

# Cisco TelePresence MCU API 2.11

## Product Programming Reference Guide

---

D15084

June 2014

---

# Contents

<b>Introduction</b> .....	<b>5</b>
API History .....	5
XML-RPC implementation .....	5
Transport protocol .....	6
Clustering .....	6
Considering API overhead when writing applications .....	6
<b>Overview</b> .....	<b>8</b>
Encoding .....	9
Authentication .....	9
Message flow .....	10
Participant identification .....	11
Enumerate methods .....	12
Enumerate filters .....	12
Revision numbers .....	13
<b>Feedback</b> .....	<b>15</b>
Feedback receivers .....	16
Feedback messages .....	17
Feedback events .....	18
<b>API commands</b> .....	<b>20</b>
Deprecations .....	23
addressBookEntry.enumerate .....	24
auditlog.delete .....	30
auditlog.query .....	31
autoAttendant.destroy .....	32
autoAttendant.enumerate .....	33
autoAttendant.status .....	34
callHome.configure .....	35
callHome.query .....	36
cdrlog.delete .....	37
cdrlog.enumerate .....	38
cdrlog.query .....	40
conference.create .....	41
conference.destroy .....	46
conference.end .....	47
conference.enumerate .....	48
conference.floor.modify .....	55
conference.floor.query .....	56
conference.metadata.modify .....	57
conference.metadata.status .....	58
conference.modify .....	59
conference.paneplacement.modify .....	63
conference.paneplacement.query .....	65
conference.resetCleanupTimeout .....	67
conference.status .....	68
conference.streaming.modify .....	73
conference.streaming.query .....	74
conferenceme.modify .....	77

conferenceme.query .....	78
device.content.modify .....	79
device.content.query .....	80
device.encryption.modify .....	81
device.encryption.query .....	82
device.features.add .....	83
device.features.remove .....	84
device.health.query .....	85
device.network.modify .....	86
device.network.query .....	88
device.query .....	91
device.restart .....	93
device.restartlog.query .....	94
device.status .....	95
device.time.modify .....	96
device.time.query .....	97
feedbackReceiver.configure .....	98
feedbackReceiver.query .....	99
feedbackReceiver.reconfigure .....	100
feedbackReceiver.remove .....	101
gatekeeper.modify .....	102
gatekeeper.query .....	104
gateway.enumerate .....	106
participant.add .....	107
participant.connect .....	111
participant.diagnostics .....	112
participant.disconnect .....	115
participant.enumerate .....	116
participant.enumerate (deprecated) .....	125
participant.fecc .....	129
participant.message .....	130
participant.modify .....	131
participant.move .....	135
participant.remove .....	136
participant.statistics .....	137
participant.status .....	143
participant.status (deprecated) .....	151
route.add .....	155
route.delete .....	156
route.enumerate .....	157
route.preferences.modify .....	158
route.preferences.query .....	159
services.modify .....	160
services.query .....	162
sip.modify .....	163
sip.query .....	165
streaming.modify .....	166
streaming.query .....	167
template.create .....	168
template.delete .....	172
template.enumerate .....	173

template.modify .....	177
template.status .....	181
<b>Related information .....</b>	<b>185</b>
system.xml file .....	186
Fault codes .....	188
Disconnect reasons .....	190
HTTP keep-alives .....	192
Conference layouts .....	193
Linking conferences across MCUs .....	195
<b>Index of parameters .....</b>	<b>200</b>
Index of parameters: A .....	201
Index of parameters: B .....	210
Index of parameters: C .....	211
Index of parameters: D .....	225
Index of parameters: E .....	229
Index of parameters: F .....	232
Index of parameters: G .....	235
Index of parameters: H .....	237
Index of parameters: I .....	240
Index of parameters: J .....	244
Index of parameters: K .....	245
Index of parameters: L .....	246
Index of parameters: M .....	250
Index of parameters: N .....	254
Index of parameters: O .....	257
Index of parameters: P .....	259
Index of parameters: Q .....	264
Index of parameters: R .....	265
Index of parameters: S .....	272
Index of parameters: T .....	276
Index of parameters: U .....	279
Index of parameters: V .....	281
Index of parameters: W .....	286
<b>API Change history .....</b>	<b>287</b>
Version 2.11 changes .....	288
Version 2.10 changes .....	290
Version 2.9 changes .....	292
Version 2.8 changes .....	300
<b>References .....</b>	<b>304</b>

# 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 4.5 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
2.11 (this version)	4.5 and later
2.10	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](#)<sup>[1]</sup> for more information.

## Transport protocol

The device implements HTTP/1.1 as defined by [RFC 2616](#)<sup>[2]</sup>. 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 ..... 9
- Authentication ..... 9
- Message flow .....10
- Participant identification ..... 11
- Enumerate methods .....12
- Enumerate filters ..... 12
- Revision numbers ..... 13

## 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**.

# 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 \[p.98\]](#) to register a receiver to listen for one or more [feedback events](#).
- Use [feedbackReceiver.query \[p.99\]](#) to return a list of receivers that are configured on the device.
- Use [feedbackReceiver.reconfigure \[p.100\]](#) to change the configuration of an existing feedback receiver.
- Use [feedbackReceiver.remove \[p.101\]](#) 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.

<b>Event</b>	<b>Description</b>
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)**.

---

Deprecations .....	23
addressBookEntry.enumerate .....	24
auditlog.delete .....	30
auditlog.query .....	31
autoAttendant.destroy .....	32
autoAttendant.enumerate .....	33
autoAttendant.status .....	34
callHome.configure .....	35
callHome.query .....	36
cdrlog.delete .....	37
cdrlog.enumerate .....	38
cdrlog.query .....	40
conference.create .....	41
conference.destroy .....	46
conference.end .....	47
conference.enumerate .....	48
conference.floor.modify .....	55
conference.floor.query .....	56
conference.metadata.modify .....	57
conference.metadata.status .....	58
conference.modify .....	59
conference.paneplacement.modify .....	63
conference.paneplacement.query .....	65
conference.resetCleanupTimeout .....	67
conference.status .....	68

conference.streaming.modify .....	73
conference.streaming.query .....	74
conferenceme.modify .....	77
conferenceme.query .....	78
device.content.modify .....	79
device.content.query .....	80
device.encryption.modify .....	81
device.encryption.query .....	82
device.features.add .....	83
device.features.remove .....	84
device.health.query .....	85
device.network.modify .....	86
device.network.query .....	88
device.query .....	91
device.restart .....	93
device.restartlog.query .....	94
device.status .....	95
device.time.modify .....	96
device.time.query .....	97
feedbackReceiver.configure .....	98
feedbackReceiver.query .....	99
feedbackReceiver.reconfigure .....	100
feedbackReceiver.remove .....	101
gatekeeper.modify .....	102
gatekeeper.query .....	104
gateway.enumerate .....	106
participant.add .....	107
participant.connect .....	111
participant.diagnostics .....	112
participant.disconnect .....	115
participant.enumerate .....	116
participant.enumerate (deprecated) .....	125
participant.fecc .....	129
participant.message .....	130
participant.modify .....	131
participant.move .....	135
participant.remove .....	136
participant.statistics .....	137
participant.status .....	143
participant.status (deprecated) .....	151
route.add .....	155
route.delete .....	156
route.enumerate .....	157
route.preferences.modify .....	158
route.preferences.query .....	159
services.modify .....	160
services.query .....	162
sip.modify .....	163
sip.query .....	165
streaming.modify .....	166
streaming.query .....	167

template.create .....	168
template.delete .....	172
template.enumerate .....	173
template.modify .....	177
template.status .....	181

## Deprecations

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

Deprecations	Superseded by this command/parameter in newer versions
<code>conference.participant.add</code>	<a href="#">participant.add [p.107]</a>
<code>conference.participant.modify</code>	<a href="#">participant.modify [p.131]</a>
<code>conference.participant.remove</code>	<a href="#">participant.remove [p.136]</a>
<code>conference.query</code>	<a href="#">conference.enumerate [p.48]</a> , <a href="#">participant.enumerate [p.116]</a>
<a href="#">participant.enumerate (deprecated) [p.125]</a>	<a href="#">participant.enumerate [p.116]</a> . This call is not technically deprecated, but there is deprecated behavior if the call does not provide the <code>operationScope</code> parameter.
<a href="#">participant.status (deprecated) [p.151]</a>	<a href="#">participant.status [p.143]</a> . 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>	<a href="#">conference.enumerate [p.48]</a> , <a href="#">device.query [p.91]</a>
<a href="#">participant.diagnostics [p.112]</a>	<a href="#">participant.statistics [p.137]</a> . 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.
<code>registrarUsage</code>	<code>outboundMode</code>
<code>configuredRegistrar</code>	<code>outboundDomain</code>
<code>configuredProxy</code>	<code>outboundAddress</code>
as used in <a href="#">sip.modify [p.163]</a> and <a href="#">sip.query [p.165]</a>	as used in <a href="#">sip.modify [p.163]</a> and <a href="#">sip.query [p.165]</a>

# 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
<b>enumerateID</b>	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. <a href="#">more...</a>

## 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
<b>enumerateID</b>	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. <a href="#">more...</a>
<b>addressBookEntries</b>	array	Each array member is a struct representing a single addressbook entry.
<b>name</b> ( <i>endpoint</i> )	string	The name of the endpoint.
<b>address</b> ( <i>endpoint</i> )	string (63)	The address of the endpoint; may be hostname, IP address, E.164 number, SIP URI, or H.323 ID.
<b>protocol</b> ( <i>signaling</i> )	string	The signaling protocol used in the call. One of <b>h323</b> , <b>sip</b> , or <b>vnc</b> .
<b>gatewayName</b>	string	Present in entries for H.323 endpoints which are configured to use a gateway. This name corresponds to the <b>name</b> parameter of a gateway returned by <b>gateway.enumerate</b> .
<b>gatewayAddress</b>	string (63)	The address of an H.323 gateway, if required. Only used if protocol is <b>h323</b> . This corresponds to the <b>address</b> parameter of the gateway as returned by <b>gateway.enumerate</b> .

<b>dtmfSequence</b>	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 href="#">more...</a>
<b>useSIPRegistrar</b>	boolean	Not valid unless the protocol is SIP. <b>true</b> if the endpoint uses the SIP registrar. Defaults to <b>false</b> .
<b>password</b>	string	The password for VNC endpoints.
<b>portNumber</b>	integer	The port number for VNC endpoints.
<b>callInParams</b>	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.
<b>name (endpoint)</b>	string	The name of the endpoint.
<b>address (endpoint)</b>	string (63)	The address of the endpoint; may be hostname, IP address, E.164 number, SIP URI, or H.323 ID.
<b>e164</b>	string	An E.164 number.
<b>videoToUse</b>	struct	Collection of parameters that uniquely identify the participant whose video will display in place of this participant's video by default.
<b>name (endpoint)</b>	string	The name of the endpoint.
<b>protocol (signaling)</b>	string	The signaling protocol used in the call. One of <b>h323</b> , <b>sip</b> , or <b>vnc</b> .
<b>conferencingParameters</b>	struct	A structure containing the conferencing parameters of the enumerated item, e.g. gateway or endpoint.
<b>useDefaultMotionSharpness</b>	boolean	<b>true</b> means this endpoint will use box-wide default motion sharpness settings.
<b>minFrameRateMotionSharpness</b>	integer	Specifies the minimum frame rate for this endpoint. This parameter is only present if <b>useDefaultMotionSharpness</b> is <b>false</b> .
<b>useDefaultVideoTransmitResolutions</b>	boolean	<b>true</b> means this endpoint will use box-wide default video transmit resolutions.
<b>videoTransmitResolutions</b>	string	Overrides the default setting for video resolution the MCU may send to the endpoint. One of <b>allowAll</b> , <b>4to3Only</b> , <b>4to3WidescreenOverride</b> , or <b>16to9Only</b> . <a href="#">more...</a>
<b>maxMediaTxBitRate</b>	integer	The maximum media transmission speed from this device, in kbps. 0 means the device uses the default.
<b>maxMediaRxBitRate</b>	integer	The maximum media reception speed of this device, in kbps. 0 means the device uses the default.

<code>defaultLayout</code>	string	Describes the participant's default conference view layout if configured. One of <code>default</code> , <code>familyIndex</code> , <code>layoutIndex</code> , <code>conferenceCustom</code> . <a href="#">more...</a>
<code>layoutControlDefault</code>	boolean	<code>true</code> means the endpoint inherits the default layout control setting. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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 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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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.

<b>autoDisconnect</b>	boolean	<p><b>true</b> 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. <b>false</b> means this endpoint requires manual disconnection.</p> <p>When a participant disconnects from a conference and only participants who have <b>autoDisconnect</b> set to <b>true</b> remain, the MCU disconnects all the remaining participants.</p>
<b>borderWidth</b>	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. <a href="#">more...</a>
<b>addAsGuest</b>	boolean	Defines whether the MCU designates guest or chair status to the participant when it invites the participant in to the conference. <b>true</b> means the participant joins as a guest when invited in; <b>false</b> means the participant joins as a chair when invited in.
<b>actAsRecorder</b>	boolean	Defines whether this participant appears as a recorder to other participants.
<b>displayNameOverrideStatus</b>	boolean	<b>true</b> if the endpoint uses the <b>displayNameOverrideValue</b> text to identify itself to other participants.
<b>displayNameOverrideValue</b>	string (63)	This value overrides the participant's display name if <b>displayNameOverrideStatus</b> is <b>true</b> .
<b>suppressAudioDuringDTMF</b>	string	<b>outgoing</b> or <b>all</b> defines which audio the MCU suppresses while it sends the DTMF connection sequence to the endpoint. <a href="#">more...</a>
<b>suppressDtmfEx</b>	string	Controls the muting of in-band DTMF tones. One of <b>fecc</b> , <b>always</b> , or <b>never</b> . <a href="#">more...</a>
<b>customCodecSelection</b>	boolean	Indicates whether the device advertises a custom set of codecs.
<b>customCodecs</b>	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 <b>customCodecSelection</b> is <b>false</b> .
<b>audioTx</b>	struct	A choice of audio codecs advertised by the MCU.
<b>audioRx</b>	struct	A choice of audio codecs received from the participant's endpoint.
<b>g711</b>	boolean	Defines whether or not the device advertises that it will send (or accept) media streams encoded with this codec.

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

---

<b>h264</b>	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
<b>firstIndex</b>	integer	The index of the oldest stored event.
<b>deletableIndex</b>	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.
<b>numEvents</b> ( <i>audit log</i> )	integer	The total number of events stored.
<b>percentageCapacity</b>	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. <a href="#">more...</a>
<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. <a href="#">more...</a>
<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.

# callHome.configure

Configures the MCU to automatically report diagnostic data to the Cisco Call Home service. This feature is disabled by default, but we strongly recommend that you enable it to ensure the best support for your device.

---

**Note:** The MCU currently only supports anonymous reporting.

---

## Input parameters

### Optional inputs

Parameter name	Type	Short description
<code>mode</code>	string	Set the Call Home mode. One of <code>disabled</code> or <code>anonymous</code> . Can only be set to <code>anonymous</code> if the encryption feature key is present. Defaults to <code>disabled</code> if it has never been configured. Omit the parameter to leave the current setting unchanged.
<code>automatic</code>	boolean	Controls automatic Call Home. <code>true</code> enables automatic Call Home. <code>false</code> disables automatic Call Home. Only has effect when mode is <code>anonymous</code> . Omit the parameter to leave the current setting unchanged.

## callHome.query

Queries the MCU to retrieve its Call Home configuration. This feature reports diagnostic data to the Cisco Call Home service.

---

**Note:** The MCU currently only supports anonymous reporting .

---

### Returned data

Parameter name	Type	Short description
<code>mode</code>	string	Call Home mode. One of <b>disabled</b> or <b>anonymous</b> . Defaults to <b>disabled</b> if it has never been configured.
<code>automatic</code>	boolean	<b>true</b> if automatic Call Home is enabled. <b>false</b> if automatic Call Home is disabled. Only has effect if mode is <b>anonymous</b> . Defaults to <b>false</b> if it has never been configured.

## 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
<b>deleteIndex</b> ( <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 <b>cdrlog.query.deleteableIndex</b> , 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>adhocConferenceStarted</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.

---

<b>eventsRemaining</b>	boolean	Whether there is data remaining after this. Provided to avoid putting all data in a single call.
<b>currentTime</b>	dateTime. iso8601	The system's current date and time.
<b>events(CDR)</b>	array	List of the new events; these are structures with some common fields (time, type, index) and other fields specific to the event type.
<b>time (CDR log)</b>	dateTime. iso8601	The date and time when the event was logged, for example 20110119T13:52:42.
<b>type (event)</b>	string	The name of the event type.
<b>index (CDR log)</b>	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
<b>firstIndex</b>	integer	The index of the oldest stored event.
<b>deletableIndex</b>	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.
<b>numEvents</b> ( <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.
<b>percentageCapacity</b>	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 \[p.46\]](#) 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 <b>Visibility</b> 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. <a href="#">more...</a>
-----------------------------	---------	--

<b>numericId</b>	string	The numeric ID of the conference. Used for registration with H.323 gatekeeper / SIP registrar, and to dial in to the conference.
<b>guestNumericId</b>	string	If it is configured, this value is used by guests (instead of <b>numericId</b> ) to access the conference.
<b>registerWithGatekeeper</b>	boolean	Defines whether or not this conference registers its <b>numericId</b> with the H.323 gatekeeper.
<b>registerWithSIPRegistrar</b>	boolean	Defines whether or not this conference registers its <b>numericId</b> with the SIP registrar.
<b>startTime</b>	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.

<b>durationSeconds</b>	integer	The period of time, in seconds, for which this item is active (up to a maximum value of 8639999).
------------------------	---------	---

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).

<b>pin</b>	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.

<b>guestPin</b>	string	Security PIN that a guest can use to gain access to this conference.
<b>description</b>	string	Additional information about the conference.
<b>startLocked</b>	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.
<b>conferenceMeEnabled</b>	boolean	Whether or not ConferenceMe is enabled for this conference.
<b>automaticLectureMode</b>	string	Defines automatic lecture mode. One of <b>type1</b> , <b>type2</b> , or <b>disabled</b> . <a href="#">more...</a>
<b>automaticLectureModeEnabled</b>	boolean	Defines whether automatic lecture mode is enabled for this conference. Deprecated by <b>automaticLectureMode</b> . <a href="#">more...</a>
<b>automaticLectureModeTimeout</b>	integer	If <b>automaticLectureMode</b> is <b>type1</b> , this integer defines the period of time for which a speaker must be talking before lecture mode begins. <a href="#">more...</a>
<b>multicastStreamingEnabled</b>	boolean	Defines whether or not the conference can be multicast.
<b>unicastStreamingEnabled</b>	boolean	Defines whether or not this conference can be unicast to streaming viewers.
<b>contentMode</b>	string	Defines the content mode of the conference. Either <b>disabled</b> , <b>passthrough</b> , <b>transcoded</b> or <b>hybrid</b> . <a href="#">more...</a>

<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>lastChairmanLeavesDisconnect</code>	boolean	Defines whether or not this conference disconnects guests when the last chairperson leaves. Corresponds to the <b>When only guests remain</b> 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. <a href="#">more...</a>
<code>preconfiguredParticipantsDefer</code>	boolean	<code>true</code> if the MCU defers inviting preconfigured participants until at least one other participant is present. <a href="#">more...</a>
<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. <a href="#">more...</a>
<code>contentTxMinimumBitRate</code>	string	The minimum bit rate to use for transmitting content, in bps. 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> , or <code>1500000</code> .
<code>contentPassthroughLimit</code>	string	Possible values of <code>none</code> , <code>sd</code> , <code>hd</code> , and <code>fullhd</code> . This is the minimum resolution passed on to endpoints. If an endpoint that can only receive content below the minimum joins a conference, the endpoint will not be sent a passthrough content stream or a transcoded content stream. If the MCU receives content with a resolution below this setting, then it will use the received resolution as the minimum instead.
<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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> .

<b>weekDays</b>	string	Required if <b>repetition</b> is <b>weekly</b> or <b>everyTwoWeeks</b> . The parameter accepts a comma separated string of weekday names, e.g. <b>monday, wednesday, friday</b> .
<b>terminationType</b>	string	Defines how a repeating conference eventually terminates. One of <b>noTermination</b> , <b>afterNRepeats</b> or <b>endOnGivenDate</b> . <a href="#">more...</a>
<b>terminationDate</b>	dateTime. iso8601	Required if <b>terminationType</b> is <b>endOnGivenDate</b> . This is the date when conference repetition will cease.
<b>numberOfRepeats</b>	integer	Defines the number of times the conference repeats. Required if <b>terminationType</b> is set to <b>afterNRepeats</b> .
<b>customLayoutEnabled</b>	boolean	<b>true</b> if the custom layout is enabled, <b>false</b> otherwise.
<b>layoutControlEx</b>	string	Defines how the view layout can be controlled. One of <b>disabled</b> , <b>feccOnly</b> , <b>dtmfOnly</b> , <b>feccWithDtmfFallback</b> , or <b>bothFeccAndDtmf</b> . <a href="#">more...</a>
<b>cameraControl</b>	string	Defines how the endpoint camera(s) within your API call's context can be controlled. If present, it may be <b>disabled</b> , <b>feccOnly</b> , <b>dtmfOnly</b> , <b>feccWithDtmfFallback</b> , <b>bothFeccAndDtmf</b> , or <b>default</b> . <a href="#">more...</a>
<b>newParticipantsCustomLayout</b>	boolean	<b>true</b> if new participants use the custom layout, <b>false</b> otherwise. Only valid if <b>customLayoutEnabled</b> is <b>true</b> .
<b>customLayout</b>	integer	The index of the layout associated with the conference. This is seen by participants if they are using the conference custom layout.  See <a href="#">Conference layouts [p.193]</a> for a list of available layouts and corresponding index values.
<b>chairControl</b>	string	The chair control setting for this conference. One of <b>none</b> , <b>floorControlOnly</b> , or <b>chairAndFloorControl</b> . <a href="#">more...</a>
<b>suppressDtmfEx</b>	string	Controls the muting of in-band DTMF tones. One of <b>fecc</b> , <b>always</b> , or <b>never</b> . <a href="#">more...</a>
<b>inCallMenuControlChair</b>	string	Defines the level of control a chairperson has over the in call menu. One of <b>off</b> , <b>local</b> , <b>conference</b> , or <b>advanced</b> . <a href="#">more...</a>
<b>inCallMenuControlGuest</b>	string	Defines the level of control a guest has over the in call menu. Either <b>off</b> or <b>local</b> . <a href="#">more...</a>
<b>automaticLectureModeEnabled</b>	boolean	Defines whether automatic lecture mode is enabled for this conference. Deprecated by <b>automaticLectureMode</b> . <a href="#">more...</a>
<b>automaticLectureModeTimeout</b>	integer	If <b>automaticLectureMode</b> is <b>type1</b> , this integer defines the period of time for which a speaker must be talking before lecture mode begins. <a href="#">more...</a>
<b>encryptionRequired</b>	boolean	The encryption setting for this conference, if the encryption feature key is enabled. If <b>true</b> , encryption is required for this conference. Otherwise, encryption is optional.
<b>contentContribution</b>	boolean	Defines whether or not endpoints are permitted to contribute the content channel to this conference. <b>true</b> if content contribution is enabled.

<b>contentTransmitResolutions</b>	string	The resolution for the content channel that will be transmitted to endpoints in this conference. One of <b>4to3Only</b> , <b>16to9Only</b> , or <b>allowAll</b> . <a href="#">more...</a>
-----------------------------------	--------	---

## Deprecated parameters

Parameter name	Type	Short description
<b>dtmfMuteControl</b>	boolean	Deprecated by <b>inCallMenuControlChair</b> and <b>inCallMenuControlGuest</b> . Defines whether or not a participant can mute audio by pressing *6 on the remote control.
<b>conferenceID</b>	string	Deprecated by <b>numericId</b> .
<b>endTime</b>	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 <b>durationSeconds</b> instead.
<b>layoutControlEnabled</b>	boolean	Deprecated by <b>layoutControlEx</b> . Defines whether the endpoint's participant will have control over the layout if <b>layoutControlDefault</b> is <b>false</b> . <a href="#">more...</a>

# 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 \[p.47\]](#)).

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 \[p.46\]](#) 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. <a href="#">more...</a>
<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	<code>true</code> 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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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 <b>disabled</b> , <b>passthrough</b> , <b>transcoded</b> or <b>hybrid</b> . <a href="#">more...</a>
<code>h239Enabled</code>	boolean	Deprecated by <code>contentMode</code> . If you set <code>h239Enabled</code> to <b>true</b> , <code>contentMode</code> will be set to <b>transcoded</b> . If you set <code>h239Enabled</code> to <b>false</b> , <code>contentMode</code> will be set to <b>disabled</b> .
<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 <b>h263+</b> , <b>h264</b> , or <b>automatic</b> (default). This setting does not apply in <b>passthrough</b> mode. <a href="#">more...</a>
<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>contentPassthroughLimit</code>	string	Possible values of <b>none</b> , <b>sd</b> , <b>hd</b> , and <b>fullhd</b> . This is the minimum resolution passed on to endpoints. If an endpoint that can only receive content below the minimum joins a conference, the endpoint will not be sent a passthrough content stream or a transcoded content stream. If the MCU receives content with a resolution below this setting, then it will use the received resolution as the minimum instead.
<code>lastChairmanLeavesDisconnect</code>	boolean	Defines whether or not this conference disconnects guests when the last chairperson leaves. Corresponds to the <b>When only guests remain</b> conference setting in the web UI.
<code>preconfiguredParticipantsDefer</code>	boolean	<b>true</b> if the MCU defers inviting preconfigured participants until at least one other participant is present. <a href="#">more...</a>
<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. <a href="#">more...</a>
<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 layout associated with the conference. This is seen by participants if they are using the conference custom layout.  See <a href="#">Conference layouts [p.193]</a> 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 <b>Visibility</b> 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> . <a href="#">more...</a>
<code>suppressDtmfEx</code>	string	Controls the muting of in-band DTMF tones. One of <code>fecc</code> , <code>always</code> , or <code>never</code> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<code>automaticLectureMode</code>	string	Defines automatic lecture mode. One of <code>type1</code> , <code>type2</code> , or <code>disabled</code> . <a href="#">more...</a>

<b>automaticLectureModeEnabled</b>	boolean	Defines whether automatic lecture mode is enabled for this conference. Deprecated by <b>automaticLectureMode</b> . <a href="#">more...</a>
<b>automaticLectureModeTimeout</b>	integer	If <b>automaticLectureMode</b> is <b>type1</b> , this integer defines the period of time for which a speaker must be talking before lecture mode begins. <a href="#">more...</a>
<b>encryptionRequired</b>	boolean	The encryption setting for this conference, if the encryption feature key is enabled. If <b>true</b> , encryption is required for this conference. Otherwise, encryption is optional.
<b>contentContribution</b>	boolean	Defines whether or not endpoints are permitted to contribute the content channel to this conference. <b>true</b> if content contribution is enabled.
<b>floorStatus</b>	string	One of <b>inactive</b> , <b>active</b> , or <b>assigned</b> . If it is <b>active</b> or <b>assigned</b> , a <b>floorParticipant</b> struct will be included in the response.
<b>floorParticipant</b>	struct	A structure that identifies which participant has the floor.
<b>participantName</b>	string	The unique name of a participant. <a href="#">more...</a>
<b>participantProtocol</b>	string	<b>h323</b> , <b>sip</b> , or <b>vnc</b> .
<b>participantType</b>	string	One of: <b>by_address</b> , <b>by_name</b> , or <b>ad_hoc</b> . <a href="#">more...</a>
<b>conferenceName</b>	string	The name of the conference.
If the participant is in a conference, the response includes the <b>conferenceName</b> ; if the participant is in an autoattendant, the response includes the <b>autoAttendantUniqueId</b> instead. The response will not include both parameters.		
<b>autoAttendantUniqueId</b>	string	Unique identifier for the auto attendant.
<b>connectionUniqueId</b>	integer	Corresponds to the <b>uniqueId</b> returned by a conference or autoattendant.
<b>chairParticipant</b>	struct	A structure containing parameters that uniquely identify the participant who is the chairperson.
<b>participantName</b>	string	The unique name of a participant. <a href="#">more...</a>
<b>participantProtocol</b>	string	<b>h323</b> , <b>sip</b> , or <b>vnc</b> .
<b>participantType</b>	string	One of: <b>by_address</b> , <b>by_name</b> , or <b>ad_hoc</b> . <a href="#">more...</a>
<b>conferenceName</b>	string	The name of the conference.
If the participant is in a conference, the response includes the <b>conferenceName</b> ; if the participant is in an autoattendant, the response includes the <b>autoAttendantUniqueId</b> instead. The response will not include both parameters.		
<b>autoAttendantUniqueId</b>	string	Unique identifier for the auto attendant.

<b>connectionUniqueId</b>	integer	Corresponds to the <code>uniqueId</code> returned by a conference or autoattendant.
---------------------------	---------	---

### Conditionally returned for `scheduled` conferences only:

Parameter name	Type	Short description
<b>startTime</b>	dateTime. iso8601	Start time of the item, e.g. 20110106T14:00:00.
<b>durationSeconds</b>	integer	The period of time, in seconds, for which this item is active (up to a maximum value of 8639999).
<b>repetition</b>	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> . <a href="#">more...</a>
<b>weekDay</b>	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.
<b>whichWeek</b>	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> .
<b>weekDays</b>	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> .
<b>terminationType</b>	string	Defines how a repeating conference eventually terminates. One of <code>noTermination</code> , <code>afterNRepeats</code> or <code>endOnGivenDate</code> . <a href="#">more...</a>
<b>terminationDate</b>	dateTime. iso8601	Required if <code>terminationType</code> is <code>endOnGivenDate</code> . This is the date when conference repetition will cease.

### Conditionally returned for `active` conferences only:

Parameter name	Type	Short description
<b>activeStartTime</b>	dateTime. iso8601	If the conference is currently active, this parameter contains the time that the current session started.
<b>activeEndTime</b>	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.

---

<b>activeConferenceId</b>	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 <b>activeConferenceId</b> .
---------------------------	--------	--

---

## Deprecated parameters

---

Parameter name	Type	Short description
<b>dtmfMuteControl</b>	boolean	Deprecated by <b>inCallMenuControlChair</b> and <b>inCallMenuControlGuest</b> . 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. <a href="#">more...</a>
<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> . <a href="#">more...</a>

## 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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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 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. <a href="#">more...</a>
<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 (up to a maximum value of 8639999).
<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 <b>disabled</b> , <b>passthrough</b> , <b>transcoded</b> or <b>hybrid</b> . <a href="#">more...</a>
<code>h239Enabled</code>	boolean	Deprecated by <code>contentMode</code> . If you set <code>h239Enabled</code> to <b>true</b> , <code>contentMode</code> will be set to <b>transcoded</b> . If you set <code>h239Enabled</code> to <b>false</b> , <code>contentMode</code> will be set to <b>disabled</b> .
<code>contentTxCodec</code>	string	The codec used to transmit content. If content is being transcoded, it is the output format of the transcoder; either <b>h263+</b> , <b>h264</b> , or <b>automatic</b> (default). This setting does not apply in <b>passthrough</b> mode. <a href="#">more...</a>
<code>contentTxMinimumBitRate</code>	string	The minimum bit rate to use for transmitting content, in bps. One of: <b>0</b> , <b>64000</b> , <b>128000</b> , <b>192000</b> , <b>256000</b> , <b>320000</b> , <b>384000</b> , <b>512000</b> , <b>768000</b> , <b>1000000</b> , <b>1250000</b> , or <b>1500000</b> .
<code>contentPassthroughLimit</code>	string	Possible values of <b>none</b> , <b>sd</b> , <b>hd</b> , and <b>fullhd</b> . This is the minimum resolution passed on to endpoints. If an endpoint that can only receive content below the minimum joins a conference, the endpoint will not be sent a passthrough content stream or a transcoded content stream. If the MCU receives content with a resolution below this setting, then it will use the received resolution as the minimum instead.
<code>conferenceMeEnabled</code>	boolean	Whether or not ConferenceMe is enabled for this conference.
<code>preconfiguredParticipantsDefer</code>	boolean	<b>true</b> if the MCU defers inviting preconfigured participants until at least one other participant is present. <a href="#">more...</a>
<code>lastChairmanLeavesDisconnect</code>	boolean	Defines whether or not this conference disconnects guests when the last chairperson leaves. Corresponds to the <b>When only guests remain</b> conference setting in the web UI.
<code>private</code>	boolean	Defines whether the conference is public or private. <b>true</b> if the conference is private. Corresponds to the <b>Visibility</b> 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. <a href="#">more...</a>
<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 <b>none</b> , <b>daily</b> , <b>weekly</b> , <b>everyTwoWeeks</b> , or <b>monthly</b> . <a href="#">more...</a>
<code>weekDay</code>	string	Must be present if <code>repetition</code> is <b>monthly</b> . One of <b>monday</b> , <b>tuesday</b> , <b>wednesday</b> , <b>thursday</b> , <b>friday</b> , <b>saturday</b> or <b>sunday</b> . Note that if <code>repetition</code> is not <b>weekly</b> or <b>everyTwoWeeks</b> , the <code>weekDays</code> parameter should be used.

<b>whichWeek</b>	string	Required if <b>repetition</b> is <b>monthly</b> . Defines which week the repeating conference will fall in; one of <b>first</b> , <b>second</b> , <b>third</b> , <b>fourth</b> , or <b>last</b> .
<b>weekDays</b>	string	Required if <b>repetition</b> is <b>weekly</b> or <b>everyTwoWeeks</b> . The parameter accepts a comma separated string of weekday names, e.g. <b>monday</b> , <b>wednesday</b> , <b>friday</b> .
<b>terminationType</b>	string	Defines how a repeating conference eventually terminates. One of <b>noTermination</b> , <b>afterNRepeats</b> or <b>endOnGivenDate</b> . <a href="#">more...</a>
<b>terminationDate</b>	dateTime. iso8601	Required if <b>terminationType</b> is <b>endOnGivenDate</b> . This is the date when conference repetition will cease.
<b>numberOfRepeats</b>	integer	Defines the number of times the conference repeats. Required if <b>terminationType</b> is set to <b>afterNRepeats</b> .
<b>contentImportant</b>	boolean	Whether or not content is set to be important.
<b>h239Important</b>	boolean	Whether the H.239 channel is set to be important. Consider this setting deprecated by <b>contentImportant</b> . The setting will still work however, even if the content channel is SIP or VNC or content from a main video participant.
<b>locked</b>	boolean	Defines whether or not the conference is locked.
<b>startLocked</b>	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.
<b>layoutControlEx</b>	string	Defines how the view layout can be controlled. One of <b>disabled</b> , <b>feccOnly</b> , <b>dtmfOnly</b> , <b>feccWithDtmfFallback</b> , or <b>bothFeccAndDtmf</b> . <a href="#">more...</a>
<b>cameraControl</b>	string	Defines how the endpoint camera(s) within your API call's context can be controlled. If present, it may be <b>disabled</b> , <b>feccOnly</b> , <b>dtmfOnly</b> , <b>feccWithDtmfFallback</b> , <b>bothFeccAndDtmf</b> , or <b>default</b> . <a href="#">more...</a>
<b>newParticipantsCustomLayout</b>	boolean	<b>true</b> if new participants use the custom layout, <b>false</b> otherwise. Only valid if <b>customLayoutEnabled</b> is <b>true</b> .
<b>customLayout</b>	integer	The index of the layout associated with the conference. This is seen by participants if they are using the conference custom layout.  See <a href="#">Conference layouts [p.193]</a> for a list of available layouts and corresponding index values.
<b>chairControl</b>	string	The chair control setting for this conference. One of <b>none</b> , <b>floorControlOnly</b> , or <b>chairAndFloorControl</b> . <a href="#">more...</a>
<b>enforceMaximumAudioPorts</b>	boolean	Defines whether the conference enforces the <b>maximumAudioPorts</b> limit. Assumed to be <b>true</b> if absent.
<b>enforceMaximumVideoPorts</b>	boolean	Defines whether the conference enforces the <b>maximumVideoPorts</b> limit. Assumed to be <b>true</b> if absent.
<b>suppressDtmfEx</b>	string	Controls the muting of in-band DTMF tones. One of <b>fecc</b> , <b>always</b> , or <b>never</b> . <a href="#">more...</a>
<b>inCallMenuControlChair</b>	string	Defines the level of control a chairperson has over the in call menu. One of <b>off</b> , <b>local</b> , <b>conference</b> , or <b>advanced</b> . <a href="#">more...</a>

<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> . <a href="#">more...</a>
<code>automaticLectureMode</code>	string	Defines automatic lecture mode. One of <code>type1</code> , <code>type2</code> , or <code>disabled</code> . <a href="#">more...</a>
<code>automaticLectureModeEnabled</code>	boolean	Defines whether automatic lecture mode is enabled for this conference. Deprecated by <code>automaticLectureMode</code> . <a href="#">more...</a>
<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. <a href="#">more...</a>
<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>customLayoutEnabled</code>	boolean	Defines whether custom layout is enabled. <code>true</code> if the custom layout is enabled, <code>false</code> otherwise.
<code>setAllParticipantsToCustomLayout</code>	boolean	<code>true</code> sets all participants to immediately see the conference custom layout. If <code>false</code> nothing happens. Only valid if <code>customLayoutEnabled</code> is <code>true</code>

## 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>oldConferenceName</code>	string	Deprecated conference renaming scheme - new code should use <code>conferenceName</code> and <code>newConferenceName</code> as above.
<code>conferenceName</code>	string	The name of the conference.
<code>conferenceID</code>	string	Deprecated by <code>numericId</code> .
<code>endTime</code>	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 <code>durationSeconds</code> instead.
<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> . <a href="#">more...</a>

# 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> . <a href="#">more...</a>

### 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> . <a href="#">more...</a>
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>participantName</code>	string	The unique name of a participant. <a href="#">more...</a>

## Returned data

### Always returned

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

---

<b>Parameter name</b>	<b>Type</b>	<b>Short description</b>
<b>panesModified</b>	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> . <a href="#">more...</a>

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

<code>participantName</code>	string	The unique name of a participant. <a href="#">more...</a>
<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> . <a href="#">more...</a>



# conference.resetCleanupTimeout

Resets the cleanup timeout on the named conference.

## Input parameters

Parameter name	Type	Short description
conferenceName	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> . <a href="#">more...</a>
<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. <a href="#">more...</a>
<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 <b>When only guests remain</b> 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. <a href="#">more...</a>
<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. <a href="#">more...</a>
<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 layout associated with the conference. This is seen by participants if they are using the conference custom layout. See <a href="#">Conference layouts [p.193]</a> 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 <b>Visibility</b> setting on the web UI, which can have the value <i>Public</i> or <i>Private</i> .
<code>joinAGC</code>	boolean	Whether AGC should be used by default for participants joining this conference
<code>chairControl</code>	string	The chair control setting for this conference. One of <code>none</code> , <code>floorControlOnly</code> , or <code>chairAndFloorControl</code> . <a href="#">more...</a>
<code>suppressDtmfEx</code>	string	Controls the muting of in-band DTMF tones. One of <code>fecc</code> , <code>always</code> , or <code>never</code> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<code>automaticLectureMode</code>	string	Defines automatic lecture mode. One of <code>type1</code> , <code>type2</code> , or <code>disabled</code> . <a href="#">more...</a>
<code>automaticLectureModeEnabled</code>	boolean	Defines whether automatic lecture mode is enabled for this conference. Deprecated by <code>automaticLectureMode</code> . <a href="#">more...</a>
<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. <a href="#">more...</a>
<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>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.
<code>floorParticipant</code>	struct	A structure that identifies which participant has the floor.
<code>participantName</code>	string	The unique name of a participant. <a href="#">more...</a>

<b>participantProtocol</b>	string	h323, sip, or vnc.
<b>participantType</b>	string	One of: <b>by_address</b> , <b>by_name</b> , or <b>ad_hoc</b> . <a href="#">more...</a>
<b>conferenceName</b>	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.

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

<b>chairParticipant</b>	struct	A structure containing parameters that uniquely identify the participant who is the chairperson.
-------------------------	--------	--

<b>participantName</b>	string	The unique name of a participant. <a href="#">more...</a>
<b>participantProtocol</b>	string	h323, sip, or vnc.
<b>participantType</b>	string	One of: <b>by_address</b> , <b>by_name</b> , or <b>ad_hoc</b> . <a href="#">more...</a>
<b>conferenceName</b>	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.

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

### Conditionally returned for scheduled conferences only:

Parameter name	Type	Short description
<b>startTime</b>	dateTime. iso8601	Start time of the item, e.g. 20110106T14:00:00.
<b>durationSeconds</b>	integer	The period of time, in seconds, for which this item is active (up to a maximum value of 8639999).
<b>repetition</b>	string	Defines the repetition frequency of a scheduled conference. One of <b>none</b> , <b>daily</b> , <b>weekly</b> , <b>everyTwoWeeks</b> , or <b>monthly</b> . <a href="#">more...</a>
<b>weekDay</b>	string	Must be present if <b>repetition</b> is <b>monthly</b> . One of <b>monday</b> , <b>tuesday</b> , <b>wednesday</b> , <b>thursday</b> , <b>friday</b> , <b>saturday</b> or <b>sunday</b> . Note that if <b>repetition</b> is not <b>weekly</b> or <b>everyTwoWeeks</b> , the <b>weekDays</b> parameter should be used.
<b>whichWeek</b>	string	Required if <b>repetition</b> is <b>monthly</b> . Defines which week the repeating conference will fall in; one of <b>first</b> , <b>second</b> , <b>third</b> , <b>fourth</b> , or <b>last</b> .

<b>weekDays</b>	string	Required if <b>repetition</b> is <b>weekly</b> or <b>everyTwoWeeks</b> . The parameter accepts a comma separated string of weekday names, e.g. <b>monday, wednesday, friday</b> .
<b>terminationType</b>	string	Defines how a repeating conference eventually terminates. One of <b>noTermination</b> , <b>afterNRepeats</b> or <b>endOnGivenDate</b> . <a href="#">more...</a>
<b>terminationDate</b>	dateTime. iso8601	Required if <b>terminationType</b> is <b>endOnGivenDate</b> . This is the date when conference repetition will cease.

### Conditionally returned for active conferences only:

Parameter name	Type	Short description
<b>activeStartTime</b>	dateTime. iso8601	If the conference is currently active, this parameter contains the time that the current session started.
<b>activeEndTime</b>	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.
<b>activeConferenceId</b>	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 <b>activeConferenceId</b> .

## Deprecated parameters

Parameter name	Type	Short description
<b>dtmfMuteControl</b>	boolean	Deprecated by <b>inCallMenuControlChair</b> and <b>inCallMenuControlGuest</b> . 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 is the layout for the video sent to the participant. Refer to <a href="#">Conference layouts [p.193]</a> for details.
<code>borderWidth</code>	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. <a href="#">more...</a>
<code>focusType</code>	string	Indicates the endpoint's focus. One of <code>participant</code> , <code>voiceActivated</code> , or <code>h239</code> . <a href="#">more...</a>

### 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. <a href="#">more...</a>
<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> . <a href="#">more...</a>

# 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. Refer to <a href="#">Conference layouts [p.193]</a> 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> . <a href="#">more...</a>
<code>borderWidth</code>	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. <a href="#">more...</a>
<code>focusType</code>	string	Indicates the endpoint's focus. One of <code>participant</code> , <code>voiceActivated</code> , or <code>h239</code> . <a href="#">more...</a>

## Conditionally returned

### focusParticipant struct

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

Parameter name	Type	Short description
<b>focusParticipant</b>	struct	The structure contains participant parameters that identify which participant displays in the largest pane if <b>focusType</b> is <b>participant</b> .
<b>participantName</b>	string	The unique name of a participant. <a href="#">more...</a>
<b>participantProtocol</b>	string	<b>h323</b> , <b>sip</b> , or <b>vnc</b> .
<b>participantType</b>	string	One of: <b>by_address</b> , <b>by_name</b> , or <b>ad_hoc</b> . <a href="#">more...</a>
<b>conferenceName</b>	string	The name of the conference.
If the participant is in a conference, the response includes the <b>conferenceName</b> ; if the participant is in an autoattendant, the response includes the <b>autoAttendantUniqueId</b> instead. The response will not include both parameters.		
<b>autoAttendantUniqueId</b>	string	Unique identifier for the auto attendant.
<b>connectionUniqueId</b>	integer	Corresponds to the <b>uniqueId</b> 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
<b>audioStreams</b>	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:

<b>codec</b>	string	The codec in use, or <b>other</b> for undefined codecs.
<b>count</b>	integer	The number of users of this codec.
<b>videoStreams</b>	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:

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

Adds a feature/license key to the MCU.

## Input parameters

### Required inputs

Parameter name	Type	Short description
key	string	The feature/license key to add/remove.

# device.features.remove

Removes a feature/license key from the MCU.

## Input parameters

### Required inputs

Parameter name	Type	Short description
key	string	The feature/license key to add/remove.

## 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. (Not returned on the MCU 5300 Series.)
<code>videoLoad</code>	integer	A percentage value representing the proportion of the device's video processing capacity that is currently in use. (Not returned on the MCU 5300 Series.)
<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> . <a href="#">more...</a>
<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 (79)	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 (79)	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.

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

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

#### Returned only if the interface is enabled and configured:

Parameter name	Type	Short description
<b>dhcpv4</b>	boolean	Defines whether or not to use DHCP to obtain an IPv4 address. Deprecates <b>dhcp</b> .
<b>ipv4Address</b>	string (79)	IPv4 address in dotted-quad format.
<b>ipv4SubnetMask</b>	string (31)	The IPv4 subnet mask in dotted quad format. Deprecates <b>subnetMask</b> .
<b>defaultIpv4Gateway</b>	string (79)	The device's IPv4 default gateway in dotted quad format. Deprecates <b>defaultGateway</b> .
<b>domainName</b>	string (255)	The domain name (DNS suffix).
<b>nameServer</b>	string (79)	The IP address of the name server, in dotted quad format (IPv4) or CIDR format (IPv6).
<b>nameServerSecondary</b>	string (79)	The IP address of the secondary name server, in dotted quad format (IPv4) or CIDR format (IPv6).
<b>ipv6Conf</b>	string (10)	Indicates how the IPv6 address is assigned; either <b>automatic</b> (by SLAAC/DHCPv6) or <b>manual</b> .
<b>ipv6Address</b>	string (79)	The IPv6 address in CIDR format.
<b>ipv6PrefixLength</b>	integer	The length of the IPv6 address prefix.
<b>defaultIpv6Gateway</b>	string (79)	The address of the IPv6 default gateway in CIDR format.
<b>linkLocalIpv6Address</b>	string(63)	The link local IPv6 address in CIDR format.
<b>linkLocalIpv6PrefixLength</b>	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 date and time.
<code>restartTime</code>	dateTime. iso8601	The system's date and time when it started.
<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, <b>Encryption</b> . <a href="#">more...</a>
<code>activatedLicenses</code>	array	Array of activated licenses (i.e. not expired and ports number>0) (optional, only present if licenses installed and active) <a href="#">more...</a>
<code>license</code>	string	License name
<code>ports</code>	integer	Number of license ports installed
<code>expiry</code>	Date Time	(optional) Expiry date when present. Time is set at 23:59:59
<code>clusterType</code>	string	The role that this MCU plays in a cluster. One of <b>master</b> , <b>slave</b> , or <b>unclustered</b> . 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 <b>enabled</b> or <b>disabled</b> . Corresponds to the Media port reservation setting on the web interface. Only present on MCU products.
<code>maxVideoResolution</code>	string	Either <code>cif</code> or <code>4cif</code> . <a href="#">more...</a>

<b>videoPortAllocation</b>	array	An array of <b>structs</b> , each of which defines the type and count of video ports that are allocated on this MCU.
<b>type</b> ( <i>videoports</i> )	string	One of <b>nhd</b> , <b>sd</b> , <b>hd</b> , <b>hdPlus</b> or <b>fullhd</b>
<b>count</b> ( <i>videoports</i> )	integer	The allocated number of video ports of this <b>type</b> .
<b>shutdownStatus</b>	string	Indicates the status of a shutdown operation. One of <b>shutdown</b> , <b>shutdownInProgress</b> , or <b>notShutdown</b> .
<b>rebootRequired</b>	boolean	The device returns this parameter as <b>true</b> if it needs to reboot. <a href="#">more...</a>
<b>finishedBooting</b>	boolean	<b>true</b> after the device is fully booted. Will not revert to <b>false</b> until a reboot starts.
<b>mediaResources</b>	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.
<b>mediaResourceRestarts</b>	integer	The number of DSP/compute card restarts that have occurred since the unit last booted

# 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 date and time.
<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 date and time.
<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
<b>receiverURI</b>	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 <a href="#">XML-RPC.com</a> ). 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
<b>sourceIdentifier</b>	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.

<b>receiverIndex</b>	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.*

<b>events (feedback)</b>	struct	Each member of the <b>events</b> struct associates a string (feedback event name) to a boolean (true to subscribe). <a href="#">events (feedback) [p.231]</a>
--------------------------	--------	---

## Returned data

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

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
<code>receivers</code>	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
<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 <a href="http://XML-RPC.com">XML-RPC.com</a> ). 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.

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

<code>index (feedback receiver)</code>	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 <a href="#">XML-RPC.com</a> ). 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 (feedback)</code>	struct	Each member of the <code>events</code> struct associates a string (feedback event name) to a boolean (true to subscribe). <a href="#">events (feedback) [p.231]</a>

# 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
<b>receiverIndex</b>	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 <b>disabled</b> , <b>enabled</b> , or <b>required</b> . <a href="#">more...</a>
<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 <b>gateway</b> , <b>terminalGateway</b> , <b>gatewayCisco</b> , <b>mcuStandard</b> , or <b>mcuCompatible</b> . <a href="#">more...</a>
<code>portAssociationAv4</code>	boolean	<b>true</b> if interface 'PortA IPv4' is associated with the H.323 gatekeeper.
<code>portAssociationAv6</code>	boolean	<b>true</b> if interface 'PortA IPv6' is associated with the H.323 gatekeeper.
<code>portAssociationBv4</code>	boolean	<b>true</b> if interface 'PortB IPv4' is associated with the H.323 gatekeeper.
<code>portAssociationBv6</code>	boolean	<b>true</b> if interface 'PortB IPv6' is associated with the H.323 gatekeeper.
<code>h323ID</code>	string	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	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 <b>enabled</b> or <b>disabled</b> . Corresponds to the <b>ID registration for scheduled conferences</b> 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 <b>all</b> in a <code>gatekeeper.modify</code> call.

---

<b>availabilityThresholdVideoPorts</b>	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 <b>all</b> in a <b>gatekeeper.modify</b> 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 <b>disabled</b> , <b>enabled</b> , or <b>required</b> . <a href="#">more...</a>

### Conditionally returned

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

Parameter name	Type	Short description
<code>address</code> ( <i>gatekeeper</i> )	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 <b>inProgress</b> , <b>resolved</b> , or <b>failed</b> .
<code>ip</code>	string	the IP address of the gatekeeper (if <code>dnsStatus</code> is <b>resolved</b> )
<code>activeRegistrations</code>	integer	The number of active registrations.
<code>pendingRegistrations</code>	integer	The number of registrations in progress
<code>registrationPrefix</code>	string	A string of digits that serves as the device's registration prefix.
<code>registrationType</code>	string	The gatekeeper registration type. One of <b>gateway</b> , <b>terminalGateway</b> , <b>gatewayCisco</b> , <b>mcuStandard</b> , or <b>mcuCompatible</b> . <a href="#">more...</a>
<code>portAssociationAv4</code>	boolean	<b>true</b> if interface 'PortA IPv4' is associated with the H.323 gatekeeper.
<code>portAssociationAv6</code>	boolean	<b>true</b> if interface 'PortA IPv6' is associated with the H.323 gatekeeper.
<code>portAssociationBv4</code>	boolean	<b>true</b> if interface 'PortB IPv4' is associated with the H.323 gatekeeper.
<code>portAssociationBv6</code>	boolean	<b>true</b> 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	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 <b>ID registration for scheduled conferences</b> option on the web interface.
<code>h323IDStatus</code>	string	The current status of the ID registration process. <a href="#">more...</a>
<code>mcuServicePrefixStatus</code>	string	The current status of the service prefix registration process. <a href="#">more...</a>
<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.

**Note:** `gateway.enumerate` is not supported on the MCU 5300 Series.

## 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. <a href="#">more...</a>

## 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. <a href="#">more...</a>
<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	<code>true</code> 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 <code>useDefaultMotionSharpness</code> is <code>false</code> .
<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 \[p.136\]](#).

---

**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
<code>conferenceName</code>	string	The name of the conference.
<code>participantName</code>	string	The unique name of a participant. <a href="#">more...</a>

### 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
<code>addResponse</code>	boolean	<code>true</code> to return the details of the added 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> or <code>ad_hoc</code> . <a href="#">more...</a>
<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>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>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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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 (63)	This value overrides the participant's display name if <code>displayNameOverrideStatus</code> is <code>true</code> .
<code>cpLayout</code>	string	This is the layout for the video sent to the participant.
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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.

<b>videoRxMuted</b>	boolean	<b>true</b> means that video from this participant will not be seen by other conference participants.
<b>videoTxMuted</b>	boolean	<b>true</b> means that the MCU does not send the video part of the conference to this participant.
<b>videoTxWidescreen</b>	boolean	If <b>true</b> , the MCU sends video in a form suitable for a widescreen 16:9 display to this participant.
<b>videoTxMaxResolution</b>	string	The maximum resolution transmitted to this endpoint. One of <b>cif</b> , <b>4cif</b> , or <b>max</b> . <a href="#">more...</a>
<b>videoRxMaxResolution</b>	string	The maximum resolution of the received video. One of <b>cif</b> , <b>4cif</b> , or <b>max</b> . <a href="#">more...</a>
<b>autoConnect</b>	boolean	<b>true</b> allows endpoints to automatically connect to this conference when they dial in and are recognized. <a href="#">more...</a>
<b>autoDisconnect</b>	boolean	<b>true</b> 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. <b>false</b> means this endpoint requires manual disconnection.  When a participant disconnects from a conference and only participants who have <b>autoDisconnect</b> set to <b>true</b> remain, the MCU disconnects all the remaining participants.
<b>borderWidth</b>	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. <a href="#">more...</a>
<b>dtmfSequence</b>	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 href="#">more...</a>
<b>suppressAudioDuringDTMF</b>	string	<b>outgoing</b> or <b>all</b> defines which audio the MCU suppresses while it sends the DTMF connection sequence to the endpoint. <a href="#">more...</a>
<b>linkType</b>	string	This parameter is ignored unless <b>participantType</b> is <b>by_address</b> . Either <b>cascadeSlaveToMaster</b> or <b>default</b>
<b>suppressDtmfEx</b>	string	Controls the muting of in-band DTMF tones. One of <b>fecc</b> , <b>always</b> , or <b>never</b> . <a href="#">more...</a>
<b>h239Negotiation</b>	string	Defines how the MCU presents itself for h239 token negotiation. One of <b>As master</b> , <b>As slave</b> , or <b>Mimic slave</b> . <a href="#">more...</a>
<b>videoToUse</b>	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:

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

---

**toOverride** string (63) This value overrides the SIP To-URI.

---

## Returned data

### Conditionally returned

Parameter name	Type	Short description
<b>participant</b>	struct	Contains the parameters that, when considered together, uniquely identify a participant.
<b>participantName</b>	string	The unique name of a participant. <a href="#">more...</a>
<b>participantType</b>	string	One of: <b>by_address</b> or <b>ad_hoc</b> . <a href="#">more...</a>
<b>participantProtocol</b>	string	<b>h323</b> , <b>sip</b> , or <b>vnc</b> .
<b>conferenceName</b>	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 does not include both parameters.

---

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

---

### Deprecated parameters

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

---

# 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. <a href="#">more...</a>
<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> . <a href="#">more...</a>

# 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. <a href="#">more...</a>
<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> . <a href="#">more...</a>

## 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. <a href="#">more...</a>
<code>videoTxWidth</code>	integer	Width in pixels of the transmitted video.
<code>videoTxHeight</code>	integer	Height in pixels of the transmitted video.

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

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

## 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. <a href="#">more...</a>
<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> . <a href="#">more...</a>

# 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 \(deprecated\) \[p.125\]](#) 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> . <a href="#">more...</a>

### 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. <a href="#">more...</a>
<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.

<b>connected</b>	boolean	<b>true</b> if the participant is currently connected to a conference.
<b>disconnected</b>	boolean	<b>true</b> 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 dialed 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
<b>enumerateID</b>	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. <a href="#">more...</a>
<b>currentRevision</b>	integer	A number that indicates the current revision of this enumeration. You can use this as a <b>lastRevision</b> input to a future <b>enumerate</b> call to retrieve only the changes between the two enumerations.
<b>participants</b>	array	An array of structures that represent participants.

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

<b>participantName</b>	string	The unique name of a participant. <a href="#">more...</a>
<b>participantProtocol</b>	string	<b>h323</b> , <b>sip</b> , or <b>vnc</b> .
<b>participantType</b>	string	One of: <b>by_address</b> , <b>by_name</b> , or <b>ad_hoc</b> . <a href="#">more...</a>
<b>conferenceName</b>	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.

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

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.

---

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

---

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
<b>address</b> ( <i>endpoint</i> )	string (63)	The address of the endpoint; may be hostname, IP address, E.164 number, SIP URI, or H.323 ID.
<b>gatewayAddress</b>	string (63)	The address of an H.323 gateway, if required. Only used if protocol is <b>h323</b> . This corresponds to the <b>address</b> parameter of the gateway as returned by <b>gateway.enumerate</b> .
<b>ipAddress</b>	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.
<b>displayName</b>	string	The display name of the participant.  If this parameter is longer than 31 characters, only the first 31 characters are returned.
<b>displayNameEx</b>	string (63)	The display name of the participant.
<b>guest</b>	boolean	<b>true</b> if the participant is a guest, <b>false</b> if the participant is a chair.
<b>remoteLinkType</b>	string	One of <b>slave</b> , <b>conference</b> , <b>autoAttendant</b> , <b>recording</b> , or <b>playback</b> .
<b>displayNameOverrideStatus</b>	boolean	<b>true</b> if the endpoint uses the <b>displayNameOverrideValue</b> text to identify itself to other participants.
<b>maxBitRateToMCU</b>	integer	Maximum bandwidth to the MCU (kbps).
<b>maxBitRateFromMCU</b>	integer	Maximum bandwidth from the MCU (kbps).
<b>motionSharpnessTradeoff</b>	string	Defines preference for motion vs. sharpness. One of <b>preferMotion</b> , <b>preferSharpness</b> , <b>balanced</b> , or <b>default</b> . <a href="#">more...</a>
<b>callStateEx</b>	string	One of <b>dormant</b> , <b>proceeding</b> , <b>alerting</b> , <b>connected</b> , <b>pending</b> , or <b>disconnected</b> . <a href="#">more...</a>
<b>connectTime</b>	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.
<b>disconnectTime</b>	dateTime. iso8601	Only returned after the participant has disconnected.
<b>disconnectReason</b>	string	Only returned after the participant has disconnected; this contains one of the <a href="#">Disconnect reasons [p.190]</a> .

<b>connectPending</b>	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).
<b>redial</b>	string	Defines the MCU's redial behavior when calls out to this participant drop. One of <b>never</b> , <b>connect</b> , <b>unexpected</b> , <b>any</b> , or <b>default</b> . <a href="#">more...</a>
<b>redialLimit</b>	string	Defines whether a redial limit is used with the redial behavior. One of <b>enabled</b> , <b>disabled</b> , or <b>default</b> . <a href="#">more...</a>
<b>audioRxCodec</b>	string	Receive audio codec.
<b>audioRxLost</b>	integer	Count of the audio packets lost by the MCU.
<b>audioRxReceived</b>	integer	Count of audio packets received by the MCU.
<b>audioRxMuted</b>	boolean	<b>true</b> means that audio from this participant will not be heard by other conference participants.
<b>audioRxMutedRemotely</b>	boolean	Whether this endpoint is muted remotely.
<b>audioRxGainMode</b>	string	<b>none</b> , <b>automatic</b> , <b>default</b> , or <b>fixed</b> . <a href="#">more...</a>
<b>audioRxGainMillidB</b>	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.
<b>audioTxCodec</b>	string	The codec used on the audio transmission.
<b>audioTxReportedLost</b>	integer	The count of audio packets reported lost by the far end.
<b>audioTxSent</b>	integer	Count of the audio packets sent to this endpoint.
<b>audioTxMuted</b>	boolean	<b>true</b> means that the MCU is not transmitting the audio part of the conference to this participant.
<b>audioRxEnergyMillidB</b>	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).
<b>videoRxCodec</b>	string	The codec used on the received video.
<b>videoRxLost</b>	integer	Count of video packets lost en route to the MCU from this endpoint.
<b>videoRxMuted</b>	boolean	<b>true</b> means that video from this participant will not be seen by other conference participants.
<b>videoRxReceived</b>	integer	Count of video packets received from this endpoint.
<b>videoTxCodec</b>	string	The codec used on the transmitted video.
<b>videoTxReportedLost</b>	integer	The count of video packets reported lost by the far end.
<b>videoTxSent</b>	integer	Count of the video packets sent to the endpoint.
<b>videoTxMuted</b>	boolean	<b>true</b> means that the MCU does not send the video part of the conference to this participant.
<b>videoTxWidescreen</b>	boolean	If <b>true</b> , the MCU sends video in a form suitable for a widescreen 16:9 display to this participant.
<b>contentRxType</b>	string	Type of content received. One of <b>none</b> , <b>h239</b> , or <b>bfcp</b> . <a href="#">more...</a>

<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> . <a href="#">more...</a>
<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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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. Refer to <a href="#">Conference layouts [p.193]</a> 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> . <a href="#">more...</a>
<code>callDirection</code>	string	Either <code>incoming</code> or <code>outgoing</code> . <a href="#">more...</a>

<b>previewURL</b>	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.
<b>focusType</b>	string	Indicates the endpoint's focus. One of <code>participant</code> , <code>voiceActivated</code> , or <code>h239</code> . <a href="#">more...</a>

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

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

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

<b>callIdentifier</b>	base64	The base64 encoded GUID (globally unique identifier) of the active H.323 call from this endpoint.
<b>borderWidth</b>	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. <a href="#">more...</a>
<b>suppressAudioDuringDTMF</b>	string	<code>outgoing</code> or <code>all</code> defines which audio the MCU suppresses while it sends the DTMF connection sequence to the endpoint. <a href="#">more...</a>
<b>autoAttendantConfiguredName</b>	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.

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

<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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.

<b>audioRxGainMode</b>	string	<b>none, automatic, default, or fixed.</b> <a href="#">more...</a>
<b>audioRxGainMillidB</b>	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.
<b>videoRxMuted</b>	boolean	<b>true</b> means that video from this participant will not be seen by other conference participants.
<b>videoTxMuted</b>	boolean	<b>true</b> means that the MCU does not send the video part of the conference to this participant.
<b>videoTxWidescreen</b>	boolean	If <b>true</b> , the MCU sends video in a form suitable for a widescreen 16:9 display to this participant.
<b>layoutControlEx</b>	string	Defines how the view layout can be controlled. One of <b>disabled, feccOnly, dtmfOnly, feccWithDtmfFallback, or bothFeccAndDtmf.</b> <a href="#">more...</a>
<b>actAsRecorder</b>	boolean	Defines whether this participant appears as a recorder to other participants.
<b>cpLayout</b>	string	This is the layout for the video sent to the participant.
<b>autoConnect</b>	boolean	<b>true</b> allows endpoints to automatically connect to this conference when they dial in and are recognized. <a href="#">more...</a>
<b>autoDisconnect</b>	boolean	<b>true</b> 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. <b>false</b> means this endpoint requires manual disconnection.  When a participant disconnects from a conference and only participants who have <b>autoDisconnect</b> set to <b>true</b> remain, the MCU disconnects all the remaining participants.
<b>borderWidth</b>	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. <a href="#">more...</a>
<b>linkType</b>	string	This parameter is ignored unless <b>participantType</b> is <b>by_address</b> . Either <b>cascadeSlaveToMaster</b> or <b>default</b>
<b>dtmfSequence</b>	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 href="#">more...</a>
<b>suppressAudioDuringDTMF</b>	string	<b>outgoing</b> or <b>all</b> defines which audio the MCU suppresses while it sends the DTMF connection sequence to the endpoint. <a href="#">more...</a>
<b>suppressDtmfEx</b>	string	Controls the muting of in-band DTMF tones. One of <b>fecc, always, or never.</b> <a href="#">more...</a>
<b>h239Negotiation</b>	string	Defines how the MCU presents itself for h239 token negotiation. One of <b>As master, As slave, or Mimic slave.</b> <a href="#">more...</a>
<b>videoToUse</b>	struct	Collection of parameters that uniquely identify the participant whose video will display in place of this participant's video by default.
<b>participantName</b>	string	The unique name of a participant. <a href="#">more...</a>

<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . <a href="#">more...</a>
<code>participantProtocol</code>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<code>displayName</code>	string	The display name of the participant. If this parameter is longer than 31 characters, only the first 31 characters are returned.

## Deprecated parameters

Parameter name	Type	Short description
<code>callState</code>	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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>

## 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 \[p. 116\]](#).

### 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. <a href="#">more...</a>
<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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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).

<b>maxBitRateFromMCU</b>	integer	Maximum bandwidth from the MCU (kbps).
<b>callState</b>	string	Deprecated by <b>callStateEx</b> . State of the call between the MCU and this participant. One of <b>dormant</b> , <b>alerting</b> , <b>connected</b> , or <b>disconnected</b> . <a href="#">more...</a>
<b>connectTime</b>	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.
<b>disconnectTime</b>	dateTime. iso8601	Only returned after the participant has disconnected.
<b>disconnectReason</b>	string	Only returned after the participant has disconnected; this contains one of the <a href="#">Disconnect reasons [p.190]</a> .
<b>connectPending</b>	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).
<b>initialAudioMuted</b>	boolean	<b>true</b> if the endpoint's audio is initially muted.
<b>initialVideoMuted</b>	boolean	<b>true</b> if the endpoint's video is initially muted.
<b>audioRxCodec</b>	string	Receive audio codec.
<b>audioRxLost</b>	integer	Count of the audio packets lost by the MCU.
<b>audioRxReceived</b>	integer	Count of audio packets received by the MCU.
<b>audioTxCodec</b>	string	The codec used on the audio transmission.
<b>audioTxReportedLost</b>	integer	The count of audio packets reported lost by the far end.
<b>audioTxSent</b>	integer	Count of the audio packets sent to this endpoint.
<b>audioRxMuted</b>	boolean	<b>true</b> means that audio from this participant will not be heard by other conference participants.
<b>audioRxGainMode</b>	string	<b>none</b> , <b>automatic</b> , <b>default</b> , or <b>fixed</b> . <a href="#">more...</a>
<b>audioRxGainMillidB</b>	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.
<b>videoRxCodec</b>	string	The codec used on the received video.
<b>videoRxLost</b>	integer	Count of video packets lost en route to the MCU from this endpoint.
<b>videoRxReceived</b>	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> . <a href="#">more...</a>
<code>cpLayout</code>	string	This is the layout for the video sent to the participant.
<code>currentLayout</code>	integer	The actual layout in use for the video stream being sent by the MCU. Refer to <a href="#">Conference layouts [p.193]</a> for details.
<code>callDirection</code>	string	Either <code>incoming</code> or <code>outgoing</code> . <a href="#">more...</a>

---

## 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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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 (up to a maximum value of 8639999).

# 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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<code>operationScope</code>	string	Either of the strings <code>activeState</code> or <code>configuredState</code> . <a href="#">more...</a>

### 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> . <a href="#">more...</a>
<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 (63)	This value overrides the participant's display name if <code>displayNameOverrideStatus</code> is <code>true</code> .
<code>cpLayout</code>	string	This is the layout for the video sent to the participant.
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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>videoRxMuted</code>	boolean	<code>true</code> means that video from this participant will not be seen by other conference participants.
<code>videoTxMuted</code>	boolean	<code>true</code> means that the MCU does not send the video part of the conference to this participant.
<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>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>dtmfSequence</code>	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 href="#">more...</a>
<code>suppressAudioDuringDTMF</code>	string	<code>outgoing</code> or <code>all</code> defines which audio the MCU suppresses while it sends the DTMF connection sequence to the endpoint. <a href="#">more...</a>
<code>videoToUse</code>	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. <a href="#">more...</a>
------------------------------	--------	---

<code>participantType</code>	string	One of: <code>by_address</code> or <code>ad_hoc</code> . <a href="#">more...</a>
<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. <a href="#">more...</a>
<code>focusType</code>	string	Indicates the endpoint's focus. One of <code>participant</code> , <code>voiceActivated</code> , or <code>h239</code> . <a href="#">more...</a>
<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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<code>suppressDtmfEx</code>	string	Controls the muting of in-band DTMF tones. One of <code>fecc</code> , <code>always</code> , or <code>never</code> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>

## 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> .

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

## Deprecated parameters

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

## 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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<code>newConferenceName</code>	string	The new conference name. <a href="#">more...</a>

## 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. <a href="#">more...</a>
<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> . <a href="#">more...</a>

# 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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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>contentRxSelectedBitRate</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> .

<b>frameErrors</b>	integer	Count of frames with errors in this stream.
<b>framesTransferred</b>	integer	Count of audio, video, or content frames received, depending on where the parameter occurs.
<b>temporalSpatial</b>	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.
<b>contentType</b>	string	The type of content being sent or received.
<b>contentError</b>	string	Information about problems with outgoing content. One of: <b>notAllowed</b> , <b>noCommonCodecs</b> , <b>noCommonFormats</b> , <b>noCommonSymmetricCodecs</b> , <b>modeMismatch</b> , <b>bitRateMismatch</b> , <b>encryptionNotPossible</b> , <b>notPossible</b> .
<b>lipSyncDelayApplied</b>	integer	The amount of delay added to either audio or video output stream to correct for <b>rtcpLipSyncDelay</b> reported between incoming audio and video streams.
<b>rtcpLipSyncDelay</b>	integer	The reported delay between the incoming audio and video streams from this endpoint.
<b>Interlaced</b>	boolean	Defines whether or not the video in this sent or received stream is interlaced. Deprecates <b>videoTxInterlaced</b> and <b>videoRxInterlaced</b> .
<b>fecRecovered</b>	integer	Only returned if FEC (forward error correction) is negotiated and enabled. (Only on receive media.) This is the number of lost packets recovered by ClearPath FEC.
<b>fecOverhead</b>	integer	Only returned if FEC (forward error correction) is negotiated and enabled. (Sent and received media.) This is the overhead of sending ClearPath FEC packets, expressed as a percentage in addition to the media stream.
<b>repairFrames</b>	integer	The number of ClearPath repair frames sent/received for video. (Sent and received media).

## Control statistics

### Commonly applicable statistics

Parameter name	Type	Short description
<b>rtcpReceiveAddress</b>	string	Address of the RTCP receiver.
<b>rtcpReceivePort</b>	integer	Port number used by the receiver to accept RTCP messages.
<b>rtcpTransmitAddress</b>	string	The IP address and port to which the MCU is sending RTCP packets about this stream.
<b>rtcpTransmitPort</b>	integer	Port number used for transmitting RTCP messages to the endpoint. Absent if <b>rtcpTransmitAddress</b> is unspecified.
<b>rtcpReceiverReports</b>	integer	Count of the RTCP receiver reports seen by the MCU.

<code>rtcpPacketLossReported</code>	integer	The count of media packets reported lost, by the far end, in a receiver report sent to the MCU.
<code>rtcpFecRecoveredReported</code>	integer	The number of lost packets reported as recovered by ClearPath FEC at the far end for video and audio control structs.
<code>rtcpSenderReports</code>	integer	Count of the RTCP sender reports seen by the MCU.
<code>rtcpOtherReports</code>	integer	Count of the RTCP reports seen by the MCU that are neither sender nor receiver reports.
<code>rtcpPacketsSent</code>	integer	Count of RTCP packets sent by the MCU to this endpoint.

### Video- and content-specific control statistics

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

## Deprecated parameters

<code>participant.diagnostics</code> parameters	Deprecated by this parameter	Found in these structs
<code>videoRxCCodec</code> , <code>contentRxCCodec</code> , <code>videoTxCodec</code> , <code>contentTxCodec</code>	codec	All media stats, either direction
	address	All media stats, either direction
	port	All media stats, either direction
<code>audioRxReceived</code> , <code>videoRxReceived</code> , <code>contentRxReceived</code> , <code>videoTxSent</code> , <code>contentTxSent</code>	<code>packetsTransferred</code>	All media stats, either direction
	encryption	All media stats, either direction
<code>videoRxWidth</code> , <code>contentRxWidth</code> , <code>videoTxWidth</code> , <code>contentTxWidth</code>	width	Video and content stats, either direction
<code>videoRxHeight</code> , <code>contentRxHeight</code> , <code>videoTxHeight</code> , <code>contentTxHeight</code>	height	Video and content stats, either direction
<code>videoRxChannelBitRate</code> , <code>contentRxChannelBitRate</code> , <code>videoTxChannelBitRate</code> , <code>contentTxChannelBitRate</code>	<code>channelBitRate</code>	Video and content stats, either direction

<b>participant.diagnostics parameters</b>	<b>Deprecated by this parameter</b>	<b>Found in these structs</b>
videoRxSelectedBitRate, contentRSelectedBitRate, videoTxSelectedBitRate, contentTxSelectedBitRate	selectedBitRate	Video and content stats, either direction
videoRxActualBitRate, contentRxActualBitRate, videoTxActualBitRate, contentTxActualBitRate	actualBitRate	Video and content stats, either direction
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, either direction
	repairFrames	Video stats, either direction
	rtcpReceiveAddress	All control structs
	rtcpReceivePort	All control structs
	rtcpTransmitAddress	All control structs
	rtcpTransmitPort	All control structs

<b>participant.diagnostics parameters</b>	<b>Deprecated by this parameter</b>	<b>Found in these structs</b>
	rtcpReceiverReports	All control structs
audioTxReportedLost, videoTxReportedLost, contentTxReportedLost	rtcpPacketLossReported	All control structs
	rtcpFecRecoveredReported	All control structs
	rtcpSenderReports	All control structs
	rtcpOtherReports	All control structs
	rtcpPacketsSent	All control structs
	fursSent	Video and content control 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 \[p.116\]](#), 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 \(deprecated\) \[p.151\]](#) 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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>

## 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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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>currentState</code>	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.

<code>configuredState</code>	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
<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>ipAddress</code>	string	IPv4 address in dotted-quad format.
<code>displayName</code>	string	The display name of the participant. If this parameter is longer than 31 characters, only the first 31 characters are returned.
<code>displayNameEx</code>	string (63)	The display name of the participant.
<code>guest</code>	boolean	<code>true</code> if the participant is a guest, <code>false</code> if the participant is a chair.
<code>remoteLinkType</code>	string	One of <code>slave</code> , <code>conference</code> , <code>autoAttendant</code> , <code>recording</code> , or <code>playback</code> .
<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> . <a href="#">more...</a>
<code>callStateEx</code>	string	One of <code>dormant</code> , <code>proceeding</code> , <code>alerting</code> , <code>connected</code> , <code>pending</code> , or <code>disconnected</code> . <a href="#">more...</a>
<code>connectTime</code>	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.

<b>disconnectTime</b>	dateTime. iso8601	Only returned after the participant has disconnected.
<b>disconnectReason</b>	string	Only returned after the participant has disconnected; this contains one of the <a href="#">Disconnect reasons [p.190]</a> .
<b>connectPending</b>	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).
<b>redial</b>	string	Defines the MCU's redial behavior when calls out to this participant drop. One of <b>never</b> , <b>connect</b> , <b>unexpected</b> , <b>any</b> , or <b>default</b> . <a href="#">more...</a>
<b>redialLimit</b>	string	Defines whether a redial limit is used with the redial behavior. One of <b>enabled</b> , <b>disabled</b> , or <b>default</b> . <a href="#">more...</a>
<b>audioRxCodec</b>	string	Receive audio codec.
<b>audioRxLost</b>	integer	Count of the audio packets lost by the MCU.
<b>audioRxReceived</b>	integer	Count of audio packets received by the MCU.
<b>audioTxCodec</b>	string	The codec used on the audio transmission.
<b>audioTxReportedLost</b>	integer	The count of audio packets reported lost by the far end.
<b>audioTxSent</b>	integer	Count of the audio packets sent to this endpoint.
<b>audioRxMuted</b>	boolean	<b>true</b> means that audio from this participant will not be heard by other conference participants.
<b>audioTxMuted</b>	boolean	<b>true</b> means that the MCU is not transmitting the audio part of the conference to this participant.
<b>audioRxGainMode</b>	string	<b>none</b> , <b>automatic</b> , <b>default</b> , or <b>fixed</b> . <a href="#">more...</a>
<b>audioRxGainMillidB</b>	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.
<b>videoRxCodec</b>	string	The codec used on the received video.
<b>videoRxLost</b>	integer	Count of video packets lost en route to the MCU from this endpoint.
<b>videoRxReceived</b>	integer	Count of video packets received from this endpoint.
<b>videoTxCodec</b>	string	The codec used on the transmitted video.
<b>videoTxReportedLost</b>	integer	The count of video packets reported lost by the far end.
<b>videoTxSent</b>	integer	Count of the video packets sent to the endpoint.
<b>videoRxMuted</b>	boolean	<b>true</b> means that video from this participant will not be seen by other conference participants.
<b>videoTxMuted</b>	boolean	<b>true</b> means that the MCU does not send the video part of the conference to this participant.
<b>videoTxWidescreen</b>	boolean	If <b>true</b> , the MCU sends video in a form suitable for a widescreen 16:9 display to this participant.
<b>contentRxType</b>	string	Type of content received. One of <b>none</b> , <b>h239</b> , or <b>bfcp</b> . <a href="#">more...</a>

<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> . <a href="#">more...</a>
<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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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. Refer to <a href="#">Conference layouts [p.193]</a> 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> . <a href="#">more...</a>
<code>callDirection</code>	string	Either <code>incoming</code> or <code>outgoing</code> . <a href="#">more...</a>

<b>previewURL</b>	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.
<b>focusType</b>	string	Indicates the endpoint's focus. One of <code>participant</code> , <code>voiceActivated</code> , or <code>h239</code> . <a href="#">more...</a>

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

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

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

<b>callIdentifier</b>	base64	The base64 encoded GUID (globally unique identifier) of the active H.323 call from this endpoint.
<b>borderWidth</b>	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. <a href="#">more...</a>
<b>autoAttendantConfiguredName</b>	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.

<b>mediaEncryption</b>	string	One of <code>encrypted</code> , <code>unencrypted</code> , <code>mixed</code> , or <code>unknown</code> . <a href="#">more...</a>
<b>audioRxEnergyMillidB</b>	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).
<b>audioRxMutedRemotely</b>	boolean	Whether this endpoint is muted remotely.
<b>suppressAudioDuringDTMF</b>	string	<code>outgoing</code> or <code>all</code> defines which audio the MCU suppresses while it sends the DTMF connection sequence to the endpoint. <a href="#">more...</a>
<b>packetLossWarning</b>	boolean	This will be true if any packet loss has been seen within the last 15 seconds.
<b>packetLossCritical</b>	boolean	This will be true if any packet loss above a certain level (5%) is seen within the last five seconds.

<b>videoToUse</b>	struct	Collection of parameters that uniquely identify the participant whose video will display in place of this participant's video by default.
<b>participantName</b>	string	The unique name of a participant. <a href="#">more...</a>
<b>participantType</b>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . <a href="#">more...</a>
<b>participantProtocol</b>	string	<code>h323</code> , <code>sip</code> , or <code>vnc</code> .
<b>conferenceName</b>	string	The name of the conference.
<b>connectionUniqueId</b>	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
<b>address</b> ( <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.

<b>gatewayAddress</b>	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> .
<b>useSIPRegistrar</b>	boolean	Not valid unless the protocol is SIP. <code>true</code> if the endpoint uses the SIP registrar. Defaults to <code>false</code> .
<b>transportProtocol</b>	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> .
<b>password</b>	string	The password for VNC endpoints.
<b>deferConnection</b>	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> .
<b>redial</b>	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> . <a href="#">more...</a>
<b>redialLimit</b>	string	Defines whether a redial limit is used with the redial behavior. One of <code>enabled</code> , <code>disabled</code> , or <code>default</code> . <a href="#">more...</a>
<b>displayNameOverrideStatus</b>	boolean	<code>true</code> if the endpoint uses the <code>displayNameOverrideValue</code> text to identify itself to other participants.
<b>maxBitRateToMCU</b>	integer	Maximum bandwidth to the MCU (kbps).
<b>maxBitRateFromMCU</b>	integer	Maximum bandwidth from the MCU (kbps).

<b>motionSharpnessTradeoff</b>	string	Defines preference for motion vs. sharpness. One of <b>preferMotion</b> , <b>preferSharpness</b> , <b>balanced</b> , or <b>default</b> . <a href="#">more...</a>
<b>audioRxMuted</b>	boolean	<b>true</b> means that audio from this participant will not be heard by other conference participants.
<b>audioTxMuted</b>	boolean	<b>true</b> means that the MCU is not transmitting the audio part of the conference to this participant.
<b>audioRxGainMode</b>	string	<b>none</b> , <b>automatic</b> , <b>default</b> , or <b>fixed</b> . <a href="#">more...</a>
<b>audioRxGainMillidB</b>	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.
<b>videoRxMuted</b>	boolean	<b>true</b> means that video from this participant will not be seen by other conference participants.
<b>videoTxMuted</b>	boolean	<b>true</b> means that the MCU does not send the video part of the conference to this participant.
<b>videoTxWidescreen</b>	boolean	If <b>true</b> , the MCU sends video in a form suitable for a widescreen 16:9 display to this participant.
<b>layoutControlEx</b>	string	Defines how the view layout can be controlled. One of <b>disabled</b> , <b>feccOnly</b> , <b>dtmfOnly</b> , <b>feccWithDtmfFallback</b> , or <b>bothFeccAndDtmf</b> . <a href="#">more...</a>
<b>actAsRecorder</b>	boolean	Defines whether this participant appears as a recorder to other participants.
<b>cpLayout</b>	string	This is the layout for the video sent to the participant.
<b>autoConnect</b>	boolean	<b>true</b> allows endpoints to automatically connect to this conference when they dial in and are recognized. <a href="#">more...</a>
<b>autoDisconnect</b>	boolean	<b>true</b> 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. <b>false</b> means this endpoint requires manual disconnection.  When a participant disconnects from a conference and only participants who have <b>autoDisconnect</b> set to <b>true</b> remain, the MCU disconnects all the remaining participants.
<b>borderWidth</b>	integer	Controls the width of the outer border of a preconfigured participant's layout. 0 is no border. <a href="#">more...</a>
<b>linkType</b>	string	This parameter is ignored unless <b>participantType</b> is <b>by_address</b> . Either <b>cascadeSlaveToMaster</b> or <b>default</b>
<b>dtmfSequence</b>	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 href="#">more...</a>
<b>suppressAudioDuringDTMF</b>	string	<b>outgoing</b> or <b>all</b> defines which audio the MCU suppresses while it sends the DTMF connection sequence to the endpoint. <a href="#">more...</a>
<b>suppressDtmfEx</b>	string	Controls the muting of in-band DTMF tones. One of <b>fecc</b> , <b>always</b> , or <b>never</b> . <a href="#">more...</a>

<b>h239Negotiation</b>	string	Defines how the MCU presents itself for h239 token negotiation. One of <b>As master</b> , <b>As slave</b> , or <b>Mimic slave</b> . <a href="#">more...</a>
<b>videoToUse</b>	struct	Collection of parameters that uniquely identify the participant whose video will display in place of this participant's video by default.
<b>participantName</b>	string	The unique name of a participant. <a href="#">more...</a>
<b>participantType</b>	string	One of: <b>by_address</b> , <b>by_name</b> , or <b>ad_hoc</b> . <a href="#">more...</a>
<b>participantProtocol</b>	string	<b>h323</b> , <b>sip</b> , or <b>vnc</b> .
<b>displayName</b>	string	The display name of the participant.

If this parameter is longer than 31 characters, only the first 31 characters are returned.

## Deprecated parameters

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

## 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 \[p.143\]](#).

Returns information about an individual participant. This call returns a participant struct as described in [participant.enumerate \(deprecated\) \[p.125\]](#), 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. <a href="#">more...</a>
<code>participantType</code>	string	One of: <code>by_address</code> , <code>by_name</code> , or <code>ad_hoc</code> . <a href="#">more...</a>
<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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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.

<b>autoAttendantUniqueID</b>	string	Unique identifier for the auto attendant.
<b>connectionUniqueId</b>	integer	Corresponds to the <code>uniqueId</code> returned by a conference or autoattendant.
<b>address (endpoint)</b>	string (63)	The address of the endpoint; may be hostname, IP address, E.164 number, SIP URI, or H.323 ID.
<b>gatewayAddress</b>	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> .
<b>deferConnection</b>	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> .
<b>displayName</b>	string	The display name of the participant.  If <code>displayName</code> is longer than 31 characters, only the first 31 characters are returned.
<b>displayNameOverrideStatus</b>	boolean	<code>true</code> if the endpoint uses the <code>displayNameOverrideValue</code> text to identify itself to other participants.
<b>maxBitRateToMCU</b>	integer	Maximum bandwidth to the MCU (kbps).
<b>maxBitRateFromMCU</b>	integer	Maximum bandwidth from the MCU (kbps).
<b>callState</b>	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> . <a href="#">more...</a>
<b>connectTime</b>	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.
<b>disconnectTime</b>	dateTime. iso8601	Only returned after the participant has disconnected.
<b>disconnectReason</b>	string	Only returned after the participant has disconnected; this contains one of the <a href="#">Disconnect reasons [p.190]</a> .
<b>connectPending</b>	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).
<b>audioRxCCodec</b>	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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<code>cpLayout</code>	string	This is the layout for the video sent to the participant.

---

<b>currentLayout</b>	integer	The actual layout in use for the video stream being sent by the MCU. Refer to <a href="#">Conference layouts [p.193]</a> for details.
<b>callDirection</b>	string	Either <b>incoming</b> or <b>outgoing</b> . <a href="#">more...</a>

---

# 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
<b>destination</b>	string	IP address of the route's destination.
<b>prefixLength</b>	integer	The prefix length of the destination IP range for this route (the number of fixed bits in the address).
<b>gateway</b>	string	One of <b>A</b> or <b>B</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 <b>destination</b> ).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
<b>status</b> ( <i>success</i> )	string	<b>Operation successful</b>
<b>newRouteId</b>	integer	A number selected by the device to identify the newly added route. Pass this parameter as <b>routeId</b> 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
<b>filter</b> ( <i>route</i> )	string	Filters the returned routes by the route type. One of <b>configured</b> , <b>automatic</b> , or <b>both</b> . Defaults to <b>both</b> .

## Returned data

### Always returned

Parameter name	Type	Short description
<b>ipv4Routes</b>	array	An array of structs, each of which represents an IPv4 route.
<b>destination</b>	string	IP address of the route's destination.
<b>prefixLength</b>	integer	The prefix length of the destination IP range for this route (the number of fixed bits in the address).
<b>gateway</b>	string	The IP address of the gateway (or next hop) of this route.
<b>port</b> ( <i>Ethernet</i> )	string	Identifies the Ethernet port. May be A or B.
<b>type</b> ( <i>route</i> )	string	The type of route. One of <b>automatic</b> , <b>configuredByGateway</b> or <b>configuredByPort</b> .
<b>active</b> ( <i>route</i> )	boolean	<b>true</b> if the route is currently active. <b>false</b> if the route is inactive (e.g. a route pointing to Port B when port B is disabled). Applies to configured routes only.
<b>routeId</b>	integer	A number that identifies a route. The device assigns a number to each manually configured route.
<b>ipv6Routes</b>	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 <b>A</b> or <b>B</b> , indicates which Ethernet port is preferred for traffic bound for IPv4 destinations.
<code>ipv6Preference</code>	string	Either <b>A</b> or <b>B</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 <b>A</b> or <b>B</b> , indicates which Ethernet port is preferred for traffic bound for IPv4 destinations.
<code>ipv6Preference</code>	string	Either <b>A</b> or <b>B</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>  <a href="#">more...</a>
<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	<b>IPv4</b> or <b>IPv6</b> .
<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	<b>true</b> if this feature or item is enabled.
<code>type (service)</code>	string	The type of service. Either <b>tcp</b> or <b>udp</b> .
<code>name (service)</code>	string	The name of the service. One of the following:  TCP services: <b>http</b> , <b>https</b> , <b>ftp</b> , <b>h225</b> , <b>rtsp</b> , <b>mms</b> , <b>sip_tcp</b> , <b>sips_tcp</b> , <b>cdep</b>  UDP services: <b>sip_udp</b> , <b>snmp</b> , <b>gatekeeper</b> , <b>tunnel</b>  <a href="#">more...</a>
<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.

---

**Note:** Deprecated fields are ignored if any of the new fields have been provided.

---

## Input parameters

### Required inputs

If you set `outboundConfiguration` to `registrar`, then `outboundAddress` and `registrarContactURI` are mandatory.

If you set `outboundConfiguration` to `trunk`, then `outboundAddress` is mandatory.

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

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

### Optional or conditional inputs

Parameter name	Type	Short description
<code>outboundConfiguration</code>	String (registrar, trunk, direct)	Defines how outbound SIP calls will be made. Replaces the functionality of <code>registrarUsage</code> .
<code>outboundAddress</code>	String	The address all outbound SIP calls should be sent to when in registrar or trunk mode. Replaces <code>configuredProxy</code>
<code>outboundDomain</code>	String	The domain that should be used during SIP invites and registration. Replaces <code>configuredRegistrar</code> .
<code>registrarUsage</code>	boolean (deprecated)	Defines if the MCU should register to a registrar. Sets (deprecated) <code>outboundConfiguration</code> .
<code>configuredRegistrar</code>	string (255) (deprecated)	The <code>outboundDomain</code> that should be used for SIP invites (deprecated) and registrations, also sets the <code>outboundAddress</code> if no proxy/address is provided.
<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 <b>Username</b> setting on the <a href="#">Settings &gt; SIP</a> 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 <b>Allow numeric ID registration for conferences</b> on the <a href="#">Settings &gt; SIP</a> page of the web interface.

---

<b>configuredProxy</b>	string (255)	The <b>outboundAddress</b> that SIP invites and registrations (deprecated) should be sent to.
<b>maxOcsBitrate</b>	integer	The bitrate to use for ocs and lcs clients, in bits per second. Accepts <b>64000, 128000, 192000, 256000, 320000, 384000, 512000, 768000, 1000000, 1250000, 1500000, 1750000, 2000000, 2500000, 3000000, 3500000, or 4000000</b> . Set this to 0 to disable the limit.
<b>outgoingTransport</b>	string	The outgoing transport protocol. One of <b>udp, tcp, or tls</b> .
<b>useLocalCertificate</b>	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>outboundConfiguration</code>	String (registrar, trunk, direct)	Defines how outbound SIP calls will be made. Replaces the functionality of <code>registrarUsage</code> .
<code>outboundAddress</code>	String	The address all outbound SIP calls should be sent to when in registrar or trunk mode. Replaces <code>configuredProxy</code>
<code>outboundDomain</code>	String	The domain that should be used during SIP invites and registration. Replaces <code>configuredRegistrar</code> .
<code>configuredRegistrar</code>	string (255) (deprecated)	The <code>outboundDomain</code> that should be used for SIP invites and registrations, also sets the <code>outboundAddress</code> if no proxy/address is provided.
<code>configuredProxy</code>	string (255) (deprecated)	The <code>outboundAddress</code> that SIP invites and registrations should be sent to.
<code>registrarContactURI</code>	string (255)	The URI provided to the SIP registrar to register this device. Corresponds to the <b>Username</b> setting on the <a href="#">Settings &gt; SIP</a> web page.
<code>registrarContactDomain</code>	string	This value is generated from the <code>registrarContactURI</code> ( <b>Username</b> in the web interface) and the <code>configuredRegistrar</code> ( <b>SIP Registrar domain</b> 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 <b>enabled</b> or <b>disabled</b> . Corresponds to <b>Allow numeric ID registration for conferences</b> on the <a href="#">Settings &gt; SIP</a> page of the web interface.
<code>registrarUsage</code>	boolean (deprecated)	Defines if the MCU should register to a registrar. Sets <code>outboundConfiguration</code> . <b>true</b> if it is a registrar, otherwise <b>false</b> .
<code>registrarType</code>	string (10)	The type of SIP registrar. Either <b>normal</b> or <b>lcs</b> .
<code>maxOcsBitrate</code>	integer	The bitrate to use for ocs and lcs clients, in bits per second. Accepts <b>64000, 128000, 192000, 256000, 320000, 384000, 512000, 768000, 1000000, 1250000, 1500000, 1750000, 2000000, 2500000, 3000000, 3500000, or 4000000</b> . Set this to 0 to disable the limit.
<code>outgoingTransport</code>	string	The outgoing transport protocol. One of <b>udp, tcp, or tls</b> .
<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 <b>registering, registered, unregistered, or unknown</b> .

# 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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> .

<b>contentTransmitResolutions</b> ( <i>template</i> )	string	The resolution for the content channel that will be transmitted to endpoints in conferences based on this template. One of <b>4to3Only</b> , <b>16to9Only</b> , <b>allowAll</b> , or <b>default</b> . <a href="#">more...</a>
<b>contentTxCodec</b> ( <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; <b>h263+</b> , <b>h264</b> , <b>automatic</b> , or <b>default</b> . This setting does not apply in <b>passthrough</b> mode. <a href="#">more...</a>
<b>contentTxMinimumBitRate</b> ( <i>template</i> )	string	The minimum bit rate to use for transmitting content, in bps, in conferences based on this template. One of: <b>0</b> , <b>64000</b> , <b>128000</b> , <b>192000</b> , <b>256000</b> , <b>320000</b> , <b>384000</b> , <b>512000</b> , <b>768000</b> , <b>1000000</b> , <b>1250000</b> , <b>1500000</b> , or <b>default</b> (inherit this setting from the parent template).
<b>contentPassthroughLimit</b> ( <i>template</i> )	string	One of <b>none</b> , <b>sd</b> , <b>hd</b> , <b>fullhd</b> , or <b>default</b> (inherit this setting from the parent template). This is the minimum resolution passed on to endpoints. If an endpoint that can only receive content below the minimum joins a conference, the endpoint will not be sent a passthrough content stream or a transcoded content stream. If the MCU receives content with a resolution below this setting, then it will use the received resolution as the minimum instead.
<b>joinAudioMuted</b> ( <i>template</i> )	string	Mutes audio on join. One of <b>true</b> , <b>false</b> , or <b>default</b> to inherit this setting from the parent template.
<b>joinVideoMuted</b> ( <i>template</i> )	string	Mutes video on join. One of <b>true</b> , <b>false</b> , or <b>default</b> to inherit this setting from the parent template.
<b>joinAGC</b> ( <i>template</i> )	string	Whether AGC should be used by default for participants joining this conference. One of <b>true</b> , <b>false</b> , or <b>default</b> if this template inherits the <b>joinAGC</b> setting of its parent template.
<b>layoutControlEx</b> ( <i>template</i> )	string	Defines how the view layout can be controlled. One of <b>disabled</b> , <b>feccOnly</b> , <b>dtmfOnly</b> , <b>feccWithDtmfFallback</b> , or <b>bothFeccAndDtmf</b> , or <b>default</b> . <a href="#">more...</a>
<b>cameraControl</b> ( <i>template</i> )	string	Defines how the endpoint camera(s) in conferences based on this template can be controlled. If present, it may be <b>disabled</b> , <b>feccOnly</b> , <b>dtmfOnly</b> , <b>feccWithDtmfFallback</b> , <b>bothFeccAndDtmf</b> , or <b>default</b> . <a href="#">more...</a>
<b>dtmfMuteControl</b> ( <i>template</i> )	string	Deprecated by <b>inCallMenuControlChair</b> ( <i>template</i> ) and <b>inCallMenuControlGuest</b> ( <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 <b>true</b> , <b>false</b> , or <b>default</b> (inherit this setting from the parent template).
<b>encryptionRequired</b> ( <i>template</i> )	string	The encryption setting for conferences based on this template, if the encryption feature key is enabled. If <b>true</b> , encryption is required for these conferences. Otherwise, encryption is optional. <b>default</b> causes the template to inherit this setting from its parent template.

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

## 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. <a href="#">more...</a>
<code>status (success)</code>	string	<code>Operation successful</code>

### Deprecated parameters

Parameter name	Type	Short description
<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).

# 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 \[p.188\]](#)).

## 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. <a href="#">more...</a>

## 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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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. <a href="#">more...</a>
<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>contentPassthroughLimit (template)</code>	string	One of <code>none</code> , <code>sd</code> , <code>hd</code> , <code>fullhd</code> , or <code>default</code> (inherit this setting from the parent template). This is the minimum resolution passed on to endpoints. If an endpoint that can only receive content below the minimum joins a conference, the endpoint will not be sent a passthrough content stream or a transcoded content stream. If the MCU receives content with a resolution below this setting, then it will use the received resolution as the minimum instead.
<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. One of <code>true</code> , <code>false</code> , or <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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>

<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).
<code>encryptionRequired</code> ( <i>template</i> )	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> . <a href="#">more...</a>
<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> ). <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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.

<b>enforceMaximumAudioPorts</b> ( <i>template</i> )	boolean	Defines whether conferences based on this template will enforce the <b>maximumAudioPorts</b> limit. One of <b>true</b> , <b>false</b> , or <b>default</b> . <a href="#">more...</a>
<b>useReservedPortsFromParent</b>	boolean	Cannot be set to true for template 0
<b>reserveVideoPorts</b>	boolean	Determines whether the template should have a value for the reserved video ports setting. Has no effect if the request sets <b>usePortsFromParent</b> to <b>true</b> .
<b>reservedVideoPorts</b>	integer	The number of video ports to reserve for a conference if in port reservation mode.
<b>reserveAudioPorts</b>	boolean	Determines if the template should have a value for the reserved audio ports setting. Has no effect if the request sets <b>usePortsFromParent</b> to <b>true</b> .
<b>reservedAudioPorts</b>	integer	The number of audio only ports to reserve for a conference if in port reservation mode. <a href="#">more...</a>

## Deprecated parameters

Parameter name	Type	Short description
<b>dtmfMuteControl</b> ( <i>template</i> )	string	Deprecated by <b>inCallMenuControlChair</b> ( <i>template</i> ) and <b>inCallMenuControlGuest</b> ( <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 <b>true</b> , <b>false</b> , or <b>default</b> (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. <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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. <a href="#">more...</a>
<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>contentPassthroughLimit (template)</code>	string	One of <code>none</code> , <code>sd</code> , <code>hd</code> , <code>fullhd</code> , or <code>default</code> (inherit this setting from the parent template). This is the minimum resolution passed on to endpoints. If an endpoint that can only receive content below the minimum joins a conference, the endpoint will not be sent a passthrough content stream or a transcoded content stream. If the MCU receives content with a resolution below this setting, then it will use the received resolution as the minimum instead.
<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. One of <code>true</code> , <code>false</code> , or <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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>
<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).

<b>encryptionRequired</b> ( <i>template</i> )	string	The encryption setting for conferences based on this template, if the encryption feature key is enabled. If <b>true</b> , encryption is required for these conferences. Otherwise, encryption is optional. <b>default</b> causes the template to inherit this setting from its parent template.
<b>suppressDtmfEx</b> ( <i>template</i> )	string	Controls the muting of in-band DTMF tones for conferences based on this template. One of <b>fecc</b> , <b>always</b> , <b>never</b> , or <b>default</b> . <a href="#">more...</a>
<b>automaticLectureModeEnabled</b> ( <i>template</i> )	string	Defines whether automatic lecture mode is enabled for conferences based on this template. Deprecated by <b>automaticLectureMode</b> ( <i>template</i> ). <a href="#">more...</a>
<b>automaticLectureMode</b> ( <i>template</i> )	string	Defines automatic lecture mode. One of <b>type1</b> , <b>type2</b> , <b>disabled</b> , or <b>default</b> . <a href="#">more...</a>
<b>automaticLectureModeTimeout</b>	integer	If <b>automaticLectureMode</b> is <b>type1</b> , this integer defines the period of time for which a speaker must be talking before lecture mode begins. <a href="#">more...</a>
<b>chairControl</b> ( <i>template</i> )	string	The chair control setting for conferences based on this template. One of <b>none</b> , <b>floorControlOnly</b> , <b>chairAndFloorControl</b> , or <b>default</b> . <a href="#">more...</a>
<b>lastChairmanLeavesDisconnect</b> ( <i>template</i> )	string	Defines whether conferences based on this template disconnect guests when the last chairperson leaves. One of <b>true</b> , <b>false</b> , or <b>default</b> . <a href="#">more...</a>
<b>preconfiguredParticipantsDefer</b> ( <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 <b>true</b> , <b>false</b> , or <b>default</b> . <a href="#">more...</a>
<b>useMaximumPortsFromParent</b>	boolean	Cannot be set to true for template 0
<b>enforceMaximumVideoPorts</b> ( <i>template</i> )	string	Defines whether conferences based on this template will enforce the <b>maximumVideoPorts</b> limit. One of <b>true</b> , <b>false</b> , or <b>default</b> . <a href="#">more...</a>
<b>maximumVideoPorts</b>	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.
<b>maximumAudioPorts</b>	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.
<b>enforceMaximumAudioPorts</b> ( <i>template</i> )	boolean	Defines whether conferences based on this template will enforce the <b>maximumAudioPorts</b> limit. One of <b>true</b> , <b>false</b> , or <b>default</b> . <a href="#">more...</a>
<b>useReservedPortsFromParent</b>	boolean	Cannot be set to true for template 0
<b>reserveVideoPorts</b>	boolean	Determines whether the template should have a value for the reserved video ports setting. Has no effect if the request sets <b>usePortsFromParent</b> to <b>true</b> .
<b>reservedVideoPorts</b>	integer	The number of video ports to reserve for a conference if in port reservation mode.
<b>reserveAudioPorts</b>	boolean	Determines if the template should have a value for the reserved audio ports setting. Has no effect if the request sets <b>usePortsFromParent</b> to <b>true</b> .

<b>reservedAudioPorts</b>	integer	The number of audio only ports to reserve for a conference if in port reservation mode. <a href="#">more...</a>
---------------------------	---------	---

## Returned data

### Always returned

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

## Deprecated parameters

Parameter name	Type	Short description
<b>dtmfMuteControl</b> ( <i>template</i> )	string	Deprecated by <b>inCallMenuControlChair</b> ( <i>template</i> ) and <b>inCallMenuControlGuest</b> ( <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 <b>true</b> , <b>false</b> , or <b>default</b> (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. <a href="#">more...</a>

## 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> . <a href="#">more...</a>
<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> . <a href="#">more...</a>

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

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

## 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 .....	186
Fault codes .....	188
Disconnect reasons .....	190
HTTP keep-alives .....	192
Conference layouts .....	193
Linking conferences across MCUs .....	195

# 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 <b>Cisco</b> on newer hardware or <b>Codian</b> .
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 <b>clusterType</b> is not <b>slave</b> 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 <b>totalVideoPorts</b> is 0. Contains a subnode for each type of video port allocated. These nodes contain the number of ports of that type. e.g. <b>&lt;hd&gt;10&lt;/hd&gt;</b> , <b>&lt;nhd&gt;40&lt;/nhd&gt;</b> , <b>&lt;sd&gt;40&lt;/sd&gt;</b> , or <b>&lt;hdPlus&gt;10&lt;/hdPlus&gt;</b> . The subnodes are only included if the count of allocated ports of that type is greater than 0.
portReservationMode	<b>enabled</b> or <b>disabled</b> determines whether the MCU allows conferences to reserve media ports. Only included if <b>clusterType</b> is not <b>slave</b> .

Node name	Node contents
maxVideoResolution	<b>max</b> if the hardware is HD capable or has the 4cif key installed, or <b>cif</b> otherwise. Only included if <b>clusterType</b> is not <b>slave</b> .
uptimeSeconds	Time since the MCU booted.
clusterType	The role of this system in a backplane cluster. May be <b>unclustered</b> , <b>master</b> , or <b>slave</b> . This entry is not included in <b>system.xml</b> 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	<b>method not supported.</b> This method is not supported on this device.
2	<b>duplicate conference name.</b> A conference name was specified, but is already in use.
3	<b>duplicate participant name.</b> A participant name was specified, but is already in use.
4	<b>no such conference or auto attendant.</b> The conference or auto attendant identification given does not match any conference or auto attendant.
5	<b>no such participant.</b> The participant identification given does not match any participants.
6	<b>too many conferences.</b> The device has reached the limit of the number of conferences that can be configured.
7	<b>too many participants.</b> There are already too many participants configured and no more can be created.
8	<b>no conference name or auto attendant id supplied.</b> A conference name or auto attendant identifier was required, but was not present.
9	<b>no participant name supplied.</b> A participant name is required but was not present.
10	<b>no participant address supplied.</b> A participant address is required but was not present.
11	<b>invalid start time specified.</b> A conference start time is not valid.
12	<b>invalid end time specified.</b> A conference end time is not valid.
13	<b>invalid PIN specified.</b> A PIN specified is not a valid series of digits.
14	<b>authorization failed.</b> The requested operation is not permitted on this device.
15	<b>insufficient privileges.</b> The specified user id and password combination is not valid for the attempted operation.
16	<b>invalid enumerateID value.</b> An enumerate ID passed to an enumerate method invocation was invalid. Only values returned by the device should be used in enumerate methods.
17	<b>port reservation failure.</b> This is in the case that reservedAudioPorts or reservedVideoPorts value is set too high, and the device cannot support this.
18	<b>duplicate numeric ID.</b> A numeric ID was given, but this ID is already in use.
19	<b>unsupported protocol.</b> 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	<b>unsupported participant type.</b> A participant type was used which does not correspond to any participant type known to the device.
25	<b>new port limit lower than currently active</b>
26	<b>floor control not enabled for this conference</b>
27	<b>no such template.</b> The specified template wasn't found.

---

30	<b>unsupported bit rate.</b> A call tried to set a bit rate that the device does not support.
31	<b>template name in use.</b> This occurs when trying to create or rename a template to have the same name as an existing template.
32	<b>too many templates.</b> This occurs when trying to create a new template after the limit of 100 templates has been reached.
36	<b>required value missing.</b> The call has omitted a value that the MCU requires to make the change requested by the call.
42	<b>port conflict.</b> The call attempts to set a port number that is already in use by another service.
43	<b>route already exists.</b> The call attempts to add a route that has the same <b>destination</b> and <b>prefixLength</b> as a route that already exists on the MCU.
44	<b>route rejected.</b> The call attempts to add a route to a forbidden subnet.
45	<b>too many routes.</b> The call can not add the route because doing so would exceed the allowed number of routes.
46	<b>no such route.</b> The MCU has no record of a route that has the provided <b>routeId</b> .
48	<b>IP address overflows prefix length.</b> The call attempts to make a route <b>destination</b> more specific than the range defined by the <b>prefixLength</b> .
49	<b>operation would disable active interface.</b>
60	<b>invalid manual IP settings.</b> Manual IP settings provided in the device. <b>network.modify</b> API call were invalid for some reason. Details of the specific reason are given in the fault string.
61	<b>cannot remove this feature key.</b> This is returned when a feature key cannot be removed - the reason why is stated in specific text in most cases.
62	<b>participant not currently active.</b> This is returned when attempting to disconnect an already disconnected participant using <b>participant.disconnect</b> .
101	<b>missing parameter.</b> 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	<b>invalid parameter.</b> 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	<b>malformed parameter.</b> 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	<b>mismatched parameters.</b> The call provides related parameters that, when considered together, are not expected/supported.
201	<b>operation failed.</b> 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
handshakeFailed	A VNC server exits the handshake process, normally due to the server not supporting the MCU's protocol version. For example, v3.3 does not support encryption, but the server is configured to use encryption.
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

Reason	Description
messageQueueOverflow	An excess of information in the message buffer has caused it to run out of space and overflow
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**

---

**Note:** This header (**Connection: Keep-Alive**) is not required for HTTP 1.1 clients.

---

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. An HTTP 1.1 client should use **Connection: close** if it explicitly does not want the connection to be maintained.

The device will not keep a connection alive if:

- the current connection has already been open for over the allowed amount of time (300 seconds) when a request comes in
- the number of persistent connections would exceed the allowed number (3) if this connection were to be maintained
- no requests have been received over the connection for a period of time (32 seconds)

These restrictions are in place to limit the resources associated with open connections. If a connection is terminated for the first reason, 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 32 seconds. There is little benefit to keeping unused connections open for such long periods.

---

## Conference layouts

Some API calls allow a particular layout to be sent to that participant via the `cpLayout` and `customLayout` parameters. The device can return information about the layout via the `cpLayout`, `customLayout` and `currentLayout` parameters.

The `cpLayout` parameter can take the following values:

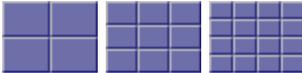
- **default**: use the MCU's default view family
- **family<index>**: use the specified layout family
- **layout<index>**: use a specific layout
- **conferenceCustom**: use the conference custom layout. Only valid if the `conferenceLayoutEnabled` parameter is set to `true` for the conference associated with the participant.

The `customLayout` and `currentLayout` parameters can take an integer value between 1 and 59. This corresponds to the index for the layout<index>.

## 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	
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" \[p.195\]](#), and it is configured with numeric id "1234"; the creation of "linked2" is shown in [Example message 2 - creating conference "linked2" on "mdu2" \[p.196\]](#), 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" \[p.197\]](#) - 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 \[p.198\]](#) 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>
      <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>
      <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>
```

# Index of parameters

---

Index of parameters: A .....	201
Index of parameters: B .....	210
Index of parameters: C .....	211
Index of parameters: D .....	225
Index of parameters: E .....	229
Index of parameters: F .....	232
Index of parameters: G .....	235
Index of parameters: H .....	237
Index of parameters: I .....	240
Index of parameters: J .....	244
Index of parameters: K .....	245
Index of parameters: L .....	246
Index of parameters: M .....	250
Index of parameters: N .....	254
Index of parameters: O .....	257
Index of parameters: P .....	259
Index of parameters: Q .....	264
Index of parameters: R .....	265
Index of parameters: S .....	272
Index of parameters: T .....	276
Index of parameters: U .....	279
Index of parameters: V .....	281
Index of parameters: W .....	286

# 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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

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

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

**activatedFeatures** is used in: [device.query \[p.91\]](#).

<b>activatedLicenses</b>	array	Array of activated licenses (i.e. not expired and ports number>0) (optional, only present if licenses installed and active)
<b>license</b>	string	License name
<b>license</b> is used in: <a href="#">device.query [p.91]</a> .		
<b>ports</b>	integer	Number of license ports installed
<b>ports</b> is used in: <a href="#">device.query [p.91]</a> .		
<b>expiry</b>	Date Time	(optional) Expiry date when present. Time is set at 23:59:59
<b>expiry</b> is used in: <a href="#">device.query [p.91]</a> .		
<b>activatedLicenses</b> is used in: <a href="#">device.query [p.91]</a> .		
<b>active (route)</b>	boolean	<b>true</b> if the route is currently active. <b>false</b> if the route is inactive (e.g. a route pointing to Port B when port B is disabled). Applies to configured routes only.
<b>active (route)</b> is used in: <a href="#">route.enumerate [p.157]</a> .		
<b>activeConferenceId</b>	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 <b>activeConferenceId</b> .
<b>activeConferenceId</b> is used in: <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.status [p.68]</a> , <a href="#">participant.enumerate [p.116]</a> , <a href="#">participant.status [p.143]</a> .		
<b>activeEndTime</b>	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.
<b>activeEndTime</b> is used in: <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.status [p.68]</a> .		
<b>active</b>	boolean	<b>true</b> to request only active conferences.
<b>active</b> is used in: <a href="#">conference.enumerate [p.48]</a> .		
<b>activeRegistrations</b>	integer	The number of active registrations.
<b>activeRegistrations</b> is used in: <a href="#">gatekeeper.query [p.104]</a> .		
<b>activeSpeaker</b>	boolean	<b>true</b> if the participant is currently the active speaker in the conference.
<b>activeSpeaker</b> is used in: <a href="#">participant.enumerate [p.116]</a> , <a href="#">participant.enumerate (deprecated) [p.125]</a> , <a href="#">participant.status [p.143]</a> , <a href="#">participant.status (deprecated) [p.151]</a> .		
<b>activeStartTime</b>	dateTime. iso8601	If the conference is currently active, this parameter contains the time that the current session started.

**activeStartTime** is used in: [conference.enumerate \[p.48\]](#), [conference.status \[p.68\]](#).

---

**actualBitRate** integer The measured bit rate of this stream, in bits per second (bps).

**actualBitRate** is used in: [participant.statistics \[p.137\]](#),

---

**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 \[p.107\]](#), [participant.modify \[p.131\]](#).

---

**addResponse** boolean **true** to return the details of the added participant.

**addResponse** is used in: [participant.add \[p.107\]](#).

---

**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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.modify \[p.131\]](#), [participant.statistics \[p.137\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#),

---

**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 \[p.102\]](#), [gatekeeper.query \[p.104\]](#).

---

**address (gateway)** string (63) The address of the gateway.

**address (gateway)** is used in: [gateway.enumerate \[p.106\]](#).

---

**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 \[p.168\]](#), [template.enumerate \[p.173\]](#), [template.modify \[p.177\]](#), [template.status \[p.181\]](#).

---

**alternateGatekeepers** integer The number of alternate gatekeepers

**alternateGatekeepers** is used in: [gatekeeper.query \[p.104\]](#).

---

**apiVersion** string The version number of the API implemented by this device.

**apiVersion** is used in: [device.query \[p.91\]](#).

---

**audioCodec** string The codec used on the audio stream. Either **RTSP** or **MMS**.

**audioCodec** is used in: [streaming.query \[p.167\]](#).

---

**audioControl** boolean Defaults to **false**. Set **true** to return **audioControl** statistics.

**audioControl** is used in: [participant.statistics \[p.137\]](#).

---

**audioLoad** integer A percentage value representing the proportion of the device's audio processing capacity that is currently in use. (Not returned on the MCU 5300 Series.)

**audioLoad** is used in: [device.health.query \[p.85\]](#).

---

**audioMedia** boolean Defaults to **false**. Set **true** to return **audioMedia** statistics.

**audioMedia** is used in: [participant.statistics \[p.137\]](#).

---

**audioRTCPOther** integer The number of other RTCP packets seen for the audio streams.

**audioRTCPOther** is used in: [conference.streaming.query \[p.74\]](#).

---

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

**audioRTCPPacketsSent** is used in: [conference.streaming.query \[p.74\]](#).

---

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

**audioRTCPReceiverReports** is used in: [conference.streaming.query \[p.74\]](#).

---

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

**audioRTCPSenderReports** is used in: [conference.streaming.query \[p.74\]](#).

---

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

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

---

**audioRxCCodec** string Receive audio codec.

**audioRxCCodec** is used in: [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**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 \[p.116\]](#), [participant.status \[p.143\]](#).

---

**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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate](#)

(deprecated) [p.125] [participant.modify](#) [p.131], [participant.status](#) [p.143], [participant.status \(deprecated\)](#) [p.151].

---

**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.107], [participant.enumerate](#) [p.116], [participant.enumerate \(deprecated\)](#) [p.125], [participant.modify](#) [p.131], [participant.status](#) [p.143], [participant.status \(deprecated\)](#) [p.151].

---

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

**audioRxLost** is used in: [participant.diagnostics](#) [p.112], [participant.enumerate](#) [p.116], [participant.enumerate \(deprecated\)](#) [p.125], [participant.status](#) [p.143], [participant.status \(deprecated\)](#) [p.151].

---

**audioRxMuted** boolean **true** means that audio from this participant will not be heard by other conference participants.

**audioRxMuted** is used in: [participant.add](#) [p.107], [participant.enumerate](#) [p.116], [participant.enumerate \(deprecated\)](#) [p.125], [participant.modify](#) [p.131], [participant.status](#) [p.143], [participant.status \(deprecated\)](#) [p.151].

---

**audioRxMutedRemotely** boolean Whether this endpoint is muted remotely.

**audioRxMutedRemotely** is used in: [participant.enumerate](#) [p.116], [participant.status](#) [p.143].

---

**audioRxReceived** integer Count of audio packets received by the MCU.

**audioRxReceived** is used in: [participant.diagnostics](#) [p.112], [participant.enumerate](#) [p.116], [participant.enumerate \(deprecated\)](#) [p.125], [participant.status](#) [p.143], [participant.status \(deprecated\)](#) [p.151].

---

**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.74].

---

**audioTx** struct A choice of audio codecs advertised by the MCU.

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

---

**audioTxCodec** string The codec used on the audio transmission.

**audioTxCodec** is used in: [participant.enumerate](#) [p.116], [participant.enumerate \(deprecated\)](#) [p.125], [participant.status](#) [p.143], [participant.status \(deprecated\)](#) [p.151].

---

**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.107], [participant.enumerate](#) [p.116], [participant.modify](#) [p.131], [participant.status](#) [p.143].

---

**audioTxReportedLost** integer The count of audio packets reported lost by the far end.

**audioTxReportedLost** is used in: [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**audioTxSent** integer Count of the audio packets sent to this endpoint.

**audioTxSent** is used in: [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**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 \[p.9\]](#).

---

**authenticationUser** string Name of a user with sufficient privilege for the operation being performed. The name is case sensitive.

**authenticationUser** is used in: [Authentication \[p.9\]](#).

---

**autoAttendantConfiguredName** string The name of the auto attendant.

**autoAttendantConfiguredName** is used in: [autoAttendant.enumerate \[p.33\]](#), [autoAttendant.status \[p.34\]](#), [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

---

**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 \[p.48\]](#), [conference.floor.query \[p.56\]](#), [conference.status \[p.68\]](#), [conference.streaming.query \[p.74\]](#), [participant.add \[p.107\]](#), [participant.connect \[p.111\]](#), [participant.diagnostics \[p.112\]](#), [participant.disconnect \[p.115\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.fecc \[p.129\]](#), [participant.message \[p.130\]](#), [participant.modify \[p.131\]](#), [participant.move \[p.135\]](#), [participant.remove \[p.136\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**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**.

---

**Note:** call-in matching is not supported for the H.323 ID.

---

Value	Description
true	When a participant that matches this call's <b>address</b> parameter dials in to the MCU, it is automatically moved to the conference identified by <b>conferenceName</b> 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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

---

**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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

---

**automatic**                                boolean      Controls automatic Call Home.

**true** enables automatic Call Home. **false** disables automatic Call Home. Only has effect when mode is **anonymous**.

Omit the parameter to leave the current setting unchanged.

**automatic** is used in: [callHome.configure \[p.35\]](#), [callHome.query \[p.36\]](#).

---

**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 <b>automaticLectureModeTimeout</b> . 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 <b>automaticLectureModeTimeout</b> parameter; it is always 0 for this automatic lecture mode.
disabled	The MCU never applies lecture mode. You do not need to provide the <b>automaticLectureModeTimeout</b> parameter; the MCU ignores it.

**automaticLectureMode** is used in: [conference.create \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

---

**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 <b>automaticLectureModeTimeout</b> . 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 <b>automaticLectureModeTimeout</b> parameter; it is always 0 for this automatic lecture mode.
disabled	The MCU never applies lecture mode. You do not need to provide the <b>automaticLectureModeTimeout</b> parameter; the MCU ignores it.
default	Inherit this setting from the parent template.

**automaticLectureMode** (*template*) is used in: [template.modify \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

---

**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 <b>automaticLectureModeTimeout</b> parameter is required.
false	Automatic lecture mode is disabled.

**automaticLectureModeEnabled** is used in: [conference.create \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

---

**automaticLectureModeEnabled** (*template*)      string      Defines whether automatic lecture mode is enabled for conferences based on this template. Deprecated by **automaticLectureMode** (*template*).

---

**Note:** This parameter is deprecated by **automaticLectureMode**.

Automatic lecture mode shows the speaker full screen.

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

**automaticLectureModeEnabled** (*template*) is used in: [template.modify \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

---

**automaticLectureModeTimeout** integer If **automaticLectureMode** is **type1**, 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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.modify \[p.177\]](#).

---

**availabilityThresholdConferences** string (8) or 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.

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.

**availabilityThresholdConferences** is used in: [gatekeeper.modify \[p.102\]](#), [gatekeeper.query \[p.104\]](#).

---

**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 \[p.102\]](#), [gatekeeper.query \[p.104\]](#).

# 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 \[p.74\]](#), [streaming.modify \[p.166\]](#), [streaming.query \[p.167\]](#).

**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 \[p.137\]](#).

**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 \[p.73\]](#), [conference.streaming.query \[p.74\]](#), [participant.add \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

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

**buildVersion** is used in: [device.query \[p.91\]](#).

**bytesReceived** integer The number of bytes received by the device.

**bytesReceived** is used in: [device.network.query \[p.88\]](#).

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

**bytesReceived64** is used in: [device.network.query \[p.88\]](#).

**bytesSent** integer The number of bytes sent by the device.

**bytesSent** is used in: [device.network.query \[p.88\]](#).

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

**bytesSent64** is used in: [device.network.query \[p.88\]](#).

# 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 \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

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

**callIdentifier** is used in: [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

---

**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 \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**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 \[p.116\]](#), [participant.status \[p.143\]](#).

---

<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 <b>disabled</b> , <b>feccOnly</b> , <b>dtmfOnly</b> , <b>feccWithDtmfFallback</b> , <b>bothFeccAndDtmf</b> , or <b>default</b> .
--	--------	--

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 \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

---

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

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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#), [participant.modify \[p.131\]](#).

---

<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 \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

---

<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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

---

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

**chairParticipant** is used in: [conference.enumerate \[p.48\]](#), [conference.floor.modify \[p.55\]](#), [conference.status \[p.68\]](#).

---

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

**channelBitRate** is used in: [participant.statistics \[p.137\]](#),

---

**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 \[p.41\]](#).

---

**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 \[p.91\]](#).

---

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

**codec** is used in: [conference.streaming.query \[p.74\]](#), [participant.statistics \[p.137\]](#),

---

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

**codecBitRate** is used in: [participant.statistics \[p.137\]](#).

---

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

**collisions** is used in: [device.network.query \[p.88\]](#).

---

**completed** boolean **true** if the conference has finished.

**completed** is used in: [conference.enumerate \[p.48\]](#).

<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: <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.status [p.68]</a> .		
<code>conferenceID</code>	string	Deprecated by <code>numericId</code> .
<code>conferenceID</code> is used in: <a href="#">conference.create [p.41]</a> , <a href="#">conference.modify [p.59]</a> .		
<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: <a href="#">template.modify [p.177]</a> , <a href="#">template.enumerate [p.173]</a> , <a href="#">template.create [p.168]</a> , <a href="#">template.status [p.181]</a> .		
<code>conferenceMeEnabled</code>	boolean	Whether or not ConferenceMe is enabled for this conference.
<code>conferenceMeEnabled</code> is used in: <a href="#">conference.create [p.41]</a> , <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.modify [p.59]</a> , <a href="#">conference.status [p.68]</a> .		
<code>conferenceName</code>	string	The name of the conference.
<code>conferenceName</code> is used in: <a href="#">conference.create [p.41]</a> , <a href="#">conference.destroy [p.46]</a> , <a href="#">conference.end [p.47]</a> , <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.floor.modify [p.55]</a> , <a href="#">conference.floor.query [p.56]</a> , <a href="#">conference.metadata.modify [p.57]</a> , <a href="#">conference.metadata.status [p.58]</a> , <a href="#">conference.modify [p.59]</a> , <a href="#">conference.paneplacement.modify [p.63]</a> , <a href="#">conference.paneplacement.query [p.65]</a> , <a href="#">conference.resetCleanupTimeout [p.67]</a> , <a href="#">conference.status [p.68]</a> , <a href="#">conference.streaming.modify [p.73]</a> , <a href="#">conference.streaming.query [p.74]</a> , <a href="#">participant.add [p.107]</a> , <a href="#">participant.connect [p.111]</a> , <a href="#">participant.diagnostics [p.112]</a> , <a href="#">participant.disconnect [p.115]</a> , <a href="#">participant.enumerate [p.116]</a> , <a href="#">participant.enumerate (deprecated) [p.125]</a> , <a href="#">participant.fecc [p.129]</a> , <a href="#">participant.message [p.130]</a> , <a href="#">participant.modify [p.131]</a> , <a href="#">participant.move [p.135]</a> , <a href="#">participant.remove [p.136]</a> , <a href="#">participant.statistics [p.137]</a> , <a href="#">participant.status [p.143]</a> , <a href="#">participant.status (deprecated) [p.151]</a> .		
<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 <b>Allow numeric ID registration for conferences</b> on the <b>Settings &gt; SIP</b> page of the web interface.
<code>conferenceRegistration</code> is used in: <a href="#">sip.modify [p.163]</a> , <a href="#">sip.query [p.165]</a> .		
<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: <a href="#">conference.enumerate [p.48]</a> ,		
<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: <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.status [p.68]</a> .		

<b>conferencingParameters</b>	struct	A structure containing the conferencing parameters of the enumerated item, e.g. gateway or endpoint.  <b>conferencingParameters</b> is used in: <a href="#">addressBookEntry.enumerate [p.24]</a> , <a href="#">gateway.enumerate [p.106]</a> .
<b>configuredProxy</b>	string (255)	The <b>outboundAddress</b> that SIP invites and registrations (deprecated) should be sent to.  <b>configuredProxy</b> is used in: <a href="#">sip.modify [p.163]</a> , <a href="#">sip.query [p.165]</a> .
<b>configuredRegistrar</b>	string (255)	The <b>outboundDomain</b> that should be used for SIP invites (deprecated) and registrations, also sets the <b>outboundAddress</b> if no proxy/address is provided.  <b>configuredRegistrar</b> is used in: <a href="#">sip.modify [p.163]</a> , <a href="#">sip.query [p.165]</a> .
<b>configuredState</b>	struct	The stored configuration of the participant, if it exists. <b>configuredState</b> is only present if requested in the <b>operationScope</b> .  <b>configuredState</b> is used in: <a href="#">participant.enumerate [p.116]</a> , <a href="#">participant.status [p.143]</a> .
<b>connected</b>	boolean	<b>true</b> if the participant is currently connected to a conference.  <b>connected</b> is used in: <a href="#">participant.enumerate [p.116]</a> , <a href="#">participant.enumerate (deprecated) [p.125]</a> .
<b>connecting</b>	boolean	<b>true</b> if the scheduled participant is in the process of connecting or is pending a retry. <b>connecting</b> is <b>true</b> for participants whose <b>callStateEx</b> values are <b>proceeding</b> , <b>alerting</b> , or <b>pending</b> . It may also be true for some participants whose <b>callState</b> (deprecated) is <b>dormant</b> or <b>disconnected</b> , because these values are also mapped to the new <b>proceeding</b> and <b>pending</b> states, respectively, that were introduced by the persistence feature in MCU 4.4.  <b>connecting</b> is used in: <a href="#">participant.enumerate [p.116]</a> , <a href="#">participant.enumerate (deprecated) [p.125]</a> .
<b>connectionUniqueId</b>	integer	Corresponds to the <b>uniqueId</b> returned by a conference or autoattendant.  <b>connectionUniqueId</b> is used in: <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.floor.query [p.56]</a> , <a href="#">conference.status [p.68]</a> , <a href="#">conference.streaming.query [p.74]</a> , <a href="#">participant.enumerate [p.116]</a> , <a href="#">participant.enumerate (deprecated) [p.125]</a> , <a href="#">participant.status [p.143]</a> , <a href="#">participant.status (deprecated) [p.151]</a> .
<b>connectPending</b>	boolean	<b>true</b> 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).  <b>connectPending</b> is used in: <a href="#">participant.enumerate [p.116]</a> , <a href="#">participant.enumerate (deprecated) [p.125]</a> , <a href="#">participant.status [p.143]</a> , <a href="#">participant.status (deprecated) [p.151]</a> .
<b>connectTime</b>	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 \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**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 \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

---

**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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

---

**contentControl**      boolean      Defaults to **false**. Set **true** to return **contentControl** statistics.

**contentControl** is used in: [participant.statistics \[p.137\]](#).

---

**contentEnabled**      string      One of **enabled**, **h239Only** or **disabled**.

**contentEnabled** is used in: [device.content.modify \[p.79\]](#), [device.content.query \[p.80\]](#).

---

**contentError**      string      Information about problems with outgoing content. One of: **notAllowed**, **noCommonCodecs**, **noCommonFormats**, **noCommonSymmetricCodecs**, **modeMismatch**, **bitRateMismatch**, **encryptionNotPossible**, **notPossible**.

**contentError** is used in: [participant.statistics \[p.137\]](#).

---

**contentHandoverEnabled**      boolean      **true** if automatic content handover is enabled.

**contentHandoverEnabled** is used in: [device.content.modify \[p.79\]](#), [device.content.query \[p.80\]](#).

---

**contentImportant**      boolean      Whether or not content is set to be important.

**contentImportant** is used in: [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

---

**contentInMainVideo**      boolean      **true** if the content can display in the main video channel.

**contentInMainVideo** is used in: [device.content.modify \[p.79\]](#), [device.content.query \[p.80\]](#).

---

**contentMarkupEnabled**      boolean      **true** if content markup is enabled.

**contentMarkupEnabled** is used in: [device.content.modify \[p.79\]](#), [device.content.query \[p.80\]](#).

---

**contentMedia**      boolean      Defaults to **false**. Set **true** to return **contentMedia** statistics.

**contentMedia** is used in: [participant.statistics \[p.137\]](#).

---

**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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#), [template.modify \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

---

**contentPassthroughLimit** string Possible values of `none`, `sd`, `hd`, and `fullhd`. This is the minimum resolution passed on to endpoints. If an endpoint that can only receive content below the minimum joins a conference, the endpoint will not be sent a passthrough content stream or a transcoded content stream. If the MCU receives content with a resolution below this setting, then it will use the received resolution as the minimum instead.

**contentPassthroughLimit** is used in: [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.create \[p.41\]](#).

---

**contentPassthroughLimit (template)** string One of `none`, `sd`, `hd`, `fullhd`, or `default` (inherit this setting from the parent template). This is the minimum resolution passed on to endpoints. If an endpoint that can only receive content below the minimum joins a conference, the endpoint will not be sent a passthrough content stream or a transcoded content stream. If the MCU receives content with a resolution below this setting, then it will use the received resolution as the minimum instead.

**contentPassthroughLimit (template)** is used in: [template.enumerate \[p.173\]](#), [template.modify \[p.177\]](#), [template.create \[p.168\]](#).

---

**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 \[p.112\]](#).

---

**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 \[p.112\]](#).

---

`contentRxChannelBitRate` integer Capacity of channel in bps

`contentRxChannelBitRate` is used in: [participant.diagnostics \[p.112\]](#).

---

`contentRxCodec` string The codec used on the incoming content stream.

`contentRxCodec` is used in: [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

---

`contentRxFrameRate` integer Frame rate of incoming content

`contentRxFrameRate` is used in: [participant.diagnostics \[p.112\]](#).

---

`contentRxFramesReceived` integer Number of received content frames

`contentRxFramesReceived` is used in: [participant.diagnostics \[p.112\]](#).

---

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

`contentRxFramesReceivedWithErrors` is used in: [participant.diagnostics \[p.112\]](#).

---

`contentRxHeight` integer Vertical resolution of incoming content

`contentRxHeight` is used in: [participant.diagnostics \[p.112\]](#).

---

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

`contentRxJitter` is used in: [participant.diagnostics \[p.112\]](#).

---

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

`contentRxLost` is used in: [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

---

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

`contentRxReceived` is used in: [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

---

`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 \[p.112\]](#).

---

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

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

`contentRxType` is used in: [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

`contentRxWidth` integer Horizontal resolution of incoming content.

`contentRxWidth` is used in: [participant.diagnostics \[p.112\]](#).

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

`contentStreamingSetting` is used in: [device.content.query \[p.80\]](#).

`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 \[p.80\]](#).

`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 \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

`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 \[p.41\]](#).

---

`contentTxActualBitRate` integer Actual speed of outgoing content in bps

`contentTxActualBitRate` is used in: [participant.diagnostics \[p.112\]](#).

---

`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 \[p.112\]](#).

---

`contentTxChannelBitRate` integer Capacity of channel in bps

`contentTxChannelBitRate` is used in: [participant.diagnostics \[p.112\]](#).

---

`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 \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

---

`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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#), [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

---

**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 \[p.112\]](#).

---

**contentTxFrameRate** integer Frame rate of outgoing content

**contentTxFrameRate** is used in: [participant.diagnostics \[p.112\]](#).

---

**contentTxHeight** integer Vertical resolution of outgoing content

**contentTxHeight** is used in: [participant.diagnostics \[p.112\]](#).

---

**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 \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

---

**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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

---

**contentTxReportedLost** integer Number of content packets reported as lost.

**contentTxReportedLost** is used in: [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

---

**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 \[p.112\]](#).

---

**contentTxSent** integer Number of content packets sent.

**contentTxSent** is used in: [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

---

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

Value	Description
none	MCU is not sending content. No other <b>contentTx</b> 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 <b>contentTx</b> fields will be returned.

**contentTxType** is used in: [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

---

**contentTxWidth** integer Horizontal resolution of outgoing content

**contentTxWidth** is used in: [participant.diagnostics \[p.112\]](#).

---

**contentType** string The type of content being sent or received.

**contentType** is used in: [participant.statistics \[p.137\]](#).

---

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

**count (videoports)** is used in: [device.query \[p.91\]](#).

---

**count** integer The number of users of this codec.

**count** is used in: [conference.streaming.query \[p.74\]](#).

---

**cpLayout** string This is the layout for the video sent to the participant. Refer to [Conference layouts \[p.193\]](#) for details.

**cpLayout** is used in: [conference.streaming.modify \[p.73\]](#), [participant.add \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**cpuLoad** integer The CPU load as a percentage of the maximum.

**cpuLoad** is used in: [device.health.query \[p.85\]](#).

---

**currentLayout** integer The actual layout in use for the video stream being sent by the MCU. Refer to [Conference layouts \[p.193\]](#) for details.

**currentLayout** is used in: [conference.streaming.query \[p.74\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**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.

**currentRevision** is used in: [autoAttendant.enumerate \[p.33\]](#), [conference.enumerate \[p.48\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#),

---

**currentState** struct The current state of the participant. This is only present if requested in the **operationScope**.

**currentState** is used in: [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

---

**currentTime** dateTime. The system's current date and time.  
iso8601

**currentTime** is used in: [cdrlog.enumerate \[p.38\]](#), [device.query \[p.91\]](#), [device.time.modify \[p.96\]](#), [device.time.query \[p.97\]](#).

---

**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**.

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

---

**customCodecSelection** 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 layout associated with the conference. This is seen by participants if they are using the conference custom layout.  
  
See [Conference layouts \[p.193\]](#) for a list of available layouts and corresponding index values.

**customLayout** is used in: [conference.create \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

---

**customLayoutEnabled** boolean **true** if the custom layout is enabled, **false** otherwise.

**customLayoutEnabled** is used in: [conference.create \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.status \[p.68\]](#), [conference.modify \[p.59\]](#).

# 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 \[p.88\]](#).

**defaultIpv4Gateway** string (79) The device's IPv4 default gateway in dotted quad format. Deprecates **defaultGateway**.  
**defaultIpv4Gateway** is used in: [device.network.modify \[p.86\]](#), [device.network.query \[p.88\]](#).

**defaultIpv6Gateway** string (79) The address of the IPv6 default gateway in CIDR format.  
**defaultIpv6Gateway** is used in: [device.network.modify \[p.86\]](#), [device.network.query \[p.88\]](#).

**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. <b>family2</b> , 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'. <b>layout3</b> , 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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

**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 \[p.40\]](#).

<b>deleteIndex</b> ( <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.  <code>deleteIndex</code> ( <i>CDR log</i> ) is used in: <a href="#">cdrlog.delete [p.37]</a> .
<b>deleteIndex</b> ( <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.  <code>deleteIndex</code> ( <i>audit log</i> ) is used in: <a href="#">auditlog.delete [p.30]</a> .
<b>description</b>	string	Additional information about the conference.  <code>description</code> is used in: <a href="#">conference.create [p.41]</a> , <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.modify [p.59]</a> , <a href="#">conference.status [p.68]</a> .
<b>destination</b>	string	IP address of the route's destination.  <code>destination</code> is used in: <a href="#">route.add [p.155]</a> , <a href="#">route.enumerate [p.157]</a> .
<b>dhcp</b>	boolean	Defines whether or not to use DHCP to obtain an IPv4 address.  <code>dhcp</code> is used in: <a href="#">device.network.query [p.88]</a> .
<b>dhcpv4</b>	boolean	Defines whether or not to use DHCP to obtain an IPv4 address. Deprecates <code>dhcp</code> .  <code>dhcpv4</code> is used in: <a href="#">device.network.modify [p.86]</a> , <a href="#">device.network.query [p.88]</a> .
<b>direction</b>	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> .  <code>direction</code> is used in: <a href="#">participant.fecc [p.129]</a> .
<b>disconnected</b>	boolean	<code>true</code> if the participant has been connected to a conference, but is now disconnected.  <code>disconnected</code> is used in: <a href="#">participant.enumerate [p.116]</a> , <a href="#">participant.enumerate (deprecated) [p.125]</a> .
<b>disconnectReason</b>	string	Only returned after the participant has disconnected; this contains one of the <a href="#">Disconnect reasons [p.190]</a> .  <code>disconnectReason</code> is used in: <a href="#">participant.enumerate [p.116]</a> , <a href="#">participant.enumerate (deprecated) [p.125]</a> , <a href="#">participant.status [p.143]</a> , <a href="#">participant.status (deprecated) [p.151]</a> .
<b>disconnectTime</b>	dateTime. iso8601	Only returned after the participant has disconnected.

**disconnectTime** is used in: [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**displayName** string The display name of the participant.

**displayName** is used in: [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**displayNameEx** string (63) The display name of the participant.

**displayNameEx** is used in: [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#),

---

**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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**displayNameOverrideValue** string (63) This value overrides the participant's display name if **displayNameOverrideStatus** is **true**.

**displayNameOverrideValue** is used in: [addressBookEntry.enumerate \[p.24\]](#), [participant.add \[p.107\]](#), [participant.modify \[p.131\]](#).

---

**dns** struct The struct members represent the device's DNS parameters.

**dns** is used in: [device.network.modify \[p.86\]](#), [device.network.query \[p.88\]](#).

---

**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 \[p.86\]](#).

---

**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 \[p.104\]](#).

---

**domainName** string (255) The domain name (DNS suffix).

**domainName** is used in: [device.network.modify \[p.86\]](#), [device.network.query \[p.88\]](#).

---

**domainName** string (255) The domain name (DNS suffix).

**domainName** is used in: [device.network.modify \[p.86\]](#), [device.network.query \[p.88\]](#).

---

**dormant** boolean **true** if the pre-configured participant is not trying to connect.

**dormant** is used in: [participant.enumerate \[p.116\]](#).

---

**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 \[p.168\]](#), [template.enumerate \[p.173\]](#), [template.modify \[p.177\]](#), [template.status \[p.181\]](#).

---

**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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

---

**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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

---

**durationSeconds**    integer                      The period of time, in seconds, for which this item is active (up to a maximum value of 8639999).

**durationSeconds** is used in: [conference.create \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#), [participant.message \[p.130\]](#).

# 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](#)

---

<b>e164</b>	string	An E.164 number.
<b>e164</b> is used in: <a href="#">addressBookEntry.enumerate [p.24]</a> .		
<b>enabled</b>	boolean	<b>true</b> if this feature or item is enabled.
<b>enabled</b> is used in: <a href="#">conference.floor.query [p.56]</a> , <a href="#">conference.paneplacement.modify [p.63]</a> , <a href="#">conference.paneplacement.query [p.65]</a> , <a href="#">conferenceme.query [p.78]</a> , <a href="#">device.encryption.query [p.82]</a> , <a href="#">device.network.query [p.88]</a> , <a href="#">services.query [p.162]</a> , <a href="#">streaming.query [p.167]</a> .		
<b>encryption</b>	boolean	Defines whether or not the received or transmitted stream is encrypted. This parameter could apply to content, audio or video streams.
<b>encryption</b> is used in: <a href="#">participant.statistics [p.137]</a> .		
<b>encryptionRequired</b> ( <i>template</i> )	string	The encryption setting for conferences based on this template, if the encryption feature key is enabled. If <b>true</b> , encryption is required for these conferences. Otherwise, encryption is optional. <b>default</b> causes the template to inherit this setting from its parent template.
<b>encryptionRequired</b> ( <i>template</i> ) is used in: <a href="#">template.create [p.168]</a> , <a href="#">template.enumerate [p.173]</a> , <a href="#">template.modify [p.177]</a> , <a href="#">template.status [p.181]</a> .		
<b>encryptionRequired</b>	boolean	The encryption setting for this conference, if the encryption feature key is enabled. If <b>true</b> , encryption is required for this conference. Otherwise, encryption is optional.
<b>encryptionRequired</b> is used in: <a href="#">conference.create [p.41]</a> , <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.modify [p.59]</a> , <a href="#">conference.status [p.68]</a> .		
<b>endTime</b>	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 <b>durationSeconds</b> instead.
<b>endTime</b> is used in: <a href="#">conference.create [p.41]</a> , <a href="#">conference.modify [p.59]</a> .		
<b>energyMillidB</b>	integer	The received audio energy in millidecibels.
<b>energyMillidB</b> is used in: <a href="#">participant.statistics [p.137]</a> .		
<b>enforceMaximumAudioPorts</b> ( <i>template</i> )	boolean	Defines whether conferences based on this template will enforce the <b>maximumAudioPorts</b> limit. One of <b>true</b> , <b>false</b> , or <b>default</b> .
Assumed to be <b>true</b> 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 \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

`enforceMaximumAudioPorts` boolean Defines whether the conference enforces the `maximumAudioPorts` limit. Assumed to be `true` if absent.

`enforceMaximumAudioPorts` is used in: [conference.create \[p.41\]](#), [conference.modify \[p.59\]](#).

`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 \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

`enforceMaximumVideoPorts` boolean Defines whether the conference enforces the `maximumVideoPorts` limit. Assumed to be `true` if absent.

`enforceMaximumVideoPorts` is used in: [conference.create \[p.41\]](#), [conference.modify \[p.59\]](#).

`enumerateFilter` string A filter expression. The enumeration results depend on the supplied expression.

`enumerateFilter` is used in: [conference.enumerate \[p.48\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#),

`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 \[p.48\]](#), [gateway.enumerate \[p.106\]](#), [participant.enumerate \[p.116\]](#),

[participant.enumerate \(deprecated\) \[p.125\]](#),

---

**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 \[p.86\]](#).

---

**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 \[p.38\]](#).

---

**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>
    </struct>
  </value>
</param>
```

**events** is used in: [feedbackReceiver.configure \[p.98\]](#), [feedbackReceiver.reconfigure \[p.100\]](#).

---

**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 \[p.38\]](#).

# 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](#)

<b>fanStatus</b>	string	One of <code>ok</code> , <code>outOfSpec</code> , or <code>critical</code> .
<b>fanStatus</b> is used in: <a href="#">device.health.query [p.85]</a> .		
<b>fanStatusWorst</b>	string	One of <code>ok</code> , <code>outOfSpec</code> , or <code>critical</code> .
<b>fanStatusWorst</b> is used in: <a href="#">device.health.query [p.85]</a> .		
<b>fecOverhead</b>	integer	Only returned if FEC (forward error correction) is negotiated and enabled. (Sent and received media.) This is the overhead of sending ClearPath FEC packets, expressed as a percentage in addition to the media stream.
<b>fecOverhead</b> is used in: <a href="#">participant.statistics [p.137]</a> .		
<b>fecRecovered</b>	integer	Only returned if FEC (forward error correction) is negotiated and enabled. (Only on receive media.) This is the number of lost packets recovered by ClearPath FEC.
<b>fecRecovered</b> is used in: <a href="#">participant.statistics [p.137]</a> .		
<b>filter (route)</b>	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> .
<b>filter (route)</b> is used in: <a href="#">route.enumerate [p.157]</a> .		
<b>filter</b>	struct	A struct that contains boolean switches to filter the statistics. All the switches default to <code>false</code> (do not return these statistics).
<b>filter (stats)</b> is used in: <a href="#">participant.statistics [p.137]</a> .		
<b>filter</b>	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>
<b>filter</b> is used in: <a href="#">cdrlog.enumerate [p.38]</a> .		
<b>finishedBootng</b>	boolean	<code>true</code> after the device is fully booted. Will not revert to <code>false</code> until a reboot starts.
<b>finishedBootng</b> is used in: <a href="#">device.query [p.91]</a> .		
<b>firstIndex</b>	integer	The index of the oldest stored event.
<b>firstIndex</b> is used in: <a href="#">auditlog.query [p.31]</a> , <a href="#">cdrlog.query [p.40]</a> .		
<b>floorParticipant</b>	struct	A structure that identifies which participant has the floor.

**floorParticipant** is used in: [conference.enumerate \[p.48\]](#), [conference.floor.modify \[p.55\]](#), [conference.floor.query \[p.56\]](#), [conference.status \[p.68\]](#).

---

**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 \[p.48\]](#), [conference.floor.modify \[p.55\]](#), [conference.floor.query \[p.56\]](#), [conference.status \[p.68\]](#).

---

**flowControlReceived** integer Count of flow control requests received.

**flowControlReceived** is used in: [participant.statistics \[p.137\]](#).

---

**flowControlSent** integer Count of flow control requests sent.

**flowControlSent** is used in: [participant.statistics \[p.137\]](#).

---

**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 \[p.73\]](#), [conference.streaming.query \[p.74\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

---

**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 \[p.73\]](#), [conference.streaming.query \[p.74\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

---

**format** string One of `wmp`, `qt64`, `qt70`, or `realPlayer`. The `format` determines the `audioCodec` and `videoCodec`.

**format** is used in: [streaming.modify \[p.166\]](#), [streaming.query \[p.167\]](#).

---

**format1** struct A struct whose contents define a streaming format.

**format1** is used in: [streaming.modify \[p.166\]](#), [streaming.query \[p.167\]](#).

---

**format2** struct A struct whose contents define a streaming format.

**format2** is used in: [streaming.modify \[p.166\]](#), [streaming.query \[p.167\]](#).

---

**frameErrors** integer Count of frames with errors in this stream.

**frameErrors** is used in: [participant.statistics \[p.137\]](#).

---

<b>frameRate</b>	integer	The frame rate of the video or content stream, in frames per second (fps).
<b>frameRate</b> is used in: <a href="#">participant.statistics [p.137]</a> ,		
<b>framesTransferred</b>	integer	Count of audio, video, or content frames received, depending on where the parameter occurs.
<b>framesTransferred</b> is used in: <a href="#">participant.statistics [p.137]</a> .		
<b>fullDuplex</b>	boolean	<b>true</b> if the port supports a full-duplex connection, <b>false</b> for half-duplex.
<b>fullDuplex</b> is used in: <a href="#">device.network.modify [p.86]</a> , <a href="#">device.network.query [p.88]</a> .		
<b>furFilteringEnabled</b>	boolean	<b>true</b> if video fast update request filtering is enabled.
<b>furFilteringEnabled</b> is used in: <a href="#">device.content.modify [p.79]</a> , <a href="#">device.content.query [p.80]</a> .		
<b>fursReceived</b>	integer	Count of fast update requests (FURs) received by the device (this statistic is only present for video or content control).
<b>fursReceived</b> is used in: <a href="#">participant.statistics [p.137]</a> .		
<b>fursSent</b>	integer	Count of fast update requests (FURs) sent by the device (this statistic is only present for video or content control).
<b>fursSent</b> is used in: <a href="#">participant.statistics [p.137]</a> .		

---

# 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 \[p.102\]](#), [gatekeeper.query \[p.104\]](#).

**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 \[p.155\]](#), [route.enumerate \[p.157\]](#).

---

**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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#),

---

**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 \[p.106\]](#).

---

**guest** boolean **true** if the participant is a guest, **false** if the participant is a chair.

**guest** is used in: [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

---

**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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

---

**guestPin** string Security PIN that a guest can use to gain access to this conference.

**guestPin** is used in: [conference.create \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

# 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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#), [template.modify \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

---

**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 \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

---

**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
<b>As master</b>	The MCU only acts as master in H.239 token negotiation.
<b>As slave</b>	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.
<b>Mimic slave</b>	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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

---

**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 The H.323 ID used by the device to register with the gatekeeper.

**h323ID** is used in: [gatekeeper.modify \[p.102\]](#), [gatekeeper.query \[p.104\]](#).

---

**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 \[p.104\]](#).

---

**height** integer The maximum width and height of this stream. Only present for defined video streams

**height** is used in: [conference.streaming.query \[p.74\]](#), [participant.statistics \[p.137\]](#).

---

**hostName** string (255) The host name of queried device. Deprecated in API version 2.8.

**hostName** is used in: [device.network.modify \[p.86\]](#), [device.network.query \[p.88\]](#).

# 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 \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

**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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

**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 \[p.168\]](#), [template.enumerate \[p.173\]](#), [template.modify \[p.177\]](#), [template.status \[p.181\]](#).

**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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

**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 \[p.168\]](#), [template.enumerate \[p.173\]](#), [template.modify \[p.177\]](#), [template.status \[p.181\]](#).

**index** (*CDR log*)      integer      The index of the CDR log message.

**index** (*CDR log*) is used in: [cdrlog.enumerate \[p.38\]](#).

**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 \[p.38\]](#).

**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 \[p.99\]](#).

**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 \[p.63\]](#), [conference.paneplacement.query \[p.65\]](#).

**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 \[p.24\]](#).

<b>Interlaced</b>	boolean	Defines whether or not the video in this sent or received stream is interlaced. Deprecates <b>videoTxInterlaced</b> and <b>videoRxInterlaced</b> .
<b>Interlaced</b> is used in: <a href="#">participant.statistics [p.137]</a> .		
<b>ip</b>	string	the IP address of the gatekeeper (if <b>dnsStatus</b> is <b>resolved</b> )
<b>ip</b> is used in: <a href="#">gatekeeper.query [p.104]</a> .		
<b>ipAddress</b>	string	IPv4 address in dotted-quad format.
<b>ipAddress</b> is used in: <a href="#">device.network.query [p.88]</a> , <a href="#">participant.enumerate [p.116]</a> , <a href="#">participant.status [p.143]</a> .		
<b>ipRangeFinish</b>	string	The last IP address in the multicast range.
<b>ipRangeFinish</b> is used in: <a href="#">streaming.modify [p.166]</a> , <a href="#">streaming.query [p.167]</a> .		
<b>ipRangeStart</b>	string	The first IP address in the multicast range.
<b>ipRangeStart</b> is used in: <a href="#">streaming.modify [p.166]</a> , <a href="#">streaming.query [p.167]</a> .		
<b>ipv4Address</b>	string (79)	IPv4 address in dotted-quad format.
<b>ipv4Address</b> is used in: <a href="#">device.network.modify [p.86]</a> , <a href="#">device.network.query [p.88]</a> .		
<b>ipv4Enabled</b>	boolean	<b>true</b> if IPv4 interface is enabled.
<b>ipv4Enabled</b> is used in: <a href="#">device.network.modify [p.86]</a> , <a href="#">device.network.query [p.88]</a> .		
<b>ipv4MulticastRange</b>	struct	Contains parameters that define an IPv4 multicast range.
<b>ipv4MulticastRange</b> is used in: <a href="#">streaming.modify [p.166]</a> , <a href="#">streaming.query [p.167]</a> .		
<b>ipv4Preference</b>	string	Either <b>A</b> or <b>B</b> , indicates which Ethernet port is preferred for traffic bound for IPv4 destinations.
<b>ipv4Preference</b> is used in: <a href="#">route.preferences.modify [p.158]</a> , <a href="#">route.preferences.query [p.159]</a> .		
<b>ipv4Routes</b>	array	An array of structs, each of which represents an IPv4 route.
<b>ipv4Routes</b> is used in: <a href="#">route.enumerate [p.157]</a> .		
<b>ipv4SubnetMask</b>	string (31)	The IPv4 subnet mask in dotted quad format. Deprecates <b>subnetMask</b> .
<b>ipv4SubnetMask</b> is used in: <a href="#">device.network.modify [p.86]</a> , <a href="#">device.network.query [p.88]</a> .		
<b>ipv6Address</b>	string (79)	The IPv6 address in CIDR format.
<b>ipv6Address</b> is used in: <a href="#">device.network.modify [p.86]</a> , <a href="#">device.network.query [p.88]</a> .		
<b>ipv6Conf</b>	string (10)	Indicates how the IPv6 address is assigned; either <b>automatic</b> (by SLAAC/DHCPv6) or <b>manual</b> .
<b>ipv6Conf</b> is used in: <a href="#">device.network.modify [p.86]</a> , <a href="#">device.network.query [p.88]</a> .		
<b>ipv6Enabled</b>	boolean	<b>true</b> if IPv6 interface is enabled.

**ipv6Enabled** is used in: [device.network.modify \[p.86\]](#), [device.network.query \[p.88\]](#).

---

<b>ipv6MulticastRange</b>	struct	Contains parameters that define an IPv6 multicast range.
---------------------------	--------	--

**ipv6MulticastRange** is used in: [streaming.modify \[p.166\]](#), [streaming.query \[p.167\]](#).

---

<b>ipv6Preference</b>	string	Either <b>A</b> or <b>B</b> , indicates which Ethernet port is preferred for traffic bound for IPv6 destinations.
-----------------------	--------	---

**ipv6Preference** is used in: [route.preferences.modify \[p.158\]](#), [route.preferences.query \[p.159\]](#).

---

<b>ipv6PrefixLength</b>	integer	The length of the IPv6 address prefix.
-------------------------	---------	--

**ipv6PrefixLength** is used in: [device.network.modify \[p.86\]](#), [device.network.query \[p.88\]](#).

---

<b>ipv6Routes</b>	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 \[p.157\]](#).

# Index of parameters: J

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

<b> jitter </b>	integer	Current jitter in this stream, measured in milliseconds (ms).
<i> jitter </i> is used in: <a href="#">participant.statistics [p.137]</a> ,		
<b> jitterBuffer </b>	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.
<i> jitterBuffer </i> is used in: <a href="#">participant.statistics [p.137]</a> .		
<b> joinAGC </b>	boolean	Whether AGC should be used by default for participants joining this conference
<i> joinAGC </i> is used in: <a href="#">conference.create [p.41]</a> , <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.modify [p.59]</a> , <a href="#">conference.status [p.68]</a> .		
<b> joinAGC (template) </b>	string	Whether AGC should be used by default for participants joining this conference. One of <b>true</b> , <b>false</b> , or <b>default</b> if this template inherits the <b>joinAGC</b> setting of its parent template.
<i> joinAGC (template) </i> is used in: <a href="#">template.create [p.168]</a> , <a href="#">template.enumerate [p.173]</a> , <a href="#">template.modify [p.177]</a> , <a href="#">template.status [p.181]</a> .		
<b> joinAudioMuted (template) </b>	string	Mutes audio on join. One of <b>true</b> , <b>false</b> , or <b>default</b> to inherit this setting from the parent template.
<i> joinAudioMuted (template) </i> is used in: <a href="#">template.modify [p.177]</a> , <a href="#">template.enumerate [p.173]</a> , <a href="#">template.create [p.168]</a> , <a href="#">template.status [p.181]</a> .		
<b> joinAudioMuted </b>	boolean	Audio mute on join.
<i> joinAudioMuted </i> is used in: <a href="#">conference.create [p.41]</a> , <a href="#">conference.enumerate [p.48]</a> .		
<b> joinVideoMuted (template) </b>	string	Mutes video on join. One of <b>true</b> , <b>false</b> , or <b>default</b> to inherit this setting from the parent template.
<i> joinVideoMuted (template) </i> is used in: <a href="#">template.modify [p.177]</a> , <a href="#">template.enumerate [p.173]</a> , <a href="#">template.create [p.168]</a> , <a href="#">template.status [p.181]</a> .		
<b> joinVideoMuted </b>	boolean	Video mute on join.
<i> joinVideoMuted </i> is used in: <a href="#">conference.create [p.41]</a> , <a href="#">conference.enumerate [p.48]</a> .		

# Index of parameters: K

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

Parameter name	Type	Short description
<code>key</code>	string	The feature/license key to add/remove.

# 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 \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

**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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

**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 \[p.48\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#),

**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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [participant.add \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

`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	<b>Deprecated by <code>feccWithDtmfFallback</code>.</b> 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 \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

`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	<b>Deprecated by <code>feccWithDtmfFallback</code>.</b> 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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#), [participant.add \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

`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 \[p.74\]](#), [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

`lecturer` boolean `true` if the participant is the lecturer.

`lecturer` is used in: [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

`linkLocalIpv6Address` string(63) The link local IPv6 address in CIDR format.

`linkLocalIpv6Address` is used in: [device.network.query \[p.88\]](#).

`linkLocalIpv6PrefixLength` integer Length of the link local IPv6 address prefix.

`linkLocalIpv6PrefixLength` is used in: [device.network.query \[p.88\]](#).

`linkStatus` boolean `true` if the ethernet connection to this port is active.

`linkStatus` is used in: [device.network.query \[p.88\]](#).

`linkType` string This parameter is ignored unless `participantType` is `by_address`. Either `cascadeSlaveToMaster` or `default`

`linkType` is used in: [participant.add \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

---

<b>lipSyncDelayApplied</b>	integer	The amount of delay added to either audio or video output stream to correct for <b>rtcpLipSyncDelay</b> reported between incoming audio and video streams.
----------------------------	---------	--

**lipSyncDelayApplied** is used in: [participant.statistics \[p.137\]](#).

---

<b>locked</b>	boolean	Defines whether or not the conference is locked.
---------------	---------	--

**locked** is used in: [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#),

---

<b>log</b>	array	Each member of the array contains log information (called system log in the user interface).
------------	-------	--

**log** is used in: [device.restartlog.query \[p.94\]](#).

# 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](#)

<b>macAddress</b>	string	The MAC address of this interface. A 12 character string of hex digits with no separators.
<b>macAddress</b> is used in: <a href="#">device.network.query [p.88]</a> .		
<b>maxBitRateFromMCU</b>	integer	Maximum bandwidth from the MCU (kbps).
<b>maxBitRateFromMCU</b> is used in: <a href="#">confrenceme.modify [p.77]</a> , <a href="#">confrenceme.query [p.78]</a> , <a href="#">participant.add [p.107]</a> , <a href="#">participant.enumerate [p.116]</a> , <a href="#">participant.enumerate (deprecated) [p.125]</a> , <a href="#">participant.modify [p.131]</a> , <a href="#">participant.status [p.143]</a> , <a href="#">participant.status (deprecated) [p.151]</a> .		
<b>maxBitRateToMCU</b>	integer	Maximum bandwidth to the MCU (kbps).
<b>maxBitRateToMCU</b> is used in: <a href="#">confrenceme.modify [p.77]</a> , <a href="#">confrenceme.query [p.78]</a> , <a href="#">participant.add [p.107]</a> , <a href="#">participant.enumerate [p.116]</a> , <a href="#">participant.enumerate (deprecated) [p.125]</a> , <a href="#">participant.modify [p.131]</a> , <a href="#">participant.status [p.143]</a> , <a href="#">participant.status (deprecated) [p.151]</a> .		
<b>maxConferenceSize</b>	integer	The maximum number of participants that can be hosted in a single conference at the time of the response.
<b>maxConferenceSize</b> is used in: <a href="#">device.query [p.91]</a> .		
<b>maximumAudioPorts</b>	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.
<b>maximumAudioPorts</b> is used in: <a href="#">conference.create [p.41]</a> , <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.modify [p.59]</a> , <a href="#">conference.status [p.68]</a> , <a href="#">template.modify [p.177]</a> , <a href="#">template.enumerate [p.173]</a> , <a href="#">template.create [p.168]</a> , <a href="#">template.status [p.181]</a> .		
<b>maximumVideoPorts</b>	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.
<b>maximumVideoPorts</b> is used in: <a href="#">conference.create [p.41]</a> , <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.modify [p.59]</a> , <a href="#">conference.status [p.68]</a> , <a href="#">template.modify [p.177]</a> , <a href="#">template.enumerate [p.173]</a> , <a href="#">template.create [p.168]</a> , <a href="#">template.status [p.181]</a> .		
<b>maxMediaRxBitRate</b>	integer	The maximum media reception speed of this device, in kbps. 0 means the device uses the default.
<b>maxMediaRxBitRate</b> is used in: <a href="#">addressBookEntry.enumerate [p.24]</a> , <a href="#">gateway.enumerate [p.106]</a> .		
<b>maxMediaTxBitRate</b>	integer	The maximum media transmission speed from this device, in kbps. 0 means the device uses the default.
<b>maxMediaTxBitRate</b> is used in: <a href="#">addressBookEntry.enumerate [p.24]</a> , <a href="#">gateway.enumerate [p.106]</a> .		
<b>maxOcsBitrate</b>	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 \[p.163\]](#), [sip.query \[p.165\]](#).

---

**maxParticipants** integer The maximum number of ConferenceMe connections allowed.

**maxParticipants** is used in: [conferenceme.modify \[p.77\]](#), [conferenceme.query \[p.78\]](#).

---

**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 \[p.91\]](#).

---

**mcuServicePrefix** string The service prefix used by the MCU.

**mcuServicePrefix** is used in: [gatekeeper.modify \[p.102\]](#), [gatekeeper.query \[p.104\]](#).

---

**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 \[p.104\]](#).

---

**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 \[p.116\]](#), [participant.status \[p.143\]](#).

<b>mediaLoad</b>	integer	A percentage value representing the proportion of the device's media processing capacity that is currently in use.  <b>mediaLoad</b> is used in: <a href="#">device.health.query [p.85]</a> .
<b>mediaOverTcp</b>	boolean	<b>true</b> allows ConferenceMe to fall back to media over TCP if it cannot do media over UDP.  <b>mediaOverTcp</b> is used in: <a href="#">conferenceme.modify [p.77]</a> , <a href="#">conferenceme.query [p.78]</a> .
<b>mediaResources</b>	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.  <b>mediaResources</b> is used in: <a href="#">device.query [p.91]</a> .
<b>mediaResourceRestarts</b>	integer	The number of DSP/compute card restarts that have occurred since the unit last booted  <b>mediaResourceRestarts</b> is used in: <a href="#">device.query [p.91]</a> .
<b>message</b>	string (255)	The string to send to the participant.  <b>message</b> is used in: <a href="#">participant.message [p.130]</a> .
<b>metadata</b>	string (4095)	A string of up to 4095 unicode characters stored on the device and associated with the named conference.  <b>metadata</b> is used in: <a href="#">conference.metadata.modify [p.57]</a> , <a href="#">conference.metadata.status [p.58]</a> .
<b>minFrameRateMotionSharpness</b>	integer	Specifies the minimum frame rate for this endpoint. This parameter is only present if <b>useDefaultMotionSharpness</b> is <b>false</b> .  <b>minFrameRateMotionSharpness</b> is used in: <a href="#">addressBookEntry.enumerate [p.24]</a> , <a href="#">gateway.enumerate [p.106]</a> .
<b>mode</b>	string	Set the Call Home mode. One of <b>disabled</b> or <b>anonymous</b> . Can only be set to <b>anonymous</b> if the encryption feature key is present. Defaults to <b>disabled</b> if it has never been configured.  Omit the parameter to leave the current setting unchanged.  <b>mode</b> is used in: <a href="#">callHome.configure [p.35]</a> , <a href="#">callHome.query [p.36]</a> .
<b>model</b>	string	The model number.  <b>model</b> is used in: <a href="#">device.query [p.91]</a> .
<b>moreThanFour</b>	boolean	Enables the call to return more than four conferences (up to 24).  <b>moreThanFour</b> is used in: <a href="#">conference.enumerate [p.48]</a> .
<b>motionSharpnessTradeoff</b>	string	Defines preference for motion vs. sharpness. One of <b>preferMotion</b> , <b>preferSharpness</b> , <b>balanced</b> , or <b>default</b> .



# 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 \[p.106\]](#).

---

**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
<b>http</b>	
<b>https</b>	
<b>ftp</b>	
<b>h225</b>	Not supported on slaves.
<b>rtsp</b>	Not supported on slaves.
<b>mms</b>	Not supported on slaves.
<b>sip_tcp</b>	Not supported on slaves.
<b>sips_tcp</b>	Not supported on slaves.
<b>cdep</b>	Requires ConferenceMe activation. Not supported on slaves.
<b>sip_udp</b>	Not supported on slaves.
<b>snmp</b>	
<b>gatekeeper</b>	Not supported on slaves.
<b>tunnel</b>	Requires ConferenceMe activation. Not supported on slaves.

**name (service)** is used in: [services.modify \[p.160\]](#), [services.query \[p.162\]](#).

---

**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 \[p.86\]](#), [device.network.query \[p.88\]](#).

---

**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 \[p.86\]](#), [device.network.query \[p.88\]](#).

---

**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 \[p.59\]](#), [participant.move \[p.135\]](#).

---

<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 \[p.41\]](#), [conference.modify \[p.59\]](#).

---

<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 \[p.155\]](#).

---

<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 \[p.177\]](#).

---

<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 \[p.38\]](#).

---

<code>ntpEnabled</code>	boolean	Defines whether or not the device may synchronize with an NTP server.
-------------------------	---------	---

`ntpEnabled` is used in: [device.time.modify \[p.96\]](#), [device.time.query \[p.97\]](#).

---

<code>ntpHost</code>	string	DNS or IP address of an NTP server
----------------------	--------	------------------------------------

`ntpHost` is used in: [device.time.modify \[p.96\]](#), [device.time.query \[p.97\]](#).

---

<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 \[p.97\]](#).

---

<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 \[p.41\]](#), [conference.modify \[p.59\]](#).

---

<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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#),

---

<code>numEvents (audit log)</code>	integer	The total number of events stored.
------------------------------------	---------	------------------------------------

**numEvents** (*audit log*) is used in: [auditlog.query \[p.31\]](#).

---

<b>numEvents</b> ( <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 \[p.40\]](#).

---

<b>numEvents</b> ( <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 \[p.38\]](#).

# 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 \[p.59\]](#).

---

**operationalStatus** string One of `active`, `shuttingDown`, or `shutdown`.

`operationalStatus` is used in: [device.health.query \[p.85\]](#).

---

**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
<code>activeState</code>	The operation scope is limited to the active configuration of the participant.
<code>currentState</code>	The operation scope is limited to the active configuration of the participant.
<code>configuredState</code>	The operation scope is limited to the stored configuration of the participant.
Both <code>activeState</code> and <code>configuredState</code>	The scope is not limited to either state. That is, the participant structure will contain a <code>currentState</code> and <code>configuredState</code> structure, but the structures may be empty if the endpoints are not active or preconfigured, respectively.

`operationScope` is used in: [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

---

**outboundConfiguration** String (registrar, trunk, direct) Defines how outbound SIP calls will be made. Replaces the functionality of `registrarUsage`.

`outboundConfiguration` is used in: [sip.modify \[p.163\]](#), [sip.query \[p.165\]](#).

---

**outboundAddress** String The address all outbound SIP calls should be sent to when in registrar or trunk mode. Replaces `configuredProxy`

`outboundAddress` is used in: [sip.modify \[p.163\]](#), [sip.query \[p.165\]](#).

---

**outboundDomain** String The domain that should be used during SIP invites and registration. Replaces `configuredRegistrar`.

**outboundDomain** is used in: [sip.modify \[p.163\]](#), [sip.query \[p.165\]](#).

---

**outgoingTransport** string The outgoing transport protocol. One of **udp**, **tcp**, or **tls**.

**outgoingTransport** is used in: [sip.modify \[p.163\]](#), [sip.query \[p.165\]](#).

# 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 \[p.116\]](#), [participant.status \[p.143\]](#).

---

**packetLossWarning**                      boolean      This will be true if any packet loss has been seen within the last 15 seconds.

**packetLossWarning** is used in: [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

---

**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 \[p.137\]](#).

---

**packetsReceived**                              integer      The number of packets received on this Ethernet port.

**packetsReceived** is used in: [device.network.query \[p.88\]](#).

---

**packetsSent**                                      integer      The number of packets sent from this Ethernet port.

**packetsSent** is used in: [device.network.query \[p.88\]](#).

---

**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 \[p.137\]](#).

---

**panes**    array        An array of **structs**, each of which defines a particular pane within the layout.

**panes** is used in: [conference.paneplacement.modify \[p.63\]](#), [conference.paneplacement.query \[p.65\]](#).

---

**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 \[p.63\]](#).

---

**parent**    string      The name of the parent template. Defaults to Top Level template if omitted.

**parent** is used in: [template.create \[p.168\]](#), [template.modify \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.status \[p.181\]](#).

---

**participant**                                      struct      Contains the parameters that, when considered together, uniquely identify a participant.

**participant** is used in: [participant.add \[p.107\]](#), [participant.statistics \[p.137\]](#).

---

**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 \[p.48\]](#), [conference.floor.modify \[p.55\]](#), [conference.floor.query \[p.56\]](#), [conference.paneplacement.modify \[p.63\]](#), [conference.paneplacement.query \[p.65\]](#), [conference.status \[p.68\]](#), [conference.streaming.modify \[p.73\]](#), [conference.streaming.query \[p.74\]](#), [participant.add \[p.107\]](#), [participant.connect \[p.111\]](#), [participant.diagnostics \[p.112\]](#), [participant.disconnect \[p.115\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.fecc \[p.129\]](#), [participant.message \[p.130\]](#), [participant.modify \[p.131\]](#), [participant.move \[p.135\]](#), [participant.remove \[p.136\]](#), [participant.statistics \[p.137\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**participantProtocol** string **h323, sip, or vnc.**

**participantProtocol** is used in: [conference.enumerate \[p.48\]](#), [conference.floor.modify \[p.55\]](#), [conference.floor.query \[p.56\]](#), [conference.paneplacement.modify \[p.63\]](#), [conference.paneplacement.query \[p.65\]](#), [conference.status \[p.68\]](#), [conference.streaming.modify \[p.73\]](#), [conference.streaming.query \[p.74\]](#), [participant.add \[p.107\]](#), [participant.connect \[p.111\]](#), [participant.diagnostics \[p.112\]](#), [participant.disconnect \[p.115\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.fecc \[p.129\]](#), [participant.message \[p.130\]](#), [participant.modify \[p.131\]](#), [participant.move \[p.135\]](#), [participant.remove \[p.136\]](#), [participant.statistics \[p.137\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#),

---

**participants** array An array of structures that represent participants.

**participants** is used in: [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#),

---

**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 <b>participant.add</b> will be of type <b>by_address</b> unless they are explicitly added as temporary <b>ad_hoc</b> 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 \[p.48\]](#), [conference.floor.modify \[p.55\]](#), [conference.floor.query \[p.56\]](#), [conference.paneplacement.modify \[p.63\]](#), [conference.paneplacement.query \[p.65\]](#), [conference.status \[p.68\]](#), [conference.streaming.modify \[p.73\]](#), [conference.streaming.query \[p.74\]](#), [participant.add \[p.107\]](#), [participant.connect \[p.111\]](#), [participant.diagnostics \[p.112\]](#), [participant.disconnect \[p.115\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.fecc \[p.129\]](#), [participant.message \[p.130\]](#), [participant.modify \[p.131\]](#), [participant.move \[p.135\]](#), [participant.remove \[p.136\]](#), [participant.statistics \[p.137\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#),

---

**password (SIP)** string (63) The password used for SIP registration.

**password (SIP)** is used in: [sip.modify \[p.163\]](#).

---

**password** string The password for VNC endpoints.

**password** is used in: [addressBookEntry.enumerate \[p.24\]](#), [participant.add \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

---

**password** (*gatekeeper*)                      string              The password that the device uses to register with the gatekeeper, if required.

**password** (*gatekeeper*) is used in: [gatekeeper.modify \[p.102\]](#).

---

**pendingRegistrations**                      integer              The number of registrations in progress

**pendingRegistrations** is used in: [gatekeeper.query \[p.104\]](#).

---

**percentageCapacity**                      integer              The percentage of the total available capacity being used by the log.

**percentageCapacity** is used in: [auditlog.query \[p.31\]](#), [cdrlog.query \[p.40\]](#).

---

**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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#),

---

**port** (*IP*)                                      integer              Identifies the IP port.

**port** (*service*) is used in: [participant.statistics \[p.137\]](#), [services.modify \[p.160\]](#), [services.query \[p.162\]](#).

---

**port** (*Ethernet*)                              string              Identifies the Ethernet port. May be A or B.

**port** is used in: [route.enumerate \[p.157\]](#), [services.modify \[p.160\]](#), [services.query \[p.162\]](#).

---

**portA**    struct              A structure that contains configuration and status information for Ethernet port A on the device.

**portA** is used in: [device.network.modify \[p.86\]](#), [device.network.query \[p.88\]](#).

---

**portAssociationA**                              boolean              **true** if interface 'PortA IPv4' is associated with the H.323 gatekeeper.

**portAssociationA** is used in: [gatekeeper.query \[p.104\]](#).

---

**portAssociationAv4**                              boolean              **true** if interface 'PortA IPv4' is associated with the H.323 gatekeeper.

**portAssociationAv4** is used in: [gatekeeper.modify \[p.102\]](#), [gatekeeper.query \[p.104\]](#).

---

**portAssociationAv6**                              boolean              **true** if interface 'PortA IPv6' is associated with the H.323 gatekeeper.

**portAssociationAv6** is used in: [gatekeeper.modify \[p.102\]](#), [gatekeeper.query \[p.104\]](#).

---

**portAssociationB**                              boolean              **true** if interface 'PortB IPv4' is associated with the H.323 gatekeeper.

**portAssociationB** is used in: [gatekeeper.query \[p.104\]](#).

---

**portAssociationBv4**                              boolean              **true** if interface 'PortB IPv4' is associated with the H.323 gatekeeper.

**portAssociationBv4** is used in: [gatekeeper.modify \[p.102\]](#), [gatekeeper.query \[p.104\]](#).

---

**portAssociationBv6**                      boolean    **true** if interface 'PortB IPv6' is associated with the H.323 gatekeeper.

**portAssociationBv6** is used in: [gatekeeper.modify \[p.102\]](#), [gatekeeper.query \[p.104\]](#).

---

**portB**                                      struct      A structure that contains configuration and status information for Ethernet port B on the device.

**portB** is used in: [device.network.modify \[p.86\]](#), [device.network.query \[p.88\]](#).

---

**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 \[p.166\]](#), [streaming.query \[p.167\]](#).

---

**portRangeStart**                         integer     The first port number in the multicast port range.

**portRangeStart** is used in: [streaming.modify \[p.166\]](#), [streaming.query \[p.167\]](#).

---

**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 \[p.91\]](#).

---

**ports**                                      array        An array whose members are structures representing the Ethernet ports on the device

**ports** is used in: [services.modify \[p.160\]](#), [services.query \[p.162\]](#).

---

**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 \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

---

**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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

`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 \[p.155\]](#), [route.enumerate \[p.157\]](#).

`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 \[p.116\]](#), [participant.status \[p.143\]](#).

`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 \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

`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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

`protocol (IP)` string **IPv4** or **IPv6**.

`protocol (IP)` is used in: [services.modify \[p.160\]](#), [services.query \[p.162\]](#).

`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 \[p.88\]](#).

# 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](#)

---

<b>reason</b>	string	<p>An explanation for the restart. One of:</p> <ul style="list-style-type: none"> <li>▪ <code>User requested shutdown</code></li> <li>▪ <code>User requested reboot from web interface</code></li> <li>▪ <code>User requested upgrade</code></li> <li>▪ <code>User requested reboot from console</code></li> <li>▪ <code>User requested reboot from API</code></li> <li>▪ <code>User requested reboot from FTP</code></li> <li>▪ <code>User requested shutdown from supervisor</code></li> <li>▪ <code>User requested reboot from supervisor</code></li> <li>▪ <code>User reset configuration</code></li> <li>▪ <code>Cold boot</code></li> <li>▪ <code>unknown</code></li> </ul>
---------------	--------	---

**reason** is used in: [device.restartlog.query \[p.94\]](#).

---

<b>rebootRequired</b>	boolean	The device returns this parameter as <code>true</code> 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 \[p.91\]](#).

---

<b>receiveErrors</b>	integer	The count of receive errors on this interface.
----------------------	---------	--

**receiveErrors** is used in: [device.network.query \[p.88\]](#).

---

<b>receiverIndex</b>	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 \[p.98\]](#), [feedbackReceiver.reconfigure \[p.100\]](#), [feedbackReceiver.remove \[p.101\]](#).

---

<b>receivers</b>	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 \[p.99\]](#).

---

**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 \[p.98\]](#), [feedbackReceiver.query \[p.99\]](#), [feedbackReceiver.reconfigure \[p.100\]](#),

---

**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 <b>Settings &gt; Conference</b> 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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#).

---

**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 <b>redial</b> is <b>any</b>). 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 <b>ad_hoc</b>.</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 <b>redialLimit</b> inherits the value from the MCU-wide setting, as configured on the <b>Settings &gt; Conference</b> page of the web interface, when the participant joins.</p> <p><b>default</b> is therefore only possible for the configured state of the participant.</p> <p>This value is the default for participants of type <b>by_address</b>.</p>

**redialLimit** is used in: [participant.add \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#).

<b>registeredAddress</b>	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 \[p.104\]](#).

<b>registerWithGatekeeper</b> ( <i>template</i> )	string	Defines whether or not the conferences based on this template register their <b>numericIds</b> with the H.323 gatekeeper. One of <b>true</b> , <b>false</b> , or <b>default</b> (inherit this setting from the parent template).
---	--------	--

**registerWithGatekeeper** (*template*) is used in: [template.modify \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

<b>registerWithGatekeeper</b>	boolean	Defines whether or not this conference registers its <b>numericId</b> with the H.323 gatekeeper.
-------------------------------	---------	--

**registerWithGatekeeper** is used in: [conference.create \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#),

<b>registerWithSIPRegistrar</b> ( <i>template</i> )	string	Defines whether conferences based on this template register with the SIP registrar. One of <b>true</b> , <b>false</b> , or <b>default</b> (inherit this setting from the parent template).
---	--------	--

**registerWithSIPRegistrar** (*template*) is used in: [template.modify \[p.177\]](#), [template.enumerate \[p.173\]](#),

[template.create \[p.168\]](#), [template.status \[p.181\]](#).

---

**registerWithSIPRegistrar**                      boolean      Defines whether or not this conference registers its **numericId** with the SIP registrar.

**registerWithSIPRegistrar** is used in: [conference.create \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#),

---

**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 \[p.165\]](#).

---

**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 \[p.163\]](#), [sip.query \[p.165\]](#).

---

**registrarType**                                      string (10)    The type of SIP registrar. Either **normal** or **lcs**.

**registrarType** is used in: [sip.modify \[p.163\]](#), [sip.query \[p.165\]](#).

---

**registrarUsage**                                    boolean        Defines if the MCU should register to a registrar. Sets (deprecated) **outboundConfiguration**.

**registrarUsage** is used in: [sip.modify \[p.163\]](#), [sip.query \[p.165\]](#).

---

**registrationPrefix**                              string        A string of digits that serves as the device's registration prefix.

**registrationPrefix** is used in: [gatekeeper.modify \[p.102\]](#), [gatekeeper.query \[p.104\]](#).

---

**registrationStatus**                              string        The SIP registration status. One of **registering**, **registered**, **unregistered**, or **unknown**.

**registrationStatus** is used in: [sip.query \[p.165\]](#).

---

**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
<b>terminalGateway</b>	Corresponds to <i>Terminal / gateway</i> on the web UI.
<b>gateway</b>	Corresponds to <i>Gateway</i> on the web UI.
<b>gatewayCisco</b>	Corresponds to <i>Gateway (Cisco GK compatible)</i> .
<b>mcuStandard</b>	Corresponds to <i>MCU (Standard)</i> .
<b>mcuCompatible</b>	Corresponds to <i>MCU (Compatible)</i> .

**registrationType** is used in: [gatekeeper.modify \[p.102\]](#), [gatekeeper.query \[p.104\]](#).

---

**remoteLinkType** string One of **slave**, **conference**, **autoAttendant**, **recording**, or **playback**.

**remoteLinkType** is used in: [participant.enumerate \[p.116\]](#), [participant.status \[p.143\]](#).

---

**repairFrames** integer The number of ClearPath repair frames sent/received for video. (Sent and received media).

**repairFrames** is used in: [participant.statistics \[p.137\]](#).

---

**repetition** string Defines the repetition frequency of a scheduled conference. One of **none**, **daily**, **weekly**, **everyTwoWeeks**, or **monthly**.

Value	Description
none	The conference does not repeat.
daily	The conference repeats every day at the given <b>startTime</b> .
weekly	The conference repeats at least once per week, at the given <b>startTime</b> on the given <b>weekDays</b> .
everyTwoWeeks	The conference repeats at least once every two weeks, at the given <b>startTime</b> on the given <b>weekDays</b> .
monthly	The conference repeats once a month, at the given <b>startTime</b> on a given <b>weekDay</b> in the given week of the month ( <b>whichWeek</b> ).

**repetition** is used in: [conference.create \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

---

**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 \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

---

**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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#), [template.modify \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

---

**reservedVideoPorts** integer The number of video ports to reserve for a conference if in port reservation mode.

**reservedVideoPorts** is used in: [conference.create \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify](#)

[\[p.59\]conference.status](#) [\[p.68\]](#), [template.modify](#) [\[p.177\]](#), [template.enumerate](#) [\[p.173\]](#), [template.create](#) [\[p.168\]](#), [template.status](#) [\[p.181\]](#).

---

**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](#) [\[p.177\]](#), [template.enumerate](#) [\[p.173\]](#), [template.create](#) [\[p.168\]](#), [template.status](#) [\[p.181\]](#).

---

**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](#) [\[p.104\]](#).

---

**restartTime**                              dateTime.    The system's date and time when it started.  
iso8601

**restartTime** is used in: [device.query](#) [\[p.91\]](#).

---

**routeId**                                    integer       A number that identifies a route. The device assigns a number to each manually configured route.

**routeId** is used in: [route.delete](#) [\[p.156\]](#), [route.enumerate](#) [\[p.157\]](#).

---

**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](#) [\[p.85\]](#).

---

**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](#) [\[p.85\]](#).

---

**rtcpFecRecoveredReported**            integer       The number of lost packets reported as recovered by ClearPath FEC at the far end for video and audio control structs.

**rtcpFecRecoveredReported** is used in: [participant.statistics](#) [\[p.137\]](#).

---

**rtcpLipSyncDelay**                        integer       The reported delay between the incoming audio and video streams from this endpoint.

**rtcpLipSyncDelay** is used in: [participant.statistics](#) [\[p.137\]](#).

---

**rtcpOtherReports**                        integer       Count of the RTCP reports seen by the MCU that are neither sender nor receiver reports.

**rtcpOtherReports** is used in: [participant.statistics](#) [\[p.137\]](#).

---

**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 \[p.137\]](#).

---

<b>rtcpPacketsSent</b>	integer	Count of RTCP packets sent by the MCU to this endpoint.
------------------------	---------	---

**rtcpPacketsSent** is used in: [participant.statistics \[p.137\]](#).

---

<b>rtcpReceiveAddress</b>	string	Address of the RTCP receiver.
---------------------------	--------	-------------------------------

**rtcpReceiveAddress** is used in: [participant.statistics \[p.137\]](#).

---

<b>rtcpReceivePort</b>	integer	Port number used by the receiver to accept RTCP messages.
------------------------	---------	---

**rtcpReceivePort** is used in: [participant.statistics \[p.137\]](#).

---

<b>rtcpReceiverReports</b>	integer	Count of the RTCP receiver reports seen by the MCU.
----------------------------	---------	---

**rtcpReceiverReports** is used in: [participant.statistics \[p.137\]](#).

---

<b>rtcpSenderReports</b>	integer	Count of the RTCP sender reports seen by the MCU.
--------------------------	---------	---

**rtcpSenderReports** is used in: [participant.statistics \[p.137\]](#).

---

<b>rtcpTransmitAddress</b>	string	The IP address and port to which the MCU is sending RTCP packets about this stream.
----------------------------	--------	---

**rtcpTransmitAddress** is used in: [participant.statistics \[p.137\]](#).

---

<b>rtcpTransmitPort</b>	integer	Port number used for transmitting RTCP messages to the endpoint. Absent if <b>rtcpTransmitAddress</b> is unspecified.
-------------------------	---------	---

**rtcpTransmitPort** is used in: [participant.statistics \[p.137\]](#).

# 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](#)

<b>scheduled</b>	boolean	<b>true</b> if the conference is a scheduled conference (regardless of whether or not it is completed).
<b>scheduled</b> is used in: <a href="#">conference.enumerate [p.48]</a> .		
<b>scheduledConferenceIDRegistration</b>	string (8)	Defines whether or not ID registration is enabled for scheduled conferences. Either <b>enabled</b> or <b>disabled</b> . Corresponds to the <b>ID registration for scheduled conferences</b> option on the web interface.
<b>scheduledConferenceIDRegistration</b> is used in: <a href="#">gatekeeper.modify [p.102]</a> , <a href="#">gatekeeper.query [p.104]</a> .		
<b>selectedBitRate</b>	integer	The selected bit rate for the media stream. Applies to sent and received video and content streams. Deprecates <b>videoRxSelectedBitRate</b> , <b>contentRSelectedBitRate</b> , <b>videoTxSelectedBitRate</b> , and <b>contentTxSelectedBitRate</b> .
<b>selectedBitRate</b> is used in: <a href="#">participant.statistics [p.137]</a> .		
<b>sendResourceAvailabilityIndications</b>	boolean	Defines whether or not the MCU will send resource availability indications.
<b>sendResourceAvailabilityIndications</b> is used in: <a href="#">gatekeeper.modify [p.102]</a> , <a href="#">gatekeeper.query [p.104]</a> .		
<b>serial</b>	string	The serial number of this device or 'unknown'.
<b>serial</b> is used in: <a href="#">device.query [p.91]</a> .		
<b>services</b>	array	An array whose members represent the services provided on the particular port and protocol.
<b>services</b> is used in: <a href="#">services.modify [p.160]</a> , <a href="#">services.query [p.162]</a> .		
<b>setAllParticipantsToCustomLayout</b>	boolean	<b>true</b> sets all participants to immediately see the conference custom layout. If <b>false</b> nothing happens. Only valid if <b>customLayoutEnabled</b> is <b>true</b>
<b>setAllParticipantsToCustomLayout</b> is used in: <a href="#">conference.modify [p.59]</a> .		
(missing or bad snippet)		
<b>setting</b>	boolean	Defines whether or not this feature is intended to be enabled, irrespective of whether it is actually enabled or requires a feature key.
<b>setting</b> is used in: <a href="#">conferenceme.modify [p.77]</a> , <a href="#">conferenceme.query [p.78]</a> , <a href="#">device.encryption.modify [p.81]</a> , <a href="#">device.encryption.query [p.82]</a> , <a href="#">services.modify [p.160]</a> , <a href="#">services.query [p.162]</a> , <a href="#">streaming.modify [p.166]</a> , <a href="#">streaming.query [p.167]</a> .		
<b>shutdownOnly</b>	boolean	If <b>true</b> , the device will shut down when it receives <b>device.restart</b> and will not restart. Defaults to <b>false</b> .

**shutdownOnly** is used in: [device.restart \[p.93\]](#).

---

<b>shutdownStatus</b>	string	Indicates the status of a shutdown operation. One of <b>shutdown</b> , <b>shutdownInProgress</b> , or <b>notShutdown</b> .
-----------------------	--------	--

**shutdownStatus** is used in: [device.query \[p.91\]](#).

---

<b>sipMediaEncryption</b>	string	Defines whether SIP media is encrypted and, if so, for which transport protocols. One of <b>disabled</b> , <b>allTransports</b> or <b>tlsOnly</b> .
---------------------------	--------	---

**sipMediaEncryption** is used in: [device.encryption.modify \[p.81\]](#), [device.encryption.query \[p.82\]](#).

---

<b>siren14</b>	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 \[p.24\]](#).

---

<b>softwareVersion</b>	string	The version number of the software running on the device.
------------------------	--------	---

**softwareVersion** is used in: [device.query \[p.91\]](#).

---

<b>sourceIdentifier</b>	string	The originating device uses this parameter to identify itself to the listening receiver/s.
-------------------------	--------	--

**sourceIdentifier** is used in: [feedbackReceiver.configure \[p.98\]](#), [feedbackReceiver.query \[p.99\]](#), [feedbackReceiver.reconfigure \[p.100\]](#),

---

<b>speed</b>	integer	Speed of the connection on this Ethernet interface. One of 10, 100 or 1000, in Mbps.
--------------	---------	--

**speed** is used in: [device.network.modify \[p.86\]](#), [device.network.query \[p.88\]](#).

---

<b>startIndex</b>	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 <b>startIndex</b> with the index provided gives the number of dropped records.
-------------------	---------	--

**startIndex** is used in: [cdrlog.enumerate \[p.38\]](#).

---

<b>startLocked (template)</b>	string	Defines whether conferences based on this template should be locked when they start. One of <b>true</b> , <b>false</b> , or <b>default</b> (inherit this setting from the parent template).
-------------------------------	--------	---

**startLocked (template)** is used in: [template.modify \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

---

<b>startLocked</b>	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 \[p.41\]](#), [conference.modify \[p.59\]](#).

---

<b>startTime</b>	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 \[p.41\]](#),

[conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#),

---

**status (success)** string **Operation successful**

**status (success)** is used in: [conference.metadata.modify \[p.57\]](#), [feedbackReceiver.configure \[p.98\]](#), [route.add \[p.155\]](#), [template.create \[p.168\]](#), [template.modify \[p.177\]](#).

---

**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 \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

---

**subnetMask** string The IPv4 subnet mask in dotted quad format.

**subnetMask** is used in: [device.network.query \[p.88\]](#).

---

**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 \[p.98\]](#), [feedbackReceiver.reconfigure \[p.100\]](#).

---

**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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

---

**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 \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#), [participant.add \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

**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 \[p.177\]](#), [template.enumerate \[p.173\]](#), [template.create \[p.168\]](#), [template.status \[p.181\]](#).

# 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 \[p.85\]](#).

---

**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 \[p.85\]](#).

---

**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 \[p.41\]](#), [template.create \[p.168\]](#), [template.delete \[p.172\]](#), [template.enumerate \[p.173\]](#), [template.status \[p.181\]](#).

---

**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 \[p.41\]](#), [template.create \[p.168\]](#), [template.delete \[p.172\]](#), [template.modify \[p.177\]](#), [template.status \[p.181\]](#).

---

**templates** array of structs Each array element is a struct that contains the parameters that define a template.

`templates` is used in: [template.create \[p.168\]](#), [template.enumerate \[p.173\]](#).

---

**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 \[p.137\]](#).

---

**terminationDate** dateTime. iso8601 Required if **terminationType** is **endOnGivenDate**. This is the date when conference repetition will cease.

**terminationDate** is used in: [conference.create \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

---

**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 <b>numberOfRepeats</b> .
endOnGivenDate	The conference will repeat, according to the given <b>repetition</b> and relevant parameters, until the given <b>terminationDate</b> .

**terminationType** is used in: [conference.create \[p.41\]](#), [conference.enumerate \[p.48\]](#), [conference.modify \[p.59\]](#), [conference.status \[p.68\]](#).

---

**time (CDR log)** dateTime. iso8601 The date and time when the event was logged, for example 20110119T13:52:42.

**time (CDR log)** is used in: [cdrlog.enumerate \[p.38\]](#).

---

**time (restart log)** dateTime. iso8601 The date and time when the device restarted. For example, 20110119T13:52:42 is in the format *yyyymmddThh:mm:ss*.

**time** is used in: [device.restartlog.query \[p.94\]](#).

---

**toOverride** string (63) This value overrides the SIP To-URI.

**toOverride** is used in: [participant.add \[p.107\]](#),

---

**totalAudioOnlyPorts** integer The total number of additional audio-only ports on the device.

**totalAudioOnlyPorts** is used in: [device.query \[p.91\]](#).

---

**totalPlaybackPorts** integer The number of ports this device uses for playback.

**totalPlaybackPorts** is used in: [device.query \[p.91\]](#).

---

**totalRecordingPorts** integer The number of ports this device uses for recording.

**totalRecordingPorts** is used in: [device.query \[p.91\]](#).

---

**totalStreamingAndContentPorts** integer The total number of streaming and content ports on the MCU. Only provided if non-zero.

**totalStreamingAndContentPorts** is used in: [device.query \[p.91\]](#).

---

**totalVideoPorts** integer The total number of video ports on the device.

**totalVideoPorts** is used in: [device.query \[p.91\]](#).

---

**transmitErrors** integer The count of transmission errors on this Ethernet interface.

**transmitErrors** is used in: [device.network.query \[p.88\]](#).

---

**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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

---

**type (event)** string The name of the event type.

**type (event)** is used in: [cdrlog.enumerate \[p.38\]](#).

---

**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 \[p.63\]](#), [conference.paneplacement.query \[p.65\]](#).

---

**type (route)** string The type of route. One of **automatic**, **configuredByGateway** or **configuredByPort**.

**type (route)** is used in: [route.enumerate \[p.157\]](#).

---

**type (service)** string The type of service. Either **tcp** or **udp**.

**type (service)** is used in: [services.modify \[p.160\]](#), [services.query \[p.162\]](#).

---

**type (videoports)** string One of **nhd**, **sd**, **hd**, **hdPlus** or **fullhd**

**type (videoports)** is used in: [device.query \[p.91\]](#).

# 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: <a href="#">conference.create [p.41]</a> , <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.modify [p.59]</a> , <a href="#">conference.status [p.68]</a> .
<code>unicastViewers</code>	integer	The count of unicast streaming viewers.  <code>unicastViewers</code> is used in: <a href="#">conference.streaming.query [p.74]</a> ,
<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: <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.status [p.68]</a> .
<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: <a href="#">device.status [p.95]</a> .
<code>useDefaultMotionSharpness</code>	boolean	<code>true</code> means this endpoint will use box-wide default motion sharpness settings.  <code>useDefaultMotionSharpness</code> is used in: <a href="#">addressBookEntry.enumerate [p.24]</a> , <a href="#">gateway.enumerate [p.106]</a> .
<code>useDefaultVideoTransmitResolutions</code>	boolean	<code>true</code> means this endpoint will use box-wide default video transmit resolutions.  <code>useDefaultVideoTransmitResolutions</code> is used in: <a href="#">addressBookEntry.enumerate [p.24]</a> .
<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: <a href="#">sip.modify [p.163]</a> , <a href="#">sip.query [p.165]</a> .
<code>useMaximumPortsFromParent</code>	boolean	Cannot be set to true for template 0  <code>useMaximumPortsFromParent</code> is used in: <a href="#">template.enumerate [p.173]</a> , <a href="#">template.create [p.168]</a> , <a href="#">template.modify [p.177]</a> .
<code>usePassword</code>	boolean	Indicates whether or not the device uses its configured password for gatekeeper registration.  <code>usePassword</code> is used in: <a href="#">gatekeeper.modify [p.102]</a> , <a href="#">gatekeeper.query [p.104]</a> .
<code>useReservedPortsFromParent</code>	boolean	Cannot be set to true for template 0  <code>useReservedPortsFromParent</code> is used in: <a href="#">template.modify [p.177]</a> , <a href="#">template.enumerate [p.173]</a> , <a href="#">template.create [p.168]</a> , <a href="#">template.status [p.181]</a> .
<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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#),

---

<b>useWebService</b>	boolean	<b>true</b> 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 \[p.77\]](#), [conferenceme.query \[p.78\]](#).

---

<b>utcOffsetHours</b>	integer	Number between -12 and +14 (inclusive) that, together with <b>utcOffsetMinutes</b> , defines the UTC offset of the device's clock.
-----------------------	---------	--

**utcOffsetHours** is used in: [device.time.modify \[p.96\]](#), [device.time.query \[p.97\]](#).

---

<b>utcOffsetMinutes</b>	integer	Number between 0 and 59 (inclusive) that, together with <b>utcOffsetHours</b> , defines the UTC offset of the device's clock.
-------------------------	---------	---

**utcOffsetMinutes** is used in: [device.time.modify \[p.96\]](#), [device.time.query \[p.97\]](#).

# 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 \[p.130\]](#).

---

**videoCodec** string The video codec for this streaming connection. Either **RTSP** or **MMS**.

**videoCodec** is used in: [streaming.query \[p.167\]](#).

---

**videoControl** boolean Defaults to **false**. Set **true** to return **videoControl** statistics.

**videoControl** is used in: [participant.statistics \[p.137\]](#).

---

**videoLoad** integer A percentage value representing the proportion of the device's video processing capacity that is currently in use. (Not returned on the MCU 5300 Series.)

**videoLoad** is used in: [device.health.query \[p.85\]](#).

---

**videoMedia** boolean Defaults to **false**. Set **true** to return **videoMedia** statistics.

**videoMedia** is used in: [participant.statistics \[p.137\]](#).

---

**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 \[p.91\]](#).

---

**videoRTCPOther** integer As for the audio equivalents.

**videoRTCPOther** is used in: [conference.streaming.query \[p.74\]](#).

---

**videoRTCPPacketsSent** integer As for the audio equivalents.

**videoRTCPPacketsSent** is used in: [conference.streaming.query \[p.74\]](#).

---

**videoRTCPReceiverReports** integer As for the audio equivalents.

**videoRTCPReceiverReports** is used in: [conference.streaming.query \[p.74\]](#).

---

**videoRTCPSenderReports** integer As for the audio equivalents.

**videoRTCPSenderReports** is used in: [conference.streaming.query \[p.74\]](#).

---

**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 \[p.112\]](#).

---

**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 \[p.112\]](#).

---

**videoRxChannelBitRate** integer The negotiated available bandwidth for the video stream coming from the endpoint.

**videoRxChannelBitRate** is used in: [participant.diagnostics \[p.112\]](#).

---

**videoRxCodec** string The codec used on the received video.

**videoRxCodec** is used in: [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**videoRxFrameRate** integer The frame rate of the received video (frames per second).

**videoRxFrameRate** is used in: [participant.diagnostics \[p.112\]](#).

---

**videoRxFramesReceived** integer The number of video frames received from this endpoint.

**videoRxFramesReceived** is used in: [participant.diagnostics \[p.112\]](#).

---

**videoRxFramesReceivedWithErrors** string The number of video frames received from this endpoint that were not successfully decoded.

**videoRxFramesReceivedWithErrors** is used in: [participant.diagnostics \[p.112\]](#).

---

**videoRxHeight** integer Height in pixels of the received video.

**videoRxHeight** is used in: [participant.diagnostics \[p.112\]](#).

---

**videoRxInterlaced** boolean **true** if the MCU is receiving interlaced video from this endpoint.

**videoRxInterlaced** is used in: [participant.diagnostics \[p.112\]](#).

---

**videoRxJitter** integer Represents the variability of the timing of received video packets.

**videoRxJitter** is used in: [participant.diagnostics \[p.112\]](#).

---

**videoRxLost** integer Count of video packets lost en route to the MCU from this endpoint.

**videoRxLost** is used in: [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**videoRxMaxResolution** string The maximum resolution of the received video. One of **cif**, **4cif**, or **max**.

Value	Description
cif	This endpoint sends <b>cif</b> or lower resolution to the MCU.
4cif	This endpoint sends <b>4cif</b> or lower resolution to the MCU.
max	Send the maximum resolution that both sides can support.

**videoRxMaxResolution** is used in: [participant.add \[p.107\]](#).

---

**videoRxMuted** boolean **true** means that video from this participant will not be seen by other conference participants.

**videoRxMuted** is used in: [participant.add \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**videoRxReceived** integer Count of video packets received from this endpoint.

**videoRxReceived** is used in: [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

**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 \[p.112\]](#).

---

**videoRxWidth** integer Width in pixels of the received video.

**videoRxWidth** is used in: [participant.diagnostics \[p.112\]](#).

---

**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 \[p.74\]](#).

---

**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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#).

---

**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 \[p.112\]](#).

`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 \[p.112\]](#).

`videoTxChannelBitRate` integer The negotiated available bandwidth for the video stream going to the endpoint.

`videoTxChannelBitRate` is used in: [participant.diagnostics \[p.112\]](#).

`videoTxCodec` string The codec used on the transmitted video.

`videoTxCodec` is used in: [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

`videoTxFrameRate` integer Frame rate of the transmitted video (frames per second).

`videoTxFrameRate` is used in: [participant.diagnostics \[p.112\]](#).

`videoTxHeight` integer Height in pixels of the transmitted video.

`videoTxHeight` is used in: [participant.diagnostics \[p.112\]](#).

`videoTxInterlaced` boolean `true` if the MCU is sending interlaced video to this endpoint.

`videoTxInterlaced` is used in: [participant.diagnostics \[p.112\]](#).

`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 \[p.107\]](#).

---

`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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.modify \[p.131\]](#).

---

`videoTxReportedLost` integer The count of video packets reported lost by the far end.

`videoTxReportedLost` is used in: [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

`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 \[p.112\]](#).

---

`videoTxSent` integer Count of the video packets sent to the endpoint.

`videoTxSent` is used in: [participant.diagnostics \[p.112\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

`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 \[p.107\]](#), [participant.enumerate \[p.116\]](#), [participant.enumerate \(deprecated\) \[p.125\]](#), [participant.modify \[p.131\]](#), [participant.status \[p.143\]](#), [participant.status \(deprecated\) \[p.151\]](#).

---

`videoTxWidth` integer Width in pixels of the transmitted video.

`videoTxWidth` is used in: [participant.diagnostics \[p.112\]](#).

---

`voltagesStatus` string `ok`, `outOfSpec` (the voltage is currently outside the normal range), or `critical`.

`voltagesStatus` is used in: [device.health.query \[p.85\]](#).

---

`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 \[p.85\]](#).

# 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](#)

<b>webAppletBandwidth</b>	integer	The bandwidth of the content stream sent to streaming viewers.
<b>webAppletBandwidth</b> is used in: <a href="#">device.content.modify [p.79]</a> , <a href="#">device.content.query [p.80]</a> .		
<b>weekDay</b>	string	Must be present if <b>repetition</b> is <b>monthly</b> . One of <b>monday</b> , <b>tuesday</b> , <b>wednesday</b> , <b>thursday</b> , <b>friday</b> , <b>saturday</b> or <b>sunday</b> . Note that if <b>repetition</b> is not <b>weekly</b> or <b>everyTwoWeeks</b> , the <b>weekDays</b> parameter should be used.
<b>weekDay</b> is used in: <a href="#">conference.create [p.41]</a> , <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.modify [p.59]</a> , <a href="#">conference.status [p.68]</a> .		
<b>weekDays</b>	string	Required if <b>repetition</b> is <b>weekly</b> or <b>everyTwoWeeks</b> . The parameter accepts a comma separated string of weekday names, e.g. <b>monday</b> , <b>wednesday</b> , <b>friday</b> .
<b>weekDays</b> is used in: <a href="#">conference.create [p.41]</a> , <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.modify [p.59]</a> , <a href="#">conference.status [p.68]</a> .		
<b>whichWeek</b>	string	Required if <b>repetition</b> is <b>monthly</b> . Defines which week the repeating conference will fall in; one of <b>first</b> , <b>second</b> , <b>third</b> , <b>fourth</b> , or <b>last</b> .
<b>whichWeek</b> is used in: <a href="#">conference.create [p.41]</a> , <a href="#">conference.enumerate [p.48]</a> , <a href="#">conference.modify [p.59]</a> , <a href="#">conference.status [p.68]</a> .		
<b>width</b>	integer	The maximum width and height of this stream. Only present for defined video streams
<b>width</b> is used in: <a href="#">conference.streaming.query [p.74]</a> , <a href="#">participant.statistics [p.137]</a> .		
<b>wmpProtocol</b>	string	Describes the behavior of the wmpProtocol when streaming to the endpoint. One of <b>auto</b> , <b>mmsOverUdp</b> , <b>mmsOverTcp</b> , or <b>http</b> .
<b>wmpProtocol</b> is used in: <a href="#">streaming.modify [p.166]</a> , <a href="#">streaming.query [p.167]</a> .		

# API Change history

---

This section details the changes in each version of the API.

Version 2.11 changes .....	288
Version 2.10 changes .....	290
Version 2.9 changes .....	292
Version 2.8 changes .....	300

## Version 2.11 changes

Category	Name	Change
Command	<a href="#">device.features.add [p.83]</a>	Introduced
Command	<a href="#">device.features.remove [p.84]</a>	Introduced
Command	<a href="#">device.query [p.91]</a>	Parameters added
Command	<a href="#">device.network.modify [p.86]</a>	Modified
Command	<a href="#">sip.modify [p.163]</a>	Parameters added
Command	<a href="#">sip.query [p.165]</a>	Parameters added
Command	<a href="#">callHome.configure [p.35]</a>	Introduced
Command	<a href="#">callHome.query [p.36]</a>	Introduced
Command	<a href="#">participant.statistics [p.137]</a>	Parameters added
Command	<a href="#">conference.modify [p.59]</a>	Parameters added and modified
Fault	<b>60: invalid manual IP settings</b> <a href="#">Fault codes [p.188]</a>	Introduced
Fault	<b>61: cannot remove this feature key</b> <a href="#">Fault codes [p.188]</a>	Introduced
Fault	<b>62: participant not currently active</b> <a href="#">Fault codes [p.188]</a>	Introduced
Parameter	<a href="#">key [p.245]</a>	Introduced
Parameter	<a href="#">activatedLicenses [p.202]</a>	Introduced
Parameter	<a href="#">mediaResourceRestarts [p.252]</a>	Introduced
Parameter	<a href="#">ipv4Address [p.242]</a>	Modified
Parameter	<a href="#">defaultIpv4Gateway [p.225]</a>	Modified
Parameter	<a href="#">outboundConfiguration [p.257]</a>	Introduced
Parameter	<a href="#">outboundAddress [p.257]</a>	Introduced
Parameter	<a href="#">outboundDomain [p.257]</a>	Introduced
Parameter	<a href="#">registrarUsage [p.268]</a>	Deprecated
Parameter	<a href="#">configuredRegistrar [p.216]</a>	Deprecated
Parameter	<a href="#">configuredProxy [p.216]</a>	Deprecated
Parameter	<a href="#">mode [p.252]</a>	Introduced
Parameter	<a href="#">automatic [p.207]</a>	Introduced
Parameter	<a href="#">rtcpFecRecoveredReported [p.270]</a>	Introduced
Parameter	<a href="#">repairFrames [p.269]</a>	Introduced
Parameter	<a href="#">customLayoutEnabled [p.224]</a>	Modified

Category	Name	Change
Parameter	<a href="#">setAllParticipantsToCustomLayout [p.272]</a>	Introduced
Parameter	<a href="#">displayNameEx [p.118]</a>	Introduced
Parameter	<a href="#">toOverride [p.277]</a>	Introduced
Parameter	<a href="#">durationSeconds [p.228]</a>	Modified
Parameter	<a href="#">contentPassthroughLimit [p.218]</a>	Omission
Parameter	<a href="#">contentPassthroughLimit (template) [p.218]</a>	Omission
Related information	<a href="#">Conference layouts [p.193]</a>	Modified
Related information	<a href="#">HTTP keep-alives [p.192]</a>	Modified

## Version 2.10 changes

Category	Name	Change
Command	<a href="#">addressBookEntry.enumerate [p.24]</a>	Parameters added
Command	<a href="#">conference.create [p.41]</a>	Parameters added
Command	<a href="#">conference.enumerate [p.48]</a>	Parameters added
Command	<a href="#">conference.metadata.modify [p.57]</a>	Parameters added
Command	<a href="#">conference.modify [p.59]</a>	Parameters added
Command	<a href="#">conference.paneplacement.query [p.65]</a>	Modified
Command	<a href="#">conference.status [p.68]</a>	Parameters added
Command	<a href="#">device.query [p.91]</a>	Parameters added
Command	<a href="#">device.status [p.95]</a>	Introduced
Command	<a href="#">participant.add [p.107]</a>	Parameters added
Command	<a href="#">participant.enumerate [p.116]</a>	Parameters added
Command	<a href="#">participant.enumerate (deprecated) [p.125]</a>	Parameters added
Command	<a href="#">participant.modify [p.131]</a>	Parameters added
Command	<a href="#">participant.status [p.143]</a>	Parameters added
Command	<a href="#">participant.status (deprecated) [p.151]</a>	Parameters added
Command	<a href="#">template.create [p.168]</a>	Parameters added
Command	<a href="#">template.delete [p.172]</a>	Parameters added
Command	<a href="#">template.enumerate [p.173]</a>	Parameters added
Command	<a href="#">template.modify [p.177]</a>	Parameters added
Command	<a href="#">template.status [p.181]</a>	Parameters added
Filter	<a href="#">connecting [p.216]</a>	Modified
Filter	<a href="#">dormant [p.227]</a>	Introduced
Parameter	<a href="#">audioRxGainMode [p.205]</a>	Modified
Parameter	<a href="#">audioTxMuted [p.205]</a>	Modified
Parameter	<a href="#">callState [p.211]</a>	Deprecated
Parameter	<a href="#">callStateEx [p.211]</a>	Introduced
Parameter	<a href="#">contentContribution (template) [p.217]</a>	Modified
Parameter	<a href="#">contentReceive [p.218]</a>	Introduced
Parameter	<a href="#">deferConnection [p.225]</a>	Modified
Parameter	<a href="#">dtmfSequence [p.228]</a>	Modified
Parameter	<a href="#">joinAGC [p.244]</a>	Introduced
Parameter	<a href="#">joinAGC (template) [p.244]</a>	Introduced
Parameter	<a href="#">maxConferenceSize [p.250]</a>	Introduced

Category	Name	Change
Parameter	<a href="#">redial [p.266]</a>	Introduced
Parameter	<a href="#">redialLimit [p.266]</a>	Introduced
Parameter	<a href="#">suppressAudioDuringDTMF [p.274]</a>	Introduced
Parameter	<a href="#">unnamed (device.status) [p.279]</a>	Introduced
Parameter	<a href="#">videoToUse [p.283]</a>	Introduced
Parameter	<a href="#">videoTxMuted [p.285]</a>	Introduced

## Version 2.9 changes

Category	Name	Change
Command	<a href="#">addressBookEntry.enumerate [p.24]</a>	Parameters added
Command	<a href="#">cdrlog.enumerate [p.38]</a>	Parameters added
Command	<a href="#">conference.create [p.41]</a>	Parameters replaced
Command	<a href="#">conference.enumerate [p.48]</a>	Parameters added, replaced
Command	<a href="#">conference.modify [p.59]</a>	Parameters replaced
Command	<a href="#">conference.status [p.68]</a>	Parameters added, replaced
Command	<a href="#">conferenceme.modify [p.77]</a>	Introduced
Command	<a href="#">conferenceme.query [p.78]</a>	Parameters added
Command	<a href="#">device.content.modify [p.79]</a>	Introduced
Command	<a href="#">device.content.query [p.80]</a>	Introduced
Command	<a href="#">device.encryption.modify [p.81]</a>	Introduced
Command	<a href="#">device.encryption.query [p.82]</a>	Introduced
Command	<a href="#">device.network.modify [p.86]</a>	Introduced
Command	<a href="#">device.query [p.91]</a>	Parameters added
Command	<a href="#">device.restart [p.93]</a>	Introduced
Command	<a href="#">device.time.modify [p.96]</a>	Introduced
Command	<a href="#">device.time.query [p.97]</a>	Introduced
Command	<a href="#">feedbackReceiver.reconfigure [p.100]</a>	Introduced
Command	<a href="#">feedbackReceiver.remove [p.101]</a>	Introduced
Command	<a href="#">gatekeeper.modify [p.102]</a>	Introduced
Command	<a href="#">gatekeeper.query [p.104]</a>	Parameters added
Command	<a href="#">participant.add [p.107]</a>	Parameters added
Command	<a href="#">participant.diagnostics [p.112]</a>	Deprecated
Command	<a href="#">participant.statistics [p.137]</a>	Introduced
Command	<a href="#">route.add [p.155]</a>	Introduced
Command	<a href="#">route.delete [p.156]</a>	Introduced
Command	<a href="#">route.enumerate [p.157]</a>	Introduced
Command	<a href="#">route.preferences.modify [p.158]</a>	Introduced
Command	<a href="#">route.preferences.query [p.159]</a>	Introduced
Command	<a href="#">services.modify [p.160]</a>	Introduced
Command	<a href="#">services.query [p.162]</a>	Parameters added

Category	Name	Change
Command	<a href="#">sip.modify [p.163]</a>	Introduced
Command	<a href="#">streaming.modify [p.166]</a>	Introduced
Command	<a href="#">streaming.query [p.167]</a>	Introduced
Command	<a href="#">template.create [p.168]</a>	Parameters replaced
Command	<a href="#">template.enumerate [p.173]</a>	Parameters replaced
Command	<a href="#">template.modify [p.177]</a>	Parameters replaced
Command	<a href="#">template.status [p.181]</a>	Parameters replaced
Fault	36: XML_RPC_FAULT_REQUIRED_VALUE_MISSING <a href="#">Fault codes [p.188]</a>	Introduced
Fault	42: XML_RPC_FAULT_PORT_CONFLICT <a href="#">Fault codes [p.188]</a>	Introduced
Fault	43: XML_RPC_FAULT_ROUTE_ALIASES_EXISTING <a href="#">Fault codes [p.188]</a>	Introduced
Fault	44: XML_RPC_FAULT_ROUTE_REJECTED <a href="#">Fault codes [p.188]</a>	Introduced
Fault	45: XML_RPC_FAULT_TOO_MANY_ROUTES <a href="#">Fault codes [p.188]</a>	Introduced
Fault	46: XML_RPC_FAULT_NO_SUCH_ROUTE <a href="#">Fault codes [p.188]</a>	Introduced
Fault	48: XML_RPC_FAULT_IP_ADDRESS_OVERFLOW_MASK <a href="#">Fault codes [p.188]</a>	Introduced
Fault	49: XML_RPC_FAULT_DISABLE_ACTIVE_INTERFACE <a href="#">Fault codes [p.188]</a>	Introduced
Fault	104: XML_RPC_FAULT_MISMATCHED_PARAMETERS <a href="#">Fault codes [p.188]</a>	Introduced
Fault	105: XML_RPC_FAULT_REQUEST_TOO_LARGE <a href="#">Fault codes [p.188]</a>	Introduced
Feedback event	sipChanged <a href="#">Feedback events [p.18]</a>	Introduced
Feedback event	h323Changed <a href="#">Feedback events [p.18]</a>	Introduced
Feedback event	floorChanged <a href="#">Feedback events [p.18]</a>	Introduced
Feedback event	chairChanged <a href="#">Feedback events [p.18]</a>	Introduced
Feedback event	encryptionChanged <a href="#">Feedback events [p.18]</a>	Introduced
Feedback event	contentChanged <a href="#">Feedback events [p.18]</a>	Introduced
Feedback event	streamingChanged <a href="#">Feedback events [p.18]</a>	Introduced
Feedback event	conferenceMeChanged <a href="#">Feedback events [p.18]</a>	Introduced
Feedback event	networkChanged <a href="#">Feedback events [p.18]</a>	Introduced
Feedback event	servicesChanged <a href="#">Feedback events [p.18]</a>	Introduced
Feedback event	routesChanged <a href="#">Feedback events [p.18]</a>	Introduced
Feedback event	deviceStatusChanged <a href="#">Feedback events [p.18]</a>	Introduced
Feedback event	rebooting <a href="#">Feedback events [p.18]</a>	Introduced

Category	Name	Change
Feedback event	timeChanged <a href="#">Feedback events [p.18]</a>	Introduced
Parameter	<a href="#">aac-Ic [p.201]</a>	Introduced
Parameter	<a href="#">aac-Id [p.201]</a>	Introduced
Parameter	<a href="#">active (route) [p.202]</a>	Introduced
Parameter	<a href="#">actualBitRate [p.203]</a>	Introduced
Parameter	<a href="#">addResponse [p.203]</a>	Introduced
Parameter	<a href="#">audioCodec [p.203]</a>	Introduced
Parameter	<a href="#">audioControl [p.204]</a>	Introduced
Parameter	<a href="#">audioMedia [p.204]</a>	Introduced
Parameter	<a href="#">audioRx [p.204]</a>	Introduced
Parameter	<a href="#">audioRxLost [p.205]</a>	Deprecated
Parameter	<a href="#">audioRxReceived [p.205]</a>	Deprecated
Parameter	<a href="#">audioTx [p.205]</a>	Introduced
Parameter	<a href="#">audioTxReportedLost [p.205]</a>	Deprecated
Parameter	<a href="#">bitRateLimitReason [p.210]</a>	Introduced
Parameter	<a href="#">cameraControl [p.212]</a>	Modified
Parameter	<a href="#">chairControl [p.213]</a>	Modified
Parameter	<a href="#">chairControl (template) [p.213]</a>	Modified
Parameter	<a href="#">chairParticipant [p.214]</a>	Introduced
Parameter	<a href="#">channelBitRate [p.214]</a>	Introduced
Parameter	<a href="#">codecBitRate [p.214]</a>	Introduced
Parameter	<a href="#">contentControl [p.217]</a>	Introduced
Parameter	<a href="#">contentEnabled [p.217]</a>	Introduced
Parameter	<a href="#">contentError [p.217]</a>	Introduced
Parameter	<a href="#">contentHandoverEnabled [p.217]</a>	Introduced
Parameter	<a href="#">contentInMainVideo [p.217]</a>	Introduced
Parameter	<a href="#">contentMarkupEnabled [p.217]</a>	Introduced
Parameter	<a href="#">contentMedia [p.217]</a>	Introduced
Parameter	<a href="#">contentRxSelectedBitRate [p.219]</a>	Deprecated
Parameter	<a href="#">contentRxActualBitRate [p.218]</a>	Deprecated
Parameter	<a href="#">contentRxBitRateLimitReason [p.218]</a>	Deprecated
Parameter	<a href="#">contentRxChannelBitRate [p.219]</a>	Deprecated
Parameter	<a href="#">contentRxCodec [p.219]</a>	Deprecated
Parameter	<a href="#">contentRxFrameRate [p.219]</a>	Deprecated

Category	Name	Change
Parameter	<a href="#">contentRxFramesReceived [p.219]</a>	Deprecated
Parameter	<a href="#">contentRxFramesReceivedWithErrors [p.219]</a>	Deprecated
Parameter	<a href="#">contentRxHeight [p.219]</a>	Deprecated
Parameter	<a href="#">contentRxJitter [p.219]</a>	Deprecated
Parameter	<a href="#">contentRxLost [p.219]</a>	Deprecated
Parameter	<a href="#">contentRxReceived [p.219]</a>	Deprecated
Parameter	<a href="#">contentRxType [p.219]</a>	Deprecated
Parameter	<a href="#">contentRxWidth [p.220]</a>	Deprecated
Parameter	<a href="#">contentStreamingSetting [p.220]</a>	Introduced
Parameter	<a href="#">contentStreamingStatus [p.220]</a>	Introduced
Parameter	<a href="#">contentTxActualBitRate [p.221]</a>	Deprecated
Parameter	<a href="#">contentTxBitRateLimitReason [p.221]</a>	Deprecated
Parameter	<a href="#">contentTxChannelBitRate [p.221]</a>	Deprecated
Parameter	<a href="#">contentTxCodec [p.221]</a>	Deprecated
Parameter	<a href="#">contentTxError [p.222]</a>	Deprecated
Parameter	<a href="#">contentTxFrameRate [p.222]</a>	Deprecated
Parameter	<a href="#">contentTxHeight [p.222]</a>	Deprecated
Parameter	<a href="#">contentTxReportedLost [p.222]</a>	Deprecated
Parameter	<a href="#">contentTxSelectedBitRate [p.222]</a>	Deprecated
Parameter	<a href="#">contentTxSent [p.222]</a>	Deprecated
Parameter	<a href="#">contentTxType [p.223]</a>	Deprecated
Parameter	<a href="#">contentTxWidth [p.223]</a>	Deprecated
Parameter	<a href="#">contentType [p.223]</a>	Introduced
Parameter	<a href="#">customCodecs [p.224]</a>	Introduced
Parameter	<a href="#">customCodecSelection [p.224]</a>	Introduced
Parameter	<a href="#">defaultGateway [p.225]</a>	Deprecated
Parameter	<a href="#">defaultIpv4Gateway [p.225]</a>	Introduced
Parameter	<a href="#">destination [p.226]</a>	Introduced
Parameter	<a href="#">dhcp [p.226]</a>	Deprecated
Parameter	<a href="#">dhcpv4 [p.226]</a>	Introduced
Parameter	<a href="#">dnsConfiguration [p.227]</a>	Introduced
Parameter	<a href="#">dtmfMuteControl [p.228]</a>	Deprecated
Parameter	<a href="#">dtmfMuteControl (template) [p.228]</a>	Deprecated
Parameter	<a href="#">encryption [p.229]</a>	Introduced

Category	Name	Change
Parameter	<a href="#">energyMillidB [p.229]</a>	Introduced
Parameter	<a href="#">ethernetAutomatic [p.231]</a>	Introduced
Parameter	<a href="#">fecOverhead [p.232]</a>	Introduced
Parameter	<a href="#">fecRecovered [p.232]</a>	Introduced
Parameter	<a href="#">filter (route) [p.232]</a>	Introduced
Parameter	<a href="#">filter [p.232]</a>	Introduced
Parameter	<a href="#">finishedBooting [p.232]</a>	Introduced
Parameter	<a href="#">flowControlReceived [p.233]</a>	Introduced
Parameter	<a href="#">flowControlSent [p.233]</a>	Introduced
Parameter	<a href="#">format [p.233]</a>	Introduced
Parameter	<a href="#">format1 [p.233]</a>	Introduced
Parameter	<a href="#">format2 [p.233]</a>	Introduced
Parameter	<a href="#">frameErrors [p.233]</a>	Introduced
Parameter	<a href="#">frameRate [p.234]</a>	Introduced
Parameter	<a href="#">framesTransferred [p.234]</a>	Introduced
Parameter	<a href="#">furFilteringEnabled [p.234]</a>	Introduced
Parameter	<a href="#">fursReceived [p.234]</a>	Introduced
Parameter	<a href="#">fursSent [p.234]</a>	Introduced
Parameter	<a href="#">g711 [p.235]</a>	Introduced
Parameter	<a href="#">g722 [p.235]</a>	Introduced
Parameter	<a href="#">g722.1 [p.235]</a>	Introduced
Parameter	<a href="#">g722.1c [p.235]</a>	Introduced
Parameter	<a href="#">g723.1 [p.235]</a>	Introduced
Parameter	<a href="#">g728 [p.235]</a>	Introduced
Parameter	<a href="#">g729 [p.235]</a>	Introduced
Parameter	<a href="#">gateway [p.235]</a>	Introduced
Parameter	<a href="#">h261 [p.238]</a>	Introduced
Parameter	<a href="#">h263 [p.238]</a>	Introduced
Parameter	<a href="#">h263+ [p.238]</a>	Introduced
Parameter	<a href="#">h263i [p.238]</a>	Introduced
Parameter	<a href="#">h264 [p.238]</a>	Introduced
Parameter	<a href="#">inCallMenuControlChair [p.240]</a>	Introduced
Parameter	<a href="#">inCallMenuControlChair (template) [p.240]</a>	Introduced
Parameter	<a href="#">inCallMenuControlGuest [p.240]</a>	Introduced

Category	Name	Change
Parameter	<a href="#">inCallMenuControlGuest (template) [p.241]</a>	Introduced
Parameter	<a href="#">Interlaced [p.242]</a>	Introduced
Parameter	<a href="#">ipAddress [p.242]</a>	Deprecated
Parameter	<a href="#">ipRangeFinish [p.242]</a>	Introduced
Parameter	<a href="#">ipRangeStart [p.242]</a>	Introduced
Parameter	<a href="#">ipv4Address [p.242]</a>	Introduced
Parameter	<a href="#">ipv4MulticastRange [p.242]</a>	Introduced
Parameter	<a href="#">ipv4Preference [p.242]</a>	Introduced
Parameter	<a href="#">ipv4Routes [p.242]</a>	Introduced
Parameter	<a href="#">ipv4SubnetMask [p.242]</a>	Introduced
Parameter	<a href="#">ipv6MulticastRange [p.243]</a>	Introduced
Parameter	<a href="#">ipv6Preference [p.243]</a>	Introduced
Parameter	<a href="#">ipv6Routes [p.243]</a>	Introduced
Parameter	<a href="#">jitter [p.244]</a>	Introduced
Parameter	<a href="#">jitterBuffer [p.244]</a>	Introduced
Parameter	<a href="#">layoutControlEx [p.247]</a>	Modified
Parameter	<a href="#">lipSyncDelayApplied [p.249]</a>	Introduced
Parameter	<a href="#">maxParticipants [p.251]</a>	Introduced
Parameter	<a href="#">mediaResources [p.252]</a>	Introduced
Parameter	<a href="#">multicast [p.253]</a>	Introduced
Parameter	<a href="#">newRouteld [p.255]</a>	Introduced
Parameter	<a href="#">ntpEnabled [p.255]</a>	Introduced
Parameter	<a href="#">ntpHost [p.255]</a>	Introduced
Parameter	<a href="#">ntpStatus [p.255]</a>	Introduced
Parameter	<a href="#">numEvents (per enumeration) [p.256]</a>	Introduced
Parameter	<a href="#">packetsErrors [p.259]</a>	Introduced
Parameter	<a href="#">packetsTransferred [p.259]</a>	Introduced
Parameter	<a href="#">password (gatekeeper) [p.261]</a>	Introduced
Parameter	<a href="#">portAssociationA [p.261]</a>	Deprecated
Parameter	<a href="#">portAssociationAv4 [p.261]</a>	Introduced
Parameter	<a href="#">portAssociationB [p.261]</a>	Deprecated
Parameter	<a href="#">portAssociationBv4 [p.261]</a>	Introduced
Parameter	<a href="#">portRangeFinish [p.262]</a>	Introduced
Parameter	<a href="#">portRangeStart [p.262]</a>	Introduced

Category	Name	Change
Parameter	<a href="#">prefixLength [p.263]</a>	Introduced
Parameter	<a href="#">rebootRequired [p.265]</a>	Introduced
Parameter	<a href="#">routeld [p.270]</a>	Introduced
Parameter	<a href="#">rtcpLipSyncDelay [p.270]</a>	Introduced
Parameter	<a href="#">rtcpOtherReports [p.270]</a>	Introduced
Parameter	<a href="#">rtcpPacketLossReported [p.270]</a>	Introduced
Parameter	<a href="#">rtcpPacketsSent [p.271]</a>	Introduced
Parameter	<a href="#">rtcpReceiveAddress [p.271]</a>	Introduced
Parameter	<a href="#">rtcpReceivePort [p.271]</a>	Introduced
Parameter	<a href="#">rtcpReceiverReports [p.271]</a>	Introduced
Parameter	<a href="#">rtcpSenderReports [p.271]</a>	Introduced
Parameter	<a href="#">rtcpTransmitAddress [p.271]</a>	Introduced
Parameter	<a href="#">rtcpTransmitPort [p.271]</a>	Introduced
Parameter	<a href="#">selectedBitRate [p.272]</a>	Introduced
Parameter	<a href="#">setting [p.272]</a>	Introduced
Parameter	<a href="#">shutdownOnly [p.272]</a>	Introduced
Parameter	<a href="#">shutdownStatus [p.273]</a>	Introduced
Parameter	<a href="#">sipMediaEncryption [p.273]</a>	Introduced
Parameter	<a href="#">siren14 [p.273]</a>	Introduced
Parameter	<a href="#">subnetMask [p.274]</a>	Deprecated
Parameter	<a href="#">temporalSpatial [p.277]</a>	Introduced
Parameter	<a href="#">type (route) [p.278]</a>	Introduced
Parameter	<a href="#">usePassword [p.279]</a>	Introduced
Parameter	<a href="#">utcOffsetHours [p.280]</a>	Introduced
Parameter	<a href="#">utcOffsetMinutes [p.280]</a>	Introduced
Parameter	<a href="#">videoCodec [p.281]</a>	Introduced
Parameter	<a href="#">videoControl [p.281]</a>	Introduced
Parameter	<a href="#">videoMedia [p.281]</a>	Introduced
Parameter	<a href="#">videoRx [p.281]</a>	Introduced
Parameter	<a href="#">videoRxActualBitRate [p.282]</a>	Deprecated
Parameter	<a href="#">videoRxBitRateLimitReason [p.282]</a>	Deprecated
Parameter	<a href="#">videoRxChannelBitRate [p.282]</a>	Deprecated
Parameter	<a href="#">videoRxCodec [p.282]</a>	Deprecated
Parameter	<a href="#">videoRxFrameRate [p.282]</a>	Deprecated

Category	Name	Change
Parameter	<a href="#">videoRxFramesReceived [p.282]</a>	Deprecated
Parameter	<a href="#">videoRxFramesReceivedWithErrors [p.282]</a>	Deprecated
Parameter	<a href="#">videoRxHeight [p.282]</a>	Deprecated
Parameter	<a href="#">videoRxInterlaced [p.282]</a>	Deprecated
Parameter	<a href="#">videoRxJitter [p.282]</a>	Deprecated
Parameter	<a href="#">videoRxLost [p.283]</a>	Deprecated
Parameter	<a href="#">videoRxReceived [p.283]</a>	Deprecated
Parameter	<a href="#">videoRxSelectedBitRate [p.283]</a>	Deprecated
Parameter	<a href="#">videoRxWidth [p.283]</a>	Deprecated
Parameter	<a href="#">videoTx [p.284]</a>	Introduced
Parameter	<a href="#">videoTxActualBitRate [p.284]</a>	Deprecated
Parameter	<a href="#">videoTxBitRateLimitReason [p.284]</a>	Deprecated
Parameter	<a href="#">videoTxChannelBitRate [p.284]</a>	Deprecated
Parameter	<a href="#">videoTxCodec [p.284]</a>	Deprecated
Parameter	<a href="#">videoTxFrameRate [p.284]</a>	Deprecated
Parameter	<a href="#">videoTxHeight [p.284]</a>	Deprecated
Parameter	<a href="#">videoTxInterlaced [p.284]</a>	Deprecated
Parameter	<a href="#">videoTxReportedLost [p.285]</a>	Deprecated
Parameter	<a href="#">videoTxSelectedBitRate [p.285]</a>	Deprecated
Parameter	<a href="#">videoTxSent [p.285]</a>	Deprecated
Parameter	<a href="#">videoTxWidth [p.285]</a>	Deprecated
Parameter	<a href="#">webAppletBandwidth [p.286]</a>	Introduced
Parameter	<a href="#">wmpProtocol [p.286]</a>	Introduced

## Version 2.8 changes

Category	Name	Change
Command	<a href="#">addressBookEntry.enumerate [p.24]</a>	Parameters added
Command	<a href="#">cdrlog.enumerate [p.38]</a>	Introduced
Command	<a href="#">conference.create [p.41]</a>	Parameters added
Command	<a href="#">conference.enumerate [p.48]</a>	Parameters added
Command	<a href="#">conference.modify [p.59]</a>	Parameters added
Command	<a href="#">conference.resetCleanupTimeout [p.67]</a>	Introduced
Command	<a href="#">conference.status [p.68]</a>	Parameters added
Command	<a href="#">conferenceme.query [p.78]</a>	Parameters added
Command	<a href="#">device.network.query [p.88]</a>	Restructured, parameters added
Command	<a href="#">gatekeeper.query [p.104]</a>	Parameters added
Command	<a href="#">participant.add [p.107]</a>	Parameters added
Command	<a href="#">participant.diagnostics [p.112]</a>	Parameters added
Command	<a href="#">participant.enumerate [p.116]</a>	Parameters added
Command	<a href="#">participant.modify [p.131]</a>	Parameters added
Command	<a href="#">participant.status [p.143]</a>	Parameters added
Command	<a href="#">services.query [p.162]</a>	Introduced
Command	<a href="#">sip.query [p.165]</a>	Parameters added
Command	<a href="#">template.create [p.168]</a>	Introduced
Command	<a href="#">template.delete [p.172]</a>	Introduced
Command	<a href="#">template.enumerate [p.173]</a>	Parameters added
Command	<a href="#">template.modify [p.177]</a>	Parameters added
Command	<a href="#">template.status [p.181]</a>	Parameters added
Fault	27 No such template <a href="#">Fault codes [p.188]</a>	Introduced
Fault	31 Template name in use <a href="#">Fault codes [p.188]</a>	Introduced
Fault	32 Too many templates <a href="#">Fault codes [p.188]</a>	Introduced
Parameter	<a href="#">actAsRecorder [p.201]</a>	Introduced
Parameter	<a href="#">addAsGuest [p.203]</a>	Introduced
Parameter	<a href="#">alternateGatekeepers [p.203]</a>	Introduced
Parameter	<a href="#">availabilityThresholdConferences [p.209]</a>	Introduced
Parameter	<a href="#">availabilityThresholdVideoPorts [p.209]</a>	Introduced
Parameter	<a href="#">cameraControl [p.212]</a>	Introduced
Parameter	<a href="#">cameraControlDefault [p.213]</a>	Introduced
Parameter	<a href="#">cleanupTimeout [p.214]</a>	Introduced

Category	Name	Change
Parameter	<a href="#">conferenceMeEnabled [p.215]</a>	Introduced
Parameter	<a href="#">contentImportant [p.217]</a>	Introduced
Parameter	<a href="#">contentRxActualBitRate [p.218]</a>	Introduced
Parameter	<a href="#">contentRxBitRateLimitReason [p.218]</a>	Introduced
Parameter	<a href="#">contentRxChannelBitRate [p.219]</a>	Introduced
Parameter	<a href="#">contentRxCodec [p.219]</a>	Introduced
Parameter	<a href="#">contentRxFrameRate [p.219]</a>	Introduced
Parameter	<a href="#">contentRxFramesReceived [p.219]</a>	Introduced
Parameter	<a href="#">contentRxFramesReceivedWithErrors [p.219]</a>	Introduced
Parameter	<a href="#">contentRxHeight [p.219]</a>	Introduced
Parameter	<a href="#">contentRxJitter [p.219]</a>	Introduced
Parameter	<a href="#">contentRxLost [p.219]</a>	Introduced
Parameter	<a href="#">contentRxReceived [p.219]</a>	Introduced
Parameter	<a href="#">contentRxSelectedBitRate [p.219]</a>	Introduced
Parameter	<a href="#">contentRxType [p.219]</a>	Introduced
Parameter	<a href="#">contentRxWidth [p.220]</a>	Introduced
Parameter	<a href="#">contentTxActualBitRate [p.221]</a>	Introduced
Parameter	<a href="#">contentTxBitRateLimitReason [p.221]</a>	Introduced
Parameter	<a href="#">contentTxChannelBitRate [p.221]</a>	Introduced
Parameter	<a href="#">contentTxCodec [p.221]</a>	Introduced
Parameter	<a href="#">contentTxError [p.222]</a>	Introduced
Parameter	<a href="#">contentTxFrameRate [p.222]</a>	Introduced
Parameter	<a href="#">contentTxHeight [p.222]</a>	Introduced
Parameter	<a href="#">contentTxMinimumBitRate [p.222]</a>	Introduced
Parameter	<a href="#">contentTxReportedLost [p.222]</a>	Introduced
Parameter	<a href="#">contentTxSelectedBitRate [p.222]</a>	Introduced
Parameter	<a href="#">contentTxSent [p.222]</a>	Introduced
Parameter	<a href="#">contentTxType [p.223]</a>	Introduced
Parameter	<a href="#">contentTxWidth [p.223]</a>	Introduced
Parameter	<a href="#">defaultIpv6Gateway [p.225]</a>	Introduced
Parameter	<a href="#">dns [p.227]</a>	Introduced
Parameter	<a href="#">events (feedback) [p.231]</a>	Introduced
Parameter	<a href="#">eventsRemaining [p.231]</a>	Introduced
Parameter	<a href="#">filter [p.232]</a>	Introduced

Category	Name	Change
Parameter	<a href="#">guest [p.236]</a>	Introduced
Parameter	<a href="#">h239Important [p.237]</a>	Deprecated
Parameter	<a href="#">h239Negotiation [p.237]</a>	Introduced
Parameter	<a href="#">index (CDR log enumerate call) [p.241]</a>	Introduced
Parameter	<a href="#">ipv4Enabled [p.242]</a>	Introduced
Parameter	<a href="#">ipv6Address [p.242]</a>	Introduced
Parameter	<a href="#">ipv6Enabled [p.242]</a>	Introduced
Parameter	<a href="#">ipv6PrefixLength [p.243]</a>	Introduced
Parameter	<a href="#">lastChairmanLeavesDisconnect [p.246]</a>	Introduced
Parameter	<a href="#">layoutControlEx [p.247]</a>	Modified
Parameter	<a href="#">linkLocalIpv6Address [p.248]</a>	Introduced
Parameter	<a href="#">linkLocalIpv6PrefixLength [p.248]</a>	Introduced
Parameter	<a href="#">maxOcsBitrate [p.250]</a>	Introduced
Parameter	<a href="#">moreThanFour [p.252]</a>	Introduced
Parameter	<a href="#">newTemplateName [p.255]</a>	Introduced
Parameter	<a href="#">nextIndex [p.255]</a>	Introduced
Parameter	<a href="#">numEvents (CDR log) [p.256]</a>	Introduced
Parameter	<a href="#">outgoingTransport [p.258]</a>	Introduced
Parameter	<a href="#">parent [p.259]</a>	Introduced
Parameter	<a href="#">portA [p.261]</a>	Modified
Parameter	<a href="#">portAssociationA [p.261]</a>	Introduced
Parameter	<a href="#">portAssociationAv6 [p.261]</a>	Introduced
Parameter	<a href="#">portAssociationB [p.261]</a>	Introduced
Parameter	<a href="#">portAssociationBv6 [p.262]</a>	Introduced
Parameter	<a href="#">portB [p.262]</a>	Modified
Parameter	<a href="#">preconfiguredParticipantsDefer [p.262]</a>	Introduced
Parameter	<a href="#">registeredAddress [p.267]</a>	Introduced
Parameter	<a href="#">registrarType [p.268]</a>	Introduced
Parameter	<a href="#">registrarUsage [p.268]</a>	Introduced
Parameter	<a href="#">registrationStatus [p.268]</a>	Introduced
Parameter	<a href="#">registrationType [p.268]</a>	Introduced
Parameter	<a href="#">remoteLinkType [p.269]</a>	Introduced
Parameter	<a href="#">reserveAudioPorts [p.269]</a>	Introduced
Parameter	<a href="#">reserveVideoPorts [p.270]</a>	Introduced

Category	Name	Change
Parameter	<a href="#">resourceAvailabilityStatus [p.270]</a>	Introduced
Parameter	<a href="#">sendResourceAvailabilityIndications [p.272]</a>	Introduced
Parameter	<a href="#">startIndex [p.273]</a>	Introduced
Parameter	<a href="#">startLocked [p.273]</a>	Introduced
Parameter	<a href="#">templateName [p.276]</a>	Introduced
Parameter	<a href="#">templateNumber [p.276]</a>	Modified
Parameter	<a href="#">useLocalCertificate [p.279]</a>	Introduced
Parameter	<a href="#">useMaximumPortsFromParent [p.279]</a>	Introduced
Parameter	<a href="#">useReservedPortsFromParent [p.279]</a>	Introduced
Parameter	<a href="#">useWebService [p.280]</a>	Introduced
Parameter	<a href="#">videoPortAllocation [p.281]</a>	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](http://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.

© 2014 Cisco Systems, Inc. All rights reserved.