

Cisco TelePresence MCU API 2.10

Product Programming Reference Guide

D14997

January 2013

Contents

Introduction	5
API History	5
XML-RPC implementation	5
Transport protocol	6
Clustering	6
Considering API overhead when writing applications	6
Overview	7
Encoding	8
Authentication	8
Message flow	9
Participant identification	10
Enumerate methods	11
Enumerate filters	11
Revision numbers	12
Feedback	15
Feedback receivers	16
Feedback messages	17
Feedback events	18
API commands	20
Deprecated commands	23
addressBookEntry.enumerate	24
auditlog.delete	30
auditlog.query	31
autoAttendant.destroy	32
autoAttendant.enumerate	33
autoAttendant.status	34
cdrlog.delete	35
cdrlog.enumerate	36
cdrlog.query	38
conference.create	39
conference.destroy	44
conference.end	45
conference.enumerate	46
conference.floor.modify	53
conference.floor.query	54
conference.metadata.modify	55
conference.metadata.status	56
conference.modify	57
conference.paneplacement.modify	61
conference.paneplacement.query	63
conference.resetCleanupTimeout	65
conference.status	66
conference.streaming.modify	71
conference.streaming.query	72
conferenceme.modify	75
conferenceme.query	76
device.content.modify	77

device.content.query	78
device.encryption.modify	79
device.encryption.query	80
device.health.query	81
device.network.modify	82
device.network.query	84
device.query	87
device.restart	89
device.restartlog.query	90
device.status	91
device.time.modify	92
device.time.query	93
feedbackReceiver.configure	94
feedbackReceiver.query	95
feedbackReceiver.reconfigure	96
feedbackReceiver.remove	97
gatekeeper.modify	98
gatekeeper.query	100
gateway.enumerate	102
participant.add	103
participant.connect	107
participant.diagnostics	108
participant.disconnect	111
participant.enumerate	112
participant.enumerate (deprecated)	121
participant.fecc	125
participant.message	126
participant.modify	127
participant.move	131
participant.remove	132
participant.statistics	133
participant.status	139
participant.status (deprecated)	147
route.add	151
route.delete	152
route.enumerate	153
route.preferences.modify	154
route.preferences.query	155
services.modify	156
services.query	158
sip.modify	159
sip.query	160
streaming.modify	161
streaming.query	162
template.create	163
template.delete	167
template.enumerate	168
template.modify	172
template.status	176
Related information	180
system.xml file	181

Fault codes	183
Disconnect reasons	185
HTTP keep-alives	187
Conference layouts	188
Linking conferences across MCUs	190
Index of parameters	195
Index of parameters: A	196
Index of parameters: B	205
Index of parameters: C	206
Index of parameters: D	220
Index of parameters: E	224
Index of parameters: F	227
Index of parameters: G	230
Index of parameters: H	232
Index of parameters: I	235
Index of parameters: J	239
Index of parameters: L	240
Index of parameters: M	244
Index of parameters: N	248
Index of parameters: O	251
Index of parameters: P	252
Index of parameters: Q	257
Index of parameters: R	258
Index of parameters: S	265
Index of parameters: T	269
Index of parameters: U	272
Index of parameters: V	274
Index of parameters: W	279
API Change history	280
Version 2.10 changes	281
Version 2.9 changes	283
Version 2.8 changes	291
References	295

Introduction

This document accompanies the latest version of the remote management API for the Cisco TelePresence MCU software (respectively referred to as API and MCU in this document). The following Cisco TelePresence products support this API when they are running MCU version 4.4 and later:

- Cisco TelePresence MCU 4200 Series
- Cisco TelePresence MCU 4500 Series
- Cisco TelePresence MCU 5300 Series
- Cisco TelePresence MCU MSE 8420
- Cisco TelePresence MCU MSE 8510

API History

The following table shows the device's software versions and the corresponding supported API versions:

API version	MCU version
2.10 (this version)	4.4 and later
2.9	4.3 and later
2.8	4.2 and later
2.7	4.1 and later

XML-RPC implementation

API calls and responses are implemented using the XML-RPC protocol. This simple protocol does remote procedure calling using HTTP (or HTTPS) as the transport and XML as the encoding. It is extremely simple although it does still allow for complex data structures. XML-RPC is stateless and is not platform-dependent; it was chosen in favor of SOAP (Simple Object Access Protocol) because of its simplicity.

Your application must either regularly poll the device or continually listen to the device - if it is configured to publish feedback events - if you want it to monitor the device's activity.

The API implements all parameters and returned data as `<struct>` elements, each of which is explicitly named. For example, `device.query` returns (amongst other data) the current time as:

```
<member>
  <name>currentTime</name>
  <value><dateTime.iso8601>20110121T13:31:26</dateTime.iso8601></value>
</member>
```

rather than simply

<dateTime.iso8601>20110121T13:31:26<dateTime.iso8601>

Note: Unless otherwise stated, assume strings have a maximum length of 31 characters. Signed 32 bit integers are used, hence a maximum value of 2147483647 is accepted or returned for integer parameters.

Refer to the [XML-RPC specification](#)^[1] for more information.

Transport protocol

The device implements HTTP/1.1 as defined by [RFC 2616](#)^[2]. It expects to receive communications over TCP/IP connections to port 80 (default HTTP port) or port 443 (default HTTPS port).

Your application should send HTTP POST messages to the URL defined by path `/RPC2` on the device's IP address, for example `https://10.0.0.53/RPC2`.

You can configure the device to receive HTTP and HTTPS on non-standard TCP port numbers if necessary, in which case append the non-standard port number to the IP address.

Clustering

From version 4.1 of the MCU software onwards you can configure MCU blades in a cluster in order to increase the maximum number of conference participants. One MCU acts as a master controlling up to two slave MCUs.

The MCU 5300 Series can be stacked, to a maximum of two appliances per stack, with one appliance acting as master for the stack.

Considering API overhead when writing applications

Every API command that your application sends incurs a processing overhead within the device's own application. The exact amount of overhead varies widely with the command type and the parameters sent. It is important to bear this in mind when designing your application's architecture and software. If the device receives a high number of API commands every second, its overall performance could be seriously impaired – in the same way that it would be if several users accessed it simultaneously via the web interface.

The current implementation of the MCU API will accept a maximum of four concurrent XML RPC requests and is limited to a maximum of eight concurrent TCP connections.

For this reason, the best architecture is a single server running the API application and sending commands to the device. If multiple users need to use the application simultaneously, provide a web interface on that server or write a client that communicates with the server. The server would then manage the clients' requests and send API commands directly to the device. Implement some form of control in the API application on your server to prevent the device being overloaded with API commands. This provides much more control than having the clients send API commands directly and will prevent the device's performance being impaired by unmanageable numbers of API requests.

Furthermore, the API is designed to have as little impact as possible on the network when responding to requests. The device's responses do not routinely include data that is not relevant, or empty data structures where the data is not available. Your application should take responsibility for checking whether the response includes what you expected, and you should design it to gracefully handle any situations where the device does not respond with the expected data.

Overview

- Encoding 8
- Authentication 8
- Message flow 9
- Participant identification 10
- Enumerate methods 11
- Enumerate filters 11
- Revision numbers 12

Encoding

Your application can encode messages as ASCII text or as UTF-8 Unicode. If you do not specify the encoding, the API assumes ASCII encoding. You can specify the encoding in a number of ways:

Specify encoding with HTTP headers

There are two ways of specifying UTF-8 in the HTTP headers:

- Use the **Accept-Charset: utf-8** header
- Modify the **Content-Type** header to read **Content-Type: text/xml; charset=utf-8**

Specify encoding with XML header

The `<?xml>` tag is required at the top of each XML file. The API will accept an encoding attribute for this tag; that is, `<?xml version="1.0" encoding="UTF-8"?>`.

Authentication

The application must authenticate itself to the MCU. Also, because the interface is stateless, the application must authenticate with the MCU every time it issues a command to the API.

Unless the device is configured to allow (or require) certificate-based login, all messages must contain a user name and password as follows:

Parameter name	Type	Short description
<code>authenticationUser</code>	string	Name of a user with sufficient privilege for the operation being performed. The name is case sensitive.
<code>authenticationPassword</code>	string	The password that corresponds with the given <code>authenticationUser</code> . The API ignores this parameter if the stored user has no password.

Note: Authentication information is sent using plain text and should only be sent over a trusted network.

Certificate-based authentication modes

Client certificate security option	API authentication rules
Not required	No effect on API.
Verify certificate	Messages must have valid username and password values (<code>authenticationUser</code> and <code>authenticationPassword</code> parameters). To successfully make an HTTPS connection, the messages must also contain a valid client certificate that was issued by an authority that the MCU trusts.

Certificate-based authentication allowed	<p>If the common name in the client certificate matches a username in the device configuration file, the API request is allowed access with the privileges assigned to that username. Messages do not need username and password values, which are ignored if present.</p> <p>If the common name does not match a username, all messages must include valid username and password values.</p>
Certificate-based authentication required	<p>Any username and password fields in the messages are always ignored. If the common name in the client certificate matches a username in the device configuration file, the API request is logged in with the privileges assigned to that username. If the common name does not match a username, the API request is rejected.</p>

Message flow

The application initiates the communication and sends a correctly formatted XML-RPC command to the device.

Example command

```
<?xml version='1.0' encoding='UTF-8'?>
  <methodCall>
    <methodName>recording.delete</methodName>
    <params>
      <param>
        <value>
          <struct>
            <member>
              <name>authenticationPassword</name>
              <value><string></string></value>
            </member>
            <member>
              <name>recordingId</name>
              <value><int>101</int></value>
            </member>
            <member>
              <name>authenticationUser</name>
              <value><string>admin</string></value>
            </member>
          </struct>
        </value>
      </param>
    </params>
  </methodCall>
```

Assuming the command was well formed, and that the device is responsive, the device will respond in one of these ways:

- With an XML **methodResponse** message that may or may not contain data, depending on the command.
- With an XML **methodResponse** that includes only a fault code message.

Example success

```
<?xml version="1.0"?>
  <methodResponse>
```

```

<params>
  <param>
    <value>
      <struct>
        <member>
          <name>status</name>
          <value>
            <string>operation successful</string>
          </value>
        </member>
      </struct>
    </value>
  </param>
</params>
</methodResponse>

```

Example fault code

```

<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value>
            <int>1</int>
          </value>
        </member>
        <member>
          <name>faultString</name>
          <value>
            <string>method not supported</string>
          </value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>

```

Participant identification

The following parameters uniquely identify a particular participant for the purposes of many MCU API calls.

When reading or modifying the parameters of a specific endpoint, you must supply `participantName`, `participantProtocol` and `participantType`, along with either a `conferenceName` or an `autoAttendantUniqueId`.

You can use `participant.enumerate` to retrieve these parameters.

Parameter name	Type	Short description
<code>participantName</code>	string	The unique name of a participant.
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> .
<code>conferenceName</code>	string	The name of the conference.

If the participant is in a conference, the call may require the `conferenceName`; if the participant is in an autoattendant, the call may require the `autoAttendantUniqueId` instead. The call will not require both parameters.

<code>autoAttendantUniqueId</code>	string	Unique identifier for the auto attendant.
------------------------------------	--------	---

Enumerate methods

Enumerate methods have the potential to return a large volume of data, so these calls have a control mechanism to limit the number of enumerated items per call.

Each enumerate call may take and return an `enumerateID` parameter which tells the API or calling application where to start the enumeration. The mechanism works as follows:

1. The application calls an enumerate method without an `enumerateID` parameter.
2. The device returns an array containing the enumerated items, and possibly an `enumerateID`. The response will always include an `enumerateID` if the device enumerated more items than it included in the response.
3. If there is an `enumerateID`, the application should call the enumerate method again, supplying the `enumerateID` as returned by the previous call.
4. The application should repeat this process until the response fails to include an `enumerateID`. This means that the enumeration is complete.

Note: Do not supply your own `enumerateID` values; make sure you only use the values returned by the device.

Enumerate filters

Enumerate methods will accept an optional `enumerateFilter` parameter, which allows you to filter the response. The parameter must contain a filter expression, which is built from criteria and operators.

The filter criteria that a call will accept vary depending on the call, but the syntax for using those criteria in expressions is the same for all methods that allow filtering. The reference information for methods that allow filtering includes acceptable filter criteria.

If the filter expression evaluates to true for the enumerated item, the item will be included in the device's response. If the expression evaluates false, the enumerated item will be filtered out of the response.

Filter expressions consist of atomic expressions combined with operators and parentheses. Whitespace is ignored. Functions are valid, and any parameters are in a comma separated list in parentheses after the function name, for example, `function(expression1,expression2)`.

For example, if the expression `(inProgress && internal)` is used to filter the response to `recording.enumerate`, the returned array of recordings will only include those which are both `inProgress` and `internal`.

The integer 0 evaluates to false and all other integers to true. Integers can be expressed using any string of valid digits. Prefix hex digits with `0x`, decimal with `0t` and binary with `0z`. The API assumes decimal if you don't supply a prefix.

Binary operators

The following binary operators are valid, in order of priority (lowest priority first):

Operator	Description
	Boolean or
&&	Boolean and
	Bitwise or
^	Bitwise exclusive or
&	Bitwise and
==	Equality
!=	Inequality
<	Less than
<=	Less than or equal
>=	Greater than or equal
>	Greater than
<<	Bitwise left shift
>>	Bitwise right shift
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo

Unary operators

The following unary operators are valid. All of these bind tighter than any binary operator.

Operator	Description
-	Unary minus
+	Unary plus
!	Logical negation
~	Bitwise negation

Revision numbers

To reduce the size of responses when querying the device, some of the enumeration methods support a revision number system.

When the device responds to a call that supports revision numbers, it returns an extra integer field called **currentRevision**. For example:

```
<member>
  <name>currentRevision</name>
```

```
<value>
  <int>18</int>
</value>
</member>
```

The revision number increases every time any API query is made on the device. To reduce the size of subsequent query responses, you may pass in the `lastRevision` parameter. For example:

```
<member>
  <name>lastRevision</name>
  <value>
    <int>18</int>
  </value>
</member>
```

The device returns only those records that have changed since `lastRevision`. For example, if you provide a `lastRevision` parameter in a `connection.enumerate` call, the device's enumeration response only includes connections that changed since its revision number was set to the value you provided.

Using revision numbers with enumerate methods

When you use revision numbers with enumerate methods, you should use the same value of the `lastRevision` parameter for each stage of the enumeration, *despite that a newer `currentRevision` parameter is returned at each stage*. If you update `lastRevision` to use the newer `currentRevision`, the device will not return the rest of the changes you were interested in; it will only look for changes since you started the enumeration.

Similarly, if you want to store a new value to use as `lastRevision` in a future enumeration, you should use the `currentRevision` number that the device returned in the first response to your current enumeration. You need to do this to ensure that your future enumeration catches any changes that that happen while you are doing the current enumeration. However, it does mean that occasionally a record is reported more than once.

Discovering record removal

The problem with the revision number feature only returning changed records is that the calling application can't tell whether a record has been *removed altogether*.

One approach to solving this problem is the `listAll` parameter, which a client application may set to `true` to tell the device to return every record available. This allows the client to synchronize with the device because it can safely assume that any record not returned by this request (or series of requests, in the case of enumerations) no longer exists on the device.

For example, you can assume that any connections not returned by `connection.enumerate` when `listAll` is set to `true` have been removed from the device.

You can use the `listAll` parameter in conjunction with the `lastRevision` parameter. In this case, the device returns every record it has but may remove data from members whose records have not changed since `lastRevision`. The API inserts a parameter named `changed` instead, with its value set to `false`; the calling application can ignore those members because they haven't changed since `lastRevision`, and the response is still much smaller than it would otherwise be with `listAll`.

Dead records

Another approach to the record removal problem is the **dead** parameter. The device maintains a cache of records that have been removed and are no longer considered active in any sense. It will return the **dead** parameter, with value **true**, instead of those records if those records would otherwise have been required by the response.

The device will never return a dead record unless revision numbers are being used. The device will also never return a dead record if **listAll** is set to **true**.

Furthermore, dead records are only cached for a few minutes.

The device only returns a dead record under the following conditions:

- **listAll** is not set, or is set **false**
- The call supports revision numbers and **lastRevision** is supplied
- The record was removed at some point after the supplied **lastRevision**
- That record has not yet been cleared from the cache.

When these conditions are met, the query response includes the minimum of information required to identify the record as well as the **dead** parameter, set to **true**. The calling application can safely assume that the device will soon remove any trace of this record.

However, unless the client is doing frequent, regular polling, we recommend using the **listAll** parameter, as described above, to verify removed records.

Feedback

- Feedback receivers 16
- Feedback messages 17
- Feedback events 18

Feedback receivers

The API allows you to register your application as a feedback receiver. This means that the application doesn't have to constantly poll the device if it wants to monitor activity.

The device publishes events when they occur. If the device knows that your application is listening for these events, it will send XML-RPC messages to your application's interface when the events occur.

- Use [feedbackReceiver.configure \[p.94\]](#) to register a receiver to listen for one or more [feedback events](#).
- Use [feedbackReceiver.query \[p.95\]](#) to return a list of receivers that are configured on the device.
- Use [feedbackReceiver.reconfigure \[p.96\]](#) to change the configuration of an existing feedback receiver.
- Use [feedbackReceiver.remove \[p.97\]](#) to remove an existing feedback receiver.

After registering as a feedback receiver, the application will receive [feedback messages](#) on the specified interface.

Feedback messages

The feedback messages follow the format used by the device for XML-RPC responses.

The messages contain two parameters:

- **sourceIdentifier** is a string that identifies the device, which may have been set by **feedbackReceiver.configure** or otherwise will be the device's MAC address.
- **events** is an array of strings that contain the names of the feedback events that have occurred.

Example feedback message

```
<params>
  <param>
    <value>
      <struct>
        <member>
          <name>sourceIdentifier</name>
          <value><string>00D7C000C66</string></value>
        </member>
        <member>
          <name>events</name>
          <value>
            <array>
              <data>
                <value><string>restart</string></value>
              </data>
            </array>
          </value>
        </member>
      </struct>
    </value>
  </param>
</params>
```

Feedback events

The following table lists the feedback events that the MCU can publish.

Event	Description
restart	The source publishes this event when it starts up.
configureAck	The source publishes this event to acknowledge that an application has successfully configured a feedback receiver.
networkChanged	Any change in IP, Ethernet or DNS configuration or status will trigger this. The feedback device should then poll <code>device.network.query</code> .
servicesChanged	Will be sent whenever a setting in <code>device.services.query</code> changes. Note that this is only generated when configuration changes and does not reflect a change in the actual bind status. Will generate a feedback message for each interface.
routesChanged	Will be sent whenever a setting in <code>device.routes.query</code> changes.
deviceStatusChanged	This event will be generated whenever an MCU is shutdown, the <code>bootComplete</code> or when <code>rebootRequired</code> changes. Also sent if a feature key is added or removed. All of these should result in a <code>device.query</code> being issued.
rebooting	Should be sent just before the device restarts. Should not be relied upon because it won't be sent if the box crashes.
timeChanged	Will be sent whenever a setting in <code>device.time.query</code> changes or whenever the time is changed manually (NTP updates shouldn't be covered as they should happen frequently with little/no noticeable change).
conferenceStarted	One or more conferences have been created.
conferenceFinished	One or more conferences have been deleted.
participantJoined	One or more participants have joined a conference.
participantLeft	One or more participants have left a conference.
conferenceConfigurationChanged	This event is generated when the active parameters of one or more ad hoc or scheduled conferences have changed. This includes changes to the conference name, streaming, H.239, privacy, chair control and custom layout.
autoAttendantStarted	An auto attendant has started.
autoAttendantChanged	A participant moved from one auto attendant to another.
autoAttendantFinished	An auto attendant has finished.
participantConnected	One or more participants have connected to the MCU.
participantDisconnected	One or more participants disconnected from the MCU.
participantAudioMuteChanged	One or more participants changed their audio mute setting.
participantVideoMuteChanged	One or more participants changed their video mute setting.
participantAudioRemoteMuteChanged	One or more participants changed their remote audio mute setting.

Event	Description
importanceChanged	A participant's important status changed; either the participant has been made important or has stopped being important.
activeSpeakerChanged	The loudest speaker has changed in one or more conferences.
sipChanged	The source publishes this event when a SIP parameter changes (parameters as returned by sip.query).
h323Changed	A change of any parameter returned in gatekeeper.query will result in this event being returned, including h323IdStatus and mcuServicePrefixStatus but excluding the number of registrations.
floorChanged	This event will be returned when floor status for a conference changes. This should result in a conference.enumerate being issued by the feedback device.
chairChanged	This event will be returned when the chair for a conference changes. This should result in a conference.enumerate being issued by the feedback device.
encryptionChanged	Will be sent whenever a setting in device.encryption.query changes.
contentChanged	Will be sent whenever a setting in device.content.query changes.
streamingChanged	Will be sent whenever anything returned in the streaming.query command changes.
conferenceMeChanged	Will be sent whenever anything returned in the conferenceme.query command changes.

API commands

This section contains a reference to each of the API calls supported by the MCU.

The calls are grouped alphabetically by the objects which they query or modify. The following information is provided for each call:

- Description of the call's effect
- Accepted parameters, and whether they are required or optional
- Returned parameters, and whether they are always or conditionally returned
- Deprecated parameters

Note: In some cases, parameter names are the same even though the parameters are used in different contexts. To avoid ambiguity, these parameters have an extra word of explanation next to their names. For example, the parameter **type** is used in several contexts and thus appears in the document as **type (service)**, **type (pane)**, or **type (event)**.

Deprecated commands	23
addressBookEntry.enumerate	24
auditlog.delete	30
auditlog.query	31
autoAttendant.destroy	32
autoAttendant.enumerate	33
autoAttendant.status	34
cdrlog.delete	35
cdrlog.enumerate	36
cdrlog.query	38
conference.create	39
conference.destroy	44
conference.end	45
conference.enumerate	46
conference.floor.modify	53
conference.floor.query	54
conference.metadata.modify	55
conference.metadata.status	56
conference.modify	57
conference.paneplacement.modify	61
conference.paneplacement.query	63
conference.resetCleanupTimeout	65
conference.status	66
conference.streaming.modify	71
conference.streaming.query	72

conferenceme.modify	75
conferenceme.query	76
device.content.modify	77
device.content.query	78
device.encryption.modify	79
device.encryption.query	80
device.health.query	81
device.network.modify	82
device.network.query	84
device.query	87
device.restart	89
device.restartlog.query	90
device.status	91
device.time.modify	92
device.time.query	93
feedbackReceiver.configure	94
feedbackReceiver.query	95
feedbackReceiver.reconfigure	96
feedbackReceiver.remove	97
gatekeeper.modify	98
gatekeeper.query	100
gateway.enumerate	102
participant.add	103
participant.connect	107
participant.diagnostics	108
participant.disconnect	111
participant.enumerate	112
participant.enumerate (deprecated)	121
participant.fecc	125
participant.message	126
participant.modify	127
participant.move	131
participant.remove	132
participant.statistics	133
participant.status	139
participant.status (deprecated)	147
route.add	151
route.delete	152
route.enumerate	153
route.preferences.modify	154
route.preferences.query	155
services.modify	156
services.query	158
sip.modify	159
sip.query	160
streaming.modify	161
streaming.query	162
template.create	163
template.delete	167
template.enumerate	168
template.modify	172

template.status 176

Deprecated commands

The following commands were supported in earlier versions of the MCU API but have since been superseded.

Deprecated command	Superseded by this command in newer versions
<code>conference.participant.add</code>	participant.add [p.103]
<code>conference.participant.modify</code>	participant.modify [p.127]
<code>conference.participant.remove</code>	participant.remove [p.132]
<code>conference.query</code>	conference.enumerate [p.46] , participant.enumerate [p.112]
participant.enumerate (deprecated) [p.121]	participant.enumerate [p.112] . This call is not technically deprecated, but there is deprecated behavior if the call does not provide the <code>operationScope</code> parameter.
participant.status (deprecated) [p.147]	participant.status [p.139] . This call is not technically deprecated, but there is deprecated behavior if the call does not provide the <code>operationScope</code> parameter.
<code>system.query</code>	conference.enumerate [p.46] , device.query [p.87]
participant.diagnostics [p.108]	participant.statistics [p.133] . The <code>participant.diagnostics</code> call will continue to work as it did in MCU 4.1 to ensure backwards compatibility with third party products.

addressBookEntry.enumerate

Enumerates the configured endpoints on the MCU. Each struct in the **addressBookEntries** array represents a known endpoint, and details its call in parameters and conferencing parameters in nested structures.

Input parameters

Optional or conditional inputs

Parameter name	Type	Short description
enumerateID	string	The device returns this index if the requested data is too large for one response. Pass this parameter in a repeat of the call to return the next batch of data. more...

Returned data

Conditionally returned

If there are entries to return, the method returns them in an array. If there are more entries than can be returned in one response, you'll get the next **enumerateID** up from the one you provided.

Parameter name	Type	Short description
enumerateID	string	The device returns this index if the requested data is too large for one response. Pass this parameter in a repeat of the call to return the next batch of data. more...
addressBookEntries	array	Each array member is a struct representing a single addressbook entry.
name (<i>endpoint</i>)	string	The name of the endpoint.
address (<i>endpoint</i>)	string (63)	The address of the endpoint; may be hostname, IP address, E.164 number, SIP URI, or H.323 ID.
protocol (<i>signaling</i>)	string	The signaling protocol used in the call. One of h323 , sip , or vnc .
gatewayName	string	Present in entries for H.323 endpoints which are configured to use a gateway. This name corresponds to the name parameter of a gateway returned by gateway.enumerate .
gatewayAddress	string (63)	The address of an H.323 gateway, if required. Only used if protocol is h323 . This corresponds to the address parameter of the gateway as returned by gateway.enumerate .

dtmfSequence	string (127)	A string of characters that will be converted to DTMF signals, allowing the device to navigate through audio menus. The sequence may contain 0–9, *, #, and ,. The comma becomes a two second pause. more...
useSIPRegistrar	boolean	Not valid unless the protocol is SIP. true if the endpoint uses the SIP registrar. Defaults to false .
password	string	The password for VNC endpoints.
portNumber	integer	The port number for VNC endpoints.
callInParams	struct	A structure containing the call in parameters of the endpoint. These parameters are used to match incoming calls to pre-configured participants. For a positive match, a participant must match fields which have values. Blank fields are not considered in the comparison.
name (endpoint)	string	The name of the endpoint.
address (endpoint)	string (63)	The address of the endpoint; may be hostname, IP address, E.164 number, SIP URI, or H.323 ID.
e164	string	An E.164 number.
conferencingParameters	struct	A structure containing the conferencing parameters of the enumerated item, e.g. gateway or endpoint.
useDefaultMotionSharpness	boolean	true means this endpoint will use box-wide default motion sharpness settings.
minFrameRateMotionSharpness	integer	Specifies the minimum frame rate for this endpoint. This parameter is only present if useDefaultMotionSharpness is false .
useDefaultVideoTransmitResolutions	boolean	true means this endpoint will use box-wide default video transmit resolutions.
videoTransmitResolutions	string	Overrides the default setting for video resolution the MCU may send to the endpoint. One of allowAll , 4to3Only , 4to3WidescreenOverride , or 16to9Only . more...
maxMediaTxBitRate	integer	The maximum media transmission speed from this device, in kbps. 0 means the device uses the default.
maxMediaRxBitRate	integer	The maximum media reception speed of this device, in kbps. 0 means the device uses the default.
defaultLayout	string	Describes the participant's default conference view layout if configured. One of default , familyIndex , layoutIndex , conferenceCustom . more...
layoutControlDefault	boolean	true means the endpoint inherits the default layout control setting. more...

<code>layoutControlEnabled</code>	boolean	Deprecated by <code>layoutControlEx</code> . Defines whether the endpoint's participant will have control over the layout if <code>layoutControlDefault</code> is <code>false</code> . more...
<code>layoutControlEx</code>	string	Defines how the view layout can be controlled. One of <code>disabled</code> , <code>feccOnly</code> , <code>dtmfOnly</code> , <code>feccWithDtmfFallback</code> , or <code>bothFeccAndDtmf</code> . more...
<code>cameraControlDefault</code>	boolean	<code>true</code> means the endpoint uses the default camera control setting of the conference or template. <code>false</code> means the endpoint explicitly sends another type of camera control to participants.
<code>cameraControl</code>	string	Defines how the endpoint camera(s) within your API call's context can be controlled. If present, it may be <code>disabled</code> , <code>feccOnly</code> , <code>dtmfOnly</code> , <code>feccWithDtmfFallback</code> , <code>bothFeccAndDtmf</code> , or <code>default</code> . more...
<code>h239ContributionDefault</code>	boolean	Defines whether or not the endpoint will use the box-wide H.239 contribution setting.
<code>h239ContributionEnabled</code>	boolean	Defines whether or not the endpoint will be able to contribute H.239, if <code>h239ContributionDefault</code> is <code>false</code> .
<code>h239Negotiation</code>	string	Defines how the MCU presents itself for h239 token negotiation. One of <code>As master</code> , <code>As slave</code> , or <code>Mimic slave</code> . more...
<code>contentReceive</code>	boolean	<code>true</code> if the endpoint is allowed to receive a separate content stream when participating in a conference.
<code>initialAudioMuted</code>	boolean	<code>true</code> if the endpoint's audio is initially muted.
<code>initialVideoMuted</code>	boolean	<code>true</code> if the endpoint's video is initially muted.
<code>audioRxGainMode</code>	string	<code>none</code> , <code>automatic</code> , <code>default</code> , or <code>fixed</code> . more...
<code>audioTxMuted</code>	boolean	<code>true</code> means that the MCU is not transmitting the audio part of the conference to this participant.
<code>videoTxMuted</code>	boolean	<code>true</code> means that the MCU does not send the video part of the conference to this participant.
<code>autoDisconnect</code>	boolean	<code>true</code> allows the device to automatically disconnect the endpoint, and all remaining endpoints that have this property, when none of the remaining endpoints require manual disconnection. <code>false</code> means this endpoint requires manual disconnection. When a participant disconnects from a conference and only participants who have <code>autoDisconnect</code> set to <code>true</code> remain, the MCU disconnects all the remaining participants.

borderWidth	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. more...
addAsGuest	boolean	Defines whether the MCU designates guest or chair status to the participant when it invites the participant in to the conference. true means the participant joins as a guest when invited in; false means the participant joins as a chair when invited in.
actAsRecorder	boolean	Defines whether this participant appears as a recorder to other participants.
displayNameOverrideStatus	boolean	true if the endpoint uses the displayNameOverrideValue text to identify itself to other participants.
displayNameOverrideValue	string	This value overrides the participant's display name if displayNameOverrideStatus is true .
suppressAudioDuringDTMF	string	outgoing or all defines which audio the MCU suppresses while it sends the DTMF connection sequence to the endpoint. more...
suppressDtmfEx	string	Controls the muting of in-band DTMF tones. One of fecc , always , or never . more...
videoToUse	struct	Collection of parameters that uniquely identify the participant whose video will display in place of this participant's video by default.
name (endpoint)	string	The name of the endpoint.
protocol (signaling)	string	The signaling protocol used in the call. One of h323 , sip , or vnc .
customCodecSelection	boolean	Indicates whether the device advertises a custom set of codecs.
customCodecs	struct	A collection of structs that indicate which codecs the device advertises that it can use to send and receive audio and video. The struct is absent if customCodecSelection is false .
audioTx	struct	A choice of audio codecs advertised by the MCU.
audioRx	struct	A choice of audio codecs received from the participant's endpoint.
g711	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.
g722	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

g722.1	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.
g722.1c	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.
g723.1	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.
g728	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.
g729	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.
siren14	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.
aac-ld	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.
aac-lc	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.
videoTx	struct	A choice of video codecs advertised by the MCU.
videoRx	struct	A choice of video codecs received from the participant's endpoint.
h261	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.
h263	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.
h263+	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.
h263i	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.
h264	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

auditlog.delete

Deletes entries from the device's audit log.

Input parameters

Parameter name	Type	Short description
<code>deleteIndex</code> (<i>audit log</i>)	integer	You can delete logs in chunks of 400. To delete logs, you can enter the value returned by <code>auditlog.query.deleteableIndex</code> . This will delete all complete chunks (400 logs each) below this number, leaving the residuals. Alternatively, you can delete less than this amount by picking a number below the value of <code>deleteableIndex</code> . This will delete all complete chunks (400 logs) below that number, leaving any residuals.

auditlog.query

Queries the device for statistics about the audit log.

Returned data

Parameter name	Type	Short description
<code>firstIndex</code>	integer	The index of the oldest stored event.
<code>deletableIndex</code>	integer	The log index of the most recent event that was archived into a log file. The delete command works on whole files, so you can delete up to the last event that went into a file.
<code>numEvents</code> (<i>audit log</i>)	integer	The total number of events stored.
<code>percentageCapacity</code>	integer	The percentage of the total available capacity being used by the log.

autoAttendant.destroy

This call destroys an auto attendant.

Input parameters

Parameter name	Type	Short description
autoAttendantUniqueID	string	Unique identifier for the auto attendant.

autoAttendant.enumerate

Input parameters

Optional or conditional inputs

The call has no valid enumerate filter expressions.

Parameter name	Type	Short description
<code>enumerateID</code>	string	The device returns this index if the requested data is too large for one response. Pass this parameter in a repeat of the call to return the next batch of data. more...
<code>lastRevision</code>	integer	This number identifies an earlier set of enumeration data to compare against your current call. If you supply this parameter using the <code>currentRevision</code> value returned by a previous enumeration, the current <code>enumerate</code> call will return only the differences since that previous call. If you don't supply this parameter, the device assumes that you want a full enumeration.

Returned data

Conditionally returned

If there are entries to return, the method returns them in an array. If there are more entries than can be returned in one response, you'll get the next `enumerateID` up from the one you provided.

Parameter name	Type	Short description
<code>enumerateID</code>	string	The device returns this index if the requested data is too large for one response. Pass this parameter in a repeat of the call to return the next batch of data. more...
<code>currentRevision</code>	integer	A number that indicates the current revision of this enumeration. You can use this as a <code>lastRevision</code> input to a future <code>enumerate</code> call to retrieve only the changes between the two enumerations.
<code>autoAttendants</code>	array	A collection of <code>autoAttendant</code> structures.
<code>autoAttendantUniqueID</code>	string	Unique identifier for the auto attendant.
<code>autoAttendantConfiguredName</code>	string	The name of the auto attendant.
<code>startTime</code>	dateTime. iso8601	Start time of the item, e.g. 20110106T14:00:00.

autoAttendant.status

This call returns a struct, as described in [autoAttendant.enumerate \[p.33\]](#), for the selected auto attendant.

A fault code of “no such conference” is returned if there is no auto attendant with the given identifier.

Input parameters

Parameter name	Type	Short description
<code>autoAttendantUniqueID</code>	string	Unique identifier for the auto attendant.

Returned data

Parameter name	Type	Short description
<code>autoAttendantUniqueID</code>	string	Unique identifier for the auto attendant.
<code>autoAttendantConfiguredName</code>	string	The name of the auto attendant.
<code>startTime</code>	dateTime. iso8601	Start time of the item, e.g. 20110106T14:00:00.

cdrlog.delete

Permanently deletes stored CDR log files. The files may contain up to 400 entries each.

The call deletes all whole log files whose highest log indexes are lower than the supplied **deleteIndex**.

Input parameters

Parameter name	Type	Short description
deleteIndex (<i>CDR log</i>)	integer	An event identifier that selects which whole CDR files will be deleted. Any whole files whose highest index is below the supplied value will be deleted from CDR log storage. If you supply the value returned in cdrlog.query.deleteableIndex , you will delete all the files stored at the time of that query.

cdrlog.enumerate

This call allows the calling application to download CDR log data without having to return the entire CDR log. The call returns a subset of the CDR log based on the optional `filter`, `index` and `numEvents` parameters.

Note: The [CDR log reference guide](#) describes the CDR log in its XML form, as downloaded in `cdr_log.xml` via the web interface. When the same events are enumerated with this call, the event type names use camelCase for multiple words rather than using underscores. For example, `conference_finished` in `cdr_log.xml` is the same event type as `conferenceFinished` in this response.

Input parameters

Optional or conditional inputs

Parameter name	Type	Short description
<code>filter</code>	array	An array of strings, which contain the names of event types by which to filter the response. Omit <code>filter</code> to return all event types or include a subset of the following: <code>scheduledConferenceStarted</code> , <code>ad-hocConferenceStarted</code> , <code>conferenceFinished</code> , <code>participantJoined</code> , <code>participantLeft</code>
<code>index</code> (CDR log enumerate call)	integer	Index from which to get events. The device returns the <code>nextIndex</code> so the application can use it to retrieve the next enumeration of CDR data. If <code>index</code> is omitted, negative, or greater (by 2 or more) than the highest index, then the device will enumerate events from the beginning of the CDR log.
<code>numEvents</code> (per enumeration)	integer	Specifies maximum number of events to be returned per enumeration. If omitted (or not between 1 - 20 inclusive), a maximum of 20 events will be returned per enumeration.

Returned data

The response provides reference information such as time and log position, and an array of events that meet the parameters provided in the call. If there are no events to enumerate, the `events` array is returned empty.

Each event in the array contains parameters that are common to all CDR log events and also contains any information that is specific to that type of event. See the [CDR log reference guide](#) for details of the MCU event types.

Parameter name	Type	Short description
<code>startIndex</code>	integer	Either the index provided, or if that is lower than the index of the first record the device has, it will be the first record it does know about. In this case, comparing the <code>startIndex</code> with the index provided gives the number of dropped records.
<code>nextIndex</code>	integer	Revision number of the data being provided, reusable in a subsequent call to the API.

eventsRemaining	boolean	Whether there is data remaining after this. Provided to avoid putting all data in a single call.
currentTime	dateTime. iso8601	The system's current time (UTC).
events(CDR)	array	List of the new events; these are structures with some common fields (time, type, index) and other fields specific to the event type.
time (CDR log)	dateTime. iso8601	The date and time when the event was logged, for example 20110119T13:52:42.
type (event)	string	The name of the event type.
index (CDR log)	integer	The index of the CDR log message.

cdrlog.query

This call queries for statistics about the CDR log.

This call takes no parameters.

Returned data

Parameter name	Type	Short description
firstIndex	integer	The index of the oldest stored event.
deletableIndex	integer	The log index of the most recent event that was archived into a log file. The delete command works on whole files, so you can delete up to the last event that went into a file.
numEvents (<i>CDR log</i>)	integer	The difference between the index numbers of the most recent record and the oldest record, irrespective of whether or not the intervening records have been permanently stored.
percentageCapacity	integer	The percentage of the total available capacity being used by the log.

conference.create

This call creates a new conference on the MCU. Conferences created via the API will appear in the list of conferences accessible via the web interface, and vice versa.

This call returns an error if the total number of ports exceeds the maximum conference size (currently 80). The maximum conference size check is also performed for reserved ports.

The MCU allows a maximum number of conferences, which varies by model as follows:

- MCU 4200 Series, MCU 4500 Series, and MCU MSE 8420: 200 conferences maximum
- MCU 5300 Series and MCU MSE 8510: 500 conferences maximum

Use [conference.destroy \[p.44\]](#) to remove unwanted conferences and thus avoid reaching this limit.

Input parameters

Required inputs

Provide a unique name when creating a conference.

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.

Optional or conditional inputs

Parameter name	Type	Short description
<code>private</code>	boolean	Defines whether the conference is public or private. <code>true</code> if the conference is private. Corresponds to the Visibility setting on the web UI, which can have the value <i>Public</i> or <i>Private</i> .
<code>joinAudioMuted</code>	boolean	Audio mute on join.
<code>joinVideoMuted</code>	boolean	Video mute on join.
<code>joinAGC</code>	boolean	Whether AGC should be used by default for participants joining this conference
<code>enforceMaximumAudioPorts</code>	boolean	Defines whether the conference enforces the <code>maximumAudioPorts</code> limit. Assumed to be <code>true</code> if absent.
<code>enforceMaximumVideoPorts</code>	boolean	Defines whether the conference enforces the <code>maximumVideoPorts</code> limit. Assumed to be <code>true</code> if absent.
<code>templateName</code>	string	The name of the template. When passed in a call, this parameter identifies the template that is used for the purpose of the call.

Pass either `templateName` or `templateNumber` if you want to create a conference based on a template. You can omit both parameters to create the conference using the default template.

<code>templateNumber</code>	integer	An index that uniquely identifies the template. Template numbers are not preserved when the MCU reboots. more...
-----------------------------	---------	--

numericId	string	The numeric ID of the conference. Used for registration with H.323 gatekeeper / SIP registrar, and to dial in to the conference.
guestNumericId	string	If it is configured, this value is used by guests (instead of numericId) to access the conference.
registerWithGatekeeper	boolean	Defines whether or not this conference registers its numericId with the H.323 gatekeeper.
registerWithSIPRegistrar	boolean	Defines whether or not this conference registers its numericId with the SIP registrar.
startTime	dateTime. iso8601	Start time of the item, e.g. 20110106T14:00:00.

If you don't specify a **startTime** parameter, the conference will start immediately.

durationSeconds	integer	The period of time, in seconds, for which this item is active.
------------------------	---------	--

If you omit this parameter in **conference.create**, or set it to 0, the conference will be permanent. If you supply the duration, the conference will be active for one or more instances of the supplied number of seconds (see the **startTime** and **repetition** parameters).

pin	string	The PIN for this conference. A string of numeric digits that must be entered to access the conference.
------------	--------	--

Supply a PIN if you want to restrict the conference to participants who know the PIN.

guestPin	string	Security PIN that a guest can use to gain access to this conference.
description	string	Additional information about the conference.
startLocked	boolean	Defines whether or not the conference should be locked when it starts. Set true if you want it to start in the locked state.
conferenceMeEnabled	boolean	Whether or not ConferenceMe is enabled for this conference.
automaticLectureMode	string	Defines automatic lecture mode. One of type1 , type2 , or disabled . more...
automaticLectureModeEnabled	boolean	Defines whether automatic lecture mode is enabled for this conference. Deprecated by automaticLectureMode . more...
automaticLectureModeTimeout	integer	If automaticLectureMode is type1 , this integer defines the period of time for which a speaker must be talking before lecture mode begins. more...
multicastStreamingEnabled	boolean	Defines whether or not the conference can be multicast.
unicastStreamingEnabled	boolean	Defines whether or not this conference can be unicast to streaming viewers.
contentMode	string	Defines the content mode of the conference. Either disabled , passthrough , transcoded or hybrid . more...
h239Enabled	boolean	Deprecated by contentMode . If you set h239Enabled to true , contentMode will be set to transcoded . If you set h239Enabled to false , contentMode will be set to disabled .

<code>lastChairmanLeavesDisconnect</code>	boolean	Defines whether or not this conference disconnects guests when the last chairperson leaves. Corresponds to the When only guests remain conference setting in the web UI.
<code>cleanupTimeout</code>	integer	Allows the MCU to automatically delete a conference which has ended or been empty for this number of seconds. more...
<code>preconfiguredParticipantsDefer</code>	boolean	<code>true</code> if the MCU defers inviting preconfigured participants until at least one other participant is present. more...
<code>contentTxCodec</code>	string	The codec used to transmit content. If content is being transcoded, it is the output format of the transcoder; either <code>h263+</code> , <code>h264</code> , or <code>automatic</code> (default). This setting does not apply in <code>passthrough</code> mode. more...
<code>contentTxMinimumBitRate</code>	string	The minimum bit rate to use for transmitting content, in bps. One of: 0, 64000, 128000, 192000, 256000, 320000, 384000, 512000, 768000, 1000000, 1250000, or 1500000.
<code>maximumAudioPorts</code>	integer	The maximum number of audio-only ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
<code>maximumVideoPorts</code>	integer	The maximum number of video ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
<code>reservedAudioPorts</code>	integer	The number of audio only ports to reserve for a conference if in port reservation mode. more...
<code>reservedVideoPorts</code>	integer	The number of video ports to reserve for a conference if in port reservation mode.
<code>repetition</code>	string	Defines the repetition frequency of a scheduled conference. One of <code>none</code> , <code>daily</code> , <code>weekly</code> , <code>everyTwoWeeks</code> , or <code>monthly</code> . more...
<code>weekDay</code>	string	Must be present if <code>repetition</code> is <code>monthly</code> . One of <code>monday</code> , <code>tuesday</code> , <code>wednesday</code> , <code>thursday</code> , <code>friday</code> , <code>saturday</code> or <code>sunday</code> . Note that if <code>repetition</code> is not <code>weekly</code> or <code>everyTwoWeeks</code> , the <code>weekDays</code> parameter should be used.
<code>whichWeek</code>	string	Required if <code>repetition</code> is <code>monthly</code> . Defines which week the repeating conference will fall in; one of <code>first</code> , <code>second</code> , <code>third</code> , <code>fourth</code> , or <code>last</code> .
<code>weekDays</code>	string	Required if <code>repetition</code> is <code>weekly</code> or <code>everyTwoWeeks</code> . The parameter accepts a comma separated string of weekday names, e.g. <code>monday, wednesday, friday</code> .
<code>terminationType</code>	string	Defines how a repeating conference eventually terminates. One of <code>noTermination</code> , <code>afterNRepeats</code> or <code>endOnGivenDate</code> . more...
<code>terminationDate</code>	dateTime. iso8601	Required if <code>terminationType</code> is <code>endOnGivenDate</code> . This is the date when conference repetition will cease.
<code>numberOfRepeats</code>	integer	Defines the number of times the conference repeats. Required if <code>terminationType</code> is set to <code>afterNRepeats</code> .
<code>customLayoutEnabled</code>	boolean	<code>true</code> if the custom layout is enabled, <code>false</code> otherwise.

<code>layoutControlEx</code>	string	Defines how the view layout can be controlled. One of <code>disabled</code> , <code>feccOnly</code> , <code>dtmfOnly</code> , <code>feccWithDtmfFallback</code> , or <code>bothFeccAndDtmf</code> . more...
<code>cameraControl</code>	string	Defines how the endpoint camera(s) within your API call's context can be controlled. If present, it may be <code>disabled</code> , <code>feccOnly</code> , <code>dtmfOnly</code> , <code>feccWithDtmfFallback</code> , <code>bothFeccAndDtmf</code> , or <code>default</code> . more...
<code>newParticipantsCustomLayout</code>	boolean	<code>true</code> if new participants use the custom layout, <code>false</code> otherwise. Only valid if <code>customLayoutEnabled</code> is <code>true</code> .
<code>customLayout</code>	integer	The index of the video layout seen by the participant(s), depending on the parameter's context. See Conference layouts [p.188] for a list of available layouts and corresponding index values.
<code>chairControl</code>	string	The chair control setting for this conference. One of <code>none</code> , <code>floorControlOnly</code> , or <code>chairAndFloorControl</code> . more...
<code>suppressDtmfEx</code>	string	Controls the muting of in-band DTMF tones. One of <code>fecc</code> , <code>always</code> , or <code>never</code> . more...
<code>inCallMenuControlChair</code>	string	Defines the level of control a chairperson has over the in call menu. One of <code>off</code> , <code>local</code> , <code>conference</code> , or <code>advanced</code> . more...
<code>inCallMenuControlGuest</code>	string	Defines the level of control a guest has over the in call menu. Either <code>off</code> or <code>local</code> . more...
<code>automaticLectureModeEnabled</code>	boolean	Defines whether automatic lecture mode is enabled for this conference. Deprecated by <code>automaticLectureMode</code> . more...
<code>automaticLectureModeTimeout</code>	integer	If <code>automaticLectureMode</code> is <code>type1</code> , this integer defines the period of time for which a speaker must be talking before lecture mode begins. more...
<code>encryptionRequired</code>	boolean	The encryption setting for this conference, if the encryption feature key is enabled. If <code>true</code> , encryption is required for this conference. Otherwise, encryption is optional.
<code>contentContribution</code>	boolean	Defines whether or not endpoints are permitted to contribute the content channel to this conference. <code>true</code> if content contribution is enabled.
<code>contentTransmitResolutions</code>	string	The resolution for the content channel that will be transmitted to endpoints in this conference. One of <code>4to3Only</code> , <code>16to9Only</code> , or <code>allowAll</code> . more...

Deprecated parameters

Parameter name	Type	Short description
<code>dtmfMuteControl</code>	boolean	Deprecated by <code>inCallMenuControlChair</code> and <code>inCallMenuControlGuest</code> . Defines whether or not a participant can mute audio by pressing *6 on the remote control.
<code>conferenceID</code>	string	Deprecated by <code>numericId</code> .

endTime	dateTime. iso8601	If you do not specify an end time, then the conference will be permanent (until it is explicitly deleted). Your application code should use durationSeconds instead.
layoutControlEnabled	boolean	Deprecated by layoutControlEx . Defines whether the endpoint's participant will have control over the layout if layoutControlDefault is false . more...

conference.destroy

This call destroys a conference on the MCU. The conference whose name you provide is removed from the list of conferences (compare with [conference.end \[p.45\]](#)).

A conference can be destroyed at any time; that is, before the conference has begun, during the conference or after the conference has ended. Destroyed conferences are removed entirely from the system; this includes all future repetitions of the conference.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.

conference.end

This call ends a conference on the MCU. A conference remains in the list of conferences even after the conference has ended — until [conference.destroy \[p.44\]](#) is called.

You can use this call to end an instance of a conference without deleting all future repetitions.

Input parameters

Required inputs

Parameter name	Type	Short description
conferenceName	string	The name of the conference.

conference.enumerate

Returns some or all conferences scheduled, running or completed on the MCU.

Input parameters

Optional or conditional inputs

Parameter name	Type	Short description
<code>enumerateID</code>	string	The device returns this index if the requested data is too large for one response. Pass this parameter in a repeat of the call to return the next batch of data. more...
<code>lastRevision</code>	integer	This number identifies an earlier set of enumeration data to compare against your current call. If you supply this parameter using the <code>currentRevision</code> value returned by a previous enumeration, the current <code>enumerate</code> call will return only the differences since that previous call. If you don't supply this parameter, the device assumes that you want a full enumeration.
<code>moreThanFour</code>	boolean	Enables the call to return more than four conferences (up to 24).
<code>enumerateFilter</code>	string	A filter expression. The enumeration results depend on the supplied expression.

`enumerateFilter` filters on:

Parameter name	Type	Short description
<code>active</code>	boolean	<code>true</code> to request only active conferences.
<code>completed</code>	boolean	True if the conference has finished.
<code>scheduled</code>	boolean	<code>true</code> if the conference is a scheduled conference (regardless of whether or not it is completed).

Returned data

Conditionally returned

If there are entries to return, the method returns them in an array. If there are more entries than can be returned in one response, you'll get the next `enumerateID` up from the one you provided.

Parameter name	Type	Short description
<code>enumerateID</code>	string	The device returns this index if the requested data is too large for one response. Pass this parameter in a repeat of the call to return the next batch of data. more...
<code>currentRevision</code>	integer	A number that indicates the current revision of this enumeration. You can use this as a <code>lastRevision</code> input to a future <code>enumerate</code> call to retrieve only the changes between the two enumerations.

<code>joinAudioMuted</code>	boolean	Audio mute on join.
<code>joinVideoMuted</code>	boolean	Video mute on join.
<code>joinAGC</code>	boolean	Whether AGC should be used by default for participants joining this conference
<code>layoutControlEnabled</code>	boolean	Deprecated by <code>layoutControlEx</code> . Defines whether the endpoint's participant will have control over the layout if <code>layoutControlDefault</code> is <code>false</code> . more...
<code>conferences</code>	array	An array of structs, each of which contains all the returned information about a single conference.
<code>conferenceName</code>	string	The name of the conference.
<code>conferenceType</code>	string	Indicates whether a conference is or was <code>scheduled</code> , or <code>ad_hoc</code> (which means it was started without being scheduled).
<code>uniqueId</code>	integer	An ID that is unique among all scheduled and ad hoc conferences. Each instance of a repeating conference has the same <code>uniqueId</code> .
<code>conferenceActive</code>	boolean	Indicates whether conference is currently active. <code>true</code> if the conference is currently active. <code>false</code> if the conference is currently inactive. Permanent conferences are always active; completed conferences, or those that have not yet started, are inactive.
<code>description</code>	string	Additional information about the conference.
<code>pin</code>	string	The PIN for this conference. A string of numeric digits that must be entered to access the conference.
<code>guestPin</code>	string	Security PIN that a guest can use to gain access to this conference.
<code>numericId</code>	string	The numeric ID of the conference. Used for registration with H.323 gatekeeper / SIP registrar, and to dial in to the conference.
<code>guestNumericId</code>	string	If it is configured, this value is used by guests (instead of <code>numericId</code>) to access the conference.
<code>registerWithGatekeeper</code>	boolean	Defines whether or not this conference registers its <code>numericId</code> with the H.323 gatekeeper.
<code>registerWithSIPRegistrar</code>	boolean	Defines whether or not this conference registers its <code>numericId</code> with the SIP registrar.
<code>multicastStreamingEnabled</code>	boolean	Defines whether or not the conference can be multicast.

<code>unicastStreamingEnabled</code>	boolean	Defines whether or not this conference can be unicast to streaming viewers.
<code>conferenceMeEnabled</code>	boolean	Whether or not ConferenceMe is enabled for this conference.
<code>contentMode</code>	string	Defines the content mode of the conference. Either disabled , passthrough , transcoded or hybrid . more...
<code>h239Enabled</code>	boolean	Deprecated by <code>contentMode</code> . If you set <code>h239Enabled</code> to true , <code>contentMode</code> will be set to transcoded . If you set <code>h239Enabled</code> to false , <code>contentMode</code> will be set to disabled .
<code>contentImportant</code>	boolean	Whether or not content is set to be important.
<code>h239Important</code>	boolean	Whether the H.239 channel is set to be important. Consider this setting deprecated by <code>contentImportant</code> . The setting will still work however, even if the content channel is SIP or VNC or content from a main video participant.
<code>contentTxCodec</code>	string	The codec used to transmit content. If content is being transcoded, it is the output format of the transcoder; either h263+ , h264 , or automatic (default). This setting does not apply in passthrough mode. more...
<code>contentTxMinimumBitRate</code>	string	The minimum bit rate to use for transmitting content, in bps. One of: 0, 64000, 128000, 192000, 256000, 320000, 384000, 512000, 768000, 1000000, 1250000, or 1500000.
<code>lastChairmanLeavesDisconnect</code>	boolean	Defines whether or not this conference disconnects guests when the last chairperson leaves. Corresponds to the When only guests remain conference setting in the web UI.
<code>preconfiguredParticipantsDefer</code>	boolean	true if the MCU defers inviting preconfigured participants until at least one other participant is present. more...
<code>locked</code>	boolean	Defines whether or not the conference is locked.
<code>maximumAudioPorts</code>	integer	The maximum number of audio-only ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
<code>maximumVideoPorts</code>	integer	The maximum number of video ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
<code>reservedAudioPorts</code>	integer	The number of audio only ports to reserve for a conference if in port reservation mode. more...

<code>reservedVideoPorts</code>	integer	The number of video ports to reserve for a conference if in port reservation mode.
<code>customLayoutEnabled</code>	boolean	<code>true</code> if the custom layout is enabled, <code>false</code> otherwise.
<code>customLayout</code>	integer	The index of the video layout seen by the participant(s), depending on the parameter's context. See Conference layouts [p.188] for a list of available layouts and corresponding index values.
<code>private</code>	boolean	Defines whether the conference is public or private. <code>true</code> if the conference is private. Corresponds to the Visibility setting on the web UI, which can have the value <i>Public</i> or <i>Private</i> .
<code>chairControl</code>	string	The chair control setting for this conference. One of <code>none</code> , <code>floorControlOnly</code> , or <code>chairAndFloorControl</code> . more...
<code>suppressDtmfEx</code>	string	Controls the muting of in-band DTMF tones. One of <code>fecc</code> , <code>always</code> , or <code>never</code> . more...
<code>layoutControlEx</code>	string	Defines how the view layout can be controlled. One of <code>disabled</code> , <code>feccOnly</code> , <code>dtmfOnly</code> , <code>feccWithDtmfFallback</code> , or <code>bothFeccAndDtmf</code> . more...
<code>cameraControl</code>	string	Defines how the endpoint camera(s) within your API call's context can be controlled. If present, it may be <code>disabled</code> , <code>feccOnly</code> , <code>dtmfOnly</code> , <code>feccWithDtmfFallback</code> , <code>bothFeccAndDtmf</code> , or <code>default</code> . more...
<code>inCallMenuControlChair</code>	string	Defines the level of control a chairperson has over the in call menu. One of <code>off</code> , <code>local</code> , <code>conference</code> , or <code>advanced</code> . more...
<code>inCallMenuControlGuest</code>	string	Defines the level of control a guest has over the in call menu. Either <code>off</code> or <code>local</code> . more...
<code>automaticLectureMode</code>	string	Defines automatic lecture mode. One of <code>type1</code> , <code>type2</code> , or <code>disabled</code> . more...
<code>automaticLectureModeEnabled</code>	boolean	Defines whether automatic lecture mode is enabled for this conference. Deprecated by <code>automaticLectureMode</code> . more...
<code>automaticLectureModeTimeout</code>	integer	If <code>automaticLectureMode</code> is <code>type1</code> , this integer defines the period of time for which a speaker must be talking before lecture mode begins. more...
<code>encryptionRequired</code>	boolean	The encryption setting for this conference, if the encryption feature key is enabled. If <code>true</code> , encryption is required for this conference. Otherwise, encryption is optional.

contentContribution	boolean	Defines whether or not endpoints are permitted to contribute the content channel to this conference. true if content contribution is enabled.
floorStatus	string	One of inactive , active , or assigned . If it is active or assigned , a floorParticipant struct will be included in the response.
floorParticipant	struct	A structure that identifies which participant has the floor.
participantName	string	The unique name of a participant. more...
participantProtocol	string	h323 , sip , or vnc .
participantType	string	One of: by_address , by_name , or ad_hoc . more...
conferenceName	string	The name of the conference.
If the participant is in a conference, the response includes the conferenceName ; if the participant is in an autoattendant, the response includes the autoAttendantUniqueId instead. The response will not include both parameters.		
autoAttendantUniqueId	string	Unique identifier for the auto attendant.
connectionUniqueId	integer	Corresponds to the uniqueId returned by a conference or autoattendant.
chairParticipant	struct	A structure containing parameters that uniquely identify the participant who is the chairperson.
participantName	string	The unique name of a participant. more...
participantProtocol	string	h323 , sip , or vnc .
participantType	string	One of: by_address , by_name , or ad_hoc . more...
conferenceName	string	The name of the conference.
If the participant is in a conference, the response includes the conferenceName ; if the participant is in an autoattendant, the response includes the autoAttendantUniqueId instead. The response will not include both parameters.		
autoAttendantUniqueId	string	Unique identifier for the auto attendant.
connectionUniqueId	integer	Corresponds to the uniqueId returned by a conference or autoattendant.

Conditionally returned for scheduled conferences only:

Parameter name	Type	Short description
startTime	dateTime. iso8601	Start time of the item, e.g. 20110106T14:00:00.
durationSeconds	integer	The period of time, in seconds, for which this item is active.

repetition	string	Defines the repetition frequency of a scheduled conference. One of none , daily , weekly , everyTwoWeeks , or monthly . more...
weekDay	string	Must be present if repetition is monthly . One of monday , tuesday , wednesday , thursday , friday , saturday or sunday . Note that if repetition is not weekly or everyTwoWeeks , the weekDays parameter should be used.
whichWeek	string	Required if repetition is monthly . Defines which week the repeating conference will fall in; one of first , second , third , fourth , or last .
weekDays	string	Required if repetition is weekly or everyTwoWeeks . The parameter accepts a comma separated string of weekday names, e.g. monday , wednesday , friday .
terminationType	string	Defines how a repeating conference eventually terminates. One of noTermination , afterNRepeats or endOnGivenDate . more...
terminationDate	dateTime. iso8601	Required if terminationType is endOnGivenDate . This is the date when conference repetition will cease.

Conditionally returned for active conferences only:

Parameter name	Type	Short description
activeStartTime	dateTime. iso8601	If the conference is currently active, this parameter contains the time that the current session started.
activeEndTime	dateTime. iso8601	If the conference is currently active, this field contains the time of the response, to delimits the time span since the start of the current session. This parameter is absent if the conference is permanent.
activeConferenceId	string	An ID that is unique to each period of activity for a permanent conference. The instance of the conference will retain this ID even if, for example, the conference is renamed while it is active. Each scheduled repeat of the conference has a different activeConferenceId .

Deprecated parameters

Parameter name	Type	Short description
dtmfMuteControl	boolean	Deprecated by inCallMenuControlChair and inCallMenuControlGuest . Defines whether or not a participant can mute audio by pressing *6 on the remote control.

conference.floor.modify

This call modifies the status of the conference floor control.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.
<code>floorStatus</code>	string	One of <code>inactive</code> or <code>assign</code> . If you set <code>floorStatus</code> to <code>assign</code> you must provide a <code>floorParticipant</code> struct.

Optional or conditional inputs

Parameter name	Type	Short description
<code>floorParticipant</code>	struct	A structure that identifies which participant has the floor.
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> or <code>ad_hoc</code> . more...

Returned data

No data. Success or fault message only.

conference.floor.query

This call queries the status of the conference floor control.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.

Returned data

Always returned

Parameter name	Type	Short description
<code>enabled</code>	boolean	<code>true</code> if this feature or item is enabled.
<code>floorStatus</code>	string	One of <code>inactive</code> , <code>active</code> , or <code>assigned</code> . If it is <code>active</code> or <code>assigned</code> , a <code>floorParticipant</code> struct will be included in the response.

Conditionally returned

If `floorStatus` is not `inactive`, then the response includes a `struct` to identify which participant 'has the floor'.

Parameter name	Type	Short description
<code>floorParticipant</code>	struct	A structure that identifies which participant has the floor.
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...
<code>conferenceName</code>	string	The name of the conference.

If the participant is in a conference, the response includes the `conferenceName`; if the participant is in an autoattendant, the response includes the `autoAttendantUniqueId` instead. The response will not include both parameters.

<code>autoAttendantUniqueId</code>	string	Unique identifier for the auto attendant.
<code>connectionUniqueId</code>	integer	Corresponds to the <code>uniqueId</code> returned by a conference or autoattendant.

conference.metadata.modify

Conferences may hold up to 4095 characters of unicode metadata, which are set or cleared with this call.

There is a limit to the number of conferences which can hold metadata. This limit is defined by half the maximum port capacity of the MCU, which varies by media port mode.

The call will return an error if this limit is reached.

For example, an MCU MSE 8510 in SD mode has a maximum port capacity of 80, irrespective of the number of licenses. In this case, up to 40 conferences may hold metadata.

If the call is successful, the device overwrites existing metadata (if any) with the value of `metadata`. Send an empty string to clear the metadata. If you omit the metadata parameter, the device does not modify the existing metadata, but still returns a success message.

Note: The metadata stored against a conference may have been set by an integrated system such as the Cisco TelePresence Conductor. Do not modify metadata that is required by other parts of your wider solution.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.

Optional or conditional inputs

Parameter name	Type	Short description
<code>metadata</code>	string (4095)	A string of up to 4095 unicode characters stored on the device and associated with the named conference.

Returned data

Parameter name	Type	Short description
<code>status (success)</code>	string	<code>Operation successful</code>

conference.metadata.status

Returns the metadata stored against the conference referenced by the supplied `conferenceName` parameter.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.

Returned data

Parameter name	Type	Short description
<code>metadata</code>	string (4095)	A string of up to 4095 unicode characters stored on the device and associated with the named conference.

conference.modify

This call modifies the settings of an existing conference. Conferences created through the management API will appear in the list of conferences accessible via the web interface. Therefore, the API can be used to modify conferences scheduled via the web interface, and vice versa.

This call returns an error if both `maximumVideoPorts` and `maximumAudioPorts` are set to 0 or if the total number of ports exceeds the maximum conference size (currently 80). The maximum conference size check is also performed for reserved ports.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.

Optional or conditional inputs

Parameter name	Type	Short description
<code>newConferenceName</code>	string	The new conference name. more...
<code>numericId</code>	string	The numeric ID of the conference. Used for registration with H.323 gatekeeper / SIP registrar, and to dial in to the conference.
<code>guestNumericId</code>	string	If it is configured, this value is used by guests (instead of <code>numericId</code>) to access the conference.
<code>pin</code>	string	The PIN for this conference. A string of numeric digits that must be entered to access the conference.
<code>guestPin</code>	string	Security PIN that a guest can use to gain access to this conference.
<code>registerWithGatekeeper</code>	boolean	Defines whether or not this conference registers its <code>numericId</code> with the H.323 gatekeeper.
<code>registerWithSIPRegistrar</code>	boolean	Defines whether or not this conference registers its <code>numericId</code> with the SIP registrar.
<code>startTime</code>	dateTime. iso8601	Start time of the item, e.g. 20110106T14:00:00.
<code>durationSeconds</code>	integer	The period of time, in seconds, for which this item is active.
<code>description</code>	string	Additional information about the conference.
<code>joinAGC</code>	boolean	Whether AGC should be used by default for participants joining this conference
<code>multicastStreamingEnabled</code>	boolean	Defines whether or not the conference can be multicast.
<code>unicastStreamingEnabled</code>	boolean	Defines whether or not this conference can be unicast to streaming viewers.

<code>contentMode</code>	string	Defines the content mode of the conference. Either disabled , passthrough , transcoded or hybrid . more...
<code>h239Enabled</code>	boolean	Deprecated by <code>contentMode</code> . If you set <code>h239Enabled</code> to true , <code>contentMode</code> will be set to transcoded . If you set <code>h239Enabled</code> to false , <code>contentMode</code> will be set to disabled .
<code>contentTxCodec</code>	string	The codec used to transmit content. If content is being transcoded, it is the output format of the transcoder; either h263+ , h264 , or automatic (default). This setting does not apply in passthrough mode. more...
<code>contentTxMinimumBitRate</code>	string	The minimum bit rate to use for transmitting content, in bps. One of: 0, 64000, 128000, 192000, 256000, 320000, 384000, 512000, 768000, 1000000, 1250000, or 1500000.
<code>conferenceMeEnabled</code>	boolean	Whether or not ConferenceMe is enabled for this conference.
<code>preconfiguredParticipantsDefer</code>	boolean	true if the MCU defers inviting preconfigured participants until at least one other participant is present. more...
<code>lastChairmanLeavesDisconnect</code>	boolean	Defines whether or not this conference disconnects guests when the last chairperson leaves. Corresponds to the When only guests remain conference setting in the web UI.
<code>private</code>	boolean	Defines whether the conference is public or private. true if the conference is private. Corresponds to the Visibility setting on the web UI, which can have the value <i>Public</i> or <i>Private</i> .
<code>reservedAudioPorts</code>	integer	The number of audio only ports to reserve for a conference if in port reservation mode. more...
<code>reservedVideoPorts</code>	integer	The number of video ports to reserve for a conference if in port reservation mode.
<code>maximumAudioPorts</code>	integer	The maximum number of audio-only ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
<code>maximumVideoPorts</code>	integer	The maximum number of video ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
<code>repetition</code>	string	Defines the repetition frequency of a scheduled conference. One of none , daily , weekly , everyTwoWeeks , or monthly . more...
<code>weekDay</code>	string	Must be present if <code>repetition</code> is monthly . One of monday , tuesday , wednesday , thursday , friday , saturday or sunday . Note that if <code>repetition</code> is not weekly or everyTwoWeeks , the <code>weekDays</code> parameter should be used.
<code>whichWeek</code>	string	Required if <code>repetition</code> is monthly . Defines which week the repeating conference will fall in; one of first , second , third , fourth , or last .
<code>weekDays</code>	string	Required if <code>repetition</code> is weekly or everyTwoWeeks . The parameter accepts a comma separated string of weekday names, e.g. monday , wednesday , friday .

terminationType	string	Defines how a repeating conference eventually terminates. One of noTermination , afterNRepeats or endOnGivenDate . more...
terminationDate	dateTime. iso8601	Required if terminationType is endOnGivenDate . This is the date when conference repetition will cease.
numberOfRepeats	integer	Defines the number of times the conference repeats. Required if terminationType is set to afterNRepeats .
contentImportant	boolean	Whether or not content is set to be important.
h239Important	boolean	Whether the H.239 channel is set to be important. Consider this setting deprecated by contentImportant . The setting will still work however, even if the content channel is SIP or VNC or content from a main video participant.
locked	boolean	Defines whether or not the conference is locked.
startLocked	boolean	Defines whether or not the conference should be locked when it starts. Set true if you want it to start in the locked state.
layoutControlEx	string	Defines how the view layout can be controlled. One of disabled , feccOnly , dtmfOnly , feccWithDtmfFallback , or bothFeccAndDtmf . more...
cameraControl	string	Defines how the endpoint camera(s) within your API call's context can be controlled. If present, it may be disabled , feccOnly , dtmfOnly , feccWithDtmfFallback , bothFeccAndDtmf , or default . more...
newParticipantsCustomLayout	boolean	true if new participants use the custom layout, false otherwise. Only valid if customLayoutEnabled is true .
customLayout	integer	The index of the video layout seen by the participant(s), depending on the parameter's context. See Conference layouts [p.188] for a list of available layouts and corresponding index values.
chairControl	string	The chair control setting for this conference. One of none , floorControlOnly , or chairAndFloorControl . more...
enforceMaximumAudioPorts	boolean	Defines whether the conference enforces the maximumAudioPorts limit. Assumed to be true if absent.
enforceMaximumVideoPorts	boolean	Defines whether the conference enforces the maximumVideoPorts limit. Assumed to be true if absent.
suppressDtmfEx	string	Controls the muting of in-band DTMF tones. One of fecc , always , or never . more...
inCallMenuControlChair	string	Defines the level of control a chairperson has over the in call menu. One of off , local , conference , or advanced . more...
inCallMenuControlGuest	string	Defines the level of control a guest has over the in call menu. Either off or local . more...
automaticLectureMode	string	Defines automatic lecture mode. One of type1 , type2 , or disabled . more...
automaticLectureModeEnabled	boolean	Defines whether automatic lecture mode is enabled for this conference. Deprecated by automaticLectureMode . more...

automaticLectureModeTimeout	integer	If automaticLectureMode is type1 , this integer defines the period of time for which a speaker must be talking before lecture mode begins. more...
encryptionRequired	boolean	The encryption setting for this conference, if the encryption feature key is enabled. If true , encryption is required for this conference. Otherwise, encryption is optional.
contentContribution	boolean	Defines whether or not endpoints are permitted to contribute the content channel to this conference. true if content contribution is enabled.

Deprecated parameters

Parameter name	Type	Short description
dtmfMuteControl	boolean	Deprecated by inCallMenuControlChair and inCallMenuControlGuest . Defines whether or not a participant can mute audio by pressing *6 on the remote control.
oldConferenceName	string	Deprecated conference renaming scheme - new code should use conferenceName and newConferenceName as above.
conferenceName	string	The name of the conference.
conferenceID	string	Deprecated by numericId .
endTime	dateTime. iso8601	If you do not specify an end time, then the conference will be permanent (until it is explicitly deleted). Your application code should use durationSeconds instead.
layoutControlEnabled	boolean	Deprecated by layoutControlEx . Defines whether the endpoint's participant will have control over the layout if layoutControlDefault is false . more...

conference.paneplacement.modify

Modifies the pane placement for a particular conference.

The panes array contains structures which define the specific panes and their contents. If you do not supply a particular pane index in the array, then that pane remains unchanged in the layout.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.

Optional or conditional inputs

Parameter name	Type	Short description
<code>enabled</code>	boolean	<code>true</code> if this feature or item is enabled.

Set `true` to enable pane placement.

Parameter name	Type	Short description
<code>panes</code>	array	An array of <code>structs</code> , each of which defines a particular pane within the layout.
<code>index (pane)</code>	integer	A number that identifies the pane with respect to other panes. A value between 0 and 19, where lower numbers are generally more prominent in the layout.
<code>type (pane)</code>	string	Defines how the MCU fills the pane. One of <code>default</code> , <code>blank</code> , <code>loudest</code> , <code>rolling</code> , <code>h239</code> , or <code>participant</code> . more...

Conditionally required

The following parameters are required to identify the participant if you set `type` to `participant`.

Parameter name	Type	Short description
<code>participantType</code>	string	One of: <code>by_address</code> or <code>ad_hoc</code> . more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantName</code>	string	The unique name of a participant. more...

Returned data

Always returned

Because not all panes are guaranteed to be changed, this call returns the following structure:

Parameter name	Type	Short description
panesModified	integer	The number of panes successfully modified. This will be the number of elements in the panes array on complete success, and zero if there is no panes array.

conference.paneplacement.query

Queries the current pane placement configuration. Returns whether pane placement is enabled and, if so, an array of panes detailing the current pane placement.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.

Returned data

The response contains the `enabled` parameter and the `panes` array. If `enabled` is `true`, the `panes` array contains a struct for each placed pane. The array is returned empty if pane placement is disabled.

Always returned

Parameter name	Type	Short description
<code>enabled</code>	boolean	<code>true</code> if this feature or item is enabled.
<code>panes</code>	array	An array of <code>structs</code> , each of which defines a particular pane within the layout.

Conditionally returned

The `panes` array contains data if pane placement is enabled. The number of panes in the array corresponds with the number of panes in the current conference custom layout:

Parameter name	Type	Short description
<code>panes</code>	array	An array of <code>structs</code> , each of which defines a particular pane within the layout.
<code>index (pane)</code>	integer	A number that identifies the pane with respect to other panes. A value between 0 and 19, where lower numbers are generally more prominent in the layout.
<code>type (pane)</code>	string	Defines how the MCU fills the pane. One of <code>default</code> , <code>blank</code> , <code>loudest</code> , <code>rolling</code> , <code>h239</code> , or <code>participant</code> . more...

The following are also returned if the pane `type` is `participant`:

<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...

conference.resetCleanupTimeout

Resets the cleanup timeout on the named conference.

Input parameters

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.

conference.status

Returns information about a named conference on the MCU.

This call returns an error if both `maximumVideoPorts` and `maximumAudioPorts` are set to 0 or if the total number of ports exceeds the maximum conference size (currently 80). The maximum conference size check is also performed for reserved ports.

The MCU returns a “no such conference” fault if it can not find a conference with the supplied `conferenceName`.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.

Returned data

A struct containing the status parameters of the named conference.

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.
<code>conferenceType</code>	string	Indicates whether a conference is or was <code>scheduled</code> , or <code>ad_hoc</code> (which means it was started without being scheduled).
<code>uniqueId</code>	integer	An ID that is unique among all scheduled and ad hoc conferences. Each instance of a repeating conference has the same <code>uniqueId</code> .
<code>conferenceActive</code>	boolean	Indicates whether conference is currently active. <code>true</code> if the conference is currently active. <code>false</code> if the conference is currently inactive. Permanent conferences are always active; completed conferences, or those that have not yet started, are inactive.
<code>description</code>	string	Additional information about the conference.
<code>pin</code>	string	The PIN for this conference. A string of numeric digits that must be entered to access the conference.
<code>guestPin</code>	string	Security PIN that a guest can use to gain access to this conference.
<code>numericId</code>	string	The numeric ID of the conference. Used for registration with H.323 gatekeeper / SIP registrar, and to dial in to the conference.
<code>guestNumericId</code>	string	If it is configured, this value is used by guests (instead of <code>numericId</code>) to access the conference.

<code>registerWithGatekeeper</code>	boolean	Defines whether or not this conference registers its <code>numericId</code> with the H.323 gatekeeper.
<code>registerWithSIPRegistrar</code>	boolean	Defines whether or not this conference registers its <code>numericId</code> with the SIP registrar.
<code>multicastStreamingEnabled</code>	boolean	Defines whether or not the conference can be multicast.
<code>unicastStreamingEnabled</code>	boolean	Defines whether or not this conference can be unicast to streaming viewers.
<code>conferenceMeEnabled</code>	boolean	Whether or not ConferenceMe is enabled for this conference.
<code>contentMode</code>	string	Defines the content mode of the conference. Either <code>disabled</code> , <code>passthrough</code> , <code>transcoded</code> or <code>hybrid</code> . more...
<code>h239Enabled</code>	boolean	Deprecated by <code>contentMode</code> . If you set <code>h239Enabled</code> to <code>true</code> , <code>contentMode</code> will be set to <code>transcoded</code> . If you set <code>h239Enabled</code> to <code>false</code> , <code>contentMode</code> will be set to <code>disabled</code> .
<code>contentImportant</code>	boolean	Whether or not content is set to be important.
<code>h239Important</code>	boolean	Whether the H.239 channel is set to be important. Consider this setting deprecated by <code>contentImportant</code> . The setting will still work however, even if the content channel is SIP or VNC or content from a main video participant.
<code>contentTxCodec</code>	string	The codec used to transmit content. If content is being transcoded, it is the output format of the transcoder; either <code>h263+</code> , <code>h264</code> , or <code>automatic</code> (default). This setting does not apply in <code>passthrough</code> mode. more...
<code>contentTxMinimumBitRate</code>	string	The minimum bit rate to use for transmitting content, in bps. One of: 0, 64000, 128000, 192000, 256000, 320000, 384000, 512000, 768000, 1000000, 1250000, or 1500000.
<code>lastChairmanLeavesDisconnect</code>	boolean	Defines whether or not this conference disconnects guests when the last chairperson leaves. Corresponds to the When only guests remain conference setting in the web UI.
<code>preconfiguredParticipantsDefer</code>	boolean	<code>true</code> if the MCU defers inviting preconfigured participants until at least one other participant is present. more...
<code>locked</code>	boolean	Defines whether or not the conference is locked.
<code>maximumAudioPorts</code>	integer	The maximum number of audio-only ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
<code>maximumVideoPorts</code>	integer	The maximum number of video ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
<code>reservedAudioPorts</code>	integer	The number of audio only ports to reserve for a conference if in port reservation mode. more...
<code>reservedVideoPorts</code>	integer	The number of video ports to reserve for a conference if in port reservation mode.
<code>customLayoutEnabled</code>	boolean	<code>true</code> if the custom layout is enabled, <code>false</code> otherwise.

customLayout	integer	The index of the video layout seen by the participant(s), depending on the parameter's context. See Conference layouts [p.188] for a list of available layouts and corresponding index values.
private	boolean	Defines whether the conference is public or private. true if the conference is private. Corresponds to the Visibility setting on the web UI, which can have the value <i>Public</i> or <i>Private</i> .
joinAGC	boolean	Whether AGC should be used by default for participants joining this conference
chairControl	string	The chair control setting for this conference. One of none , floorControlOnly , or chairAndFloorControl . more...
suppressDtmfEx	string	Controls the muting of in-band DTMF tones. One of fecc , always , or never . more...
layoutControlEx	string	Defines how the view layout can be controlled. One of disabled , feccOnly , dtmfOnly , feccWithDtmfFallback , or bothFeccAndDtmf . more...
cameraControl	string	Defines how the endpoint camera(s) within your API call's context can be controlled. If present, it may be disabled , feccOnly , dtmfOnly , feccWithDtmfFallback , bothFeccAndDtmf , or default . more...
inCallMenuControlChair	string	Defines the level of control a chairperson has over the in call menu. One of off , local , conference , or advanced . more...
inCallMenuControlGuest	string	Defines the level of control a guest has over the in call menu. Either off or local . more...
automaticLectureMode	string	Defines automatic lecture mode. One of type1 , type2 , or disabled . more...
automaticLectureModeEnabled	boolean	Defines whether automatic lecture mode is enabled for this conference. Deprecated by automaticLectureMode . more...
automaticLectureModeTimeout	integer	If automaticLectureMode is type1 , this integer defines the period of time for which a speaker must be talking before lecture mode begins. more...
encryptionRequired	boolean	The encryption setting for this conference, if the encryption feature key is enabled. If true , encryption is required for this conference. Otherwise, encryption is optional.
contentContribution	boolean	Defines whether or not endpoints are permitted to contribute the content channel to this conference. true if content contribution is enabled.
floorStatus	string	One of inactive , active , or assigned . If it is active or assigned , a floorParticipant struct will be included in the response.
floorParticipant	struct	A structure that identifies which participant has the floor.
participantName	string	The unique name of a participant. more...
participantProtocol	string	h323 , sip , or vnc .

participantType	string	One of: by_address , by_name , or ad_hoc . more...
conferenceName	string	The name of the conference.
If the participant is in a conference, the response includes the conferenceName ; if the participant is in an autoattendant, the response includes the autoAttendantUniqueId instead. The response will not include both parameters.		
autoAttendantUniqueId	string	Unique identifier for the auto attendant.
connectionUniqueId	integer	Corresponds to the uniqueId returned by a conference or autoattendant.
chairParticipant	struct	A structure containing parameters that uniquely identify the participant who is the chairperson.
participantName	string	The unique name of a participant. more...
participantProtocol	string	h323 , sip , or vnc .
participantType	string	One of: by_address , by_name , or ad_hoc . more...
conferenceName	string	The name of the conference.
If the participant is in a conference, the response includes the conferenceName ; if the participant is in an autoattendant, the response includes the autoAttendantUniqueId instead. The response will not include both parameters.		
autoAttendantUniqueId	string	Unique identifier for the auto attendant.
connectionUniqueId	integer	Corresponds to the uniqueId returned by a conference or autoattendant.

Conditionally returned for **scheduled** conferences only:

Parameter name	Type	Short description
startTime	dateTime. iso8601	Start time of the item, e.g. 20110106T14:00:00.
durationSeconds	integer	The period of time, in seconds, for which this item is active.
repetition	string	Defines the repetition frequency of a scheduled conference. One of none , daily , weekly , everyTwoWeeks , or monthly . more...
weekDay	string	Must be present if repetition is monthly . One of monday , tuesday , wednesday , thursday , friday , saturday or sunday . Note that if repetition is not weekly or everyTwoWeeks , the weekDays parameter should be used.
whichWeek	string	Required if repetition is monthly . Defines which week the repeating conference will fall in; one of first , second , third , fourth , or last .
weekDays	string	Required if repetition is weekly or everyTwoWeeks . The parameter accepts a comma separated string of weekday names, e.g. monday , wednesday , friday .

terminationType	string	Defines how a repeating conference eventually terminates. One of noTermination , afterNRRepeats or endOnGivenDate . more...
terminationDate	dateTime. iso8601	Required if terminationType is endOnGivenDate . This is the date when conference repetition will cease.

Conditionally returned for active conferences only:

Parameter name	Type	Short description
activeStartTime	dateTime. iso8601	If the conference is currently active, this parameter contains the time that the current session started.
activeEndTime	dateTime. iso8601	If the conference is currently active, this field contains the time of the response, to delimits the time span since the start of the current session. This parameter is absent if the conference is permanent.
activeConferenceId	string	An ID that is unique to each period of activity for a permanent conference. The instance of the conference will retain this ID even if, for example, the conference is renamed while it is active. Each scheduled repeat of the conference has a different activeConferenceId .

Deprecated parameters

Parameter name	Type	Short description
dtmfMuteControl	boolean	Deprecated by inCallMenuControlChair and inCallMenuControlGuest . Defines whether or not a participant can mute audio by pressing *6 on the remote control.

conference.streaming.modify

Modifies the parameters of the layout being streamed from the specified conference.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.

Optional or conditional inputs

Parameter name	Type	Short description
<code>cpLayout</code>	string	This sets the initial conference view layout for the video sent to the participant. Refer to Conference layouts [p.188] for details. Refer to Conference layouts [p.188] for details.
<code>borderWidth</code>	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. more...
<code>focusType</code>	string	Indicates the endpoint's focus. One of <code>participant</code> , <code>voiceActivated</code> , or <code>h239</code> . more...

Conditionally required

The following parameters are required to identify the participant if you set `focusType` to `participant`.

Parameter name	Type	Short description
<code>focusParticipant</code>	struct	The structure contains participant parameters that identify which participant displays in the largest pane if <code>focusType</code> is <code>participant</code> .
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> or <code>ad_hoc</code> . more...

conference.streaming.query

Returns details on the current state of streaming viewers for a conference.

This call will return a fault code of "no such conference" if there is no *active* conference with the given name, regardless of the presence of a configured but inactive conference of that name.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.

Returned data

Always returned

The response includes a structure with the following fields:

Parameter name	Type	Short description
<code>unicastViewers</code>	integer	The count of unicast streaming viewers.
<code>multicastViewers</code>	integer	The count of multicast streaming viewers.
<code>audioRTCPReceiverReports</code>	integer	The number of RTCP receiver reports for the audio streams seen by the MCU.
<code>audioRTCPSenderReports</code>	integer	The number of RTCP sender reports for the audio streams seen by the MCU.
<code>audioRTCPOther</code>	integer	The number of other RTCP packets seen for the audio streams.
<code>audioRTCPPacketsSent</code>	integer	The number of RTCP packets sent by the MCU.
<code>videoRTCPReceiverReports</code>	integer	As for the audio equivalents.
<code>videoRTCPSenderReports</code>	integer	As for the audio equivalents.
<code>videoRTCPOther</code>	integer	As for the audio equivalents.
<code>videoRTCPPacketsSent</code>	integer	As for the audio equivalents.
<code>currentLayout</code>	integer	The actual layout in use for the video stream being sent by the MCU to streaming viewers. Refer to Conference layouts [p.188] for details.
<code>layoutSource</code>	string	Describes the reason for the current layout, and is only present if <code>currentLayout</code> is present. One of <code>familyx</code> , <code>conferenceCustom</code> , or <code>participantCustom</code> . more...
<code>borderWidth</code>	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. more...
<code>focusType</code>	string	Indicates the endpoint's focus. One of <code>participant</code> , <code>voiceActivated</code> , or <code>h239</code> . more...

Conditionally returned

focusParticipant struct

The following parameters identify the participant if the **focusType** is **participant**.

Parameter name	Type	Short description
focusParticipant	struct	The structure contains participant parameters that identify which participant displays in the largest pane if focusType is participant .
participantName	string	The unique name of a participant. more...
participantProtocol	string	h323 , sip , or vnc .
participantType	string	One of: by_address , by_name , or ad_hoc . more...
conferenceName	string	The name of the conference.

If the participant is in a conference, the response includes the **conferenceName**; if the participant is in an autoattendant, the response includes the **autoAttendantUniqueId** instead. The response will not include both parameters.

autoAttendantUniqueId	string	Unique identifier for the auto attendant.
connectionUniqueId	integer	Corresponds to the uniqueId returned by a conference or autoattendant.

stream structs

If there are active audio streams or video streams at the time of the response, then the response will include an array of **stream** structures for each collection of streams.

Parameter name	Type	Short description
audioStreams	array	An array of stream structs (defined below). These are only present if there are any streams of either type currently in use.

The **stream** structures in the **audioStreams** array include the following details:

codec	string	The codec in use, or other for undefined codecs.
count	integer	The number of users of this codec.
videoStreams	array	An array of stream structs. The structs are only present if there are any streams of either type currently in use.

The **stream** structures in the **videoStreams** array include the following details:

codec	string	The codec in use, or other for undefined codecs.
count	integer	The number of users of this codec.
bitRate	integer	The bitrate of this stream in bits/second. This is only present for video streams with a defined codec.
width	integer	The maximum width and height of this stream. Only present for defined video streams
height	integer	The maximum width and height of this stream. Only present for defined video streams

conferenceme.modify

If `setting` is `true`, this call will enable conferenceMe but disable streaming. This call is not supported on slave blades.

Input parameters

Optional or conditional inputs

Parameter name	Type	Short description
<code>setting</code>	boolean	Defines whether or not this feature is intended to be enabled, irrespective of whether it is actually enabled or requires a feature key.
<code>mediaOverTcp</code>	boolean	<code>true</code> allows ConferenceMe to fall back to media over TCP if it cannot do media over UDP.
<code>maxBitRateFromMCU</code>	integer	Maximum bandwidth from the MCU (kbps).
<code>maxBitRateToMCU</code>	integer	Maximum bandwidth to the MCU (kbps).
<code>useWebService</code>	boolean	<code>true</code> if ConferenceMe may use web service to connect clients to a conference. Corresponds to the "Allow ConferenceMe to use web service" checkbox on the web interface.
<code>maxParticipants</code>	integer	The maximum number of ConferenceMe connections allowed.

conferenceme.query

Queries for information about ConferenceMe.

Accepts no parameters. Returns whether ConferenceMe is enabled and, if so, the ConferenceMe parameters.

Returned data

Always returned

Parameter name	Type	Short description
<code>enabled</code>	boolean	<code>true</code> if this feature or item is enabled.
<code>setting</code>	boolean	Defines whether or not this feature is intended to be enabled, irrespective of whether it is actually enabled or requires a feature key.
<code>maxBitRateFromMCU</code>	integer	Maximum bandwidth from the MCU (kbps).
<code>maxBitRateToMCU</code>	integer	Maximum bandwidth to the MCU (kbps).
<code>mediaOverTcp</code>	boolean	<code>true</code> allows ConferenceMe to fall back to media over TCP if it cannot do media over UDP.
<code>useWebService</code>	boolean	<code>true</code> if ConferenceMe may use web service to connect clients to a conference. Corresponds to the "Allow ConferenceMe to use web service" checkbox on the web interface.
<code>maxParticipants</code>	integer	The maximum number of ConferenceMe connections allowed.

device.content.modify

Modifies the device's content settings. Not supported on slave blades.

Input parameters

Optional or conditional inputs

Parameter name	Type	Short description
<code>contentEnabled</code>	string	One of <code>enabled</code> , <code>h239Only</code> or <code>disabled</code> .
<code>contentInMainVideo</code>	boolean	<code>true</code> if the content can display in the main video channel.
<code>furFilteringEnabled</code>	boolean	<code>true</code> if video fast update request filtering is enabled.
<code>webAppletBandwidth</code>	integer	The bandwidth of the content stream sent to streaming viewers.
<code>contentMarkupEnabled</code>	boolean	<code>true</code> if content markup is enabled.
<code>contentHandoverEnabled</code>	boolean	<code>true</code> if automatic content handover is enabled.

device.content.query

Queries the device for its content settings. Not supported on slave blades.

Returned data

Always returned

Parameter name	Type	Short description
<code>contentEnabled</code>	string	One of <code>enabled</code> , <code>h239Only</code> or <code>disabled</code> .
<code>contentInMainVideo</code>	boolean	<code>true</code> if the content can display in the main video channel.
<code>furFilteringEnabled</code>	boolean	<code>true</code> if video fast update request filtering is enabled.
<code>contentStreamingStatus</code>	boolean	<code>true</code> if the web conferencing feature key is present and <code>contentEnabled</code> is either <code>enabled</code> or <code>h239Only</code> .
<code>contentStreamingSetting</code>	boolean	<code>true</code> if <code>contentEnabled</code> is <code>enabled</code> or <code>h239Only</code> .
<code>webAppletBandwidth</code>	integer	The bandwidth of the content stream sent to streaming viewers.
<code>contentMarkupEnabled</code>	boolean	<code>true</code> if content markup is enabled.
<code>contentHandoverEnabled</code>	boolean	<code>true</code> if automatic content handover is enabled.

device.encryption.modify

Modifies the device's encryption settings. Not supported on slave blades.

Input parameters

Optional or conditional inputs

Parameter name	Type	Short description
<code>setting</code>	boolean	Defines whether or not this feature is intended to be enabled, irrespective of whether it is actually enabled or requires a feature key.
<code>sipMediaEncryption</code>	string	Defines whether SIP media is encrypted and, if so, for which transport protocols. One of <code>disabled</code> , <code>allTransports</code> or <code>tlsOnly</code> .

device.encryption.query

Queries the device for its encryption settings. Not supported on slave blades.

Returned data

Always returned

Parameter name	Type	Short description
<code>enabled</code>	boolean	<code>true</code> if this feature or item is enabled.
<code>setting</code>	boolean	Defines whether or not this feature is intended to be enabled, irrespective of whether it is actually enabled or requires a feature key.
<code>sipMediaEncryption</code>	string	Defines whether SIP media is encrypted and, if so, for which transport protocols. One of <code>disabled</code> , <code>allTransports</code> or <code>tlsOnly</code> .

device.health.query

Returns the current status of the device, such as health monitors and CPU load.

Returned data

Parameter name	Type	Short description
<code>cpuLoad</code>	integer	The CPU load as a percentage of the maximum.
<code>mediaLoad</code>	integer	A percentage value representing the proportion of the device's media processing capacity that is currently in use.
<code>audioLoad</code>	integer	A percentage value representing the proportion of the device's audio processing capacity that is currently in use.
<code>videoLoad</code>	integer	A percentage value representing the proportion of the device's video processing capacity that is currently in use.
<code>fanStatus</code>	string	One of <code>ok</code> , <code>outOfSpec</code> , or <code>critical</code> .
<code>fanStatusWorst</code>	string	One of <code>ok</code> , <code>outOfSpec</code> , or <code>critical</code> .
<code>temperatureStatus</code>	string	The current temperature status. One of <code>ok</code> , <code>outOfSpec</code> , or <code>critical</code> . The device will shutdown if the <code>critical</code> status persists.
<code>temperatureStatusWorst</code>	string	The worst temperature status recorded on this device since it booted. One of <code>ok</code> , <code>outOfSpec</code> , or <code>critical</code> . more...
<code>rtcBatteryStatus</code>	string	The current status of the RTC battery (Real Time Clock). One of <code>ok</code> , <code>outOfSpec</code> (the battery is operating outside of the normal range, and may require service), or <code>critical</code> .
<code>rtcBatteryStatusWorst</code>	string	The worst recorded status of the RTC battery. One of <code>ok</code> , <code>outOfSpec</code> (the battery has operated outside of the normal range at some time since the device was booted), or <code>critical</code> .
<code>voltagesStatus</code>	string	<code>ok</code> , <code>outOfSpec</code> (the voltage is currently outside the normal range), or <code>critical</code> .
<code>voltagesStatusWorst</code>	string	<code>ok</code> , <code>outOfSpec</code> (the voltage has been outside the normal range at some time since the device last booted), or <code>critical</code> .
<code>operationalStatus</code>	string	One of <code>active</code> , <code>shuttingDown</code> , or <code>shutdown</code> .

device.network.modify

Modifies the device's network information. You may supply only the parameters that you want to change but, in some cases, you must supply a parameter (depending on the value you set for another parameter).

Include the parameters you want to modify in the appropriate struct; `portA`, `portB`, or `dns`. The `portA` and `portB` structs take the same parameters.

Note: The device returns a success message after successfully parsing your call but before implementing the settings. Also, you will generate a fault if you attempt to disable the active interface.

Input parameters

Required inputs

If you set `ipv4Enabled` to `true`, you must supply `dhcpv4`. If you set `dhcpv4` to `false`, you must supply `ipv4Address` and `ipv4SubnetMask`.

If you set `ipv6Enabled` to `true`, you must supply `ipv6Conf`. If you set `ipv6Conf` to `manual`, you must supply `ipv6Address` and `ipv6PrefixLength`.

If you set `ethernetAutomatic` to `false`, you must supply `speed` and `fullDuplex`.

Optional or conditional inputs

Parameter name	Type	Short description
<code>portA</code>	struct	A structure that contains configuration and status information for Ethernet port A on the device.
<code>portB</code>	struct	A structure that contains configuration and status information for Ethernet port B on the device.
<code>ipv4Enabled</code>	boolean	<code>true</code> if IPv4 interface is enabled.
<code>dhcpv4</code>	boolean	Defines whether or not to use DHCP to obtain an IPv4 address. Deprecates <code>dhcp</code> .
<code>ipv4Address</code>	string (31)	IPv4 address in dotted-quad format.
<code>ipv4SubnetMask</code>	string (31)	The IPv4 subnet mask in dotted quad format. Deprecates <code>subnetMask</code> .
<code>defaultIpv4Gateway</code>	string (31)	The device's IPv4 default gateway in dotted quad format. Deprecates <code>defaultGateway</code> .
<code>ipv6Enabled</code>	boolean	<code>true</code> if IPv6 interface is enabled.
<code>ipv6Conf</code>	string (10)	Indicates how the IPv6 address is assigned; either <code>automatic</code> (by SLAAC/DHCPv6) or <code>manual</code> .
<code>ipv6Address</code>	string (79)	The IPv6 address in CIDR format.
<code>ipv6PrefixLength</code>	integer	The length of the IPv6 address prefix.

defaultIpv6Gateway	string (79)	The address of the IPv6 default gateway in CIDR format.
ethernetAutomatic	boolean	true for the Ethernet interface to configure itself automatically. If you set this to false you must supply the speed and fullDuplex parameters.
speed	integer	Speed of the connection on this Ethernet interface. One of 10, 100 or 1000, in Mbps.
fullDuplex	boolean	true if the port supports a full-duplex connection, false for half-duplex.
dns	struct	The struct members represent the device's DNS parameters.
dnsConfiguration	string (10)	Defines how the device gets its DNS configuration; one of portAIPv4 , portAIPv6 , portBIPv4 , portBIPv6 or manual . If manual , you must supply a name server address. more...
hostName	string (255)	The host name of queried device. Deprecated in API version 2.8.
nameServer	string (79)	The IP address of the name server, in dotted quad format (IPv4) or CIDR format (IPv6).
nameServerSecondary	string (79)	The IP address of the secondary name server, in dotted quad format (IPv4) or CIDR format (IPv6).
domainName	string (255)	The domain name (DNS suffix).

device.network.query

Queries the device for its network information. The call takes no parameters and returns three data structures: `dns`, `portA`, and `portB`. Some of the data listed below will be omitted if the interface is not enabled or configured. The query returns empty strings or dashes for addresses that are not configured.

Returned data

Parameter name	Type	Short description
<code>dns</code>	struct	The struct members represent the device's DNS parameters.
<code>hostName</code>	string (255)	The host name of queried device. Deprecated in API version 2.8.
<code>nameServer</code>	string (79)	The IP address of the name server, in dotted quad format (IPv4) or CIDR format (IPv6).
<code>nameServerSecondary</code>	string (79)	The IP address of the secondary name server, in dotted quad format (IPv4) or CIDR format (IPv6).
<code>domainName</code>	string (255)	The domain name (DNS suffix).
<code>portA</code>	struct	A structure that contains configuration and status information for Ethernet port A on the device.
<code>portB</code>	struct	A structure that contains configuration and status information for Ethernet port B on the device.
<code>enabled</code>	boolean	<code>true</code> if this feature or item is enabled.
<code>ipv4Enabled</code>	boolean	<code>true</code> if IPv4 interface is enabled.
<code>ipv6Enabled</code>	boolean	<code>true</code> if IPv6 interface is enabled.
<code>linkStatus</code>	boolean	<code>true</code> if the ethernet connection to this port is active.
<code>speed</code>	integer	Speed of the connection on this Ethernet interface. One of 10, 100 or 1000, in Mbps.
<code>fullDuplex</code>	boolean	<code>true</code> if the port supports a full-duplex connection, <code>false</code> for half-duplex.
<code>macAddress</code>	string	The MAC address of this interface. A 12 character string of hex digits with no separators.
<code>packetsSent</code>	integer	The number of packets sent from this Ethernet port.
<code>packetsReceived</code>	integer	The number of packets received on this Ethernet port.
<code>multicastPacketsSent</code>	integer	Number of multicast packets sent from this Ethernet interface.

multicastPacketsReceived	integer	Number of multicast packets received on this Ethernet interface.
bytesSent	integer	The number of bytes sent by the device.
bytesReceived	integer	The number of bytes received by the device.
queueDrops	integer	Number of packets dropped from the queue on this network interface.
collisions	integer	Count of the network collisions recorded by the device.
transmitErrors	integer	The count of transmission errors on this Ethernet interface.
receiveErrors	integer	The count of receive errors on this interface.
bytesSent64	string	64 bit versions of the bytesSent statistic, using a string rather than an integer.
bytesReceived64	string	64 bit versions of the bytesReceived statistic, using a string rather than an integer.

Returned only if the interface is enabled and configured:

Parameter name	Type	Short description
dhcpv4	boolean	Defines whether or not to use DHCP to obtain an IPv4 address. Deprecates dhcp .
ipv4Address	string (31)	IPv4 address in dotted-quad format.
ipv4SubnetMask	string (31)	The IPv4 subnet mask in dotted quad format. Deprecates subnetMask .
defaultIpv4Gateway	string (31)	The device's IPv4 default gateway in dotted quad format. Deprecates defaultGateway .
domainName	string (255)	The domain name (DNS suffix).
nameServer	string (79)	The IP address of the name server, in dotted quad format (IPv4) or CIDR format (IPv6).
nameServerSecondary	string (79)	The IP address of the secondary name server, in dotted quad format (IPv4) or CIDR format (IPv6).
ipv6Conf	string (10)	Indicates how the IPv6 address is assigned; either automatic (by SLAAC/DHCPv6) or manual .
ipv6Address	string (79)	The IPv6 address in CIDR format.
ipv6PrefixLength	integer	The length of the IPv6 address prefix.
defaultIpv6Gateway	string (79)	The address of the IPv6 default gateway in CIDR format.
linkLocalIpv6Address	string(63)	The link local IPv6 address in CIDR format.
linkLocalIpv6PrefixLength	integer	Length of the link local IPv6 address prefix.

Deprecated parameters

These are replaced by their explicitly named `ipv4` equivalents.

Parameter name	Type	Short description
<code>dhcp</code>	boolean	Defines whether or not to use DHCP to obtain an IPv4 address.
<code>ipAddress</code>	string	IPv4 address in dotted-quad format.
<code>subnetMask</code>	string	The IPv4 subnet mask in dotted quad format.
<code>defaultGateway</code>	string	The device's IPv4 default gateway in dotted quad format.

device.query

Returns high level status information about the device. Accepts no parameters.

Returned data

Parameter name	Type	Short description
<code>currentTime</code>	dateTime. iso8601	The system's current time (UTC).
<code>restartTime</code>	dateTime. iso8601	The date and time when the system was last restarted.
<code>serial</code>	string	The serial number of this device or 'unknown'.
<code>softwareVersion</code>	string	The version number of the software running on the device.
<code>buildVersion</code>	string	The build version of the software running on the device.
<code>model</code>	string	The model number.
<code>apiVersion</code>	string	The version number of the API implemented by this device.
<code>activatedFeatures</code>	array	Each member contains a string named <code>feature</code> containing a short description of that feature, for example, Encryption . more...
<code>clusterType</code>	string	The role that this MCU plays in a cluster. One of master , slave , or unclustered . The parameter is absent if the device is incapable of belonging to a cluster.
<code>maxConferenceSize</code>	integer	The maximum number of participants that can be hosted in a single conference at the time of the response.
<code>totalVideoPorts</code>	integer	The total number of video ports on the device.
<code>totalAudioOnlyPorts</code>	integer	The total number of additional audio-only ports on the device.
<code>totalStreamingAndContentPorts</code>	integer	The total number of streaming and content ports on the MCU. Only provided if non-zero.
<code>portReservationMode</code>	string	Defines whether port reservation mode is enabled or disabled . Corresponds to the Media port reservation setting on the web interface. Only present on MCU products.
<code>maxVideoResolution</code>	string	Either cif or 4cif . more...
<code>videoPortAllocation</code>	array	An array of structs , each of which defines the type and count of video ports that are allocated on this MCU.
<i>type (videoports)</i>	string	One of nhd , sd , hd , hdPlus or fullhd
<i>count (videoports)</i>	integer	The allocated number of video ports of this type .
<code>shutdownStatus</code>	string	Indicates the status of a shutdown operation. One of shutdown , shutdownInProgress , or notShutdown .
<code>rebootRequired</code>	boolean	The device returns this parameter as true if it needs to reboot. more...

finishedBooting	boolean	true after the device is fully booted. Will not revert to false until a reboot starts.
mediaResources	integer	The percentage of DSP resources that are available (i.e. successfully booted and not failed) to the unclustered device or the master blade of a cluster. Slave blades don't return this value.

device.restart

Restarts the device, or shuts it down without a restart.

Input parameters

Optional or conditional inputs

Parameter name	Type	Short description
<code>shutdownOnly</code>	boolean	If <code>true</code> , the device will shut down when it receives <code>device.restart</code> and will not restart. Defaults to <code>false</code> .

device.restartlog.query

Returns the restart log - also known as the system log on the web interface.

Returned data

Parameter name	Type	Short description
<code>log</code>	array	Each member of the array contains log information (called system log in the user interface).
<code>time (restart log)</code>	dateTime. iso8601	The date and time when the device restarted. For example, 20110119T13:52:42 is in the format <code>yyyymmddThh:mm:ss</code> .
<code>reason</code>	string	An explanation for the restart. One of: User requested shutdown User requested reboot from web interface User requested upgrade User requested reboot from console User requested reboot from API User requested reboot from FTP User requested shutdown from supervisor User requested reboot from supervisor User reset configuration Cold boot unknown

device.status

This command takes no data inputs, although it must be authenticated like all other commands. It is a lightweight status command that does not conform to the common struct-based format (described in [XML-RPC implementation \[p.5\]](#)) that is used in all other commands.

device.status returns a correctly formatted XML-RPC **methodResponse** that contains only one value - a string that is a delimited list of status monitors and their values.

Note: This command is solely intended for troubleshooting and is subject to change at any time. Your applications should not rely on the format of the command or its returned data.

Returned data

Always returned

Parameter name	Type	Short description
unnamed (<i>device.status</i>)	string	A semi-colon delimited list of status monitors and their values at the time of the response.

Example response

```
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
  <params>
    <param>
      <value>
        <string>
          uptime=85952;cpu_load=0.2%;media_load=0%;video_load=0%;audio_load=0%;free_
memory=119192K;free=10712K;cached_fbs=108480k;one=37.5;two=31.0;
        </string>
      </value>
    </param>
  </params>
</methodResponse>
```

device.time.modify

Modifies the device's time settings.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>currentTime</code>	dateTime. iso8601	The system's current time (UTC).
<code>ntpEnabled</code>	boolean	Defines whether or not the device may synchronize with an NTP server.
<code>utcOffsetHours</code>	integer	Number between -12 and +14 (inclusive) that, together with <code>utcOffsetMinutes</code> , defines the UTC offset of the device's clock.
<code>utcOffsetMinutes</code>	integer	Number between 0 and 59 (inclusive) that, together with <code>utcOffsetHours</code> , defines the UTC offset of the device's clock.
<code>ntpHost</code>	string	DNS or IP address of an NTP server

device.time.query

Queries the device for its time settings.

Parameter name	Type	Short description
<code>currentTime</code>	dateTime. iso8601	The system's current time (UTC).
<code>ntpEnabled</code>	boolean	Defines whether or not the device may synchronize with an NTP server.
<code>utcOffsetHours</code>	integer	Number between -12 and +14 (inclusive) that, together with <code>utcOffsetMinutes</code> , defines the UTC offset of the device's clock.
<code>utcOffsetMinutes</code>	integer	Number between 0 and 59 (inclusive) that, together with <code>utcOffsetHours</code> , defines the UTC offset of the device's clock.
<code>ntpHost</code>	string	DNS or IP address of an NTP server
<code>ntpStatus</code>	string	The NTP client's current status; one of <code>disabled</code> , <code>synchronizing</code> , <code>synchronized</code> or <code>error</code> .

feedbackReceiver.configure

This call configures the device to send feedback about the specified **events** to the specified **receiverURI**. See the list of [Feedback events \[p.18\]](#) when you define the **events** struct.

If you omit the **events** struct, then the receiver will be configured to receive the default notification messages (all notifications except **activeSpeakerChanged**)

Input parameters

Required inputs

Parameter name	Type	Short description
receiverURI	string	Fully-qualified URI that identifies the listening application's XML-RPC interface (protocol, address, and port), for example, <code>http://tms1:8080/RPC2</code> . Must end in <code>/RPC2</code> (see XML-RPC.com). You can use <code>http</code> or <code>https</code> and, if no port number is specified, the device will use the protocol defaults (80 and 443 respectively).

Optional or conditional inputs

Parameter name	Type	Short description
sourceIdentifier	string	The originating device uses this parameter to identify itself to the listening receiver/s.

If **sourceIdentifier** is not explicitly set, the device identifies itself with the MAC address of its Ethernet port A interface.

receiverIndex	integer	A number between 1 and 20 defining the position of this feedback receiver in the device's table of feedback receivers.
----------------------	---------	--

Set this to `-1` to use any available position. The value `1` is assumed if you don't supply **receiverIndex** - *which will overwrite any existing entry in position 1.*

events (feedback)	struct	Each member of the events struct associates a string (feedback event name) to a boolean (true to subscribe). events (feedback) [p.226]
--------------------------	--------	---

Returned data

Parameter name	Type	Short description
receiverIndex	integer	A number between 1 and 20 defining the position of this feedback receiver in the device's table of feedback receivers.
status (success)	string	Operation successful

The call returns the allocated **receiverIndex**.

feedbackReceiver.query

This call asks the device for a list of all the feedback receivers that have previously been configured. It does not accept parameters other than the authentication strings.

Returned data

Always returned

If there are no feedback receivers to enumerate, then **feedbackReceiver.query** returns an empty **receivers** array.

Parameter name	Type	Short description
receivers	array	An array of feedback receivers, with members corresponding to the entries in the receivers table on the device's web interface.

Conditionally returned

If **receivers** is not empty, then each receiver in the response contains the following parameters:

Parameter name	Type	Short description
receiverURI	string	Fully-qualified URI that identifies the listening application's XML-RPC interface (protocol, address, and port), for example, <code>http://tms1:8080/RPC2</code> . Must end in <code>/RPC2</code> (see XML-RPC.com). You can use <code>http</code> or <code>https</code> and, if no port number is specified, the device will use the protocol defaults (80 and 443 respectively).
sourceIdentifier	string	The originating device uses this parameter to identify itself to the listening receiver/s.

If **sourceIdentifier** is not explicitly set, the device identifies itself with the MAC address of its Ethernet port A interface.

index (feedback receiver)	integer	A number between 1 and 20 (inclusive) that indicates the position of this feedback receiver in the device's table of feedback receivers.
----------------------------------	---------	--

feedbackReceiver.reconfigure

This call reconfigures an existing feedback receiver. This call only reconfigures the receiver parameters that you specify; the MCU retains the original values for any parameters that you omit.

See [Feedback events \[p.18\]](#) for a list of events published by the MCU to which receivers can subscribe.

The call returns a fault if there is no feedback receiver at the specified `receiverIndex`.

If you omit `receiverURI` altogether, the original value persists. However if you supply an empty `receiverURI`, the call generates a fault.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>receiverIndex</code>	integer	A number between 1 and 20 defining the position of this feedback receiver in the device's table of feedback receivers.

Optional or conditional inputs

Parameter name	Type	Short description
<code>receiverURI</code>	string	Fully-qualified URI that identifies the listening application's XML-RPC interface (protocol, address, and port), for example, <code>http://tms1:8080/RPC2</code> . Must end in <code>/RPC2</code> (see XML-RPC.com). You can use <code>http</code> or <code>https</code> and, if no port number is specified, the device will use the protocol defaults (80 and 443 respectively).
<code>sourceIdentifier</code>	string	The originating device uses this parameter to identify itself to the listening receiver/s.
<code>events</code> (<i>feedback</i>)	struct	Each member of the <code>events</code> struct associates a string (feedback event name) to a boolean (true to subscribe). events (feedback) [p.226]

feedbackReceiver.remove

Removes the specified feedback receiver.

The call returns a fault if there is no feedback receiver at the specified **receiverIndex**.

Input parameters

Required inputs

Parameter name	Type	Short description
receiverIndex	integer	A number between 1 and 20 defining the position of this feedback receiver in the device's table of feedback receivers.

gatekeeper.modify

Modifies the device's H.323 gatekeeper settings.

Input parameters

Optional or conditional inputs

Parameter name	Type	Short description
<code>gatekeeperUsage</code>	string (8)	Defines how the gatekeeper is used. One of disabled , enabled , or required . more...
<code>address</code> (<i>gatekeeper</i>)	string (255)	The address of the gatekeeper. It may be a DNS hostname or an IP address.
<code>registrationType</code>	string	The gatekeeper registration type. One of gateway , terminalGateway , gatewayCisco , mcuStandard , or mcuCompatible . more...
<code>portAssociationAv4</code>	boolean	true if interface 'PortA IPv4' is associated with the H.323 gatekeeper.
<code>portAssociationAv6</code>	boolean	true if interface 'PortA IPv6' is associated with the H.323 gatekeeper.
<code>portAssociationBv4</code>	boolean	true if interface 'PortB IPv4' is associated with the H.323 gatekeeper.
<code>portAssociationBv6</code>	boolean	true if interface 'PortB IPv6' is associated with the H.323 gatekeeper.
<code>h323ID</code>	string (255)	The H.323 ID used by the device to register with the gatekeeper.
<code>usePassword</code>	boolean	Indicates whether or not the device uses its configured password for gatekeeper registration.
<code>password</code> (<i>gatekeeper</i>)	string	The password that the device uses to register with the gatekeeper, if required.
<code>registrationPrefix</code>	string (255)	A string of digits that serves as the device's registration prefix.
<code>mcuServicePrefix</code>	string	The service prefix used by the MCU.
<code>scheduledConferenceIDRegistration</code>	string (8)	Defines whether or not ID registration is enabled for scheduled conferences. Either enabled or disabled . Corresponds to the ID registration for scheduled conferences option on the web interface.
<code>sendResourceAvailabilityIndications</code>	boolean	Defines whether or not the MCU will send resource availability indications.
<code>availabilityThresholdConferences</code>	string (8)	A threshold beyond which the device will stop indicating resource availability. It is a number between 0 and the maximum number of conferences that can be hosted on the device. You can set this string to a number or all in a <code>gatekeeper.modify</code> call.

availabilityThresholdVideoPorts	string (8)	A threshold beyond which the device stops indicating resource availability. It is a number between 0 and the maximum number of video ports available on the device. You can set this string to a number or all in a gatekeeper.modify call.
--	------------	--

gatekeeper.query

Retrieves the gatekeeper settings and current status of the device.

Returned data

Always returned

Parameter name	Type	Short description
<code>gatekeeperUsage</code>	string (8)	Defines how the gatekeeper is used. One of disabled , enabled , or required . more...

Conditionally returned

The following parameters are not present if `gatekeeperUsage` is **disabled**.

Parameter name	Type	Short description
<code>address (gatekeeper)</code>	string (255)	The address of the gatekeeper. It may be a DNS hostname or an IP address.
<code>dnsStatus</code>	string	The status of the DNS lookup of the gatekeeper's address. One of inProgress , resolved , or failed .
<code>ip</code>	string	the IP address of the gatekeeper (if <code>dnsStatus</code> is resolved)
<code>activeRegistrations</code>	integer	The number of active registrations.
<code>pendingRegistrations</code>	integer	The number of registrations in progress
<code>registrationPrefix</code>	string (255)	A string of digits that serves as the device's registration prefix.
<code>registrationType</code>	string	The gatekeeper registration type. One of gateway , terminalGateway , gatewayCisco , mcuStandard , or mcuCompatible . more...
<code>portAssociationAv4</code>	boolean	true if interface 'PortA IPv4' is associated with the H.323 gatekeeper.
<code>portAssociationAv6</code>	boolean	true if interface 'PortA IPv6' is associated with the H.323 gatekeeper.
<code>portAssociationBv4</code>	boolean	true if interface 'PortB IPv4' is associated with the H.323 gatekeeper.
<code>portAssociationBv6</code>	boolean	true if interface 'PortB IPv6' is associated with the H.323 gatekeeper.
<code>sendResourceAvailabilityIndications</code>	boolean	Defines whether or not the MCU will send resource availability indications.

<code>availabilityThresholdConferences</code>	integer	A threshold beyond which the device will stop indicating resource availability. It is a number between 0 and the maximum number of conferences that can be hosted on the device. This threshold value is returned as an integer by <code>gatekeeper.query</code> . It is not returned if it has been set to <code>all</code> . It is not returned if the MCU is not configured to send resource availability indications.
<code>availabilityThresholdVideoPorts</code>	integer	A threshold beyond which the device stops indicating resource availability. It is a number between 0 and the maximum number of video ports available on the device. This threshold value is returned as an integer by <code>gatekeeper.query</code> . It is not returned if it has been set to <code>all</code> . It is not returned if the MCU is not configured to send resource availability indications.
<code>registeredAddress</code>	string	The IP address and port that the MCU has registered with the gateway. This value is only returned if the MCU is registered.
<code>alternateGatekeepers</code>	integer	The number of alternate gatekeepers
<code>resourceAvailabilityStatus</code>	string	Indicates the availability of resources on the MCU. One of <code>available</code> , <code>unavailable</code> , or <code>disabled</code> (resource availability indications are not enabled).
<code>h323ID</code>	string (255)	The H.323 ID used by the device to register with the gatekeeper.
<code>mcuServicePrefix</code>	string	The service prefix used by the MCU.
<code>scheduledConferenceIDRegistration</code>	string (8)	Defines whether or not ID registration is enabled for scheduled conferences. Either <code>enabled</code> or <code>disabled</code> . Corresponds to the ID registration for scheduled conferences option on the web interface.
<code>h323IDStatus</code>	string	The current status of the ID registration process. more...
<code>mcuServicePrefixStatus</code>	string	The current status of the service prefix registration process. more...
<code>usePassword</code>	boolean	Indicates whether or not the device uses its configured password for gatekeeper registration.

Deprecated parameters

These are replaced by their explicitly named v4 equivalents.

Parameter name	Type	Short description
<code>portAssociationA</code>	boolean	<code>true</code> if interface 'PortA IPv4' is associated with the H.323 gatekeeper.
<code>portAssociationB</code>	boolean	<code>true</code> if interface 'PortB IPv4' is associated with the H.323 gatekeeper.

gateway.enumerate

Enumerates configured H.323 gateways on the device.

Input parameters

Optional or conditional inputs

Parameter name	Type	Short description
<code>enumerateID</code>	string	The device returns this index if the requested data is too large for one response. Pass this parameter in a repeat of the call to return the next batch of data. more...

Returned data

If there are entries to return, the method returns them in an array. If there are more entries than can be returned in one response, you'll get the next `enumerateID` up from the one you provided.

Parameter name	Type	Short description
<code>enumerateID</code>	string	The device returns this index if the requested data is too large for one response. Pass this parameter in a repeat of the call to return the next batch of data. more...
<code>gateways</code>	array	A collection of structures, each of which describes a gateway.
<code>name (gateway)</code>	string	The name of the gateway.
<code>address (gateway)</code>	string (63)	The address of the gateway.
<code>conferencingParameters</code>	struct	A structure containing the conferencing parameters of the enumerated item, e.g. gateway or endpoint.
<code>useDefaultMotionSharpness</code>	boolean	true means this endpoint will use box-wide default motion sharpness settings.
<code>minFrameRateMotionSharpness</code>	integer	Specifies the minimum frame rate for this endpoint. This parameter is only present if useDefaultMotionSharpness is false .
<code>maxMediaTxBitRate</code>	integer	The maximum media transmission speed from this device, in kbps. 0 means the device uses the default.
<code>maxMediaRxBitRate</code>	integer	The maximum media reception speed of this device, in kbps. 0 means the device uses the default.

participant.add

Adds a participant to a conference. All participants in a conference must have a **participantName** that is unique to the conference but it need not be unique across all conferences. Add the participant as type **by_address** unless you are adding the participant to an ad hoc conference.

Send the **addResponse** parameter if you want the call to return the details of the added participant (in a **participant** struct.)

Participants can be added before or during a conference. A participant which is added at any time via the API will be added to the configured list of participants, and thus will be called at the start of the conference by the MCU for any conference which has any sort of repetition; to avoid this, a participant must be removed directly using **participant.remove**. Also, the MCU allows a maximum of 500 API-configured participants, and participants will persist after conferences have ended unless you use [participant.remove \[p.132\]](#).

Note: If a **participantName** matches the name of an endpoint in the list of configured endpoints (go to **Endpoints** in the web interface) the two are not necessarily related. This is because the MCU uses the combination of both **participantName** and **participantType** to ensure unique participants.

Input parameters

Required inputs

Parameter name	Type	Short description
conferenceName	string	The name of the conference.
participantName	string	The unique name of a participant. more...

Optional or conditional inputs

All of the following parameters are optional, and control the conferencing behavior of the MCU with respect to the endpoint in question; for example, the maximum resolution of the video streams used, or whether the participant is able to control their conference view layout.

Parameter name	Type	Short description
addResponse	boolean	true to return the details of the added participant.
participantProtocol	string	h323 , sip , or vnc .
participantType	string	One of: by_address or ad_hoc . more...
address (endpoint)	string (63)	The address of the endpoint; may be hostname, IP address, E.164 number, SIP URI, or H.323 ID.
gatewayAddress	string (63)	The address of an H.323 gateway, if required. Only used if protocol is h323 . This corresponds to the address parameter of the gateway as returned by gateway.enumerate .
useSIPRegistrar	boolean	Not valid unless the protocol is SIP. true if the endpoint uses the SIP registrar. Defaults to false .

<code>transportProtocol</code>	string	Defines the SIP transport protocol. This parameter is ignored if the communication protocol is not SIP. One of <code>default</code> , <code>tcp</code> , <code>udp</code> , or <code>tls</code> .
<code>redial</code>	string	Defines the MCU's redial behavior when calls out to this participant drop. One of <code>never</code> , <code>connect</code> , <code>unexpected</code> , <code>any</code> , or <code>default</code> . more...
<code>redialLimit</code>	string	Defines whether a redial limit is used with the redial behavior. One of <code>enabled</code> , <code>disabled</code> , or <code>default</code> . more...
<code>password</code>	string	The password for VNC endpoints.
<code>deferConnection</code>	boolean	If <code>true</code> , don't call out to this participant immediately, but wait for a <code>participant.connect</code> command. You cannot set <code>deferConnection</code> to true for participants where <code>participantType</code> is <code>ad_hoc</code> .
<code>addAsGuest</code>	boolean	Defines whether the MCU designates guest or chair status to the participant when it invites the participant in to the conference. <code>true</code> means the participant joins as a guest when invited in; <code>false</code> means the participant joins as a chair when invited in.
<code>actAsRecorder</code>	boolean	Defines whether this participant appears as a recorder to other participants.
<code>maxBitRateToMCU</code>	integer	Maximum bandwidth to the MCU (kbps).
<code>maxBitRateFromMCU</code>	integer	Maximum bandwidth from the MCU (kbps).
<code>motionSharpnessTradeoff</code>	string	Defines preference for motion vs. sharpness. One of <code>preferMotion</code> , <code>preferSharpness</code> , <code>balanced</code> , or <code>default</code> . more...
<code>displayNameOverrideStatus</code>	boolean	<code>true</code> if the endpoint uses the <code>displayNameOverrideValue</code> text to identify itself to other participants.
<code>displayNameOverrideValue</code>	string	This value overrides the participant's display name if <code>displayNameOverrideStatus</code> is <code>true</code> .
<code>cpLayout</code>	string	This sets the initial conference view layout for the video sent to the participant. Refer to Conference layouts [p.188] for details.
<code>layoutControlEx</code>	string	Defines how the view layout can be controlled. One of <code>disabled</code> , <code>feccOnly</code> , <code>dtmfOnly</code> , <code>feccWithDtmfFallback</code> , or <code>bothFeccAndDtmf</code> . more...
<code>audioRxMuted</code>	boolean	<code>true</code> means that audio from this participant will not be heard by other conference participants.
<code>audioTxMuted</code>	boolean	<code>true</code> means that the MCU is not transmitting the audio part of the conference to this participant.

Note: The endpoint may not always detect DTMF tones from the MCU after you mute the outgoing audio.

<code>audioRxGainMode</code>	string	<code>none</code> , <code>automatic</code> , <code>default</code> , or <code>fixed</code> . more...
<code>audioRxGainMillidB</code>	integer	If audio gain mode is fixed, this is the number of decibels of gain applied, multiplied by 1000, and can be a negative value.

videoRxMuted	boolean	true means that video from this participant will not be seen by other conference participants.
videoTxMuted	boolean	true means that the MCU does not send the video part of the conference to this participant.
videoTxWidescreen	boolean	If true , the MCU sends video in a form suitable for a widescreen 16:9 display to this participant.
videoTxMaxResolution	string	The maximum resolution transmitted to this endpoint. One of cif , 4cif , or max . more...
videoRxMaxResolution	string	The maximum resolution of the received video. One of cif , 4cif , or max . more...
autoConnect	boolean	true allows endpoints to automatically connect to this conference when they dial in and are recognized. more...
autoDisconnect	boolean	true allows the device to automatically disconnect the endpoint, and all remaining endpoints that have this property, when none of the remaining endpoints require manual disconnection. false means this endpoint requires manual disconnection. When a participant disconnects from a conference and only participants who have autoDisconnect set to true remain, the MCU disconnects all the remaining participants.
borderWidth	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. more...
dtmfSequence	string (127)	A string of characters that will be converted to DTMF signals, allowing the device to navigate through audio menus. The sequence may contain 0–9, *, #, and ,. The comma becomes a two second pause. more...
suppressAudioDuringDTMF	string	outgoing or all defines which audio the MCU suppresses while it sends the DTMF connection sequence to the endpoint. more...
linkType	string	This parameter is ignored unless participantType is by_address . Either cascadeSlaveToMaster or default
suppressDtmfEx	string	Controls the muting of in-band DTMF tones. One of fecc , always , or never . more...
h239Negotiation	string	Defines how the MCU presents itself for h239 token negotiation. One of As master , As slave , or Mimic slave . more...
videoToUse	struct	Collection of parameters that uniquely identify the participant whose video will display in place of this participant's video by default.

To define a participant whose video source will display by default in place of this participant's video, you need to populate the **videoToUse** struct with the following parameters:

participantName	string	The unique name of a participant. more...
participantType	string	One of: by_address or ad_hoc . more...
participantProtocol	string	h323 , sip , or vnc .

Returned data

Conditionally returned

Parameter name	Type	Short description
<code>participant</code>	struct	Contains the parameters that, when considered together, uniquely identify a participant.
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantType</code>	string	One of: <code>by_address</code> or <code>ad_hoc</code> . more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>conferenceName</code>	string	The name of the conference.
<p>If the participant is in a conference, the response includes the <code>conferenceName</code>; if the participant is in an autoattendant, the response includes the <code>autoAttendantUniqueId</code> instead. The response does not include both parameters.</p>		
<code>autoAttendantUniqueId</code>	string	Unique identifier for the auto attendant.

Deprecated parameters

Parameter name	Type	Short description
<code>layoutControlEnabled</code>	boolean	Deprecated by <code>layoutControlEx</code> . Defines whether the endpoint's participant will have control over the layout if <code>layoutControlDefault</code> is <code>false</code> . more...

participant.connect

Used primarily for API-configured participants with `deferConnection` set to `TRUE`, but can also be used to reconnect disconnected participants.

Input parameters

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.
<code>autoAttendantUniqueID</code>	string	Unique identifier for the auto attendant.
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...

participant.diagnostics

Returns diagnostic information about a given participant.

Input parameters

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.
<code>autoAttendantUniqueID</code>	string	Unique identifier for the auto attendant.
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...

Returned data

Parameter name	Type	Short description
<code>videoTxFrameRate</code>	integer	Frame rate of the transmitted video (frames per second).
<code>videoRxFrameRate</code>	integer	The frame rate of the received video (frames per second).
<code>videoRxFramesReceived</code>	integer	The number of video frames received from this endpoint.
<code>videoTxChannelBitRate</code>	integer	The negotiated available bandwidth for the video stream going to the endpoint.
<code>videoTxSelectedBitRate</code>	integer	The bit rate at which the MCU is attempting to send video to this endpoint (bits per second). This value may be lower than <code>videoTxChannelBitRate</code> which is an effective maximum.
<code>videoTxActualBitRate</code>	integer	The most recently measured bit rate of the outgoing video stream to this endpoint (bits per second).
<code>videoTxBitRateLimitReason</code>	string	Indicates why the bit rate of the transmitted video stream was limited by the device. One of <code>notLimited</code> , <code>viewedSize</code> , <code>quality</code> , <code>aggregateBandwidth</code> , <code>flowControl</code> , or <code>endpointLimitation</code> .
<code>videoRxChannelBitRate</code>	integer	The negotiated available bandwidth for the video stream coming from the endpoint.
<code>videoRxSelectedBitRate</code>	integer	The bit rate which the MCU has requested for the video stream from this endpoint (bits per second).
<code>videoRxActualBitRate</code>	integer	The most recently measured bit rate of the incoming video stream from this endpoint (bits per second).
<code>videoRxBitRateLimitReason</code>	string	Indicates why the bit rate of the received video stream was limited by the device. more...
<code>videoTxWidth</code>	integer	Width in pixels of the transmitted video.
<code>videoTxHeight</code>	integer	Height in pixels of the transmitted video.

videoTxInterlaced	boolean	<code>true</code> if the MCU is sending interlaced video to this endpoint.
videoRxWidth	integer	Width in pixels of the received video.
videoRxHeight	integer	Height in pixels of the received video.
videoRxInterlaced	boolean	<code>true</code> if the MCU is receiving interlaced video from this endpoint.
videoTxReportedLost	integer	The count of video packets reported lost by the far end.
videoRxCodec	string	The codec used on the received video.
videoRxJitter	integer	Represents the variability of the timing of received video packets.
audioTxReportedLost	integer	The count of audio packets reported lost by the far end.
videoTxSent	integer	Count of the video packets sent to the endpoint.
audioRxLost	integer	Count of the audio packets lost by the MCU.
audioRxReceived	integer	Count of audio packets received by the MCU.
videoTxCodec	string	The codec used on the transmitted video.
videoRxFramesReceivedWithErrors	string	The number of video frames received from this endpoint that were not successfully decoded.
audioTxSent	integer	Count of the audio packets sent to this endpoint.
videoRxReceived	integer	Count of video packets received from this endpoint.
videoRxLost	integer	Count of video packets lost en route to the MCU from this endpoint.
contentRxType	string	Type of content received. One of <code>none</code> , <code>h239</code> , or <code>bfcp</code> . more...
contentRxCodec	string	The codec used on the incoming content stream.
contentRxWidth	integer	Horizontal resolution of incoming content.
contentRxHeight	integer	Vertical resolution of incoming content
contentRxFrameRate	integer	Frame rate of incoming content
contentRxActualBitRate	integer	Actual speed of incoming content in bps
contentRxChannelBitRate	integer	Capacity of channel in bps
contentRxSelectedBitRate	integer	Participant-selected content bitrate. If one is not set, the MCU assumes the content should be received as fast as possible.
contentRxBitRateLimitReason	string	Indicates why the bit rate of the received content stream was limited by the device. more...
contentRxJitter	integer	A measure of the jitter in the received content
contentRxFramesReceived	integer	Number of received content frames
contentRxFramesReceivedWithErrors	integer	Number of received content frames that had errors
contentRxReceived	integer	Number of content packets received from this participant.
contentRxLost	integer	Number of content packets that should have been received from this participant that were not.

<code>contentTxType</code>	string	Type of content transmitted. One of <code>none</code> , <code>h239</code> , <code>bfcf</code> , or <code>mainVideo</code> . more...
<code>contentTxCodec</code>	string	The codec used to transmit content. If content is being transcoded, it is the output format of the transcoder; either <code>h263+</code> , <code>h264</code> , or <code>automatic</code> (default). This setting does not apply in <code>passthrough</code> mode. more...
<code>contentTxWidth</code>	integer	Horizontal resolution of outgoing content
<code>contentTxHeight</code>	integer	Vertical resolution of outgoing content
<code>contentTxFrameRate</code>	integer	Frame rate of outgoing content
<code>contentTxActualBitRate</code>	integer	Actual speed of outgoing content in bps
<code>contentTxChannelBitRate</code>	integer	Capacity of channel in bps
<code>contentTxSelectedBitRate</code>	integer	Participant-selected content bitrate. If one is not set, the MCU assumes the content should be sent as fast as possible.
<code>contentTxBitRateLimitReason</code>	string	Indicates why the bit rate of the transmitted content stream was limited by the device. more...
<code>contentTxSent</code>	integer	Number of content packets sent.
<code>contentTxReportedLost</code>	integer	Number of content packets reported as lost.
<code>contentTxError</code>	string	Provides a reason for a content transmission error. more...

participant.disconnect

This call causes the MCU to tear down its connection to the specified participant, if such a connection exists. This is different from `participant.remove` above because:

- In the case of configured participants, it does not remove the configuration (thus allowing later re-connection with `participant.connect`).
- In the case of ad hoc participants, it does not remove the record of the previous connection.

Input parameters

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.
<code>autoAttendantUniqueID</code>	string	Unique identifier for the auto attendant.
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...

participant.enumerate

Returns data about participants in conferences on the MCU. Several calls may be required to receive data about all participants; see the notes on `enumerateID` below.

Note: The device will respond to `participant.enumerate` if you omit `operationScope`. However, this behavior is deprecated and may not be supported in future versions. See [participant.enumerate \(deprecated\) \[p.121\]](#) for details.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>operationScope</code>	array	The array should contain one or two string parameters. That is, it should contain either or both of the strings <code>currentState</code> or <code>configuredState</code> . more...

Optional or conditional inputs

Parameter name	Type	Short description
<code>enumerateID</code>	string	The device returns this index if the requested data is too large for one response. Pass this parameter in a repeat of the call to return the next batch of data. more...
<code>lastRevision</code>	integer	This number identifies an earlier set of enumeration data to compare against your current call. If you supply this parameter using the <code>currentRevision</code> value returned by a previous enumeration, the current <code>enumerate</code> call will return only the differences since that previous call. If you don't supply this parameter, the device assumes that you want a full enumeration.
<code>enumerateFilter</code>	string	A filter expression. The enumeration results depend on the supplied expression.

`enumerateFilter` filters on:

Parameter name	Type	Short description
<code>dormant</code>	boolean	<code>true</code> if the pre-configured participant is not trying to connect.
<code>connecting</code>	boolean	<code>true</code> if the scheduled participant is in the process of connecting or is pending a retry. <code>connecting</code> is <code>true</code> for participants whose <code>callStateEx</code> values are <code>proceeding</code> , <code>alerting</code> , or <code>pending</code> . It may also be true for some participants whose <code>callState</code> (deprecated) is <code>dormant</code> or <code>disconnected</code> , because these values are also mapped to the new <code>proceeding</code> and <code>pending</code> states, respectively, that were introduced by the persistence feature in MCU 4.4.

connected	boolean	true if the participant is currently connected to a conference.
disconnected	boolean	true if the participant has been connected to a conference, but is now disconnected.

Returned data

Conditionally returned

The response only includes the **participants** array if there are participants to enumerate.

Note: This participant information is returned for all participants added to the conference using the **participant.add** call, even after they have disconnected. However, this information is only returned for other participants (i.e. those added via the web interface or those who dialled into the conference) whilst they are connected but not after they have disconnected.

If there are participants to enumerate, the response may include some or all of the following data:

Parameter name	Type	Short description
enumerateID	string	The device returns this index if the requested data is too large for one response. Pass this parameter in a repeat of the call to return the next batch of data. more...
currentRevision	integer	A number that indicates the current revision of this enumeration. You can use this as a lastRevision input to a future enumerate call to retrieve only the changes between the two enumerations.
participants	array	An array of structures that represent participants.

Members of the **participants** array may contain the following data:

participantName	string	The unique name of a participant. more...
participantProtocol	string	h323 , sip , or vnc .
participantType	string	One of: by_address , by_name , or ad_hoc . more...
conferenceName	string	The name of the conference.

If the participant is in a conference, the response includes the **conferenceName**; if the participant is in an autoattendant, the response includes the **autoAttendantUniqueId** instead. The response will not include both parameters.

autoAttendantUniqueId	string	Unique identifier for the auto attendant.
connectionUniqueId	integer	Corresponds to the uniqueId returned by a conference or autoattendant.
currentState	struct	The current state of the participant. This is only present if requested in the operationScope .

The **currentState** structure may or may not be included in the participant structure, and it may be empty if it is included, depending on the provided value of **operationScope** and whether there is any data to return. [Details of the struct](#) are listed below.

configuredState	struct	The stored configuration of the participant, if it exists. configuredState is only present if requested in the operationScope .
------------------------	--------	---

The **configuredState** structure may or may not be included in the participant structure, and it may be empty if it is included, depending on the provided value of **operationScope** and whether there is any data to return. [Details of the struct](#) are listed below.

The **currentState** structure

Parameter name	Type	Short description
address (<i>endpoint</i>)	string (63)	The address of the endpoint; may be hostname, IP address, E.164 number, SIP URI, or H.323 ID.
gatewayAddress	string (63)	The address of an H.323 gateway, if required. Only used if protocol is h323 . This corresponds to the address parameter of the gateway as returned by gateway.enumerate .
ipAddress	string	IPv4 address in dotted-quad format.
This is the IP address to which the MCU is connected for this endpoint; it will usually be the endpoint itself, but may be a gatekeeper or gateway.		
displayName	string	The display name of the participant.
If this parameter is longer than 31 characters, only the first 31 characters are returned.		
guest	boolean	true if the participant is a guest, false if the participant is a chair.
remoteLinkType	string	One of slave , conference , autoAttendant , recording , or playback .
displayNameOverrideStatus	boolean	true if the endpoint uses the displayNameOverrideValue text to identify itself to other participants.
maxBitRateToMCU	integer	Maximum bandwidth to the MCU (kbps).
maxBitRateFromMCU	integer	Maximum bandwidth from the MCU (kbps).
motionSharpnessTradeoff	string	Defines preference for motion vs. sharpness. One of preferMotion , preferSharpness , balanced , or default . more...
callStateEx	string	One of dormant , proceeding , alerting , connected , pending , or disconnected . more...
connectTime	dateTime. iso8601	Only returned after the participant is connected. This value is always present if the call state is connected. It may or may not be defined for participants in the disconnected state, depending on whether they were ever connected.
disconnectTime	dateTime. iso8601	Only returned after the participant has disconnected.
disconnectReason	string	Only returned after the participant has disconnected; this contains one of the Disconnect reasons [p.185] .

<code>connectPending</code>	boolean	true if sending a "participant.connect" command for this participant will cause either the initial connection to that endpoint (in the event that it was configured with "deferConnection" set) or a re-connection to that endpoint (in the event that it has disconnected).
<code>redial</code>	string	Defines the MCU's redial behavior when calls out to this participant drop. One of never , connect , unexpected , any , or default . more...
<code>redialLimit</code>	string	Defines whether a redial limit is used with the redial behavior. One of enabled , disabled , or default . more...
<code>audioRxCodec</code>	string	Receive audio codec.
<code>audioRxLost</code>	integer	Count of the audio packets lost by the MCU.
<code>audioRxReceived</code>	integer	Count of audio packets received by the MCU.
<code>audioRxMuted</code>	boolean	true means that audio from this participant will not be heard by other conference participants.
<code>audioRxMutedRemotely</code>	boolean	Whether this endpoint is muted remotely.
<code>audioRxGainMode</code>	string	none , automatic , default , or fixed . more...
<code>audioRxGainMillidB</code>	integer	If audio gain mode is fixed, this is the number of decibels of gain applied, multiplied by 1000, and can be a negative value.
<code>audioTxCodec</code>	string	The codec used on the audio transmission.
<code>audioTxReportedLost</code>	integer	The count of audio packets reported lost by the far end.
<code>audioTxSent</code>	integer	Count of the audio packets sent to this endpoint.
<code>audioTxMuted</code>	boolean	true means that the MCU is not transmitting the audio part of the conference to this participant.
<code>audioRxEnergyMillidB</code>	integer	The measured energy of a participant's audio sent to the MCU. Typically this will be a negative value in the range -30000 (-30dB for very quiet) and 0 (very loud).
<code>videoRxCodec</code>	string	The codec used on the received video.
<code>videoRxLost</code>	integer	Count of video packets lost en route to the MCU from this endpoint.
<code>videoRxMuted</code>	boolean	true means that video from this participant will not be seen by other conference participants.
<code>videoRxReceived</code>	integer	Count of video packets received from this endpoint.
<code>videoTxCodec</code>	string	The codec used on the transmitted video.
<code>videoTxReportedLost</code>	integer	The count of video packets reported lost by the far end.
<code>videoTxSent</code>	integer	Count of the video packets sent to the endpoint.
<code>videoTxMuted</code>	boolean	true means that the MCU does not send the video part of the conference to this participant.
<code>videoTxWidescreen</code>	boolean	If true , the MCU sends video in a form suitable for a widescreen 16:9 display to this participant.
<code>contentRxType</code>	string	Type of content received. One of none , h239 , or bfcp . more...

<code>contentRxCodec</code>	string	The codec used on the incoming content stream.
<code>contentRxReceived</code>	integer	Number of content packets received from this participant.
<code>contentRxLost</code>	integer	Number of content packets that should have been received from this participant that were not.
<code>contentTxType</code>	string	Type of content transmitted. One of <code>none</code> , <code>h239</code> , <code>bfcf</code> , or <code>mainVideo</code> . more...
<code>contentTxCodec</code>	string	The codec used to transmit content. If content is being transcoded, it is the output format of the transcoder; either <code>h263+</code> , <code>h264</code> , or <code>automatic</code> (default). This setting does not apply in <code>passthrough</code> mode. more...
<code>contentTxSent</code>	integer	Number of content packets sent.
<code>contentTxReportedLost</code>	integer	Number of content packets reported as lost.
<code>initialAudioMuted</code>	boolean	<code>true</code> if the endpoint's audio is initially muted.
<code>initialVideoMuted</code>	boolean	<code>true</code> if the endpoint's video is initially muted.
<code>autoDisconnect</code>	boolean	<code>true</code> allows the device to automatically disconnect the endpoint, and all remaining endpoints that have this property, when none of the remaining endpoints require manual disconnection. <code>false</code> means this endpoint requires manual disconnection. When a participant disconnects from a conference and only participants who have <code>autoDisconnect</code> set to <code>true</code> remain, the MCU disconnects all the remaining participants.
<code>important</code>	boolean	<code>true</code> means this participant's video is important; it will dominate the layout.
<code>activeSpeaker</code>	boolean	<code>true</code> if the participant is currently the active speaker in the conference.
<code>lecturer</code>	boolean	<code>true</code> if the participant is the lecturer.
<code>layoutControlEx</code>	string	Defines how the view layout can be controlled. One of <code>disabled</code> , <code>feccOnly</code> , <code>dtmfOnly</code> , <code>feccWithDtmfFallback</code> , or <code>bothFeccAndDtmf</code> . more...
<code>activeConferenceId</code>	string	An ID that is unique to each period of activity for a permanent conference. The instance of the conference will retain this ID even if, for example, the conference is renamed while it is active. Each scheduled repeat of the conference has a different <code>activeConferenceId</code> .
<code>activeConferenceId</code> is only present if this participant is currently in an active conference.		
<code>currentLayout</code>	integer	The actual layout in use for the video stream being sent by the MCU to streaming viewers. Refer to Conference layouts [p.188] for details.
<code>currentLayout</code> is not present if the participant is in an auto attendant or if the MCU is not sending video to the participant.		
<code>layoutSource</code>	string	Describes the reason for the current layout, and is only present if <code>currentLayout</code> is present. One of <code>familyx</code> , <code>conferenceCustom</code> , or <code>participantCustom</code> . more...

callDirection	string	Either incoming or outgoing . more...
previewURL	string	The location of the preview image; this is not a complete URL, and requires a prefix of http://hostname (where hostname is the hostname of this MCU) before it is used.
focusType	string	Indicates the endpoint's focus. One of participant , voiceActivated , or h239 . more...

The following parameters identify the participant if **focusType** is **participant**.

focusParticipant	struct	The structure contains participant parameters that identify which participant displays in the largest pane if focusType is participant .
participantName	string	The unique name of a participant. more...
participantProtocol	string	h323 , sip , or vnc .
participantType	string	One of: by_address , by_name , or ad_hoc . more...
conferenceName	string	The name of the conference.

If the participant is in a conference, the response includes the **conferenceName**; if the participant is in an autoattendant, the response includes the **autoAttendantUniqueId** instead. The response will not include both parameters.

autoAttendantUniqueId	string	Unique identifier for the auto attendant.
connectionUniqueId	integer	Corresponds to the uniqueId returned by a conference or autoattendant.
callIdentifier	base64	The base64 encoded GUID (globally unique identifier) of the active H.323 call from this endpoint.
borderWidth	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. more...
suppressAudioDuringDTMF	string	outgoing or all defines which audio the MCU suppresses while it sends the DTMF connection sequence to the endpoint. more...
autoAttendantConfiguredName	string	The name of the auto attendant.

Holds the name of the auto attendant if the participant is connected to an auto attendant; may change as the participant navigates the auto attendant menus.

mediaEncryption	string	One of encrypted , unencrypted , mixed , or unknown . more...
packetLossWarning	boolean	This will be true if any packet loss has been seen within the last 15 seconds.
packetLossCritical	boolean	This will be true if any packet loss above a certain level (5%) is seen within the last five seconds.
videoToUse	struct	Collection of parameters that uniquely identify the participant whose video will display in place of this participant's video by default.
participantName	string	The unique name of a participant. more...

<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>conferenceName</code>	string	The name of the conference.
<code>connectionUniqueId</code>	integer	Corresponds to the <code>uniqueId</code> returned by a conference or autoattendant.

The `configuredState` structure

If the endpoint is not pre-configured, the `configuredState` structure is empty; otherwise it is structured as follows:

Parameter name	Type	Short description
<code>address</code> (<i>endpoint</i>)	string (63)	The address of the endpoint; may be hostname, IP address, E.164 number, SIP URI, or H.323 ID.

The `address` is not returned if it is not known.

<code>gatewayAddress</code>	string (63)	The address of an H.323 gateway, if required. Only used if protocol is <code>h323</code> . This corresponds to the <code>address</code> parameter of the gateway as returned by <code>gateway.enumerate</code> .
<code>useSIPRegistrar</code>	boolean	Not valid unless the protocol is SIP. <code>true</code> if the endpoint uses the SIP registrar. Defaults to <code>false</code> .
<code>transportProtocol</code>	string	Defines the SIP transport protocol. This parameter is ignored if the communication protocol is not SIP. One of <code>default</code> , <code>tcp</code> , <code>udp</code> , or <code>tls</code> .
<code>password</code>	string	The password for VNC endpoints.
<code>deferConnection</code>	boolean	If <code>true</code> , don't call out to this participant immediately, but wait for a <code>participant.connect</code> command. You cannot set <code>deferConnection</code> to true for participants where <code>participantType</code> is <code>ad_hoc</code> .
<code>redial</code>	string	Defines the MCU's redial behavior when calls out to this participant drop. One of <code>never</code> , <code>connect</code> , <code>unexpected</code> , <code>any</code> , or <code>default</code> . more...
<code>redialLimit</code>	string	Defines whether a redial limit is used with the redial behavior. One of <code>enabled</code> , <code>disabled</code> , or <code>default</code> . more...
<code>displayNameOverrideStatus</code>	boolean	<code>true</code> if the endpoint uses the <code>displayNameOverrideValue</code> text to identify itself to other participants.
<code>maxBitRateToMCU</code>	integer	Maximum bandwidth to the MCU (kbps).
<code>maxBitRateFromMCU</code>	integer	Maximum bandwidth from the MCU (kbps).
<code>motionSharpnessTradeoff</code>	string	Defines preference for motion vs. sharpness. One of <code>preferMotion</code> , <code>preferSharpness</code> , <code>balanced</code> , or <code>default</code> . more...
<code>audioRxMuted</code>	boolean	<code>true</code> means that audio from this participant will not be heard by other conference participants.

audioTxMuted	boolean	true means that the MCU is not transmitting the audio part of the conference to this participant.
audioRxGainMode	string	none, automatic, default, or fixed. more...
audioRxGainMillidB	integer	If audio gain mode is fixed, this is the number of decibels of gain applied, multiplied by 1000, and can be a negative value.
videoRxMuted	boolean	true means that video from this participant will not be seen by other conference participants.
videoTxMuted	boolean	true means that the MCU does not send the video part of the conference to this participant.
videoTxWidescreen	boolean	If true , the MCU sends video in a form suitable for a widescreen 16:9 display to this participant.
layoutControlEx	string	Defines how the view layout can be controlled. One of disabled, feccOnly, dtmfOnly, feccWithDtmfFallback, or bothFeccAndDtmf. more...
actAsRecorder	boolean	Defines whether this participant appears as a recorder to other participants.
cpLayout	string	This sets the initial conference view layout for the video sent to the participant. Refer to Conference layouts [p.188] for details.
autoConnect	boolean	true allows endpoints to automatically connect to this conference when they dial in and are recognized. more...
autoDisconnect	boolean	true allows the device to automatically disconnect the endpoint, and all remaining endpoints that have this property, when none of the remaining endpoints require manual disconnection. false means this endpoint requires manual disconnection. When a participant disconnects from a conference and only participants who have autoDisconnect set to true remain, the MCU disconnects all the remaining participants.
borderWidth	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. more...
linkType	string	This parameter is ignored unless participantType is by_address . Either cascadeSlaveToMaster or default
dtmfSequence	string (127)	A string of characters that will be converted to DTMF signals, allowing the device to navigate through audio menus. The sequence may contain 0–9, *, #, and ,. The comma becomes a two second pause. more...
suppressAudioDuringDTMF	string	outgoing or all defines which audio the MCU suppresses while it sends the DTMF connection sequence to the endpoint. more...
suppressDtmfEx	string	Controls the muting of in-band DTMF tones. One of fecc, always, or never. more...
h239Negotiation	string	Defines how the MCU presents itself for h239 token negotiation. One of As master, As slave, or Mimic slave. more...

videoToUse	struct	Collection of parameters that uniquely identify the participant whose video will display in place of this participant's video by default.
participantName	string	The unique name of a participant. more...
participantType	string	One of: by_address , by_name , or ad_hoc . more...
participantProtocol	string	h323 , sip , or vnc .

Deprecated parameters

Parameter name	Type	Short description
callState	string	Deprecated by callStateEx . State of the call between the MCU and this participant. One of dormant , alerting , connected , or disconnected . more...
layoutControlEnabled	boolean	Deprecated by layoutControlEx . Defines whether the endpoint's participant will have control over the layout if layoutControlDefault is false . more...

participant.enumerate (deprecated)

Returns data about participants in conferences on the MCU. Several calls may be required to receive data about all participants; see the notes on `enumerateID` below.

Note: The `participant.enumerate` call now requires the `operationScope` parameter in the call. This topic explains the response of the device when you omit `operationScope`. This use of the call is deprecated and may not be supported in future versions. See [participant.enumerate \[p. 112\]](#).

Input parameters

Optional or conditional inputs

Parameter name	Type	Short description
<code>enumerateID</code>	string	The device returns this index if the requested data is too large for one response. Pass this parameter in a repeat of the call to return the next batch of data. more...
<code>lastRevision</code>	integer	This number identifies an earlier set of enumeration data to compare against your current call. If you supply this parameter using the <code>currentRevision</code> value returned by a previous enumeration, the current <code>enumerate</code> call will return only the differences since that previous call. If you don't supply this parameter, the device assumes that you want a full enumeration.
<code>enumerateFilter</code>	string	A filter expression. The enumeration results depend on the supplied expression.

`enumerateFilter` filters on:

Parameter name	Type	Short description
<code>connected</code>	boolean	<code>true</code> if the participant is currently connected to a conference.
<code>disconnected</code>	boolean	<code>true</code> if the participant has been connected to a conference, but is now disconnected.
<code>connecting</code>	boolean	<code>true</code> if the scheduled participant is in the process of connecting or is pending a retry. <code>connecting</code> is <code>true</code> for participants whose <code>callStateEx</code> values are <code>proceeding</code> , <code>alerting</code> , or <code>pending</code> . It may also be true for some participants whose <code>callState</code> (deprecated) is <code>dormant</code> or <code>disconnected</code> , because these values are also mapped to the new <code>proceeding</code> and <code>pending</code> states, respectively, that were introduced by the persistence feature in MCU 4.4.

Returned data

The response only includes the `participants` array if there are participants to enumerate.

Note: This participant information is returned for all participants added to the conference using the `participant.add` call, even after they have disconnected. However, this information is only returned for other participants (i.e. those added via the web interface or those who dialled into the conference) whilst they are connected but not after they have disconnected.

Parameter name	Type	Short description
<code>currentRevision</code>	integer	A number that indicates the current revision of this enumeration. You can use this as a <code>lastRevision</code> input to a future <code>enumerate</code> call to retrieve only the changes between the two enumerations.
<code>participants</code>	array	An array of structures that represent participants.

If there are participants to enumerate, each corresponding struct in the array may include some or all of the following data:

<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...
<code>conferenceName</code>	string	The name of the conference.

If the participant is in a conference, the response includes the `conferenceName`; if the participant is in an autoattendant, the response includes the `autoAttendantUniqueId` instead. The response will not include both parameters.

<code>autoAttendantUniqueId</code>	string	Unique identifier for the auto attendant.
<code>connectionUniqueId</code>	integer	Corresponds to the <code>uniqueId</code> returned by a conference or autoattendant.
<code>address (endpoint)</code>	string (63)	The address of the endpoint; may be hostname, IP address, <code>sip</code> , E.164 number, SIP URI, or H.323 ID.
<code>gatewayAddress</code>	string (63)	The address of an H.323 gateway, if required. Only used if protocol is <code>h323</code> . This corresponds to the <code>address</code> parameter of the gateway as returned by <code>gateway.enumerate</code> .
<code>deferConnection</code>	boolean	If <code>true</code> , don't call out to this participant immediately, but wait for a <code>participant.connect</code> command. You cannot set <code>deferConnection</code> to true for participants where <code>participantType</code> is <code>ad_hoc</code> .
<code>displayName</code>	string	The display name of the participant. If <code>displayName</code> is longer than 31 characters, only the first 31 characters are returned.
<code>displayNameOverrideStatus</code>	boolean	<code>true</code> if the endpoint uses the <code>displayNameOverrideValue</code> text to identify itself to other participants.
<code>maxBitRateToMCU</code>	integer	Maximum bandwidth to the MCU (kbps).

maxBitRateFromMCU	integer	Maximum bandwidth from the MCU (kbps).
callState	string	Deprecated by callStateEx . State of the call between the MCU and this participant. One of dormant , alerting , connected , or disconnected . more...
connectTime	dateTime. iso8601	Only returned after the participant is connected. This value is always present if the call state is connected. It may or may not be defined for participants in the disconnected state, depending on whether they were ever connected.
disconnectTime	dateTime. iso8601	Only returned after the participant has disconnected.
disconnectReason	string	Only returned after the participant has disconnected; this contains one of the Disconnect reasons [p.185] .
connectPending	boolean	true if sending a "participant.connect" command for this participant will cause either the initial connection to that endpoint (in the event that it was configured with "deferConnection" set) or a re-connection to that endpoint (in the event that it has disconnected).
initialAudioMuted	boolean	true if the endpoint's audio is initially muted.
initialVideoMuted	boolean	true if the endpoint's video is initially muted.
audioRxCodec	string	Receive audio codec.
audioRxLost	integer	Count of the audio packets lost by the MCU.
audioRxReceived	integer	Count of audio packets received by the MCU.
audioTxCodec	string	The codec used on the audio transmission.
audioTxReportedLost	integer	The count of audio packets reported lost by the far end.
audioTxSent	integer	Count of the audio packets sent to this endpoint.
audioRxMuted	boolean	true means that audio from this participant will not be heard by other conference participants.
audioRxGainMode	string	none , automatic , default , or fixed . more...
audioRxGainMillidB	integer	If audio gain mode is fixed, this is the number of decibels of gain applied, multiplied by 1000, and can be a negative value.
videoRxCodec	string	The codec used on the received video.
videoRxLost	integer	Count of video packets lost en route to the MCU from this endpoint.
videoRxReceived	integer	Count of video packets received from this endpoint.

<code>videoTxCodec</code>	string	The codec used on the transmitted video.
<code>videoTxReportedLost</code>	integer	The count of video packets reported lost by the far end.
<code>videoTxSent</code>	integer	Count of the video packets sent to the endpoint.
<code>videoRxMuted</code>	boolean	<code>true</code> means that video from this participant will not be seen by other conference participants.
<code>videoTxWidescreen</code>	boolean	If <code>true</code> , the MCU sends video in a form suitable for a widescreen 16:9 display to this participant.
<code>important</code>	boolean	<code>true</code> means this participant's video is important; it will dominate the layout.
<code>activeSpeaker</code>	boolean	<code>true</code> if the participant is currently the active speaker in the conference.
<code>layoutControlEnabled</code>	boolean	Deprecated by <code>layoutControlEx</code> . Defines whether the endpoint's participant will have control over the layout if <code>layoutControlDefault</code> is <code>false</code> . more...
<code>cpLayout</code>	string	This sets the initial conference view layout for the video sent to the participant. Refer to Conference layouts [p.188] for details.
<code>currentLayout</code>	integer	The actual layout in use for the video stream being sent by the MCU to streaming viewers. Refer to Conference layouts [p.188] for details.
<code>callDirection</code>	string	Either <code>incoming</code> or <code>outgoing</code> . more...

participant.fecc

Controls far end camera. Sends a direction to the identified camera.

Input parameters

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.
<code>autoAttendantUniqueID</code>	string	Unique identifier for the auto attendant.
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...
<code>direction</code>	string	One of <code>up</code> , <code>down</code> , <code>left</code> , <code>right</code> , <code>zoomIn</code> , <code>zoomOut</code> , <code>focusIn</code> , or <code>focusOut</code> .

participant.message

Puts a message on the display of a given participant.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.
<code>autoAttendantUniqueID</code>	string	Unique identifier for the auto attendant.
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...
<code>message</code>	string (255)	The string to send to the participant.

Optional or conditional inputs

Parameter name	Type	Short description
<code>verticalPosition</code>	string	Specifies where to show the message in relation to the screen. The message is always horizontally centred, and is vertically positioned to either <code>top</code> , <code>middle</code> (default), or <code>bottom</code> .
<code>durationSeconds</code>	integer	The period of time, in seconds, for which this item is active.

participant.modify

Depending on the `operationScope` parameter, this call modifies the configuration of a participant (`configuredState`), or the active state of a participant in a conference (`activeState`).

For example, if the parameter `layoutControlEnabled` is included in a call to `participant.modify`, then the effect of the call will depend on the operation scope as follows:

- `operationScope` is `activeState`: the active participant's ability to control their layout will immediately change, but the configured value will remain unchanged, so that if they were to reconnect later, the state of `layoutControlEnabled` would revert back to how it is in the configuration.
- If `operationScope` is `configuredState`, the participant's current ability to control their layout will be unaffected, but their configuration will be changed so that in future occurrences of the conference (or when the participant is reconnected) they will have the newly configured state.

Note: If there is no `operationScope` parameter, the MCU will attempt to change both active and configured states. This is deprecated behavior, and should not be relied upon.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.

If the participant is in a conference, the call requires the `conferenceName`; if the participant is in an autoattendant, the call requires the `autoAttendantUniqueId` instead. The call does not require both parameters.

<code>autoAttendantUniqueId</code>	string	Unique identifier for the auto attendant.
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> or <code>ad_hoc</code> . more...
<code>operationScope</code>	string	Either of the strings <code>activeState</code> or <code>configuredState</code> . more...

Optional or conditional inputs (for either/both states)

You may provide the following parameters, irrespective of the `operationScope`. The call will then attempt to modify the participant's parameters in the state(s) you provide in `operationScope`.

Parameter name	Type	Short description
<code>motionSharpnessTradeoff</code>	string	Defines preference for motion vs. sharpness. One of <code>preferMotion</code> , <code>preferSharpness</code> , <code>balanced</code> , or <code>default</code> . more...
<code>displayNameOverrideStatus</code>	boolean	<code>true</code> if the endpoint uses the <code>displayNameOverrideValue</code> text to identify itself to other participants.

displayNameOverrideValue	string	This value overrides the participant's display name if displayNameOverrideStatus is true .
cpLayout	string	This sets the initial conference view layout for the video sent to the participant. Refer to Conference layouts [p.188] for details.
cameraControl	string	Defines how the endpoint camera(s) within your API call's context can be controlled. If present, it may be disabled , feccOnly , dtmfOnly , feccWithDtmfFallback , bothFeccAndDtmf , or default . more...
audioRxMuted	boolean	true means that audio from this participant will not be heard by other conference participants.
audioTxMuted	boolean	true means that the MCU is not transmitting the audio part of the conference to this participant.

Note: The endpoint may not always detect DTMF tones from the MCU after you mute the outgoing audio.

audioRxGainMode	string	none , automatic , default , or fixed . more...
audioRxGainMillidB	integer	If audio gain mode is fixed, this is the number of decibels of gain applied, multiplied by 1000, and can be a negative value.
videoRxMuted	boolean	true means that video from this participant will not be seen by other conference participants.
videoTxMuted	boolean	true means that the MCU does not send the video part of the conference to this participant.
videoTxWidescreen	boolean	If true , the MCU sends video in a form suitable for a widescreen 16:9 display to this participant.
autoDisconnect	boolean	true allows the device to automatically disconnect the endpoint, and all remaining endpoints that have this property, when none of the remaining endpoints require manual disconnection. false means this endpoint requires manual disconnection. When a participant disconnects from a conference and only participants who have autoDisconnect set to true remain, the MCU disconnects all the remaining participants.
dtmfSequence	string (127)	A string of characters that will be converted to DTMF signals, allowing the device to navigate through audio menus. The sequence may contain 0-9, *, #, and ,. The comma becomes a two second pause. more...
suppressAudioDuringDTMF	string	outgoing or all defines which audio the MCU suppresses while it sends the DTMF connection sequence to the endpoint. more...
videoToUse	struct	Collection of parameters that uniquely identify the participant whose video will display in place of this participant's video by default.

To define or change the participant whose video source will display by default in place of this participant's video, you need to populate the **videoToUse** struct with the following parameters.

Send an empty struct if you want to clear a pre-existing **videoToUse** link.

<code>participantName</code>	string	The unique name of a participant. more...
<code>participantType</code>	string	One of: <code>by_address</code> or <code>ad_hoc</code> . more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .

Optional or conditional inputs (for `activeState` only)

Parameter name	Type	Short description
<code>important</code>	boolean	<code>true</code> means this participant's video is important; it will dominate the layout.
<code>borderWidth</code>	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. more...
<code>focusType</code>	string	Indicates the endpoint's focus. One of <code>participant</code> , <code>voiceActivated</code> , or <code>h239</code> . more...
<code>focusParticipant</code>	struct	The structure contains participant parameters that identify which participant displays in the largest pane if <code>focusType</code> is <code>participant</code> .
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> or <code>ad_hoc</code> . more...
<code>suppressDtmfEx</code>	string	Controls the muting of in-band DTMF tones. One of <code>fecc</code> , <code>always</code> , or <code>never</code> . more...
<code>layoutControlEx</code>	string	Defines how the view layout can be controlled. One of <code>disabled</code> , <code>feccOnly</code> , <code>dtmfOnly</code> , <code>feccWithDtmfFallback</code> , or <code>bothFeccAndDtmf</code> . more...
<code>h239Negotiation</code>	string	Defines how the MCU presents itself for h239 token negotiation. One of <code>As master</code> , <code>As slave</code> , or <code>Mimic slave</code> . more...

Optional or conditional inputs (for `configuredState` only)

You may provide the following parameters to modify the participant's stored configuration (`configuredState`). Do not provide these if you have set `operationScope` to `activeState`.

Parameter name	Type	Short description
<code>address</code> (<i>endpoint</i>)	string (63)	The address of the endpoint; may be hostname, IP address, E.164 number, SIP URI, or H.323 ID.
<code>gatewayAddress</code>	string (63)	The address of an H.323 gateway, if required. Only used if protocol is <code>h323</code> . This corresponds to the <code>address</code> parameter of the gateway as returned by <code>gateway.enumerate</code> .
<code>useSIPRegistrar</code>	boolean	Not valid unless the protocol is SIP. <code>true</code> if the endpoint uses the SIP registrar. Defaults to <code>false</code> .

transportProtocol	string	Defines the SIP transport protocol. This parameter is ignored if the communication protocol is not SIP. One of default , tcp , udp , or tls .
password	string	The password for VNC endpoints.
deferConnection	boolean	If true , don't call out to this participant immediately, but wait for a participant.connect command. You cannot set deferConnection to true for participants where participantType is ad_hoc .
autoConnect	boolean	true allows endpoints to automatically connect to this conference when they dial in and are recognized. more...
linkType	string	This parameter is ignored unless participantType is by_address . Either cascadeSlaveToMaster or default
redial	string	Defines the MCU's redial behavior when calls out to this participant drop. One of never , connect , unexpected , any , or default . more...
redialLimit	string	Defines whether a redial limit is used with the redial behavior. One of enabled , disabled , or default . more...
maxBitRateToMCU	integer	Maximum bandwidth to the MCU (kbps).
maxBitRateFromMCU	integer	Maximum bandwidth from the MCU (kbps).
addAsGuest	boolean	Defines whether the MCU designates guest or chair status to the participant when it invites the participant in to the conference. true means the participant joins as a guest when invited in; false means the participant joins as a chair when invited in.
actAsRecorder	boolean	Defines whether this participant appears as a recorder to other participants.
layoutControlEx	string	Defines how the view layout can be controlled. One of disabled , feccOnly , dtmfOnly , feccWithDtmfFallback , or bothFeccAndDtmf . more...

Deprecated parameters

Parameter name	Type	Short description
layoutControlEnabled	boolean	Deprecated by layoutControlEx . Defines whether the endpoint's participant will have control over the layout if layoutControlDefault is false . more...

participant.move

Moves a participant from one conference to another. This will only move an active participant. Even if this participant is preconfigured, the configuration is unchanged.

A fault code of "no such participant" is returned when the participant isn't found; "too many participants" when the conference has reached its limit, and "operation failed" for other move failures such as moving an unencrypted participant into a conference which requires encryption.

Input parameters

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.
<code>autoAttendantUniqueID</code>	string	Unique identifier for the auto attendant.
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...
<code>newConferenceName</code>	string	The new conference name. more...

participant.remove

Removes a participant from the database of configured participants, and also removes this participant from any conferences. It will also remove all records of this participant's presence in a conference.

Input parameters

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.
<code>autoAttendantUniqueID</code>	string	Unique identifier for the auto attendant.
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...

participant.statistics

Returns statistics relevant to the specified participant.

Note: This call deprecates `participant.diagnostics`. A table at the end of this topic maps the deprecated `participant.diagnostics` parameters to the new parameters detailed below.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...
<code>conferenceName</code>	string	The name of the conference.

Optional or conditional inputs

Parameter name	Type	Short description
<code>filter</code>	struct	A struct that contains boolean switches to filter the statistics. All the switches default to <code>false</code> (do not return these statistics).
<code>audioMedia</code>	boolean	Defaults to <code>false</code> . Set <code>true</code> to return <code>audioMedia</code> statistics.
<code>videoMedia</code>	boolean	Defaults to <code>false</code> . Set <code>true</code> to return <code>videoMedia</code> statistics.
<code>contentMedia</code>	boolean	Defaults to <code>false</code> . Set <code>true</code> to return <code>contentMedia</code> statistics.
<code>audioControl</code>	boolean	Defaults to <code>false</code> . Set <code>true</code> to return <code>audioControl</code> statistics.
<code>videoControl</code>	boolean	Defaults to <code>false</code> . Set <code>true</code> to return <code>videoControl</code> statistics.
<code>contentControl</code>	boolean	Defaults to <code>false</code> . Set <code>true</code> to return <code>contentControl</code> statistics.

Returned data

You will receive only those statistics that you have requested by setting the `filter` parameters. Some statistics do not apply to all media types, and some are only relevant in the receive or transmit sense.

Media statistics

Each of the audio, video, and content media structs contains two nested structs; one each for received and transmitted media. These structs contain subsets of the following statistics, depending on the media type and direction:

Parameter name	Type	Short description
<code>codec</code>	string	The codec in use, or <code>other</code> for undefined codecs.
<code>address (endpoint)</code>	string (63)	The address of the endpoint; may be hostname, IP address, E.164 number, SIP URI, or H.323 ID.
<code>port (IP)</code>	integer	Identifies the IP port.
<code>packetsTransferred</code>	integer	The count of packets transferred in a particular stream. Applies to audio, video, and content streams to and from the device. Deprecates <code>audioRxReceived</code> , <code>videoRxReceived</code> , <code>contentRxReceived</code> , <code>videoTxSent</code> and <code>contentTxSent</code> .
<code>encryption</code>	boolean	Defines whether or not the received or transmitted stream is encrypted. This parameter could apply to content, audio or video streams.
<code>width</code>	integer	The maximum width and height of this stream. Only present for defined video streams
<code>height</code>	integer	The maximum width and height of this stream. Only present for defined video streams
<code>channelBitRate</code>	integer	Bit rate of the channel in bits per second (bps).
<code>selectedBitRate</code>	integer	The selected bit rate for the media stream. Applies to sent and received video and content streams. Deprecates <code>videoRxSelectedBitRate</code> , <code>contentRSelectedBitRate</code> , <code>videoTxSelectedBitRate</code> , and <code>contentTxSelectedBitRate</code> .
<code>actualBitRate</code>	integer	The measured bit rate of this stream, in bits per second (bps).
<code>bitRateLimitReason</code>	string	Provides a reason why the bit rate of a particular stream was limited. Deprecates several more specific parameters , e.g. <code>videoRxBitRateLimitReason</code> .
<code>frameRate</code>	integer	The frame rate of the video or content stream, in frames per second (fps).
<code>codecBitRate</code>	integer	The bit rate required by the codec (bits per second)
<code>jitter</code>	integer	Current jitter in this stream, measured in milliseconds (ms).
<code>jitterBuffer</code>	integer	The jitter buffer shows the current play out delay added to outgoing media to accommodate for packet arrival jitter. Larger values indicate a longer buffer, i.e. more jitter from incoming streams.
<code>energyMillidB</code>	integer	The received audio energy in millidecibels.
<code>packetsErrors</code>	integer	Count of packets lost from a received audio, video, or content stream. Deprecates <code>audioRxLost</code> , <code>videoRxLost</code> and <code>contentRxLost</code> .

frameErrors	integer	Count of frames with errors in this stream.
framesTransferred	integer	Count of audio, video, or content frames received, depending on where the parameter occurs.
temporalSpatial	integer	Integer representing the agreed temporal / spatial trade-off between endpoint and the MCU (motion / sharpness). Value between 0 and 31 (inclusive) where 0 is prefer quality over framerate and 31 is prefer framerate over quality.
contentType	string	The type of content being sent or received.
contentError	string	Information about problems with outgoing content. One of: notAllowed , noCommonCodecs , noCommonFormats , noCommonSymmetricCodecs , modeMismatch , bitRateMismatch , encryptionNotPossible , notPossible .
lipSyncDelayApplied	integer	The amount of delay added to either audio or video output stream to correct for rtcpLipSyncDelay reported between incoming audio and video streams.
rtcpLipSyncDelay	integer	The reported delay between the incoming audio and video streams from this endpoint.
Interlaced	boolean	Defines whether or not the video in this sent or received stream is interlaced. Deprecates videoTxInterlaced and videoRxInterlaced .
fecRecovered	integer	Only returned if FEC (forward error correction) is negotiated and enabled.
fecOverhead	integer	Only returned if FEC (forward error correction) is negotiated and enabled.

Control statistics

Commonly applicable statistics

Parameter name	Type	Short description
rtcpReceiveAddress	string	Address of the RTCP receiver.
rtcpReceivePort	integer	Port number used by the receiver to accept RTCP messages.
rtcpTransmitAddress	string	The IP address and port to which the MCU is sending RTCP packets about this stream.
rtcpTransmitPort	integer	Port number used for transmitting RTCP messages to the endpoint. Absent if rtcpTransmitAddress is unspecified.
rtcpReceiverReports	integer	Count of the RTCP receiver reports seen by the MCU.
rtcpPacketLossReported	integer	The count of media packets reported lost, by the far end, in a receiver report sent to the MCU.
rtcpSenderReports	integer	Count of the RTCP sender reports seen by the MCU.
rtcpOtherReports	integer	Count of the RTCP reports seen by the MCU that are neither sender nor receiver reports.

rtcpPacketsSent	integer	Count of RTCP packets sent by the MCU to this endpoint.
------------------------	---------	---

Video- and content-specific control statistics

Parameter name	Type	Short description
fursSent	integer	Count of fast update requests (FURs) sent by the device (this statistic is only present for video or content control).
fursReceived	integer	Count of fast update requests (FURs) received by the device (this statistic is only present for video or content control).
flowControlReceived	integer	Count of flow control requests received.
flowControlSent	integer	Count of flow control requests sent.

Deprecated parameters

participant.diagnostics parameters	Deprecated by this parameter	Found in these structs
videoRxCodec, contentRxCodec, videoTxCodec, contentTxCodec	codec	All media stats, either direction
	address	All media stats, either direction
	port	All media stats, either direction
audioRxReceived, videoRxReceived, contentRxReceived, videoTxSent, contentTxSent	packetsTransferred	All media stats, either direction
	encryption	All media stats, either direction
videoRxWidth, contentRxWidth, videoTxWidth, contentTxWidth	width	Video and content stats, either direction
videoRxHeight, contentRxHeight, videoTxHeight, contentTxHeight	height	Video and content stats, either direction
videoRxChannelBitRate, contentRxChannelBitRate, videoTxChannelBitRate, contentTxChannelBitRate	channelBitRate	Video and content stats, either direction
videoRxSelectedBitRate, contentRSelectedBitRate, videoTxSelectedBitRate, contentTxSelectedBitRate	selectedBitRate	Video and content stats, either direction
videoRxActualBitRate, contentRxActualBitRate, videoTxActualBitRate, contentTxActualBitRate	actualBitRate	Video and content stats, either direction

participant.diagnostics parameters	Deprecated by this parameter	Found in these structs
videoRxBitRateLimitReason, contentRxBitRateLimitReason, videoTxBitRateLimitReason, contentTxBitRateLimitReason	bitRateLimitReason	Video and content stats, either direction
videoRxFrameRate, contentRxFrameRate, videoTxFrameRate, contentTxFrameRate	frameRate	Video and content stats, either direction
	codecBitRate	Audio stats, either direction
videoRxJitter, contentRxJitter	jitter	All media stats, Receive only
	jitterBuffer	All media stats, Receive only
	energyMillidB	Audio stats, Receive only
audioRxLost, videoRxLost, contentRxLost	packetsErrors	All media stats, Receive only
videoRXFramesReceivedWithErrors, contentRXFramesReceivedWithErrors	frameErrors	All media stats, Receive only
videoRxFramesReceived, contentRxFramesReceived	framesTransferred	All media stats, Receive only
	temporalSpatial	Video and content stats, Transmit only
contentRxType, contentTxType	contentType	Content stats, either direction
contentTxError	contentError	Content stats, Transmit only
	lipSyncDelayApplied	Video stats, Receive only
	rtcpLipSyncDelay	Video stats, Receive only
videoTxInterlaced, videoRxInterlaced	interlaced	Video stats, either direction
	fecRecovered	Video and audio stats, Receive only
	fecOverhead	Video and audio stats, Transmit only
	rtcpReceiveAddress	All control structs
	rtcpReceivePort	All control structs
	rtcpTransmitAddress	All control structs
	rtcpTransmitPort	All control structs
	rtcpReceiverReports	All control structs
audioTxReportedLost, videoTxReportedLost, contentTxReportedLost	rtcpPacketLossReported	All control structs
	rtcpSenderReports	All control structs
	rtcpOtherReports	All control structs
	rtcpPacketsSent	All control structs
	fursSent	Video and content control structs

participant.diagnostics parameters	Deprecated by this parameter	Found in these structs
	fursReceived	Video and content control structs
	flowControlReceived	Video and content control structs
	flowControlSent	Video and content control structs

participant.status

Returns information about an individual participant. This call returns a participant struct as described in [participant.enumerate \[p. 112\]](#), containing information about the participant identified by the call.

A fault code of “no such participant” is returned if the participant does not exist.

Note: The device will respond to `participant.status` if you omit `operationScope`. However, this behavior is deprecated and may not be supported in future versions. See [participant.status \(deprecated\) \[p. 147\]](#) for details.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>conferenceName</code>	string	The name of the conference.
If the participant is in a conference, the call requires the <code>conferenceName</code> ; if the participant is in an autoattendant, the call requires the <code>autoAttendantUniqueId</code> instead. The call does not require both parameters.		
<code>autoAttendantUniqueId</code>	string	Unique identifier for the auto attendant.
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...
<code>operationScope</code>	array	The array should contain one or two string parameters. That is, it should contain either or both of the strings <code>currentState</code> or <code>configuredState</code> . more...

Returned data

Conditionally returned

The response struct may contain the following data:

Parameter name	Type	Short description
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...
<code>conferenceName</code>	string	The name of the conference.

If the participant is in a conference, the response includes the `conferenceName`; if the participant is in an autoattendant, the response includes the `autoAttendantUniqueId` instead. The response will not include both parameters.

<code>autoAttendantUniqueId</code>	string	Unique identifier for the auto attendant.
------------------------------------	--------	---

connectionUniqueId	integer	Corresponds to the <code>uniqueId</code> returned by a conference or autoattendant.
currentState	struct	The current state of the participant. This is only present if requested in the <code>operationScope</code> .

The `currentState` structure may or may not be included in the participant structure, and it may be empty if it is included, depending on the provided value of `operationScope` and whether there is any data to return.

configuredState	struct	The stored configuration of the participant, if it exists. <code>configuredState</code> is only present if requested in the <code>operationScope</code> .
------------------------	--------	---

The `configuredState` structure may or may not be included in the participant structure, and it may be empty if it is included, depending on the provided value of `operationScope` and whether there is any data to return.

The `currentState` structure:

Parameter name	Type	Short description
address (<i>endpoint</i>)	string (63)	The address of the endpoint; may be hostname, IP address, E.164 number, SIP URI, or H.323 ID.
gatewayAddress	string (63)	The address of an H.323 gateway, if required. Only used if protocol is <code>h323</code> . This corresponds to the <code>address</code> parameter of the gateway as returned by <code>gateway.enumerate</code> .
ipAddress	string	IPv4 address in dotted-quad format.
displayName	string	The display name of the participant. If this parameter is longer than 31 characters, only the first 31 characters are returned.
guest	boolean	<code>true</code> if the participant is a guest, <code>false</code> if the participant is a chair.
remoteLinkType	string	One of <code>slave</code> , <code>conference</code> , <code>autoAttendant</code> , <code>recording</code> , or <code>playback</code> .
displayNameOverrideStatus	boolean	<code>true</code> if the endpoint uses the <code>displayNameOverrideValue</code> text to identify itself to other participants.
maxBitRateToMCU	integer	Maximum bandwidth to the MCU (kbps).
maxBitRateFromMCU	integer	Maximum bandwidth from the MCU (kbps).
motionSharpnessTradeoff	string	Defines preference for motion vs. sharpness. One of <code>preferMotion</code> , <code>preferSharpness</code> , <code>balanced</code> , or <code>default</code> . more...
callStateEx	string	One of <code>dormant</code> , <code>proceeding</code> , <code>alerting</code> , <code>connected</code> , <code>pending</code> , or <code>disconnected</code> . more...
connectTime	dateTime. iso8601	Only returned after the participant is connected. This value is always present if the call state is connected. It may or may not be defined for participants in the disconnected state, depending on whether they were ever connected.
disconnectTime	dateTime. iso8601	Only returned after the participant has disconnected.

disconnectReason	string	Only returned after the participant has disconnected; this contains one of the Disconnect reasons [p.185] .
connectPending	boolean	true if sending a "participant.connect" command for this participant will cause either the initial connection to that endpoint (in the event that it was configured with "deferConnection" set) or a re-connection to that endpoint (in the event that it has disconnected).
redial	string	Defines the MCU's redial behavior when calls out to this participant drop. One of never , connect , unexpected , any , or default . more...
redialLimit	string	Defines whether a redial limit is used with the redial behavior. One of enabled , disabled , or default . more...
audioRxCodec	string	Receive audio codec.
audioRxLost	integer	Count of the audio packets lost by the MCU.
audioRxReceived	integer	Count of audio packets received by the MCU.
audioTxCodec	string	The codec used on the audio transmission.
audioTxReportedLost	integer	The count of audio packets reported lost by the far end.
audioTxSent	integer	Count of the audio packets sent to this endpoint.
audioRxMuted	boolean	true means that audio from this participant will not be heard by other conference participants.
audioTxMuted	boolean	true means that the MCU is not transmitting the audio part of the conference to this participant.
audioRxGainMode	string	none , automatic , default , or fixed . more...
audioRxGainMillidB	integer	If audio gain mode is fixed, this is the number of decibels of gain applied, multiplied by 1000, and can be a negative value.
videoRxCodec	string	The codec used on the received video.
videoRxLost	integer	Count of video packets lost en route to the MCU from this endpoint.
videoRxReceived	integer	Count of video packets received from this endpoint.
videoTxCodec	string	The codec used on the transmitted video.
videoTxReportedLost	integer	The count of video packets reported lost by the far end.
videoTxSent	integer	Count of the video packets sent to the endpoint.
videoRxMuted	boolean	true means that video from this participant will not be seen by other conference participants.
videoTxMuted	boolean	true means that the MCU does not send the video part of the conference to this participant.
videoTxWidescreen	boolean	If true , the MCU sends video in a form suitable for a widescreen 16:9 display to this participant.
contentRxType	string	Type of content received. One of none , h239 , or bfcp . more...
contentRxCodec	string	The codec used on the incoming content stream.
contentRxReceived	integer	Number of content packets received from this participant.

<code>contentRxLost</code>	integer	Number of content packets that should have been received from this participant that were not.
<code>contentTxType</code>	string	Type of content transmitted. One of <code>none</code> , <code>h239</code> , <code>bfcp</code> , or <code>mainVideo</code> . more...
<code>contentTxCodec</code>	string	The codec used to transmit content. If content is being transcoded, it is the output format of the transcoder; either <code>h263+</code> , <code>h264</code> , or <code>automatic</code> (default). This setting does not apply in <code>passthrough</code> mode. more...
<code>contentTxSent</code>	integer	Number of content packets sent.
<code>contentTxReportedLost</code>	integer	Number of content packets reported as lost.
<code>initialAudioMuted</code>	boolean	<code>true</code> if the endpoint's audio is initially muted.
<code>initialVideoMuted</code>	boolean	<code>true</code> if the endpoint's video is initially muted.
<code>autoDisconnect</code>	boolean	<code>true</code> allows the device to automatically disconnect the endpoint, and all remaining endpoints that have this property, when none of the remaining endpoints require manual disconnection. <code>false</code> means this endpoint requires manual disconnection. When a participant disconnects from a conference and only participants who have <code>autoDisconnect</code> set to <code>true</code> remain, the MCU disconnects all the remaining participants.
<code>important</code>	boolean	<code>true</code> means this participant's video is important; it will dominate the layout.
<code>activeSpeaker</code>	boolean	<code>true</code> if the participant is currently the active speaker in the conference.
<code>lecturer</code>	boolean	<code>true</code> if the participant is the lecturer.
<code>layoutControlEx</code>	string	Defines how the view layout can be controlled. One of <code>disabled</code> , <code>feccOnly</code> , <code>dtmfOnly</code> , <code>feccWithDtmfFallback</code> , or <code>bothFeccAndDtmf</code> . more...
<code>activeConferenceId</code>	string	An ID that is unique to each period of activity for a permanent conference. The instance of the conference will retain this ID even if, for example, the conference is renamed while it is active. Each scheduled repeat of the conference has a different <code>activeConferenceId</code> .
<code>activeConferenceId</code> is only present if this participant is currently in an active conference.		
<code>currentLayout</code>	integer	The actual layout in use for the video stream being sent by the MCU to streaming viewers. Refer to Conference layouts [p.188] for details.
<code>currentLayout</code> is not present if the participant is in an auto attendant or if the MCU is not sending video to the participant.		
<code>layoutSource</code>	string	Describes the reason for the current layout, and is only present if <code>currentLayout</code> is present. One of <code>familyx</code> , <code>conferenceCustom</code> , or <code>participantCustom</code> . more...
<code>callDirection</code>	string	Either <code>incoming</code> or <code>outgoing</code> . more...

previewURL	string	The location of the preview image; this is not a complete URL, and requires a prefix of <code>http://hostname</code> (where <code>hostname</code> is the hostname of this MCU) before it is used.
focusType	string	Indicates the endpoint's focus. One of <code>participant</code> , <code>voiceActivated</code> , or <code>h239</code> . more...

If **focusType** is `participant`, then a **focusParticipant** struct is included:

focusParticipant	struct	The structure contains participant parameters that identify which participant displays in the largest pane if focusType is <code>participant</code> .
participantName	string	The unique name of a participant. more...
participantProtocol	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
participantType	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...
conferenceName	string	The name of the conference.

If the participant is in a conference, the response includes the **conferenceName**; if the participant is in an autoattendant, the response includes the **autoAttendantUniqueId** instead. The response will not include both parameters.

autoAttendantUniqueId	string	Unique identifier for the auto attendant.
connectionUniqueId	integer	Corresponds to the <code>uniqueId</code> returned by a conference or autoattendant.

callIdentifier	base64	The base64 encoded GUID (globally unique identifier) of the active H.323 call from this endpoint.
borderWidth	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. more...
autoAttendantConfiguredName	string	The name of the auto attendant.

Holds the name of the auto attendant if the participant is connected to an auto attendant; may change as the participant navigates the auto attendant menus.

mediaEncryption	string	One of <code>encrypted</code> , <code>unencrypted</code> , <code>mixed</code> , or <code>unknown</code> . more...
audioRxEnergyMillidB	integer	The measured energy of a participant's audio sent to the MCU. Typically this will be a negative value in the range -30000 (-30dB for very quiet) and 0 (very loud).
audioRxMutedRemotely	boolean	Whether this endpoint is muted remotely.
suppressAudioDuringDTMF	string	<code>outgoing</code> or <code>all</code> defines which audio the MCU suppresses while it sends the DTMF connection sequence to the endpoint. more...
packetLossWarning	boolean	This will be true if any packet loss has been seen within the last 15 seconds.
packetLossCritical	boolean	This will be true if any packet loss above a certain level (5%) is seen within the last five seconds.

videoToUse	struct	Collection of parameters that uniquely identify the participant whose video will display in place of this participant's video by default.
participantName	string	The unique name of a participant. more...
participantType	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...
participantProtocol	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
conferenceName	string	The name of the conference.
connectionUniqueId	integer	Corresponds to the <code>uniqueId</code> returned by a conference or autoattendant.

The `configuredState` structure

If the endpoint is not pre-configured, the `configuredState` structure is empty; otherwise it contains the following entries:

Parameter name	Type	Short description
address (<i>endpoint</i>)	string (63)	The address of the endpoint; may be hostname, IP address, E.164 number, SIP URI, or H.323 ID.

The `address` is not returned if it is not known.

gatewayAddress	string (63)	The address of an H.323 gateway, if required. Only used if protocol is <code>h323</code> . This corresponds to the <code>address</code> parameter of the gateway as returned by <code>gateway.enumerate</code> .
useSIPRegistrar	boolean	Not valid unless the protocol is SIP. <code>true</code> if the endpoint uses the SIP registrar. Defaults to <code>false</code> .
transportProtocol	string	Defines the SIP transport protocol. This parameter is ignored if the communication protocol is not SIP. One of <code>default</code> , <code>tcp</code> , <code>udp</code> , or <code>tls</code> .
password	string	The password for VNC endpoints.
deferConnection	boolean	If <code>true</code> , don't call out to this participant immediately, but wait for a <code>participant.connect</code> command. You cannot set <code>deferConnection</code> to true for participants where <code>participantType</code> is <code>ad_hoc</code> .
redial	string	Defines the MCU's redial behavior when calls out to this participant drop. One of <code>never</code> , <code>connect</code> , <code>unexpected</code> , <code>any</code> , or <code>default</code> . more...
redialLimit	string	Defines whether a redial limit is used with the redial behavior. One of <code>enabled</code> , <code>disabled</code> , or <code>default</code> . more...
displayNameOverrideStatus	boolean	<code>true</code> if the endpoint uses the <code>displayNameOverrideValue</code> text to identify itself to other participants.
maxBitRateToMCU	integer	Maximum bandwidth to the MCU (kbps).
maxBitRateFromMCU	integer	Maximum bandwidth from the MCU (kbps).

motionSharpnessTradeoff	string	Defines preference for motion vs. sharpness. One of preferMotion , preferSharpness , balanced , or default . more...
audioRxMuted	boolean	true means that audio from this participant will not be heard by other conference participants.
audioTxMuted	boolean	true means that the MCU is not transmitting the audio part of the conference to this participant.
audioRxGainMode	string	none , automatic , default , or fixed . more...
audioRxGainMillidB	integer	If audio gain mode is fixed, this is the number of decibels of gain applied, multiplied by 1000, and can be a negative value.
videoRxMuted	boolean	true means that video from this participant will not be seen by other conference participants.
videoTxMuted	boolean	true means that the MCU does not send the video part of the conference to this participant.
videoTxWidescreen	boolean	If true , the MCU sends video in a form suitable for a widescreen 16:9 display to this participant.
layoutControlEx	string	Defines how the view layout can be controlled. One of disabled , feccOnly , dtmfOnly , feccWithDtmfFallback , or bothFeccAndDtmf . more...
actAsRecorder	boolean	Defines whether this participant appears as a recorder to other participants.
cpLayout	string	This sets the initial conference view layout for the video sent to the participant. Refer to Conference layouts [p.188] for details.
autoConnect	boolean	true allows endpoints to automatically connect to this conference when they dial in and are recognized. more...
autoDisconnect	boolean	true allows the device to automatically disconnect the endpoint, and all remaining endpoints that have this property, when none of the remaining endpoints require manual disconnection. false means this endpoint requires manual disconnection. When a participant disconnects from a conference and only participants who have autoDisconnect set to true remain, the MCU disconnects all the remaining participants.
borderWidth	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. more...
linkType	string	This parameter is ignored unless participantType is by_address . Either cascadeSlaveToMaster or default
dtmfSequence	string (127)	A string of characters that will be converted to DTMF signals, allowing the device to navigate through audio menus. The sequence may contain 0-9, *, #, and ,. The comma becomes a two second pause. more...
suppressAudioDuringDTMF	string	outgoing or all defines which audio the MCU suppresses while it sends the DTMF connection sequence to the endpoint. more...

suppressDtmfEx	string	Controls the muting of in-band DTMF tones. One of fecc , always , or never . more...
h239Negotiation	string	Defines how the MCU presents itself for h239 token negotiation. One of As master , As slave , or Mimic slave . more...
videoToUse	struct	Collection of parameters that uniquely identify the participant whose video will display in place of this participant's video by default.
participantName	string	The unique name of a participant. more...
participantType	string	One of: by_address , by_name , or ad_hoc . more...
participantProtocol	string	h323 , sip , or vnc .

Deprecated parameters

Parameter name	Type	Short description
callState	string	Deprecated by callStateEx . State of the call between the MCU and this participant. One of dormant , alerting , connected , or disconnected . more...
layoutControlEnabled	boolean	Deprecated by layoutControlEx . Defines whether the endpoint's participant will have control over the layout if layoutControlDefault is false . more...

participant.status (deprecated)

Note: The `participant.status` call now requires the `operationScope` parameter in the call. This topic explains the response of the device when you omit `operationScope`. This use of the call is deprecated and may not be supported in future versions. See [participant.status \[p.139\]](#).

Returns information about an individual participant. This call returns a participant struct as described in [participant.enumerate \(deprecated\) \[p.121\]](#), containing information about the participant identified by the call.

A fault code of “no such participant” is returned if the participant does not exist.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>conferenceName</code>	string	The name of the conference.

If the participant is in a conference, the call requires the `conferenceName`; if the participant is in an autoattendant, the call requires the `autoAttendantUniqueId` instead. The call does not require both parameters.

<code>autoAttendantUniqueId</code>	string	Unique identifier for the auto attendant.
------------------------------------	--------	---

Returned data

Conditionally returned

The response struct may contain the following data:

Parameter name	Type	Short description
<code>participantName</code>	string	The unique name of a participant. more...
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . more...
<code>conferenceName</code>	string	The name of the conference.

If the participant is in a conference, the response includes the `conferenceName`; if the participant is in an autoattendant, the response includes the `autoAttendantUniqueId` instead. The response will not

include both parameters.

autoAttendantUniqueID	string	Unique identifier for the auto attendant.
connectionUniqueId	integer	Corresponds to the <code>uniqueId</code> returned by a conference or autoattendant.
address (endpoint)	string (63)	The address of the endpoint; may be hostname, IP address, E.164 number, SIP URI, or H.323 ID.
gatewayAddress	string (63)	The address of an H.323 gateway, if required. Only used if protocol is <code>h323</code> . This corresponds to the <code>address</code> parameter of the gateway as returned by <code>gateway.enumerate</code> .
deferConnection	boolean	If <code>true</code> , don't call out to this participant immediately, but wait for a <code>participant.connect</code> command. You cannot set <code>deferConnection</code> to true for participants where <code>participantType</code> is <code>ad_hoc</code> .
displayName	string	The display name of the participant. If <code>displayName</code> is longer than 31 characters, only the first 31 characters are returned.
displayNameOverrideStatus	boolean	<code>true</code> if the endpoint uses the <code>displayNameOverrideValue</code> text to identify itself to other participants.
maxBitRateToMCU	integer	Maximum bandwidth to the MCU (kbps).
maxBitRateFromMCU	integer	Maximum bandwidth from the MCU (kbps).
callState	string	Deprecated by <code>callStateEx</code> . State of the call between the MCU and this participant. One of <code>dormant</code> , <code>alerting</code> , <code>connected</code> , or <code>disconnected</code> . more...
connectTime	dateTime. iso8601	Only returned after the participant is connected. This value is always present if the call state is connected. It may or may not be defined for participants in the disconnected state, depending on whether they were ever connected.
disconnectTime	dateTime. iso8601	Only returned after the participant has disconnected.
disconnectReason	string	Only returned after the participant has disconnected; this contains one of the Disconnect reasons [p.185] .
connectPending	boolean	<code>true</code> if sending a "participant.connect" command for this participant will cause either the initial connection to that endpoint (in the event that it was configured with "deferConnection" set) or a re-connection to that endpoint (in the event that it has disconnected).
audioRxCCodec	string	Receive audio codec.

<code>audioRxLost</code>	integer	Count of the audio packets lost by the MCU.
<code>audioRxReceived</code>	integer	Count of audio packets received by the MCU.
<code>audioTxCodec</code>	string	The codec used on the audio transmission.
<code>audioTxReportedLost</code>	integer	The count of audio packets reported lost by the far end.
<code>audioTxSent</code>	integer	Count of the audio packets sent to this endpoint.
<code>audioRxMuted</code>	boolean	<code>true</code> means that audio from this participant will not be heard by other conference participants.
<code>audioRxGainMode</code>	string	<code>none</code> , <code>automatic</code> , <code>default</code> , or <code>fixed</code> . more...
<code>audioRxGainMillidB</code>	integer	If audio gain mode is fixed, this is the number of decibels of gain applied, multiplied by 1000, and can be a negative value.
<code>videoRxCodec</code>	string	The codec used on the received video.
<code>videoRxLost</code>	integer	Count of video packets lost en route to the MCU from this endpoint.
<code>videoRxReceived</code>	integer	Count of video packets received from this endpoint.
<code>videoTxCodec</code>	string	The codec used on the transmitted video.
<code>videoTxReportedLost</code>	integer	The count of video packets reported lost by the far end.
<code>videoTxSent</code>	integer	Count of the video packets sent to the endpoint.
<code>videoRxMuted</code>	boolean	<code>true</code> means that video from this participant will not be seen by other conference participants.
<code>videoTxWidescreen</code>	boolean	If <code>true</code> , the MCU sends video in a form suitable for a widescreen 16:9 display to this participant.
<code>initialAudioMuted</code>	boolean	<code>true</code> if the endpoint's audio is initially muted.
<code>initialVideoMuted</code>	boolean	<code>true</code> if the endpoint's video is initially muted.
<code>important</code>	boolean	<code>true</code> means this participant's video is important; it will dominate the layout.
<code>activeSpeaker</code>	boolean	<code>true</code> if the participant is currently the active speaker in the conference.
<code>layoutControlEnabled</code>	boolean	Deprecated by <code>layoutControlEx</code> . Defines whether the endpoint's participant will have control over the layout if <code>layoutControlDefault</code> is <code>false</code> . more...
<code>cpLayout</code>	string	This sets the initial conference view layout for the video sent to the participant. Refer to Conference layouts [p.188] for details.

currentLayout	integer	The actual layout in use for the video stream being sent by the MCU to streaming viewers. Refer to Conference layouts [p.188] for details.
callDirection	string	Either incoming or outgoing . more...

route.add

Adds a route, via the gateway specified, to a range of destination IP addresses. Returns a **newRouteId** if successful.

Input parameters

Required inputs

Parameter name	Type	Short description
destination	string	IP address of the route's destination.
prefixLength	integer	The prefix length of the destination IP range for this route (the number of fixed bits in the address).
gateway	string	One of A or B (to use the default gateway configured for that ethernet port), or the IP address of the gateway of this route (must be a valid IP address of the same type as destination).The IP address of the gateway (or next hop) of this route.

Returned data

Conditionally returned

If the call is successful, the response includes the following data:

Parameter name	Type	Short description
status (<i>success</i>)	string	Operation successful
newRouteId	integer	A number selected by the device to identify the newly added route. Pass this parameter as routeId to any calls that require identification of the new route.

route.delete

Deletes the specified route. You can delete manually configured routes but you can not delete automatically configured routes.

Input parameters

Required inputs

Parameter name	Type	Short description
routeId	integer	A number that identifies a route. The device assigns a number to each manually configured route.

route.enumerate

Queries the device for its IP routes. You can filter the response by the type of route - **automatic** or **configured** - but the response includes **both** types by default. The response contains an array of IPv4 routes and an array of IPv6 routes.

Input parameters

Optional or conditional inputs

Parameter name	Type	Short description
filter (<i>route</i>)	string	Filters the returned routes by the route type. One of configured , automatic , or both . Defaults to both .

Returned data

Always returned

Parameter name	Type	Short description
ipv4Routes	array	An array of structs, each of which represents an IPv4 route.
destination	string	IP address of the route's destination.
prefixLength	integer	The prefix length of the destination IP range for this route (the number of fixed bits in the address).
gateway	string	The IP address of the gateway (or next hop) of this route.
port (<i>Ethernet</i>)	string	Identifies the Ethernet port. May be A or B.
type (<i>route</i>)	string	The type of route. One of automatic , configuredByGateway or configuredByPort .
active (<i>route</i>)	boolean	true if the route is currently active. false if the route is inactive (e.g. a route pointing to Port B when port B is disabled). Applies to configured routes only.
routeId	integer	A number that identifies a route. The device assigns a number to each manually configured route.
ipv6Routes	array	An array of structs, each of which represents an IPv6 route (the structs are the same as described above for the IPv4 routes array).

route.preferences.modify

Changes the routing preferences for IPv4 and IPv6 traffic to the specified ethernet interfaces.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>ipv4Preference</code>	string	Either A or B , indicates which Ethernet port is preferred for traffic bound for IPv4 destinations.
<code>ipv6Preference</code>	string	Either A or B , indicates which Ethernet port is preferred for traffic bound for IPv6 destinations.

route.preferences.query

Queries the device's routing preferences for IPv4 and IPv6 traffic.

Returned data

Always returned

Parameter name	Type	Short description
<code>ipv4Preference</code>	string	Either A or B , indicates which Ethernet port is preferred for traffic bound for IPv4 destinations.
<code>ipv6Preference</code>	string	Either A or B , indicates which Ethernet port is preferred for traffic bound for IPv6 destinations.

services.modify

This call modifies the services information as seen on the [Network > Services](#) web page. The call accepts an array named `ports` which must contain at least one struct representing an Ethernet port on the MCU, but may contain two.

Each struct must contain a `port` string to identify the ethernet port (A or B), a `protocol` string (IPv4 or IPv6), and a `services` array whose members represent the web services that you wish to modify on that port and IP protocol.

Exclude from the `services` array any services that you do not wish to modify with this call.

Note: The device returns a success message after successfully parsing the call but before applying the settings.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>ports</code>	array	An array whose members are structures representing the Ethernet ports on the device
<code>port (Ethernet)</code>	string	Identifies the Ethernet port. May be A or B.
<code>protocol (IP)</code>	string	IPv4 or IPv6.
<code>services</code>	array	An array whose members represent the services provided on the particular port and protocol.
<code>name (service)</code>	string	The name of the service. One of the following: TCP services: <code>http</code> , <code>https</code> , <code>ftp</code> , <code>h225</code> , <code>rtsp</code> , <code>mms</code> , <code>sip_tcp</code> , <code>sips_tcp</code> , <code>cdep</code> UDP services: <code>sip_udp</code> , <code>snmp</code> , <code>gatekeeper</code> , <code>tunnel</code> more...
<code>type (service)</code>	string	The type of service. Either <code>tcp</code> or <code>udp</code> .

Optional or conditional inputs

For each service in the `services` array, you may choose to include the following parameters:

Parameter name	Type	Short description
<code>setting</code>	boolean	Defines whether or not this feature is intended to be enabled, irrespective of whether it is actually enabled or requires a feature key.
<code>port (IP)</code>	integer	Identifies the IP port.

The port number is required if `setting` is `true`.

services.query

This call returns the services information as seen on the [Network > Services](#) web page. The response contains an array named `ports` which contains a struct for each Ethernet port on the MCU. Each struct contains a `port` string which identifies the port (A or B), a `protocol` string (IPv4 or IPv6), and an array of structs that contain the details of services provided on that port and protocol.

Returned data

Parameter name	Type	Short description
<code>ports</code>	array	An array whose members are structures representing the Ethernet ports on the device
<code>port (Ethernet)</code>	string	Identifies the Ethernet port. May be A or B.
<code>protocol (IP)</code>	string	IPv4 or IPv6 .
<code>services</code>	array	An array whose members represent the services provided on the particular port and protocol.
<code>setting</code>	boolean	Defines whether or not this feature is intended to be enabled, irrespective of whether it is actually enabled or requires a feature key.
<code>enabled</code>	boolean	true if this feature or item is enabled.
<code>type (service)</code>	string	The type of service. Either tcp or udp .
<code>name (service)</code>	string	The name of the service. One of the following: TCP services: http , https , ftp , h225 , rtsp , mms , sip_tcp , sips_tcp , cdep UDP services: sip_udp , snmp , gatekeeper , tunnel more...
<code>port (IP)</code>	integer	Identifies the IP port.

The port number is only returned if `enabled` is **true**.

sip.modify

Modifies the device's SIP configuration.

A success response to `sip.modify` does not imply that the MCU has successfully registered using the new settings - only that the settings have successfully been modified.

Input parameters

Required inputs

If you set `registrarUsage` to `true`, then you must supply `configuredRegistrar` and `registrarContactURI`.

If you set `registrarType` to `lcs`, then you must supply a fully qualified SIP URI for `registrarContactURI`.

Optional or conditional inputs

Parameter name	Type	Short description
<code>registrarUsage</code>	boolean	Defines whether or not SIP registrar usage is enabled.
<code>configuredRegistrar</code>	string (255)	The SIP domain. Corresponds to SIP registrar domain on the Settings > SIP web page. The parameter contains an empty string value if there is no currently configured SIP domain.
<code>registrarType</code>	string (10)	The type of SIP registrar. Either <code>normal</code> or <code>lcs</code> .
<code>registrarContactURI</code>	string (255)	The URI provided to the SIP registrar to register this device. Corresponds to the Username setting on the Settings > SIP web page.
<code>password (SIP)</code>	string (63)	The password used for SIP registration.
<code>conferenceRegistration</code>	string (8)	Defines whether or not the MCU may register conferences' numeric IDs with the configured SIP registrar. Either <code>enabled</code> or <code>disabled</code> . Corresponds to Allow numeric ID registration for conferences on the Settings > SIP page of the web interface.
<code>configuredProxy</code>	string (255)	The SIP proxy address, either as a DNS hostname or IP address. Corresponds to the SIP proxy address on the Settings > SIP web page. The parameter contains an empty string value if there is no currently configured SIP proxy.
<code>maxOcsBitrate</code>	integer	The bitrate to use for ocs and lcs clients, in bits per second. Accepts <code>64000</code> , <code>128000</code> , <code>192000</code> , <code>256000</code> , <code>320000</code> , <code>384000</code> , <code>512000</code> , <code>768000</code> , <code>1000000</code> , <code>1250000</code> , <code>1500000</code> , <code>1750000</code> , <code>2000000</code> , <code>2500000</code> , <code>3000000</code> , <code>3500000</code> , or <code>4000000</code> . Set this to 0 to disable the limit.
<code>outgoingTransport</code>	string	The outgoing transport protocol. One of <code>udp</code> , <code>tcp</code> , or <code>tls</code> .
<code>useLocalCertificate</code>	boolean	Shows whether or not the MCU has been set to use the local certificate for connections and registrations

sip.query

Retrieves information about SIP configuration on the device.

Returned data

Parameter name	Type	Short description
<code>configuredRegistrar</code>	string (255)	The SIP domain. Corresponds to SIP registrar domain on the Settings > SIP web page. The parameter contains an empty string value if there is no currently configured SIP domain.
<code>configuredProxy</code>	string (255)	The SIP proxy address, either as a DNS hostname or IP address. Corresponds to the SIP proxy address on the Settings > SIP web page. The parameter contains an empty string value if there is no currently configured SIP proxy.
<code>registrarContactURI</code>	string (255)	The URI provided to the SIP registrar to register this device. Corresponds to the Username setting on the Settings > SIP web page.
<code>registrarContactDomain</code>	string	This value is generated from the <code>registrarContactURI</code> (Username in the web interface) and the <code>configuredRegistrar</code> (SIP Registrar domain in web interface.)
<code>conferenceRegistration</code>	string (8)	Defines whether or not the MCU may register conferences' numeric IDs with the configured SIP registrar. Either enabled or disabled . Corresponds to Allow numeric ID registration for conferences on the Settings > SIP page of the web interface.
<code>registrarUsage</code>	boolean	Defines whether or not SIP registrar usage is enabled.
<code>registrarType</code>	string (10)	The type of SIP registrar. Either normal or lcs .
<code>maxOcsBitrate</code>	integer	The bitrate to use for ocs and lcs clients, in bits per second. Accepts 64000, 128000, 192000, 256000, 320000, 384000, 512000, 768000, 1000000, 1250000, 1500000, 1750000, 2000000, 2500000, 3000000, 3500000, or 4000000. Set this to 0 to disable the limit.
<code>outgoingTransport</code>	string	The outgoing transport protocol. One of udp , tcp , or tls .
<code>useLocalCertificate</code>	boolean	Shows whether or not the MCU has been set to use the local certificate for connections and registrations
<code>registrationStatus</code>	string	The SIP registration status. One of registering , registered , unregistered , or unknown .

streaming.modify

Modifies the device's streaming settings. If you set `setting` to `true`, the call will enable streaming and disable conferenceMe.

The call accepts two structs which define streaming formats. It also accepts a struct for each of the IP protocol versions to define the range of multicast addresses.

Notes:

- Multicast is not allowed with the `wmp` format.
- The multicast addresses supplied in a multicast range must all be valid, and must all be of the same IP version.
- The IP version for which multicast is enabled must be active on the device.

Input parameters

Optional or conditional inputs

Parameter name	Type	Short description
<code>setting</code>	boolean	Defines whether or not this feature is intended to be enabled, irrespective of whether it is actually enabled or requires a feature key.
<code>format1</code>	struct	A struct whose contents define a streaming format.
<code>format2</code>	struct	A struct whose contents define a streaming format.
<code>name (endpoint)</code>	string	The name of the endpoint.
<code>format</code>	string	One of <code>wmp</code> , <code>qt64</code> , <code>qt70</code> , or <code>realPlayer</code> . The <code>format</code> determines the <code>audioCodec</code> and <code>videoCodec</code> .
<code>bitRate</code>	integer	The bitrate of this stream in bits/second. This is only present for video streams with a defined codec.
<code>multicast</code>	boolean	Defines whether or not multicast streaming is enabled for this format.
<code>wmpProtocol</code>	string	Describes the behavior of the <code>wmpProtocol</code> when streaming to the endpoint. One of <code>auto</code> , <code>mmsOverUdp</code> , <code>mmsOverTcp</code> , or <code>http</code> .
<code>ipv4MulticastRange</code>	struct	Contains parameters that define an IPv4 multicast range.
<code>ipv6MulticastRange</code>	struct	Contains parameters that define an IPv6 multicast range.
<code>ipRangeStart</code>	string	The first IP address in the multicast range.
<code>ipRangeFinish</code>	string	The last IP address in the multicast range.
<code>portRangeStart</code>	integer	The first port number in the multicast port range.
<code>portRangeFinish</code>	integer	The last port number in the multicast port range.

streaming.query

Queries the device for its streaming settings. The response includes up two structs, `format1` and `format2`, which define streaming formats and a struct each for the defined IPv4 and IPv6 multicast ranges.

Returned data

Always returned

Parameter name	Type	Short description
<code>enabled</code>	boolean	<code>true</code> if this feature or item is enabled.
<code>setting</code>	boolean	Defines whether or not this feature is intended to be enabled, irrespective of whether it is actually enabled or requires a feature key.
<code>format1</code>	struct	A struct whose contents define a streaming format.
<code>format2</code>	struct	A struct whose contents define a streaming format.
<code>name (endpoint)</code>	string	The name of the endpoint.
<code>format</code>	string	One of <code>wmp</code> , <code>qt64</code> , <code>qt70</code> , or <code>realPlayer</code> . The <code>format</code> determines the <code>audioCodec</code> and <code>videoCodec</code> .
<code>bitRate</code>	integer	The bitrate of this stream in bits/second. This is only present for video streams with a defined codec.
<code>audioCodec</code>	string	The codec used on the audio stream. Either <code>RTSP</code> or <code>MMS</code> .
<code>videoCodec</code>	string	The video codec for this streaming connection. Either <code>RTSP</code> or <code>MMS</code> .
<code>multicast</code>	boolean	Defines whether or not multicast streaming is enabled for this format.
<code>wmpProtocol</code>	string	Describes the behavior of the <code>wmpProtocol</code> when streaming to the endpoint. One of <code>auto</code> , <code>mmsOverUdp</code> , <code>mmsOverTcp</code> , or <code>http</code> .
<code>ipv4MulticastRange</code>	struct	Contains parameters that define an IPv4 multicast range.
<code>ipv6MulticastRange</code>	struct	Contains parameters that define an IPv6 multicast range.
<code>ipRangeStart</code>	string	The first IP address in the multicast range.
<code>ipRangeFinish</code>	string	The last IP address in the multicast range.
<code>portRangeStart</code>	integer	The first port number in the multicast port range.
<code>portRangeFinish</code>	integer	The last port number in the multicast port range.

template.create

Creates a new template with the required `templateName` string parameter and returns the `templateNumber` of the new template. If you don't specify the `parent` parameter (takes the `templateName` of the parent template), then the new template will use the top level template as its parent.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>templateName</code>	string	The name of the template. When passed in a call, this parameter identifies the template that is used for the purpose of the call.

Optional or conditional inputs

Parameter name	Type	Short description
<code>parent</code>	string	The name of the parent template. Defaults to Top Level template if omitted.
<code>startLocked</code> (<i>template</i>)	string	Defines whether conferences based on this template should be locked when they start. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).
<code>registerWithGatekeeper</code> (<i>template</i>)	string	Defines whether or not the conferences based on this template register their <code>numericIds</code> with the H.323 gatekeeper. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).
<code>registerWithSIPRegistrar</code> (<i>template</i>)	string	Defines whether conferences based on this template register with the SIP registrar. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).
<code>private</code> (<i>template</i>)	string	Defines whether or not conferences based on this template are private. One of <code>true</code> , <code>false</code> , or <code>default</code> . more...
<code>streaming</code>	string	Specifies the type of streaming to be used on the conference. One of <code>none</code> , <code>unicast</code> , <code>multicast</code> , <code>unicastAndMulticast</code> , or <code>default</code> .
<code>conferenceMeEnabled</code> (<i>template</i>)	string	Whether or not ConferenceMe is enabled for conferences based on this template. <code>true</code> , <code>false</code> , or <code>default</code> (Inherit this setting from the parent template)
<code>contentMode</code>	string	Defines the content mode of the conference. Either <code>disabled</code> , <code>passthrough</code> , <code>transcoded</code> or <code>hybrid</code> . more...
<code>h239Enabled</code>	boolean	Deprecated by <code>contentMode</code> . If you set <code>h239Enabled</code> to <code>true</code> , <code>contentMode</code> will be set to <code>transcoded</code> . If you set <code>h239Enabled</code> to <code>false</code> , <code>contentMode</code> will be set to <code>disabled</code> .
<code>contentContribution</code> (<i>template</i>)	string	Defines whether endpoints are permitted to contribute the content channel to conferences based on this template. One of <code>true</code> , <code>false</code> , or <code>default</code> .

contentTransmitResolutions (<i>template</i>)	string	The resolution for the content channel that will be transmitted to endpoints in conferences based on this template. One of 4to3Only , 16to9Only , allowAll , or default . more...
contentTxCodec (<i>template</i>)	string	The codec used to transmit content in conferences based on this template. If content is to be transcoded, this is the output format of the transcoder; h263+ , h264 , automatic , or default . This setting does not apply in passthrough mode. more...
contentTxMinimumBitRate (<i>template</i>)	string	The minimum bit rate to use for transmitting content, in bps, in conferences based on this template. One of: 0 , 64000 , 128000 , 192000 , 256000 , 320000 , 384000 , 512000 , 768000 , 1000000 , 1250000 , 1500000 , or default (inherit this setting from the parent template).
joinAudioMuted (<i>template</i>)	string	Mutes audio on join. One of true , false , or default to inherit this setting from the parent template.
joinVideoMuted (<i>template</i>)	string	Mutes video on join. One of true , false , or default to inherit this setting from the parent template.
joinAGC (<i>template</i>)	string	Whether AGC should be used by default for participants joining this conference. default if this template inherits the joinAGC setting of its parent template.
layoutControlEx (<i>template</i>)	string	Defines how the view layout can be controlled. One of disabled , feccOnly , dtmfOnly , feccWithDtmfFallback , or bothFeccAndDtmf , or default . more...
cameraControl (<i>template</i>)	string	Defines how the endpoint camera(s) in conferences based on this template can be controlled. If present, it may be disabled , feccOnly , dtmfOnly , feccWithDtmfFallback , bothFeccAndDtmf , or default . more...
dtmfMuteControl (<i>template</i>)	string	Deprecated by inCallMenuControlChair (<i>template</i>) and inCallMenuControlGuest (<i>template</i>). Defines whether or not participants, in conferences based on this template, can mute audio by pressing *6 on the remote control. One of true , false , or default (inherit this setting from the parent template).
encryptionRequired (<i>template</i>)	string	The encryption setting for conferences based on this template, if the encryption feature key is enabled. If true , encryption is required for these conferences. Otherwise, encryption is optional. default causes the template to inherit this setting from its parent template.
suppressDtmfEx (<i>template</i>)	string	Controls the muting of in-band DTMF tones for conferences based on this template. One of fecc , always , never , or default . more...
automaticLectureModeEnabled (<i>template</i>)	string	Defines whether automatic lecture mode is enabled for conferences based on this template. Deprecated by automaticLectureMode (<i>template</i>). more...
automaticLectureMode (<i>template</i>)	string	Defines automatic lecture mode. One of type1 , type2 , disabled , or default . more...

<code>automaticLectureModeTimeout</code>	integer	If <code>automaticLectureMode</code> is <code>type1</code> , this integer defines the period of time for which a speaker must be talking before lecture mode begins. more...
<code>chairControl</code> (<i>template</i>)	string	The chair control setting for conferences based on this template. One of <code>none</code> , <code>floorControlOnly</code> , <code>chairAndFloorControl</code> , or <code>default</code> . more...
<code>lastChairmanLeavesDisconnect</code> (<i>template</i>)	string	Defines whether conferences based on this template disconnect guests when the last chairperson leaves. One of <code>true</code> , <code>false</code> , or <code>default</code> . more...
<code>preconfiguredParticipantsDefer</code> (<i>template</i>)	string	Defines whether conferences based on this template defer inviting preconfigured participants until at least one other participant is present. One of <code>true</code> , <code>false</code> , or <code>default</code> . more...
<code>useMaximumPortsFromParent</code>	boolean	Cannot be set to true for template 0
<code>enforceMaximumVideoPorts</code> (<i>template</i>)	string	Defines whether conferences based on this template will enforce the <code>maximumVideoPorts</code> limit. One of <code>true</code> , <code>false</code> , or <code>default</code> . more...
<code>maximumVideoPorts</code>	integer	The maximum number of video ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
<code>maximumAudioPorts</code>	integer	The maximum number of audio-only ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
<code>enforceMaximumAudioPorts</code> (<i>template</i>)	boolean	Defines whether conferences based on this template will enforce the <code>maximumAudioPorts</code> limit. One of <code>true</code> , <code>false</code> , or <code>default</code> . more...
<code>useReservedPortsFromParent</code>	boolean	Cannot be set to true for template 0
<code>reserveVideoPorts</code>	boolean	Determines whether the template should have a value for the reserved video ports setting. Has no effect if the request sets <code>usePortsFromParent</code> to <code>true</code> .
<code>reservedVideoPorts</code>	integer	The number of video ports to reserve for a conference if in port reservation mode.
<code>reserveAudioPorts</code>	boolean	Determines if the template should have a value for the reserved audio ports setting. Has no effect if the request sets <code>usePortsFromParent</code> to <code>true</code> .
<code>reservedAudioPorts</code>	integer	The number of audio only ports to reserve for a conference if in port reservation mode. more...

Returned data

Always returned

Parameter name	Type	Short description
<code>templateNumber</code>	integer	An index that uniquely identifies the template. Template numbers are not preserved when the MCU reboots. more...
<code>status</code> (<i>success</i>)	string	Operation successful

Deprecated parameters

Parameter name	Type	Short description
<code>dtmfMuteControl</code> (<i>template</i>)	string	Deprecated by <code>inCallMenuControlChair</code> (<i>template</i>) and <code>inCallMenuControlGuest</code> (<i>template</i>). Defines whether or not participants, in conferences based on this template, can mute audio by pressing *6 on the remote control. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).

template.delete

Deletes a template with the provided `templateName` or `templateNumber` parameter. You may only pass one reference.

You can't delete the top level or ad hoc templates. The call will return an error if it can't find the template or if you pass an invalid reference (see [Fault codes \[p.183\]](#)).

Input parameters

Required inputs

The call requires one of the following template identifier parameters.

Parameter name	Type	Short description
<code>templateName</code>	string	The name of the template. When passed in a call, this parameter identifies the template that is used for the purpose of the call.
<code>templateNumber</code>	integer	An index that uniquely identifies the template. Template numbers are not preserved when the MCU reboots. more...

template.enumerate

The `template.enumerate` function returns an array of template structures, each of which contains the settings of a template. The call does not take any parameters.

Parameter name	Type	Short description
<code>templates</code>	array of structs	Each array element is a struct that contains the parameters that define a template.
<code>templateName</code>	string	The name of the template. When passed in a call, this parameter identifies the template that is used for the purpose of the call.
<code>parent</code>	string	The name of the parent template. Defaults to Top Level template if omitted.
<code>adHocDefault</code>	boolean	<code>true</code> means that the MCU uses this template for ad hoc conferences. All templates have this parameter, and it can only be <code>true</code> for one template on the MCU. <code>false</code> for all other templates.
<code>startLocked (template)</code>	string	Defines whether conferences based on this template should be locked when they start. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).
<code>registerWithGatekeeper (template)</code>	string	Defines whether or not the conferences based on this template register their <code>numericIds</code> with the H.323 gatekeeper. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).
<code>registerWithSIPRegistrar (template)</code>	string	Defines whether conferences based on this template register with the SIP registrar. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).
<code>private (template)</code>	string	Defines whether or not conferences based on this template are private. One of <code>true</code> , <code>false</code> , or <code>default</code> . more...
<code>streaming</code>	string	Specifies the type of streaming to be used on the conference. One of <code>none</code> , <code>unicast</code> , <code>multicast</code> , <code>unicastAndMulticast</code> , or <code>default</code> .
<code>conferenceMeEnabled (template)</code>	string	Whether or not ConferenceMe is enabled for conferences based on this template. <code>true</code> , <code>false</code> , or <code>default</code> (Inherit this setting from the parent template)
<code>contentMode</code>	string	Defines the content mode of the conference. Either <code>disabled</code> , <code>passthrough</code> , <code>transcoded</code> or <code>hybrid</code> . more...
<code>h239Enabled</code>	boolean	Deprecated by <code>contentMode</code> . If you set <code>h239Enabled</code> to <code>true</code> , <code>contentMode</code> will be set to <code>transcoded</code> . If you set <code>h239Enabled</code> to <code>false</code> , <code>contentMode</code> will be set to <code>disabled</code> .

<code>contentContribution (template)</code>	string	Defines whether endpoints are permitted to contribute the content channel to conferences based on this template. One of <code>true</code> , <code>false</code> , or <code>default</code> .
<code>contentTransmitResolutions (template)</code>	string	The resolution for the content channel that will be transmitted to endpoints in conferences based on this template. One of <code>4to3Only</code> , <code>16to9Only</code> , <code>allowAll</code> , or <code>default</code> . more...
<code>contentTxCodec (template)</code>	string	The codec used to transmit content in conferences based on this template. If content is to be transcoded, this is the output format of the transcoder; <code>h263+</code> , <code>h264</code> , <code>automatic</code> , or <code>default</code> . This setting does not apply in <code>passthrough</code> mode. more...
<code>contentTxMinimumBitRate (template)</code>	string	The minimum bit rate to use for transmitting content, in bps, in conferences based on this template. One of: 0, 64000, 128000, 192000, 256000, 320000, 384000, 512000, 768000, 1000000, 1250000, 1500000, or <code>default</code> (inherit this setting from the parent template).
<code>joinAudioMuted (template)</code>	string	Mutes audio on join. One of <code>true</code> , <code>false</code> , or <code>default</code> to inherit this setting from the parent template.
<code>joinVideoMuted (template)</code>	string	Mutes video on join. One of <code>true</code> , <code>false</code> , or <code>default</code> to inherit this setting from the parent template.
<code>joinAGC (template)</code>	string	Whether AGC should be used by default for participants joining this conference. <code>default</code> if this template inherits the <code>joinAGC</code> setting of its parent template.
<code>layoutControlEx (template)</code>	string	Defines how the view layout can be controlled. One of <code>disabled</code> , <code>feccOnly</code> , <code>dtmfOnly</code> , <code>feccWithDtmfFallback</code> , or <code>bothFeccAndDtmf</code> , or <code>default</code> . more...
<code>cameraControl (template)</code>	string	Defines how the endpoint camera(s) in conferences based on this template can be controlled. If present, it may be <code>disabled</code> , <code>feccOnly</code> , <code>dtmfOnly</code> , <code>feccWithDtmfFallback</code> , <code>bothFeccAndDtmf</code> , or <code>default</code> . more...
<code>dtmfMuteControl (template)</code>	string	Deprecated by <code>inCallMenuControlChair(template)</code> and <code>inCallMenuControlGuest(template)</code> . Defines whether or not participants, in conferences based on this template, can mute audio by pressing *6 on the remote control. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).
<code>encryptionRequired (template)</code>	string	The encryption setting for conferences based on this template, if the encryption feature key is enabled. If <code>true</code> , encryption is required for these conferences. Otherwise, encryption is optional. <code>default</code> causes the template to inherit this setting from its parent template.

<code>suppressDtmfEx</code> (<i>template</i>)	string	Controls the muting of in-band DTMF tones for conferences based on this template. One of <code>fecc</code> , <code>always</code> , <code>never</code> , or <code>default</code> . more...
<code>automaticLectureModeEnabled</code> (<i>template</i>)	string	Defines whether automatic lecture mode is enabled for conferences based on this template. Deprecated by <code>automaticLectureMode</code> (<i>template</i>). more...
<code>automaticLectureMode</code> (<i>template</i>)	string	Defines automatic lecture mode. One of <code>type1</code> , <code>type2</code> , <code>disabled</code> , or <code>default</code> . more...
<code>automaticLectureModeTimeout</code>	integer	If <code>automaticLectureMode</code> is <code>type1</code> , this integer defines the period of time for which a speaker must be talking before lecture mode begins. more...
<code>chairControl</code> (<i>template</i>)	string	The chair control setting for conferences based on this template. One of <code>none</code> , <code>floorControlOnly</code> , <code>chairAndFloorControl</code> , or <code>default</code> . more...
<code>lastChairmanLeavesDisconnect</code> (<i>template</i>)	string	Defines whether conferences based on this template disconnect guests when the last chairperson leaves. One of <code>true</code> , <code>false</code> , or <code>default</code> . more...
<code>preconfiguredParticipantsDefer</code> (<i>template</i>)	string	Defines whether conferences based on this template defer inviting preconfigured participants until at least one other participant is present. One of <code>true</code> , <code>false</code> , or <code>default</code> . more...
<code>useMaximumPortsFromParent</code>	boolean	Cannot be set to true for template 0
<code>enforceMaximumVideoPorts</code> (<i>template</i>)	string	Defines whether conferences based on this template will enforce the <code>maximumVideoPorts</code> limit. One of <code>true</code> , <code>false</code> , or <code>default</code> . more...
<code>maximumVideoPorts</code>	integer	The maximum number of video ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
<code>maximumAudioPorts</code>	integer	The maximum number of audio-only ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
<code>enforceMaximumAudioPorts</code> (<i>template</i>)	boolean	Defines whether conferences based on this template will enforce the <code>maximumAudioPorts</code> limit. One of <code>true</code> , <code>false</code> , or <code>default</code> . more...
<code>useReservedPortsFromParent</code>	boolean	Cannot be set to true for template 0
<code>reserveVideoPorts</code>	boolean	Determines whether the template should have a value for the reserved video ports setting. Has no effect if the request sets <code>usePortsFromParent</code> to <code>true</code> .
<code>reservedVideoPorts</code>	integer	The number of video ports to reserve for a conference if in port reservation mode.

reserveAudioPorts	boolean	Determines if the template should have a value for the reserved audio ports setting. Has no effect if the request sets <code>usePortsFromParent</code> to <code>true</code> .
reservedAudioPorts	integer	The number of audio only ports to reserve for a conference if in port reservation mode. more...

Deprecated parameters

Parameter name	Type	Short description
dtmfMuteControl (<i>template</i>)	string	Deprecated by <code>inCallMenuControlChair(<i>template</i>)</code> and <code>inCallMenuControlGuest(<i>template</i>)</code> . Defines whether or not participants, in conferences based on this template, can mute audio by pressing *6 on the remote control. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).

template.modify

This call modifies the settings for conference templates. The settings you modify will be applied to any conferences based on the modified template. If you pass the `default` value for a parameter, the template will inherit its parent template's setting for that parameter.

This call returns an error if both `maximumVideoPorts` and `maximumAudioPorts` are set to 0 or if the total number of ports exceeds the maximum conference size (currently 80).

Input parameters

Optional or conditional inputs

Parameter name	Type	Short description
<code>templateNumber</code>	integer	An index that uniquely identifies the template. Template numbers are not preserved when the MCU reboots. more...
<code>newTemplateName</code>	string	Use this parameter to change the name of the template. The call will return an error if another template exists that has this name.
<code>parent</code>	string	The name of the parent template. Defaults to Top Level template if omitted.

You can modify this parameter to move a template within the tree (hierarchy can be seen on the MCU's [Conferences > Templates](#) page). The ad hoc template can be moved this way.

<code>startLocked (template)</code>	string	Defines whether conferences based on this template should be locked when they start. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).
<code>registerWithGatekeeper (template)</code>	string	Defines whether or not the conferences based on this template register their <code>numericIds</code> with the H.323 gatekeeper. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).
<code>registerWithSIPRegistrar (template)</code>	string	Defines whether conferences based on this template register with the SIP registrar. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).
<code>private (template)</code>	string	Defines whether or not conferences based on this template are private. One of <code>true</code> , <code>false</code> , or <code>default</code> . more...
<code>streaming</code>	string	Specifies the type of streaming to be used on the conference. One of <code>none</code> , <code>unicast</code> , <code>multicast</code> , <code>unicastAndMulticast</code> , or <code>default</code> .
<code>conferenceMeEnabled (template)</code>	string	Whether or not ConferenceMe is enabled for conferences based on this template. <code>true</code> , <code>false</code> , or <code>default</code> (Inherit this setting from the parent template)
<code>contentMode</code>	string	Defines the content mode of the conference. Either <code>disabled</code> , <code>passthrough</code> , <code>transcoded</code> or <code>hybrid</code> . more...
<code>h239Enabled</code>	boolean	Deprecated by <code>contentMode</code> . If you set <code>h239Enabled</code> to <code>true</code> , <code>contentMode</code> will be set to <code>transcoded</code> . If you set <code>h239Enabled</code> to <code>false</code> , <code>contentMode</code> will be set to <code>disabled</code> .

<code>contentContribution (template)</code>	string	Defines whether endpoints are permitted to contribute the content channel to conferences based on this template. One of <code>true</code> , <code>false</code> , or <code>default</code> .
<code>contentTransmitResolutions (template)</code>	string	The resolution for the content channel that will be transmitted to endpoints in conferences based on this template. One of <code>4to3Only</code> , <code>16to9Only</code> , <code>allowAll</code> , or <code>default</code> . more...
<code>contentTxCodec (template)</code>	string	The codec used to transmit content in conferences based on this template. If content is to be transcoded, this is the output format of the transcoder; <code>h263+</code> , <code>h264</code> , <code>automatic</code> , or <code>default</code> . This setting does not apply in <code>passthrough</code> mode. more...
<code>contentTxMinimumBitRate (template)</code>	string	The minimum bit rate to use for transmitting content, in bps, in conferences based on this template. One of: <code>0</code> , <code>64000</code> , <code>128000</code> , <code>192000</code> , <code>256000</code> , <code>320000</code> , <code>384000</code> , <code>512000</code> , <code>768000</code> , <code>1000000</code> , <code>1250000</code> , <code>1500000</code> , or <code>default</code> (inherit this setting from the parent template).
<code>joinAudioMuted (template)</code>	string	Mutes audio on join. One of <code>true</code> , <code>false</code> , or <code>default</code> to inherit this setting from the parent template.
<code>joinVideoMuted (template)</code>	string	Mutes video on join. One of <code>true</code> , <code>false</code> , or <code>default</code> to inherit this setting from the parent template.
<code>joinAGC (template)</code>	string	Whether AGC should be used by default for participants joining this conference. <code>default</code> if this template inherits the <code>joinAGC</code> setting of its parent template.
<code>layoutControlEx (template)</code>	string	Defines how the view layout can be controlled. One of <code>disabled</code> , <code>feccOnly</code> , <code>dtmfOnly</code> , <code>feccWithDtmfFallback</code> , or <code>bothFeccAndDtmf</code> , or <code>default</code> . more...
<code>cameraControl (template)</code>	string	Defines how the endpoint camera(s) in conferences based on this template can be controlled. If present, it may be <code>disabled</code> , <code>feccOnly</code> , <code>dtmfOnly</code> , <code>feccWithDtmfFallback</code> , <code>bothFeccAndDtmf</code> , or <code>default</code> . more...
<code>dtmfMuteControl (template)</code>	string	Deprecated by <code>inCallMenuControlChair(template)</code> and <code>inCallMenuControlGuest(template)</code> . Defines whether or not participants, in conferences based on this template, can mute audio by pressing *6 on the remote control. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).
<code>encryptionRequired (template)</code>	string	The encryption setting for conferences based on this template, if the encryption feature key is enabled. If <code>true</code> , encryption is required for these conferences. Otherwise, encryption is optional. <code>default</code> causes the template to inherit this setting from its parent template.
<code>suppressDtmfEx (template)</code>	string	Controls the muting of in-band DTMF tones for conferences based on this template. One of <code>fecc</code> , <code>always</code> , <code>never</code> , or <code>default</code> . more...
<code>automaticLectureModeEnabled (template)</code>	string	Defines whether automatic lecture mode is enabled for conferences based on this template. Deprecated by <code>automaticLectureMode (template)</code> . more...

automaticLectureMode (<i>template</i>)	string	Defines automatic lecture mode. One of type1 , type2 , disabled , or default . more...
automaticLectureModeTimeout	integer	If automaticLectureMode is type1 , this integer defines the period of time for which a speaker must be talking before lecture mode begins. more...
chairControl (<i>template</i>)	string	The chair control setting for conferences based on this template. One of none , floorControlOnly , chairAndFloorControl , or default . more...
lastChairmanLeavesDisconnect (<i>template</i>)	string	Defines whether conferences based on this template disconnect guests when the last chairperson leaves. One of true , false , or default . more...
preconfiguredParticipantsDefer (<i>template</i>)	string	Defines whether conferences based on this template defer inviting preconfigured participants until at least one other participant is present. One of true , false , or default . more...
useMaximumPortsFromParent	boolean	Cannot be set to true for template 0
enforceMaximumVideoPorts (<i>template</i>)	string	Defines whether conferences based on this template will enforce the maximumVideoPorts limit. One of true , false , or default . more...
maximumVideoPorts	integer	The maximum number of video ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
maximumAudioPorts	integer	The maximum number of audio-only ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
enforceMaximumAudioPorts (<i>template</i>)	boolean	Defines whether conferences based on this template will enforce the maximumAudioPorts limit. One of true , false , or default . more...
useReservedPortsFromParent	boolean	Cannot be set to true for template 0
reserveVideoPorts	boolean	Determines whether the template should have a value for the reserved video ports setting. Has no effect if the request sets usePortsFromParent to true .
reservedVideoPorts	integer	The number of video ports to reserve for a conference if in port reservation mode.
reserveAudioPorts	boolean	Determines if the template should have a value for the reserved audio ports setting. Has no effect if the request sets usePortsFromParent to true .
reservedAudioPorts	integer	The number of audio only ports to reserve for a conference if in port reservation mode. more...

Returned data

Always returned

Parameter name	Type	Short description
status (<i>success</i>)	string	Operation successful

Deprecated parameters

Parameter name	Type	Short description
<code>dtmfMuteControl</code> (<i>template</i>)	string	Deprecated by <code>inCallMenuControlChair</code> (<i>template</i>) and <code>inCallMenuControlGuest</code> (<i>template</i>). Defines whether or not participants, in conferences based on this template, can mute audio by pressing *6 on the remote control. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).

template.status

The `template.status` call returns a structure containing all the settings of the selected template.

Input parameters

Required inputs

Parameter name	Type	Short description
<code>templateNumber</code>	integer	An index that uniquely identifies the template. Template numbers are not preserved when the MCU reboots. more...

Returned data

A structure containing the settings of the selected template.

Parameter name	Type	Short description
<code>templateName</code>	string	The name of the template. When passed in a call, this parameter identifies the template that is used for the purpose of the call.
<code>parent</code>	string	The name of the parent template. Defaults to Top Level template if omitted.
<code>adHocDefault</code>	boolean	<code>true</code> means that the MCU uses this template for ad hoc conferences. All templates have this parameter, and it can only be <code>true</code> for one template on the MCU. <code>false</code> for all other templates.
<code>startLocked (template)</code>	string	Defines whether conferences based on this template should be locked when they start. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).
<code>registerWithGatekeeper (template)</code>	string	Defines whether or not the conferences based on this template register their <code>numericIds</code> with the H.323 gatekeeper. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).
<code>registerWithSIPRegistrar (template)</code>	string	Defines whether conferences based on this template register with the SIP registrar. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).
<code>private (template)</code>	string	Defines whether or not conferences based on this template are private. One of <code>true</code> , <code>false</code> , or <code>default</code> . more...
<code>streaming</code>	string	Specifies the type of streaming to be used on the conference. One of <code>none</code> , <code>unicast</code> , <code>multicast</code> , <code>unicastAndMulticast</code> , or <code>default</code> .
<code>conferenceMeEnabled (template)</code>	string	Whether or not ConferenceMe is enabled for conferences based on this template. <code>true</code> , <code>false</code> , or <code>default</code> (Inherit this setting from the parent template)
<code>contentMode</code>	string	Defines the content mode of the conference. Either <code>disabled</code> , <code>passthrough</code> , <code>transcoded</code> or <code>hybrid</code> . more...

h239Enabled	boolean	Deprecated by contentMode . If you set h239Enabled to true , contentMode will be set to transcoded . If you set h239Enabled to false , contentMode will be set to disabled .
contentContribution (<i>template</i>)	string	Defines whether endpoints are permitted to contribute the content channel to conferences based on this template. One of true , false , or default .
contentTransmitResolutions (<i>template</i>)	string	The resolution for the content channel that will be transmitted to endpoints in conferences based on this template. One of 4to3Only , 16to9Only , allowAll , or default . more...
contentTxCodec (<i>template</i>)	string	The codec used to transmit content in conferences based on this template. If content is to be transcoded, this is the output format of the transcoder; h263+ , h264 , automatic , or default . This setting does not apply in passthrough mode. more...
contentTxMinimumBitRate (<i>template</i>)	string	The minimum bit rate to use for transmitting content, in bps, in conferences based on this template. One of: 0, 64000, 128000, 192000, 256000, 320000, 384000, 512000, 768000, 1000000, 1250000, 1500000, or default (inherit this setting from the parent template).
joinAudioMuted (<i>template</i>)	string	Mutes audio on join. One of true , false , or default to inherit this setting from the parent template.
joinVideoMuted (<i>template</i>)	string	Mutes video on join. One of true , false , or default to inherit this setting from the parent template.
joinAGC (<i>template</i>)	string	Whether AGC should be used by default for participants joining this conference. default if this template inherits the joinAGC setting of its parent template.
layoutControlEx (<i>template</i>)	string	Defines how the view layout can be controlled. One of disabled , feccOnly , dtmfOnly , feccWithDtmfFallback , or bothFeccAndDtmf , or default . more...
cameraControl (<i>template</i>)	string	Defines how the endpoint camera(s) in conferences based on this template can be controlled. If present, it may be disabled , feccOnly , dtmfOnly , feccWithDtmfFallback , bothFeccAndDtmf , or default . more...
dtmfMuteControl (<i>template</i>)	string	Deprecated by inCallMenuControlChair (<i>template</i>) and inCallMenuControlGuest (<i>template</i>). Defines whether or not participants, in conferences based on this template, can mute audio by pressing *6 on the remote control. One of true , false , or default (inherit this setting from the parent template).
encryptionRequired (<i>template</i>)	string	The encryption setting for conferences based on this template, if the encryption feature key is enabled. If true , encryption is required for these conferences. Otherwise, encryption is optional. default causes the template to inherit this setting from its parent template.

suppressDtmfEx (<i>template</i>)	string	Controls the muting of in-band DTMF tones for conferences based on this template. One of fecc , always , never , or default . more...
automaticLectureModeEnabled (<i>template</i>)	string	Defines whether automatic lecture mode is enabled for conferences based on this template. Deprecated by automaticLectureMode (<i>template</i>). more...
automaticLectureMode (<i>template</i>)	string	Defines automatic lecture mode. One of type1 , type2 , disabled , or default . more...
automaticLectureModeTimeout	integer	If automaticLectureMode is type1 , this integer defines the period of time for which a speaker must be talking before lecture mode begins. more...
chairControl (<i>template</i>)	string	The chair control setting for conferences based on this template. One of none , floorControlOnly , chairAndFloorControl , or default . more...
lastChairmanLeavesDisconnect (<i>template</i>)	string	Defines whether conferences based on this template disconnect guests when the last chairperson leaves. One of true , false , or default . more...
preconfiguredParticipantsDefer (<i>template</i>)	string	Defines whether conferences based on this template defer inviting preconfigured participants until at least one other participant is present. One of true , false , or default . more...
useMaximumPortsFromParent	boolean	Cannot be set to true for template 0
enforceMaximumVideoPorts (<i>template</i>)	string	Defines whether conferences based on this template will enforce the maximumVideoPorts limit. One of true , false , or default . more...
maximumVideoPorts	integer	The maximum number of video ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
maximumAudioPorts	integer	The maximum number of audio-only ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
enforceMaximumAudioPorts (<i>template</i>)	boolean	Defines whether conferences based on this template will enforce the maximumAudioPorts limit. One of true , false , or default . more...
useReservedPortsFromParent	boolean	Cannot be set to true for template 0
reserveVideoPorts	boolean	Determines whether the template should have a value for the reserved video ports setting. Has no effect if the request sets usePortsFromParent to true .
reservedVideoPorts	integer	The number of video ports to reserve for a conference if in port reservation mode.
reserveAudioPorts	boolean	Determines if the template should have a value for the reserved audio ports setting. Has no effect if the request sets usePortsFromParent to true .
reservedAudioPorts	integer	The number of audio only ports to reserve for a conference if in port reservation mode. more...

Deprecated parameters

Parameter name	Type	Short description
<code>dtmfMuteControl</code> (<i>template</i>)	string	Deprecated by <code>inCallMenuControlChair</code> (<i>template</i>) and <code>inCallMenuControlGuest</code> (<i>template</i>). Defines whether or not participants, in conferences based on this template, can mute audio by pressing *6 on the remote control. One of <code>true</code> , <code>false</code> , or <code>default</code> (inherit this setting from the parent template).

Related information

system.xml file	181
Fault codes	183
Disconnect reasons	185
HTTP keep-alives	187
Conference layouts	188
Linking conferences across MCUs	190

system.xml file

You can derive some information about the MCU from its **system.xml** file. You can download this file via HTTP from the MCU's root.

Example system.xml

```
<?xml version="1.0"?>
  <system>
    <manufacturer>Cisco</manufacturer>
    <model>MCU 5320</model>
    <serial>SM220074</serial>
    <softwareVersion>4.3(1.14)</softwareVersion>
    <buildVersion>6.18(1.14)</buildVersion>
    <hostName></hostName>
    <totalVideoPorts>50</totalVideoPorts>
    <totalAudioOnlyPorts>50</totalAudioOnlyPorts>
    <totalStreamingAndContentPorts>50</totalStreamingAndContentPorts>
    <videoPortAllocation>
      <hd>50</hd>
    </videoPortAllocation>
    <portReservationMode>disabled</portReservationMode>
    <maxVideoResolution>max</maxVideoResolution>
    <uptimeSeconds>109887</uptimeSeconds>
    <clusterType>unclustered</clusterType>
  </system>
```

System XML contents

Node name	Node contents
manufacturer	Name of the manufacturer. May be Cisco on newer hardware or Codian .
model	Model number.
serial	Unique serial number if known, blank otherwise.
softwareVersion	Software version (release reference number).
buildVersion	Software build version (internal reference number).
hostName	DNS name of the MCU if known, blank otherwise.
totalVideoPorts	Count of all video ports.
totalAudioOnlyPorts	Count of all audio-only ports. Only included if clusterType is not slave and the count is greater than 0.
totalStreamingAndContentPorts	Count of all dedicated streaming and content ports, if it is greater than 0. Excluded otherwise.
videoPortAllocation	Excluded if totalVideoPorts is 0. Contains a subnode for each type of video port allocated. These nodes contain the number of ports of that type. e.g. <hd>10</hd> , <nhd>40</nhd> , <sd>40</sd> , or <hdPlus>10</hdPlus> . The subnodes are only included if the count of allocated ports of that type is greater than 0.
portReservationMode	enabled or disabled determines whether the MCU allows conferences to reserve media ports. Only included if clusterType is not slave .

Node name	Node contents
maxVideoResolution	max if the hardware is HD capable or has the 4cif key installed, or cif otherwise. Only included if clusterType is not slave .
uptimeSeconds	Time since the MCU booted.
clusterType	The role of this system in a backplane cluster. May be unclustered , master , or slave . This entry is not included in system.xml if the MCU is incapable of belonging to a cluster.

Fault codes

The Cisco TelePresence MCU returns a fault code when it encounters a problem with processing an XML-RPC request.

The following table lists the fault codes that may be returned by the MCU and their most common interpretations.

Fault Code	Description
1	method not supported. This method is not supported on this device.
2	duplicate conference name. A conference name was specified, but is already in use.
3	duplicate participant name. A participant name was specified, but is already in use.
4	no such conference or auto attendant. The conference or auto attendant identification given does not match any conference or auto attendant.
5	no such participant. The participant identification given does not match any participants.
6	too many conferences. The device has reached the limit of the number of conferences that can be configured.
7	too many participants. There are already too many participants configured and no more can be created.
8	no conference name or auto attendant id supplied. A conference name or auto attendant identifier was required, but was not present.
9	no participant name supplied. A participant name is required but was not present.
10	no participant address supplied. A participant address is required but was not present.
11	invalid start time specified. A conference start time is not valid.
12	invalid end time specified. A conference end time is not valid.
13	invalid PIN specified. A PIN specified is not a valid series of digits.
14	authorization failed. The requested operation is not permitted on this device.
15	insufficient privileges. The specified user id and password combination is not valid for the attempted operation.
16	invalid enumerateID value. An enumerate ID passed to an enumerate method invocation was invalid. Only values returned by the device should be used in enumerate methods.
17	port reservation failure. This is in the case that reservedAudioPorts or reservedVideoPorts value is set too high, and the device cannot support this.
18	duplicate numeric ID. A numeric ID was given, but this ID is already in use.
19	unsupported protocol. A protocol was used which does not correspond to any valid protocol for this method. In particular, this is used for participant identification where an invalid protocol is specified.
20	unsupported participant type. A participant type was used which does not correspond to any participant type known to the device.
25	new port limit lower than currently active
26	floor control not enabled for this conference
27	no such template. The specified template wasn't found.

30	unsupported bit rate. A call tried to set a bit rate that the device does not support.
31	template name in use. This occurs when trying to create or rename a template to have the same name as an existing template.
32	too many templates. This occurs when trying to create a new template after the limit of 100 templates has been reached.
36	required value missing. The call has omitted a value that the MCU requires to make the change requested by the call.
42	port conflict. The call attempts to set a port number that is already in use by another service.
43	route already exists. The call attempts to add a route that has the same destination and prefixLength as a route that already exists on the MCU.
44	route rejected. The call attempts to add a route to a forbidden subnet.
45	too many routes. The call can not add the route because doing so would exceed the allowed number of routes.
46	no such route. The MCU has no record of a route that has the provided routeId .
48	IP address overflows prefix length. The call attempts to make a route destination more specific than the range defined by the prefixLength .
49	operation would disable active interface.
101	missing parameter. This is given when a required parameter is absent. The parameter in question is given in the fault string in the format "missing parameter - parameter_name".
102	invalid parameter. This is given when a parameter was successfully parsed, is of the correct type, but falls outside the valid values; for example an integer is too high or a string value for a protocol contains an invalid protocol. The parameter in question is given in the fault string in the format "invalid parameter - parameter_name".
103	malformed parameter. This is given when a parameter of the correct name is present, but cannot be read for some reason; for example the parameter is supposed to be an integer, but is given as a string. The parameter in question is given in the fault string in the format "malformed parameter - parameter_name".
104	mismatched parameters. The call provides related parameters that, when considered together, are not expected/supported.
201	operation failed. This is a generic fault for when an operation does not succeed as required.

Disconnect reasons

These are the possible values for `disconnectReason`:

Reason	Description
authenticationFailed	VNC authentication failed. Check username and password
busy	The endpoint is in another call
capabilityNegotiationError	Unable to negotiate a common capability set between endpoint and MCU. For example there is no video codec that both sides support
destinationUnreachable	The destination endpoint could not be reached or did not respond
disconnectAll	The MCU disconnected all calls. This occurs at the end of a scheduled conference or a user initiates a disconnect all from the web interface
dnsFailed	A DNS lookup has failed. This can occur when dialling by DNS name
failedToConnectToServer	Unable to connect to VNC server. This can be due to a network problem or if a VNC server is not listening on the specified host
gatekeeperEnded	The gatekeeper ended the call
gatekeeperError	The gatekeeper refused to let the call complete or did not respond
gatekeeperForced	The gatekeeper forced the call to disconnect. For example the end call option was selected on the gatekeeper
gatekeeperRequiredButAbsent	No gatekeeper has been configured but MCU settings require that one be present
h225DecodeError	Error decoding incoming H.225 message. For example the MCU was unable to decode the incoming H.225 message
h225ProtocolError	There has been an H.225 protocol error. For example the endpoint has sent an invalid H.255 message to the MCU
h225SocketError	There has been an error establishing a TCP connection to the H.225 socket on the endpoint. For example there is no route to the desired IP address
h245DecodeError	Error decoding incoming H.245 message. For example the MCU was unable to decode the incoming H.245 message
h245ProtocolError	There has been an H.245 protocol error. For example, the endpoint has sent an invalid H.245 message to the MCU
h245SocketError	There has been an error establishing a TCP connection to the H.245 socket on the endpoint. For example, the endpoint is not listening on the H.245 port it had previously specified
incompatibleVncVersion	VNC version is incompatible with MCU. Check knowledge base for details of supported versions
localGatekeeperRefused	"The local gatekeeper refused the call. This maybe because the destination is not registered to the gatekeeper, for example when dialling direct by IP address"
localTeardown	The MCU disconnected the call
messageQueueOverflow	An excess of information in the message buffer has caused it to run out of space and overflow

Reason	Description
moved	The endpoint has moved to a different conference
networkError	There has been an unspecified network error
noAnswer	The endpoint started ringing but the call was not accepted by the user
noGatekeeperForDN	No gatekeeper has been found for dialed number. This can occur when attempting a call to an invalid E164 number
portAllocationExceeded	The number of available ports (both audio and video) on the MCU has been exceeded
protocolError	There has been an unspecified protocol error
q931DecodeError	Error decoding incoming Q.931 message. For example the MCU was unable to decode the incoming Q.931 message
q931ProtocolError	There has been a Q.931 protocol error. For example the endpoint has sent an invalid Q.931 message to the MCU
rejected	The endpoint chose to reject an incoming call instead of answering
rejectedImmediately	The endpoint rejected the call without ringing
remoteGatekeeperRefused	The remote gatekeeper refused the call. This maybe because the MCU is not registered to the the gatekeeper required by the endpoint
remoteGatekeeperUnreachable	The remote gatekeeper did not respond to the endpoint that the MCU was trying to call
remoteGatewayResources	The remote gateway has insufficient resources to let the call complete. For example the call is being routed to an ISDN gateway with insufficient channels to allow the call to complete
remoteTeardown	The endpoint disconnected the call
serviceUnavailable	The requested service is unavailable. This directly corresponds to an H.323 or SIP message received from the far end to indicate that the call is unable to proceed. The far end could have made this decision for any one of a number of reasons, including lack of resource availability or a call routing policy that prevents the MCU from calling the destination number
timeout	Could not establish call due to network timeout
unspecified	This is a "catch all" reason used when no extra information can be provided
unspecifiedError	This is a "catch all" reason used when no extra information can be provided
videoPortAllocationExceeded	The number of available video ports on the MCU has been exceeded

HTTP keep-alives

Note: This feature is available from API version 2.4 onwards.

Your application can use HTTP keep-alives to reduce the amount of TCP traffic that results from constantly polling the device. Any client which supports HTTP keep-alives may include the following line in the HTTP header of an API request:

Connection: Keep-Alive

This indicates to the device that the client supports HTTP keep-alives. The device may then choose to maintain the TCP connection after it has responded. If the device will close the connection it returns the following HTTP header in its response:

Connection: close

If this line is not in the HTTP header of the response, the client may use the same connection for a subsequent request.

The device will not keep a connection alive if:

- the current connection has already serviced the allowed number of requests
- the current connection has already been open for the allowed amount of time
- the number of open connections exceeds the allowed number if this connection is maintained

These restrictions are in place to limit the resources associated with open connections. If a connection is terminated for either of the first two reasons, the client will probably find that the connection is maintained after the next request.

Note: The client should never assume a connection will be maintained. Also, the device will close an open connection if the client does not make any further requests within a minute. There is little benefit to keeping unused connections open for such long periods.

Conference layouts

Some API calls allow a particular layout to be specified for video sent to that participant via the cpLayout, currentLayout, customLayout parameters. These parameter can take the following values:

- **default**: use the MCU's default view family
- **family<index>**: use the specified layout family
- **layout<index>**: use a specific layout
- **conferenceCustom**: use the conference custom layout

Layout families

The <index> values for family<index> correspond to the following pane arrangements:

Layout families

index	Example layouts
1	
2	
3	
4	
5	

Specific layouts

The <index> values for layout<index> correspond to the following pane arrangements:

Specific layouts

index	Layout	index	Layout	index	Layout	index	Layout
1		16		31		46	
2		17		32		47	
3		18		33		48	

index	Layout	index	Layout	index	Layout	index	Layout
4		19		34		49	
5		20		35		50	
6		21		36		51	
7		22		37		52	
8		23		38		53	
9		24		39		54	
10		25		40		55	
11		26		41		56	
12		27		42		57	
13		28		43		58	
14		29		44		59	
15		30		45			

Linking conferences across MCUs

For the purposes of this description, two conferences are said to be linked if there is a bi-directional H.323 connection between them and each MCU is sending a video channel to the other, showing the active speaker full screen. The audio communicated between the MCUs will be the usual mix of active speakers. For clarification, the linked conferences are given different names ("linked1" and "linked2") in the explanation, but they can have the same name.

The first step is to set up the two conferences. It is important to ensure that the conferences have a numeric id set (the "conferenceId" field in "conference.create"), because, without this configured field, it is not possible to call in directly to a conference. In this example both conferences are given a numeric id, though strictly it is only necessary on the target MCU (i.e. the one that is called rather than the one calling).

In this specific example, "linked1" is set up on "mdu1" and "linked2" set up on "mdu2". The creation of "linked1" is shown in [Example message 1 - creating conference "linked1" on "mdu1" \[p.190\]](#), and it is configured with numeric id "1234"; the creation of "linked2" is shown in [Example message 2 - creating conference "linked2" on "mdu2" \[p.191\]](#), and this conference is given the numeric id "5678".

Next, a participant needs to be added to the "linked1" conference and connected to "linked2" on the target MCU. The most reliable way to accomplish this, which does not rely on the target MCU's gatekeeper usage, is to call from "mdu1" into the target conference using "mdu2" as a gateway and the target conference's numeric id as the remote address. The participant addition is shown in [Example message 3 - calling into "linked2" from "linked1" \[p.192\]](#) - as well as the address and gateway. It also configures the view layout to be full screen (by setting "cpLayout" to "layout1") to make sure that just the active speaker from "linked1" is sent to "linked2".

The final step is slightly more complex — it involves modifying the new "linked2" participant on "mdu2" which was the result of the call from "mdu1". The modification required is to change the view layout setting (for the video sent from "linked2" to "linked1") to full screen so that a view of the "linked2" active speaker is sent.

The complication here is that the "linked2" participant in question is not a participant created via the API, and so the API does not know the name in advance. Therefore, it is necessary to:

- poll membership of "linked2" after the connection from "linked1" has been made
- identify the participant corresponding to the call
- use its name in a "participant.modify" call to set the view layout

The simplest way to identify the participant is to look for an absence of the "address" field in a "conference.query" response: for incoming, non-API, connections this will not be present. [Example message 4 - setting the new "linked2" participant to use a full screen view layout \[p.192\]](#) shows such a "participant.modify" call; in this case the participant name needed was "1_Cisco MCU 4210".

Example message 1 - creating conference "linked1" on "mdu1"

```
<?xml version="1.0"?>
<methodCall>
  <methodName>conference.create</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>authenticationUser</name>
```

```
        <value>
          <string>admin</string>
        </value>
      </member>
    </member>
    <member>
      <name>conferenceName</name>
      <value>
        <string>linked1</string>
      </value>
    </member>
  </struct>
</value>
</param>
</params>
</methodCall>
```

Example message 2 - creating conference "linked2" on "mcu2"

```
<?xml version="1.0"?>
<methodCall>
  <methodName>conference.create</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>authenticationUser</name>
            <value>
              <string>admin</string>
            </value>
          </member>
          <member>
            <name>conferenceName</name>
            <value>
              <string>linked2</string>
            </value>
          </member>
          <member>
            <name>conferenceID</name>
            <value>
              <string>5678</string>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

Example message 3 - calling into "linked2" from "linked1"

```
<?xml version="1.0"?>
<methodCall>
  <methodName>participant.add</methodName>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>authenticationUser</name>
            <value>
              <string>admin</string>
            </value>
          </member>
          <member>
            <name>conferenceName</name>
            <value>
              <string>linked1</string>
            </value>
          </member>
          <member>
            <name>participantName</name>
            <value>
              <string>remote_mcu</string>
            </value>
          </member>
          <member>
            <name>address</name>
            <value>
              <string>5678</string>
            </value>
          </member>
          <member>
            <name>gatewayAddress</name>
            <value>
              <string>10.2.1.27</string>
            </value>
          </member>
          <member>
            <name>cpLayout</name>
            <value>
              <string>layout1</string>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>
```

Example message 4 - setting the new "linked2" participant to use a full screen view layout

```
<?xml version="1.0"?>
<methodCall>
```



```

<methodName>
participant.modify</methodName>
<params>
  <param>
    <value>
      <struct>
        <member>
          <name>authenticationUser</name>
          <value>
            <string>admin</string>
          </value>
        </member>
        <member>
          <name>conferenceName</name>
          <value>
            <string>linked2</string>
          </value>
        </member>
        <member>
          <name>participantName</name>
          <value>
            <string>1_Cisco MCU 4210</string>
          </value>
        </member>
        <member>
          <name>operationScope</name>
          <value>
            <string>active</string>
          </value>
        </member>
        <member>
          <name>cpLayout</name>
          <value>
            <string>layout1</string>
          </value>
        </member>
      </struct>
    </value>
  </param>
</params>
</methodCall>

```

Message responses

The response to each of the above method invocations should be the same normal success indication:

```

<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value>
        <struct>
          <member>
            <name>status</name>
            <value>
              <string>operation successful</string>
            </value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodResponse>

```

```
        </struct>
      </value>
    </param>
  </params>
</methodResponse>
```

Index of parameters

Index of parameters: A	196
Index of parameters: B	205
Index of parameters: C	206
Index of parameters: D	220
Index of parameters: E	224
Index of parameters: F	227
Index of parameters: G	230
Index of parameters: H	232
Index of parameters: I	235
Index of parameters: J	239
Index of parameters: L	240
Index of parameters: M	244
Index of parameters: N	248
Index of parameters: O	251
Index of parameters: P	252
Index of parameters: Q	257
Index of parameters: R	258
Index of parameters: S	265
Index of parameters: T	269
Index of parameters: U	272
Index of parameters: V	274
Index of parameters: W	279

Index of parameters: A

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

aac boolean Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

aac is used in: [addressBookEntry.enumerate \[p.24\]](#).

aac-1c boolean Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

aac_1c is used in: [addressBookEntry.enumerate \[p.24\]](#).

aac-1d boolean Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

aac_1d is used in: [addressBookEntry.enumerate \[p.24\]](#).

actAsRecorder boolean Defines whether this participant appears as a recorder to other participants.

actAsRecorder is used in: [addressBookEntry.enumerate \[p.24\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

activatedFeatures array Each member contains a string named **feature** containing a short description of that feature, for example, **Encryption**.

Feature name	Description
<Product code> activation	Required to activate the product. <i>Product code</i> depends on the type of product for which this key is used, e.g. MSE 8420 activation is the name of the activation key for a Cisco TelePresence MCU MSE 8420 blade.
Video firewall	Required to use Ethernet port B, if present.
4CIF	
Management Application	Required for Conference Director feature.
Web conferencing	Required for ConferenceMe feature.
Encryption	Required for HTTPS, SSL, and TLS.
Gatekeeper I	Required to improve capacity of embedded gatekeeper.
Gatekeeper II	Required to improve capacity of embedded gatekeeper.
6 to 12 port	Required for upgrade of Cisco TelePresence MCU 4501.
1080p capacity upgrade	Required to double the HD+ port count.
Backplane support	Required to enable clustering on Cisco TelePresence MCU MSE 8510 blades.
Full HD mode	Required to enable Full HD mode on MCU 4500 series and MCU MSE 8510 blades.

activatedFeatures is used in: [device.query \[p.87\]](#).

active (<i>route</i>)	boolean	true if the route is currently active. false if the route is inactive (e.g. a route pointing to Port B when port B is disabled). Applies to configured routes only.
active (<i>route</i>) is used in: route.enumerate [p.153] .		
activeConferenceId	string	An ID that is unique to each period of activity for a permanent conference. The instance of the conference will retain this ID even if, for example, the conference is renamed while it is active. Each scheduled repeat of the conference has a different activeConferenceId .
activeConferenceId is used in: conference.enumerate [p.46] , conference.status [p.66] , participant.enumerate [p.112] , participant.status [p.139] .		
activeEndTime	dateTime. iso8601	If the conference is currently active, this field contains the time of the response, to delimits the time span since the start of the current session. This parameter is absent if the conference is permanent.
activeEndTime is used in: conference.enumerate [p.46] , conference.status [p.66] .		
active	boolean	true to request only active conferences.
active is used in: conference.enumerate [p.46] .		
activeRegistrations	integer	The number of active registrations.
activeRegistrations is used in: gatekeeper.query [p.100] .		
activeSpeaker	boolean	true if the participant is currently the active speaker in the conference.
activeSpeaker is used in: participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] , participant.status [p.139] , participant.status (deprecated) [p.147] .		
activeStartTime	dateTime. iso8601	If the conference is currently active, this parameter contains the time that the current session started.
activeStartTime is used in: conference.enumerate [p.46] , conference.status [p.66] .		
actualBitRate	integer	The measured bit rate of this stream, in bits per second (bps).
actualBitRate is used in: participant.statistics [p.133] ,		
addAsGuest	boolean	Defines whether the MCU designates guest or chair status to the participant when it invites the participant in to the conference. true means the participant joins as a guest when invited in; false means the participant joins as a chair when invited in.
addAsGuest is used in: addressBookEntry.enumerate [p.24] , participant.add [p.103] , participant.modify [p.127] .		
addResponse	boolean	true to return the details of the added participant.
addResponse is used in: participant.add [p.103] .		

address (endpoint)	string (63)	The address of the endpoint; may be hostname, IP address, E.164 number, SIP URI, or H.323 ID. address (endpoint) is used in: addressBookEntry.enumerate [p.24] , participant.add [p.103] , participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] , participant.modify [p.127] , participant.statistics [p.133] , participant.status [p.139] , participant.status (deprecated) [p.147] .
address (gatekeeper)	string (255)	The address of the gatekeeper. It may be a DNS hostname or an IP address. address (gatekeeper) is used in: gatekeeper.modify [p.98] , gatekeeper.query [p.100] .
address (gateway)	string (63)	The address of the gateway. address (gateway) is used in: gateway.enumerate [p.102] .
addressBookEntries	array	Each array member is a struct representing a single addressbook entry. addressBookEntries is used in: addressBookEntry.enumerate [p.24] .
adHocDefault	boolean	true means that the MCU uses this template for ad hoc conferences. All templates have this parameter, and it can only be true for one template on the MCU. false for all other templates. adHocDefault is used in: template.create [p.163] , template.enumerate [p.168] , template.modify [p.172] , template.status [p.176] .
alternateGatekeepers	integer	The number of alternate gatekeepers alternateGatekeepers is used in: gatekeeper.query [p.100] .
apiVersion	string	The version number of the API implemented by this device. apiVersion is used in: device.query [p.87] .
audioCodec	string	The codec used on the audio stream. Either RTSP or MMS . audioCodec is used in: streaming.query [p.162] .
audioControl	boolean	Defaults to false . Set true to return audioControl statistics. audioControl is used in: participant.statistics [p.133] .
audioLoad	integer	A percentage value representing the proportion of the device's audio processing capacity that is currently in use. audioLoad is used in: device.health.query [p.81] .
audioMedia	boolean	Defaults to false . Set true to return audioMedia statistics. audioMedia is used in: participant.statistics [p.133] .
audioRTCPOther	integer	The number of other RTCP packets seen for the audio streams.

audioRTCPOther is used in: [conference.streaming.query \[p.72\]](#).

audioRTCPPacketsSent integer The number of RTCP packets sent by the MCU.

audioRTCPPacketsSent is used in: [conference.streaming.query \[p.72\]](#).

audioRTCPReceiverReports integer The number of RTCP receiver reports for the audio streams seen by the MCU.

audioRTCPReceiverReports is used in: [conference.streaming.query \[p.72\]](#).

audioRTCPSenderReports integer The number of RTCP sender reports for the audio streams seen by the MCU.

audioRTCPSenderReports is used in: [conference.streaming.query \[p.72\]](#).

audioRx struct A choice of audio codecs received from the participant's endpoint.

audioRx (*address book entry*) is used in: [addressBookEntry.enumerate \[p.24\]](#).

audioRxCodec string Receive audio codec.

audioRxCodec is used in: [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

audioRxEnergyMillidB integer The measured energy of a participant's audio sent to the MCU. Typically this will be a negative value in the range -30000 (-30dB for very quiet) and 0 (very loud).

audioRxEnergyMillidB is used in: [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

audioRxGainMillidB integer If audio gain mode is fixed, this is the number of decibels of gain applied, multiplied by 1000, and can be a negative value.

audioRxGainMillidB is used in: [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

audioRxGainMode string **none, automatic, default, or fixed.**

Value	Description
none	No extra gain applied
automatic	Automatic gain control applied
fixed	Fixed number of dBs of gain applied
default	The gain mode is inherited from the conference configuration

audioRxGainMode is used in: [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

audioRxLost integer Count of the audio packets lost by the MCU.

audioRxLost is used in: [participant.diagnostics \[p.108\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

audioRxMuted	boolean	true means that audio from this participant will not be heard by other conference participants.
audioRxMuted is used in: participant.add [p.103] , participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] , participant.modify [p.127] , participant.status [p.139] , participant.status (deprecated) [p.147] .		
audioRxMutedRemotely	boolean	Whether this endpoint is muted remotely.
audioRxMutedRemotely is used in: participant.enumerate [p.112] , participant.status [p.139] .		
audioRxReceived	integer	Count of audio packets received by the MCU.
audioRxReceived is used in: participant.diagnostics [p.108] , participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] , participant.status [p.139] , participant.status (deprecated) [p.147] .		
audioStreams	array	An array of stream structs (defined below). These are only present if there are any streams of either type currently in use.
audioStreams is used in: conference.streaming.query [p.72] .		
audioTx	struct	A choice of audio codecs advertised by the MCU.
audioTx (<i>address book entry</i>) is used in: addressBookEntry.enumerate [p.24] .		
audioTxCodec	string	The codec used on the audio transmission.
audioTxCodec is used in: participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] , participant.status [p.139] , participant.status (deprecated) [p.147] .		
audioTxMuted	boolean	true means that the MCU is not transmitting the audio part of the conference to this participant.
audioTxMuted is used in: participant.add [p.103] , participant.enumerate [p.112] , participant.modify [p.127] , participant.status [p.139] .		
audioTxReportedLost	integer	The count of audio packets reported lost by the far end.
audioTxReportedLost is used in: participant.diagnostics [p.108] , participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] , participant.status [p.139] , participant.status (deprecated) [p.147] .		
audioTxSent	integer	Count of the audio packets sent to this endpoint.
audioTxSent is used in: participant.diagnostics [p.108] , participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] , participant.status [p.139] , participant.status (deprecated) [p.147] .		
authenticationPassword	string	The password that corresponds with the given authenticationUser . The API ignores this parameter if the stored user has no password.
authenticationPassword is used in: Authentication [p.8] .		
authenticationUser	string	Name of a user with sufficient privilege for the operation being performed. The name is case sensitive.
authenticationUser is used in: Authentication [p.8] .		
autoAttendantConfiguredName	string	The name of the auto attendant.

autoAttendantConfiguredName is used in: [autoAttendant.enumerate \[p.33\]](#), [autoAttendant.status \[p.34\]](#), [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

autoAttendants array A collection of **autoAttendant** structures.

autoAttendants is used in: [autoAttendant.enumerate \[p.33\]](#).

autoAttendantUniqueID string Unique identifier for the auto attendant.

autoAttendantUniqueID is used in: [autoAttendant.destroy \[p.32\]](#), [autoAttendant.enumerate \[p.33\]](#), [autoAttendant.status \[p.34\]](#), [conference.enumerate \[p.46\]](#), [conference.floor.query \[p.54\]](#), [conference.status \[p.66\]](#), [conference.streaming.query \[p.72\]](#), [participant.add \[p.103\]](#), [participant.connect \[p.107\]](#), [participant.diagnostics \[p.108\]](#), [participant.disconnect \[p.111\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.fecc \[p.125\]](#), [participant.message \[p.126\]](#), [participant.modify \[p.127\]](#), [participant.move \[p.131\]](#), [participant.remove \[p.132\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

autoConnect boolean **true** allows endpoints to automatically connect to this conference when they dial in and are recognized.

If this is true and a participant whose E.164, DNS, or IP address* matches this participant's address dials into the MCU, it will be moved directly to this conference. In order to stop the MCU dialing out to the participant, as the conference starts, use `deferConnection`.

Value	Description
true	When a participant that matches this call's address parameter dials in to the MCU, it is automatically moved to the conference identified by conferenceName in this call.
false	

* Call matching fails on IP address if the participant's autoconnect attempt is routed via a gatekeeper that is in call routing mode. This is because the gatekeeper replaces the IP address of the endpoint with its own IP address.

autoConnect is used in: [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

autoDisconnect boolean **true** allows the device to automatically disconnect the endpoint, and all remaining endpoints that have this property, when none of the remaining endpoints require manual disconnection. **false** means this endpoint requires manual disconnection.

When a participant disconnects from a conference and only participants who have **autoDisconnect** set to **true** remain, the MCU disconnects all the remaining participants.

autoDisconnect is used in: [addressBookEntry.enumerate \[p.24\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

automaticLectureMode string Defines automatic lecture mode. One of **type1**, **type2**, or **disabled**.

Automatic lecture mode shows the speaker full screen. This parameter deprecates **automaticLectureModeEnabled**. If you provide both, only **automaticLectureMode** is used.

Value	Description
type1	The MCU automatically applies lecture mode, if the lecture mode conditions are met, after the period (in seconds) given by <code>automaticLectureModeTimeout</code> . You must provide a value for the timeout integer.
type2	The MCU immediately applies lecture mode when the lecture mode conditions are met. You do not need to provide the <code>automaticLectureModeTimeout</code> parameter; it is always 0 for this automatic lecture mode.
disabled	The MCU never applies lecture mode. You do not need to provide the <code>automaticLectureModeTimeout</code> parameter; the MCU ignores it.

`automaticLectureMode` is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

`automaticLectureMode` (*template*) string Defines automatic lecture mode. One of `type1`, `type2`, `disabled`, or `default`.

Automatic lecture mode shows the speaker full screen. This parameter deprecates `automaticLectureModeEnabled` (*template*). If you provide both, only `automaticLectureMode` (*template*) is used.

Value	Description
type1	The MCU automatically applies lecture mode, if the lecture mode conditions are met, after the period (in seconds) given by <code>automaticLectureModeTimeout</code> . You must provide a value for the timeout integer.
type2	The MCU immediately applies lecture mode when the lecture mode conditions are met. You do not need to provide the <code>automaticLectureModeTimeout</code> parameter; it is always 0 for this automatic lecture mode.
disabled	The MCU never applies lecture mode. You do not need to provide the <code>automaticLectureModeTimeout</code> parameter; the MCU ignores it.
default	Inherit this setting from the parent template.

`automaticLectureMode` (*template*) is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

`automaticLectureModeEnabled` boolean Defines whether automatic lecture mode is enabled for this conference. Deprecated by `automaticLectureMode`.

Note: This parameter is deprecated by `automaticLectureMode`.

Automatic lecture mode shows the speaker full screen.

Value	Description
true	Automatic lecture mode is enabled. The <code>automaticLectureModeTimeout</code> parameter is required.
false	Automatic lecture mode is disabled.

`automaticLectureModeEnabled` is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

<code>automaticLectureModeEnabled</code> (<i>template</i>)	string	Defines whether automatic lecture mode is enabled for conferences based on this template. Deprecated by <code>automaticLectureMode</code> (<i>template</i>).
---	--------	--

Note: This parameter is deprecated by `automaticLectureMode`.

Automatic lecture mode shows the speaker full screen.

Value	Description
true	Automatic lecture mode is enabled. The <code>automaticLectureModeTimeout</code> parameter is required
false	Automatic lecture mode is disabled
default	Inherit this setting from the parent template

`automaticLectureModeEnabled` (*template*) is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

<code>automaticLectureModeTimeout</code>	integer	If <code>automaticLectureMode</code> is <code>type1</code> , this integer defines the period of time for which a speaker must be talking before lecture mode begins.
--	---------	--

Does not apply unless `automaticLectureMode` is `type1`. If `automaticLectureMode` is `type1`, this integer defines the period of time for which a speaker must be talking before lecture mode begins (the speaker is shown full screen).

The parameter has a range of 0 to 60 seconds. A setting of 0 seconds will cause a new speaker to appear in full screen immediately.

`automaticLectureModeTimeout` is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.modify \[p.172\]](#).

<code>availabilityThresholdConferences</code>	string (8) or integer	<p>A threshold beyond which the device will stop indicating resource availability. It is a number between 0 and the maximum number of conferences that can be hosted on the device.</p> <p>You can set this string to a number or <code>all</code> in a <code>gatekeeper.modify</code> call.</p> <p>This threshold value is returned as an integer by <code>gatekeeper.query</code>. It is not returned if it has been set to <code>all</code>. It is not returned if the MCU is not configured to send resource availability indications.</p>
---	-----------------------------	--

availabilityThresholdConferences is used in: [gatekeeper.modify \[p.98\]](#), [gatekeeper.query \[p.100\]](#).

availabilityThresholdVideoPorts	string (8) or integer	A threshold beyond which the device stops indicating resource availability. It is a number between 0 and the maximum number of video ports available on the device. You can set this string to a number or a11 in a gatekeeper.modify call. This threshold value is returned as an integer by gatekeeper.query . It is not returned if it has been set to a11 . It is not returned if the MCU is not configured to send resource availability indications.
--	-----------------------------	--

availabilityThresholdVideoPorts is used in: [gatekeeper.modify \[p.98\]](#), [gatekeeper.query \[p.100\]](#).

Index of parameters: B

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

bitRate integer The bitrate of this stream in bits/second. This is only present for video streams with a defined codec.

bitRate is used in: [conference.streaming.query \[p.72\]](#), [streaming.modify \[p.161\]](#), [streaming.query \[p.162\]](#).

bitRateLimitReason string Provides a reason why the bit rate of a particular stream was limited. Deprecates several more specific parameters , e.g. **videoRxBitRateLimitReason**.

bitRateLimitReason is used in: [participant.statistics \[p.133\]](#).

borderWidth integer Controls the width of the outer border of a preconfigured participant's layout. 0 is no border.

Value	Description
0	No border
1	Corresponds to border +1 on the web interface
2	Corresponds to border +2 on the web interface
3	Corresponds to border +3 on the web interface

borderWidth is used in: [addressBookEntry.enumerate \[p.24\]](#), [conference.streaming.modify \[p.71\]](#), [conference.streaming.query \[p.72\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

buildVersion string The build version of the software running on the device.

buildVersion is used in: [device.query \[p.87\]](#).

bytesReceived integer The number of bytes received by the device.

bytesReceived is used in: [device.network.query \[p.84\]](#).

bytesReceived64 string 64 bit versions of the **bytesReceived** statistic, using a string rather than an integer.

bytesReceived64 is used in: [device.network.query \[p.84\]](#).

bytesSent integer The number of bytes sent by the device.

bytesSent is used in: [device.network.query \[p.84\]](#).

bytesSent64 string 64 bit versions of the **bytesSent** statistic, using a string rather than an integer.

bytesSent64 is used in: [device.network.query \[p.84\]](#).

Index of parameters: C

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

callDirection string Either **incoming** OR **outgoing**.

This parameter is not present if **callState** is **dormant**.

Value	Description
incoming	The participant called in to the MCU
outgoing	The MCU called out to the participant

callDirection is used in: [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

callIdentifier base64 The base64 encoded GUID (globally unique identifier) of the active H.323 call from this endpoint.

callIdentifier is used in: [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

callInParams struct A structure containing the call in parameters of the endpoint. These parameters are used to match incoming calls to pre-configured participants. For a positive match, a participant must match fields which have values. Blank fields are not considered in the comparison.

callInParams is used in: [addressBookEntry.enumerate \[p.24\]](#).

callState string Deprecated by **callStateEx**. State of the call between the MCU and this participant. One of **dormant**, **alerting**, **connected**, or **disconnected**.

Value	Description
dormant	There is currently no attempt to connect a call.
alerting	The call is connecting and a reply has been received.
connected	Call has been set up successfully.
disconnected	The call has ended or the connection has failed. A further connection attempt may or may not occur.

callState is used in: [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

callStateEx string One of **dormant**, **proceeding**, **alerting**, **connected**, **pending**, or **disconnected**.

Value	Description
dormant	There is currently no attempt to connect a call.

Value	Description
proceeding	The call is connecting but no reply has yet been received.
alerting	The call is connecting and a reply has been received.
connected	Call has been set up successfully.
pending	The connection has failed and another connection attempt will occur.
disconnected	The call has ended or the connection has failed. No more connection attempts will occur.

`callStateEx` is used in: [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

<code>cameraControl</code> (<i>template</i>)	string	Defines how the endpoint camera(s) in conferences based on this template can be controlled. If present, it may be disabled , feccOnly , dtmfOnly , feccWithDtmfFallback , bothFeccAndDtmf , or default .
--	--------	--

In calls to the MCU this parameter defines how the endpoint camera(s) within the call's context can be controlled.

In responses from the MCU the parameter may be absent if it is not explicitly configured; that is, if `cameraControlDefault` is **true** in the context of the response.

If `cameraControlDefault` is **false**, `cameraControl` can be:

Value	Description
disabled	Camera control is disabled
feccOnly	Camera control via FECC only
dtmfOnly	Camera control via DTMF only
feccWithDtmfFallback	Camera control via FECC when it is available and via DTMF for endpoints which don't have FECC
bothFeccAndDtmf	Camera control via FECC and via DTMF
default	Inherit this setting from the parent template

`cameraControl` (*template*) is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

<code>cameraControl</code>	string	Defines how the endpoint camera(s) within your API call's context can be controlled. If present, it may be disabled , feccOnly , dtmfOnly , feccWithDtmfFallback , bothFeccAndDtmf , or default .
----------------------------	--------	---

In calls to the MCU this parameter defines how the endpoint camera(s) within the call's context can be controlled.

In responses from the MCU the parameter may be absent if it is not explicitly configured; that is, if `cameraControlDefault` is **true** in the context of the response.

If `cameraControlDefault` is **false**, `cameraControl` can be:

Value	Description
disabled	Camera control is disabled
default	Inherit camera control setting
feccOnly	Camera control via FECC only
dtmfOnly	Camera control via DTMF only
feccWithDtmfFallback	Camera control via FECC when it is available and via DTMF for endpoints which don't have FECC
bothFeccAndDtmf	Camera control via FECC and via DTMF

`cameraControl` is used in: [addressBookEntry.enumerate \[p.24\]](#), [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#), [participant.modify \[p.127\]](#).

<code>cameraControlDefault</code>	boolean	<code>true</code> means the endpoint uses the default camera control setting of the conference or template. <code>false</code> means the endpoint explicitly sends another type of camera control to participants.
-----------------------------------	---------	--

`cameraControlDefault` is used in: [addressBookEntry.enumerate \[p.24\]](#).

<code>chairControl (template)</code>	string	The chair control setting for conferences based on this template. One of <code>none</code> , <code>floorControlOnly</code> , <code>chairAndFloorControl</code> , or <code>default</code> .
--------------------------------------	--------	--

This setting corresponds to the "Floor and chair control" setting on the web interface. If this parameter is not specified, the chair control setting defaults to *Allow floor control only*.

Value	Description
none	Do not allow floor or chair control
floorControlOnly	Allow floor control only (default value)
chairAndFloorControl	Allow floor and chair control
default	Inherit this setting from the parent template

`chairControl (template)` is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

<code>chairControl</code>	string	The chair control setting for this conference. One of <code>none</code> , <code>floorControlOnly</code> , or <code>chairAndFloorControl</code> .
---------------------------	--------	--

This setting corresponds to the **Floor and chair control** setting on the web interface. If this parameter is not specified, the chair control setting defaults to *Allow floor control only*.

Value	Description
none	Do not allow floor or chair control
floorControlOnly	Allow floor control only (default value)
chairAndFloorControl	Allow floor and chair control

chairControl is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

chairParticipant struct A structure containing parameters that uniquely identify the participant who is the chairperson.

chairParticipant is used in: [conference.enumerate \[p.46\]](#), [conference.floor.modify \[p.53\]](#), [conference.status \[p.66\]](#).

channelBitRate integer Bit rate of the channel in bits per second (bps).

channelBitRate is used in: [participant.statistics \[p.133\]](#),

cleanupTimeout integer Allows the MCU to automatically delete a conference which has ended or been empty for this number of seconds.

If the conference has an end time, the timeout will only start after the end time, even if it is empty before that time.

Permanent conferences will be deleted when they become empty and remain empty for the timeout ('empty' excludes recorders, streamers and slave to master links).

Scheduled conferences won't be deleted before their scheduled start time.

Value	Description
0	Disable automatic deletion
n (positive integer)	Allow automatic deletion of a conference, n seconds after it ends or becomes and remains empty

cleanupTimeout is used in: [conference.create \[p.39\]](#).

clusterType string The role that this MCU plays in a cluster. One of **master**, **slave**, or **unclustered**. The parameter is absent if the device is incapable of belonging to a cluster.

clusterType is used in: [device.query \[p.87\]](#).

codec string The codec in use, or **other** for undefined codecs.

codec is used in: [conference.streaming.query \[p.72\]](#), [participant.statistics \[p.133\]](#),

codecBitRate integer The bit rate required by the codec (bits per second)

codecBitRate is used in: [participant.statistics \[p.133\]](#).

collisions integer Count of the network collisions recorded by the device.

collisions is used in: [device.network.query \[p.84\]](#).

completed boolean True if the conference has finished.

completed is used in: [conference.enumerate \[p.46\]](#).

<code>conferenceActive</code>	boolean	Indicates whether conference is currently active. <code>true</code> if the conference is currently active. <code>false</code> if the conference is currently inactive. Permanent conferences are always active; completed conferences, or those that have not yet started, are inactive. <code>conferenceActive</code> is used in: conference.enumerate [p.46] , conference.status [p.66] .
<code>conferenceID</code>	string	Deprecated by <code>numericId</code> . <code>conferenceID</code> is used in: conference.create [p.39] , conference.modify [p.57] .
<code>conferenceMeEnabled</code> (<i>template</i>)	string	Whether or not ConferenceMe is enabled for conferences based on this template. <code>true</code> , <code>false</code> , or <code>default</code> (Inherit this setting from the parent template) <code>conferenceMeEnabled</code> (<i>template</i>) is used in: template.modify [p.172] , template.enumerate [p.168] , template.create [p.163] , template.status [p.176] .
<code>conferenceMeEnabled</code>	boolean	Whether or not ConferenceMe is enabled for this conference. <code>conferenceMeEnabled</code> is used in: conference.create [p.39] , conference.enumerate [p.46] , conference.modify [p.57] , conference.status [p.66] .
<code>conferenceName</code>	string	The name of the conference. <code>conferenceName</code> is used in: conference.create [p.39] , conference.destroy [p.44] , conference.end [p.45] , conference.enumerate [p.46] , conference.floor.modify [p.53] , conference.floor.query [p.54] , conference.metadata.modify [p.55] , conference.metadata.status [p.56] , conference.modify [p.57] , conference.paneplacement.modify [p.61] , conference.paneplacement.query [p.63] , conference.resetCleanupTimeout [p.65] , conference.status [p.66] , conference.streaming.modify [p.71] , conference.streaming.query [p.72] , participant.add [p.103] , participant.connect [p.107] , participant.diagnostics [p.108] , participant.disconnect [p.111] , participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] , participant.fecc [p.125] , participant.message [p.126] , participant.modify [p.127] , participant.move [p.131] , participant.remove [p.132] , participant.statistics [p.133] , participant.status [p.139] , participant.status (deprecated) [p.147] .
<code>conferenceRegistration</code>	string (8)	Defines whether or not the MCU may register conferences' numeric IDs with the configured SIP registrar. Either <code>enabled</code> or <code>disabled</code> . Corresponds to Allow numeric ID registration for conferences on the Settings > SIP page of the web interface. <code>conferenceRegistration</code> is used in: sip.modify [p.159] , sip.query [p.160] .
<code>conferences</code>	array	An array of structs, each of which contains all the returned information about a single conference. <code>conferences</code> is used in: conference.enumerate [p.46] ,
<code>conferenceType</code>	string	Indicates whether a conference is or was <code>scheduled</code> , or <code>ad_hoc</code> (which means it was started without being scheduled). <code>conferenceType</code> is used in: conference.enumerate [p.46] , conference.status [p.66] .

conferencingParameters	struct	A structure containing the conferencing parameters of the enumerated item, e.g. gateway or endpoint. conferencingParameters is used in: addressBookEntry.enumerate [p.24] , gateway.enumerate [p.102] .
configuredProxy	string (255)	The SIP proxy address, either as a DNS hostname or IP address. Corresponds to the SIP proxy address on the Settings > SIP web page. The parameter contains an empty string value if there is no currently configured SIP proxy. configuredProxy is used in: sip.modify [p.159] , sip.query [p.160] .
configuredRegistrar	string (255)	The SIP domain. Corresponds to SIP registrar domain on the Settings > SIP web page. The parameter contains an empty string value if there is no currently configured SIP domain. configuredRegistrar is used in: sip.modify [p.159] , sip.query [p.160] .
configuredState	struct	The stored configuration of the participant, if it exists. configuredState is only present if requested in the operationScope . configuredState is used in: participant.enumerate [p.112] , participant.status [p.139] .
connected	boolean	true if the participant is currently connected to a conference. connected is used in: participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] .
connecting	boolean	true if the scheduled participant is in the process of connecting or is pending a retry. connecting is true for participants whose callStateEx values are proceeding , alerting , or pending . It may also be true for some participants whose callState (deprecated) is dormant or disconnected , because these values are also mapped to the new proceeding and pending states, respectively, that were introduced by the persistence feature in MCU 4.4. connecting is used in: participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] .
connectionUniqueId	integer	Corresponds to the uniqueId returned by a conference or autoattendant. connectionUniqueId is used in: conference.enumerate [p.46] , conference.floor.query [p.54] , conference.status [p.66] , conference.streaming.query [p.72] , participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] , participant.status [p.139] , participant.status (deprecated) [p.147] .
connectPending	boolean	true if sending a "participant.connect" command for this participant will cause either the initial connection to that endpoint (in the event that it was configured with "deferConnection" set) or a re-connection to that endpoint (in the event that it has disconnected). connectPending is used in: participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] , participant.status [p.139] , participant.status (deprecated) [p.147] .

connectTime	dateTime. iso8601	Only returned after the participant is connected. This value is always present if the call state is connected. It may or may not be defined for participants in the disconnected state, depending on whether they were ever connected.
connectTime is used in: participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] , participant.status [p.139] , participant.status (deprecated) [p.147] .		
contentContribution (template)	string	Defines whether endpoints are permitted to contribute the content channel to conferences based on this template. One of true , false , or default .
contentContribution (template) is used in: template.modify [p.172] , template.enumerate [p.168] , template.create [p.163] , template.status [p.176] .		
contentContribution	boolean	Defines whether or not endpoints are permitted to contribute the content channel to this conference. true if content contribution is enabled.
contentContribution is used in: conference.create [p.39] , conference.enumerate [p.46] , conference.modify [p.57] , conference.status [p.66] .		
contentControl	boolean	Defaults to false . Set true to return contentControl statistics.
contentControl is used in: participant.statistics [p.133] .		
contentEnabled	string	One of enabled , h239Only or disabled .
contentEnabled is used in: device.content.modify [p.77] , device.content.query [p.78] .		
contentError	string	Information about problems with outgoing content. One of: notAllowed , noCommonCodecs , noCommonFormats , noCommonSymmetricCodecs , modeMismatch , bitRateMismatch , encryptionNotPossible , notPossible .
contentError is used in: participant.statistics [p.133] .		
contentHandoverEnabled	boolean	true if automatic content handover is enabled.
contentHandoverEnabled is used in: device.content.modify [p.77] , device.content.query [p.78] .		
contentImportant	boolean	Whether or not content is set to be important.
contentImportant is used in: conference.enumerate [p.46] , conference.modify [p.57] , conference.status [p.66] .		
contentInMainVideo	boolean	true if the content can display in the main video channel.
contentInMainVideo is used in: device.content.modify [p.77] , device.content.query [p.78] .		
contentMarkupEnabled	boolean	true if content markup is enabled.
contentMarkupEnabled is used in: device.content.modify [p.77] , device.content.query [p.78] .		
contentMedia	boolean	Defaults to false . Set true to return contentMedia statistics.

contentMedia is used in: [participant.statistics \[p.133\]](#).

contentMode string Defines the content mode of the conference. Either **disabled**, **passthrough**, **transcoded** or **hybrid**.

Value	Description
disabled	Content is not transmitted.
transcoded	Content is always transcoded. The MCU sends out a single, transcoded content stream.
passthrough	Content is not decoded and is simply repackaged and sent out to each eligible endpoint in the conference.
hybrid	The MCU sends out two content streams: a passed through higher resolution one, and a lower resolution stream transcoded and scaled down for any endpoints that are unable to support the higher stream.

contentMode is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#), [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

contentReceive boolean **true** if the endpoint is allowed to receive a separate content stream when participating in a conference.

contentReceive is used in: [addressBookEntry.enumerate \[p.24\]](#).

contentRxActualBitRate integer Actual speed of incoming content in bps

contentRxActualBitRate is used in: [participant.diagnostics \[p.108\]](#).

contentRxBitRateLimitReason string Indicates why the bit rate of the received content stream was limited by the device.

One of:

- notLimited
- viewedSize
- quality
- aggregateBandwidth
- flowControl
- endpointLimitation

contentRxBitRateLimitReason is used in: [participant.diagnostics \[p.108\]](#).

contentRxChannelBitRate integer Capacity of channel in bps

contentRxChannelBitRate is used in: [participant.diagnostics \[p.108\]](#).

contentRxCodec string The codec used on the incoming content stream.

contentRxCodec is used in: [participant.diagnostics \[p.108\]](#), [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

contentRxFrameRate integer Frame rate of incoming content

`contentRxFrameRate` is used in: [participant.diagnostics \[p.108\]](#).

`contentRxFramesReceived` integer Number of received content frames

`contentRxFramesReceived` is used in: [participant.diagnostics \[p.108\]](#).

`contentRxFramesReceivedWithErrors` integer Number of received content frames that had errors

`contentRxFramesReceivedWithErrors` is used in: [participant.diagnostics \[p.108\]](#).

`contentRxHeight` integer Vertical resolution of incoming content

`contentRxHeight` is used in: [participant.diagnostics \[p.108\]](#).

`contentRxJitter` integer A measure of the jitter in the received content

`contentRxJitter` is used in: [participant.diagnostics \[p.108\]](#).

`contentRxLost` integer Number of content packets that should have been received from this participant that were not.

`contentRxLost` is used in: [participant.diagnostics \[p.108\]](#), [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

`contentRxReceived` integer Number of content packets received from this participant.

`contentRxReceived` is used in: [participant.diagnostics \[p.108\]](#), [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

`contentRxSelectedBitRate` integer Participant-selected content bitrate. If one is not set, the MCU assumes the content should be received as fast as possible.

`contentRxSelectedBitRate` is used in: [participant.diagnostics \[p.108\]](#).

`contentRxType` string Type of content received. One of `none`, `h239`, or `bfcf`.

Value	Description
<code>none</code>	Participant is not sending content. No other <code>contentRx</code> fields will be returned.
<code>h239</code>	Participant is sending H.239 content.
<code>bfcf</code>	Participant is sending BFCF content.

`contentRxType` is used in: [participant.diagnostics \[p.108\]](#), [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

`contentRxWidth` integer Horizontal resolution of incoming content.

`contentRxWidth` is used in: [participant.diagnostics \[p.108\]](#).

`contentStreamingSetting` boolean `true` if `contentEnabled` is `enabled` or `h239Only`.

`contentStreamingSetting` is used in: [device.content.query \[p.78\]](#).

`contentStreamingStatus` boolean `true` if the web conferencing feature key is present and `contentEnabled` is either `enabled` or `h239Only`.

contentStreamingStatus is used in: [device.content.query \[p.78\]](#).

contentTransmitResolutions (template) string The resolution for the content channel that will be transmitted to endpoints in conferences based on this template. One of **4to3Only**, **16to9Only**, **allowAll**, or **default**.

Value	Description
4to3Only	The MCU will encode the content and transmit it in a resolution of ratio 4:3 only
16to9Only	The MCU will encode the content and transmit it in a resolution of ratio 16:9 only
allowAll	The MCU will decide on the most optimal resolution depending on information about capabilities sent by the endpoints in the conference
default	Inherit this setting from the parent template

contentTransmitResolutions (template) is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

contentTransmitResolutions string The resolution for the content channel that will be transmitted to endpoints in this conference. One of **4to3Only**, **16to9Only**, or **allowAll**.

The resolution for the content channel that will be transmitted to endpoints in this conference.

Value	Description
4to3Only	The MCU will encode the content and transmit it in a resolution of ratio 4:3 only
16to9Only	The MCU will encode the content and transmit it in a resolution of ratio 16:9 only
allowAll	The MCU will decide on the most optimal resolution depending on information about capabilities sent by the endpoints in the conference.

contentTransmitResolutions is used in: [conference.create \[p.39\]](#).

contentTxActualBitRate integer Actual speed of outgoing content in bps

contentTxActualBitRate is used in: [participant.diagnostics \[p.108\]](#).

contentTxBitRateLimitReason string Indicates why the bit rate of the transmitted content stream was limited by the device.

- notLimited
- viewedSize
- quality
- aggregateBandwidth
- flowControl
- endpointLimitation

contentTxBitRateLimitReason is used in: [participant.diagnostics \[p.108\]](#).

contentTxChannelBitRate integer Capacity of channel in bps

contentTxChannelBitRate is used in: [participant.diagnostics \[p.108\]](#).

contentTxCodec (template) string The codec used to transmit content in conferences based on this template. If content is to be transcoded, this is the output format of the transcoder; **h263+**, **h264**, **automatic**, or **default**. This setting does not apply in **passthrough** mode.

If the output format is **automatic**, the MCU chooses the most suitable codec, either H.263+ or H.264, and changes between them as required. **default** means the template inherits this setting from its parent template.

contentTxCodec (template) is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

contentTxCodec string The codec used to transmit content. If content is being transcoded, it is the output format of the transcoder; either **h263+**, **h264**, or **automatic** (default). This setting does not apply in **passthrough** mode.

If the output format is **automatic**, the MCU chooses the most suitable codec, either H.263+ or H.264, and changes between them as required.

contentTxCodec is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#), [participant.diagnostics \[p.108\]](#), [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

contentTxError string Provides a reason for a content transmission error.

One of:

- notAllowed
- noCommonCodecs
- noCommonFormats
- noCommonSymmetricCodecs
- modeMismatch
- bitRateMismatch
- encryptionNotPossible
- notPossible

These correspond to the messages shown on the participant page of the web UI.

contentTxError is used in: [participant.diagnostics \[p.108\]](#).

contentTxFrameRate integer Frame rate of outgoing content

contentTxFrameRate is used in: [participant.diagnostics \[p.108\]](#).

contentTxHeight integer Vertical resolution of outgoing content

contentTxHeight is used in: [participant.diagnostics \[p.108\]](#).

contentTxMinimumBitRate (*template*) string The minimum bit rate to use for transmitting content, in bps, in conferences based on this template. One of: 0, 64000, 128000, 192000, 256000, 320000, 384000, 512000, 768000, 1000000, 1250000, 1500000, or `default` (inherit this setting from the parent template).

contentTxMinimumBitRate (*template*) is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

contentTxMinimumBitRate string The minimum bit rate to use for transmitting content, in bps. One of: 0, 64000, 128000, 192000, 256000, 320000, 384000, 512000, 768000, 1000000, 1250000, or 1500000.

contentTxMinimumBitRate is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

contentTxReportedLost integer Number of content packets reported as lost.

contentTxReportedLost is used in: [participant.diagnostics \[p.108\]](#), [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

contentTxSelectedBitRate integer Participant-selected content bitrate. If one is not set, the MCU assumes the content should be sent as fast as possible.

contentTxSelectedBitRate is used in: [participant.diagnostics \[p.108\]](#).

contentTxSent integer Number of content packets sent.

contentTxSent is used in: [participant.diagnostics \[p.108\]](#), [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

contentTxType string Type of content transmitted. One of `none`, `h239`, `bfcf`, or `mainVideo`.

Value	Description
none	MCU is not sending content. No other <code>contentTx</code> fields will be returned.
h239	MCU is sending H.239 content.
bfcf	MCU is sending BFCF content.
mainVideo	MCU is sending content in main video. No other <code>contentTx</code> fields will be returned.

contentTxType is used in: [participant.diagnostics \[p.108\]](#), [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

contentTxWidth integer Horizontal resolution of outgoing content

contentTxWidth is used in: [participant.diagnostics \[p.108\]](#).

contentType string The type of content being sent or received.

contentType is used in: [participant.statistics \[p.133\]](#).

count (*videoports*) integer The allocated number of video ports of this `type`.

`count` (*videoports*) is used in: [device.query \[p.87\]](#).

<code>count</code>	integer	The number of users of this codec.
--------------------	---------	------------------------------------

`count` is used in: [conference.streaming.query \[p.72\]](#).

<code>cpLayout</code>	string	This sets the initial conference view layout for the video sent to the participant. Refer to Conference layouts [p.188] for details.
-----------------------	--------	--

`cpLayout` is used in: [conference.streaming.modify \[p.71\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

<code>cpuLoad</code>	integer	The CPU load as a percentage of the maximum.
----------------------	---------	--

`cpuLoad` is used in: [device.health.query \[p.81\]](#).

<code>currentLayout</code>	integer	The actual layout in use for the video stream being sent by the MCU to streaming viewers. Refer to Conference layouts [p.188] for details.
----------------------------	---------	--

`currentLayout` is used in: [conference.streaming.query \[p.72\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

<code>currentRevision</code>	integer	A number that indicates the current revision of this enumeration. You can use this as a <code>lastRevision</code> input to a future <code>enumerate</code> call to retrieve only the changes between the two enumerations.
------------------------------	---------	--

`currentRevision` is used in: [autoAttendant.enumerate \[p.33\]](#), [conference.enumerate \[p.46\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#),

<code>currentState</code>	struct	The current state of the participant. This is only present if requested in the <code>operationScope</code> .
---------------------------	--------	--

`currentState` is used in: [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

<code>currentTime</code>	dateTime. iso8601	The system's current time (UTC).
--------------------------	----------------------	----------------------------------

`currentTime` is used in: [cdrlog.enumerate \[p.36\]](#), [device.query \[p.87\]](#), [device.time.modify \[p.92\]](#), [device.time.query \[p.93\]](#).

<code>customCodecs</code>	struct	A collection of structs that indicate which codecs the device advertises that it can use to send and receive audio and video. The struct is absent if <code>customCodecSelection</code> is <code>false</code> .
---------------------------	--------	---

`customCodecs` is used in: [addressBookEntry.enumerate \[p.24\]](#).

<code>customCodecSelection</code>	boolean	Indicates whether the device advertises a custom set of codecs.
-----------------------------------	---------	---

`customCodecSelection` is used in: [addressBookEntry.enumerate \[p.24\]](#).

customLayout integer The index of the video layout seen by the participant(s), depending on the parameter's context. See [Conference layouts \[p.188\]](#) for a list of available layouts and corresponding index values.

customLayout is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

customLayoutEnabled boolean **true** if the custom layout is enabled, **false** otherwise.

customLayoutEnabled is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.status \[p.66\]](#).

Index of parameters: D

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

defaultGateway string The device's IPv4 default gateway in dotted quad format.
defaultGateway is used in: [device.network.query \[p.84\]](#).

defaultIpv4Gateway string (31) The device's IPv4 default gateway in dotted quad format. Deprecates **defaultGateway**.
defaultIpv4Gateway is used in: [device.network.modify \[p.82\]](#), [device.network.query \[p.84\]](#).

defaultIpv6Gateway string (79) The address of the IPv6 default gateway in CIDR format.
defaultIpv6Gateway is used in: [device.network.modify \[p.82\]](#), [device.network.query \[p.84\]](#).

defaultLayout string Describes the participant's default conference view layout if configured. One of **default**, **familyIndex**, **layoutIndex**, **conferenceCustom**.

Describes the participant's default conference view layout if configured.

Value	Description
default	The participant uses the default view family as set on the device that hosts the conference
familyIndex	The participant uses a layout from a specific family of layouts. There are 5 layout families, indexed by a number between 1 and 5. family2 , for example, includes full screen layouts.
layoutIndex	The participant uses a specific layout. There are over 50 specific layouts, indexed by the number after 'layout'. layout3 , for example, is a 3 by 3 grid of equal-sized panes.
conferenceCustom	The participant uses the conference's custom layout.

defaultLayout is used in: [addressBookEntry.enumerate \[p.24\]](#).

deferConnection boolean If **true**, don't call out to this participant immediately, but wait for a **participant.connect** command.
 You cannot set **deferConnection** to true for participants where **participantType** is **ad_hoc**.

deferConnection is used in: [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

deletableIndex integer The log index of the most recent event that was archived into a log file. The delete command works on whole files, so you can delete up to the last event that went into a file.

deletableIndex is used in: [auditlog.query \[p.31\]](#), [cdrlog.query \[p.38\]](#).

deleteIndex (<i>CDR log</i>)	integer	An event identifier that selects which whole CDR files will be deleted. Any whole files whose highest index is below the supplied value will be deleted from CDR log storage. If you supply the value returned in <code>cdrlog.query.deleteableIndex</code> , you will delete all the files stored at the time of that query.
deleteIndex (<i>CDR log</i>) is used in: cdrlog.delete [p.35] .		
deleteIndex (<i>audit log</i>)	integer	You can delete logs in chunks of 400. To delete logs, you can enter the value returned by <code>auditlog.query.deleteableIndex</code> . This will delete all complete chunks (400 logs each) below this number, leaving the residuals. Alternatively, you can delete less than this amount by picking a number below the value of <code>deleteableIndex</code> . This will delete all complete chunks (400 logs) below that number, leaving any residuals.
deleteIndex (<i>audit log</i>) is used in: auditlog.delete [p.30] .		
description	string	Additional information about the conference.
description is used in: conference.create [p.39] , conference.enumerate [p.46] , conference.modify [p.57] , conference.status [p.66] .		
destination	string	IP address of the route's destination.
destination is used in: route.add [p.151] , route.enumerate [p.153] .		
dhcp	boolean	Defines whether or not to use DHCP to obtain an IPv4 address.
dhcp is used in: device.network.query [p.84] .		
dhcpv4	boolean	Defines whether or not to use DHCP to obtain an IPv4 address. Deprecates <code>dhcp</code> .
dhcpv4 is used in: device.network.modify [p.82] , device.network.query [p.84] .		
direction	string	One of <code>up</code> , <code>down</code> , <code>left</code> , <code>right</code> , <code>zoomIn</code> , <code>zoomOut</code> , <code>focusIn</code> , or <code>focusOut</code> .
direction is used in: participant.fecc [p.125] .		
disconnected	boolean	<code>true</code> if the participant has been connected to a conference, but is now disconnected.
disconnected is used in: participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] .		
disconnectReason	string	Only returned after the participant has disconnected; this contains one of the Disconnect reasons [p.185] .
disconnectReason is used in: participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] , participant.status [p.139] , participant.status (deprecated) [p.147] .		
disconnectTime	dateTime. iso8601	Only returned after the participant has disconnected.

disconnectTime is used in: [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

displayName string The display name of the participant.

displayName is used in: [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

displayNameOverrideStatus boolean **true** if the endpoint uses the **displayNameOverrideValue** text to identify itself to other participants.

displayNameOverrideStatus is used in: [addressBookEntry.enumerate \[p.24\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

displayNameOverrideValue string This value overrides the participant's display name if **displayNameOverrideStatus** is **true**.

displayNameOverrideValue is used in: [addressBookEntry.enumerate \[p.24\]](#), [participant.add \[p.103\]](#), [participant.modify \[p.127\]](#).

dns struct The struct members represent the device's DNS parameters.

dns is used in: [device.network.modify \[p.82\]](#), [device.network.query \[p.84\]](#).

dnsConfiguration string (10) Defines how the device gets its DNS configuration; one of **portAIPv4**, **portAIPv6**, **portBIPv4**, **portBIPv6** or **manual**. If **manual**, you must supply a name server address.

For example, if you set **dnsConfiguration** to **portAIPv6**, the device will automatically get a name server address using DHCP over the IPv6 network connected to Ethernet port A.

dnsConfiguration is used in: [device.network.modify \[p.82\]](#).

dnsStatus string The status of the DNS lookup of the gatekeeper's address. One of **inProgress**, **resolved**, or **failed**.

dnsStatus is used in: [gatekeeper.query \[p.100\]](#).

domainName string (255) The domain name (DNS suffix).

domainName is used in: [device.network.modify \[p.82\]](#), [device.network.query \[p.84\]](#).

domainName string (255) The domain name (DNS suffix).

domainName is used in: [device.network.modify \[p.82\]](#), [device.network.query \[p.84\]](#).

dormant boolean **true** if the pre-configured participant is not trying to connect.

dormant is used in: [participant.enumerate \[p.112\]](#).

dtmfMuteControl (*template*) string Deprecated by `inCallMenuControlChair`(*template*) and `inCallMenuControlGuest`(*template*). Defines whether or not participants, in conferences based on this template, can mute audio by pressing *6 on the remote control. One of `true`, `false`, or `default` (inherit this setting from the parent template).

`dtmfMuteControl` (*template*) is used in: [template.create \[p.163\]](#), [template.enumerate \[p.168\]](#), [template.modify \[p.172\]](#), [template.status \[p.176\]](#).

dtmfMuteControl boolean Deprecated by `inCallMenuControlChair` and `inCallMenuControlGuest`. Defines whether or not a participant can mute audio by pressing *6 on the remote control.

`dtmfMuteControl` is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

dtmfSequence string (127) A string of characters that will be converted to DTMF signals, allowing the device to navigate through audio menus. The sequence may contain 0–9, *, #, and ,. The comma becomes a two second pause.

A DTMF sequence is used for dialing systems with keypad/tone navigation menus, such as an audio bridge. The sequence may contain the digits 0–9, the star/asterisk character *, the hash/pound character #, and the comma character ,.

After the call connects, the MCU waits for two seconds and then sends the corresponding tones, in sequence, at the rate of two tones per second. The comma character is interpreted by the MCU as a two second pause, and you can use as many of them as necessary to deliver the right tones at the right times.

For example, assume you want the MCU to dial out to a PIN protected audio conference on an audio bridge. The conference ID is 555 and the PIN is 888. The audio bridge requires that you press # after entering the ID and after entering the PIN. The DTMF sequence for this example could be 555#, , 888#.

`dtmfSequence` is used in: [addressBookEntry.enumerate \[p.24\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

durationSeconds integer The period of time, in seconds, for which this item is active.

`durationSeconds` is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#), [participant.message \[p.126\]](#).

Index of parameters: E

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

e164	string	An E.164 number.
e164 is used in: addressBookEntry.enumerate [p.24] .		
enabled	boolean	true if this feature or item is enabled.
enabled is used in: conference.floor.query [p.54] , conference.paneplacement.modify [p.61] , conference.paneplacement.query [p.63] , conferenceme.query [p.76] , device.encryption.query [p.80] , device.network.query [p.84] , services.query [p.158] , streaming.query [p.162] .		
encryption	boolean	Defines whether or not the received or transmitted stream is encrypted. This parameter could apply to content, audio or video streams.
encryption is used in: participant.statistics [p.133] .		
encryptionRequired (template)	string	The encryption setting for conferences based on this template, if the encryption feature key is enabled. If true , encryption is required for these conferences. Otherwise, encryption is optional. default causes the template to inherit this setting from its parent template.
encryptionRequired (template) is used in: template.create [p.163] , template.enumerate [p.168] , template.modify [p.172] , template.status [p.176] .		
encryptionRequired	boolean	The encryption setting for this conference, if the encryption feature key is enabled. If true , encryption is required for this conference. Otherwise, encryption is optional.
encryptionRequired is used in: conference.create [p.39] , conference.enumerate [p.46] , conference.modify [p.57] , conference.status [p.66] .		
endTime	dateTime. iso8601	If you do not specify an end time, then the conference will be permanent (until it is explicitly deleted). Your application code should use durationSeconds instead.
endTime is used in: conference.create [p.39] , conference.modify [p.57] .		
energyMillidB	integer	The received audio energy in millidecibels.
energyMillidB is used in: participant.statistics [p.133] .		
enforceMaximumAudioPorts (template)	boolean	Defines whether conferences based on this template will enforce the maximumAudioPorts limit. One of true , false , or default .
Assumed to be true if not defined.		

Value	Description
true	The MCU enforces the maximumAudioPorts limit

Value	Description
false	The MCU does not enforce the <code>maximumAudioPorts</code> limit
default	Inherit this setting from the parent template

`enforceMaximumAudioPorts` (*template*) is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

`enforceMaximumAudioPorts` boolean Defines whether the conference enforces the `maximumAudioPorts` limit. Assumed to be `true` if absent.

`enforceMaximumAudioPorts` is used in: [conference.create \[p.39\]](#), [conference.modify \[p.57\]](#).

`enforceMaximumVideoPorts` (*template*) string Defines whether conferences based on this template will enforce the `maximumVideoPorts` limit. One of `true`, `false`, or `default`.

Assumed to be `true` if absent.

Value	Description
true	The MCU enforces the <code>maximumVideoPorts</code> limit
false	The MCU does not enforce the <code>maximumVideoPorts</code> limit
default	Inherit this setting from the parent template

`enforceMaximumVideoPorts` (*template*) is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

`enforceMaximumVideoPorts` boolean Defines whether the conference enforces the `maximumVideoPorts` limit. Assumed to be `true` if absent.

`enforceMaximumVideoPorts` is used in: [conference.create \[p.39\]](#), [conference.modify \[p.57\]](#).

`enumerateFilter` string A filter expression. The enumeration results depend on the supplied expression.

`enumerateFilter` is used in: [conference.enumerate \[p.46\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#),

`enumerateID` string The device returns this index if the requested data is too large for one response. Pass this parameter in a repeat of the call to return the next batch of data.

Enumerate calls may return many results so all of them will accept this parameter and may include this parameter in the response.

If the response includes an `enumerateID`, the application should pass the ID to the subsequent `enumerate` call to retrieve the next set of results. If the response does not include an `enumerateID`, there are no more results in the enumeration.

If the application omits the `enumerateID`, the target device will start a new enumeration and return the first set of results.

`enumerateID` is used in: [addressBookEntry.enumerate \[p.24\]](#), [autoAttendant.enumerate \[p.33\]](#), [conference.enumerate \[p.46\]](#), [gateway.enumerate \[p.102\]](#), [participant.enumerate \[p.112\]](#),

[participant.enumerate \(deprecated\) \[p.121\]](#),

ethernetAutomatic	boolean	true for the Ethernet interface to configure itself automatically. If you set this to false you must supply the speed and fullDuplex parameters.
--------------------------	---------	--

ethernetAutomatic is used in: [device.network.modify \[p.82\]](#).

events(CDR)	array	List of the new events; these are structures with some common fields (time, type, index) and other fields specific to the event type.
--------------------	-------	---

events (CDR log) is used in: [cdrlog.enumerate \[p.36\]](#).

events (feedback)	struct	Each member of the events struct associates a string (feedback event name) to a boolean (true to subscribe).
--------------------------	--------	---

For example, the following XML fragment shows how you would define a member of the **events** struct so that the receiver subscribes to **restart** events.

```
<param>
  <value>
    <struct>
      ...
      <member>
        <name>events</name>
        <value>
          <struct>
            <member>
              <name>restart</name>
              <value>
                <boolean>true</boolean>
              </value>
            </member>
          </struct>
        </value>
      </member>
    </struct>
  </value>
</param>
```

events is used in: [feedbackReceiver.configure \[p.94\]](#), [feedbackReceiver.reconfigure \[p.96\]](#).

eventsRemaining	boolean	Whether there is data remaining after this. Provided to avoid putting all data in a single call.
------------------------	---------	--

eventsRemaining is used in: [cdrlog.enumerate \[p.36\]](#).

Index of parameters: F

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

fanStatus	string	One of <code>ok</code> , <code>outOfSpec</code> , or <code>critical</code> .
fanStatus is used in: device.health.query [p.81] .		
fanStatusWorst	string	One of <code>ok</code> , <code>outOfSpec</code> , or <code>critical</code> .
fanStatusWorst is used in: device.health.query [p.81] .		
fecOverhead	integer	Only returned if FEC (forward error correction) is negotiated and enabled.
fecOverhead is used in: participant.statistics [p.133] .		
fecRecovered	integer	Only returned if FEC (forward error correction) is negotiated and enabled.
fecRecovered is used in: participant.statistics [p.133] .		
filter (route)	string	Filters the returned routes by the route type. One of <code>configured</code> , <code>automatic</code> , or <code>both</code> . Defaults to <code>both</code> .
filter (route) is used in: route.enumerate [p.153] .		
filter	struct	A struct that contains boolean switches to filter the statistics. All the switches default to <code>false</code> (do not return these statistics).
filter (stats) is used in: participant.statistics [p.133] .		
filter	array	An array of strings, which contain the names of event types by which to filter the response. Omit <code>filter</code> to return all event types or include a subset of the following: <code>scheduledConferenceStarted</code> , <code>ad-hocConferenceStarted</code> , <code>conferenceFinished</code> , <code>participantJoined</code> , <code>participantLeft</code>
filter is used in: cdrlog.enumerate [p.36] .		
finishedBooting	boolean	<code>true</code> after the device is fully booted. Will not revert to <code>false</code> until a reboot starts.
finishedBooting is used in: device.query [p.87] .		
firstIndex	integer	The index of the oldest stored event.
firstIndex is used in: auditlog.query [p.31] , cdrlog.query [p.38] .		
floorParticipant	struct	A structure that identifies which participant has the floor.
floorParticipant is used in: conference.enumerate [p.46] , conference.floor.modify [p.53] , conference.floor.query [p.54] , conference.status [p.66] .		

floorStatus string One of `inactive` or `assign`. If you set `floorStatus` to `assign` you must provide a `floorParticipant` struct. One of `inactive`, `active`, or `assigned`. If it is `active` or `assigned`, a `floorParticipant` struct will be included in the response.

`floorStatus` is used in: [conference.enumerate \[p.46\]](#), [conference.floor.modify \[p.53\]](#), [conference.floor.query \[p.54\]](#), [conference.status \[p.66\]](#).

flowControlReceived integer Count of flow control requests received.

`flowControlReceived` is used in: [participant.statistics \[p.133\]](#).

flowControlSent integer Count of flow control requests sent.

`flowControlSent` is used in: [participant.statistics \[p.133\]](#).

focusParticipant struct The structure contains participant parameters that identify which participant displays in the largest pane if `focusType` is `participant`.

`focusParticipant` is used in: [conference.streaming.modify \[p.71\]](#), [conference.streaming.query \[p.72\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

focusType string Indicates the endpoint's focus. One of `participant`, `voiceActivated`, or `h239`.

Value	Description
<code>participant</code>	The focus remains on a particular participant.
<code>voiceActivated</code>	The focus changes to show the loudest speaker.
<code>h239</code>	The focus remains on the content channel.

`focusType` is used in: [conference.streaming.modify \[p.71\]](#), [conference.streaming.query \[p.72\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

format string One of `wmp`, `qt64`, `qt70`, or `realPlayer`. The `format` determines the `audioCodec` and `videoCodec`.

`format` is used in: [streaming.modify \[p.161\]](#), [streaming.query \[p.162\]](#).

format1 struct A struct whose contents define a streaming format.

`format1` is used in: [streaming.modify \[p.161\]](#), [streaming.query \[p.162\]](#).

format2 struct A struct whose contents define a streaming format.

`format2` is used in: [streaming.modify \[p.161\]](#), [streaming.query \[p.162\]](#).

frameErrors integer Count of frames with errors in this stream.

`frameErrors` is used in: [participant.statistics \[p.133\]](#),

frameRate integer The frame rate of the video or content stream, in frames per second (fps).

frameRate is used in: [participant.statistics \[p.133\]](#),

framesTransferred	integer	Count of audio, video, or content frames received, depending on where the parameter occurs.
--------------------------	---------	---

framesTransferred is used in: [participant.statistics \[p.133\]](#).

fullDuplex	boolean	true if the port supports a full-duplex connection, false for half-duplex.
-------------------	---------	--

fullDuplex is used in: [device.network.modify \[p.82\]](#), [device.network.query \[p.84\]](#).

furFilteringEnabled	boolean	true if video fast update request filtering is enabled.
----------------------------	---------	--

furFilteringEnabled is used in: [device.content.modify \[p.77\]](#), [device.content.query \[p.78\]](#).

fursReceived	integer	Count of fast update requests (FURs) received by the device (this statistic is only present for video or content control).
---------------------	---------	--

fursReceived is used in: [participant.statistics \[p.133\]](#).

fursSent	integer	Count of fast update requests (FURs) sent by the device (this statistic is only present for video or content control).
-----------------	---------	--

fursSent is used in: [participant.statistics \[p.133\]](#).

Index of parameters: G

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

g711 boolean Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

g711 is used in: [addressBookEntry.enumerate \[p.24\]](#).

g722 boolean Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

g722 is used in: [addressBookEntry.enumerate \[p.24\]](#).

g722.1 boolean Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

g722.1 is used in: [addressBookEntry.enumerate \[p.24\]](#).

g722.1c boolean Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

g722.1c is used in: [addressBookEntry.enumerate \[p.24\]](#).

g723.1 boolean Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

g723.1 is used in: [addressBookEntry.enumerate \[p.24\]](#).

g728 boolean Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

g728 is used in: [addressBookEntry.enumerate \[p.24\]](#).

g729 boolean Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

g729 is used in: [addressBookEntry.enumerate \[p.24\]](#).

gatekeeperUsage string (8) Defines how the gatekeeper is used. One of **disabled**, **enabled**, or **required**.

Value	Description
disabled	The gatekeeper is not used.
enabled	The gatekeeper is used but, if it can't match the call, the call is attempted anyway.
required	The gatekeeper must be used to match the call.

gatekeeperUsage is used in: [gatekeeper.modify \[p.98\]](#), [gatekeeper.query \[p.100\]](#).

gateway string One of **A** or **B** (to use the default gateway configured for that ethernet port), or the IP address of the gateway of this route (must be a valid IP address of the same type as **destination**).The IP address of the gateway (or next hop) of this route.

gateway is used in: [route.add \[p.151\]](#), [route.enumerate \[p.153\]](#).

gatewayAddress string (63) The address of an H.323 gateway, if required. Only used if protocol is **h323**. This corresponds to the **address** parameter of the gateway as returned by **gateway.enumerate**.

gatewayAddress is used in: [addressBookEntry.enumerate \[p.24\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#),

gatewayName string Present in entries for H.323 endpoints which are configured to use a gateway. This name corresponds to the **name** parameter of a gateway returned by **gateway.enumerate**.

gatewayName is used in: [addressBookEntry.enumerate \[p.24\]](#).

gateways array A collection of structures, each of which describes a gateway.

gateways is used in: [gateway.enumerate \[p.102\]](#).

guest boolean **true** if the participant is a guest, **false** if the participant is a chair.

guest is used in: [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

guestNumericId string If it is configured, this value is used by guests (instead of **numericId**) to access the conference.

guestNumericId is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

guestPin string Security PIN that a guest can use to gain access to this conference.

guestPin is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

Index of parameters: H

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

h239ContributionDefault boolean Defines whether or not the endpoint will use the box-wide H.239 contribution setting.

h239ContributionDefault is used in: [addressBookEntry.enumerate \[p.24\]](#).

h239ContributionEnabled boolean Defines whether or not the endpoint will be able contribute H.239, if **h239ContributionDefault** is **false**.

h239ContributionEnabled is used in: [addressBookEntry.enumerate \[p.24\]](#).

h239Enabled boolean Deprecated by **contentMode**. If you set **h239Enabled** to **true**, **contentMode** will be set to **transcoded**. If you set **h239Enabled** to **false**, **contentMode** will be set to **disabled**.

h239Enabled is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#), [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

h239Important boolean Whether the H.239 channel is set to be important. Consider this setting deprecated by **contentImportant**. The setting will still work however, even if the content channel is SIP or VNC or content from a main video participant.

h239Important is used in: [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

h239Negotiation string Defines how the MCU presents itself for h239 token negotiation. One of **As master**, **As slave**, or **Mimic slave**.

When exchanging content with an endpoint in an H.323 call, the MCU acts as a master unit and the endpoint as a slave unit for the purpose of H.239 token negotiation. However, in order for the MCU to exchange content with a cascaded third party MCU, the MCU must appear to the third party MCU to be a slave unit.

The MCU can be configured as a true slave, in which case content will only be sent if the third party MCU master accepts the token request, or as a mimic slave where content is sent to all other connected endpoints even if the third party MCU rejects the token request.

Value	Description
As master	The MCU only acts as master in H.239 token negotiation.
As slave	The MCU acts as the slave in H.239 token negotiation and can send content to a master unit if it accepts the token request.
Mimic slave	The MCU acts as a mimic slave in H.239 token negotiation and will try to send content to all other endpoints/units even if this unit (i.e. the mimic slave) rejects the token request.

h239Negotiation is used in: [addressBookEntry.enumerate \[p.24\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

h261 boolean Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

h261 is used in: [addressBookEntry.enumerate \[p.24\]](#).

h263 boolean Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

h263 is used in: [addressBookEntry.enumerate \[p.24\]](#).

h263i boolean Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

h263i is used in: [addressBookEntry.enumerate \[p.24\]](#).

h263+ boolean Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

h263+ is used in: [addressBookEntry.enumerate \[p.24\]](#).

h264 boolean Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

h264 is used in: [addressBookEntry.enumerate \[p.24\]](#).

h323ID string (255) The H.323 ID used by the device to register with the gatekeeper.

h323ID is used in: [gatekeeper.modify \[p.98\]](#), [gatekeeper.query \[p.100\]](#).

h323IDStatus string The current status of the ID registration process.

The current status of the ID registration process.

Value	Description
idle	
registering	
registered	
deregistering	
pendingReregistration	
waitingRetry	
noID	
idTooLong	

h323IDStatus is used in: [gatekeeper.query \[p.100\]](#).

height integer The maximum width and height of this stream. Only present for defined video streams

height is used in: [conference.streaming.query \[p.72\]](#), [participant.statistics \[p.133\]](#).

hostName string (255) The host name of queried device. Deprecated in API version 2.8.

hostName is used in: [device.network.modify \[p.82\]](#), [device.network.query \[p.84\]](#).

Index of parameters: I

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

important boolean `true` means this participant's video is important; it will dominate the layout.

important is used in: [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

inCallMenuControlChair string Defines the level of control a chairperson has over the in call menu. One of `off`, `local`, `conference`, or `advanced`.

Value	Description
<code>off</code>	The in call menu is disabled for this conference.
<code>local</code>	Chairpersons may use the in call menu to modify their own in call settings.
<code>conference</code>	Chairpersons may use the in call menu to modify their own in call settings, those of other participants, and some conference-wide settings.
<code>advanced</code>	Chairpersons have conference level menu control and may also modify some conference configuration features such as PINs.

inCallMenuControlChair is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

inCallMenuControlChair (template) string Defines the level of control a chairperson has over the in call menu, in conferences based on this template. One of `default`, `off`, `local`, `conference`, or `advanced`.

Value	Description
<code>default</code>	Inherit this setting from the parent template.
<code>off</code>	The in call menu is disabled for this conference.
<code>local</code>	Chairpersons may use the in call menu to modify their own in call settings.
<code>conference</code>	Chairpersons may use the in call menu to modify their own in call settings, those of other participants, and some conference-wide settings.
<code>advanced</code>	Chairpersons have conference level menu control and may also modify some conference configuration features such as PINs.

inCallMenuControlChair (template) is used in: [template.create \[p.163\]](#), [template.enumerate \[p.168\]](#), [template.modify \[p.172\]](#), [template.status \[p.176\]](#).

inCallMenuControlGuest string Defines the level of control a guest has over the in call menu. Either `off` or `local`.

Value	Description
off	The in call menu is disabled for guests.
local	Guests may use the in call menu to modify their own in call settings.

inCallMenuControlGuest is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

inCallMenuControlGuest (*template*) string Defines the level of control a guest has over the in call menu, in conferences based on this template. One of **default**, **off**, or **local**.

Value	Description
default	Inherit this setting from the parent template.
off	The in call menu is disabled for guests.
local	Guests may use the in call menu to modify their own in call settings.

inCallMenuControlGuest (*template*) is used in: [template.create \[p.163\]](#), [template.enumerate \[p.168\]](#), [template.modify \[p.172\]](#), [template.status \[p.176\]](#).

index (*CDR log*) integer The index of the CDR log message.

index (*CDR log*) is used in: [cdrlog.enumerate \[p.36\]](#).

index (*CDR log enumerate call*) integer Index from which to get events. The device returns the **nextIndex** so the application can use it to retrieve the next enumeration of CDR data.
If **index** is omitted, negative, or greater (by 2 or more) than the highest index, then the device will enumerate events from the beginning of the CDR log.

index (*CDR log enumerate call*) is used in: [cdrlog.enumerate \[p.36\]](#).

index (*feedback receiver*) integer A number between 1 and 20 (inclusive) that indicates the position of this feedback receiver in the device's table of feedback receivers.

index (*feedback receiver*) is used in: [feedbackReceiver.query \[p.95\]](#).

index (*pane*) integer A number that identifies the pane with respect to other panes. A value between 0 and 19, where lower numbers are generally more prominent in the layout.

index (*pane*) is used in: [conference.paneplacement.modify \[p.61\]](#), [conference.paneplacement.query \[p.63\]](#).

initialAudioMuted boolean **true** if the endpoint's audio is initially muted.

initialAudioMuted is used in: [addressBookEntry.enumerate \[p.24\]](#).

initialVideoMuted boolean **true** if the endpoint's video is initially muted.

initialVideoMuted is used in: [addressBookEntry.enumerate \[p.24\]](#).

Interlaced	boolean	Defines whether or not the video in this sent or received stream is interlaced. Deprecates videoTxInterlaced and videoRxInterlaced .
Interlaced is used in: participant.statistics [p.133] .		
ip	string	the IP address of the gatekeeper (if dnsStatus is resolved)
ip is used in: gatekeeper.query [p.100] .		
ipAddress	string	IPv4 address in dotted-quad format.
ipAddress is used in: device.network.query [p.84] , participant.enumerate [p.112] , participant.status [p.139] .		
ipRangeFinish	string	The last IP address in the multicast range.
ipRangeFinish is used in: streaming.modify [p.161] , streaming.query [p.162] .		
ipRangeStart	string	The first IP address in the multicast range.
ipRangeStart is used in: streaming.modify [p.161] , streaming.query [p.162] .		
ipv4Address	string (31)	IPv4 address in dotted-quad format.
ipv4Address is used in: device.network.modify [p.82] , device.network.query [p.84] .		
ipv4Enabled	boolean	true if IPv4 interface is enabled.
ipv4Enabled is used in: device.network.modify [p.82] , device.network.query [p.84] .		
ipv4MulticastRange	struct	Contains parameters that define an IPv4 multicast range.
ipv4MulticastRange is used in: streaming.modify [p.161] , streaming.query [p.162] .		
ipv4Preference	string	Either A or B , indicates which Ethernet port is preferred for traffic bound for IPv4 destinations.
ipv4Preference is used in: route.preferences.modify [p.154] , route.preferences.query [p.155] .		
ipv4Routes	array	An array of structs, each of which represents an IPv4 route.
ipv4Routes is used in: route.enumerate [p.153] .		
ipv4SubnetMask	string (31)	The IPv4 subnet mask in dotted quad format. Deprecates subnetMask .
ipv4SubnetMask is used in: device.network.modify [p.82] , device.network.query [p.84] .		
ipv6Address	string (79)	The IPv6 address in CIDR format.
ipv6Address is used in: device.network.modify [p.82] , device.network.query [p.84] .		
ipv6Conf	string (10)	Indicates how the IPv6 address is assigned; either automatic (by SLAAC/DHCPv6) or manual .
ipv6Conf is used in: device.network.modify [p.82] , device.network.query [p.84] .		
ipv6Enabled	boolean	true if IPv6 interface is enabled.

ipv6Enabled is used in: [device.network.modify \[p.82\]](#), [device.network.query \[p.84\]](#).

ipv6MulticastRange	struct	Contains parameters that define an IPv6 multicast range.
---------------------------	--------	--

ipv6MulticastRange is used in: [streaming.modify \[p.161\]](#), [streaming.query \[p.162\]](#).

ipv6Preference	string	Either A or B , indicates which Ethernet port is preferred for traffic bound for IPv6 destinations.
-----------------------	--------	---

ipv6Preference is used in: [route.preferences.modify \[p.154\]](#), [route.preferences.query \[p.155\]](#).

ipv6PrefixLength	integer	The length of the IPv6 address prefix.
-------------------------	---------	--

ipv6PrefixLength is used in: [device.network.modify \[p.82\]](#), [device.network.query \[p.84\]](#).

ipv6Routes	array	An array of structs, each of which represents an IPv6 route (the structs are the same as described above for the IPv4 routes array).
-------------------	-------	--

ipv6Routes is used in: [route.enumerate \[p.153\]](#).

Index of parameters: J

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [!](#) | [m](#) | [n](#) | [o](#) | [p](#) | [g](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

 jitter 	integer	Current jitter in this stream, measured in milliseconds (ms).
jitter is used in: participant.statistics [p.133] ,		
 jitterBuffer 	integer	The jitter buffer shows the current play out delay added to outgoing media to accommodate for packet arrival jitter. Larger values indicate a longer buffer, i.e. more jitter from incoming streams.
jitterBuffer is used in: participant.statistics [p.133] .		
 joinAGC 	boolean	Whether AGC should be used by default for participants joining this conference
joinAGC is used in: conference.create [p.39] , conference.enumerate [p.46] , conference.modify [p.57] , conference.status [p.66] .		
 joinAGC (template) 	string	Whether AGC should be used by default for participants joining this conference. default if this template inherits the joinAGC setting of its parent template.
joinAGC (template) is used in: template.create [p.163] , template.enumerate [p.168] , template.modify [p.172] , template.status [p.176] .		
 joinAudioMuted (template) 	string	Mutes audio on join. One of true , false , or default to inherit this setting from the parent template.
joinAudioMuted (template) is used in: template.modify [p.172] , template.enumerate [p.168] , template.create [p.163] , template.status [p.176] .		
 joinAudioMuted 	boolean	Audio mute on join.
joinAudioMuted is used in: conference.create [p.39] , conference.enumerate [p.46] .		
 joinVideoMuted (template) 	string	Mutes video on join. One of true , false , or default to inherit this setting from the parent template.
joinVideoMuted (template) is used in: template.modify [p.172] , template.enumerate [p.168] , template.create [p.163] , template.status [p.176] .		
 joinVideoMuted 	boolean	Video mute on join.
joinVideoMuted is used in: conference.create [p.39] , conference.enumerate [p.46] .		

Index of parameters: L

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

lastChairmanLeavesDisconnect (template) string Defines whether conferences based on this template disconnect guests when the last chairperson leaves. One of **true**, **false**, or **default**.

Corresponds to the "When only guests remain" conference setting in the web UI.

Value	Description
true	Disconnect all participants after the last chairman leaves the conference
false	Take no action when only guests remain in the conference
default	Inherit this setting from the parent template

lastChairmanLeavesDisconnect (template) is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

lastChairmanLeavesDisconnect boolean Defines whether or not this conference disconnects guests when the last chairperson leaves. Corresponds to the **When only guests remain** conference setting in the web UI.

lastChairmanLeavesDisconnect is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

lastRevision integer This number identifies an earlier set of enumeration data to compare against your current call. If you supply this parameter using the **currentRevision** value returned by a previous enumeration, the current **enumerate** call will return only the differences since that previous call. If you don't supply this parameter, the device assumes that you want a full enumeration.

lastRevision is used in: [autoAttendant.enumerate \[p.33\]](#), [conference.enumerate \[p.46\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#),

layoutControlDefault boolean **true** means the endpoint inherits the default layout control setting.

Value	Description
true	The endpoint uses the layout control setting of the conference or template.
false	The endpoint does not use the layout control setting of the conference or template.

layoutControlDefault is used in: [addressBookEntry.enumerate \[p.24\]](#).

layoutControlEnabled boolean Deprecated by **layoutControlEx**. Defines whether the endpoint's participant will have control over the layout if **layoutControlDefault** is **false**.

Note: This parameter is deprecated by `layoutControlEx`.

Indicates whether the participant will have control over their layout. Only present if `layoutControlDefault` is `false`.

`layoutControlEnabled` is `false` if `layoutControlEx` is disabled, but `true` for any other value of `layoutControlEx`.

Value	Description
true	The participant may change the layout on their endpoint.
false	The participant may not change the layout on their endpoint.

`layoutControlEnabled` is used in: [addressBookEntry.enumerate \[p.24\]](#), [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

`layoutControlEx (template)` string Defines how the view layout can be controlled. One of `disabled`, `feccOnly`, `dtmfOnly`, `feccWithDtmfFallback`, or `bothFeccAndDtmf`, or `default`.

Value	Description
disabled	Layout control is disabled
feccOnly	Layout control via FECC only
dtmfOnly	Layout control via DTMF only
feccAndDtmf	Deprecated by <code>feccWithDtmfFallback</code>. Layout control via FECC or via DTMF if FECC is unavailable. This option is no longer supported; use <code>feccWithDtmfFallback</code> instead.
feccWithDtmfFallback	Layout control via FECC when it is available and via DTMF for endpoints which don't have FECC
bothFeccAndDtmf	Layout control via FECC and via DTMF
default	Inherit this setting from the parent template

`layoutControlEx (template)` is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

`layoutControlEx` string Defines how the view layout can be controlled. One of `disabled`, `feccOnly`, `dtmfOnly`, `feccWithDtmfFallback`, or `bothFeccAndDtmf`.

Value	Description
disabled	Layout control is disabled
default	Inherit layout control setting

Value	Description
feccOnly	Layout control via FECC only
dtmfOnly	Layout control via DTMF only
feccAndDtmf	Deprecated by <code>feccWithDtmfFallback</code>. Layout control via FECC or via DTMF if FECC is unavailable. This option is no longer supported; use <code>feccWithDtmfFallback</code> instead.
feccWithDtmfFallback	Layout control via FECC when it is available and via DTMF for endpoints which don't have FECC
bothFeccAndDtmf	Layout control via FECC and via DTMF

`layoutControlEx` is used in: [addressBookEntry.enumerate \[p.24\]](#), [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

`layoutSource` string Describes the reason for the current layout, and is only present if `currentLayout` is present. One of `familyx`, `conferenceCustom`, or `participantCustom`.

Value	Description
familyx	Current layout is determined by the layout family.
conferenceCustom	The current layout is a custom layout set for the conference.
participantCustom	The current layout is a custom layout set for the participant.

`layoutSource` is used in: [conference.streaming.query \[p.72\]](#), [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

`lecturer` boolean `true` if the participant is the lecturer.

`lecturer` is used in: [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

`linkLocalIpv6Address` string(63) The link local IPv6 address in CIDR format.

`linkLocalIpv6Address` is used in: [device.network.query \[p.84\]](#).

`linkLocalIpv6PrefixLength` integer Length of the link local IPv6 address prefix.

`linkLocalIpv6PrefixLength` is used in: [device.network.query \[p.84\]](#).

`linkStatus` boolean `true` if the ethernet connection to this port is active.

`linkStatus` is used in: [device.network.query \[p.84\]](#).

`linkType` string This parameter is ignored unless `participantType` is `by_address`. Either `cascadeSlaveToMaster` or `default`

`linkType` is used in: [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

lipSyncDelayApplied	integer	The amount of delay added to either audio or video output stream to correct for rtcpLipSyncDelay reported between incoming audio and video streams.
----------------------------	---------	--

lipSyncDelayApplied is used in: [participant.statistics \[p.133\]](#).

locked	boolean	Defines whether or not the conference is locked.
---------------	---------	--

locked is used in: [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#),

log	array	Each member of the array contains log information (called system log in the user interface).
------------	-------	--

log is used in: [device.restartlog.query \[p.90\]](#).

Index of parameters: M

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

macAddress	string	The MAC address of this interface. A 12 character string of hex digits with no separators.
macAddress is used in: device.network.query [p.84] .		
maxBitRateFromMCU	integer	Maximum bandwidth from the MCU (kbps).
maxBitRateFromMCU is used in: conferenceme.modify [p.75] , conferenceme.query [p.76] , participant.add [p.103] , participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] , participant.modify [p.127] , participant.status [p.139] , participant.status (deprecated) [p.147] .		
maxBitRateToMCU	integer	Maximum bandwidth to the MCU (kbps).
maxBitRateToMCU is used in: conferenceme.modify [p.75] , conferenceme.query [p.76] , participant.add [p.103] , participant.enumerate [p.112] , participant.enumerate (deprecated) [p.121] , participant.modify [p.127] , participant.status [p.139] , participant.status (deprecated) [p.147] .		
maxConferenceSize	integer	The maximum number of participants that can be hosted in a single conference at the time of the response.
maxConferenceSize is used in: device.query [p.87] .		
maximumAudioPorts	integer	The maximum number of audio-only ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
maximumAudioPorts is used in: conference.create [p.39] , conference.enumerate [p.46] , conference.modify [p.57] , conference.status [p.66] , template.modify [p.172] , template.enumerate [p.168] , template.create [p.163] , template.status [p.176] .		
maximumVideoPorts	integer	The maximum number of video ports for the conference if it is not in port reservation mode. May not be returned in responses unless it is explicitly configured.
maximumVideoPorts is used in: conference.create [p.39] , conference.enumerate [p.46] , conference.modify [p.57] , conference.status [p.66] , template.modify [p.172] , template.enumerate [p.168] , template.create [p.163] , template.status [p.176] .		
maxMediaRxBitRate	integer	The maximum media reception speed of this device, in kbps. 0 means the device uses the default.
maxMediaRxBitRate is used in: addressBookEntry.enumerate [p.24] , gateway.enumerate [p.102] .		
maxMediaTxBitRate	integer	The maximum media transmission speed from this device, in kbps. 0 means the device uses the default.
maxMediaTxBitRate is used in: addressBookEntry.enumerate [p.24] , gateway.enumerate [p.102] .		
maxOcsBitrate	integer	The bitrate to use for ocs and lcs clients, in bits per second. Accepts 64000, 128000, 192000, 256000, 320000, 384000, 512000, 768000, 1000000, 1250000, 1500000, 1750000, 2000000, 2500000, 3000000, 3500000, or 4000000. Set this to 0 to disable the limit.

maxOcsBitrate is used in: [sip.modify \[p.159\]](#), [sip.query \[p.160\]](#).

maxParticipants integer The maximum number of ConferenceMe connections allowed.

maxParticipants is used in: [conferenceme.modify \[p.75\]](#), [conferenceme.query \[p.76\]](#).

maxVideoResolution string Either `cif` or `4cif`.

Value	Description
<code>cif</code>	The maximum video resolution is 352 x 288
<code>4cif</code>	The maximum video resolution is 704 x 576

maxVideoResolution is used in: [device.query \[p.87\]](#).

mcuServicePrefix string The service prefix used by the MCU.

mcuServicePrefix is used in: [gatekeeper.modify \[p.98\]](#), [gatekeeper.query \[p.100\]](#).

mcuServicePrefixStatus string The current status of the service prefix registration process.

Value	Description
<code>idle</code>	
<code>registering</code>	
<code>registered</code>	
<code>deregistering</code>	
<code>pendingReregistration</code>	
<code>waitingRetry</code>	
<code>noID</code>	
<code>idTooLong</code>	

mcuServicePrefixStatus is used in: [gatekeeper.query \[p.100\]](#).

mediaEncryption string One of `encrypted`, `unencrypted`, `mixed`, or `unknown`.

Value	Description
<code>encrypted</code>	All media channels to and from this endpoint are encrypted.
<code>unencrypted</code>	All media channels to and from this endpoint are unencrypted.
<code>mixed</code>	Some of the media channels to or from this endpoint are encrypted.
<code>unknown</code>	None of the above; this may occur when a participant has very recently connected and media channels have not yet been established.

mediaEncryption is used in: [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

mediaLoad	integer	A percentage value representing the proportion of the device's media processing capacity that is currently in use. mediaLoad is used in: device.health.query [p.81] .
mediaOverTcp	boolean	true allows ConferenceMe to fall back to media over TCP if it cannot do media over UDP. mediaOverTcp is used in: conferenceme.modify [p.75] , conferenceme.query [p.76] .
mediaResources	integer	The percentage of DSP resources that are available (i.e. successfully booted and not failed) to the unclustered device or the master blade of a cluster. Slave blades don't return this value. mediaResources is used in: device.query [p.87] .
message	string (255)	The string to send to the participant. message is used in: participant.message [p.126] .
metadata	string (4095)	A string of up to 4095 unicode characters stored on the device and associated with the named conference. metadata is used in: conference.metadata.modify [p.55] , conference.metadata.status [p.56] .
minFrameRateMotionSharpness	integer	Specifies the minimum frame rate for this endpoint. This parameter is only present if useDefaultMotionSharpness is false . minFrameRateMotionSharpness is used in: addressBookEntry.enumerate [p.24] , gateway.enumerate [p.102] .
model	string	The model number. model is used in: device.query [p.87] .
moreThanFour	boolean	Enables the call to return more than four conferences (up to 24). moreThanFour is used in: conference.enumerate [p.46] .
motionSharpnessTradeoff	string	Defines preference for motion vs. sharpness. One of preferMotion , preferSharpness , balanced , or default .

Value	Description
default	Use the global default setting.
preferMotion	Prefer motion at the expense of sharpness.
preferSharpness	Prefer sharpness at the expense of motion.
balanced	Try to balance the motion and sharpness trade-off.

motionSharpnessTradeoff is used in: [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

multicast	boolean	Defines whether or not multicast streaming is enabled for this format.
------------------	---------	--

multicast is used in: [streaming.modify \[p.161\]](#), [streaming.query \[p.162\]](#).

multicastPacketsReceived	integer	Number of multicast packets received on this Ethernet interface.
---------------------------------	---------	--

multicastPacketsReceived is used in: [device.network.query \[p.84\]](#).

multicastPacketsSent	integer	Number of multicast packets sent from this Ethernet interface.
-----------------------------	---------	--

multicastPacketsSent is used in: [device.network.query \[p.84\]](#).

multicastStreamingEnabled	boolean	Defines whether or not the conference can be multicast.
----------------------------------	---------	---

multicastStreamingEnabled is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

multicastViewers	integer	The count of multicast streaming viewers.
-------------------------	---------	---

multicastViewers is used in: [conference.streaming.query \[p.72\]](#).

Index of parameters: N

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

name (endpoint) string The name of the endpoint.

name (endpoint) is used in: [addressBookEntry.enumerate \[p.24\]](#).

name (gateway) string The name of the gateway.

name (gateway) is used in: [gateway.enumerate \[p.102\]](#).

name (service) string The name of the service. One of the following:
 TCP services: **http**, **https**, **ftp**, **h225**, **rtsp**, **mms**, **sip_tcp**, **sips_tcp**, **cdep**
 UDP services: **sip_udp**, **snmp**, **gatekeeper**, **tunnel**

Service name	Comments
http	
https	
ftp	
h225	Not supported on slaves.
rtsp	Not supported on slaves.
mms	Not supported on slaves.
sip_tcp	Not supported on slaves.
sips_tcp	Not supported on slaves.
cdep	Requires ConferenceMe activation. Not supported on slaves.
sip_udp	Not supported on slaves.
snmp	
gatekeeper	Not supported on slaves.
tunnel	Requires ConferenceMe activation. Not supported on slaves.

name (service) is used in: [services.modify \[p.156\]](#), [services.query \[p.158\]](#).

nameServer string (79) The IP address of the name server, in dotted quad format (IPv4) or CIDR format (IPv6).

nameServer is used in: [device.network.modify \[p.82\]](#), [device.network.query \[p.84\]](#).

nameServerSecondary string (79) The IP address of the secondary name server, in dotted quad format (IPv4) or CIDR format (IPv6).

nameServerSecondary is used in: [device.network.modify \[p.82\]](#), [device.network.query \[p.84\]](#).

newConferenceName string The new conference name.

This parameter changes the name of the conference when it is supplied as a parameter to `conference.modify`, so must be unique in that context.

When it is supplied as a parameter to `participant.move`, it is interpreted as the destination for the moved participant and should be an existing conference name.

`newConferenceName` is used in: [conference.modify \[p.57\]](#), [participant.move \[p.131\]](#).

<code>newParticipantsCustomLayout</code>	boolean	<code>true</code> if new participants use the custom layout, <code>false</code> otherwise. Only valid if <code>customLayoutEnabled</code> is <code>true</code> .
--	---------	--

`newParticipantsCustomLayout` is used in: [conference.create \[p.39\]](#), [conference.modify \[p.57\]](#).

<code>newRouteId</code>	integer	A number selected by the device to identify the newly added route. Pass this parameter as <code>routeId</code> to any calls that require identification of the new route.
-------------------------	---------	---

`newRouteId` is used in: [route.add \[p.151\]](#).

<code>newTemplateName</code>	string	Use this parameter to change the name of the template. The call will return an error if another template exists that has this name.
------------------------------	--------	---

`newTemplateName` is used in: [template.modify \[p.172\]](#).

<code>nextIndex</code>	integer	Revision number of the data being provided, reusable in a subsequent call to the API.
------------------------	---------	---

`nextIndex` is used in: [cdrlog.enumerate \[p.36\]](#).

<code>ntpEnabled</code>	boolean	Defines whether or not the device may synchronize with an NTP server.
-------------------------	---------	---

`ntpEnabled` is used in: [device.time.modify \[p.92\]](#), [device.time.query \[p.93\]](#).

<code>ntpHost</code>	string	DNS or IP address of an NTP server
----------------------	--------	------------------------------------

`ntpHost` is used in: [device.time.modify \[p.92\]](#), [device.time.query \[p.93\]](#).

<code>ntpStatus</code>	string	The NTP client's current status; one of <code>disabled</code> , <code>synchronizing</code> , <code>synchronized</code> or <code>error</code> .
------------------------	--------	--

`ntpStatus` is used in: [device.time.query \[p.93\]](#).

<code>numberOfRepeats</code>	integer	Defines the number of times the conference repeats. Required if <code>terminationType</code> is set to <code>afterNRepeats</code> .
------------------------------	---------	---

`numberOfRepeats` is used in: [conference.create \[p.39\]](#), [conference.modify \[p.57\]](#).

<code>numericId</code>	string	The numeric ID of the conference. Used for registration with H.323 gatekeeper / SIP registrar, and to dial in to the conference.
------------------------	--------	--

`numericId` is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#),

<code>numEvents (audit log)</code>	integer	The total number of events stored.
------------------------------------	---------	------------------------------------

numEvents (*audit log*) is used in: [auditlog.query \[p.31\]](#).

numEvents (<i>CDR log</i>)	integer	The difference between the index numbers of the most recent record and the oldest record, irrespective of whether or not the intervening records have been permanently stored.
-------------------------------------	---------	--

numEvents (*CDR log*) is used in: [cdrlog.query \[p.38\]](#).

numEvents (<i>per enumeration</i>)	integer	Specifies maximum number of events to be returned per enumeration. If omitted (or not between 1 - 20 inclusive), a maximum of 20 events will be returned per enumeration.
---	---------	---

numEvents (*per enumeration*) is used in: [cdrlog.enumerate \[p.36\]](#).

Index of parameters: O

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

oldConferenceName string Deprecated conference renaming scheme - new code should use conferenceName and newConferenceName as above.

oldConferenceName is used in: [conference.modify \[p.57\]](#).

operationalStatus string One of **active**, **shuttingDown**, or **shutdown**.

operationalStatus is used in: [device.health.query \[p.81\]](#).

operationScope string Either of the strings **activeState** or **configuredState**.

operationScope array The array should contain one or two string parameters. That is, it should contain either or both of the strings **currentState** or **configuredState**.

The operationScope parameter takes either a string or an array of strings, depending on whether you are reading or setting the participant parameters. In the participant.modify sense, operationScope is a string parameter that accepts either activeState or configuredState; you can only modify the participant's parameters for one of those scopes. In the participant.status and participant.enumerate senses, operationScope accepts an array because you can read the currentState and configuredState parameters in the same call.

Value	Description
activeState	The operation scope is limited to the active configuration of the participant.
currentState	The operation scope is limited to the active configuration of the participant.
configuredState	The operation scope is limited to the stored configuration of the participant.
Both activeState and configuredState	The scope is not limited to either state. That is, the participant structure will contain a currentState and configuredState structure, but the structures may be empty if the endpoints are not active or preconfigured, respectively.

operationScope is used in: [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

outgoingTransport string The outgoing transport protocol. One of **udp**, **tcp**, or **tls**.

outgoingTransport is used in: [sip.modify \[p.159\]](#), [sip.query \[p.160\]](#).

Index of parameters: P

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

packetLossCritical boolean This will be true if any packet loss above a certain level (5%) is seen within the last five seconds.

packetLossCritical is used in: [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

packetLossWarning boolean This will be true if any packet loss has been seen within the last 15 seconds.

packetLossWarning is used in: [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

packetsErrors integer Count of packets lost from a received audio, video, or content stream. Deprecates **audioRxLost**, **videoRxLost** and **contentRxLost**.

packetsErrors is used in: [participant.statistics \[p.133\]](#).

packetsReceived integer The number of packets received on this Ethernet port.

packetsReceived is used in: [device.network.query \[p.84\]](#).

packetsSent integer The number of packets sent from this Ethernet port.

packetsSent is used in: [device.network.query \[p.84\]](#).

packetsTransferred integer The count of packets transferred in a particular stream. Applies to audio, video, and content streams to and from the device. Deprecates **audioRxReceived**, **videoRxReceived**, **contentRxReceived**, **videoTxSent** and **contentTxSent**.

packetsTransferred is used in: [participant.statistics \[p.133\]](#).

panes array An array of **structs**, each of which defines a particular pane within the layout.

panes is used in: [conference.paneplacement.modify \[p.61\]](#), [conference.paneplacement.query \[p.63\]](#).

panesModified integer The number of panes successfully modified. This will be the number of elements in the panes array on complete success, and zero if there is no panes array.

panesModified is used in: [conference.paneplacement.modify \[p.61\]](#).

parent string The name of the parent template. Defaults to Top Level template if omitted.

parent is used in: [template.create \[p.163\]](#), [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.status \[p.176\]](#).

participant struct Contains the parameters that, when considered together, uniquely identify a participant.

participant is used in: [participant.add \[p.103\]](#), [participant.statistics \[p.133\]](#).

participantName string The unique name of a participant.

An **ad_hoc** participant contains its automatically assigned global participant index in place of a **participantName**; the MCU ignores the **participantName** if you supply it for this **participantType**.

participantName is used in: [conference.enumerate \[p.46\]](#), [conference.floor.modify \[p.53\]](#), [conference.floor.query \[p.54\]](#), [conference.paneplacement.modify \[p.61\]](#), [conference.paneplacement.query \[p.63\]](#), [conference.status \[p.66\]](#), [conference.streaming.modify \[p.71\]](#), [conference.streaming.query \[p.72\]](#), [participant.add \[p.103\]](#), [participant.connect \[p.107\]](#), [participant.diagnostics \[p.108\]](#), [participant.disconnect \[p.111\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.fecc \[p.125\]](#), [participant.message \[p.126\]](#), [participant.modify \[p.127\]](#), [participant.move \[p.131\]](#), [participant.remove \[p.132\]](#), [participant.statistics \[p.133\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

participantProtocol string **h323**, **sip**, or **vnc**.

participantProtocol is used in: [conference.enumerate \[p.46\]](#), [conference.floor.modify \[p.53\]](#), [conference.floor.query \[p.54\]](#), [conference.paneplacement.modify \[p.61\]](#), [conference.paneplacement.query \[p.63\]](#), [conference.status \[p.66\]](#), [conference.streaming.modify \[p.71\]](#), [conference.streaming.query \[p.72\]](#), [participant.add \[p.103\]](#), [participant.connect \[p.107\]](#), [participant.diagnostics \[p.108\]](#), [participant.disconnect \[p.111\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.fecc \[p.125\]](#), [participant.message \[p.126\]](#), [participant.modify \[p.127\]](#), [participant.move \[p.131\]](#), [participant.remove \[p.132\]](#), [participant.statistics \[p.133\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

participants array An array of structures that represent participants.

participants is used in: [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#),

participantType string One of: **by_address**, **by_name**, or **ad_hoc**.

Value	Description
ad_hoc	The participant may have joined the conference by dialing in, by being dialed directly via the web interface, or by the API.
by_address	The participant was added to the conference via the API. API-created participants in scheduled conferences (i.e. those originated by participant.add will be of type by_address unless they are explicitly added as temporary ad_hoc participants.
by_name	The participant's endpoint is in the MCU's endpoint list. The endpoint was added to the conference's configuration as a pre-configured participant, using the web interface.

participantType is used in: [conference.enumerate \[p.46\]](#), [conference.floor.modify \[p.53\]](#), [conference.floor.query \[p.54\]](#), [conference.paneplacement.modify \[p.61\]](#), [conference.paneplacement.query \[p.63\]](#), [conference.status \[p.66\]](#), [conference.streaming.modify \[p.71\]](#), [conference.streaming.query \[p.72\]](#), [participant.add \[p.103\]](#), [participant.connect \[p.107\]](#), [participant.diagnostics \[p.108\]](#), [participant.disconnect \[p.111\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.fecc \[p.125\]](#), [participant.message \[p.126\]](#), [participant.modify \[p.127\]](#), [participant.move \[p.131\]](#), [participant.remove \[p.132\]](#), [participant.statistics \[p.133\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#),

password (SIP) string (63) The password used for SIP registration.

password (SIP) is used in: [sip.modify \[p.159\]](#).

password string The password for VNC endpoints.

password is used in: [addressBookEntry.enumerate \[p.24\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

password (*gatekeeper*) string The password that the device uses to register with the gatekeeper, if required.

password (*gatekeeper*) is used in: [gatekeeper.modify \[p.98\]](#).

pendingRegistrations integer The number of registrations in progress

pendingRegistrations is used in: [gatekeeper.query \[p.100\]](#).

percentageCapacity integer The percentage of the total available capacity being used by the log.

percentageCapacity is used in: [auditlog.query \[p.31\]](#), [cdrlog.query \[p.38\]](#).

pin string The PIN for this conference. A string of numeric digits that must be entered to access the conference.

pin is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#),

port (*IP*) integer Identifies the IP port.

port (*service*) is used in: [participant.statistics \[p.133\]](#), [services.modify \[p.156\]](#), [services.query \[p.158\]](#).

port (*Ethernet*) string Identifies the Ethernet port. May be A or B.

port is used in: [route.enumerate \[p.153\]](#), [services.modify \[p.156\]](#), [services.query \[p.158\]](#).

portA struct A structure that contains configuration and status information for Ethernet port A on the device.

portA is used in: [device.network.modify \[p.82\]](#), [device.network.query \[p.84\]](#).

portAssociationA boolean **true** if interface 'PortA IPv4' is associated with the H.323 gatekeeper.

portAssociationA is used in: [gatekeeper.query \[p.100\]](#).

portAssociationAv4 boolean **true** if interface 'PortA IPv4' is associated with the H.323 gatekeeper.

portAssociationAv4 is used in: [gatekeeper.modify \[p.98\]](#), [gatekeeper.query \[p.100\]](#).

portAssociationAv6 boolean **true** if interface 'PortA IPv6' is associated with the H.323 gatekeeper.

portAssociationAv6 is used in: [gatekeeper.modify \[p.98\]](#), [gatekeeper.query \[p.100\]](#).

portAssociationB boolean **true** if interface 'PortB IPv4' is associated with the H.323 gatekeeper.

portAssociationB is used in: [gatekeeper.query \[p.100\]](#).

portAssociationBv4 boolean **true** if interface 'PortB IPv4' is associated with the H.323 gatekeeper.

portAssociationBv4 is used in: [gatekeeper.modify \[p.98\]](#), [gatekeeper.query \[p.100\]](#).

portAssociationBv6 boolean **true** if interface 'PortB IPv6' is associated with the H.323 gatekeeper.

portAssociationBv6 is used in: [gatekeeper.modify \[p.98\]](#), [gatekeeper.query \[p.100\]](#).

portB struct A structure that contains configuration and status information for Ethernet port B on the device.

portB is used in: [device.network.modify \[p.82\]](#), [device.network.query \[p.84\]](#).

portNumber integer The port number for VNC endpoints.

portNumber is used in: [addressBookEntry.enumerate \[p.24\]](#).

portRangeFinish integer The last port number in the multicast port range.

portRangeFinish is used in: [streaming.modify \[p.161\]](#), [streaming.query \[p.162\]](#).

portRangeStart integer The first port number in the multicast port range.

portRangeStart is used in: [streaming.modify \[p.161\]](#), [streaming.query \[p.162\]](#).

portReservationMode string Defines whether port reservation mode is **enabled** or **disabled**. Corresponds to the Media port reservation setting on the web interface. Only present on MCU products.

portReservationMode is used in: [device.query \[p.87\]](#).

ports array An array whose members are structures representing the Ethernet ports on the device

ports is used in: [services.modify \[p.156\]](#), [services.query \[p.158\]](#).

preconfiguredParticipantsDefer string Defines whether conferences based on this template defer inviting preconfigured participants until at least one other participant is present. One of **true**, **false**, or **default**.

Corresponds to the "Invite preconfigured participants" conference setting in the web UI.

Value	Description
true	The MCU defers inviting preconfigured participants until at least one other participant is present
false	The MCU invites preconfigured participants as soon as the conference starts
default	Inherit this setting from the parent template

preconfiguredParticipantsDefer (*template*) is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

preconfiguredParticipantsDefer boolean **true** if the MCU defers inviting preconfigured participants until at least one other participant is present.

Corresponds to the "Invite preconfigured participants" conference setting in the web UI.

Value	Description
true	The MCU defers inviting preconfigured participants until at least one other participant is present.
false	The MCU invites preconfigured participants as soon as the conference starts.

`preconfiguredParticipantsDefer` is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

`prefixLength` integer The prefix length of the destination IP range for this route (the number of fixed bits in the address).

`prefixLength` is used in: [route.add \[p.151\]](#), [route.enumerate \[p.153\]](#).

`previewURL` string The location of the preview image; this is not a complete URL, and requires a prefix of `http://hostname` (where `hostname` is the hostname of this MCU) before it is used.

`previewURL` is used in: [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

`private (template)` string Defines whether or not conferences based on this template are private. One of `true`, `false`, or `default`.

Determines the visibility of conferences based on this template. This parameter corresponds to the "Visibility" setting on the web UI, which can have the value Public or Private.

Value	Description
true	Conferences based on this template are Private
false	Conferences based on this template are Public
default	Inherit this setting from the parent template

`private (template)` is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

`private` boolean Defines whether the conference is public or private. `true` if the conference is private. Corresponds to the **Visibility** setting on the web UI, which can have the value *Public* or *Private*.

`private` is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

`protocol (IP)` string **IPv4** or **IPv6**.

`protocol (IP)` is used in: [services.modify \[p.156\]](#), [services.query \[p.158\]](#).

`protocol (signaling)` string The signaling protocol used in the call. One of `h323`, `sip`, or `vnc`.

`protocol` is used in: [addressBookEntry.enumerate \[p.24\]](#),

Index of parameters: Q

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

queueDrops

integer

Number of packets dropped from the queue on this network interface.

queueDrops is used in: [device.network.query \[p.84\]](#).

Index of parameters: R

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

reason	string	<p>An explanation for the restart. One of:</p> <ul style="list-style-type: none"> ▪ User requested shutdown ▪ User requested reboot from web interface ▪ User requested upgrade ▪ User requested reboot from console ▪ User requested reboot from API ▪ User requested reboot from FTP ▪ User requested shutdown from supervisor ▪ User requested reboot from supervisor ▪ User reset configuration ▪ Cold boot ▪ unknown
---------------	--------	---

reason is used in: [device.restartlog.query \[p.90\]](#).

rebootRequired	boolean	The device returns this parameter as true if it needs to reboot.
-----------------------	---------	---

The device will signal that it needs a reboot under the following circumstances:

- new loader
- new main image
- certificate manager needs restart
- product modifier pending

rebootRequired is used in: [device.query \[p.87\]](#).

receiveErrors	integer	The count of receive errors on this interface.
----------------------	---------	--

receiveErrors is used in: [device.network.query \[p.84\]](#).

receiverIndex	integer	A number between 1 and 20 defining the position of this feedback receiver in the device's table of feedback receivers.
----------------------	---------	--

receiverIndex is used in: [feedbackReceiver.configure \[p.94\]](#), [feedbackReceiver.reconfigure \[p.96\]](#), [feedbackReceiver.remove \[p.97\]](#).

receivers	array	An array of feedback receivers, with members corresponding to the entries in the receivers table on the device's web interface.
------------------	-------	---

receivers is used in: [feedbackReceiver.query \[p.95\]](#).

receiverURI string Fully-qualified URI that identifies the listening application's XML-RPC interface (protocol, address, and port), for example, `http://tms1:8080/RPC2`. Must end in `/RPC2` (see [XML-RPC.com](#)). You can use `http` or `https` and, if no port number is specified, the device will use the protocol defaults (80 and 443 respectively).

receiverURI is used in: [feedbackReceiver.configure \[p.94\]](#), [feedbackReceiver.query \[p.95\]](#), [feedbackReceiver.reconfigure \[p.96\]](#),

redial string Defines the MCU's redial behavior when calls out to this participant drop. One of `never`, `connect`, `unexpected`, `any`, or `default`.

Value	Description
never	The MCU never tries to redial this participant. It only tries to connect the call once. This is the default value for participants of type <code>ad_hoc</code> .
connect	The MCU redials this participant until the connection is established. After that initial connection, it does not attempt to redial when the connection drops.
unexpected	The MCU redials this participant until the connection is made, and also on unexpected drops thereafter.
any	The MCU redials this participant until the connection is made, and also on any drops thereafter. This includes the participant deliberately ending the call.
default	The participant's <code>redial</code> inherits the value from the MCU-wide setting, as configured on the Settings > Conference page of the web interface, when the participant joins. <code>default</code> is therefore only possible for the configured state of the participant. This value is the default for participants of type <code>by_address</code> .

redial is used in: [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#).

redialLimit string Defines whether a redial limit is used with the redial behavior. One of `enabled`, `disabled`, or `default`.

The **redialLimit** only applies when **redial** is `connect`, `unexpected`, or `any`.

Value	Description
enabled	<p>The MCU follows a limited schedule when it redials a participant.</p> <p>If the call has connected but is deliberately ended by the far end, it waits 30 seconds before the first reconnection attempt (only applies when redial is any). In other cases, the MCU redials as soon as it knows the call has failed.</p> <p>Assuming the call continues failing, the MCU attempts to reconnect once per minute for four attempts after the first. If the call still fails to connect, the MCU continues trying once every five minutes for five more attempts before stopping. In total, it tries to re-establish the connection ten times in a half hour period.</p> <p>This value is the default for participants of type ad_hoc.</p>
disabled	<p>The MCU follows an unlimited schedule when it redials a participant.</p> <p>It follows the limited schedule described above for the first ten attempts; if the call continues failing after that, the MCU redials once every five minutes thereafter, indefinitely, until either the conference or the participant is no longer active.</p>
default	<p>The participant's redialLimit inherits the value from the MCU-wide setting, as configured on the Settings > Conference page of the web interface, when the participant joins.</p> <p>default is therefore only possible for the configured state of the participant.</p> <p>This value is the default for participants of type by_address.</p>

redialLimit is used in: [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#).

registeredAddress	string	The IP address and port that the MCU has registered with the gateway. This value is only returned if the MCU is registered.
--------------------------	--------	---

registeredAddress is used in: [gatekeeper.query \[p.100\]](#).

registerWithGatekeeper (<i>template</i>)	string	Defines whether or not the conferences based on this template register their numericIds with the H.323 gatekeeper. One of true , false , or default (inherit this setting from the parent template).
---	--------	--

registerWithGatekeeper (*template*) is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

registerWithGatekeeper	boolean	Defines whether or not this conference registers its numericId with the H.323 gatekeeper.
-------------------------------	---------	--

registerWithGatekeeper is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#),

registerWithSIPRegistrar (<i>template</i>)	string	Defines whether conferences based on this template register with the SIP registrar. One of true , false , or default (inherit this setting from the parent template).
---	--------	--

registerWithSIPRegistrar (*template*) is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#),

[template.create \[p.163\]](#), [template.status \[p.176\]](#).

registerWithSIPRegistrar boolean Defines whether or not this conference registers its **numericId** with the SIP registrar.

registerWithSIPRegistrar is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#),

registrarContactDomain string This value is generated from the **registrarContactURI** (**Username** in the web interface) and the **configuredRegistrar** (**SIP Registrar domain** in web interface.)

registrarContactDomain is used in: [sip.query \[p.160\]](#).

registrarContactURI string (255) The URI provided to the SIP registrar to register this device. Corresponds to the **Username** setting on the **Settings > SIP** web page.

registrarContactURI is used in: [sip.modify \[p.159\]](#), [sip.query \[p.160\]](#).

registrarType string (10) The type of SIP registrar. Either **normal** or **lcs**.

registrarType is used in: [sip.modify \[p.159\]](#), [sip.query \[p.160\]](#).

registrarUsage boolean Defines whether or not SIP registrar usage is enabled.

registrarUsage is used in: [sip.modify \[p.159\]](#), [sip.query \[p.160\]](#).

registrationPrefix string (255) A string of digits that serves as the device's registration prefix.

registrationPrefix is used in: [gatekeeper.modify \[p.98\]](#), [gatekeeper.query \[p.100\]](#).

registrationStatus string The SIP registration status. One of **registering**, **registered**, **unregistered**, or **unknown**.

registrationStatus is used in: [sip.query \[p.160\]](#).

registrationType string The gatekeeper registration type. One of **gateway**, **terminalGateway**, **gatewayCisco**, **mcuStandard**, or **mcuCompatible**.

The value of the "Gatekeeper registration type" setting as seen on **Settings > H.323** web UI page.

Value	Description
terminalGateway	Corresponds to <i>Terminal / gateway</i> on the web UI.
gateway	Corresponds to <i>Gateway</i> on the web UI.
gatewayCisco	Corresponds to <i>Gateway (Cisco GK compatible)</i> .
mcuStandard	Corresponds to <i>MCU (Standard)</i> .
mcuCompatible	Corresponds to <i>MCU (Compatible)</i> .

registrationType is used in: [gatekeeper.modify \[p.98\]](#), [gatekeeper.query \[p.100\]](#).

remoteLinkType string One of `slave`, `conference`, `autoAttendant`, `recording`, or `playback`.

remoteLinkType is used in: [participant.enumerate \[p.112\]](#), [participant.status \[p.139\]](#).

repetition string Defines the repetition frequency of a scheduled conference. One of `none`, `daily`, `weekly`, `everyTwoWeeks`, or `monthly`.

Value	Description
<code>none</code>	The conference does not repeat.
<code>daily</code>	The conference repeats every day at the given <code>startTime</code> .
<code>weekly</code>	The conference repeats at least once per week, at the given <code>startTime</code> on the given <code>weekDays</code> .
<code>everyTwoWeeks</code>	The conference repeats at least once every two weeks, at the given <code>startTime</code> on the given <code>weekDays</code> .
<code>monthly</code>	The conference repeats once a month, at the given <code>startTime</code> on a given <code>weekDay</code> in the given week of the month (<code>whichWeek</code>).

repetition is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

reserveAudioPorts boolean Determines if the template should have a value for the reserved audio ports setting. Has no effect if the request sets `usePortsFromParent` to `true`.

reserveAudioPorts is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

reservedAudioPorts integer The number of audio only ports to reserve for a conference if in port reservation mode.

If the value of the `reservedAudioPorts` parameter exceeds the total number of available audio ports, the MCU will reserve all available audio ports and reserve video ports for the remainder.

For example, if the MCU has 20 video and 20 audio only ports and a request is made to reserve 30 audio only ports, the MCU will reserve 20 audio only ports and 10 video ports.

reservedAudioPorts is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#), [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

reservedVideoPorts integer The number of video ports to reserve for a conference if in port reservation mode.

reservedVideoPorts is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#), [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

reserveVideoPorts boolean Determines whether the template should have a value for the reserved video ports setting. Has no effect if the request sets `usePortsFromParent` to `true`.

reserveVideoPorts is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#),

[template.status \[p.176\]](#).

resourceAvailabilityStatus string Indicates the availability of resources on the MCU. One of **available**, **unavailable**, or **disabled** (resource availability indications are not enabled).

resourceAvailabilityStatus is used in: [gatekeeper.query \[p.100\]](#).

restartTime dateTime. The date and time when the system was last restarted.
iso8601

restartTime is used in: [device.query \[p.87\]](#).

routeId integer A number that identifies a route. The device assigns a number to each manually configured route.

routeId is used in: [route.delete \[p.152\]](#), [route.enumerate \[p.153\]](#).

rtcBatteryStatus string The current status of the RTC battery (Real Time Clock). One of **ok**, **outOfSpec** (the battery is operating outside of the normal range, and may require service), or **critical**.

rtcBatteryStatus is used in: [device.health.query \[p.81\]](#).

rtcBatteryStatusWorst string The worst recorded status of the RTC battery. One of **ok**, **outOfSpec** (the battery has operated outside of the normal range at some time since the device was booted), or **critical**.

rtcBatteryStatusWorst is used in: [device.health.query \[p.81\]](#).

rtcpLipSyncDelay integer The reported delay between the incoming audio and video streams from this endpoint.

rtcpLipSyncDelay is used in: [participant.statistics \[p.133\]](#).

rtcpOtherReports integer Count of the RTCP reports seen by the MCU that are neither sender nor receiver reports.

rtcpOtherReports is used in: [participant.statistics \[p.133\]](#).

rtcpPacketLossReported integer The count of media packets reported lost, by the far end, in a receiver report sent to the MCU.

rtcpPacketLossReported is used in: [participant.statistics \[p.133\]](#).

rtcpPacketsSent integer Count of RTCP packets sent by the MCU to this endpoint.

rtcpPacketsSent is used in: [participant.statistics \[p.133\]](#).

rtcpReceiveAddress string Address of the RTCP receiver.

rtcpReceiveAddress is used in: [participant.statistics \[p.133\]](#).

rtcpReceivePort integer Port number used by the receiver to accept RTCP messages.

rtcpReceivePort is used in: [participant.statistics \[p.133\]](#).

rtcpReceiverReports integer Count of the RTCP receiver reports seen by the MCU.

rtcpReceiverReports is used in: [participant.statistics \[p.133\]](#).

rtcpSenderReports integer Count of the RTCP sender reports seen by the MCU.

rtcpSenderReports is used in: [participant.statistics \[p.133\]](#).

rtcpTransmitAddress string The IP address and port to which the MCU is sending RTCP packets about this stream.

rtcpTransmitAddress is used in: [participant.statistics \[p.133\]](#).

rtcpTransmitPort integer Port number used for transmitting RTCP messages to the endpoint. Absent if **rtcpTransmitAddress** is unspecified.

rtcpTransmitPort is used in: [participant.statistics \[p.133\]](#).

Index of parameters: S

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

scheduled	boolean	true if the conference is a scheduled conference (regardless of whether or not it is completed).
scheduled is used in: conference.enumerate [p.46] .		
scheduledConferenceIDRegistration	string (8)	Defines whether or not ID registration is enabled for scheduled conferences. Either enabled or disabled . Corresponds to the ID registration for scheduled conferences option on the web interface.
scheduledConferenceIDRegistration is used in: gatekeeper.modify [p.98] , gatekeeper.query [p.100] .		
selectedBitRate	integer	The selected bit rate for the media stream. Applies to sent and received video and content streams. Deprecates videoRxSelectedBitRate , contentRxSelectedBitRate , videoTxSelectedBitRate , and contentTxSelectedBitRate .
selectedBitRate is used in: participant.statistics [p.133] .		
sendResourceAvailabilityIndications	boolean	Defines whether or not the MCU will send resource availability indications.
sendResourceAvailabilityIndications is used in: gatekeeper.modify [p.98] , gatekeeper.query [p.100] .		
serial	string	The serial number of this device or 'unknown'.
serial is used in: device.query [p.87] .		
services	array	An array whose members represent the services provided on the particular port and protocol.
services is used in: services.modify [p.156] , services.query [p.158] .		
setting	boolean	Defines whether or not this feature is intended to be enabled, irrespective of whether it is actually enabled or requires a feature key.
setting is used in: conferenceme.modify [p.75] , conferenceme.query [p.76] , device.encryption.modify [p.79] , device.encryption.query [p.80] , services.modify [p.156] , services.query [p.158] , streaming.modify [p.161] , streaming.query [p.162] .		
shutdownOnly	boolean	If true , the device will shut down when it receives device.restart and will not restart. Defaults to false .
shutdownOnly is used in: device.restart [p.89] .		
shutdownStatus	string	Indicates the status of a shutdown operation. One of shutdown , shutdownInProgress , or notShutdown .
shutdownStatus is used in: device.query [p.87] .		

sipMediaEncryption	string	Defines whether SIP media is encrypted and, if so, for which transport protocols. One of disabled , allTransports or tlsOnly .
sipMediaEncryption is used in: device.encryption.modify [p.79] , device.encryption.query [p.80] .		
siren14	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.
siren14 is used in: addressBookEntry.enumerate [p.24] .		
softwareVersion	string	The version number of the software running on the device.
softwareVersion is used in: device.query [p.87] .		
sourceIdentifier	string	The originating device uses this parameter to identify itself to the listening receiver/s.
sourceIdentifier is used in: feedbackReceiver.configure [p.94] , feedbackReceiver.query [p.95] , feedbackReceiver.reconfigure [p.96] ,		
speed	integer	Speed of the connection on this Ethernet interface. One of 10, 100 or 1000, in Mbps.
speed is used in: device.network.modify [p.82] , device.network.query [p.84] .		
startIndex	integer	Either the index provided, or if that is lower than the index of the first record the device has, it will be the first record it does know about. In this case, comparing the startIndex with the index provided gives the number of dropped records.
startIndex is used in: cdrlog.enumerate [p.36] .		
startLocked (template)	string	Defines whether conferences based on this template should be locked when they start. One of true , false , or default (inherit this setting from the parent template).
startLocked (template) is used in: template.modify [p.172] , template.enumerate [p.168] , template.create [p.163] , template.status [p.176] .		
startLocked	boolean	Defines whether or not the conference should be locked when it starts. Set true if you want it to start in the locked state.
startLocked is used in: conference.create [p.39] , conference.modify [p.57] .		
startTime	dateTime. iso8601	Start time of the item, e.g. 20110106T14:00:00.
startTime is used in: autoAttendant.enumerate [p.33] , autoAttendant.status [p.34] , conference.create [p.39] , conference.enumerate [p.46] , conference.modify [p.57] , conference.status [p.66] ,		
status (success)	string	Operation successful
status (success) is used in: conference.metadata.modify [p.55] , feedbackReceiver.configure [p.94] , route.add [p.151] , template.create [p.163] , template.modify [p.172] .		

streaming string Specifies the type of streaming to be used on the conference. One of **none**, **unicast**, **multicast**, **unicastAndMulticast**, or **default**.

streaming is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

subnetMask string The IPv4 subnet mask in dotted quad format.

subnetMask is used in: [device.network.query \[p.84\]](#).

subscribedEvents array An array of strings, each of which is the name of a notification event. The array defines the events to which the receiver subscribes.

You may specify any or all of the following:

- cdrAdded
- conferenceStarted
- conferenceFinished
- conferenceActive
- conferenceInactive
- configureAck
- participantJoined
- participantLeft
- participantConnected
- participantDisconnected
- restart

subscribedEvents is used in: [feedbackReceiver.configure \[p.94\]](#), [feedbackReceiver.reconfigure \[p.96\]](#).

suppressAudioDuringDTMF string **outgoing** or **all** defines which audio the MCU suppresses while it sends the DTMF connection sequence to the endpoint.

The MCU suppresses outgoing audio to the endpoint by default, while it is sending the DTMF connection sequence to the endpoint. Use **all** to suppress incoming audio as well - so that other participants don't hear the audio from the endpoint while it is connecting.

suppressAudioDuringDTMF is used in: [addressBookEntry.enumerate \[p.24\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

suppressDtmfEx string Controls the muting of in-band DTMF tones. One of **fecc**, **always**, or **never**.

Value	Description
fecc	In-band DTMF tones are muted when DTMF is being used to control layout because far end camera control (FECC) is not available
always	In-band DTMF tones are always muted
never	In-band DTMF tones are never muted

suppressDtmfEx is used in: [addressBookEntry.enumerate \[p.24\]](#), [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

suppressDtmfEx (*template*) string Controls the muting of in-band DTMF tones for conferences based on this template. One of **fecc**, **always**, **never**, or **default**.

Value	Description
fecc	In-band DTMF tones will be muted when DTMF is being used to control layout because far end camera control (FECC) is not available
always	In-band DTMF tones will always be muted
never	In-band DTMF tones will never be muted
default	Inherit this setting from the parent template

suppressDtmfEx (*template*) is used in: [template.modify \[p.172\]](#), [template.enumerate \[p.168\]](#), [template.create \[p.163\]](#), [template.status \[p.176\]](#).

Index of parameters: T

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

temperatureStatus string The current temperature status. One of `ok`, `outOfSpec`, or `critical`. The device will shutdown if the `critical` status persists.

`temperatureStatus` is used in: [device.health.query \[p.81\]](#).

temperatureStatusWorst string The worst temperature status recorded on this device since it booted. One of `ok`, `outOfSpec`, or `critical`.

Value	Description
ok	The temperature has been within the normal operating range since the device was booted.
outOfSpec	The temperature has been outside the normal operating range at least once since the device was booted.
critical	At some point since the last boot the temperature was too high. The device will shutdown if this condition persists.

`temperatureStatusWorst` is used in: [device.health.query \[p.81\]](#).

templateName string The name of the template. When passed in a call, this parameter identifies the template that is used for the purpose of the call.

`templateName` is used in: [conference.create \[p.39\]](#), [template.create \[p.163\]](#), [template.delete \[p.167\]](#), [template.enumerate \[p.168\]](#), [template.status \[p.176\]](#).

templateNumber integer An index that uniquely identifies the template. Template numbers are not preserved when the MCU reboots.

The index number of the template. When passed in a call, this parameter identifies the template that is used for the purpose of the call.

The MCU assigns a `templateNumber` and returns it in response to a `template.create` call.

Value	Description
0	The top level template
1	The first created template
2	The second created template. <code>templateNumber</code> increments as new templates are created

`templateNumber` is used in: [conference.create \[p.39\]](#), [template.create \[p.163\]](#), [template.delete \[p.167\]](#), [template.modify \[p.172\]](#), [template.status \[p.176\]](#).

templates array of structs Each array element is a struct that contains the parameters that define a template.

`templates` is used in: [template.create \[p.163\]](#), [template.enumerate \[p.168\]](#).

temporalSpatial integer Integer representing the agreed temporal / spatial trade-off between endpoint and the MCU (motion / sharpness). Value between 0 and 31 (inclusive) where 0 is prefer quality over framerate and 31 is prefer framerate over quality.

temporalSpatial is used in: [participant.statistics \[p.133\]](#).

terminationDate dateTime. Required if **terminationType** is **endOnGivenDate**. This is iso8601 the date when conference repetition will cease.

terminationDate is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

terminationType string Defines how a repeating conference eventually terminates. One of **noTermination**, **afterNRepeats** or **endOnGivenDate**.

Value	Description
noTermination	The conference repeats indefinitely.
afterNRepeats	The conference repeats <i>N</i> times, where <i>N</i> is defined in numberOfRepeats .
endOnGivenDate	The conference will repeat, according to the given repetition and relevant parameters, until the given terminationDate .

terminationType is used in: [conference.create \[p.39\]](#), [conference.enumerate \[p.46\]](#), [conference.modify \[p.57\]](#), [conference.status \[p.66\]](#).

time (CDR log) dateTime. The date and time when the event was logged, for example iso8601 20110119T13:52:42.

time (CDR log) is used in: [cdrlog.enumerate \[p.36\]](#).

time (restart log) dateTime. The date and time when the device restarted. For example, iso8601 20110119T13:52:42 is in the format *yyyymmddThh:mm:ss*.

time is used in: [device.restartlog.query \[p.90\]](#).

totalAudioOnlyPorts integer The total number of additional audio-only ports on the device.

totalAudioOnlyPorts is used in: [device.query \[p.87\]](#).

totalPlaybackPorts integer The number of ports this device uses for playback.

totalPlaybackPorts is used in: [device.query \[p.87\]](#).

totalRecordingPorts integer The number of ports this device uses for recording.

totalRecordingPorts is used in: [device.query \[p.87\]](#).

totalStreamingAndContentPorts integer The total number of streaming and content ports on the MCU. Only provided if non-zero.

totalStreamingAndContentPorts is used in: [device.query \[p.87\]](#).

totalVideoPorts integer The total number of video ports on the device.

totalVideoPorts is used in: [device.query \[p.87\]](#).

transmitErrors integer The count of transmission errors on this Ethernet interface.

transmitErrors is used in: [device.network.query \[p.84\]](#).

transportProtocol string Defines the SIP transport protocol. This parameter is ignored if the communication protocol is not SIP. One of **default**, **tcp**, **udp**, or **tls**.

transportProtocol is used in: [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

type (event) string The name of the event type.

type (event) is used in: [cdrlog.enumerate \[p.36\]](#).

type (pane) string Defines how the MCU fills the pane. One of **default**, **blank**, **loudest**, **rolling**, **h239**, or **participant**.

Value	Description
default	The default pane behavior.
blank	The pane is always blank.
loudest	The pane shows the current loudest speaker.
rolling	The pane shows a sequence of conference participants, changing from one to the next according to the rolling interval.
h239	The pane shows the h239 content channel.
participant	The pane shows a particular participant.

type (pane) is used in: [conference.paneplacement.modify \[p.61\]](#), [conference.paneplacement.query \[p.63\]](#).

type (route) string The type of route. One of **automatic**, **configuredByGateway** or **configuredByPort**.

type (route) is used in: [route.enumerate \[p.153\]](#).

type (service) string The type of service. Either **tcp** or **udp**.

type (service) is used in: [services.modify \[p.156\]](#), [services.query \[p.158\]](#).

type (videoports) string One of **nhd**, **sd**, **hd**, **hdPlus** or **fullhd**

type (videoports) is used in: [device.query \[p.87\]](#).

Index of parameters: U

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

<code>unicastStreamingEnabled</code>	boolean	Defines whether or not this conference can be unicast to streaming viewers. <code>unicastStreamingEnabled</code> is used in: conference.create [p.39] , conference.enumerate [p.46] , conference.modify [p.57] , conference.status [p.66] .
<code>unicastViewers</code>	integer	The count of unicast streaming viewers. <code>unicastViewers</code> is used in: conference.streaming.query [p.72] ,
<code>uniqueId</code>	integer	An ID that is unique among all scheduled and ad hoc conferences. Each instance of a repeating conference has the same <code>uniqueId</code> . <code>uniqueId</code> is used in: conference.enumerate [p.46] , conference.status [p.66] .
<code>unnamed (device.status)</code>	string	A semi-colon delimited list of status monitors and their values at the time of the response. <code>unnamed (device.status)</code> is used in: device.status [p.91] .
<code>useDefaultMotionSharpness</code>	boolean	<code>true</code> means this endpoint will use box-wide default motion sharpness settings. <code>useDefaultMotionSharpness</code> is used in: addressBookEntry.enumerate [p.24] , gateway.enumerate [p.102] .
<code>useDefaultVideoTransmitResolutions</code>	boolean	<code>true</code> means this endpoint will use box-wide default video transmit resolutions. <code>useDefaultVideoTransmitResolutions</code> is used in: addressBookEntry.enumerate [p.24] .
<code>useLocalCertificate</code>	boolean	Shows whether or not the MCU has been set to use the local certificate for connections and registrations <code>useLocalCertificate</code> is used in: sip.modify [p.159] , sip.query [p.160] .
<code>useMaximumPortsFromParent</code>	boolean	Cannot be set to true for template 0 <code>useMaximumPortsFromParent</code> is used in: template.enumerate [p.168] , template.create [p.163] , template.modify [p.172] .
<code>usePassword</code>	boolean	Indicates whether or not the device uses its configured password for gatekeeper registration. <code>usePassword</code> is used in: gatekeeper.modify [p.98] , gatekeeper.query [p.100] .
<code>useReservedPortsFromParent</code>	boolean	Cannot be set to true for template 0 <code>useReservedPortsFromParent</code> is used in: template.modify [p.172] , template.enumerate [p.168] , template.create [p.163] , template.status [p.176] .
<code>useSIPRegistrar</code>	boolean	Not valid unless the protocol is SIP. <code>true</code> if the endpoint uses the SIP registrar. Defaults to <code>false</code> .

useSIPRegistrar is used in: [addressBookEntry.enumerate \[p.24\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#),

useWebService	boolean	true if ConferenceMe may use web service to connect clients to a conference. Corresponds to the "Allow ConferenceMe to use web service" checkbox on the web interface.
----------------------	---------	---

useWebService is used in: [conferenceme.modify \[p.75\]](#), [conferenceme.query \[p.76\]](#).

utcOffsetHours	integer	Number between -12 and +14 (inclusive) that, together with utcOffsetMinutes , defines the UTC offset of the device's clock.
-----------------------	---------	--

utcOffsetHours is used in: [device.time.modify \[p.92\]](#), [device.time.query \[p.93\]](#).

utcOffsetMinutes	integer	Number between 0 and 59 (inclusive) that, together with utcOffsetHours , defines the UTC offset of the device's clock.
-------------------------	---------	---

utcOffsetMinutes is used in: [device.time.modify \[p.92\]](#), [device.time.query \[p.93\]](#).

Index of parameters: V

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

verticalPosition string Specifies where to show the message in relation to the screen. The message is always horizontally centred, and is vertically positioned to either **top**, **middle** (default), or **bottom**.

verticalPosition is used in: [participant.message \[p.126\]](#).

videoCodec string The video codec for this streaming connection. Either **RTSP** or **MMS**.

videoCodec is used in: [streaming.query \[p.162\]](#).

videoControl boolean Defaults to **false**. Set **true** to return **videoControl** statistics.

videoControl is used in: [participant.statistics \[p.133\]](#).

videoLoad integer A percentage value representing the proportion of the device's video processing capacity that is currently in use.

videoLoad is used in: [device.health.query \[p.81\]](#).

videoMedia boolean Defaults to **false**. Set **true** to return **videoMedia** statistics.

videoMedia is used in: [participant.statistics \[p.133\]](#).

videoPortAllocation array An array of **structs**, each of which defines the type and count of video ports that are allocated on this MCU.

videoPortAllocation is used in: [device.query \[p.87\]](#).

videoRTCPOther integer As for the audio equivalents.

videoRTCPOther is used in: [conference.streaming.query \[p.72\]](#).

videoRTCPPacketsSent integer As for the audio equivalents.

videoRTCPPacketsSent is used in: [conference.streaming.query \[p.72\]](#).

videoRTCPReceiverReports integer As for the audio equivalents.

videoRTCPReceiverReports is used in: [conference.streaming.query \[p.72\]](#).

videoRTCPSenderReports integer As for the audio equivalents.

videoRTCPSenderReports is used in: [conference.streaming.query \[p.72\]](#).

videoRx struct A choice of video codecs received from the participant's endpoint.

videoRx (*address book entry*) is used in: [addressBookEntry.enumerate \[p.24\]](#).

videoRxActualBitRate integer The most recently measured bit rate of the incoming video stream from this endpoint (bits per second).

videoRxActualBitRate is used in: [participant.diagnostics \[p.108\]](#).

videoRxBitRateLimitReason string Indicates why the bit rate of the received video stream was limited by the device.

Value	Description
notLimited	
viewedSize	
quality	
aggregateBandwidth	
flowControl	
endpointLimitation	

videoRxBitRateLimitReason is used in: [participant.diagnostics \[p.108\]](#).

videoRxChannelBitRate integer The negotiated available bandwidth for the video stream coming from the endpoint.

videoRxChannelBitRate is used in: [participant.diagnostics \[p.108\]](#).

videoRxCodec string The codec used on the received video.

videoRxCodec is used in: [participant.diagnostics \[p.108\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

videoRxFrameRate integer The frame rate of the received video (frames per second).

videoRxFrameRate is used in: [participant.diagnostics \[p.108\]](#).

videoRxFramesReceived integer The number of video frames received from this endpoint.

videoRxFramesReceived is used in: [participant.diagnostics \[p.108\]](#).

videoRxFramesReceivedWithErrors string The number of video frames received from this endpoint that were not successfully decoded.

videoRxFramesReceivedWithErrors is used in: [participant.diagnostics \[p.108\]](#).

videoRxHeight integer Height in pixels of the received video.

videoRxHeight is used in: [participant.diagnostics \[p.108\]](#).

videoRxInterlaced boolean **true** if the MCU is receiving interlaced video from this endpoint.

videoRxInterlaced is used in: [participant.diagnostics \[p.108\]](#).

videoRxJitter integer Represents the variability of the timing of received video packets.

videoRxJitter is used in: [participant.diagnostics \[p.108\]](#).

videoRxLost integer Count of video packets lost en route to the MCU from this endpoint.

videoRxLost is used in: [participant.diagnostics \[p.108\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

videoRxMaxResolution string The maximum resolution of the received video. One of **cif**, **4cif**, or **max**.

Value	Description
cif	This endpoint sends cif or lower resolution to the MCU.
4cif	This endpoint sends 4cif or lower resolution to the MCU.
max	Send the maximum resolution that both sides can support.

videoRxMaxResolution is used in: [participant.add \[p.103\]](#).

videoRxMuted boolean **true** means that video from this participant will not be seen by other conference participants.

videoRxMuted is used in: [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

videoRxReceived integer Count of video packets received from this endpoint.

videoRxReceived is used in: [participant.diagnostics \[p.108\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

videoRxSelectedBitRate integer The bit rate which the MCU has requested for the video stream from this endpoint (bits per second).

videoRxSelectedBitRate is used in: [participant.diagnostics \[p.108\]](#).

videoRxWidth integer Width in pixels of the received video.

videoRxWidth is used in: [participant.diagnostics \[p.108\]](#).

videoStreams array An array of stream structs. The structs are only present if there are any streams of either type currently in use.

videoStreams is used in: [conference.streaming.query \[p.72\]](#).

videoToUse struct Collection of parameters that uniquely identify the participant whose video will display in place of this participant's video by default.

videoToUse is used in: [addressBookEntry.enumerate \[p.24\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#).

videoTransmitResolutions string Overrides the default setting for video resolution the MCU may send to the endpoint. One of **allowAll**, **4to3Only**, **4to3WidescreenOverride**, or **16to9Only**.

Defines the video resolution that the MCU will transmit to this endpoint. The default is to use the box-wide setting, but you can set to one of the following overrides if necessary.

Value	Description
allowAll	The MCU may transmit any of the available resolutions to the endpoint.
4to3Only	The MCU may only transmit 4:3 video to this endpoint.
4to3WidescreenOverride	The MCU may transmit 4:3 video, modified to fit widescreen, to this endpoint.
16to9Only	The MCU may only transmit 16:9 video to this endpoint.

`videoTransmitResolutions` is used in: [addressBookEntry.enumerate \[p.24\]](#).

`videoTx` struct A choice of video codecs advertised by the MCU.

`videoTx` (*address book entry*) is used in: [addressBookEntry.enumerate \[p.24\]](#).

`videoTxActualBitRate` integer The most recently measured bit rate of the outgoing video stream to this endpoint (bits per second).

`videoTxActualBitRate` is used in: [participant.diagnostics \[p.108\]](#).

`videoTxBitRateLimitReason` string Indicates why the bit rate of the transmitted video stream was limited by the device. One of `notLimited`, `viewedSize`, `quality`, `aggregateBandwidth`, `flowControl`, or `endpointLimitation`.

`videoTxBitRateLimitReason` is used in: [participant.diagnostics \[p.108\]](#).

`videoTxChannelBitRate` integer The negotiated available bandwidth for the video stream going to the endpoint.

`videoTxChannelBitRate` is used in: [participant.diagnostics \[p.108\]](#).

`videoTxCodec` string The codec used on the transmitted video.

`videoTxCodec` is used in: [participant.diagnostics \[p.108\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

`videoTxFrameRate` integer Frame rate of the transmitted video (frames per second).

`videoTxFrameRate` is used in: [participant.diagnostics \[p.108\]](#).

`videoTxHeight` integer Height in pixels of the transmitted video.

`videoTxHeight` is used in: [participant.diagnostics \[p.108\]](#).

`videoTxInterlaced` boolean `true` if the MCU is sending interlaced video to this endpoint.

`videoTxInterlaced` is used in: [participant.diagnostics \[p.108\]](#).

`videoTxMaxResolution` string The maximum resolution transmitted to this endpoint. One of `cif`, `4cif`, or `max`.

Value	Description
cif	Send <code>cif</code> or lower resolution to this endpoint.
4cif	Send <code>4cif</code> or lower resolution to this endpoint.
max	Send the maximum resolution that both sides can support.

`videoTxMaxResolution` is used in: [participant.add \[p.103\]](#).

`videoTxMuted` boolean `true` means that the MCU does not send the video part of the conference to this participant.

`videoTxMuted` is used in: [addressBookEntry.enumerate \[p.24\]](#), [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.modify \[p.127\]](#).

`videoTxReportedLost` integer The count of video packets reported lost by the far end.

`videoTxReportedLost` is used in: [participant.diagnostics \[p.108\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

`videoTxSelectedBitRate` integer The bit rate at which the MCU is attempting to send video to this endpoint (bits per second). This value may be lower than `videoTxChannelBitRate` which is an effective maximum.

`videoTxSelectedBitRate` is used in: [participant.diagnostics \[p.108\]](#).

`videoTxSent` integer Count of the video packets sent to the endpoint.

`videoTxSent` is used in: [participant.diagnostics \[p.108\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

`videoTxWidescreen` boolean If `true`, the MCU sends video in a form suitable for a widescreen 16:9 display to this participant.

`videoTxWidescreen` is used in: [participant.add \[p.103\]](#), [participant.enumerate \[p.112\]](#), [participant.enumerate \(deprecated\) \[p.121\]](#), [participant.modify \[p.127\]](#), [participant.status \[p.139\]](#), [participant.status \(deprecated\) \[p.147\]](#).

`videoTxWidth` integer Width in pixels of the transmitted video.

`videoTxWidth` is used in: [participant.diagnostics \[p.108\]](#).

`voltagesStatus` string `ok`, `outOfSpec` (the voltage is currently outside the normal range), or `critical`.

`voltagesStatus` is used in: [device.health.query \[p.81\]](#).

`voltagesStatusWorst` string `ok`, `outOfSpec` (the voltage has been outside the normal range at some time since the device last booted), or `critical`.

`voltagesStatusWorst` is used in: [device.health.query \[p.81\]](#).

Index of parameters: W

[a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#)

webAppletBandwidth	integer	The bandwidth of the content stream sent to streaming viewers.
webAppletBandwidth is used in: device.content.modify [p.77] , device.content.query [p.78] .		
weekDay	string	Must be present if repetition is monthly . One of monday , tuesday , wednesday , thursday , friday , saturday or sunday . Note that if repetition is not weekly or everyTwoWeeks , the weekDays parameter should be used.
weekDay is used in: conference.create [p.39] , conference.enumerate [p.46] , conference.modify [p.57] , conference.status [p.66] .		
weekDays	string	Required if repetition is weekly or everyTwoWeeks . The parameter accepts a comma separated string of weekday names, e.g. monday , wednesday , friday .
weekDays is used in: conference.create [p.39] , conference.enumerate [p.46] , conference.modify [p.57] , conference.status [p.66] .		
whichWeek	string	Required if repetition is monthly . Defines which week the repeating conference will fall in; one of first , second , third , fourth , or last .
whichWeek is used in: conference.create [p.39] , conference.enumerate [p.46] , conference.modify [p.57] , conference.status [p.66] .		
width	integer	The maximum width and height of this stream. Only present for defined video streams
width is used in: conference.streaming.query [p.72] , participant.statistics [p.133] .		
wmpProtocol	string	Describes the behavior of the wmpProtocol when streaming to the endpoint. One of auto , mmsOverUdp , mmsOverTcp , or http .
wmpProtocol is used in: streaming.modify [p.161] , streaming.query [p.162] .		

API Change history

This section details the changes in each version of the API.

Version 2.10 changes	281
Version 2.9 changes	283
Version 2.8 changes	291

Version 2.10 changes

Category	Name	Change
Command	addressBookEntry.enumerate [p.24]	Parameters added
Command	conference.create [p.39]	Parameters added
Command	conference.enumerate [p.46]	Parameters added
Command	conference.metadata.modify [p.55]	Parameters added
Command	conference.modify [p.57]	Parameters added
Command	conference.paneplacement.query [p.63]	Modified
Command	conference.status [p.66]	Parameters added
Command	device.query [p.87]	Parameters added
Command	device.status [p.91]	Introduced
Command	participant.add [p.103]	Parameters added
Command	participant.enumerate [p.112]	Parameters added
Command	participant.enumerate (deprecated) [p.121]	Parameters added
Command	participant.modify [p.127]	Parameters added
Command	participant.status [p.139]	Parameters added
Command	participant.status (deprecated) [p.147]	Parameters added
Command	template.create [p.163]	Parameters added
Command	template.delete [p.167]	Parameters added
Command	template.enumerate [p.168]	Parameters added
Command	template.modify [p.172]	Parameters added
Command	template.status [p.176]	Parameters added
Filter	connecting [p.211]	Modified
Filter	dormant [p.222]	Introduced
Parameter	audioRxGainMode [p.199]	Modified
Parameter	audioTxMuted [p.200]	Modified
Parameter	callState [p.206]	Deprecated
Parameter	callStateEx [p.206]	Introduced
Parameter	contentContribution (template) [p.212]	Modified
Parameter	contentReceive [p.213]	Introduced
Parameter	deferConnection [p.220]	Modified
Parameter	dtmfSequence [p.223]	Modified
Parameter	joinAGC [p.239]	Introduced
Parameter	joinAGC (template) [p.239]	Introduced
Parameter	maxConferenceSize [p.244]	Introduced

Category	Name	Change
Parameter	redial [p.259]	Introduced
Parameter	redialLimit [p.259]	Introduced
Parameter	suppressAudioDuringDTMF [p.267]	Introduced
Parameter	unnamed (device.status) [p.272]	Introduced
Parameter	videoToUse [p.276]	Introduced
Parameter	videoTxMuted [p.278]	Introduced

Version 2.9 changes

Category	Name	Change
Command	addressBookEntry.enumerate [p.24]	Parameters added
Command	cdrlog.enumerate [p.36]	Parameters added
Command	conference.create [p.39]	Parameters replaced
Command	conference.enumerate [p.46]	Parameters added, replaced
Command	conference.modify [p.57]	Parameters replaced
Command	conference.status [p.66]	Parameters added, replaced
Command	conferenceme.modify [p.75]	Introduced
Command	conferenceme.query [p.76]	Parameters added
Command	device.content.modify [p.77]	Introduced
Command	device.content.query [p.78]	Introduced
Command	device.encryption.modify [p.79]	Introduced
Command	device.encryption.query [p.80]	Introduced
Command	device.network.modify [p.82]	Introduced
Command	device.query [p.87]	Parameters added
Command	device.restart [p.89]	Introduced
Command	device.time.modify [p.92]	Introduced
Command	device.time.query [p.93]	Introduced
Command	feedbackReceiver.reconfigure [p.96]	Introduced
Command	feedbackReceiver.remove [p.97]	Introduced
Command	gatekeeper.modify [p.98]	Introduced
Command	gatekeeper.query [p.100]	Parameters added
Command	participant.add [p.103]	Parameters added
Command	participant.diagnostics [p.108]	Deprecated
Command	participant.statistics [p.133]	Introduced
Command	route.add [p.151]	Introduced
Command	route.delete [p.152]	Introduced
Command	route.enumerate [p.153]	Introduced
Command	route.preferences.modify [p.154]	Introduced
Command	route.preferences.query [p.155]	Introduced
Command	services.modify [p.156]	Introduced
Command	services.query [p.158]	Parameters added

Category	Name	Change
Command	sip.modify [p.159]	Introduced
Command	streaming.modify [p.161]	Introduced
Command	streaming.query [p.162]	Introduced
Command	template.create [p.163]	Parameters replaced
Command	template.enumerate [p.168]	Parameters replaced
Command	template.modify [p.172]	Parameters replaced
Command	template.status [p.176]	Parameters replaced
Fault	36: XML_RPC_FAULT_REQUIRED_VALUE_MISSING Fault codes [p.183]	Introduced
Fault	42: XML_RPC_FAULT_PORT_CONFLICT Fault codes [p.183]	Introduced
Fault	43: XML_RPC_FAULT_ROUTE_ALIASES_EXISTING Fault codes [p.183]	Introduced
Fault	44: XML_RPC_FAULT_ROUTE_REJECTED Fault codes [p.183]	Introduced
Fault	45: XML_RPC_FAULT_TOO_MANY_ROUTES Fault codes [p.183]	Introduced
Fault	46: XML_RPC_FAULT_NO_SUCH_ROUTE Fault codes [p.183]	Introduced
Fault	48: XML_RPC_FAULT_IP_ADDRESS_OVERFLOW_MASK Fault codes [p.183]	Introduced
Fault	49: XML_RPC_FAULT_DISABLE_ACTIVE_INTERFACE Fault codes [p.183]	Introduced
Fault	104: XML_RPC_FAULT_MISMATCHED_PARAMETERS Fault codes [p.183]	Introduced
Fault	105: XML_RPC_FAULT_REQUEST_TOO_LARGE Fault codes [p.183]	Introduced
Feedback event	sipChanged Feedback events [p.18]	Introduced
Feedback event	h323Changed Feedback events [p.18]	Introduced
Feedback event	floorChanged Feedback events [p.18]	Introduced
Feedback event	chairChanged Feedback events [p.18]	Introduced
Feedback event	encryptionChanged Feedback events [p.18]	Introduced
Feedback event	contentChanged Feedback events [p.18]	Introduced
Feedback event	streamingChanged Feedback events [p.18]	Introduced
Feedback event	conferenceMeChanged Feedback events [p.18]	Introduced
Feedback event	networkChanged Feedback events [p.18]	Introduced
Feedback event	servicesChanged Feedback events [p.18]	Introduced
Feedback event	routesChanged Feedback events [p.18]	Introduced
Feedback event	deviceStatusChanged Feedback events [p.18]	Introduced
Feedback event	rebooting Feedback events [p.18]	Introduced

Category	Name	Change
Feedback event	timeChanged Feedback events [p.18]	Introduced
Parameter	aac-Ic [p.196]	Introduced
Parameter	aac-Id [p.196]	Introduced
Parameter	active (route) [p.197]	Introduced
Parameter	actualBitRate [p.197]	Introduced
Parameter	addResponse [p.197]	Introduced
Parameter	audioCodec [p.198]	Introduced
Parameter	audioControl [p.198]	Introduced
Parameter	audioMedia [p.198]	Introduced
Parameter	audioRx [p.199]	Introduced
Parameter	audioRxLost [p.199]	Deprecated
Parameter	audioRxReceived [p.200]	Deprecated
Parameter	audioTx [p.200]	Introduced
Parameter	audioTxReportedLost [p.200]	Deprecated
Parameter	bitRateLimitReason [p.205]	Introduced
Parameter	cameraControl [p.207]	Modified
Parameter	chairControl [p.208]	Modified
Parameter	chairControl (template) [p.208]	Modified
Parameter	chairParticipant [p.209]	Introduced
Parameter	channelBitRate [p.209]	Introduced
Parameter	codecBitRate [p.209]	Introduced
Parameter	contentControl [p.212]	Introduced
Parameter	contentEnabled [p.212]	Introduced
Parameter	contentError [p.212]	Introduced
Parameter	contentHandoverEnabled [p.212]	Introduced
Parameter	contentInMainVideo [p.212]	Introduced
Parameter	contentMarkupEnabled [p.212]	Introduced
Parameter	contentMedia [p.212]	Introduced
Parameter	contentRxSelectedBitRate [p.214]	Deprecated
Parameter	contentRxActualBitRate [p.213]	Deprecated
Parameter	contentRxBitRateLimitReason [p.213]	Deprecated
Parameter	contentRxChannelBitRate [p.213]	Deprecated
Parameter	contentRxCodec [p.213]	Deprecated
Parameter	contentRxFrameRate [p.213]	Deprecated

Category	Name	Change
Parameter	contentRxFramesReceived [p.214]	Deprecated
Parameter	contentRxFramesReceivedWithErrors [p.214]	Deprecated
Parameter	contentRxHeight [p.214]	Deprecated
Parameter	contentRxJitter [p.214]	Deprecated
Parameter	contentRxLost [p.214]	Deprecated
Parameter	contentRxReceived [p.214]	Deprecated
Parameter	contentRxType [p.214]	Deprecated
Parameter	contentRxWidth [p.214]	Deprecated
Parameter	contentStreamingSetting [p.214]	Introduced
Parameter	contentStreamingStatus [p.214]	Introduced
Parameter	contentTxActualBitRate [p.215]	Deprecated
Parameter	contentTxBitRateLimitReason [p.215]	Deprecated
Parameter	contentTxChannelBitRate [p.216]	Deprecated
Parameter	contentTxCodec [p.216]	Deprecated
Parameter	contentTxError [p.216]	Deprecated
Parameter	contentTxFrameRate [p.216]	Deprecated
Parameter	contentTxHeight [p.216]	Deprecated
Parameter	contentTxReportedLost [p.217]	Deprecated
Parameter	contentTxSelectedBitRate [p.217]	Deprecated
Parameter	contentTxSent [p.217]	Deprecated
Parameter	contentTxType [p.217]	Deprecated
Parameter	contentTxWidth [p.217]	Deprecated
Parameter	contentType [p.217]	Introduced
Parameter	customCodecs [p.218]	Introduced
Parameter	customCodecSelection [p.218]	Introduced
Parameter	defaultGateway [p.220]	Deprecated
Parameter	defaultIpv4Gateway [p.220]	Introduced
Parameter	destination [p.221]	Introduced
Parameter	dhcp [p.221]	Deprecated
Parameter	dhcpv4 [p.221]	Introduced
Parameter	dnsConfiguration [p.222]	Introduced
Parameter	dtmfMuteControl [p.223]	Deprecated
Parameter	dtmfMuteControl (template) [p.223]	Deprecated
Parameter	encryption [p.224]	Introduced

Category	Name	Change
Parameter	energyMillidB [p.224]	Introduced
Parameter	ethernetAutomatic [p.226]	Introduced
Parameter	fecOverhead [p.227]	Introduced
Parameter	fecRecovered [p.227]	Introduced
Parameter	filter (route) [p.227]	Introduced
Parameter	filter [p.227]	Introduced
Parameter	finishedBooting [p.227]	Introduced
Parameter	flowControlReceived [p.228]	Introduced
Parameter	flowControlSent [p.228]	Introduced
Parameter	format [p.228]	Introduced
Parameter	format1 [p.228]	Introduced
Parameter	format2 [p.228]	Introduced
Parameter	frameErrors [p.228]	Introduced
Parameter	frameRate [p.228]	Introduced
Parameter	framesTransferred [p.229]	Introduced
Parameter	furFilteringEnabled [p.229]	Introduced
Parameter	fursReceived [p.229]	Introduced
Parameter	fursSent [p.229]	Introduced
Parameter	g711 [p.230]	Introduced
Parameter	g722 [p.230]	Introduced
Parameter	g722.1 [p.230]	Introduced
Parameter	g722.1c [p.230]	Introduced
Parameter	g723.1 [p.230]	Introduced
Parameter	g728 [p.230]	Introduced
Parameter	g729 [p.230]	Introduced
Parameter	gateway [p.230]	Introduced
Parameter	h261 [p.233]	Introduced
Parameter	h263 [p.233]	Introduced
Parameter	h263+ [p.233]	Introduced
Parameter	h263i [p.233]	Introduced
Parameter	h264 [p.233]	Introduced
Parameter	inCallMenuControlChair [p.235]	Introduced
Parameter	inCallMenuControlChair (template) [p.235]	Introduced
Parameter	inCallMenuControlGuest [p.235]	Introduced

Category	Name	Change
Parameter	inCallMenuControlGuest (template) [p.236]	Introduced
Parameter	Interlaced [p.237]	Introduced
Parameter	ipAddress [p.237]	Deprecated
Parameter	ipRangeFinish [p.237]	Introduced
Parameter	ipRangeStart [p.237]	Introduced
Parameter	ipv4Address [p.237]	Introduced
Parameter	ipv4MulticastRange [p.237]	Introduced
Parameter	ipv4Preference [p.237]	Introduced
Parameter	ipv4Routes [p.237]	Introduced
Parameter	ipv4SubnetMask [p.237]	Introduced
Parameter	ipv6MulticastRange [p.238]	Introduced
Parameter	ipv6Preference [p.238]	Introduced
Parameter	ipv6Routes [p.238]	Introduced
Parameter	jitter [p.239]	Introduced
Parameter	jitterBuffer [p.239]	Introduced
Parameter	layoutControlEx [p.241]	Modified
Parameter	lipSyncDelayApplied [p.243]	Introduced
Parameter	maxParticipants [p.245]	Introduced
Parameter	mediaResources [p.246]	Introduced
Parameter	multicast [p.247]	Introduced
Parameter	newRouteld [p.249]	Introduced
Parameter	ntpEnabled [p.249]	Introduced
Parameter	ntpHost [p.249]	Introduced
Parameter	ntpStatus [p.249]	Introduced
Parameter	numEvents (per enumeration) [p.250]	Introduced
Parameter	packetsErrors [p.252]	Introduced
Parameter	packetsTransferred [p.252]	Introduced
Parameter	password (gatekeeper) [p.254]	Introduced
Parameter	portAssociationA [p.254]	Deprecated
Parameter	portAssociationAv4 [p.254]	Introduced
Parameter	portAssociationB [p.254]	Deprecated
Parameter	portAssociationBv4 [p.254]	Introduced
Parameter	portRangeFinish [p.255]	Introduced
Parameter	portRangeStart [p.255]	Introduced

Category	Name	Change
Parameter	prefixLength [p.256]	Introduced
Parameter	rebootRequired [p.258]	Introduced
Parameter	routeId [p.263]	Introduced
Parameter	rtcpLipSyncDelay [p.263]	Introduced
Parameter	rtcpOtherReports [p.263]	Introduced
Parameter	rtcpPacketLossReported [p.263]	Introduced
Parameter	rtcpPacketsSent [p.263]	Introduced
Parameter	rtcpReceiveAddress [p.263]	Introduced
Parameter	rtcpReceivePort [p.263]	Introduced
Parameter	rtcpReceiverReports [p.264]	Introduced
Parameter	rtcpSenderReports [p.264]	Introduced
Parameter	rtcpTransmitAddress [p.264]	Introduced
Parameter	rtcpTransmitPort [p.264]	Introduced
Parameter	selectedBitRate [p.265]	Introduced
Parameter	setting [p.265]	Introduced
Parameter	shutdownOnly [p.265]	Introduced
Parameter	shutdownStatus [p.265]	Introduced
Parameter	sipMediaEncryption [p.266]	Introduced
Parameter	siren14 [p.266]	Introduced
Parameter	subnetMask [p.267]	Deprecated
Parameter	temporalSpatial [p.270]	Introduced
Parameter	type (route) [p.271]	Introduced
Parameter	usePassword [p.272]	Introduced
Parameter	utcOffsetHours [p.273]	Introduced
Parameter	utcOffsetMinutes [p.273]	Introduced
Parameter	videoCodec [p.274]	Introduced
Parameter	videoControl [p.274]	Introduced
Parameter	videoMedia [p.274]	Introduced
Parameter	videoRx [p.274]	Introduced
Parameter	videoRxActualBitRate [p.275]	Deprecated
Parameter	videoRxBitRateLimitReason [p.275]	Deprecated
Parameter	videoRxChannelBitRate [p.275]	Deprecated
Parameter	videoRxCodec [p.275]	Deprecated
Parameter	videoRxFrameRate [p.275]	Deprecated

Category	Name	Change
Parameter	videoRxFramesReceived [p.275]	Deprecated
Parameter	videoRxFramesReceivedWithErrors [p.275]	Deprecated
Parameter	videoRxHeight [p.275]	Deprecated
Parameter	videoRxInterlaced [p.275]	Deprecated
Parameter	videoRxJitter [p.275]	Deprecated
Parameter	videoRxLost [p.276]	Deprecated
Parameter	videoRxReceived [p.276]	Deprecated
Parameter	videoRxSelectedBitRate [p.276]	Deprecated
Parameter	videoRxWidth [p.276]	Deprecated
Parameter	videoTx [p.277]	Introduced
Parameter	videoTxActualBitRate [p.277]	Deprecated
Parameter	videoTxBitRateLimitReason [p.277]	Deprecated
Parameter	videoTxChannelBitRate [p.277]	Deprecated
Parameter	videoTxCodec [p.277]	Deprecated
Parameter	videoTxFrameRate [p.277]	Deprecated
Parameter	videoTxHeight [p.277]	Deprecated
Parameter	videoTxInterlaced [p.277]	Deprecated
Parameter	videoTxReportedLost [p.278]	Deprecated
Parameter	videoTxSelectedBitRate [p.278]	Deprecated
Parameter	videoTxSent [p.278]	Deprecated
Parameter	videoTxWidth [p.278]	Deprecated
Parameter	webAppletBandwidth [p.279]	Introduced
Parameter	wmpProtocol [p.279]	Introduced

Version 2.8 changes

Category	Name	Change
Command	addressBookEntry.enumerate [p.24]	Parameters added
Command	cdrlog.enumerate [p.36]	Introduced
Command	conference.create [p.39]	Parameters added
Command	conference.enumerate [p.46]	Parameters added
Command	conference.modify [p.57]	Parameters added
Command	conference.resetCleanupTimeout [p.65]	Introduced
Command	conference.status [p.66]	Parameters added
Command	conferenceme.query [p.76]	Parameters added
Command	device.network.query [p.84]	Restructured, parameters added
Command	gatekeeper.query [p.100]	Parameters added
Command	participant.add [p.103]	Parameters added
Command	participant.diagnostics [p.108]	Parameters added
Command	participant.enumerate [p.112]	Parameters added
Command	participant.modify [p.127]	Parameters added
Command	participant.status [p.139]	Parameters added
Command	services.query [p.158]	Introduced
Command	sip.query [p.160]	Parameters added
Command	template.create [p.163]	Introduced
Command	template.delete [p.167]	Introduced
Command	template.enumerate [p.168]	Parameters added
Command	template.modify [p.172]	Parameters added
Command	template.status [p.176]	Parameters added
Fault	27 No such template Fault codes [p.183]	Introduced
Fault	31 Template name in use Fault codes [p.183]	Introduced
Fault	32 Too many templates Fault codes [p.183]	Introduced
Parameter	actAsRecorder [p.196]	Introduced
Parameter	addAsGuest [p.197]	Introduced
Parameter	alternateGatekeepers [p.198]	Introduced
Parameter	availabilityThresholdConferences [p.203]	Introduced
Parameter	availabilityThresholdVideoPorts [p.204]	Introduced
Parameter	cameraControl [p.207]	Introduced
Parameter	cameraControlDefault [p.208]	Introduced
Parameter	cleanupTimeout [p.209]	Introduced

Category	Name	Change
Parameter	conferenceMeEnabled [p.210]	Introduced
Parameter	contentImportant [p.212]	Introduced
Parameter	contentRxActualBitRate [p.213]	Introduced
Parameter	contentRxBitRateLimitReason [p.213]	Introduced
Parameter	contentRxChannelBitRate [p.213]	Introduced
Parameter	contentRxCodec [p.213]	Introduced
Parameter	contentRxFrameRate [p.213]	Introduced
Parameter	contentRxFramesReceived [p.214]	Introduced
Parameter	contentRxFramesReceivedWithErrors [p.214]	Introduced
Parameter	contentRxHeight [p.214]	Introduced
Parameter	contentRxJitter [p.214]	Introduced
Parameter	contentRxLost [p.214]	Introduced
Parameter	contentRxReceived [p.214]	Introduced
Parameter	contentRxSelectedBitRate [p.214]	Introduced
Parameter	contentRxType [p.214]	Introduced
Parameter	contentRxWidth [p.214]	Introduced
Parameter	contentTxActualBitRate [p.215]	Introduced
Parameter	contentTxBitRateLimitReason [p.215]	Introduced
Parameter	contentTxChannelBitRate [p.216]	Introduced
Parameter	contentTxCodec [p.216]	Introduced
Parameter	contentTxError [p.216]	Introduced
Parameter	contentTxFrameRate [p.216]	Introduced
Parameter	contentTxHeight [p.216]	Introduced
Parameter	contentTxMinimumBitRate [p.217]	Introduced
Parameter	contentTxReportedLost [p.217]	Introduced
Parameter	contentTxSelectedBitRate [p.217]	Introduced
Parameter	contentTxSent [p.217]	Introduced
Parameter	contentTxType [p.217]	Introduced
Parameter	contentTxWidth [p.217]	Introduced
Parameter	defaultIpv6Gateway [p.220]	Introduced
Parameter	dns [p.222]	Introduced
Parameter	events (feedback) [p.226]	Introduced
Parameter	eventsRemaining [p.226]	Introduced
Parameter	filter [p.227]	Introduced

Category	Name	Change
Parameter	guest [p.231]	Introduced
Parameter	h239Important [p.232]	Deprecated
Parameter	h239Negotiation [p.232]	Introduced
Parameter	index (CDR log enumerate call) [p.236]	Introduced
Parameter	ipv4Enabled [p.237]	Introduced
Parameter	ipv6Address [p.237]	Introduced
Parameter	ipv6Enabled [p.237]	Introduced
Parameter	ipv6PrefixLength [p.238]	Introduced
Parameter	lastChairmanLeavesDisconnect [p.240]	Introduced
Parameter	layoutControlEx [p.241]	Modified
Parameter	linkLocalIpv6Address [p.242]	Introduced
Parameter	linkLocalIpv6PrefixLength [p.242]	Introduced
Parameter	maxOcsBitrate [p.244]	Introduced
Parameter	moreThanFour [p.246]	Introduced
Parameter	newTemplateName [p.249]	Introduced
Parameter	nextIndex [p.249]	Introduced
Parameter	numEvents (CDR log) [p.250]	Introduced
Parameter	outgoingTransport [p.251]	Introduced
Parameter	parent [p.252]	Introduced
Parameter	portA [p.254]	Modified
Parameter	portAssociationA [p.254]	Introduced
Parameter	portAssociationAv6 [p.254]	Introduced
Parameter	portAssociationB [p.254]	Introduced
Parameter	portAssociationBv6 [p.255]	Introduced
Parameter	portB [p.255]	Modified
Parameter	preconfiguredParticipantsDefer [p.255]	Introduced
Parameter	registeredAddress [p.260]	Introduced
Parameter	registrarType [p.261]	Introduced
Parameter	registrarUsage [p.261]	Introduced
Parameter	registrationStatus [p.261]	Introduced
Parameter	registrationType [p.261]	Introduced
Parameter	remoteLinkType [p.262]	Introduced
Parameter	reserveAudioPorts [p.262]	Introduced
Parameter	reserveVideoPorts [p.262]	Introduced

Category	Name	Change
Parameter	resourceAvailabilityStatus [p.263]	Introduced
Parameter	sendResourceAvailabilityIndications [p.265]	Introduced
Parameter	startIndex [p.266]	Introduced
Parameter	startLocked [p.266]	Introduced
Parameter	templateName [p.269]	Introduced
Parameter	templateNumber [p.269]	Modified
Parameter	useLocalCertificate [p.272]	Introduced
Parameter	useMaximumPortsFromParent [p.272]	Introduced
Parameter	useReservedPortsFromParent [p.272]	Introduced
Parameter	useWebService [p.273]	Introduced
Parameter	videoPortAllocation [p.274]	Modified

References

1. XML-RPC specification (Dave Winer, June 1999); <http://www.xmlrpc.com/spec>, accessed 24/01/2011.
2. HTTP/1.1 specification (RFC 2616, Fielding et al., June 1999); <http://www.ietf.org/rfc/rfc2616.txt>, accessed 24/01/2011.

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.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

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.

© 2012 Cisco Systems, Inc. All rights reserved.