Cisco Meeting Server

Cisco Meeting Server Release 3.2

Release Notes

13 January 2022

Contents

What's changed

Version	Change
January 13, 2021	Updated End of Software Maintenance section.
August 17, 2021	Added section 3.1. Added CSCvy02403 as an Open issue
August 05, 2021	Maintenance release 3.2.2 Hashes updated See Resolved issues.
June 1, 2021	Updated Resolved Issues.
May 26, 2021	Maintenance release 3.2.1 Hashes updated See Resolved issues.
May 19, 2021	Updated for web app call capacities and recommendations for Medium OVA Expressway.
May 03, 2021	Added version of the CE software that supports the raise hand feature.
April 22, 2021	Minor improvements. Added Section 2.18.4.
April 07, 2021	First release for version 3.2.

1 Introduction

These release notes describe the new features, improvements and changes in 3.2 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 3.2 is required with Meeting Server 3.2. Meeting Management handles the product registration and interaction with your Smart Account (if set up) for Smart Licensing support.

Note: For Meeting Server 3.2 it is recommended to use Smart Licensing via Meeting Management. However, license files hosted locally on Meeting Server are still supported via Meeting Management for existing versions. As Meeting Server and Meeting Management intend to remove support for locally hosted licenses in future releases, you are advised to plan migration to Smart Licensing.

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 3.2.

Table 1: Call capacities across Meeting Server platforms

Type of calls	Cisco Meeting Server 1000 M4	Cisco Meeting Server 1000 M5	Cisco Meeting Server 1000 M5v2	Cisco Meeting Server 2000	Cisco Meeting Server 2000 M5v2
Full HD calls 1080p60 video 720p30 content	24	24	30	175	218
Full HD calls 1080p30 video 1080p30/4K7 content	24	24	30	175	218
Full HD calls 1080p30 video 720p30 content	48	48	60	350	437
HD calls 720p30 video 720p5 content	96	96	120	700	875
SD calls 448p30 video 720p5 content	192	192	240	1000	1250
Audio calls (G.711)	1700	2200	2200	3000	3000

Table 2 provides 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 1000 M5v2	Cisco Meeting Server 2000	Cisco Meeting Server 2000 M5v2
Individual Meeting Servers or Meeting Servers in a cluster (notes 1, 2, 3, and 4) and Meeting Servers in a Call Bridge Group	1080p30 720p30 SD Audio calls	48 96 192 1700	48 96 192 2200	60 120 240 2200	350 700 1000 3000	437 875 1250 3000
	HD participants per conference per server	96	96	120	450	450
	web app call capacities (internal calling & external calling on CMS web edge): Full HD HD SD Audio calls	48 96 192 500	48 96 192 500	60 120 240 500	350 700 1000 1000	437 875 1250 1250
Meeting Servers in a Call Bridge Group	Call type supported	Inbound SIP Outbound SIP				
	Load limit	96,000	96,000	120,000	700,000	875,000

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 21,000 HD concurrent calls per cluster (24 nodes x 875 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 medium web app scale requirements (i.e. 800 calls or less). Expressway (Medium OVA) is the recommended solution for deployments with small web app scale requirements (i.e. 200 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.

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	Medium OVA Expressway
Per Cisco Expressway (X12.6 or later)	Full HD	150	150	50
	Other	200	200	50

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 Large OVA 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. These details are 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 3.2 , Cisco announced 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.
Cisco Meeting Server version 3.0.x	The last date that Cisco Engineering may release any final software maintenance releases or bug fixes for Cisco Meeting Server version 3.0.x is August 9, 2021.

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

2 New features and changes in version 3.2

Version 3.2 of the Meeting Server software introduces the following new features and changes:

- a new Email invitation API to retrieve text based meeting entry information
- improvement in display of the meeting title displayed in the lobby
- all in-conference audio prompts mixed into speech
- a new permanent in-meeting banner at the bottom of the window
- support for in-meeting chat in web app
- ability to enable detailed tracing via API
- embedding web app within a website with a user customizable Content Security Policy
- enhancements in the meeting experience via wide main video over content
- enabling a participant to virtually raise their hand during a meeting
- capability to admit participants from an endpoint via ActiveControl into a conference
- an increase in the maximum number of supported spaces per cluster
- dial-out access method for use in dial-out calls from Meeting Server
- adding the scope parameter to accessMethodTemplates
- web app media improvements that include Improved Meeting Server to web app media resilience and change in the Maximum Transmission Unit (MTU) for web app calls
- support for increased call capacities with Meeting Server M5v2 hardware versions
- support for ESXi7.0U1c
- short-term credentials for Cisco Meeting Server edge is now a fully supported feature

2.1 Email invitation API

The new e-mail invitation API is used to retrieve text based meeting entry information suitable for distributing, typically via e-mail. The template and generation of the e-mail invitation text is shared with the Cisco Meeting Server web app Custom Email Invites feature, with an exception. If using tenants, and a **webBridgeProfile** is set at the tenant level then the **ivrNumbers** and **webBridgeAddresses** settings at the tenant level will override settings at the system/profiles level. If the **ivrNumbers** or **webBridgeAddresses** in the tenant **webBridgeProfile** are not specified, then the system level **ivrNumbers** and **webBridgeAddresses** addresses will be inherited. If no **webBridgeProfile** is configured at any level then no IVR numbers or Web Bridge addresses will be present in the invitation text.

The email invitations can be sent in different languages. For more information, see the **Invitation text customization** section in Cisco Meeting Server 3.2 Customization Guidelines.

2.1.1 API additions

The Email invitation API is introduced to retrieve text based meeting entry information suitable for distributing, typically via e-mail.

- GET on **/api/v1/coSpaces/<coSpace id>/accessMethods/<access method id>/emailInvitation**

URI Parameters	Type/Value	Description/Notes
language (optional)	String	In the form of a language tag "xx" or "xx_XX" (xx language code and XX region code) or any other string between 1 and 32 characters (allowed characters: 'a'-'z', 'A'-'Z', '0'-'9', and '_'). **Note:** Refer to Cisco Meeting Server 3.1 Customization Guidelines for the list of supported languages and for details on customizing the email invite.
organizer (optional)	String	If provided, includes the organizer details in the email invitation text. The organizer details could be name or email address of the organizer/host as included in the API.

Response Elements	Type/ Value	Description/Notes
invitation	String	Email invitation text.
language	String	Language tag of email invitation. If no language is specified, then it defaults to en_ US. If the specified language is invalid, then a "400 - Bad Request" response is returned.

Failure Responses

Description	Failure Type	Example failure responses
Language para-meter invalid (empty string, invalid characters)	400 - Bad Request	**<?xml version=\"1.0\"?><failureDetails><parameterError parameter=\"language\" error=\"invalidValue\" /></-failureDetails>**
Language para-meter too long	400 - Bad Request	**<?xml version=\"1.0\"?><failureDetails><parameterError parameter=\"language\" error=\"valueTooLong\" /></-failureDetails>**

Description	Failure Type	Example failure responses
Retry later (server too busy, fetching externally hosted template). Retry after recommended **retryAfter** period in seconds.	503 - Service Unavailable	`<?xml version=\"1.0\"?><failureDetails><retryAfter=1 /></failureDetails>`

2.2 Meeting title displayed in the lobby

Version 3.2 introduces an improvement in display of the meeting title. The meeting title is now displayed as text overlay on the welcome screen for SIP endpoints unless configured not to do so. The position of the text can be customized to avoid obscuring important areas of custom lobby screens. If there is a PIN for the conference, the title is not displayed until the correct PIN is entered. The meeting title is also shown when waiting in the lobby either when the call is locked or when you are waiting for the host.

Note: The meeting title is taken from the meeting title provided in web app, Cisco TelePresence Management Suite.

2.2.1 New API addition

A new **meetingTitlePosition** API request parameter is introduced in 3.2 to implement this feature. This parameter takes the values **top** | **middle** | **bottom** to enable and place the meeting title, or **disabled** to remove it. This is introduced for the following methods:

- POST to **/callLegProfiles**
- PUT to **/callLegProfiles/<callLegProfile id>**
- GET on **/callLegProfiles/<callLegProfile id>**
- POST to **/calls/<call id>/callLegs**
- PUT to **/callLegs/<callLeg id>**
- GET on **/callLegs/<callLeg id>**
- GET on **/callLegs/<call leg id>/callLegProfileTrace**

Note: If unset in the callLeg and all levels of the callLegProfile hierarchy, its value defaults to **bottom**.

2.3 In-conference audio prompts mixed into speech

In the 3.2 release, all audio prompts playable during a meeting including participant join and leave tones, will be merged into the participants speech rather than overriding it. However, this does not apply to prompts played to participants in the lobby.

Note: This feature is not configurable. The new experience will be automatic for everyone and cannot be turned off.

2.4 Permanent in-meeting banner

Version 3.2 allows a banner to be displayed in meetings. The banner is more visible to users, its position is not configurable, and it is permanent until explicitly removed. This banner can be configured before the call starts using the **`callProfiles`** API. The in-meeting banner works for both, SIP endpoints and the Cisco Meeting Server web app.

This feature is enabled by the new **`messageBannerText`** API request parameter. It accepts a single parameter, a string which is the message to be displayed on screen. To remove a message banner, set the parameter to be an empty string. In the default setting, the string is empty.

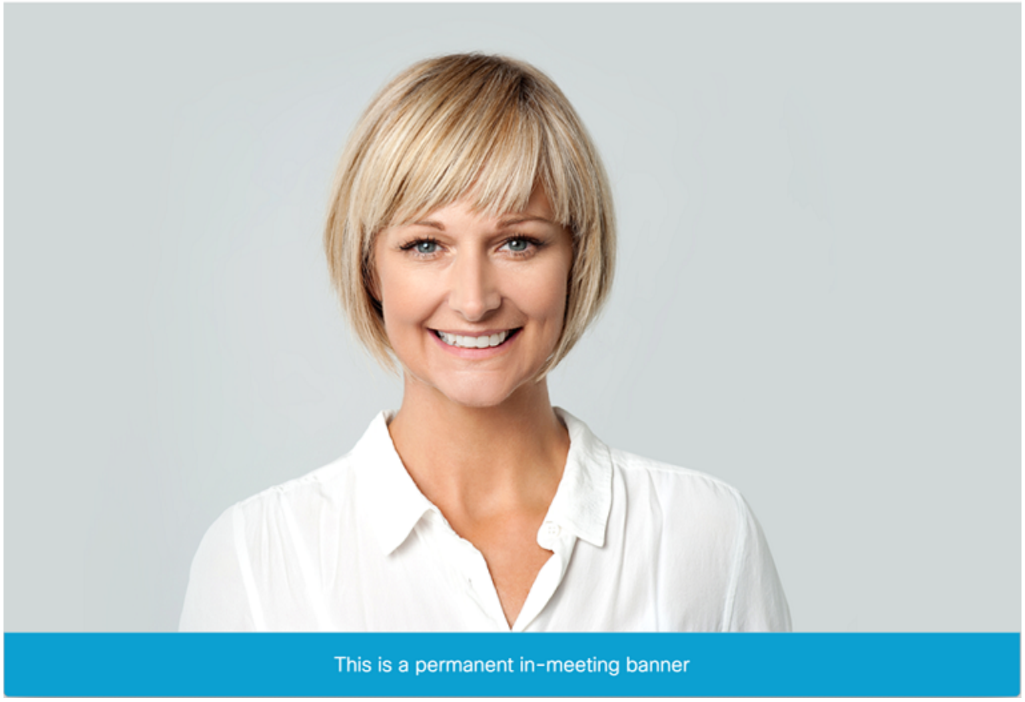
Note: While the existing **`messageText`** feature provides the functionality for overlaying text on top of a video stream with a configurable position, the in-meeting banner enabled by the new **`messageBannerText`** parameter, has a constant position at the bottom of the window.

`messageBannerText` is available for:

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

Note: The **`messageBannerText`** can be a maximum of 200 bytes.

The following image depicts the permanent meeting banner on web app:



2.5 In-meeting chat

This feature supports in-meeting chat in Cisco Meeting Server web app:

- Chat is available only during an ongoing call. Participants cannot chat in a coSpace outside of a call.
- Chat messages are not persistent. When a call ends, all chat messages sent during that call are lost permanently.
- When a participant sends a chat message, it is broadcast to all participants currently in the call whose clients are capable of receiving chat.
- Participant-to-participant chat is not supported.
- When a new participant joins a call, they receive only those chat messages that were sent since they joined the call, and not the entire chat history.
- SIP participants cannot receive chat as messages are not rendered on the screen of SIP endpoints.

Note: Chat interoperability support is limited to Skype for Business clients. Skype for Business clients can send and receive messages while participating in a Cisco Meeting Server hosted meeting. However, they cannot send any attachments. Standard SIP participants do not receive chat messages as they are not displayed on the screen.

2.5.1 New API request parameter to enable / disable chat

A new **chatAllowed API** request parameter is introduced in 3.2 to enable/disable chat at a call level. The acceptable range of values are **true**, **false** or **""** denoting <**unset**>. The parameter is supported on the following API operations:

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

The parameter is optional and like other parameters of the call profiles, the value of **chatAllowed** is dictated by the rules of inheritance in the call profile/call hierarchy as usual: explicit values in the profiles lower in the hierarchy override those set above, and if a parameter is unset, it inherits from the next profile up in the hierarchy. If the value is unset at all levels of the call profile hierarchy, then it defaults to **true**.

Request parameters	Type/ Value	Description/ Notes
chatAllowed	**true**, **false**, or <**unset**>	If the value is specified, determines whether or not chat is allowed on this call/s using this call profile.

Additionally, the administrator can control at a finer level of granularity which participants in a given call are allowed to send chat messages. A participant can send a chat message if chat is allowed on the call and the participant is allowed to contribute chat messages. This is controlled by the new parameter **chatContributionAllowed**, introduced for the following API operations:

- POST to **/callLegProfiles**
- GET on **/callLegProfiles/<call leg profile id>**
- PUT to **/callLegProfiles/<call leg profile id>**
- POST to **/calls/<call id>/callLegs**
- GET on **/callLegs/<call leg id>**
- PUT to **/callLegs/<call leg id>**
- POST to **/calls/<call id>/participants**

With this type of parameter, if a value of <**unset**> is used, the rules of inheritance in the **callLeg/callLegProfile** hierarchy are followed. If not set at any level, then it defaults to **true**.

Note: Even when this setting is **`true`**, if **`chatAllowed=false`** at the call level (or because the call inherited it from the callProfile hierarchy) then chat contribution will still not be allowed.

Request parameters	Type/ Value	Description/ Notes
chatContributionAllowed	**`true`**, **`false`**, or <**`unset`**>	If the value is specified, determines whether or not this call leg/this participant/call legs using this **`call leg profile`** are allowed to send messages on the chat.

2.6 Enable detailed tracing via API

With this feature, the existing detailed tracing available from **Logs** > **Detailed tracing** web admin page can now be enabled using the management API as well. The API and web interface operations work on the same items. For example, a change to a timed logging value on the web interface should result in the new value being read back from an API GET. Similarly, using an API PUT command to modify, for example, the DNS timed logging status should result in that change being seen on the web page too.

2.6.1 API additions

This feature introduces a new API node, **`/system/timedLogging`** to support the following operations:

- PUT to **`/system/timedLogging`**
- GET on **`/system/timedLogging`**

It supports the parameters detailed in the table below. Each parameter can be assigned an integer value, corresponding to the duration of seconds for which that logging subsystem will be activated.

Setting a parameter to 0 or to nothing will deactivate a logging subsystem. For example, a PUT to **`system/timedLogging`** with **`sip=60`** would activate detailed logging for SIP for 60 seconds. A PUT to **`system/timedLogging`** with **`sip=0`** before those 60 seconds have elapsed would deactivate the logging again. You can supply multiple parameters at the same time, for example: **`sip=600&tip=600`** to enable both SIP and TIP logging for the next 10 minutes.

The following parameters are available for this object:

Parameter	Type/ Value	Description/ Notes
activeControl	numeric	time remaining (in seconds) for which detailed Active Control logging should be enabled

Parameter	Type/ Value	Description/ Notes
activeSpeaker	numeric	time remaining (in seconds) for which detailed active speaker logging should be enabled
api	numeric	time remaining (in seconds) for which detailed API logging should be enabled
bfcp	numeric	time remaining (in seconds) for which detailed BFCP logging should be enabled
cameraControl	numeric	time remaining (in seconds) for which detailed camera control logging is enabled (0 if not enabled)
dns	numeric	time remaining (in seconds) for which detailed DNS logging should be enabled
events	numeric	time remaining (in seconds) for which detailed Events logging should be enabled
ice	numeric	time remaining (in seconds) for which detailed ICE logging should be enabled
sip	numeric	time remaining (in seconds) for which detailed SIP logging should be enabled
tip	numeric	time remaining (in seconds) for which detailed TIP logging should be enabled
webBridge	numeric	time remaining (in seconds) for which detailed web bridge logging should be enabled

2.7 User customizable Content Security Policy

From Meeting Server version 3.2 onwards, system administrators can embed the web app within a website.

The web app does not check the header contents besides checking that the characters are valid. The administrators must ensure that the content security policy header contains valid strings. The string size is limited to 1000 characters and allowed characters are **a-z A-Z 0-9_ <space>. / : ? # [ ] @ ! $ & ' ( ) * + - = ~ %**

Note: Web app can run media when embedded in the browsers that require https and not on browsers with http.

2.7.1 MMP additions

In this release, the new MMP command **`webbridge3 https frame-ancestors`** is added to Cisco Meeting Server and Cisco Meeting Server 2000. It allows administrators to specify a custom frame-ancestors value to be returned in the **`content-security-policy`** header allowing the web app to be embedded in other web pages.

Note: In a cluster setup, this command must be configured on all Web Bridges in the deployment.

```
webbridge3 https frame-ancestors <frame-ancestors space-separated
string>
```

```
webbridge3 https frame-ancestors none
```

For example,

```
webbridge3 https frame-ancestors https://*.example.com
https://customdomain.example2.com:8000
```

2.7.2 iframe example for embedded web app

Here is an example of an iframe that embeds the website with the minimum feature policies necessary to let the app run:

```
<iframe src="https://<address>:<port>/" allowusermedia
allow="microphone; camera; encrypted-media; display-
capture;"></iframe>
```

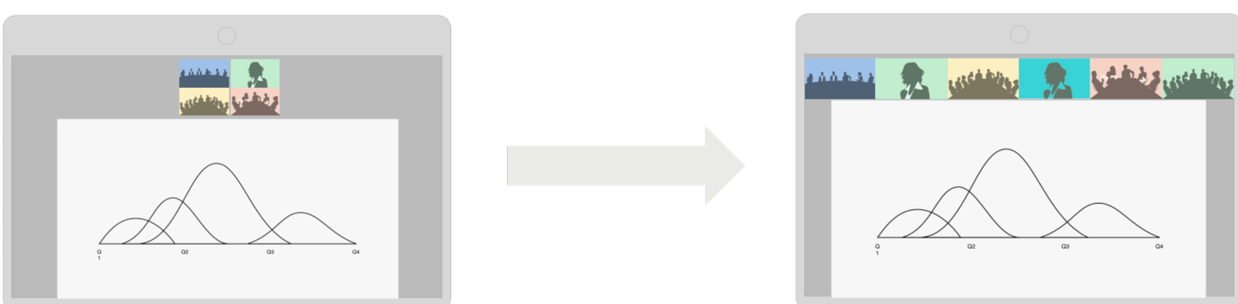
Where Web Bridge 3: `https://<address>:<port>/` is the address of the web bridge.

Note: We recommend using a certificate signed by a public Certificate Authority (CA) with the web app. If a custom certificate is used then the web app may not be visible in the embedded page until you have navigated to the original web app site and accepted the custom certificate.

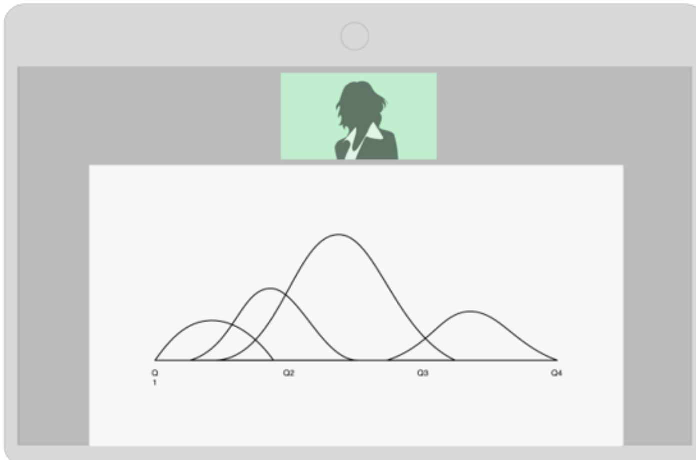
2.8 Wide main video over content

Version 3.2 introduces support for flexible video channel sizes. This feature enhances the meeting experience for single screen video endpoints by displaying the participants in a row whilst the presentation is the main focus of the meeting.

The previous stacked layout depicted on the left is now replaced with participants displayed in a row on top of the presentation as shown on the right. This gives a much clearer view of the participants in the meeting. Up to six participants can be displayed in this layout.



The stage layout has been enhanced to show only the active speaker without detracting from the presentation.



This layout takes precedence over the server configured layouts, and therefore cannot be changed from Meeting Server. However, users can still change the layout from the endpoint.

Pane placement is supported, but if pane placement is enabled or modified whilst the endpoint is in stacked layout, the changes will not be reflected.

This feature uses ActiveControl to allow other devices to specify the sizes of the main and content video channels that they would like to receive, and only works with ActiveControl endpoints that have implemented the wide main over content feature. This will be supported in a future Collaboration Endpoint Software release. For more information on implementation in the Collaboration Endpoint Software, see Cisco Collaboration Endpoint Software Release Notes.

Note: This feature is enabled by default and is not configurable.

2.9 Raise hand

Version 3.2 introduces a new feature that enables a participant to virtually raise or lower their hand during a meeting by clicking on a button or by tapping the screen. An administrator or operator with API privileges can also raise or lower a participant's hand. Meeting Server will also indicate to ActiveControl capable endpoints, the list of participants who have their hand raised so that a raised hand icon is displayed next to each participant in the roster list. ActiveControl endpoints also show the total number of raised hands.

This feature is supported via ActiveControl from Collaboration Endpoint software version 9.15.3.17 onwards. For more information on how raise hand is implemented in Collaboration Endpoint software, see Cisco Collaboration Endpoint Software Release Notes.

Note: The ability to raise / lower hand and see hand raised status is not available on web app or SIP endpoints that do not support this ActiveControl feature.

2.9.1 New API additions

This feature introduces the new **handStatus** parameter. It accepts a single value, that indicates whether a participant's hand is **raised** or **lowered**. The value is not returned if the **handStatus** has not changed during the call. The parameter is available for:

- POST to **/calls/<call id>/participants**
- GET on **/callLegs/<call leg id>**
- PUT to **/callLegs/<call leg id>**
- PUT to **/calls/<call id>/participants/***
- PUT to **/participants/<participant id>**
- PUT to **/calls/<call id>/<participant id>**
- POST to **/calls/<call id>/callLegs**

The **handStatusLastModified** parameter provides information about the date / time and indicates when the **handStatus** was last modified. The value is not returned if the **handStatus** has not changed during the call.

- GET on **/callLegs/<call leg id>**

The **raiseHandEnabled** parameter with the values **true** | **false** allows an administrator to enable or disable the feature for the whole call. By default, the parameter is <**unset**>, but if unset at all levels in the call / callProfile hierarchy, then raise hand will default to enabled.

It is supported on the following APIs:

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

2.10 Admit participants from an endpoint via ActiveControl

Cisco Meeting Server now includes an ActiveControl capability to admit participants into a conference from the lobby of a locked call. ActiveControl capable endpoints and clients that support this feature can now display a message when a participant is waiting in the lobby. A user can then admit the participant into the meeting.

Meeting Server allows administrators to control which participants can admit participants from the lobby using the **callLockAllowed** parameter that can be set on a call leg or call leg profile. To have **admit** participant permission, the participant must be in an activated state and **callLockAllowed** parameter must be enabled. This is applicable to ActiveControl endpoint versions that support this feature.

This feature enriches the experience of making calls between Collaboration Endpoint software or Cisco Jabber, and Meeting Server. It also helps create a parity between CE endpoints and web app participants, who dial into Meeting conferences. For more information on versions that support this feature, see Cisco Collaboration Endpoint Software Release Notes and Cisco Jabber Release Notes.

2.11 Increase in maximum number of supported spaces

In version 3.2, the maximum number of supported spaces per cluster in Meeting Server 1000, Meeting Server 2000, and in specification-based VM platforms* has been increased from 75,000 to 500,000.

Note: The number of supported users still remains at 75,000.

---

* For specification-based VMs, the minimum RAM requirement is 4GB but 8GB is recommended for deployments between 1000 and 75,000 spaces. For more than 75,000 coSpaces, we recommend 8GB RAM + 1GB per 100,000 coSpaces over 75,000 coSpaces. For more information on VM configuration requirements, see Virtualization for Cisco Meeting Server and *Cisco Meeting Server Installation Guide for Cisco Meeting Server 1000 and Virtualized Deployments*.

2.12 Dial-out access method

Version 3.2 introduces the ability to configure an **accessMethod** and a **defaultAccessMethod** for use on dial out calls from Meeting Server. On dial out from a coSpace, Meeting Server will use the callLegProfile and the importance value from this access method, and the outbound call will use its URI as the from address. If an **accessMethod** is not specified on dial out, a **defaultAccessMethod** will be used, provided it is configured for the coSpace. When an endpoint wants to return the call, it is made to the access method from the original outbound call.

The **accessMethod** parameter is now supported on POST to /calls/<call id>/callLegs and /calls/<call id>/participants. The new **defaultAccessMethod** parameter is supported on PUT and GET on /coSpaces/<cospaceId>, and **defaultAccessMethodTemplate** parameter is supported on PUT and GET on /coSpaceTemplates/<coSpaceTemplate id>.

2.12.1 Order of precedence for selection of access method

This feature introduces a new order of precedence for selection of access methods for outgoing calls. These rules determine which access method is going to be used on dial out (if any).

The order of precedence from highest to lowest is:

1. **`accessMethod`** as set on POST to **`/calls/<call id>/participants`** or **`/calls/<call id>/callLegs`**
2. **`defaultAccessMethod`** as set on PUT to **`/cospaces/<cospace id>`**
3. As in previous release versions, the public access method with the smallest GUID (referred to as the "primary" access method).
4. If the access method ID is unset at all levels of the hierarchy, then the coSpace parameters are used on dial out.

Once the access method is chosen, its call leg profile, URI, and importance values are employed in existing algorithms to determine the behaviour on dial out.

2.12.2 Call leg profiles

When you dial out using an access method, the settings will be inherited in the same order of precedence as in a dial in call. The order is as follows, with 1 taking the highest precedence:

1. callLeg specific settings
2. callLegProfile on the callLeg
3. callLegProfile on accessMethod
4. callLegProfile on coSpace
5. callLegProfile on tenant
6. callLegProfile in /system/profiles

The order of precedence for choosing the access method determines only which call leg profile is employed at level 3.

2.12.3 URIs

The access method chosen according to the order of precedence described earlier also provides the access method URI. On dial out, the URI where the call can be returned is selected as follows:

1. If the access method URI is configured, then its value is used.
2. Otherwise, if a primary access method (public access method with the smallest GUID) is present and has a URI configured, the URI from the primary access method is used.
    a. In the absence of a primary access method, if the cospace URI is configured, then its value gets adopted.

3. Otherwise, if an IVR is configured on the Web Admin (Configuration/General configuration/IVR numeric ID), then the IVR URI gets employed.

Note: The IVR configured via the Web Admin is separate from the IVRs that are configured by using the Admin API (via operations on /ivr).

---

4. Otherwise, the URI does not have a user part and only the IP address of the call bridge is used.

2.12.4 Importance

Configuring the importance value directly on the participant object via a POST on **`/participants`** or PUT on **`/participants/<participant id>`** overrides the access method importance. When unsetting the importance on the participant object (via PUT "" for "importance" on /participants/<participant id>), the importance value inherited from the access method, if any, still remains in place but it could be unset/overridden by explicitly setting it to "0" on the participant object.

2.12.5 Default access method for new members added to a coSpace using the web app

When adding a new member to a coSpace using the Meeting Server web app, the new member will now be configured with the call leg profile specified in the default access method of the coSpace if there is one. In previous releases, the new member would not automatically have a call leg profile configured.

If the default access method of the coSpace is changed, then the call leg profile of existing coSpace members is not changed.

2.12.6 API additions

A new **`defaultAccessMethod`** optional field is introduced on **`/coSpaces/<coSpaceId>`** to specify the default access method to be used when dialing out. This field is supported in the following methods:

- GET on **`/coSpaces/<cospaceId>`**
- PUT to **`/coSpaces/<cospaceId>`**

Parameter	Type/Value	Description/Notes
`defaultAccessMethod` (Optional)	ID | ""	Associates the specified access method as the default access method to be used for dial outs.

A new **`defaultAccessMethodTemplate`** parameter is introduced for the following methods:

- GET on **/coSpaceTemplates/<coSpaceTemplate id>**
- PUT to **/coSpaceTemplates/<coSpaceTemplate id>**

Parameter	Type/Value	Description/Notes
defaultAccessMethodTemplate	ID | ""	If specified, associates the access method template as the default one for the coSpace template. When a coSpace is instantiated from the coSpace template, the instantiated default access method template becomes the default access method for the coSpace.

A new **accessMethod** parameter is introduced on callLegs and Participants for the following methods:

- POST to **/calls/<call id>/callLegs**
- GET on **/callLegs/<callLeg id>**
- POST to **/calls/<call id>/participants**
- GET on **/participants/<participant id>**

Note: This parameter is not returned for non-coSpace calls. It is only applicable where the call corresponds to a coSpace call and the ID is an accessMethod within the context of that coSpace.

Parameter	Type/Value	Description/Notes
accessMethod	ID | "coSpace"	• On POST (optional): Associates the specified **accessMethod** as the access method for the callLeg/participant and overrides any default or primary access method on the coSpace. • On GET: Returns the access method used to join the call on dial in or as set on dial out. Where an **accessMethod** is not specified on the POST operation, the GET still returns an access method ID if **defaultAccessMethod** was configured on the coSpace or if the primary access method was employed. • The API will return "coSpace" if the coSpace was not joined through an access method. This could be on dial in or on dial out if no **accessMethod** or **defaultAccessMethod** is specified and no primary access method exists. **Note:** coSpace is only available on GET operations.

2.13 Adding scope to accessMethodTemplates

In Meeting Server 3.2, **accessMethodTemplates** can now include a **scope** parameter for controlling the visibility of any coSpace access method created using that template. It controls the visibility of this coSpace access method to users of web app who are members of the coSpace. Users with a certain access method (for example, a guest access method), cannot see the details of users with a different access method (such as host access method).

CoSpaces already created will not be changed on upgrade to 3.2, but newly created spaces using existing coSpace templates will have a default accessMethod scope of **private**.

Note: In previous releases, you could not set the scope on accessMethodTemplates. When creating a coSpace from the template, the corresponding accessMethod scope would default to **public**. This behaviour changes in 3.2.

2.13.1 API addition

The resulting video address of spaces created with this accessMethodtemplate will not have a domain appended.

- POST to **`/coSpaceTemplates/<coSpace template id>/accessMethodTemplates`**
- GET on **`/coSpaceTemplates/<coSpace template id>/accessMethodTemplates`**
- PUT to **`/coSpaceTemplates/<coSpace template id>/accessMethodTemplates/<access method template id>`**
- GET on **`/coSpaceTemplates/<coSpace template id>/accessMethodTemplates/<access method template id>`**

2.14 Web app media improvements

Version 3.2 includes improvements to increase the overall media quality of the web app.

2.14.1 Improved Meeting Server to web app media resilience

Version 3.2 includes improvements to presentation video streams between Meeting Server and web app calls, in the presence of packet loss.

Some web browsers support use of "NACK" (negative acknowledgment) for video packets, whereby on noticing a video packet missing, they ask the sender via RTCP to repeat that packet sequence number rather than requesting the entire stream resync with a key frame.

In previous versions, when Meeting Server receives an RTCP NACK packet, it sends a whole keyframe in the video stream. In the other direction, when detecting a missing packet Meeting Server asks the far end (browser contributing the video stream) for a key frame. Sending a whole key frame uses more bandwidth, increasing the probability of further loss, and can result in temporary degradation of picture quality.

From version 3.2 onwards,

- Meeting Server is able to resend individual video packets to the far end in response to RTCP NACK messages.
- When its video decoders detect loss, Meeting Server requests individual video packets to be resent.

With this improvement, in the presence of a bad network between Meeting Server and the web browser, only the parts of individual frames that have been lost are resent to the receiver.

Note: This feature is enabled by default and is not configurable.

2.14.2 Maximum Transmission Unit (MTU) changes for web app calls

In version 3.2, the payload size for outgoing web app media packets from the Meeting Server has been restricted to 1200 bytes in order to keep the overall MTU below 1280 bytes.

Note: This feature is enabled by default and is not configurable.

2.15 Support for increased call capacities with Meeting Server M5v2 hardware versions

From version 3.2 onwards, we support an increased scale on Meeting Server 1000 M5v2 and Meeting Server 2000 M5v2 hardware variants.

- The load limit for Meeting Server 1000 M5v2 has increased from 96,000 to 120,000. The Meeting Server 1000 call capacity for 720p video calls has increased from a maximum of 96 to 120 on the new platform.
- The load limit for Meeting Server 2000 M5v2 has increased from 700,000 to 875,000. The Meeting Server 2000 call capacity for 720p video calls has increased from 700 to 875 on the new platform.

The capacities for the different call resolutions have also increased to match the new load limits. The section Cisco Meeting Server platform maintenance includes full details for call capacities across Meeting Server platforms and call capacities for a single or cluster of Meeting Servers compared to load balancing calls within a Call Bridge Group.

In order to take advantage of the new scale, specify the `loadLimit` parameter on the server to the new values on these hardware variants. See Cisco Meeting Server deployment guides for more information on specifying the load limit on a cluster and enabling load balancing.

Table 5: Load limits for Meeting Server platforms

System	Load Limit
Meeting Server 2000 M5v2	875,000
Meeting Server 2000	700,000
Meeting Server 1000 M5v2	120,000
Meeting Server 1000	96,000
VM	1250 per vCPU

Note: Load limits for previous versions of Meeting Server platforms stay as they were; these changes only apply to the M5v2 variants.

2.16 ESXi support

Version 3.2 adds support on Meeting Server 1000 M4, M5, M5v2, and specs-based servers for:

- ESXi7.0U1c with Virtual Hardware version 11

Previous ESXi versions also supported by version 3.2 include ESXi6.5u2, and 6.7U3.

2.17 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. When deploying Cisco Meeting Server web edge, the API node **`/turnServers`** "type" parameter should be set to "cms". If this parameter is unset, it defaults to "standard", and tells the clients to use TCP/UDP port 443 to connect to the TURN server. For more information on the "type" parameter values, see the section *Setting up and modifying TURN servers* in Cisco Meeting Server API Reference Guide.

2.17.1 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.

2.17.2 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)

2.17.2.1 Parameter updates

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

2.17.3 Implementing short term credentials on the Meeting Server

These steps assume you have already upgraded to version 3.2.

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.

2.18 Summary of 3.2 API additions and changes

API functionality for the Meeting Server 3.2 includes:

- New API objects and parameters to support Email invitation
- New API objects and parameters for configuration of metadata on a coSpace
- New API objects and parameters to enable detailed tracing via API

2.18.1 API additions

New API functionality for the Meeting Server 3.2 include new API objects and minor API enhancements.

New API objects

- **`/coSpaces/<coSpace id>/accessMethods/<access method id>/emailInvitation`**

- **`/coSpaces/<coSpace id>/metadata`**

  Metadata is a text string which can be configured on the coSpace which allows management applications such as Cisco Meeting Management to store metadata on a coSpace.

  Note: Some Meeting Management features such as blast dial require metadata to be stored on the coSpace. Changing the metadata can cause these features to fail.

- **`/system/timedLogging`**

  Note: For more information on the parameters for this API, see Enable detailed tracing via API.

Minor API enhancements

- **`coSpaceMetaDataConfigured`** response element on
  - GET on **`/calls/<call id>`**

  This value is **`true`** if there is metadata configured on **`/cospaces/<cospace id>/metadata`** and **`false`** otherwise.

- **`confirmationStatus`** parameter on
  - GET on **`/callLegs/<callLeg id>`**

  This parameter returns **`required/notRequired/confirmed`** depending on whether the confirmation has been given. See **`confirmationStatus`** in Summary of CDR Changes for more information on confirmation.

- To control H.264 parameters used by Safari browsers for WebRTC calls, a new request parameter **`safariWebRtcH264interopMode`** is introduced for:
  - POST to **`/compatibilityProfiles`**
  - PUT to **`/compatibilityProfiles/<compatibility profile id>`**
  - GET on **`/compatibilityProfiles/<compatibility profile id>`**

  The parameter **`safariWebRtcH264interopMode`** is one of **`auto`** or **`none`**, where
  - **`auto`**: SDPs sent to WebRTC clients running on Safari will disable H.264 High Profile, and advertise Base Profile Level 5. This is the default value.
  - **`none`**: No change from previous releases.

  Note: Use this parameter only under supervision from the Cisco Technical Support team.

Note: If this parameter is changed, the new setting is applied to any new WebRTC sessions; whilst active WebRTC sessions require a page refresh and will need to rejoin the call. Ongoing WebRTC calls are unaffected.

New/modified error code reasons introduced in version 3.2

- **`accessMethodDoesNotExist`** – You tried to create a call leg or participant with an access method that does not correspond to the call's coSpace.
- **`coSpaceCallDoesNotExist`** – You tried to create a call leg or participant (with an access method specified) on a call that is not associated with a coSpace.

2.18.2 New and modified parameters

New parameters in version 3.2.

- **`meetingTitlePosition`** is introduced on
  - POST to **`/callLegProfiles/`**
  - PUT to **`/callLegProfiles/<callLegProfile id>`**
  - GET on **`/callLegProfiles/<callLegProfile id>`**
  - PUT to **`/callLegs/<callLeg id>`**
  - GET on **`/callLegs/<callLeg id>`**
  - GET on **`/callLegs/<call leg id>/callLegProfileTrace`**
  - POST to **`/calls/<call id>/participants`**
  - POST to **`/calls/<call id>/callLegs`**
- **`messageBannerText`** is introduced on
  - POST to **`/callProfiles`**
  - PUT to **`/callProfiles/<call profile id>`**
  - GET on **`/callProfiles/<call profile id>`**
  - POST to **`/calls`**
  - PUT to **`/calls/<call id>`**
  - GET on **`/calls/<call id>`**
- **`chatAllowed`** is introduced on
  - POST to **`/callProfiles`**
  - PUT to **`/callProfiles/<call profile id>`**
  - GET on **`/callProfiles/<call profile id>`**

  - POST to **/calls**
  - PUT to **/calls/<call id>**
  - GET on **/calls/<call id>**

- **chatContributionAllowed** is introduced on
  - POST to **/callLegProfiles**
  - PUT to **/callLegProfiles/<call leg profile id>**
  - GET on **/callLegProfiles/<call leg profile id>**
  - PUT to **/callLegs/<call leg id>**
  - GET on **/callLegs/<call leg id>**
  - POST **/calls/<call id>/participants**
  - POST **/calls/<call id>/callLegs**

- **handStatus** is introduced on
  - PUT to **/callLegs/<call leg id>**
  - GET on **/callLegs/<call leg id>**
  - PUT to **/participants/<participant id>**
  - GET on **/participants/<participant id>**
  - POST to **/calls/<call id>/participants**
  - POST **/calls/<call id>/callLegs**

- **handStatusLastModified** is introduced on
  - GET on **/callLegs/<call leg id>**
  - GET on **/participants/<participant id>**

- **raiseHandEnabled** is introduced on:
  - POST to **/callProfiles**
  - PUT to **/callProfiles/<call profile id>**
  - GET on **/callProfiles/<call profile id>**
  - POST to **/calls**
  - PUT to **/calls/<call id>**
  - GET on **/calls/<call id>**

- **`callBridgeGroupFilter`** URI parameter is introduced on
  - GET on **`/webBridges`** as part of the request **`/webBridges/webBridges?callBridgeGroupFilter=<call bridge group id>`**
- **`defaultAccessMethod`** parameter is introduced on
  - PUT to **`/coSpaces/<coSpace id>`**
  - GET on **`/coSpaces/<coSpace id>`**
- **`defaultAccessMethodTemplate`** parameter is introduced on
  - PUT to **`/coSpaceTemplates/<coSpace template id>`**
  - GET on **`/coSpaceTemplates/<coSpace template id>`**
- **`accessMethod`** parameter is introduced on
  - POST to **`/calls/<call id>/callLegs`**
  - GET on **`/callLegs/<callLeg id>`**
  - POST to **`/calls/<call id>/participants`**
  - GET on **`/participants/<participant id>`**
- **`canChangeScope`** parameter is introduced on
  - POST to **`/coSpaces/<coSpace id>/coSpaceUsers`**
  - PUT to **`/coSpaces/<coSpace id>/coSpaceUsers/<coSpace user id>`**
  - GET on **`/coSpaces/<coSpace id>/coSpaceUsers/<coSpace user id>`**

Modified parameter in version 3.2

The **`scope`** parameter is added to the accessMethodTemplates object with the values **`public`** | **`private`** | **`member`** | **`directory`**.

The resulting video address of spaces created with this accessMethodtemplate will not have a domain appended.

- POST to **`/coSpaceTemplates/<coSpace template id>/accessMethodTemplates`**
- GET on **`/coSpaceTemplates/<coSpace template id>/accessMethodTemplates`**
- PUT to **`/coSpaceTemplates/<coSpace template id>/accessMethodTemplates/<access method template id>`**
- GET on **`/coSpaceTemplates/<coSpace template id>/accessMethodTemplates/<access method template id>`**

2.18.3 Retrieving Email invitation text

The Email invitation API is introduced to retrieve text based meeting entry information suitable for distributing, typically via e-mail.

- GET on **/api/v1/coSpaces/<coSpace id>/accessMethods/<access method id>/emailInvitation**

URI Parameters	Type/Value	Description/Notes
language (optional)	String	In the form of a language tag "xx" or "xx_XX" (xx language code and XX region code) or any other string between 1 and 32 characters (allowed characters: 'a'-'z', 'A'-'Z', '0'-'9', and '_'). **Note:** Refer to [Cisco Meeting Server 3.1 Customization Guidelines]() for the list of supported languages and for details on customizing the email invite.
organizer (optional)	String	If provided, includes the organizer details in the email invitation text. The organizer details could be name or email address of the organizer/host as included in the API.

Response Elements	Type/ Value	Description/Notes
invitation	String	Email invitation text.
language	String	Language tag of email invitation. If no language is specified, then it defaults to en_ US. If the specified language is invalid, then a "400 - Bad Request" response is returned.

2.18.4 Configuring and retrieving coSpace metadata

Metadata is a text string which can be configured on the coSpace, which allows management applications such as Cisco Meeting Management to store metadata on a coSpace. This is supported from version 3.2 onwards with the API node **/coSpaces/<coSpace id>/metadata** on the following methods:

- PUT to **/coSpaces/<coSpace id>/metadata**
- GET on **/coSpaces/<coSpace id>/metadata**

The **coSpaceMetaDataConfigured** response element on GET on **/calls/<call id>** returns true if there is metadata configured on **/cospaces/<cospace id>/metadata** and false otherwise.

Note: Some Meeting Management features such as blast dial require metadata to be stored on the coSpace. Changing the metadata can cause these features to fail.

2.18.5 Detailed tracing via API

Version 3.2 introduces a new API node, **`/system/timedLogging`** to support the following operations:

- PUT to **`/system/timedLogging`**
- GET on **`/system/timedLogging`**

For more information and supported parameters, see the section Enable detailed tracing via API.

2.18.6 Setting and retrieving a meeting title position

A new **`meetingTitlePosition`** API request parameter is introduced in 3.2 to implement this feature on the following methods:

- POST to **`/callLegProfiles/`**
- PUT to **`/callLegProfiles/<callLegProfile id>`**
- POST to **`/calls/<call id>/callLegs`**
- POST to **`/calls/<call id>/participants`**
- PUT to **`/callLegs/<callLeg id>`**

Request parameter	Type/Value	Description/Notes
`meetingTitlePosition`	disabled | top | middle | bottom	Enables and places the meeting title at the specified position. If unspecified, it takes the value **`bottom`**. The value **`disabled`** removes the meeting title.

- GET on **`/callLegProfiles/<callLegProfile id>`**
- GET on **`/callLegs/<call leg id>/callLegProfileTrace`**
- GET on **`/callLegs/<callLeg id>`**

Response element	Type/Value	Description/Notes
`meetingTitlePosition`	top | middle | bottom | disabled	Enables and places the meeting title at the specified position. If unspecified, it takes the value **`bottom`**. The value **`disabled`** removes the meeting title.

2.18.7 Creating, modifying, and retrieving the meeting banner text

A new **`messageBannerText`** API request parameter is introduced to implement the permanent in-meeting banner feature on the following methods:

- PUT to **/callProfiles/<callProfile id>**
- POST to **/calls**
- PUT to **/calls/<call id>**
- POST to **/callProfiles**

Request parameter	Type/Value	Description/Notes
messageBannerText	String	The string is the message to be displayed on the screen. The default value is an empty string, which does not display the message banner.

- GET on **/callLegProfiles/<callLegProfile id>**
- GET on **/calls/<call id>**

Response element	Type/Value	Description/Notes
messageBannerText	String	The string is the message to be displayed on the screen. The default value is an empty string, which does not display the message banner.

2.18.8 Enabling / disabling in-meeting chat

A new **chatAllowed API** request parameter is introduced to enable/disable chat at a call level. See In-meeting chat for more information.

2.18.9 Enabling, modifying, and retrieving raise hand status

The raise hand feature introduces the new **handStatus** parameter in version 3.2. The parameter is available for:

- POST to **/calls/<call id>/callLegs**
- PUT to **/callLegs/<call leg id>**
- PUT to **/participants/<participant id>**
- POST to **/calls/<call id>/participants**
  Create a new participant for the specified call; the parameters are as per the call leg create operation, but may result in the call leg instantiation ("owned" by the new participant object) to take place on a remote clustered call bridge.

Parameter	Type/Value	Description/Notes
handStatus	**raised** | **lowered**	Specifies whether or not to raise or lower hand for this participant or call leg.

- GET on **/callLegs/<call leg id>**

- GET on **/participants/<participant id>**
  Allows modification of some properties of a participant.

Response elements	Type/Value	Description/Notes
handStatus	**raised** | **lowered**	If set, indicates whether the hand is raised or lowered for this participant or call leg. The value is not returned if the handStatus was not changed during the call.

The **handStatusLastModified** parameter is introduced to indicate when the **handStatus** was last modified.

- GET on **/participants/<participant id>**
- GET on **/callLegs/<call leg id>**

Response elements	Type/Value	Description/Notes
handStatusLastModified	string	Returns a UTC date-time for the last time the hand status was modified. The value is not returned if the handStatus was not changed during the call.

The **raiseHandEnabled** parameter is introduced for an administrator to manage the feature for the whole call. It is supported on the following APIs:

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

Parameter	Type/Value	Description/Notes
raiseHandEnabled	**true** | **false**	An administrator can enable or disable the feature for the whole call. By default, the parameter is <**unset**>, but if unset at all levels in the call / callProfile hierarchy, then it defaults to **true**.

- GET on **/callProfiles/<call profile id>**

Response elements	Type/Value	Description/Notes
raiseHandEnabled	**true** | **false**	If set, it returns true or false to indicate whether or not participants are allowed to raise their hands in this call.

- GET on **/calls/<call id>**

Response elements	Type/Value	Description/Notes
raiseHandEnabled	**true** | **false**	If set, it returns true or false to indicate whether or not participants are allowed to raise their hands in this call.

2.18.10 Retrieving call bridge group filter

The callBridgeGroupFilter parameter is introduced on GET on /webBridges.

- GET Enumeration on **/webBridges**

URI parameters	Type/Value	Description/Notes
callBridgeGroupFilter	id	If callBridgeGroupFilter is supplied, only those web bridges within the specified call bridge group will be returned.

2.18.11 Creating, modifying, and retrieving the user's ability to change scope

A new canChangeScope parameter is introduced on:

- POST to **/coSpaces/<coSpace id>/coSpaceUsers**
- PUT to **/coSpaces/<coSpace id>/coSpaceUsers/<coSpace user id>**

Request parameters	Type/Value	Description/Notes
canChangeScope	**true** | **false**	Whether this user is allowed to change the scope of access methods on the coSpace. If this parameter is not supplied in a create (POST) operation, it defaults to **false**.

- GET on **/coSpaces/<coSpace id>/coSpaceUsers/<coSpace user id>**

Response parameters	Type/Value	Description/Notes
canChangeScope	**true** | **false**	Returns true or false to indicate whether this user is allowed to change the scope of access methods on the coSpace.

2.18.12 Creating, modifying, and retrieving an access method

A new **`accessMethod`** parameter is introduced on callLegs and Participants for the following methods:

- POST to **`/calls/<call id>/callLegs`**
- POST to **`/calls/<call id>/participants`**

Parameter	Type/Value	Description/Notes
`accessMethod` (Optional)	ID	Associates the specified **`accessMethod`** as the access method for the callLeg/participant and overrides any default or primary access method on the coSpace.

Note: This parameter is not returned for non-coSpace calls. It is only applicable where the call corresponds to a coSpace call and the ID is an accessMethod within the context of that coSpace.

- GET on **`/callLegs/<callLeg id>`**
- GET on **`/participants/<participant id>`**

Response elements	Type/Value	Description/Notes
`accessMethod`	ID | "coSpace"	• Returns the access method used to join the call on dial in or as set on dial out. Where an **`accessMethod`** is not specified on the POST operation, the GET still returns an access method ID if **`defaultAccessMethod`** was configured on the coSpace or if the primary access method was employed. • The API will return "coSpace" if the coSpace was not joined through an access method. This could be on dial in or on dial out if no **`accessMethod`** or **`defaultAccessMethod`** is specified and no primary access method exists.

2.18.13 Specifying and retrieving a default access method

A new **`defaultAccessMethod`** optional field is introduced on **`/coSpaces/<cospaceId>`** to specify the default access method to be used when dialing out. This field is supported in the following methods:

- PUT to **`/coSpaces/<cospaceId>`**

Parameter	Type/Value	Description/Notes
`defaultAccessMethod` (Optional)	ID | ""	Associates the specified access method as the default access method to be used for dial outs.

- GET on **`/coSpaces/<cospaceId>`**

Response elements	Type/Value	Description/Notes
`defaultAccessMethod`	ID | ""	Associates the specified access method as the default access method to be used for dial outs.

2.18.14 Modifying and retrieving a default access method template

A new **`defaultAccessMethodTemplate`** parameter is introduced for the following methods:

- PUT to **`/coSpaceTemplates/<coSpaceTemplate id>`**

Parameter	Type/Value	Description/Notes
`defaultAccessMethodTemplate`	ID | ""	If specified, associates the access method template as the default one for the coSpace template. When a coSpace is instantiated from the coSpace template, the instantiated default access method template becomes the default access method for the coSpace.

- GET Enumeration on **`/coSpaceTemplates`** 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 <coSpaceTemplates total="N"> tag with potentially multiple <coSpaceTemplate> elements within it.

Each <coSpaceTemplate> tag may include the following element:

Parameter	Type/Value	Description/Notes
defaultAccessMethodTemplate	ID | ""	If specified, associates the access method template as the default one for the coSpace template. When a coSpace is instantiated from the coSpace template, the instantiated default access method template becomes the default access method for the coSpace.

- GET on /coSpaceTemplates/<coSpaceTemplate id> gives the following response:

Response elements	Type/Value	Description/Notes
defaultAccessMethodTemplate	ID | ""	When a coSpace is instantiated from the coSpace template, the instantiated default access method template becomes the default access method for the coSpace.

2.19 Summary of MMP additions and changes

Version 3.2 supports the MMP changes and additions described in this section.

2.19.1 MMP additions

In this release, the new MMP command **webbridge3 https frame-ancestors** is added to Cisco Meeting Server and Cisco Meeting Server 2000. It allows administrators to specify a custom frame-ancestors value to be returned in the **content-security-policy** header allowing the web app to be embedded in other web pages.

Note: In a cluster setup, this command must be configured on all Web Bridges in the deployment.

```
webbridge3 https frame-ancestors <frame-ancestors space-separated
string>
```

```
webbridge3 https frame-ancestors none
```

For example,

```
webbridge3 https frame-ancestors https://*.example.com
https://customdomain.example2.com:8000
```

2.19.2 iframe example for embedded web app

Here is an example of an iframe that embeds the website with the minimum feature policies necessary to let the app run:

```
<iframe src="https://<address>:<port>/" allowusermedia
allow="microphone; camera; encrypted-media; display-
capture;"></iframe>
```

Where Web Bridge 3: **https://<address>:<port>/** is the address of the web bridge.

Note: We recommend using a certificate signed by a public Certificate Authority (CA) with the web app. If a custom certificate is used then the web app may not be visible in the embedded page until you have navigated to the original web app site and accepted the custom certificate.

2.20 Summary of CDR Changes

Version 3.2 introduces the following additions to the Call Detail Records of the Meeting Server:

- A new field **coSpaceMetaDataConfigured** is added to the **callStart** CDR, with possible values of **true** or **false**.
  This is set to **true** when metadata has been configured on **cospaces/<cospace id>/metadata**.
- A new field **confirmationStatus** is added to the **callLegStart** and **callLegUpdate** CDRs, with possible values **required**, **notRequired** or **confirmed**. The value provided determines whether the participant owning the call leg has to confirm, or has already confirmed to join the call, as required by the call out being made with the **confirmation=true** parameter.

2.21 Summary of Event Changes

There are no new Events for version 3.2.

3 Upgrading, downgrading and deploying Cisco Meeting Server software version 3.2.2

This section assumes that you are upgrading from Cisco Meeting Server software version 3.1. If you are upgrading from an earlier version, then Cisco recommends that you upgrade to 3.1 first following the instructions in the 3.1.x release notes, before following any instructions in this Cisco Meeting Server 3.2 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 3.1.

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.

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.

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.

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.

Known issue in 3.2: When configuration backup is taken using the **`backup snapshot <filename>`** command, the private keys get corrupted. Due to this, services using those keys do not run. It is important to take the backup for your private keys manually so that they can be restored after the rollback. To recover the keys after rollback, SFTP the certificate key files to the Meeting Server and then restart all the configured services.

3.1 Before upgrading from version 3.2.x to 3.2.2

Due to a known issues in 3.2.0 and 3.2.1, when the user uses **`backup snaphot <filename>`** command, the private keys that are part of the generated snapshot is corrupted. It is necessary to re-upload the private key files using SFTP to rectify the corrupted key files. So, it is recommended to download the key files for uploading later during the following scenarios:

- When upgrading using an ova file after restoring the Meeting server configuration from the backup file and then manually upload the private key file using SFTP.
- After performing the factory reset, restoring the Meeting server configuration from the backup file and then manually upload the private key file using SFTP.

3.2 Upgrading to Release 3.2.2

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_2_2_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:

   65ecb97f08de9f9faa0943a8331e0dd62a8766094822fb04356c260bc2cc7eac

   `Cisco_Meeting_Server_3_2_2_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:

   4513cd9093e09bfe697bea97d20e4652896363e37774ace86d650235adae181e

   `Cisco_Meeting_Server_3_2_2.ova`

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

   For vSphere6, hash (SHA-512) for Cisco_Meeting_Server_3_2_2_vSphere-6_0.ova file:
   a60313c6ee6de32c00a08400d2f906abccbd2497cb5690b82172fc5def8a6d727759c352d05b8b3bb67c333
   2cc533a0a3f57c09a313f996c9db7ab77ac6f7a1e

   For vSphere6.5 and higher, hash (SHA-512) for Cisco_Meeting_Server_3_2_2_vSphere-6_5.ova file:
   3731e1d5cf569878b0a1150eaa6de5793f237edc0428da5cec4d3d2301be598fc1bf92e7dd87ecc91d687a89
   428094544d6a1db207f625554ee172be71e248c0

2. To validate the OVA file, the checksum for the 3.2 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.

Note: After upgrade, create a new backup file using **`backup snapshot <filename>`** command.

3.3 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 cms.lic 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.

3.4 Cisco Meeting Server 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.

3 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 `3.2`.
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.

3.5 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 3.2.2.

Cisco identifier	Summary
CSCvz24378	In version 3.x, for Meeting Server deployments with Skype and web app and Expressway as TURN server type, if the parameter **tcpPortNumberOverride** is not specified, the TCP port does not default to 443 and the web app service does not open.
CSCvz24375	In version 3.2, when configuration backup is taken using the **backup snapshot <filename>** command, the private keys get corrupted. Due to this, services using those keys do not run displaying the message "key file not found".
CSCvy67773	Due to high packet loss from the incoming audio stream, the participants hear a white noise during web app call.

Cisco identifier	Summary
[CSCvy61122]	The web app users are unable to join conference calls getting an error message “System is Currently Unavailable”.
[CSCvy59876]	If a participant during a web app call shares the screen for longer than 30 minutes and then shares it again later during the same call, then the screen is not visible to other participants in the call.
[CSCvy52991]	In rare cases, a peer link call in a cluster fails to establish, if the call object with no active `callLegs` may fail to be removed.
[CSCvw62217]	In rare cases, Cisco Meeting Server 2000 crashes while processing user requests with the message "guest request 4005477866 failed: operation destroyed while request in progress"in the syslog file.
[CSCvy21678]	Cisco Meeting Server ends all the peer calls (including the existing calls) for the respective cospace, if the participants exceed the call limit set in the call profile.
[CSCvx64849]	In a scenario where the Meeting Server has a customized layout and Meeting Management applies pane placement, on a video endpoint with dual monitor setup, the second monitor quickly flashes the image from 13th participant and then reverts to the background screen during the pane placement.

Issues seen in previous versions that are fixed in 3.2.1.

Cisco identifier	Summary
[CSCvy47970]	When listing IVR numbers, in case label is not provided an colon symbol is appearing in the IVR number.
[CSCvy22987]	When running the TURN Server, the log files are spammed with COUNTS+ 1 turn_report_session_usage log messages.
[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. Cisco has evaluated the impact of the vulnerability on this product and concluded that the product is affected by: • CVE-2021-3449: could allow a remote unauthenticated attacker to crash a TLS server resulting in a Denial of Service (DoS) condition. However, the product is not affected by: • 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.

Cisco identifier	Summary
[CSCvy07110]	In a web app call with three participants, after the first participant has presented and the second participant tries to present, the third participant sees a white background instead of the presentation. This issue was detected in Google Chrome and Microsoft Edge.
[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.

Issues seen in previous versions that are fixed in 3.2.

Cisco identifier	Summary
[CSCvw61465]	Web Bridge 3 to C2W stops trying to establish a connection after 300 DNS lookup failures.
[CSCvw61470]	The SSO domain is case-sensitive (it should not be case-sensitive).
[CSCvw61548]	TURN logs do not show current session counts accurately.
[CSCvi67053]	From 3.2 onwards, it is no longer possible to set a **min_password_age** greater than the **password_age**. If the **min_password_age** parameter to the user rule command is larger than the **password_age** parameter, it is not possible for that MMP user's password to be changed. If there is only one MMP user account (the admin account), then no logins are possible and theMeeting Server will need to be redeployed.
[CSCvx93381]	Unexpected Call Bridge restarts on a Meeting Server 2000 did not generate a syslog message alerting the administrator that the restart had occurred.
[CSCvx14793]	A crash could occur when accessing objects under /multipartyLicensing/activePersonalLicenses in API explorer, or when using direct API calls for /users/GUID of users under the same API path.
[CSCvw91670]	Participant labels were not appearing in the recorded video using CMS Recorder.

3.6 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
CSCvy02403	Backup rollback fails on Meeting Server 3.2. There is an issue when XMPP recorder settings were previously used prevented the backup restore.
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.

3.6.1 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.

4 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

5 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

Cisco Trademark