge for increased web app scale.
- Security enhancement with short term credentials for Cisco Meeting Server edge.
- Scheduled LDAP sync API additions and ldapSources object name label addition.

## 2.1 Video enable/disable and audio mute/unmute modes

Version 3.1 introduces a new video enable/disable and audio mute/unmute mode that separates the link between local and remote mute so that administrators can now

enable/disable video and/or mute/unmute audio for all participants.

From 3.1 administrators can select between "linked" and "separate" video enable/disable and audio mute/unmute behaviors – where "linked" is the behavior from previous releases, i.e. the local video enable/disable and audio mute/unmute status of the device (such as ActiveControl-enabled endpoints, Jabber clients 12.5 or later, or Meeting Server web app) mirrors the server video enable/disable and audio mute/unmute status. This mode means, for example, that when administrators mute participants, the participants are muted locally on their device and have to unmute themselves locally (the administrator cannot unmute them). The existing mute behavior is described in this FAQ.

The new "separate" mode is where the local and server video enable/disable and audio mute/unmute mode status are not connected. This new mute mode is introduced to support use cases where, for example, local mute is not used and administrators need the ability to mute and unmute participants without interaction from the participants.

For existing users, on upgrade to 3.1 the default behavior will be "linked" so the user experience will be the same as prior to 3.1 unless reconfigured to use "separate" video enable/disable and audio mute/unmute mode.

### 2.1.1 New audio mute mode behavior

#### *2.1.1.1 All endpoints / clients (local and server audio mute are separate)*

- When users mute themselves with the local mute button they are muted locally but not on the Meeting Server/Meeting Management:

| Endpoint / client type | User sees... | Other participants.. | Meeting Management... | Can user locally unmute? |
|---|---|---|---|---|
| CE | a local mute indicator | will not see that they are muted in the participant list | will not see that they are muted | yes |
| Jabber | (not applicable) local mute icon dimmed/disabled | (not applicable) mute icon dimmed/disabled | (not applicable) mute icon dimmed/disabled | (not applicable) |
| web app | a local mute indicator | other web app participants will see that they are muted | will not see that they are muted | yes |

- When another user unmutes them or when the actual user sends a DTMF unmute command then the server mute will be removed, and the behaviors are as follows:

| User sees.. | Other participants.. | Meeting Management... | Can user locally unmute? |
|---|---|---|---|
| mute icon disappears from the video stream | will still see them as being unmuted | will see them as being unmuted | no, user cannot unmute themselves by pressing the local mute button, but can via DTMF, if configured |

### 2.1.2 New video enable/disable mode behavior

#### *2.1.2.1 Applies for web app (local and server video enable/disable are separate)*

- When users disable their video themselves locally:

| User sees.. | Other participants.. | Meeting Management... | Can user locally enable video? |
|---|---|---|---|
| message to say video is not being sent | not notified | not notified | yes, the user can enable video locally to send video again |

### 2.1.3 New API request parameter to support the new video enable/disable and audio mute/unmute mode

A new **`muteBehavior`** API request parameter is introduced in 3.1 to implement the new mute behavior. This parameter to set the mute mode of a call with the new type of **`linked`** or **`separate`** is introduced for:

- POST to **`/callProfiles`**
- PUT to **`/callProfiles/<call profile id>`**
- GET on **`/callProfiles/<call profile id>`**

## 2.2 Support for Vbrick API v2

Cisco Meeting Server 3.1 includes support for Rev API v2.

The following table provides information about the Vbrick API versions that are supported in Cisco Meeting Server releases:

**Table 5: Vbrick API versions supported in Cisco Meeting Server releases**

| Vbrick API versions | Cisco Meeting Server releases |
|---|---|
| Rev API v2 and Rev API v1 | Cisco Meeting Server 3.1 |
| Rev API v1 | Cisco Meeting Server 3.0 |
| Rev API v1 | Cisco Meeting Server 2.9 |

**Note:** Cisco Meeting Server versions 2.9 and 3.0 only support Rev API v1, for which support will be stopped by April 30, 2021. For more information, refer to https://portal.vbrick.com/rev-developers/. Users of Cisco Meeting Server versions 2.9 and 3.0 will be unable to use future Vbrick releases, especially if they are using the Vbrick cloud, which might stop working without prior notice.

## 2.3 coSpace provisioning

Version 3.1 extends the coSpace provisioning feature that allows the administrator to provision spaces for users based on an LDAP sync.

Previously you could provision spaces as part of the LDAP sync by specifying the following parameters in the ldapMapping object as part of "user import": coSpaceUriMapping, coSpaceSecondaryUriMapping, coSpaceNameMapping, coSpaceCallIdMapping. Version 3.1 still supports this old deprecated method of provisioning spaces, however, multiple spaces can now be provisioned for users using the new and improved space provisioning method and we recommend that spaces are provisioned using this method rather than the old, deprecated, one.

**Note:** The deprecated coSpace provisioning method may be removed at some point in the future.

This feature introduces the following new API objects in version 3.1:

- **`/ldapUserProvisionedCoSpaceMappings`**
- **`/ldapUserProvisionedCoSpaceMappings/<LDAP user provisioned coSpace mapping id>`**
- **`/ldapUserProvisionedCoSpaceSources`**
- **`/ldapUserProvisionedCoSpaceSources/<LDAP user provisioned coSpace mapping id>`**
- **`/ldapUserProvisionedCoSpaceSources/<LDAP user provisioned coSpace source id>`**
- **`/users/<user id>/userProvisionedCoSpaces`**
- **`/users/<user id>/userProvisionedCoSpaces/<user provisioned coSpace id>`**

The **`userProvisionedCoSpace`** parameter is introduced to the /cospaces object.

For details of all API additions to support this feature, see the API additions summary.

### 2.3.1 How to provision coSpaces to users

Create the ldapUserProvisionedCoSpaceMapping:

1. Using the Meeting Server Web Admin interface:
    a. Log in to the Meeting Server Web Admin interface and select **Configuration > API**:
    b. From the list of API objects, tap the ► after **/api/v1/ldapUserProvisionedCospaceMappings**
    c. Click **Create new**.
    d. Define the **coSpaceUriMapping** to the required coSpaces' URI.
    e. Optional. Set the **coSpaceNameMapping**, if desired. For example, setting coSpaceNameMapping to "$cn$ personal coSpace" ensures that each user's coSpace is labelled with their name followed by "personal coSpace".
    f. Go to the **coSpaceTemplate** field and click **Choose**.
    g. From the resulting "coSpaceTemplate object selector window", click **Select** for the **object id** of the **coSpaceTemplate** that you wish to assign to the user provisioned coSpace mapping.
    h. Click **Create**.

Create the ldapUserProvisionedCoSpaceSource:

2. From the list of API objects, tap the ► after **/api/v1/ldapUserProvisionedCoSpaceSource**
    a. Click **Create new**.
    b. Go to the **ldapSource** field and click **Choose**.

    From the resulting "ldapSource object selector window", click **Select** for the **object id** of the **ldapSource** that you wish to use. (That is, the source that will provide the list of users who will be provisioned cospaces.)
    c. Go to the **ldapUserProvisionedCoSpaceMapping** field and click **Choose**.

    From the resulting "ldapUserProvisionedCoSpaceMapping object selector window", click **Select** for the **object id** of the **ldapUserProvisionedCoSpaceMapping** that you have just created in Step 1.
    d. Optional. Set **filter**, if required. This is an additional LDAP filter string to be applied when reading the source.

    ---

    **Note:** The set of users that will be applied with the coSpaceSource is defined by the set produced by the ldapSource, filtered by the ldapUserProvisionedCoSpaceSource 'filter' attribute.

    ---

    e. Click **Create**.

Now you have your LDAP source and an LDAP mapping, you can now do an LDAP sync.

3. From the list of API objects, tap the ► after **/api/v1/ldapSyncs**
   a. Click **Create new**.
   b. Go to the **ldapSource** field and click **Choose**.
   c. From the resulting "ldapSource object selector window", click **Select** for the **object id** of the **ldapSource** that you wish to sync.
   d. Click **Create** to perform the LDAP sync.
4. From the list of API objects, tap the ► after **/api/v1/users** to display a list of user provisioned coSpaces.

Web app users will now be able to activate a newly provisioned coSpace from the web app UI. For more information, see Cisco Meeting Server web app Important Information.

## 2.4 RTMPS streaming

Version 3.1 extends the RTMP support in the internal SIP streamer application to RTMPS – essentially RTMP over a TLS connection. Previously all traffic between the streamer and RTMP server was unencrypted, 3.1 RTMPS support allows this traffic to be encrypted.

The supported feature set for RTMPS and configuration steps are the same as for RTMP as introduced and documented in 3.0.

RTMPS support introduces new MMP commands to allow configuration of the TLS certificates and trusts. Additionally, support for stream URLs prefixed with **`rtmps://`** is introduced for the **`streamUrl`** parameter on the coSpaces API which is required to configure an RTMPS stream.

### 2.4.1 New MMP Commands

The existing **`tls`** MMP command is extended to optionally allow configuration of TLS trusts for RTMPS. This step is optional but recommended. If a TLS trust is not configured then the RTMPS connection will not be secure. The new output of the **`help tls`** command is:

```
cms> help tls
Configure TLS operations
Usage:

    tls
    tls (sip|ldap|dtls|webadmin|rtmps)
    tls (sip|ldap|rtmps) trust <crt bundle>
    tls (sip|ldap|rtmps) verify (enable|disable|ocsp)
    tls sip ciphers <cipher string>
    tls (sip|ldap|webadmin|rtmps) min-tls-version
    <minimum version string>
```

```
tls min-dtls-version <minimum version string>
```

The trust and verify commands can be used to set the trust store and then enable it. The configured **min-tls-version** is also supported.

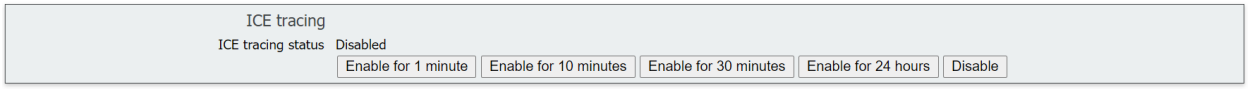
### 2.4.2 Configuring RTMPS URL

Configuring RTMP streaming requires that you configure the stream URL by setting the **streamUrl** parameter in the coSpaces API. This remains the same in 3.1, but you will now be able to enter an URL prefixed **rtmps** instead of **rtmp** to indicate that it is an RTMPS URL. For example, **rtmps://mystreamer.com/stream**

## 2.5 Serviceability enhancements

### 2.5.1 ICE tracing

There is an ICE tracing button on the Web Admin interface under **Logs > Detailed tracing** which provides log messages.




3.1 improves the usefulness of the ICE detailed tracing log messages by including full details of message contents and removing some unnecessary but verbose messages which obscured the main flow.

### 2.5.2 Packet capture improvements

Version 3.1 introduces packet capture improvements to increase the packet capture size when a server is under load. The maximum file capture size is limited to 75% of the current available file system space, or 1GB whichever is smaller, however, this limit doesn't allow for a very long capture on a server under load.

Version 3.1 introduces two new options to the **pcap** MMP command to ensure that useful packet captures can be achieved on a server under load without reaching the file size limit:

- **snaplen <length>** – this option truncates each packet captured to the maximum number of bytes if it is longer. As a result, more packets can fit into the same file-size limit. Although this means the exact packet data cannot be extracted, there are some instances where this is not an issue, for example:
  - when media is encrypted but this still allows you to count packets
  - when the presence/absence of specific traffic is what you need to check

- **filter <filter-string>** – this option selects only packets matching the criteria in the string. This reduces the capture to only packets of interest, and avoids wasting disk space on the others. The parsing of this string and the packet filtering are performed with exactly the same underlying libraries as used by tcpdump, so this has exactly the same expressive power and performance. The filter expression can be up to around 4080 characters long, if required.

These two options can be combined, using the **snaplen** command first as everything after **filter** is treated as the filter expression. Here's an example of using these two options:

```
pcap a snaplen 40 filter udp host 10.10.3.4 and portrange 40000-50000
```

This example command captures only UDP traffic on high-numbered ports to or from 10.10.3.4 and keeps only the first 40 characters of each (probably media) packet.

#### *2.5.2.1 Packet capture on multiple interfaces*

The **pcap** MMP command now accepts **any** for an interface name, this will capture on any enabled interfaces (interfaces that are not enabled will be skipped). Specifying a single interface name as before is still supported.

---

Note: When capturing from multiple interfaces, this requires additional disk space as each interface is captured to a separate temporary file and the files are then merged when the capture is stopped. So the available storage when capture on multiple interfaces is half what is available when capturing on a single interface.

---

The new option **any** is used as follows:

```
pcap <interface>|any [snaplen <length>] [filter <filter-string>]
```

### 2.5.3 Improved TURN server logging

Prior to 3.1, the TURN server generated few syslog entries in normal operation. The logging is now extended in 3.1 to track the start and end of individual media streams and the TURN server IP address is now added to the failed connection message.

## 2.6 Cisco Meeting Server web app new features and changes

Version 3.1 introduces some new features and changes on Cisco Meeting Server web app.

---

Note: For details of all new 3.1 web app features, see Cisco Meeting Server 3.1 web app Important Information. The new web app features listed below are those that may require server-side configuration or introduce API additions/changes.

---

### 2.6.1 Cisco Meeting Server web app custom email invite changes

**Note:** On upgrade to 3.1, administrators must recreate all custom email invite templates as there is no migration path for existing custom email invite templates from a previous release to 3.1.

Version 3.1 introduces the following changes to custom email invitation templates:

- Custom email invitation template language is extended to support multiple IVR numbers and Web Bridge addresses.
- Custom email invitation template content is now created using new syntax.
- All custom email invitation templates must now start with "Subject:" followed by an empty line to separate the header from the body of the email. Only "Subject:" header is supported for now.
- New default languages added for email invitation templates.

#### *2.6.1.1 Custom email invitation language extended*

In 3.1 the template language is extended to support multiples IVR numbers and Web Bridge addresses as follows:

For Web Bridges:

```
#for webbridge in web_bridge_addresses
%webbridge.label% : %webbridge.address%
%webbridge.hyperlink%
#endfor
```

For PSTN dial ins:

```
#for pstn in ivr_numbers
%pstn.label% : %pstn.number%
#endfor
```

#### *2.6.1.2 Custom email invitation content created with new syntax*

Version 3.1 introduces new syntax to create a custom email invitation.

**Note:** On upgrade to 3.1, administrators must recreate all custom email invite templates as there is no migration path for existing custom email invite templates from a previous release to 3.1.

**Header**

All custom email templates must now start with "Subject:" followed by an empty line to separate from the body text. The subject header is used to generate the email link that sets the text specified in the header as subject of the email. The header cannot contain new line characters. The syntax is:

**`Subject: <subject….>`** [followed by an empty line]

### Format

The new custom email invitation template syntax allows variables, loops, conditions, includes, callbacks, and comments (nested and combined as required).

There are escape sequences that allow you to write conditional code, loops and so on, i.e. New line escape sequence (the same as prior to 3.1). Any new line starting with # introduces a template statement. Example:

```
#if name
I can use a name variable as it is set to %name%
#endif
```

You can specify an inline statement by using "{%" and "%}". Example:

```
You have been invited to a meeting{% if name %}: %name%{% endif %}
```

You can also add comments to your templates using "{# my comment #}". So any text between those escape sequences won't be rendered. Example:

```
Hello, {# This section is just an intro #}
```

### Variables

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

%<var name>% will be substituted with the content of a variable.

The variables that are currently defined are detailed below:

Table 6: Variables in invitation template

| Variable name, description, and example |
| --- |
| name<br><br>This will be replaced by the name of the cospace. To see if it has content, check its length or whether it's not null.<br>Example:<br>`#if name`<br>`The cospace name is %name%`<br>`#endif`<br><br>`#if length(name)>0`<br>`The cospace name is %name%`<br>`#endif` |

| Variable name, description, and example |
| --- |
| uri<br><br>The cospace uri that can be used to 'join by uri' in the web app or dialed-in on endpoints.<br>It can be checked by length or if set to null.<br>`#if uri`<br>`The uri is: %uri%`<br>`#endif`<br><br>`#if length(uri) > 0`<br>`The uri is: %uri%`<br>`#endif` |
| numeric_id<br><br>This is the callId of the cospace. It is normally a numeric identifier for the cospace.<br>It can be checked on length or if set to null.<br>Example:<br>`#if numeric_id`<br>`The meeting id is %numeric_id%`<br>`#endif`<br><br>`#if length(numeric_id) > 0`<br>`The meeting id is %numeric_id%`<br>`#endif` |
| passcode<br><br>The passcode assigned to the cospace.<br>It can be checked on length or if set to null.<br>Example:<br>`#if passcode`<br>`The meeting requires a passcode: %passcode%`<br>`#endif`<br><br>`#if length(passcode) > 0`<br>`The meeting requires a passcode: %passcode%`<br>`#endif` |

| Variable name, description, and example |
| --- |
| ivr_numbers<br><br>This is an array of objects that contains label and number.<br>(It supersedes the dial_pstn variable that was used in custom email templates prior to 3.1.)<br>It can be an empty array and it needs to be looped over to access the internal variables: label and number.<br>It cannot be checked for null, only by length.<br>• label: The label assigned to this IVR number when registering it using the Meeting Server API.<br>• number: The IVR number to dial in.<br><br>Example:<br>`#if length(ivr_numbers) > 0`<br>`You can use the following dial-in number: %ivr_numbers.0.number%`<br>`%`<br>`#endif`<br><br>`#for ivr_number in ivr_numbers`<br>`%loop.index1% - %ivr_number.label%: %ivr_number.number%`<br>`#endfor` |
| web_bridge_addresses<br><br>This is an array of objects that contain label, address and hyperlink.<br>(It supersedes the hyperlink and webbridge_url variables used in custom email templates prior to 3.1.)<br>It can be empty but it cannot be checked for null, only by length.<br>• label: The label assigned to this Web Bridge Address when registering it via the Meeting Server API. Example: web app address<br>• address: The HTTPS address that can be used to access the web app on this Web Bridge. Example: https://join.mydomain.com/<br>• hyperlink: When a cospace has a callId non-empty (aka, meeting id or number_id) and a non-empty address in this object, then a unique hyperlink will be generated that can be used to directly join a given meeting without requesting a passcode.<br><br>Example:<br>`#if length(web_bridge_addresses) > 0`<br>`You can use the following web app address: %web_bridge_addresses.0.address%`<br>`#endif`<br><br>`#for wba in web_bridge_addresses`<br>`Label: %wba.label%`<br>`Address: %wba.address%`<br>`Hyperlink: %wba.hyperlink%`<br>`#endfor` |

### Loops

You can use loops by using the "for" and "endfor" keywords. Example:

```
#for ivr_number in ivr_numbers
%loop.index1% - %ivr_number.label%: %ivr_number.number%
#endfor
```

In a loop, the special variables are:

- loop.index (number): Loop iteration number starting from 0.
- loop.index1 (number): Loop iteration number starting from 1.
- loop.is_first (boolean): If the iteration is the first iteration.
- loop.is_last (boolean): If the iteration is the last iteration.
- loop.parent.*: In nested loops, the parent loop variables are available using loop.parent.<name of var> example: loop.parent.is_first.

You can also iterate over objects like web_bridge_address or ivr_number. Example:

```
#for ivr_number in ivr_numbers
{%- for key, value in ivr_number -%}
%key%: %value%
{%- endfor -%}
#endfor
```

### Conditions

Conditions support the typical "if" and "else" statements. Examples:

Print something different depending on the length of a passcode

```
{% if length(passcode) >= 3 %}…{% else if length(passcode) >= 10 %}…{% endif %}
```

Print something only if a particular field is set in a web_bridge_address

```
{% if web_bridge_address.hyperlink %}…{% endif %}
```

### Logical operations

You can use "and", "or" and "not" to generate complex conditions. Example:

```
{% if numeric_id and passcode %}…{% endif %}
{% if not name %}…{% endif %}
```

### Functions

A few functions are implemented within the template syntax. These are detailed below.

Upper and lower function, for string cases. Example:

```
Join {% upper(name) %}
Join {% lower(name) %}
```

Range function, useful for loops. Example:

```
{% for i in range(4) %} %loop.index1% {% endfor %}
```

```
{# It will show only the first two web_bridge_addresses #}
{% for i in range(2) %} %at(web_bridge_addresses, i).address% {% endfor %}
```

Get first and last element in a list. Example:

```
The first ivr_number: % first(ivr_numbers).number %
The last ivr_number: % last(ivr_numbers).number %
```

Sort a list. . Example:

```
#for sort(ivr_numbers)
...
#endfor

{# Produces [1, 2, 3] #}
% sort([3,2,1]) %
```

Round numbers to a given precision. . Example:

```
{# returns 1 #}
% round(1.4142135, 0) %
{# returns 1.4 #}
% round(1.4142135, 1) %
```

Check if a value is odd, even or divisible by a number. Example:

```
{# returns true #}
% odd(1) %
{# returns true #}
% even(2) %
{# returns true #}
% divisibleBy(42, 7) %
```

Maximum and minimum values from a list. Example:

```
{# returns 3 #}
% max([1,2,3]) %
{# returns 1 #}
% min([1,2,3]) %
```

Convert strings to numbers. Example:

```
# if int(ivr_number) >= 123123
...
#endif
```

Set default values if variables are not defined. Otherwise, the template rendering will fail and show no result. Example:

```
{% if default(has_valid_ivr_number, false) %}... {% endif %}
```

Check if a key exists in an object. Example:

```
{# returns false #}
% exists("pstn_dial_in") %
{# returns true #}
% existsIn("a", myobj) %
{# returns false #}
```

```
% existsIn("c", myobj) %
```

Check if a key is a specific type. Example:

```
{# returns true #}
% isArray(ivr_numbers) %
{# returns true #}
% isString(ivr_numbers.0.number) %
{# returns false #}
% isString(int(ivr.numbers.0.number)) %
Implemented type checks are: isArray, isBoolean, isFloat, isInteger, isNumber,
isObject, isString,
```

#### Whitespace Control

Whitespaces are removed by default. To support a more readable template style you can also remove whitespaces for both statements and expressions by hand. If you add a minus sign (-) to the start or end, the whitespaces before or after that block will be removed. Example:

```
Cospace name: % name -%        .
{# Produces: Cospace name: blah. #}
```

Stripping behind a statement or expression also removes any new lines.

### *2.6.1.3 Example invitation template*

Use the example below and customize with your specific values in the variables. Save it using the appropriate file name format.

---

**Note:** The "Subject:" header must be followed by an empty line to separate from the body text.

---

```
Subject: {% if name %}You are invited to join a meeting: %name%{% else %}You
are invited to join a meeting{% endif %}

#if numeric_id
Meeting ID: %numeric_id%
#if passcode
Meeting passcode: %passcode%
#endif
#endif

#for wba in sort(web_bridge_addresses)
#if wba.address or wba.hyperlink
#if loop.index == 0
Join from a computer, mobile phone or tablet
#endif
{% if wba.label %} %wba.label%:{% endif %} {% if wba.hyperlink
%}%wba.hyperlink%{% else %}%wba.address%{% endif %}
#endif
#endfor
```

```
#for ivrn in sort(ivr_numbers)
#if ivrn.number
#if loop.index == 0
Join by phone
#endif
{% if ivrn.label %} %ivrn.label%:{% endif %} %ivrn.number%
#endif
#endfor

#if uri
Join from a video conferencing system or application
Dial %uri%
#endif
```

#### *2.6.1.4 Invitation templates for different languages*

There are now 21 default language invitation email templates for web app on the Meeting Server to choose from. The invitation templates take the format:

- invitation_template_xx_XX.txt

---

**Note:** Where the language .txt files have the appropriate language tags for its language variant (as defined by the IANA Language Subtag Registry) – the two lower case characters indicate the language code and the two upper case characters the region code. For example, invitation_template_fr_CA.txt, where "fr" is the French language, and "CA" is the region (Canada).

---

As previously, if you want to overwrite the default templates, you can create your own language tagged template file and upload to locally hosted branding. The Meeting Server interprets these language tags to return the appropriate template option in the web app.

A web app user can only select a language template that is uploaded – a language option is not shown to the web app user in the drop-down list if it is not uploaded.

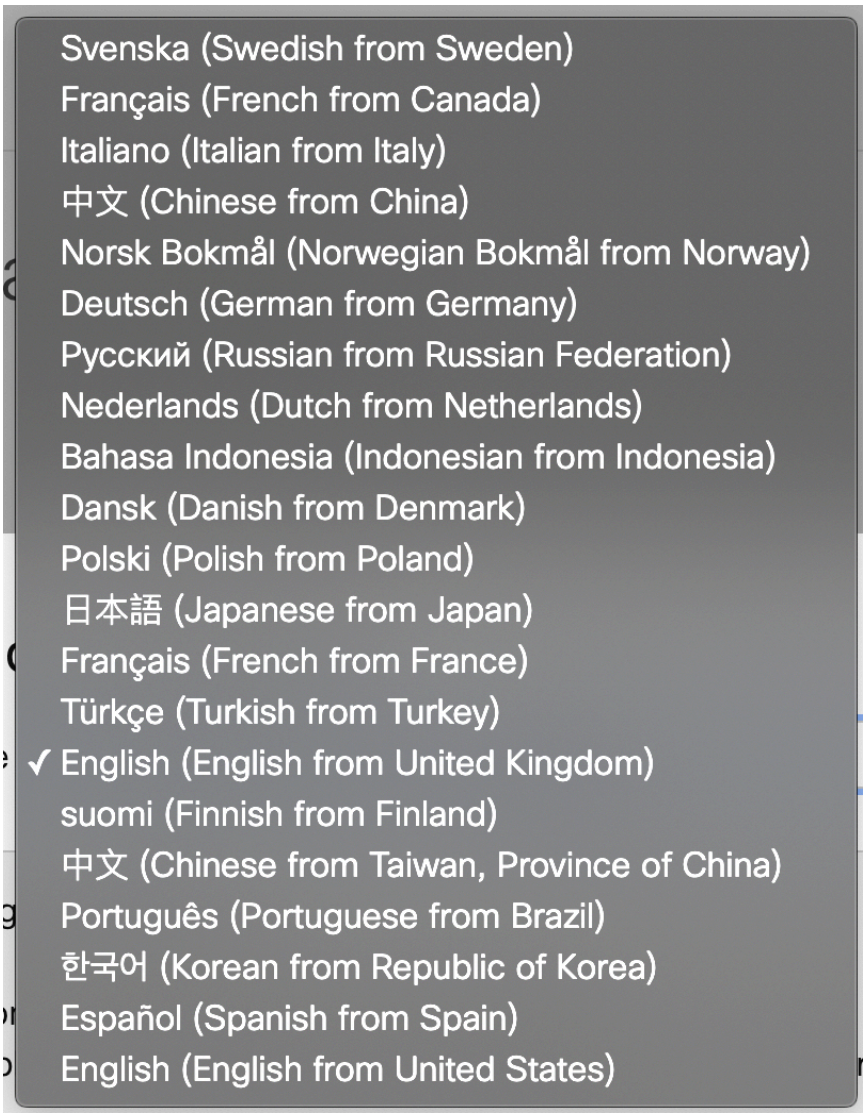
---

**Note:** You must upload the template files to all Meeting Servers in a cluster.

---

The figure below shows an example of some of the different email invites languages that are available by default:

Figure 1: Email invite options




For more information, see the Customization Guidelines.

### 2.6.2 View call join information

This feature allows a user to view the space name, call duration, and call join information whilst a call is in progress. Using this join information, the user can invite more participants to the meeting.

This feature introduces the new API types:

- on request parameter **`scope`** for POST to **`/coSpaces/<coSpace id>/accessMethods`** and PUT to **`/coSpaces/<coSpace id>/accessMethods/<access method id>`**:
  - **`member`** – details of this coSpace access method are visible to members of the coSpace

- **directory** – details of this coSpace access method can be found through search [Note: in 3.1 there is no search, so behavior is the same as **public**]

Additionally, the existing API type defintions for **public** and **private** are updated:

- **public** – details of this coSpace access method are visible to members of the coSpace and all participants in the meeting
- **private** – details of this coSpace access method is visible only to the owner of the space in web app, or visible to admin users using the Call Bridge API.

### 2.6.3 Cisco Meeting Server web app localized user interface

Version 3.1 introduces localization of the web app user interface in 21 languages. The default web app user interface language is based upon the browser's default. If a web app user then wishes to select a different language, they can then do so before signing in to the app or joining a meeting. For more information on how web app users select their preferred language, see Cisco Meeting Server 3.1(or later) web app Important Information.

To support this new feature, Cisco Meeting Server has a new web app asset: "text_strings_xx_XX.json" as described in Table 7. These languages are part of the software and require no additional configuration unless you want to customize a particular language file. If a specific language file doesn't exist, Meeting Server will default to using the text_strings.json file.

Table 7: web app asset description and specification

| File name | Description | Max file-size | Recommended sizes, formats and aspect ratios |
|---|---|---|---|
| text_strings_xx_XX.json | Text strings for a particular language; for example "text_strings_fr_CA.json" will offer the web app user interface in French Canadian.<br>Supports the same as text_strings.json.<br>Any text strings defined in this format will override those specified in "text_strings.json" for the specified language.<br>Supported strings:<br>• brand_title: Main brand name<br>• brand_subtitle: Secondary text below<br>• brand_title brand_tag_line: Tertiary text below<br>• brand_subtitle brand_browser_tab_label: The name of the tab in the browser | 16 kb | Recommended lengths:<br>• brand_title: up to 24 characters (displays on 1 line), or up to 48 characters (displays on 2 lines).<br>• brand_subtitle: up to 24 characters (displays on 1 line), or up to 48 characters (displays on 2 lines).<br>• brand_tag_line: up to 100 characters<br>• brand_browser_tab_label: up to 64 characters |

## 2.7 Single Sign On (SSO) for Cisco Meeting Server web app

This feature allows a web app user to login using an SSO provider to verify their identity.

SSO means the web app user doesn't need to enter their password every time they sign in as they can now have a single session with an identity provider (the entity responsible for authenticating users at a single place and maintaining a single session for each, for example, OAuth, gmail).

It allows the web app user to login with different SSO providers on the same Web Bridge.

This SSO mechanism uses SAML (Security Assertion Markup Language) 2.0 which is an open standard and a widely used industry standard protocol.

**Note:** Currently Meeting Server supports only HTTP-POST bindings in the SAML 2.0 protocol. This means it will only accept messages on its HTTP-POST AssertionConsumerService and it will reject Identity Providers with no HTTP-POST bindings available

**Note:** If you enable SSO login, you can no longer use LDAP login.

### 2.7.1 Configuring SSO for use on Meeting Server web app

To use SSO requires some configuration for the identity provider and on the Meeting Server (regarded as the Service Provider in the SAML 2.0 exchange) as detailed below.

*Task 1: Mapping between Identity provider and Meeting Server users*

So that Meeting Server can correctly map users on your Identity provider to its own users you will need to setup an authenticationId for every user authenticated via SSO. This can be done as part of the standard ldap sync process. The contents of this field will be verified against a custom parameter passed from the Identity provider with successful responses (see Task 2).

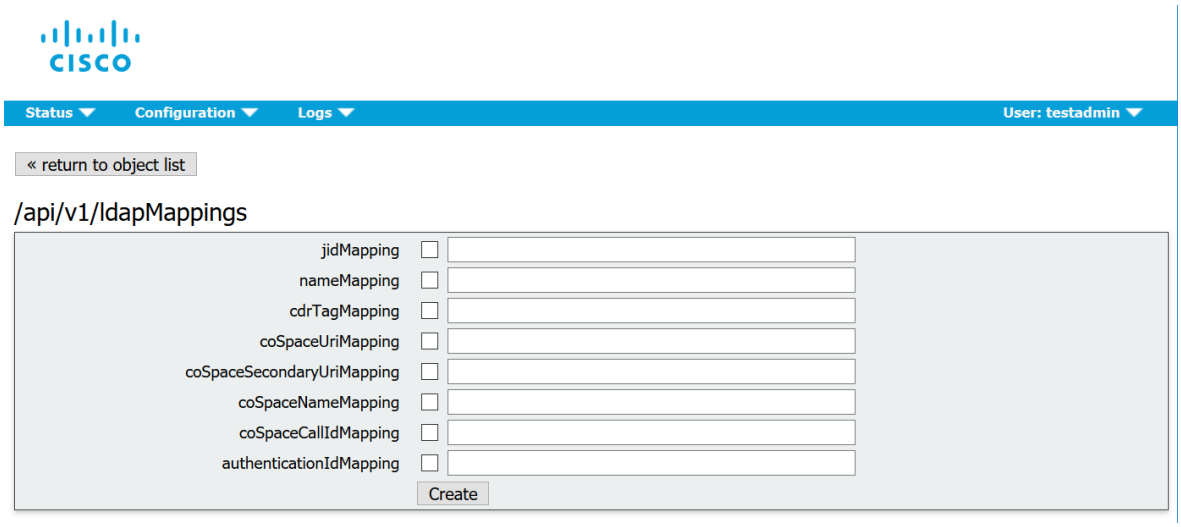
We recommend that you choose a unique identifier for each user (e.g. $sAMAccountName$). Empty values for the authenticationIds are not accepted.

To setup the authenticationId as part of an ldapSync you can create a new ldapSync or modify an existing one.

You then need to create/modify an ldapMapping and populate the **authenticationIdMapping** parameter with an appropriate value (e.g. $sAMAccountName$).

Using the Meeting Server Web Admin interface:

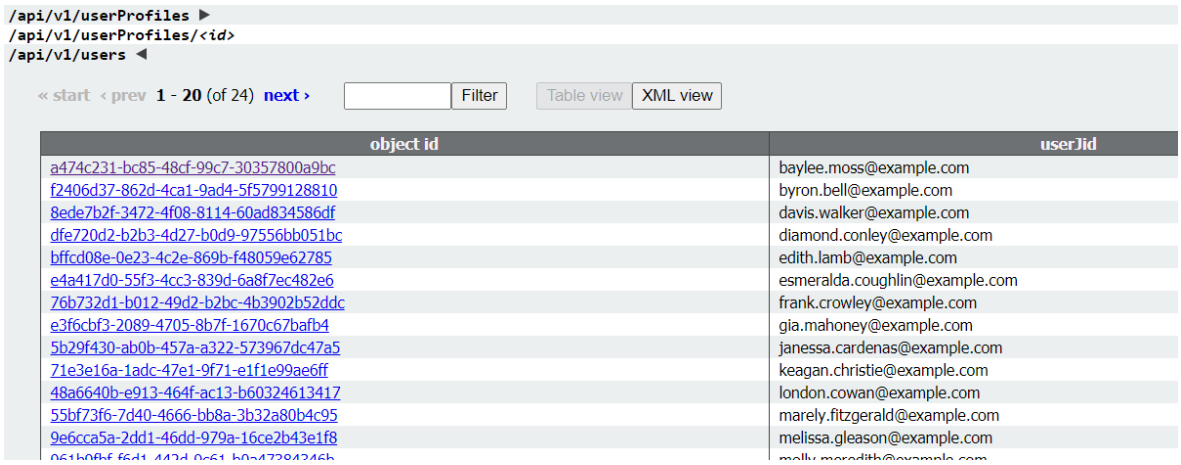
a. Log in to the Meeting Server Web Admin interface and select **Configuration > API**:
b. From the list of API objects, tap the ► after **/api/v1/ldapMappings**
c. Click **Create new** or select the ID for an existing ldap mapping to modify.



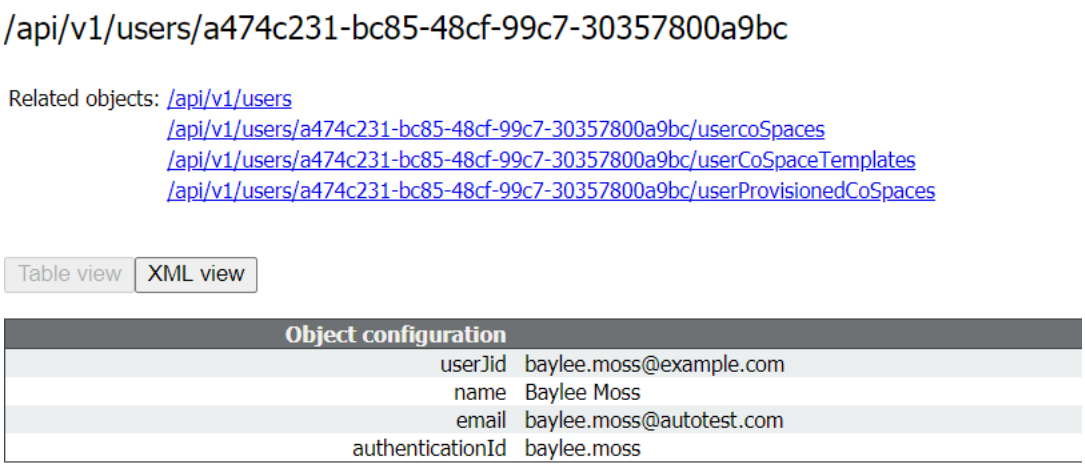

d. Populate the **authenticationIdMapping** parameter with an appropriate value (e.g. $sAMAccountName$ ) and click **Create** or **Modify**, as appropriate.

e. For the changes to take effect on the Meeting Server you now need to trigger an ldapSync. From the list of API objects, tap the ▶ after **/api/v1/ldapSyncs** and select the object ID or **Create new**, as appropriate. Once the ldapSync has finished you can verify that the process has succeeded by examining one of your Meeting Server users.

f. Firstly, from the list of API objects, tap the ▶ after **/api/v1/users**, to display a list of users as seen in this example:




g. Select one of the users that should now have authenticationId set up (you may need to use the Filter field). The user entry should now include an **authenticationId** field with the correct value from the ldapSync as shown in this example:




*Task 2: Identity Provider configuration*

1. All identity providers let you upload a metadata xml file representing the Service Provider being registered with them (i.e. the Meeting Server in this instance). Some identity providers simplify the process by allowing you to configure the most important pieces of information. Metadata xml file examples can be found here.

   The values to include in the metadata xml file to be uploaded to the identity provider are:

   a. entityID – this is the Web Bridge 3 address (i.e. https://<domain>:port). This address must be a valid Web Bridge 3 address reachable from the browsers of web app users.

      **Note:** If there are multiple Web Bridge 3s in a deployment this should be a load-balanced address.

   b. An HTTP-POST AssertionConsumerService for the Web Bridge address defined as the entityId following the format "https://<domain>:<port>/api/auth/sso/idpResponse".

   c. Optional. A public key for signing with which the identity provider will verify AuthnRequest signatures.

   d. Optional. A public key for encryption with which the identity provider will encrypt information sent back to one of the Web Bridge 3s routable through the address provided above.

   **Note:** Meeting Server requires that messages sent to it are signed by the identity provider on the Response and/or Assertion level. Unsigned communication will be discarded.

2. You need to configure a custom parameter passed from your identity provider with a successful response. For each user its contents should match the value already configured as authenticationId for that Meeting Server user (e.g. $sAMAccountName$). Usually identity providers will have a special form or dialog for that as part of creating the Service Provider entry. This parameter can be any name of your choosing, although we recommend you choose something easy to remember, such as "uid" (you will need the name in Task 3).

*Task 3: Creating SSO archive zip file*

1. To configure the Meeting Server, you need to create an archive zip file named sso_<name>.zip for each SSO you want to configure for the Web Bridge 3 on that Meeting Server. The file name must start with "sso_" followed by a meaningful name of your choice.

   Create a zip archive file containing these files:

   a. idp_config.xml – This is a file that the administrator will receive from the identity provider.

   b. config.json – includes:

      - supportedDomains (array of strings) – a list of all domains for Meeting Server users which will be authenticated against this identity provider. I.e. using the examples from Task 1, supportedDomains would contain the single entry of "example.com".
      - authenticationIdMapping (string) – name of the parameter from the identity provider responses configured as part of Task 2 (e.g. "uid") that matches to the authenticationIds in the Meeting Server. Web app users for SSO must have authenticationIds setup for them(see Task 1.)
      - ssoServiceProviderAddress (string) – the address on which the identity providers will send the responses, this will match the Web Bridge 3 specified in the entityID in Task 2.

   c. Optional. sso_sign.key – private key for the public signing key configured on the identity provider side. It will be used to sign outgoing AuthnRequests from Meeting Server which can then be verified using the public key on the identity provider's side.

   d. Optional. sso_encrypt.key – private key for the public encryption key configured on the identity provider side. It will be used to decrypt on the Meeting Server messages encrypted with the public key on the identity provider's side.

   **Note:** You will need different named zip files for different identity providers.

2. Create an archive (zip) file containing the SSO files.

> **Note:** When you zip the files, do not zip the folder containing the SSO files. If this is done, this will create an extra layer of folder (zipped file > folder > SSO files). Instead, highlight the SSO 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 SSO files without creating an extra layer of folder (e.g. zipped file > SSO files).

*Task 4: Uploading the SSO archive zip*

The SSO archive zip now needs to be uploaded and hosted on the local Web Bridge 3.

**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. For each Meeting Server with an enabled Web Bridge 3 which will locally host this zip archive:
2. 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 **sso_<name>.zip**. For example:
      ```
      PUT sso_<name>.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 3
      ```
      webbridge3 restart
      ```
3. The new SSO archive file will be picked up after the restart.

**Note:** Once a web app user is logged in they will have a separate session on the web app application from the one with the identity provider. This means that if they logout/sign out from the web app application but not from the identity provider once they enter the same username they will automatically be allowed into the web app application again. However, if they sign out from the identity provider it doesn't sign them out from the web app application and they will have to also sign out from the web app application. To ensure that you cannot log in for this browser session again you must sign out from both the web app application and the identity provider.

### 2.7.2 Example 1 config.json file

This is an example config.json file:

```
{
    "authenticationIdMapping" : "<parameter_from_task_2>",
    "ssoServiceProviderAddress" : "https://<domain>:<port>",
    "supportedDomains" : ["<domain1>","<domain2>"]
}
```

### 2.7.3 Example 2 Simple service provider metadata file.

This is an example simple service provider metadata file – note that administrators will have to modify <domain> and <port> with their relevant values.

```
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
ID="https://<domain>:<port>" entityID="https://<domain>:<port>">
  <md:SPSSODescriptor AuthnRequestsSigned="false" WantAssertionsSigned="true"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-
format:transient</md:NameIDFormat>
    <md:AssertionConsumerService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://<domain>:<port>/api/auth/sso/idpResponse" index="0"/>
  </md:SPSSODescriptor>
</md:EntityDescriptor>
```

### 2.7.4 Example 3 Comprehensive service provider metadata file.

This is a comprehensive metadata file example which includes an xml for the signing and encryption keys.

---

Note: The keys should be placed in the X509Certificate sub-elements of their corresponding KeyDescriptor elements according to the use parameter ("encryption" or "signing"). You must substitute "..." with the text contents of the key (e.g. ds:X509CertificateMIID**<omitted_key_text>**+gb</ds:X509Certificate> )

---

Note: If you include a signing certificate, the value AuthnRequestsSigned is set to "true" (it is set to "false" in the simpler metadata file in example 2).

---

```
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
ID="https://<domain>:<port>" entityID="https://<domain>:<port>">
  <md:SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="true"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:KeyDescriptor use="signing">
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>...</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:KeyDescriptor use="encryption">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>...</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-
format:transient</md:NameIDFormat>
    <md:AssertionConsumerService
```

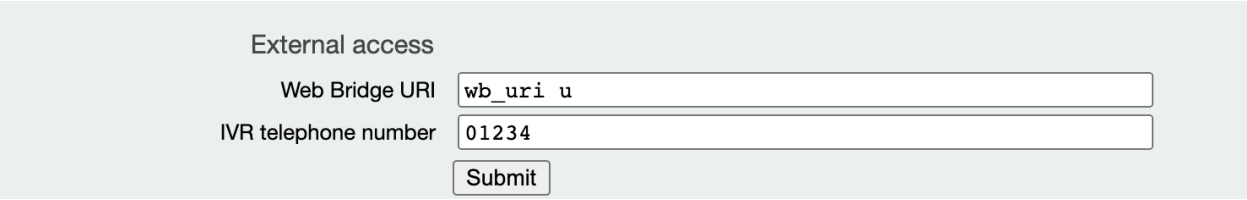
```
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://<domain>:<port>/api/auth/sso/idpResponse" index="0"/>
  </md:SPSSODescriptor>
</md:EntityDescriptor>
```

## 2.8 Web Admin user interface changes

Version 3.1 introduces a change to the Web Admin user interface for external access configuration.

**External access** on the **Configuration > General** page in the Web Admin user interface that allowed configuration of the **Web Bridge URI** and **IVR telephone number** is removed. These configuration fields are now moved to web bridge profiles.

**Figure 2: External access configuration settings removed from 3.1 Web Admin user interface**




The fields under **External access** are dealt with as follows in 3.1:

- **Web Bridge URI**: moved to webBridgeProfiles API
- **IVR telephone number**: moved to webBridgeProfiles API

You can also now specify multiple IVR numbers and Web Bridge addresses — up to 32 IVR numbers and up to 32 Web Bridge addresses per Web Bridge profile. These are used when displaying join information, and for generating email invitations.

---

**Note:** We strongly recommend that you use a webBridgeProfile at the system level or at the tenant level (if you are using multi-tenancy) for configuring the ivrNumbers and webBridgeAddresses.

---

### 2.8.1 API additions and changes

The following new API objects are introduced in version 3.1 to support these feature changes:

- **`/webBridgeProfiles/<web bridge profile id>/ivrNumbers`**
- **`/webBridgeProfiles/<web bridge profile id>/ivrNumbers/<ivr number id>`**
- **`/webBridgeProfiles/<web bridge profile id>/webBridgeAddresses`**
- **`/webBridgeProfiles/<web bridge profile id>/webBridgeAddresses/<web bridge address id>`**

Each of these objects supports **`label`** and **`number`** parameters in the form of a string.

The following new API error code reasons are introduced in version 3.1 to support these feature changes:

- **webBridgeAddressDoesNotExist** – You tried to modify, remove, or retrieve a web bridge address using an ID that did not correspond to a valid web bridge address.
- **ivrNumberDoesNotExist** – You tried to modify, remove, or retrieve an IVR number using an ID that did not correspond to a valid IVR number.
- **maxNumberOfWebBridgeAddressesReached** – You tried to add a new web bridge address for a web bridge profile that already had the maximum number of allowed entries defined. Please remove one to be able to add another one.
- **maxNumberOfIvrNumbersReached** – You tried to add a new IVR number for a web bridge profile that already had the maximum number of allowed entries defined. Please remove one to be able to add another one.

The following response values are deprecated in the **/accessQuery** response

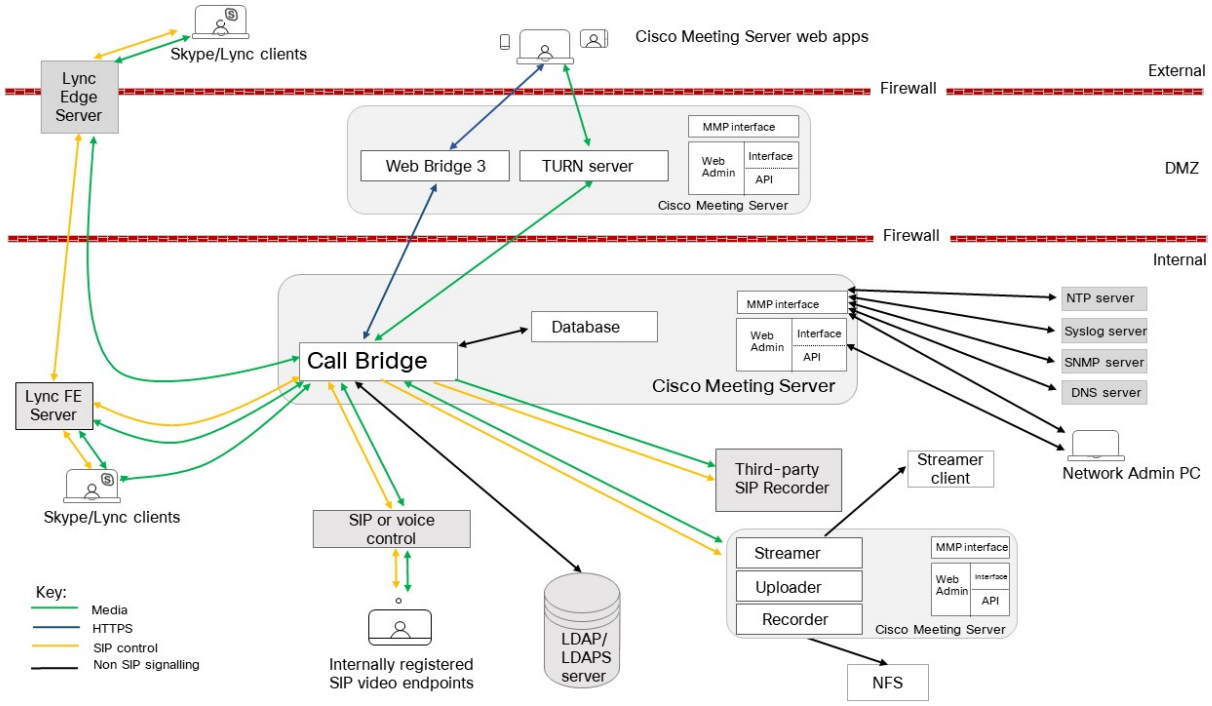
- **webAddress**
- **ivr**

## 2.9 Cisco Meeting Server web edge solution to increase scale

Expressway (Large OVA or CE1200) is the recommended solution for deployments with small to medium web app scale requirements (i.e. 800 calls or less). However, for deployments that need larger web app scale, from version 3.1 we recommend Cisco Meeting Server web edge as the required solution.

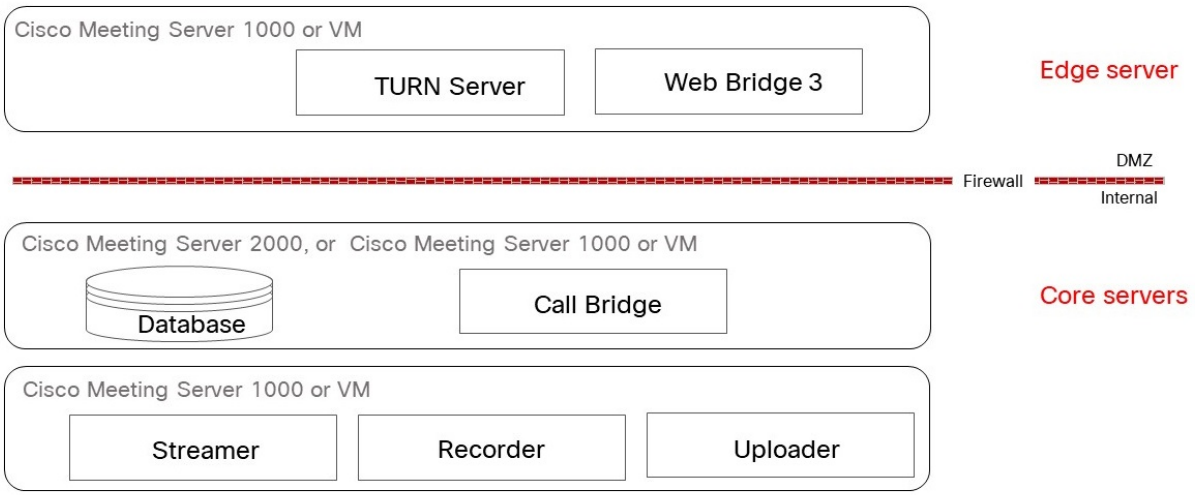
With the greater demand for remote working driving the need for increased web app scale, Cisco Meeting Server version 3.1 has been developed and tested to provide edge support for this increased web app scale. Figure 3 shows an example of how you can deploy the Meeting Server web edge solution to optimize your deployment for larger web app scale.

Figure 3: Example of a Meeting Server web edge deployment using the TURN server component in a split server deployment




When used as an edge server, the Meeting Server uses its existing TURN server and web app components (and not the Call Bridge component), as shown in Figure 4.

Figure 4: Split server deployment with Cisco Meeting Server providing TURN services




**Note:** Not all of these components need to be configured, you only need to configure the components that are appropriate to your deployment.

**Note:** Both the core and edge server must run the same version of software.

To support the Meeting Server web edge solution, a new MMP command **`turn high-capacity-mode (enable|disable)`** is introduced that enables TURN scalability mode. It is enabled by default.

We recommend that you use one of two server specifications to run your Cisco Meeting Server web edge, as detailed in Section 2.9.2. The call capacities that can be achieved using the server specifications with recommended hardware are shown in Table 8.

**Table 8: Call capacities for server specifications with recommended hardware**

| Type of calls | 1 x 4 vCPU VM call capacity | 1 x 16 vCPU VM call capacity |
|---|---|---|
| Full HD calls<br>1080p30 video | 100 | 350 |
| HD calls<br>720p30 video | 175 | 700 |
| SD calls<br>448p30 video | 250 | 1000 |
| Audio calls (G.711) | 850 | 3000 |

### 2.9.1 Important points to note:

- For web app to match SIP scale (up to 24 Call Bridge s per cluster), we support multiple edge servers. However, Call Bridge groups only support up to 10 Edge servers per group. For deployments needing more than 10 Edge servers, more than one Call Bridge group will be necessary.

  **Note:** Deployments requiring more than 8 edge servers must be reviewed by the BU.

- We recommend that all edge servers are the same capacity, i.e. all 4 vCPUs or all 16 vCPUs, not a mix of both.
- We recommend that you configure Call Bridge groups. This allows you to assign a unique group of TURN servers to each Call Bridge group which is useful for:
  - helping with load balancing
  - keeping TURN servers sensibly geolocated with Call Bridges
- Scaling edge servers – we recommend that "core Call Bridge" to "edge VM" ratios can be any of the following: many:1, 1:1, or 1:many.
- We recommend 1 vCPU to 1 physical CPU.

- Co-residency support: the edge server can be co-resident with other VMs. However, each 4 vCPU VM has a 1 Gbps NIC requirement and each 16 vCPU has a 10Gbps NIC requirement. The VM host will need sufficient NIC capacity for all applications.

  **Note:** Meeting Server 1000 M4 hardware supports 1Gbps NIC, M5 onwards hardware supports 10Gbps NIC.

- We recommend processor specification such as Intel Xeon E5 2600 running at 2.5GHz or higher.

### 2.9.2 Recommended Meeting Server web edge server specifications

**Server specification A**

- 1 x Cisco Meeting Server VM with the following specification for supported Cisco hardware; 4 GB RAM, 4 vCPUs, 1Gbps network interface.
  - Recommend using:
    - 1 x Meeting Server VM per Cisco Meeting Server 1000.
      or
    - 4 x Meeting Server VMs per Cisco Meeting Server 2000.

**Server specification B**

- 1 Cisco Meeting Server VM with the following specification for supported Cisco hardware, 8 GB RAM, 16 vCPUs, 10Gbps network interface .
  - Recommend using:
    - 1 x Meeting Server VM can serve up to 4 x Cisco Meeting Server 1000s.
      or
    - 1 x Meeting Server VM can serve 1 x Cisco Meeting Server 2000

### 2.9.3 Deploying Meeting Server web edge

Comprehensive information on configuring Meeting Server in a Split deployment, can be found in the Deployment Guide here. The following steps give a high level view of how to deploy Meeting Server web edge:

1. Configure the TURN server on the Meeting Server edge via the MMP.
2. Configure Web Bridge 3 on the Meeting Server edge via the MMP.
3. Link the Web Bridge 3 to the Call Bridge, (i.e. add the **callBridge** parameter via the Web Admin user interface under **Configuration > API** to **/api/v1/turnServers** and **/api/v1/webBridges**, and check Web Bridge 3 certificate requirements).

4. Check that connections are functioning correctly – to do this, you can test manually by logging in via the web app address, or check on the Web Admin interface under **Status > General** and look at the **Fault conditions**, and **Recent errors and warnings**. (Note that Web Bridge 3/TURN connection failure messages aren't shown.)
5. Add the firewall settings as follows:
    a. You need the TCP connection Web Bridge 3 c2w connection ports opened (as specified on the "c2w://address:port" in the API; i.e. in the **url** field on **/api/v1/webBridges**.)
    b. You must be able to establish a connection on TCP 3478 on the Meeting Server edge from the Call Bridge (i.e. able to communicate with the TURN server component).
    c. TURN relay ports on the Meeting Server edge are 50000 to 62000, so Call Bridge and external connections must be able to contact those on UDP to send media.

## 2.10 Short-term credentials for Cisco Meeting Server edge

To enhance security, 3.1 introduced short term credentials for the Cisco Meeting Server edge. When 3.1 was originally released, this was a beta feature due to limited solution testing. Testing is now completed, and the feature is fully supported. Therefore, the "beta feature" caveat has been removed. This feature is optional and when enabled, each credential set is valid for 24 hours.

By default the Meeting Server TURN server component will continue to use long-term credentials. You only need to use the new MMP commands and API parameters detailed below if you wish to try the short-term credentials feature.

**Note:** The TURN server component always supports the standard port 3478 for UDP.

### 2.10.1 MMP Additions

This feature introduces the following new MMP commands:

**`turn short_term_credentials_mode (enable|disable)`** – toggles the TURN server between short- and long-term credential mode. Default is **`disable`**.

**`turn short_term_credentials <shared secret> <realm>`** – Specifies the shared secret and realm required by the TURN server to use short-term credentials.

### 2.10.2 API Changes

The new parameters **`useShortTermCredentials`** and **`sharedSecret`** are added to the **`/turnServers`** object.

- **useShortTermCredentials** – true | false: whether or not short term credentials should be used on this TURN server. If this parameter is not supplied in a create (POST) operation, it defaults to "false".
- **sharedSecret** – is the shared secret (string) that should be used when making allocations on this TURN server (when short term credential mode is enabled)

#### *2.10.2.1 Parameter updates*

The existing **username** and **password** parameters on **/turnServers** now only apply when short term credentials mode is disabled.

### 2.10.3 Implementing short term credentials on the Meeting Server

These steps assume you have already upgraded to version 3.1.

---

Note: You can reverse Tasks 1 and 2 and perform the API configuration prior to the MMP steps, however, the sharedSecret must be the same in both places.

---

Task 1: Enabling and configuring short term credentials via the MMP

1. SSH into the MMP and login.
2. Enter **turn short_term_credentials_mode enable** to enable short term credentials mode.
3. Enter **turn short_term_credentials <shared secret> <realm>** to set the desired shared secret and realm. For example: **turn short_term_credentials mysharedsecret example.com**

Task 2: Configuring the TURN server to use short term credentials via the API

To configure the short term credentials for a TURN server using the Meeting Server Web Admin interface:

4. Log in to the Meeting Server Web Admin interface and select **Configuration > API**:
5. From the list of API objects, tap the ▶ after **/api/v1/turnServers**
6. To configure or modify an existing TURN server, either select **Create new** or the object id of the required existing TURN server and set the **useShortTermCredentials** field to **true**.
7. Enter the shared secret (as set in Step 3 of Task 1) in the **sharedSecret** field.
8. Click **Create** if configuring a new TURN server, or **Modify** if configuring an existing one.

## 2.11 Scheduled LDAP sync – API timestamp additions

Version 3.1 introduces three new timestamps to the ldapSync API object. This provides additional information for the administrator, both directly on the Meeting Server API and via Meeting Management.

The three parameters are available on GET on **/ldapSyncs/<sync id>** and are returned in the enumerated list on GET on **/ldapSyncs**. The parameters cannot be set via POST or PUT. The new parameters are:

- **creationTime** – timestamp of when the sync object was created.
- **startTime** – timestamp of when the sync operation started.
- **endTime** – timestamp of when the sync operation ended (in success or failure).

**Note:** For all three timestamps, the time is in UTC – the format / timezone specified in RFC 3339, e.g. "2014-02-11T12:10:47Z".

**Note:** Although LDAP syncs can be initiated via the Web Admin user interface, you cannot enumerate a Web Admin initiated LDAP sync in the API. Therefore Web Admin initiated LDAP syncs will not provide access to these new timestamp details.

## 2.12 Name label on ldapSources API objects

Version 3.1 allows you to configure a name label on the /ldapSources API object. This provides additional information for the administrator on the Meeting Server API, and also improves the user experience on Meeting Management.

Meeting Management provides an interface that requires the administrator to select a user import to match with a template. From 3.1, if a name label has been configured, it's easier to identify the user import required from the name labels displayed in the drop-down list.

This name label addition introduces the API parameter **name** on /ldapSources objects. The name of the ldapSources is optional and the default is an empty string. This parameter supports the following operations:

- POST to **/ldapSources**
- PUT on **/ldapSources/<ldap source id>**
- Enumerate of GET on **/ldapSources**
- GET on **/ldapSources/<ldap source id>**

## 2.13 Summary of 3.1 API Additions and Changes

New API functionality for the Meeting Server 3.1 includes:

- New API objects and parameters to support coSpace provisioning
- New API objects and parameters to support web app custom email invite changes (webBridgeProfiles)

### 2.13.1 API additions

The following new API objects are introduced in version 3.1:

- **/ldapUserProvisionedCoSpaceMappings**
- **/ldapUserProvisionedCoSpaceMappings/<LDAP user provisioned coSpace mapping id>**
- **/ldapUserProvisionedCoSpaceSources**
- **/ldapUserProvisionedCoSpaceSources/<LDAP user provisioned coSpace mapping id>**
- **/ldapUserProvisionedCoSpaceSources/<LDAP user provisioned coSpace source id>**
- **/users/<user id>/userProvisionedCoSpaces**
- **/users/<user id>/userProvisionedCoSpaces/<user provisioned coSpace id>**
- **/webBridgeProfiles/<web bridge profile id>/ivrNumbers**
- **/webBridgeProfiles/<web bridge profile id>/ivrNumbers/<ivr number id>**
- **/webBridgeProfiles/<web bridge profile id>/webBridgeAddresses**
- **/webBridgeProfiles/<web bridge profile id>/webBridgeAddresses/<web bridge address id>**

The following new API error code reasons are introduced in version 3.1:

- **ldapUserProvisionedCoSpaceMappingDoesNotExist** — You tried to modify or remove an LDAP user provisioned coSpace mapping using an ID that did not correspond to a valid LDAP user provisioned coSpace mapping.
- **userProvisionedCoSpaceDoesNotExist** — You tried to modify, remove or retrieve a user provisioned coSpace using an ID that did not correspond to a valid user provisioned coSpace for that user.
- **ldapUserProvisionedCoSpaceSourceDoesNotExist** — You tried to modify or remove an LDAP user provisioned coSpace source using an ID that did not correspond to a valid LDAP user provisioned coSpace source.
- **webBridgeAddressDoesNotExist** — You tried to modify, remove, or retrieve a web bridge address using an ID that did not correspond to a valid web bridge address.

- **ivrNumberDoesNotExist** – You tried to modify, remove, or retrieve an IVR number using an ID that did not correspond to a valid IVR number.
- **maxNumberOfWebBridgeAddressesReached** – You tried to add a new web bridge address for a web bridge profile that already had the maximum number of allowed entries defined. Please remove one to be able to add another one.
- **maxNumberOfIvrNumbersReached** – You tried to add a new IVR number for a web bridge profile that already had the maximum number of allowed entries defined. Please remove one to be able to add another one.

New API types supported:

- on request parameter **scope** for POST to **/coSpaces/<coSpace id>/accessMethods** and PUT to **/coSpaces/<coSpace id>/accessMethods/<access method id>**:
  - **member** – details of this coSpace access method are visible to members of the coSpace
  - **directory** – details of this coSpace access method can be found through search [Note: in 3.1 there is no search, so behavior is the same as **public**]

  And existing API type definitions for **public** and **private** are updated:
  - **public** – details of this coSpace access method are visible to members of the coSpace and all participants in the meeting
  - **private** – details of this coSpace access method is visible only to the owner of the space in web app, or visible to admin users using the Call Bridge API.

New API request parameters supported in 3.1:

- **muteBehavior** parameter on POST to **/callProfiles**; PUT to **/callProfiles/<call profile id>**; GET on **/callProfiles/<call profile id>**
- **passthroughMode** parameter to control whether H.264 passthrough feature is allowed. Introduced for POST on **/compatibilityProfiles**; PUT to **/compatibilityProfiles/<compatibility profile id>**, and GET on **/compatibilityProfiles/<compatibility profile id>**. Values are:
  - **enabled** – allows video to avoid being transcoded when possible (default if unset)
  - **disabled** – always transcode video
- **userProvisionedCoSpace** parameter is introduced to the /coSpaces object.
- **name** parameter introduced on POST to **/ldapSources**, PUT to **/ldapSources/<ldap source id>**, Enumerate of GET on **/ldapSources**, and GET on **/ldapSources/<ldap source id>**
- **creationTime**, **startTime**, **endTime** URI parameters are introduced on GET on **/ldapSyncs/<sync id>**

- **useShortTermCredentials** added on POST to **/turnServers** and PUT to **/turnServers/<turn server id>** objects, and GET on **/turnServers/<turn server id>**.
- **sharedSecret** are added on POST to **/turnServers** and PUT to **/turnServers/<turn server id>** objects.

Updated API parameters in 3.1:

- the existing **username** and **password** parameters on **/turnServers** now only apply when short term credentials mode is disabled.

API parameter deprecations in 3.1:

The following parameters are deprecated in the /ldapmappings object:

- **coSpaceUriMapping**
- **coSpaceSecondaryUriMapping**
- **coSpaceNameMapping**
- **coSpaceCallIdMapping**

The following parameter is deprecated in the **/system/status** response:

- **activated**

The following parameters are deprecated in the **/system/multipartyLicensing** response:

- **personalLicenseLimit**
- **sharedLicenseLimit**
- **capacityUnitLimit**

The following response values are deprecated in the **/accessQuery** response:

- **webAddress**
- **ivr**

The **resolveLyncConferenceIds** parameter that was visible but non-functional in 3.0 is now removed from the following APIs: **/webBridgeProfiles/<webBridge profile id>**, **/system/profiles/effectiveWebBridgeProfile**, **/tenant/<tenant id>/effectiveWebBridgeProfile**, and **/webBridges/<web bridge id>/effectiveWebBridgeProfile**.

The following API error code reason is removed in version 3.1:

- **messageDoesNotExist**

### 2.13.2 Creating, modifying, and retrieving LDAP user provisioned coSpace mappings

This new API object supports the following operations:

- POST to **`/ldapUserProvisionedCospaceMappings`**
- PUT to **`/ldapUserProvisionedCospaceMappings/<LDAP user provisioned coSpace mapping id>`**

| Parameter | Type/Value | Description/Notes |
|---|---|---|
| coSpaceUriMapping (*) | string | similar to the mappings of the ldapMappings object, template for generating URI of the user provisioned coSpaces. (from version 3.1) |
| coSpaceNameMapping | string | similar to the mappings of the ldapMappings object, template for generating the name of the user provisioned coSpaces. (from version 3.1) |
| coSpaceTemplate (*) | ID | coSpace template to use for the user provisioned coSpaces. (from version 3.1) |

- Enumeration of **`/ldapUserProvisionedCospaceMappings`** accepts the following URI parameters:

| URI parameters | Type/Value | Description/Notes |
|---|---|---|
| offset | | an offset and limit can be supplied to retrieve entries other than those in the first page in the notional list |
| limit | | |

Response is structured as a top-level <ldapUserProvisionedCospaceMappings total="N"> tag with potentially multiple <ldapUserProvisionedCospaceMapping> elements within it.

Each <ldapUserProvisionedCospaceMapping> tag may include the following elements:

| Response elements | Type/Value | Description/Notes |
|---|---|---|
| coSpaceUriMapping | string | similar to the mappings of the ldapMappings object, template for generating URI of the user provisioned coSpaces. (from version 3.1) |
| coSpaceNameMapping | string | similar to the mappings of the ldapMappings object, template for generating name of the user provisioned coSpaces. (from version 3.1) |
| coSpaceTemplate | ID | coSpace template used for the user provisioned coSpaces. (from version 3.1) |

- GET on individual LDAP user provisioned coSpace mappings with

**/ldapUserProvisionedCospaceMappings/<LDAP user provisioned coSpace mapping id>** gives the following response:

| Response elements | Type/Value | Description/Notes |
|---|---|---|
| coSpaceUriMapping | string | similar to the mappings of the ldapMappings object, template for generating URI of the user provisioned coSpaces. (from version 3.1) |
| coSpaceNameMapping | string | similar to the mappings of the ldapMappings object, template for generating name of the user provisioned coSpaces. (from version 3.1) |
| coSpaceTemplate | ID | coSpace template used for the user provisioned coSpaces. (from version 3.1) |

### 2.13.3 Creating, modifying, and retrieving LDAP user provisioned coSpace sources

This new API object supports the following operations:

- POST to **/ldapUserProvisionedCoSpaceSources**
- PUT to **/ldapUserProvisionedCoSpaceSources/<LDAP user provisioned coSpace source id>**

| Parameter | Type/Value | Description/Notes |
|---|---|---|
| ldapSource (*) | ID | ID of the LDAP source to be used to locate users (from version 3.1) |
| ldapUserProvisionedCoSpaceMapping (*) | ID | Mapping to be used to generate the name and uri-hint of the user provisioned coSpaces (from version 3.1) |
| filter | string | Additional LDAP filter string to be applied when reading the source (from version 3.1) |

- Enumeration of **/ldapUserProvisionedCoSpaceSources** accepts the following URI parameters:

| URI parameters | Type/Value | Description/Notes |
|---|---|---|
| offset | | an offset and limit can be supplied to retrieve entries other than those in the first page of the notional list (from version 3.1) |
| limit | | |

Response is structured as a top-level <ldapUserProvisionedCoSpaceSources total="N"> tag with potentially multiple <ldapUserProvisionedCoSpaceSource> elements within it.

Each <ldapUserProvisionedCoSpaceSource> tag may include the following elements:

| Response elements | Type/Value | Description/Notes |
|---|---|---|
| ldapSource | ID | ID of the LDAP source used to locate users (from version 3.1) |
| ldapUserProvisionedCoSpaceMapping | ID | Mapping used to generate the name and uri-hint of the user provisioned coSpaces (from version 3.1) |
| filter | string | Additional LDAP filter string applied when reading the source (from version 3.1) |

- GET on individual LDAP user provisioned coSpace sources with **`/ldapUserProvisionedCoSpaceSources/<LDAP user provisioned coSpace mapping id>`** gives the following response:

| Response elements | Type/Value | Description/Notes |
|---|---|---|
| ldapSource | ID | ID of the LDAP source used to locate users (from version 3.1) |
| ldapUserProvisionedCoSpaceMapping | ID | Mapping used to generate the name and uri-hint of the user provisioned coSpaces (from version 3.1) |
| filter | string | Additional LDAP filter string applied when reading the source (from version 3.1) |

### 2.13.4 Retrieving user provisioned coSpace information

Version 3.1 introduces this API object to support the following operations:

- Enumeration of GET on **`/users/<user id>/userProvisionedCoSpaces`**
- GET on **`/users/<user id>/userProvisionedCoSpaces/<user provisioned coSpace id>`**

Enumeration of **`/users/<user id>/userProvisionedCoSpaces`** accepts the following URI parameters:

| URI parameters | Type/Value | Description/Notes |
|---|---|---|
| offset | | An offset and limit can be supplied to retrieve user provisioned coSpaces other than those in the first page of the notional list. (from version 3.1) |
| limit | | |

Response is structured as a top-level <userProvisionedCoSpaces total="N"> tag with potentially multiple <userProvisionedCoSpace> elements within it.

Each <userProvisionedCoSpace> tag may include the following elements:

| Response elements | Type/Value | Description/Notes |
|---|---|---|
| coSpaceTemplate | ID | coSpaceTemplate that this coSpace will be based on when it is instantiated. (from version 3.1) |
| uriHint | string | Basis for the uri of this coSpace (if this clashes with other uris when the space is instantiated, a unique uri will be generated based on this hint). (from version 3.1) |
| name | string | name that this coSpace will have when it is instantiated. (from version 3.1) |
| coSpace | ID | If present, ID of the coSpace this userProvisionedCoSpace was instantiated into. (from version 3.1) |

GET on individual user provisioned coSpace with **`/users/<user id>/userProvisionedCoSpaces/<user provisioned coSpace id>`** gives the following response:

| Response elements | Type/Value | Description/Notes |
|---|---|---|
| coSpaceTemplate | ID | coSpaceTemplate that this coSpace is based on when it is instantiated. (from version 3.1) |
| uriHint | string | Basis for the uri of this coSpace (if this clashes with other uris when the space is instantiated, a unique uri will be generated based on this hint). (from version 3.1) |
| name | string | name that this coSpace has when it is instantiated. (from version 3.1) |
| coSpace | ID | If present, ID of the coSpace this userProvisionedCoSpace was instantiated into. (from version 3.1) |

#### *2.13.4.1 Instantiating a new coSpace from a user provisioned coSpace*

The new API parameter **`userProvisionedCoSpace`** is introduced with the type "ID" to instantiate a new coSpace from the user provisioned coSpace. When this parameter is present all the other parameters are ignored. It's introduced on the following operations:

- POST to **`/coSpaces`**
- PUT to **`/coSpaces/<coSpace id>`**

### 2.13.5 Creating, modifying, and retrieving Web Bridge addresses for a webBridgeProfile

This new API object supports the following operations:

- POST to **`/webBridgeProfiles/<web bridge profile id>/webBridgeAddresses`**
- PUT to **`/webBridgeProfiles/<web bridge profile id>/webBridgeAddresses/<web bridge address id>`**

| Parameter | Type/Value | Description/Notes |
|---|---|---|
| label | string | Label name that describes this web bridge address. Example: **`USA web app`** (from version 3.1) |
| address | url | Address to use when rendering email invites. Example: **`https://usa.mycompany.com/`** (from version 3.1) |

- Enumeration of **`/webBridgeProfiles/<web bridge profile id>/webBridgeAddresses`** accepts the following URI parameters:

| URI parameters | Type/Value | Description/Notes |
|---|---|---|
| offset | | an offset and limit can be supplied to retrieve web bridge addresses other than those in the first page in the notional list |
| limit | | |

Response is structured as a top-level <webBridgeAddresses total="N"> tag with potentially multiple <webBridgeAddress> elements within it.

Each <webBridgeAddress> tag may include the following elements:

| Response elements | Type/Value | Description/Notes |
|---|---|---|
| label | string | Label name that describes this web bridge address. Example: **`USA web app`** (from version 3.1) |

- GET on individual web bridge addresses on webBridgeProfiles with **`/webBridgeProfiles/<web bridge profile id>/webBridgeAddresses/<web bridge address id>`** gives the following response:

| Response elements | Type/Value | Description/Notes |
|---|---|---|
| label | string | Label name that describes this web bridge address. Example: **`USA web app`** (from version 3.1) |
| address | url | Address used when rendering email invites. Example: **`https://usa.mycompany.com/`** (from version 3.1) |

### 2.13.6 Creating, modifying, and retrieving IVR numbers for a webBridgeProfile

This new API object supports the following operations:

- POST to **`/webBridgeProfiles/<web bridge profile id>/ivrNumbers`**
- PUT to **`/webBridgeProfiles/<web bridge profile id>/ivrNumbers/<ivr number id>`**

| Parameter | Type/Value | Description/Notes |
|---|---|---|
| label | string | Label name that describes this IVR number. Example: **`USA Call-in Number`** (from version 3.1) |
| number | string | IVR number to use when rendering email invites. Example: **`888-123123`** (from version 3.1) |

- Enumeration of **`/webBridgeProfiles/<web bridge profile id>/ivrNumbers`** accepts the following URI parameters:

| URI parameters | Type/Value | Description/Notes |
|---|---|---|
| offset | | an offset and limit can be supplied to retrieve IVR numbers other than those in the first page in the notional list |
| limit | | |

Response is structured as a top-level <ivrNumbers total="N"> tag with potentially multiple <ivrNumber> elements within it.

Each <ivrNumber> tag may include the following elements:

| Response elements | Type/Value | Description/Notes |
|---|---|---|
| label | string | Label name that describes this IVR number. Example: **`USA Call-in Number`** (from version 3.1) |

- GET on individual IVR numbers on webBridgeProfiles with **`/webBridgeProfiles/<web bridge profile id>/ivrNumbers/<ivr number id>`** gives the following response:

| Response elements | Type/Value | Description/Notes |
|---|---|---|
| label | string | Label name that describes this IVR number. Example: **`USA Call-in Number`** (from version 3.1) |
| number | string | IVR number used when rendering email invites. Example: **`888-123123`** (from version 3.1) |

### 2.13.7 Setting the mute behavior of a call

This feature introduces the parameter **`muteBehavior`** for a call on the following operations:

- POST to **`/callProfiles`**
- PUT to **`/callProfiles/<call profile id>`**

| Parameter | Type/Value | Description/Notes |
|---|---|---|
| muteBehavior | linked \| separate | Defines the mute behavior of the call:<br>• linked – in this mode, when a user's call is muted on the Meeting Server, their endpoint or web app session may also automatically perform a local mute of their device; this means it is not possible to reverse the effect of the Meeting Server mute with just another API command - the user themselves must unmute their device.<br>• separate – in this mode, the mute statuses of a user's call on the Meeting Server and on their local device are independent of one another, meaning that other users/admins can video/audio mute/unmute all participants.<br>If this parameter is not supplied in a create (POST) operation, it defaults to "linked". (from version 3.1) |

- GET on **`/callProfiles/<call profile id>`**.
  Response is structured as a top-level <callProfiles total="N"> tag with potentially multiple <callProfile> elements within it.

  Each <callProfiles> tag may include the following elements:

| Response elements | Type/Value | Description/Notes |
|---|---|---|
| muteBehavior | linked \| separate | The mute behavior of the call:<br>• linked – in this mode, when a user's call is muted on the Meeting Server, their endpoint or web app session may also automatically perform a local mute of their device; this means it is not possible to reverse the effect of the Meeting Server mute with just another API command - the user themselves must unmute their device.<br>• separate – in this mode, the mute statuses of a user's call on the Meeting Server and on their local device are independent of one another, meaning that other users/admins can video/audio mute/unmute all participants.<br>If this parameter is not supplied in a create (POST) operation, it defaults to "linked". (from version 3.1) |

### 2.13.8 Configuring short term credentials for Cisco Meeting Server edge

This feature introduces two new parameters **`useShortTermCredentials`** and **`sharedSecret`** on these operations as follows:

- POST to **`/turnServers`**
- PUT to **`/turnServers/<turn server id>`**

| Parameter | Type/Value | Description/Notes |
|---|---|---|
| useShortTermCredentials | true \| false | Whether or not short term credentials should be used on this TURN server.<br>If this parameter is not supplied in a create (POST) operation, it defaults to "false". (from version 3.1) |
| sharedSecret | string | The shared secret that should be used when making allocations on this TURN server (when short term credential mode is enabled). (from version 3.1) |

**Note:** Short-term TURN credentials are never accessible via an API GET operation to /turnServers, nor is the shared secret.

- GET on individual TURN Servers with **`turnServers/<turn server id>`** gives the following response:

| Response elements | Type/Value | Description/Notes |
|---|---|---|
| useShortTermCredentials | true \| false | Whether or not short term credentials are used on this TURN server. (from version 3.1) |

## 2.14 Summary of MMP additions and changes

Version 3.1 supports these MMP changes:

### 2.14.1 RTMPS streaming

To support RTMPS streaming, the existing **`tls`** MMP command is extended to optionally allow configuration of TLS trusts for RTMPS. So MMP commands that feature **`tls <service>`** in the command line will now support **`tls rtmps`**.

### 2.14.2 Packet capture improvements

The following MMP command options are introduced in version 3.1.

Table 9: Version 3.1 pcap MMP command additions

| Command | Description |
|---|---|
| **pcap (a\|b\|c\|d\|any) [snaplen <n>] [filter <pcap-filter-expression>]** | **any** will allow packet capture on multiple interfaces, i.e. any enabled interfaces (interfaces that are not enabled will be skipped).<br>Note: When capturing from multiple interfaces, this requires additional disk space as each interface is captured to a separate temporary file and the files are then merged when the capture is stopped. So the available storage when capture on multiple interfaces is half what is available when capturing on a single interface.<br>**snaplen** truncates each packet captured to the maximum number (n) of bytes if it is longer. As a result, more packets can fit into the same file-size limit.<br>**filter** selects only packets matching the criteria in the string. This reduces the capture to only packets of interest, and avoids wasting disk space on the others. The parsing of this string and the packet filtering are performed with exactly the same underlying libraries as used by tcpdump, so this has exactly the same expressive power and performance. The filter expression can be up to around 4080 characters long, if required<br>**snaplen** and **filter** options added from version 3.1. |

Note: Although packets can be captured by the Meeting Server, due to the high packet rate that the Meeting Server operates at, packets may be dropped from the packet capture rather than disturb the normal operation of the Meeting Server in handling calls. To avoid dropped packets in the packet capture, Cisco recommends capturing packets at your network switch rather than on the Meeting Server.

### 2.14.3 Cisco Meeting Server web edge solution to increase web app scale

The following MMP command option is introduced in version 3.1.

Table 10: Version 3.1 Meeting Serverweb edge solution to increase scale MMP command additions

| Command | Description |
|---|---|
| **turn high-capacity-mode (enable\|disable)** | Implements support for increased web app scale (default enable) on the Meeting Server running TURN and web app – it allows higher packet throughput when using Meeting Server for web edge. Only disable if advised to do so by Cisco Support. (from version 3.1) |

### 2.14.4 Short term credentials for Cisco Meeting Server edge

The following MMP command options are introduced in version 3.1.

Table 11: Version 3.1 short term credentials for Meeting Server edge MMP command additions

| Command | Description |
|---|---|
| `turn short_term_credentials_mode (enable\|disable)` | Toggle the TURN server between short- and long-term credential mode. (from version 3.1) |
| `turn short_term_credentials <shared secret> <realm>` | Specifies the shared secret and realm required by the TURN server to use short-term credentials. (from version 3.1) |

## 2.15 Summary of CDR Changes

There are no CDR changes for version 3.1.

## 2.16 Summary of Event Changes

There are no new Events for version 3.1.

# 3 Upgrading, downgrading and deploying Cisco Meeting Server software version 3.1.3

This section assumes that you are upgrading from Cisco Meeting Server software version 3.0. If you are upgrading from an earlier version, then Cisco recommends that you upgrade to 3.0 first following the instructions in the 3.0.x release notes, before following any instructions in these Cisco Meeting Server 3.1.x Release Notes. This is particularly important if you have a Cisco Expressway connected to the Meeting Server.

**Note:** Cisco has not tested upgrading from a software release earlier than 3.0.

To check which version of Cisco Meeting Server software is installed on a Cisco Meeting Server 2000, Cisco Meeting Server 1000, or previously configured VM deployment, use the MMP command **`version`**.

If you are configuring a VM for the first time then follow the instructions in the Cisco Meeting Server Installation Guide for Virtualized Deployments.

## 3.1 Upgrading to Release 3.1.3

The instructions in this section apply to Meeting Server deployments which are not clustered. For deployments with clustered databases read the instructions in this FAQ, before upgrading clustered servers.

**CAUTION:** Before upgrading or downgrading Meeting Server you must take a configuration backup using the **`backup snapshot <filename>`** command and save the backup file safely on a different device. See the MMP Command Reference document for full details. Do **not** rely on the automatic backup file generated by the upgrade/downgrade process as it may be inaccessible in the event of a failed upgrade/downgrade.

Upgrading the firmware is a two-stage process: first, upload the upgraded firmware image; then issue the upgrade command. This restarts the server: the restart process interrupts all active calls running on the server; therefore, this stage should be done at a suitable time so as not to impact users – or users should be warned in advance.

**Note:**
Meeting Server 3.0 introduced a mandatory requirement to have Cisco Meeting Management 3.0 (or later). Meeting Management handles the product registration and interaction with your Smart Account (if set up) for Smart Licensing support.

To install the latest firmware on the server follow these steps:

1. Obtain the appropriate upgrade file from the software download pages of the Cisco website:

   **`Cisco_Meeting_Server_3_1_3_CMS2000.zip`**

   This file requires unzipping to a single upgrade.img file before uploading to the server. Use this file to upgrade Cisco Meeting Server 2000 servers.

   Hash (SHA-256) for upgrade.img file:
   9523ed195a5153c7523154093c55be50fb059db0b0a1338f0bbb7cf7160b04b6

   **`Cisco_Meeting_Server_3_1_3_vm-upgrade.zip`**

   This file requires unzipping to a single upgrade.img file before uploading to the server. Use this file to upgrade a Cisco Meeting Server virtual machine deployment.

   Hash (SHA-256) for upgrade.img file:
   65ea7fd412e00c88f637f2fc877787ccc8ae54b5cd0f4a5ce6ffc858c2dffa54

   **`Cisco_Meeting_Server_3_1_3.ova`**

   Use this file to deploy a new virtual machine via VMware.

   For vSphere6, hash (SHA-512) for Cisco_Meeting_Server_3_1_3_vSphere-6_0.ova file:
   b0d99dd06c33a67ac3abaeb720937913f0f69d88298cdf0228fbd66d931ec1a987466dfe35cee2bf17b4c2f37a65c8462316e5a4db7960f521756bdbc9eb1546

   For vSphere6.5 and higher, hash (SHA-512) for Cisco_Meeting_Server_3_1_3_vSphere-6_5.ova file:
   7a6739777e69f247e62870e8efdc55d49a020dcc13e80996178ec40830e61f9a6498c3db7037faaa8581e11a20a070d6a2ac80aedbb60f4fad485762a61e60a1

2. To validate the OVA file, the checksum for the 3.1.3 release is shown in a pop up box that appears when you hover over the description for the download. In addition, you can check the integrity of the download using the SHA-512 hash value listed above.
3. Using an SFTP client, log into the MMP using its IP address. The login credentials will be the ones set for the MMP admin account. If you are using Windows, we recommend using the WinSCP tool.

   **Note:** If you are using WinSCP for the file transfer, ensure that the Transfer Settings option is 'binary' not 'text'. Using the incorrect setting results in the transferred file being slightly smaller than the original and this prevents successful upgrade.

   **Note:**
   a) You can find the IP address of the MMP's interface with the **`iface a`** MMP command.
   b) The SFTP server runs on the standard port 22.

4. Copy the software to the Server/ virtualized server.
5. To validate the upgrade file, issue the **`upgrade list`** command.
   a. Establish an SSH connection to the MMP and log in.
   b. Output the available upgrade images and their checksums by executing the upgrade list command.

      **`upgrade list`**
   c. Check that this checksum matches the checksum shown above.
6. To apply the upgrade, use the SSH connection to the MMP from the previous step and initiate the upgrade by executing the **`upgrade`** command.
   a. Initiate the upgrade by executing the upgrade command.
      **`upgrade`**
   b. The Server/ virtualized server restarts automatically: allow 10 minutes for the process to complete.
7. Verify that the Meeting Server is running the upgraded image by re-establishing the SSH connection to the MMP and typing:
   **`version`**
8. Update the customization archive file when available.
9. If you are deploying a scaled or resilient deployment read the Scalability and Resilience Deployment Guide and plan the rest of your deployment order and configuration.
10. If you have deployed a database cluster, be sure to run the **`database cluster upgrade_schema`** command after upgrading. For instructions on upgrading the database schema refer to the Scalability and Resilience Deployment Guide.
11. You have completed the upgrade.

## 3.2 Downgrading

If anything unexpected occurs during or after the upgrade process you can return to the previous version of the Meeting Server software. Use the regular upgrade procedure to "downgrade" the Meeting Server to the required version using the MMP **`upgrade`** command.

1. Copy the software to the Server/ virtualized server.
2. To apply the downgrade, use the SSH connection to the MMP and start the downgrade by executing the **`upgrade <filename>`** command.

   The Server/ virtualized server will restart automatically – allow 10-12 minutes for the process to complete and for the Web Admin to be available after downgrading the server.
3. Log in to the Web Admin and go to **Status > General** and verify the new version is showing under **System status**.

4. Use the MMP command **`factory_reset app`** on the server and wait for it to reboot from the factory reset.
5. Restore the configuration backup for the older version, using the MMP command **`backup rollback <name>`** command.

   ---

   Note: The **`backup rollback`** command overwrites the existing configuration as well as the license.dat file and all certificates and private keys on the system, and reboots the Meeting Server. Therefore it should be used with caution. Make sure you copy your existing cms.lic file and certificates beforehand because they will be overwritten during the backup rollback process. The .JSON file will not be overwritten and does not need to be re-uploaded.

   ---

   The Meeting Server will reboot to apply the backup file.

   For a clustered deployment, repeat steps 1-5 for each node in the cluster.

6. a. In the case of XMPP clustering, if applicable, you need to re-cluster XMPP:
   a. Pick one node as the XMPP primary, initialize XMPP on this node
   b. Once the XMPP primary has been enabled, joining any other XMPP nodes to it.
   c. Providing you restore using the backup file that was created from the same server, the XMPP license files and certificates will match and continue to function.
7. Finally, check that:
   - the Web Admin interface on each Call Bridge can display the list of coSpaces.
   - dial plans are intact,
   - XMPP service is connected, if applicable,
   - no fault conditions are reported on the Web Admin and log files.
   - you can connect using SIP and Cisco Meeting Apps (as well as Web Bridge if that is supported).

   The downgrade of your Meeting Server deployment is now complete.

## 3.3 Cisco Meeting Server 3.1 Deployments

To simplify explaining how to deploy the Meeting Server, deployments are described in terms of three models:

- single combined Meeting Server – all Meeting Server components (Call Bridge, Web Bridge 3, Database, Recorder, Uploader, Streamer and TURN server) are available, the Call Bridge and Database are automatically enabled but the other components can be individually enabled depending upon the requirements of the deployment. All enabled components reside on a single host server.

- single split Meeting Server – in this model the TURN server and Web Bridge 3 are enabled on a Meeting Server located at the network edge in the DMZ, while the other components are enabled on another Meeting Server located in the internal (core) network.
- the third model covers deploying multiple Meeting Servers clustered together to provide greater scale and resilience in the deployment.

Deployment guides covering all three models are available here. Each deployment guide is accompanied by a separate Certificate Guidelines document.

**Points to note**:

The Cisco Meeting Server 2000 only has the Call Bridge, Web Bridge 3, and database components. It is suited for deployment on an internal network, either as a single server or a cascade of multiple servers. The Cisco Meeting Server 2000 should not be deployed in a DMZ network. Instead if a deployment requires firewall traversal support for external Cisco Meeting Server web app users, then you will need to also deploy either:

- a Cisco Expressway-C in the internal network and an Expressway-E in the DMZ, or
- a separate Cisco Meeting Server 1000 or specification-based VM server deployed in the DMZ with the TURN server enabled.

The Cisco Meeting Server 1000 and specification-based VM servers have lower call capacities than the Cisco Meeting Server 2000, but all components (Call Bridge, Web Bridge 3, Database, Recorder, Uploader, Streamer and TURN server) are available on each host server. The Web Bridge 3, Recorder, Uploader, Streamer and TURN server require enabling before they are operational.

# 4 Bug search tool, resolved and open issues

You can now use the Cisco Bug Search Tool to find information on open and resolved issues for the Cisco Meeting Server, including descriptions of the problems and available workarounds. The identifiers listed in these release notes will take you directly to a description of each issue.

1. Using a web browser, go to the Bug Search Tool.
2. Sign in with a cisco.com registered username and password.

To look for information about a specific problem mentioned in this document:

1. Enter the bug identifier in the **Search** field and click **Search**.

To look for information when you do not know the identifier:

1. Type the product name in the **Search** field and click **Search**

   or,

   in the **Product** field select **Series/Model** and start typing `Cisco Meeting Server`, then in the **Releases** field select **Fixed in these Releases** and type the releases to search for example `3.1`.
2. From the list of bugs that appears, filter the list using the *Modified Date*, *Status*, *Severity*, *Rating* drop down lists.

The Bug Search Tool help pages have further information on using the Bug Search Tool.

## 4.1 Resolved issues

**Note:** Refer to the Cisco Meeting Server web app Important information guide for information on resolved issues affecting web app.

Issues seen in previous versions that are fixed in 3.1.3.

| Cisco identifier | Summary |
|---|---|
| CSCvx82685 | On March 25, 2021, the OpenSSL Software foundation disclosed two high severity vulnerabilities affecting the OpenSSL software package identified by CVE IDs: CVE-2021-3450 and CVE-2021-3449.<br>Cisco has evaluated the impact of the vulnerability on this product and concluded that the product is affected by:<br>• CVE-2021-3449: could allow a remote unauthenticated attacker to crash a TLS server resulting in a Denial of Service (DoS) condition.<br>However, the product is not affected by:<br>• CVE-2021-3450 could allow a remote unauthenticated attacker to conduct a MiTM attack or to impersonate another user or device by providing a crafted certificate. |
| CSCvx74726 | The Meeting Server crashes in rare case when a conference is not created yet, but the callLeg is either modified due to call replacement or the API PUT operation. |
| CSCvx56476 | The initiator of an adhoc call escalating to multipoint meeting does not transmit video to Meeting Server but it receives video normally from the Meeting Server. The other two participants that are added into the CMS meeting can send and receive video normally. |
| CSCvx85827 | If no URI or an invalid URI is used when requesting the uriUsageQuery API, in certain cases it returns incorrect coSpaceID. |
| CSCvx59782 | If a web app guest user leaves a call by closing the browser tab instead of using the Leave Call button, the next attempt to join a meeting fails. |

Issues seen in previous versions that are fixed in 3.1.2.

| Cisco identifier | Summary |
|---|---|
| CSCvx32832 | When using **`toggleMuteSelfAudio`**, the initial DTMF command may not work if the participant is already muted due to **`callLegProfile`** muting that participant on join. |
| CSCvx04125 | In case of an unexpected call drop, automatic reconnect was failing. Users would see an '**unable to reconnect media**' error and a button to rejoin the meeting. |
| CSCvw98069 | User would see the screen flickering severely for some time right after one participant shares content. The flickering is caused by an unexpected restart of media on H264 high profile. |
| CSCvx19320 | An unexpected restart occurs when an activator leaves the space and disconnects all non-activators. |
| CSCvx47165 | ActiveControl is broken 15 minutes after the SIP call has been replaced. |

Issue seen in previous versions that is fixed in 3.1.1.

| Cisco identifier | Summary |
|---|---|
| CSCvw18292 | Cisco Meeting Server experiences packet loss, which recovers after a restart. |

Issues seen in previous versions that are fixed in 3.1.

| Cisco identifier | Summary |
|---|---|
| CSCvu38420 | In a conference when a participant uses Macbook Chrome to share content, the content video may change resolution occasionally. |
| CSCvv13875 | Audio-only users who have needsActivation=False dialling into a locked coSpace on Meeting Server 2.9.x will hear the audio prompt "This meeting is locked; you are waiting to be allowed in" on a continuous loop. |
| CSCvv32271 | When an external user (PSTN in this case) is transferred into a Meeting Server space, the display on Meeting Server will inaccurately show the device(s) or endpoint(s) who did the transfer versus the number of the device or endpoint number transferred into the space. |
| CSCvw61501 | The local time as returned by the MMP date command could be wrong. This issue only affects releases from 3.0 onward. |
| CSCvw03087 | The streamer will tear down the streaming call leg once an internal RTMP queue grows too large. The call will be flow controlled and once it reaches 200 kbps, it will be torn down by the streamer. |
| CSCvv19087 | The detailed tracing page option "Web Bridge connection tracing" on the Web Admin UI is still visible but now non-functional – it was used for the Web Bridge 2 component which is now removed from Meeting Server. |

## 4.2 Open issues

**Note:** Refer to the Cisco Meeting Server web app Important information guide for information on open issues affecting web app.

The following are known issues in this release of the Cisco Meeting Server software. If you require more details enter the Cisco identifier into the Search field of the Bug Search Tool.

| Cisco identifier | Summary |
|---|---|
| CSCvw61465 | Web Bridge 3 to C2W stops trying to establish a connection after 300 DNS lookup failures. |
| CSCvw61467 | If you are on the JoinCall page and your JWT expired, doing a SSO sign in wrongly takes you to the portal. |
| CSCvw61470 | The SSO domain is case-sensitive (it should not be case-sensitive). |
| CSCvw61548 | TURN logs do not show current session counts accurately |

| Cisco identifier | Summary |
|---|---|
| CSCvw61547 | On very rare occasions, calls through a Meeting Server TURN component may fail to connect or may lack a media channel. An error similar to "TURN 437 allocation mismatch in state RefreshTurnAllocationPending" will be seen in the Call Bridge syslog. |
| CSCvt74033 | When content is being shared and an event happens to trigger a Webex Room Panorama to drop from sending two video streams to one, the video frame rate being received by a remote endpoint from the Room Panorama can drop noticeably. |
| CSCvt52420 | The mediaProcessingLoad parameter returned in the system/load API on Meeting Server does not correctly account for calls using VP8 codec. When using VP8, there may be a higher actual media load on the Meeting Server than the API reports. |
| CSCvn65112 | For locally hosted branding, if the audio prompt files are omitted then the default built-in prompts are used instead. To suppress all audio prompts use a zero-byte file, rather than no file at all. |
| CSCvm56734 | In a dual homed conference, the video does not restart after the attendee unmutes the video. |
| CSCvj49594 | ActiveControl does not work after a hold/resume when a call traverses Cisco Unified Communications Manager and Cisco Expressway. |
| CSCvh23039 | The Uploader component does not work on tenanted recordings held on the NFS. |
| CSCvh23036 | DTLS1.2, which is the default DTLS setting for Meeting Server 2.4, is not supported by Cisco endpoints running CE 9.1.x. ActiveControl will only be established between Meeting Server 2.4 and the endpoints, if DTLS is changed to 1.1 using the MMP command `tls-min-dtls-version 1.0`. |
| CSCvg62497 | If the NFS is set or becomes Read Only, then the Uploader component will continuously upload the same video recording to Vbrick. This is a result of the Uploader being unable to mark the file as upload complete. To avoid this, ensure that the NFS has read/write access. |
| CSCve64225 | Cisco UCS Manager for Cisco Meeting Server 2000 should be updated to 3.1(3a) to fix OpenSSL CVE issues. |
| CSCve37087 but related to CSCvd91302 | One of the media blades of the Cisco Meeting Server 2000 occasionally fails to boot correctly. Workaround: Reboot the Fabric Interconnect modules. |

### 4.2.1 Known limitations

From version 3.1, Cisco Meeting Server supports TURN short-term credentials. This mode of operation can only be used if the TURN server also supports short-term credentials, such as the Meeting Server TURN server in version 3.1 onwards. Using Cisco Meeting Server with Expressway does not support short-term credentials.

# 5 Related user documentation

The following sites contain documents covering installation, planning and deployment, initial configuration, operation of the product, and more:

- Release notes (latest and previous releases): https://www.cisco.com/c/en/us/support/conferencing/meeting-server/products-release-notes-list.html
- Install guides (including VM installation, Meeting Server 2000, and using Installation Assistant): https://www.cisco.com/c/en/us/support/conferencing/meeting-server/products-installation-guides-list.html
- Configuration guides (including deployment planning and deployment, certificate guidelines, simplified setup, load balancing white papers, and quick reference guides for admins): https://www.cisco.com/c/en/us/support/conferencing/meeting-server/products-installation-and-configuration-guides-list.html
- Programming guides (including API, CDR, Events, and MMP reference guides and customization guidelines): https://www.cisco.com/c/en/us/support/conferencing/meeting-server/products-programming-reference-guides-list.html
- Open source licensing information: https://www.cisco.com/c/en/us/support/conferencing/meeting-server/products-licensing-information-listing.html
- Cisco Meeting Server FAQs: https://meeting-infohub.cisco.com/faq/category/25/cisco-meeting-server.html
- Cisco Meeting Server interoperability database: https://tp-tools-web01.cisco.com/interop/d459/s1790

# 6 Accessibility Notice

Cisco is committed to designing and delivering accessible products and technologies.

The Voluntary Product Accessibility Template (VPAT) for Cisco Meeting Server is available here:

http://www.cisco.com/web/about/responsibility/accessibility/legal_regulatory/vpats.html#telepresence

You can find more information about accessibility here:

www.cisco.com/web/about/responsibility/accessibility/index.html

# 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, LOST 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.

# Cisco Trademark