# Zoom Rooms Control System API

## ZR-CSAPI Features: Version 1.1

The ZR-CSAPI is a stand-alone process that runs on the Zoom Room machine. The process exposes an SSH interface that accepts a CLI terminal connection from an external automation controller. The ZR-CSAPI then connects to the Zoom Room process via a proprietary Zoom Room API. The ZR-CSAPI translates the SSH CLI commands from the automation controller to the proprietary Zoom Room API commands for the Zoom Room. Features of the ZR-CSAPI:

- The API supports SSH only, via port 2244. It supports a single SSH connection at a time. Also, the standard Zoom Room Controller (iPad / Android tablet) cannot be connected to the Zoom Room simultaneously while ZR-CSAPI is running.
- It will not support an RS-232 connection. RS-232 support will likely **not** be added in future versions.
- For interactive terminal command line editing, ZR-CSAPI supports Arrow up/down/left/right, Del, and Backspace.
- The CLI can return either CLI-style reply text or JSON reply structures.
- The SSH interface does *not* support SSH exec commands; the interaction is strictly plain text.

## Installing and launching

Because the automation controller needs to connect to the Zoom Room over Ethernet to establish an SSH connection, the automation controller will likely need to connect to the Zoom Room machine over the corporate LAN. For security purposes, you can setup a VLAN to isolate the communication between the automation controller and the Zoom Room. You can also add a separate USB NIC, to be used just with your automation controller, for total isolation; the SSH API listens on all NICs.

Install the Zoom Room on a Mac or Windows. The Zoom Room install package will install the ZR-CSAPI executable. After the Zoom Room installer finishes, it will launch the Zoom Room. Log in to the Zoom Room as you normally would to activate the Zoom Room.

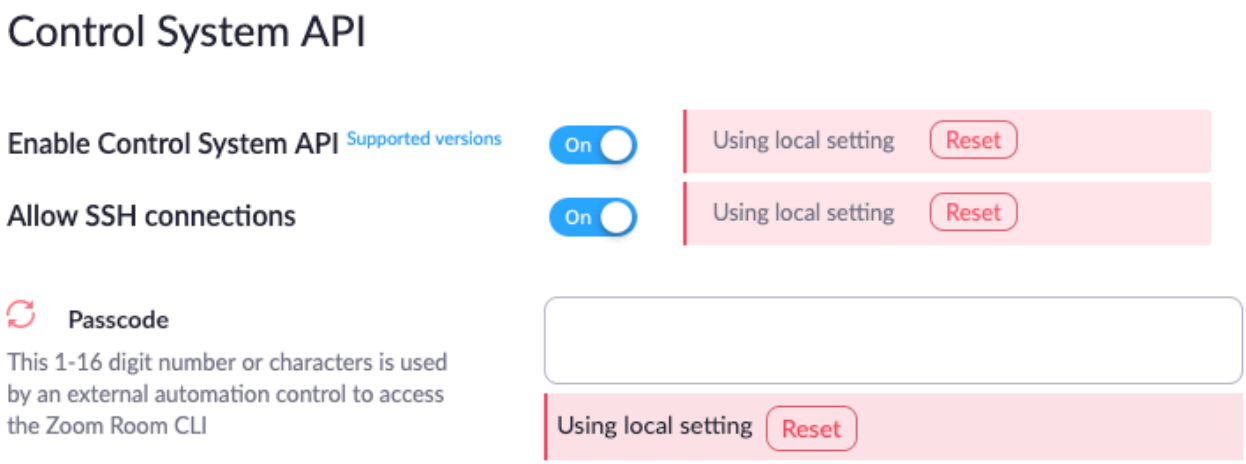
## Enabling the ZR-CSAPI for a Zoom account

Starting May 2019, the ZR-CSAPI is enabled for all Zoom Accounts. It is no longer necessary to submit a request to the Zoom operations team to enable the ZR-CSAPI for a Zoom account.

## Enabling the ZR-CSAPI per-Zoom Room

The ZR-CSAPI process must be launched by the Zoom Room process; it's not possible to manually launch the ZR-CSAPI process.

On the Zoom Web portal, in the configuration section for a Zoom Room, there is a UI to enable the ZR-CSAPI:




The customer can enable/disable the ZR-CSAPI, and set the SSH passcode, on a per-Zoom-Room basis. The Zoom Room will download this information, and launch the ZR-CSAPI with the desired SSH passcode. If the customer has many Zoom Rooms, it is possible to configure settings for a group of Zoom Rooms in bulk, at one time, using the location feature of the Web Portal.

## ZR-CSAPI Process verification

After the Zoom Room launches, you can verify that the ZR-CSAPI process is running on the Mac, and using the ps command:

```
ps | grep ZAAPI
```

And you will see a process that looks like this:

```
/Applications/ZoomPresence.app/Contents/Frameworks/ZAAPI.app/Contents/MacOS/ZAAPI -ssh -p zoomus123 -a rlb|pkg|tcp://127.0.0.1:9090 -t wxxaMzYuMDEISZyqto8LfA4r0HjYTeRepUKDu-nIb_Y.DgEWVS15Z3F0M0hTdFNtYlkxSmZZVy1PQR90Y3B8cGtnfHJsYjovLzEwLjEwLjE3LjIzMjo5MDkw
```

To run the ps command: Enable SSH login on the Mac, then remotely SSH into the Mac using the normal SSH port 22, then issue the ps command in the SSH terminal.

## Connecting the automation controller to ZR-CSAPI

When ZR-CSAPI is running, you can connect to it via SSH, over port 2244. It does not use the standard SSH port 22.

When logging into the ZR-CSAPI, Use the SSH username **zoom**.

You can set the SSH password on the Zoom web portal, under the Zoom Room configuration section.

Currently, when you turn on the ZR-SCAPI functionality, the default password is blank (the empty string). Please immediately change the password to a non-empty string.

It is possible to use the location configuration feature on the Zoom Portal to place multiple Zoom Rooms in a Location, then change the settings for all Zoom Rooms in the same location at once.

Online instructions for setting up locations is here: https://support.zoom.us/hc/en-us/articles/115000342983-Zoom-Rooms-Location-Hierarchy

Example SSH connection test: If your Zoom Room is at 10.10.1.5, connect to the ZR-CSAPI using:

```
ssh -p 2244 zoom@10.10.1.5
```

The ZR-CSAPI supports a subset of the SSH PTY pseudo-terminal, for command line interactivity. The interactive features include:

- Arrow up/down to go backward / forward in the history.
- Arrow left/right to go left/right on a command line.
- Del to delete a character after the cursor.
- Backspace to delete a character before the cursor.

The ZR-CSAPI SSH interface supports only one SSH connection at a time.

## The ZR-CSAPI SSH interface does not support SSH exec commands; the interface is strictly plain text.

While the ZR-CSAPI process is running, it's not possible to simultaneously use a Zoom Room controller: be sure to quit the Zoom Room Controller (your iPad / Android tablet) before using the ZR-CSAPI.

## ZR-CSAPI basic commands

There are several basic CLI commands:

| Command | Functionality |
|---|---|
| bye | Exit the session |
| echo on/off | Turn terminal echo on/off. Upon entry, the echo is on. |
| format json/cli | Switch between JSON format reply and CLI format reply |

## JSON format reply structure

At login, the default reply structure is CLI format. To switch to JSON reply format, issue the command:

```
format json
```

For this command:

```
zStatus SystemUnit
```

This is an example return structure in JSON format:

```
{
  "Status": {
    "message": "",
    "state": "OK"
```

```
  },
  "Sync": true,
  "SystemUnit": {
    "email": "scott.firestone_U-ygqt3HStSmbY1JfYW-OA@parasync.com",
    "login_type": "work_email",
    "meeting_number": "5526136251",
    "platform": "Mac OS X, 10.11.6",
    "room_info": {
      "account_email": "room@kanovia.com",
      "is_auto_answer_enabled": true,
      "is_auto_answer_selected": false,
      "room_name": "ZR-ScottFirestone2"
    },
    "room_version": "4.0.58637.1228"
  },
  "topKey": "SystemUnit",
  "type": "zStatus"
}
```


JSON structures have these top level keys:

- **topKey**: The name of the top level keyword in the command hierarchy; the keyword also appears at the top level of the structure.
- **Status**: error information, with a possible error message. A state of OK means success.
- **Sync**: If true, this structure is a synchronous response to a command. If false, it is an asynchronous notification.
- type:
- If **Sync** = true, the structure is a synchronous response, and this type will be zCommand, zConfiguration, or zStatus, corresponding to the original command type that triggered this response.
- If **Sync** = false, the structure is an asynchronous reply, and this type will be zCommand, zConfiguration, or zStatus if the notification was triggered by a corresponding synchronous reply. The type will be zEvent if the notification does not have a corresponding synchronous reply. One example reply with the type of zEvent is IncomingCallIndication; that notification does not have a corresponding synchronous command or reply.

# ZR-CSAPI CLI commands

There are several categories of commands:

- **zCommands**: Perform an action, or retrieve dynamic user-centric information.
- **zConfiguration**: Get/Set read/write configuration settings: Only the CLI can change these values, not external events.
- **zStatus**: Get read-only status settings. Some of these values never change, but some values can change based on external events.
- **zEvents**: Also, asynchronous notifications can alert the CLI to changes: these are called **zEvents**.

Each command specifies a path hierarchy, like the Windows registry. All commands and parameter names are case-insensitive. However, values may be case-sensitive.

## zCommands

zCommands either perform an action or retrieve dynamic user-centric information. The parameters must be provided in the order shown in the command reference. For most zCommands, all parameters are required. Boolean parameters can have values of *on* or *off*.

Example:

```
zCommand Call MuteParticipant mute: on Id: 16778240
```

In this case, the path heirarchy for this command is *Call -> MuteParticipant*, and the two required parameters are *mute* (with a value of *on*) and *Id* (with a value of *16778240*). Example reply text for this command is:

```
OK
*r CallMuteParticipantResult (status=OK):
** end
```

Command responses generally consist of *OK*, followed by a status, followed by return values, if any, followed by ***end*. For zCommands, the status and reply values are preceded by *r.

Some zCommands are In-meeting only: they can only be used when a meeting is in progress. They return an error if you attempt to call them when a meeting is not active. To find out whether the Zoom Room is in a meeting, use the zStatus command:

```
zStatus Call Status
```

After you issue a zCommand, you may receive asynchronous zEvent notifications if the command changes the state of the Zoom Room.

## zConfiguration

A zConfiguration command can get or set a single configuration setting value on the Zoom Room.

To get a value, specify the path hierarchy, down to the leaf parameter, but do not specify a value. Example:

```
zConfiguration System mute_av_begin
```

And the reply is:

```
*c zConfiguration System mute_av_begin: off
** end
OK
```

In this case, the path is *Sytem*, and the parameter is mute_av_begin. The return value is a boolean, with a value of *off*. To set this value, specify a value for the parameter:

```
zConfiguration System mute_av_begin: on
```

And the reply is:

```
** end
OK
```

zConfiguration reply text consists of ***end* followed by OK. Also, if the value changes, you will get an asynchronous notification that is value has changed, in the form of a zEvent. The zEvent has the same format as the original zConfiguration command and starts with *c. the zEvent will not end in an OK. An example of a zEvent for the *zConfiguration sytem mute_av_begin* parameter:

```
*c zConfiguration System mute_av_begin: on
** end
```

Some zConfiguration commands are In-meeting only: they get/set values for a meeting in progress; they return an error if you attempt to call them when a meeting is not active.

## zStatus

A zStatus command gets one or more read-only values. Specify the path hierarchy. Example:

zStatus Call Status

In this example, the path is *Call*, and the parameter is *Status*. The reply is:

```
*s Call Status: NOT_IN_MEETING
** end
OK
```

For zStatus commands, you must specify the path down to a certain terminal path level in the command hierarchy. The ZR-CSAPI will return all values for that level and below. The reference guide specifies the terminal path. For instance, To get a list of Audio Input devices, specify the zStatus hierarchy down to **audio Input Line** :

```
zStatus Audio Input Line
```

And you will get parameters for that level in the hierarchy, and below. This reply returns an array of devices:

```
*s Audio Input Line 1 id: Sennheiser SC70 USB CTRL
*s Audio Input Line 1 Name: Sennheiser SC70 USB CTRL
*s Audio Input Line 1 Alias:
*s Audio Input Line 2 id: HD Pro Webcam C920
*s Audio Input Line 2 Name: HD Pro Webcam C920
*s Audio Input Line 2 Alias:
*s Audio Input Line 3 id: Built-in Input
*s Audio Input Line 3 Name: Built-in Input (Line In)
*s Audio Input Line 3 Alias:
** end
OK
```

Each zStatus reply line has the *s prefix. zStatus values are read-only but can change from external changes to the state of the Zoom Room. If the value changes, you will get an asynchronous notification that the value has changed, in the form of a zEvent. The zEvent has the same format as the zStatus command and starts with *s. The zEvent will not end in an OK. An example of a zEvent for the *zStatus__Call Status* parameter:

```
*s Call Status: CONNECTING_MEETING
** end
```

## zEvent

zEvents are asynchronous notifications. They can come in several forms:

- Changes to zStatus parameters
- Changes to zConfiguration parameters
- Events triggered by zCommands
- Other asynchronous events, such as a remote participant attempting to join a conference.

As an example, if a remote participant attempts to join a meeting hosted by the Zoom Room, and if the Zoom Room is not configured to auto-accept new meeting participants, the ZR-CSAPI will issue this zEvent notification to indicate that the participant requests to enter the conference:

```
*e IncomingCallIndication callerJID: thqrasdqs4wzdnch4kcbiw@xmpp.zoom.us
*e IncomingCallIndication calleeJID:
*e IncomingCallIndication meetingID: g
*e IncomingCallIndication password:
*e IncomingCallIndication meetingOption: 6
*e IncomingCallIndication meetingNumber: 5351459175
*e IncomingCallIndication callerName: Bob Smith
*e IncomingCallIndication avatarURL:
*e IncomingCallIndication lifeTime: 60
** end
```

zEvent notifications that are not triggered by changes to zStatus or zConfiguration changes will have an *e prefix.

## Arrays

Some hierarchical paths include an array index for a level in the path. In the Command Reference, this array level is indicated by a **n** in the path specification. For example, when retrieving a list of PhoneBook entries, it is possible to Ask for a range of entries, starting at an offset index, and spaning a maximum number of return values. In this case, the return information contains an array of multiple parameters underneath it, with the index array value preceding the values under the array level:

```
zcommand phonebook list offset: 2 Limit: 2
OK
*r PhonebookListResult (status=OK):
*r PhonebookListResult resultInfo Offset: 2
*r PhonebookListResult resultInfo Limit: 2
*r PhonebookListResult resultInfo TotalRows: 2
*r PhonebookListResult Contact 3 jid: hap3piapreqhiqztagcrsq@xmpp.zoom.us
*r PhonebookListResult Contact 3 screenName: AlexBaker
*r PhonebookListResult Contact 3 firstName: AlexBaker
*r PhonebookListResult Contact 3 lastName:
*r PhonebookListResult Contact 3 phoneNumber:
*r PhonebookListResult Contact 3 email:
rooms_hAP3piAPRrqHIQzXAgCRsQ@gmail.com
*r PhonebookListResult Contact 3 avatarURL:
```

```
*r PhonebookListResult Contact 3 presence: ZoomIMPresence_Offline
*r PhonebookListResult Contact 3 onDesktop: off
*r PhonebookListResult Contact 3 onMobile: off
*r PhonebookListResult Contact 4 jid: lkfetphqtnyarhkofk4baa@xmpp.zoom.us
*r PhonebookListResult Contact 4 screenName: ZR-FifthFloor
*r PhonebookListResult Contact 4 firstName: ZR-FifthFloor
*r PhonebookListResult Contact 4 lastName:
*r PhonebookListResult Contact 4 phoneNumber:
*r PhonebookListResult Contact 4 email:
Bob.Smoth_lkFETpHqTnycRHKOFk4BaA@parasync.com
*r PhonebookListResult Contact 4 avatarURL:
*r PhonebookListResult Contact 4 presence: ZoomIMPresence_Offline
*r PhonebookListResult Contact 4 onDesktop: off
*r PhonebookListResult Contact 4 onMobile: off
** end
```

## Call Participants IDs

When a meeting is active, you can get a list of participants. Each participant has a meeting participant Id. However, the bottom 10 bits of the Id are ignored. To compare two Ids, you must mask off the bottom 10 bits.

## Controller connections

The Zoom Room can support either the ZR-CSAPI, or a Zoom Room Controller, but not both. If one type of controller connects, the other will disconnect.

Normally, the ZR-CSAPI maintains a pipe connection to the Zoom Room. But if the Zoom Room Controller connects to the same Zoom Room, then the ZR-CSAPI will disconnect its pipe from the Zoom Room. If the SSH login is active, then the ZR-CSAPI will issue this zEvent to the SSH terminal:

```
*e Pipe OtherControllerLogin
** end
```

If the SSH login is not active, then the ZR-CSAPI will still disconnect its pipe.

There are two ways to re-establish the ZR-CSAPI pipe connection of the ZR:

1. Log in to the ZR-CSAPI. If you are currently logged in, you can log out, and then log back in again.

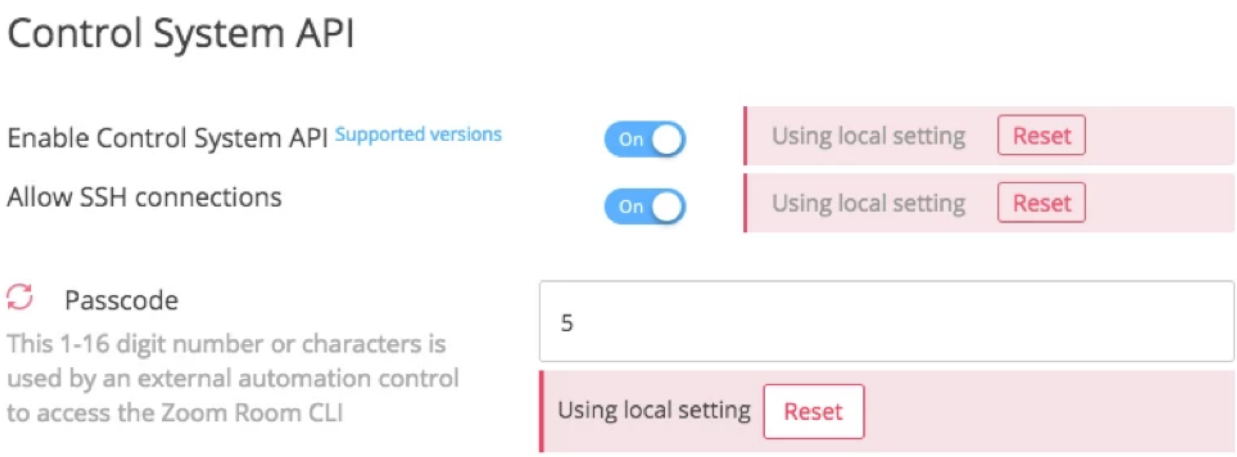
2. Issue any ZR-CSAPI command. The command will not execute, but you will get this zEvent message:

```
*e Pipe Reconnecting
** end
```

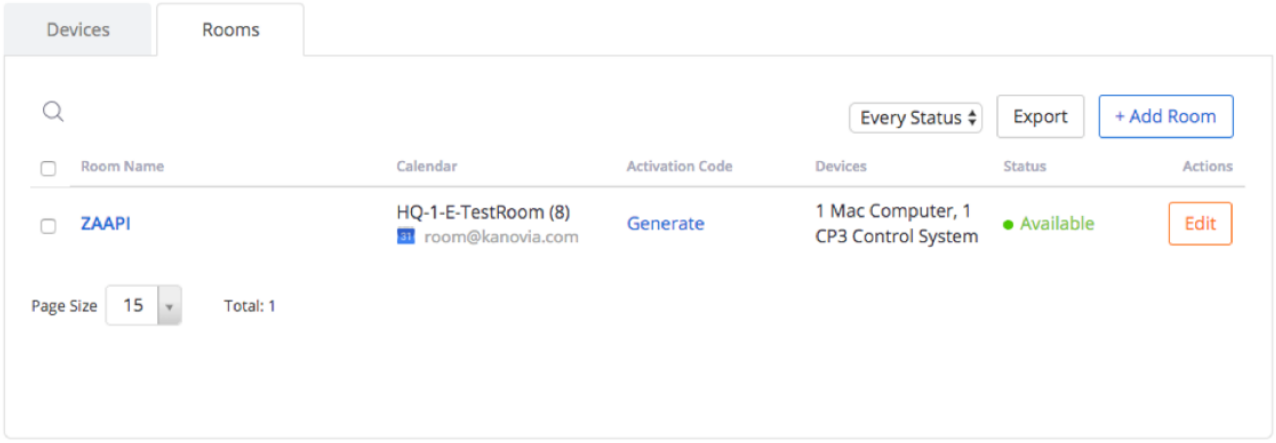
This message indicates that the ZR-CSAPI has re-connected its pipe to the ZR. You can then re-issue your previous command to execute it.

# Web Portal UI

- On the Zoom Portal website, under the Zoom Room settings, there is a UI to enable/disable ZR-CSAPI, enable/disable the SSH connection, and set the password of the SSH connection.



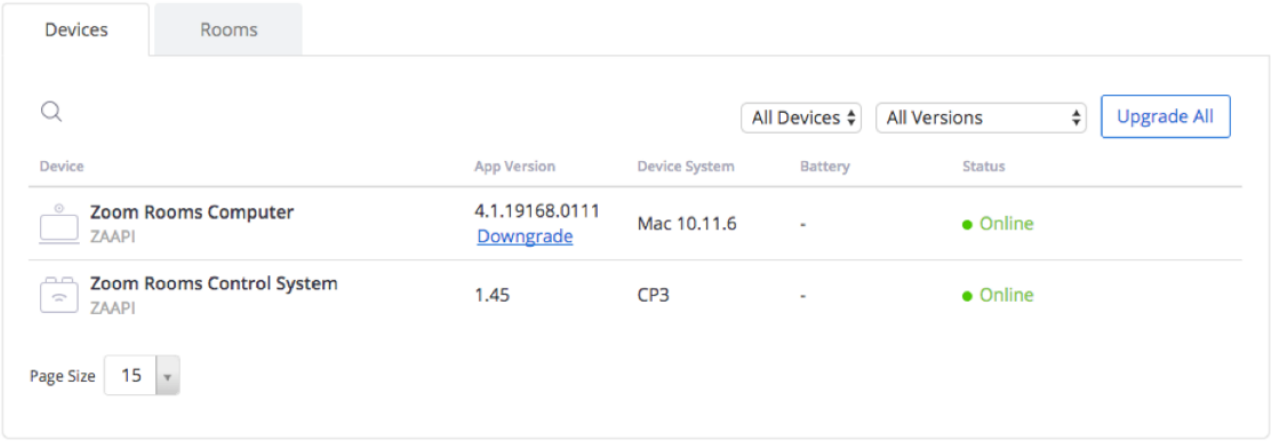

On the Zoom Portal website, under the Zoom Room settings, the Room tab will list some details on the API for that Zoom Room. In this case, the automation controller using the API has set deviceSystem to CP3:




You will also see **Zoom Rooms Automation Controllers** on the list. An external automation control system will send the Zoom Room the **App version** and **Device System** for the control system, and the Zoom Room will forward that information to the Web Portal to be listed as info for the device. For instance, to show App Version **1.45**, and Device System **CP3**, issue these ZR-CSAPI CLI commands:

```
zConfiguration Client appVersion: 1.45
zConfiguration Client deviceSystem: CP3
```

And those values will be displayed on the Devices tab on the web portal:




Also, the status of **Online** will appear, if an automation controller is connected to ZR-CSAPI via SSH.

## ZR-CSAPI log files

Log files are created automatically, to capture all severity levels.

Every time the ZR-CSAPI launches, the old log files are deleted, and a new log file is created.

When a log file reaches 20 MB in size, the logs continue with a new file; up to 4 log files at 20 MB in size may exist. The log files will be located here:

Mac: ~/Library/Logs/zoom.us/zaapi/zaapi_XXX.log

Windows: %APPDATA%\ZoomRooms\logs\zaapi_xxx.log

Please IGNORE the files that start with the uppercase ZAAPI_XXX.

# zCommand

## zCommand Dial Start

Start or join a meeting that is scheduled to occur now.

In order for the Zoom Room to be able to start the meeting before the original host joins, the meeting should be scheduled using the “Enable join before host” option. The meeting number will be an integer; however, in the future, the Zoom Room may support string-based vanity phone numbers, so this parameter is a string that currently represents an integer. It’s often useful to filter out the InfoResult Info callin_country_list, if you don’t need that info, by first invoking: **zFeedback Register Op: ex Path: /Event/InfoResult/info/callin_country_list**

Schema:

```
zCommand Dial Start meetingNumber: <string>
```

Example:

```
zCommand Dial Start meetingNumber: 3348757790
OK
*r DialStartResult (status=OK):
** end

*s Call Status: CONNECTING_MEETING
** end

...

*r InfoResult Info real_meeting_id: m71q6DwwTd6Y7b0ua5ubtQ==
*r InfoResult Info meeting_id: 371440459
*r InfoResult Info participant_id: 28
*r InfoResult Info my_userid: 16778240
*r InfoResult Info am_i_original_host: off
*r InfoResult Info is_webinar: off
*r InfoResult Info is_view_only: off
*r InfoResult Info meeting_type: NORMAL
*r InfoResult Info meeting_password:
*r InfoResult Info dialIn: +1 669 900 6833;+1 646 558 8656
*r InfoResult Info toll_free_number:
*r InfoResult Info international_url: /zoomconference
*r InfoResult Info is_toll_free_callin_list_available: on
*r InfoResult Info is_callin_country_list_available: on
*r InfoResult Info is_calling_room_system_enabled: on
*r InfoResult Info support_callout_type: NONE
*r InfoResult Info user_type: BASIC
*r InfoResult Info callin_country_list ...
*r InfoResult Info invite_email_subject: Please join Zoom meeting in
progress
```

```
*r InfoResult Info invite_email_content: You are invited to a Zoom
meeting now.

Join from PC, Mac, Linux, iOS or Android: https://zoom.us/j/371440459

Or iPhone one-tap:
    US: +16699006833,,371440459#  or +16465588656,,371440459#
Or Telephone:
    Dial(for higher quality, dial a number based on your current
location):
        US: +1 669 900 6833  or +1 646 558 8656
    Meeting ID: 371 440 459
    International numbers available: https://zoom.us/u/ackaxrkh2e

Or an H.323/SIP room system:
    H.323:
        162.255.37.11 (US West)
        162.255.36.11 (US East)
        221.122.88.195 (China)
        115.114.131.7 (India)
        213.19.144.110 (EMEA)
        202.177.207.158 (Australia)
        209.9.211.110 (Hong Kong)
        64.211.144.160 (Brazil)
        69.174.57.160 (Canada)
    Meeting ID: 371 440 459

    SIP: 371440459@zoomcrc.com


** end

...

*r ListParticipantsResult ...

*s Call Status: IN_MEETING
** end
```

**meetingNumber**: The meeting number of a meeting that is scheduled to happen today. You can get a list of scheduled meetings for this Zoom Room using **zCommand Bookings List**.

## zCommand Dial StartPmi

Start a PMI meeting, using the PMI number for this room. The duration does not dictate when the meeting ends; the duration is used only to show the intended duration of the meeting in the calendar integration. The external Zoom Rooms Scheduling Display will show this duration.

Schema:

```
zCommand Dial StartPmi Duration: <int>
```

Example:

```
zCommand Dial StartPmi Duration: 9
OK
*r DialStartPmiResult (status=OK):
** end

*s Call Status: CONNECTING_MEETING
** end

...

*r InfoResult ...
** end

...

*r ListParticipantsResult ...

*e CallConnect success: off
*e CallConnect avail_minutes: 0
** end

*s Call Status: IN_MEETING
** end
```

**Duration**: The time in minutes the meeting is intended to last. The Zoom Room will book an instant meeting, which will appear on integrated calendars.

## zCommand Dial Join

Join a meeting in progress.

Schema:

```
zCommand Dial Join meetingNumber: <string>
```

Example:

```
command dial join meetingnumber: 804452190
OK
*r DialJoinResult (status=OK):
** end

*s Call Status: CONNECTING_MEETING
** end
```

```
...

*r InfoResult ...

** end

...

*r ListParticipantsResult ...

*s Call Status: IN_MEETING
** end
```

**meetingNumber**: The meeting number to join. This meeting number will be an integer, stored as a string. In the future, Zoom may support vanity phone numbers in the form of a string that may include alphanumeric characters.

## zCommand Call Disconnect - In-Meeting

Ends the current meeting. If the ZR is not connected to a meeting, this command results in an error message.

Schema:

```
zCommand Call Disconnect
```

Example

```
zCommand Call Disconnect
OK
*r CallDisconnectResult (status=OK):
** end

*s Sharing ...

*e CallDisconnect success : on
** end

...

*s Call Status: NOT_IN_MEETING
** end
```

## zCommand Call Info - In-Meeting

Gets the info of the current meeting in progress. If the ZR is not connected to a meeting, this command results in an error message.

Schema:

```
zCommand Call Info
```

OK
*r InfoResult (status=OK):
*r InfoResult Info real_meeting_id: <string> An internal meeting Id that is only used in the Web Zoom back end. This ID is not used by the Zoom Room.
*r InfoResult Info meeting_id: <string> Meeting ID of the call; users will enter this ID to call in.
*r InfoResult Info participant_id: <int> If the participant wants to join the conference using a video connection from a desktop, and dial in via PSTN, then when dialing in, the user should enter this participant Id, so that the Zoom Room can correlate the video and audio. This value is not. the same as a user id, which is a value assigned to each meeting participant.
*r InfoResult Info my_userid: <int> A meeting participant ID: Use this ID to get/set in-meeting parameters for a participant. The bottom 10 bits are ignored.
*r InfoResult Info am_i_original_host: <on/off> The Zoom Room is the original host if it started the meeting via its PMI.
*r InfoResult Info is_webinar: <on/off> Whether this call is a webinar
*r InfoResult Info is_view_only: <on/off> Whether the Zoom Room is a view-only attendee in a webinar.
*r InfoResult Info meeting_type: <NORMAL, SHARING_LAPTOP, PSTN_CALLOUT, NONE> Type of meeting.
*r InfoResult Info meeting_password: <string> The meeting password
*r InfoResult Info dialIn: <string> PSTN phone numbers to use to dial into the meeting.
*r InfoResult Info toll_free_number: <string> The tool free PSTN number that participants use to join the conference
*r InfoResult Info international_url: <string> The web page link on zoom.us that shows the list of all international dial-in numbers.
*r InfoResult Info is_toll_free_callin_list_available: <on/off> DEPRECATED
*r InfoResult Info is_callin_country_list_available: <on/off> DEPRECATED
*r InfoResult Info is_calling_room_system_enabled: <on/off> if the customer has purchasd the CRC (Cloud Room Connector), the customer has the ability to invite H.323 endpoints that connect through the CRC. On the iOS ZRC, when inviting a new participant to a meeting, if the CRC is installed, then one tab will display the option "Invite Room System"
*r InfoResult Info support_callout_type: <US_ONLY, INTERNATIONAL, NONE> The callout types allowed for the Zoom Room. These permissions are configured on the web portal. None = no call out, US_ONLY = call to US only, INTERNATIONAL = allow international call out. To add international dial out, you need to go to My Profile -> Billing, on the Web portal, and add the Audio Conferencing Options add-on.

*r InfoResult Info user_type: <BASIC, PRO, CORP, NONE> Type of user. BASIC is a free user, PRO is a paid user with a license, and CORP is an on-premise user.
*r InfoResult Info callout_country_list n id: <string> Configured on the Web portal, if you have added the Audio Conferencing Options add-on.
[*r InfoResult Info callout_country_list n name: <string> ]Friendly Country name
[*r InfoResult Info callout_country_list n code: <string>] Multiple digit country code
[*r InfoResult Info callout_country_list n number: <string> ]Full number, including country code
[*r InfoResult Info callout_country_list n display_number: <string> ]Friendly number format
[*r InfoResult Info callin_country_list n id: <string> ]A list of phone numbers in foreign countries to use when dialing in.
[*r InfoResult Info callin_country_list n name: <string>]
[*r InfoResult Info callin_country_list n code: <string>]
[*r InfoResult Info callin_country_list n number: <string>]
[*r InfoResult Info callin_country_list n display_number: <string>]
[*r InfoResult Info toll_free_callin_list n id: <string>] Configured on the Web portal, if you have added the Audio Conferencing Options add-on.
[*r InfoResult Info toll_free_callin_list n name: <string>]
[*r InfoResult Info toll_free_callin_list n code: <string>]
[*r InfoResult Info toll_free_callin_list n number: <string>]
[*r InfoResult Info toll_free_callin_list n display_number: <string>]
*r InfoResult Info invite_email_subject: <string> The subject line for the invitation email that goes to invited participants
*r InfoResult Info invite_email_content: <string> The body of the email that goes to invited participants.
*r InfoResult Info default_callin_country: <string>
*r InfoResult Info schedule_option: <string>
*r InfoResult Info schedule_option2: <string>
*r InfoResult Info is_waiting_room: <on/off>
*r InfoResult Info meeting_list_item meetingName: <string>
*r InfoResult Info meeting_list_item meetingNumber: <string>
*r InfoResult Info meeting_list_item hostName: <string>
*r InfoResult Info meeting_list_item startTime: <string>
*r InfoResult Info meeting_list_item endTime: <string>
*r InfoResult Info meeting_list_item creatorEmail: <string>
*r InfoResult Info meeting_list_item creatorName: <string>
*r InfoResult Info meeting_list_item isPrivate: <on/off>
*r InfoResult Info meeting_list_item checkedIn: <1/0> //Whether meeting has been checked in
*r InfoResult Info meeting_list_item isInstantMeeting: <on/off>
*r InfoResult Info meeting_list_item calendarID: <string> //ID on calender

```
*r InfoResult Info meeting_list_item calendarChangeKey: <string>
*r InfoResult Info meeting_list_item accessRole: <string>
*r InfoResult Info meeting_list_item location: <string>
*r InfoResult Info meeting_list_item scheduledFrom: <string> //How the
meeting was scheduled
*r InfoResult Info meeting_list_item thirdparty service_provider:
<string>
*r InfoResult Info meeting_list_item thirdparty meeting_number: <string>
*r InfoResult Info meeting_list_item thirdparty sip_address: <string>
*r InfoResult Info meeting_list_item thirdparty h323_address: <string>
** end
```

Example

```
zcommand call info
OK
*r InfoResult (status=OK):
*r InfoResult Info real_meeting_id: dz4mtgZsTKSZ2XBb7FUJkw==
*r InfoResult Info meeting_id: 804452190
*r InfoResult Info participant_id: 15
*r InfoResult Info my_userid: 16779264
*r InfoResult Info am_i_original_host: off
*r InfoResult Info is_webinar: off
*r InfoResult Info is_view_only: off
*r InfoResult Info meeting_type: NORMAL
*r InfoResult Info meeting_password:
*r InfoResult Info dialIn: +1 646 876 9923;+1 669 900 6833
*r InfoResult Info toll_free_number: ...
*r InfoResult Info international_url: /zoomconference
*r InfoResult Info is_toll_free_callin_list_available: on
*r InfoResult Info is_callin_country_list_available: on
*r InfoResult Info is_calling_room_system_enabled: on
*r InfoResult Info support_callout_type: INTERNATIONAL
*r InfoResult Info user_type: PRO
*r InfoResult Info toll_free_callin_list 1 id: US
*r InfoResult Info toll_free_callin_list 1 name: US
*r InfoResult Info toll_free_callin_list 1 code: 1
*r InfoResult Info toll_free_callin_list 1 number: 1 8773690926
*r InfoResult Info toll_free_callin_list 1 display_number: +1 877 369
0926
*r InfoResult Info toll_free_callin_list 2 id: US
*r InfoResult Info toll_free_callin_list 2 name: US
*r InfoResult Info toll_free_callin_list 2 code: 1
*r InfoResult Info toll_free_callin_list 2 number: 1 8778535247
*r InfoResult Info toll_free_callin_list 2 display_number: +1 877 853
5247
*r InfoResult Info invite_email_subject: Please join Zoom meeting in
progress
```

```
*r InfoResult Info invite_email_content: You are invited to a Zoom
meeting now.

Join from PC, Mac, Linux, iOS or Android: https://success.zoom.us/j/
804452190

Or iPhone one-tap:
    US: +16468769923,,804452190#  or +16699006833,,804452190#
Or Telephone:
    Dial(for higher quality, dial a number based on your current
location):
        US: +1 646 876 9923  or +1 669 900 6833  or +1 877 369 0926 (Toll
Free) or +1 877 853 5247  (Toll Free)
    Meeting ID: 804 452 190
    International numbers available: https://zoom.us/u/acpJ3PkjBo

Or an H.323/SIP room system:
    H.323:
        162.255.37.11 (US West)
        162.255.36.11 (US East)
        221.122.88.195 (China)
        115.114.131.7 (India)
        213.19.144.110 (EMEA)
        202.177.207.158 (Australia)
        209.9.211.110 (Hong Kong)
        64.211.144.160 (Brazil)
        69.174.57.160 (Canada)
    Meeting ID: 804 452 190

    SIP: 804452190@zoomcrc.com

Or Skype for Business (Lync):
    https://success.zoom.us/skype/804452190


*r InfoResult Info default_callin_country: US
*r InfoResult Info schedule_option: 2382404203751407872
*r InfoResult Info schedule_option2: 1649267441664
*r InfoResult Info is_waiting_room: off
*r InfoResult Info meeting_list_item meetingName:
*r InfoResult Info meeting_list_item meetingNumber:
*r InfoResult Info meeting_list_item hostName:
*r InfoResult Info meeting_list_item startTime:
*r InfoResult Info meeting_list_item endTime:
*r InfoResult Info meeting_list_item creatorEmail:
*r InfoResult Info meeting_list_item creatorName:
```

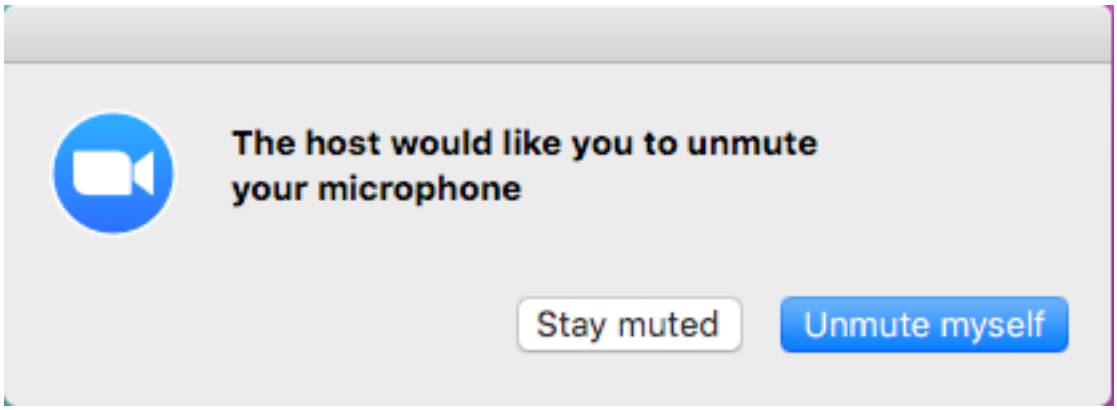
```
*r InfoResult Info meeting_list_item isPrivate: off
*r InfoResult Info meeting_list_item checkedIn: off
*r InfoResult Info meeting_list_item isInstantMeeting: off
*r InfoResult Info meeting_list_item calendarID:
*r InfoResult Info meeting_list_item calendarChangeKey:
*r InfoResult Info meeting_list_item accessRole:
*r InfoResult Info meeting_list_item location:
*r InfoResult Info meeting_list_item scheduledFrom: ZRP
*r InfoResult Info meeting_list_item thirdparty service_provider: -1
*r InfoResult Info meeting_list_item thirdparty meeting_number:
*r InfoResult Info meeting_list_item thirdparty sip_address:
*r InfoResult Info meeting_list_item thirdparty h323_address:
** end
```

## zCommand Call MuteAll - In-Meeting

Mute the audio of all participants in the current meeting.

There are several scenarios:

- If the Zoom Room is not the host, this command has no effect.
- If the Zoom Room is the host, then the first time the Zoom Room unmutes another participant's audio, that participant's client will ask permission from the participant to unmute:




- The second and subsequent times that the host Zoom Room attempts to unmute a participant's audio, that participant's audio will be automatically unmuted if the participant previously gave permission to unmute audio.

Schema:

```
zCommand Call MuteAll mute: <on/off>
```

Example:

```
zcommand call muteall mute: off
OK
*r CallMuteAllResult (status=OK):
** end
```

If the other pariticipant agrees to allow audio to be unmuted, you get this notification:

```
...
*r ListParticipantsResult audio_status state: AUDIO_UNMUTED
...
```

**Mute**: If the Zoom Room is the host of the meeting (the is_host parameter is set to on in the ListParticipant result for Zoom Room), set this parameter to true to audio mute all users in a meeting; set to false to audio unmute all users. When unmuting all users, in some cases, the remote users need to agree to the unmute operation. If the ZR is not connected to a meeting, this command results in an error message.

## zCommand Call MuteParticipant - In-Meeting

Mute the audio of a specific participant in a meeting. The same permission mechanism is active as in the MuteAll command. If the ZR is not connected to a meeting, this command results in an error message. The bottom 10 bits of the Id are not significant. Do not use this command to mute the microphone audio of the Zoom Room itself; instead, use **zConfiguration Call Microphone**.

Schema:

```
zCommand Call MuteParticipant mute: <on/off> Id: <int>
```

Example:

```
zcommand call muteparticipant mute: off id: 33555457
OK
*r CallMuteParticipantResult (status=OK):
** end
```

If the other pariticipant agrees to allow audio to be unmuted, you get this notification:

```
*r ListParticipantsResult user_name: Scott Firestone
...
*r ListParticipantsResult audio_status state: AUDIO_UNMUTED
...
```

**Mute**: set to on to mute the audio of the participant, and off to unmute the participant. When unmuting a user, the remote user may need to agree to the unmute operation.

**Id**: the ID of the participant, according to the participant list. To see the participants list, issue the command **zCommand Call ListParticipants**. The lower 10 bits of the participant ID are ignored.

## zCommand Call ListParticipants - In-Meeting

List all participants in a meeting. If the ZR is not connected to a meeting, this command results in an error message.

**Note:** When switching between the ZR-CSAPI and the Zoom Room Controller (iOS / Android tablet), make sure to exit existing meetings first. Otherwise, the participants list may not update.

**Note:** The **is_recording** and **can_record** fields are only applicable to the other participants and those who are not host, not the Zoom Room itself. If another participant is host, the **is_recording** and **can_record** fields do not apply to them either. For the Zoom Room itself, rely on the **RecordingInfo zEvent**.

Schema:

```
zcommand call listparticipants
OK
*r ListParticipantsResult (status=OK):
*r ListParticipantsResult resultInfo TotalRows: <int>
*r ListParticipantsResult n user_name: <string> // The username
*r ListParticipantsResult n user_id: <string> // The user ID, used only
in the current meeting. The bottom 10 bits are ignored.
*r ListParticipantsResult n is_host: <on/off> // Whether this particpant
started the meeting as a host.
*r ListParticipantsResult n is_myself: <on/off> // Whether the
participant is this Zoom Room.
*r ListParticipantsResult n can_record: <on/off> // Whether this
participant can start a recording. (do not use, use is_host instead)
*r ListParticipantsResult n is_recording: <on/off> // Whether this
participant has started a recording.
*r ListParticipantsResult n avatar_url: <string> // The avatar iamge URL
for the participant
*r ListParticipantsResult n local_recording_disabled: <on/off> // If this
participant is not allowed to record to a local file
*r ListParticipantsResult n is_video_can_mute_byHost: <on/off> // Whether
the host can video mute this participant.
*r ListParticipantsResult n is_video_can_unmute_byHost:  <on/off> //
Whether the host can video unmute this participant after getting
permission from the participant.
*r ListParticipantsResult n isCohost: <on/off> // A cohost has all
privileges of a host. Not supported on the ZR.
*r ListParticipantsResult n user_type: <NORMAL, H323, CALL_IN> // Type of
call
*r ListParticipantsResult n audio status type: <AUDIO_VOIP, AUDIO_TELE,
AUDIO_NONE> // Type of participant. AUDIO_VOIP: connected over
*r ListParticipantsResult n audio status state: <AUDIO_MUTED,
AUDIO_UNMUTED> // Whether audio is muted or unmuted
*r ListParticipantsResult n video status has_source: <on/off>
*r ListParticipantsResult n video status is_sending: <on/off> // Whether
the participants is sending video others. Otherwise, this person's camera
is muted.
*r ListParticipantsResult n video status is_receiving: <on/off> //
DEPRECATED
*r ListParticipantsResult n camera status can_i_request_control: <on/off>
// Whether it is possible to request control of the camera for this
participant
*r ListParticipantsResult n camera status am_i_controlling: <on/off> //
Whether the Zoom Room is controlling the camera for this participant
*r ListParticipantsResult n camera status can_switch_camera: <on/off> //
Whether the Zoom Room can switch the camera source for this particpant
```

```
*r ListParticipantsResult n camera status can_move_camera: <on/off> //
Whether the zoom Room can move the camera for this participant.
*r ListParticipantsResult n camera status can_zoom_camera: <on/off>
Whether the zoom Room can zoom the camera for this participant.
*r ListParticipantsResult n is_client_support_closed_caption: <on/off> //
Whether client supports closed caption
*r ListParticipantsResult n can_edit_closed_caption: <on/off> //Whether
client can edit closed caption
*r ListParticipantsResult n is_client_support_coHost: <on/off> //Whether
client supports co-host
*r ListParticipantsResult n hand_status is_valid: <on/off> //Whether hand
status is supported
*r ListParticipantsResult n hand_status is_raise_hand: <on/off> //Only
visible if hand_status is_valid is true, whether hand is raised
*r ListParticipantsResult n hand_status time_stamp: <int> //Only visible
if hand_status is_valid is true, timestamp of hand status
*r ListParticipantsResult n event: <None,
ZRCUserChangedEventJoinedMeeting, ZRCUserChangedEventLeftMeeting,
ZRCUserChangedEventUserInfoUpdated, ZRCUserChangedEventHostChanged> //
The event will be none, if there is no asynchronous change to the status
of the participant. Update: ZRCUserChangedEventJoinedMeeting and
ZRCUserChangedEventLeftMeeting notifications are not sent; instead, you
will get a revised total participant list via the *r
ListParticipantsResult asynchronous notification.
*r ListParticipantsResult 2 ...
...
** end
```

Example:

```
zcommand call listparticipants
OK
*r ListParticipantsResult (status=OK):
*r ListParticipantsResult resultInfo TotalRows: 2
*r ListParticipantsResult 1 user_name: ZR-ScottFirestone2
*r ListParticipantsResult 1 user_id: 16778240
*r ListParticipantsResult 1 is_host: on
*r ListParticipantsResult 1 is_myself: on
*r ListParticipantsResult 1 can_record: off
*r ListParticipantsResult 1 is_recording: off
*r ListParticipantsResult 1 avatar_url: https://zoom.us/p/U-
ygqt3HStSmbY1JfYW-OA/15d64cfd-3b0d-46c0-93c1-08985d0c96b1-2618
*r ListParticipantsResult 1 local_recording_disabled: on
*r ListParticipantsResult 1 is_video_can_mute_byHost: on
*r ListParticipantsResult 1 is_video_can_unmute_byHost: on
*r ListParticipantsResult 1 isCohost: off
*r ListParticipantsResult 1 user_type: NORMAL
*r ListParticipantsResult 1 audio_status type: AUDIO_VOIP
```

```
*r ListParticipantsResult 1 audio_status state: AUDIO_UNMUTED
*r ListParticipantsResult 1 video_status has_source: on
*r ListParticipantsResult 1 video_status is_sending: on
*r ListParticipantsResult 1 video_status is_receiving: off
*r ListParticipantsResult 1 camera_status can_i_request_control: off
*r ListParticipantsResult 1 camera_status am_i_controlling: off
*r ListParticipantsResult 1 camera_status can_switch_camera: off
*r ListParticipantsResult 1 camera_status can_move_camera: off
*r ListParticipantsResult 1 camera_status can_zoom_camera: off
*r ListParticipantsResult 1 is_client_support_closed_caption: off
*r ListParticipantsResult 1 can_edit_closed_caption: off
*r ListParticipantsResult 1 is_client_support_coHost: off
*r ListParticipantsResult 1 hand_status is_valid: off
*r ListParticipantsResult 1 event: None
*r ListParticipantsResult 2 user_name: Scott Firestone
...
** end
```

## zCommand Call Accept

You can configure a Zoom Room to NOT accept incoming calls automatically, by turning off the setting on the Web portal “Automatically accept incoming call and far end camera control”. When another user dials the Zoom Room directly via the Contacts list, the incoming call arrives, and the ZR-CSAPI will issue a **zEvent IncomingCallIndication**notification message containing a **callerJID** of the incoming call. If the client wants to accept the incoming caller, the client should issue this command with the callerJID from the **zEvent IncomingCallIndication** notification.

Schema:

```
zCommand Call Accept callerJID: <string>
```

Example: An incoming call arrives:

```
*e IncomingCallIndication callerJID: thqrasdqs4wzdnch8kcbiw@xmpp.zoom.us
*e IncomingCallIndication calleeJID:
*e IncomingCallIndication meetingID: :
*e IncomingCallIndication password:
*e IncomingCallIndication meetingOption: 6
*e IncomingCallIndication meetingNumber: 815366714
*e IncomingCallIndication callerName: Scott Firestone Testing ZR
*e IncomingCallIndication avatarURL:
*e IncomingCallIndication lifeTime: 60
** end
```

And then the CLI accepts the call:

```
zCommand Call Accept callerJID: thqrasdqs4wzdnch8kcbiw@xmpp.zoom.us
OK
*r CallAcceptRejectResult (status=OK):
** end
```

```
*s Call Status: CONNECTING_MEETING
** end

...
*r ListParticipantsResult ...
...

*s Call Status: IN_MEETING
** end
```

**callerJID**: The callerJID from the IncomingCallIndication event.

## zCommand Call Reject

Similar to **zCommand Call Accept**, but this command rejects the call. Reject incoming calls using the JID that was included in the **zEvent IncomingCallIndication** notification.

Schema:

```
zCommand Call Reject callerJID: <string>
```

Example: An incoming call arrives:

```
*e IncomingCallIndication callerJID: thqrasdqs4wzdnch8kcbiw@xmpp.zoom.us
*e IncomingCallIndication calleeJID:
*e IncomingCallIndication meetingID: :
*e IncomingCallIndication password:
*e IncomingCallIndication meetingOption: 6
*e IncomingCallIndication meetingNumber: 815366714
*e IncomingCallIndication callerName: Scott Firestone Testing ZR
*e IncomingCallIndication avatarURL:
*e IncomingCallIndication lifeTime: 60
** end
```

And then the CLI rejects the call:

```
zCommand Call Reject callerJID: thqrasdqs4wzdnch8kcbiw@xmpp.zoom.us
OK
*r CallAcceptRejectResult (status=OK):
** end
```

**callerJID**: The callerJID from the IncomingCallIndication event.

## zCommand Invite

Create a new PMI meeting, and invite these users, using the JID of each user from the phonebook. The Zoom Room will book an instant meeting, using the duration that you specify. The external Zoom Rooms Scheduling Display panel (if hooked up) will show this duration.

Schema:

```
zCommand Invite Duration: <int> user: <string> user: <string> ...
```

Example:

```
zcommand invite duration: 30 user: thqrasdqs4wzdnch8kcbiw@xmpp.zoom.us
OK
*r InviteResult (status=OK):
** end

*s Call Status: CONNECTING_MEETING
** end
...
 *r InfoResult Info ...
...
*s Call Status: IN_MEETING
** end
...
*r ListParticipantsResult ...
...
```

**Duration**: The intended duration of the meeting, in minutes. The Zoom Room will automatically book an instant meeting for this duration, and update it via calendar integration.

**user**: The jid (join ID) user ID to invite. Use **zCommand Phonebook List** to get a list of Phonebook entries, and use the jid entry for a user.

## zCommand Bookings List

Return the list of meetings scheduled for today, between 12:01 AM and midnight. It is not possible to retrieve scheduled meetings for other days.

Schema:

```
zCommand Bookings List
OK
*r BookingsListResult (status=OK):
*r BookingsListResult resultInfo TotalRows: N
*r BookingsListResult Meeting n meetingName: <string> // Name of the
meeting. If a meeting is set to "private", it is the responsibility of
the CLI user to not show the meeting name.
*r BookingsListResult Meeting n meetingNumber: <int> // meeting number
*r BookingsListResult Meeting n hostName: // RESERVED for future use
*r BookingsListResult Meeting n startTime: <ISO 8601 date> // Meeting
start time
*r BookingsListResult Meeting n endTime: <ISO 8601 date> // Meeting end
time
*r BookingsListResult Meeting n creatorEmail: <string> // Creator of the
Google Calendar item.  This parameter might be empty.
*r BookingsListResult Meeting n creatorName: // Creator of the Google
Calendar item.  This parameter might be empty.
*r BookingsListResult Meeting n isPrivate: <on/off> // On if meeting is
private. If private, then the end system should show the name of the
```

```
meeting using the CreatorName; for instance, "Bob's meeting." If the
creatorName is not available, use the creatorEmail.
*r BookingsListResult Meeting n checkedIn: <1/0> //Whether meeting has
been checked in
*r BookingsListResult Meeting n isInstantMeeting: <on/off>
*r BookingsListResult Meeting n calendarID: <string> //ID on calender
*r BookingsListResult Meeting n calendarChangeKey: <string>
*r BookingsListResult Meeting n accessRole: <string>
*r BookingsListResult Meeting n location: <string>
*r BookingsListResult Meeting n scheduledFrom: <string> //How the meeting
was scheduled
*r BookingsListResult Meeting n thirdparty service_provider: <string>
*r BookingsListResult Meeting n thirdparty meeting_number: <string>
*r BookingsListResult Meeting n thirdparty sip_address: <string>
*r BookingsListResult Meeting n thirdparty h323_address: <string>
...
** end
```

Example:

```
zcommand bookings list
OK
*r BookingsListResult (status=OK):
*r BookingsListResult resultInfo TotalRows: 2
*r BookingsListResult Meeting 1 meetingName: Instant Meeting
*r BookingsListResult Meeting 1 meetingNumber: 6225245297
*r BookingsListResult Meeting 1 hostName:
*r BookingsListResult Meeting 1 startTime: 2019-01-04T01:50:32Z
*r BookingsListResult Meeting 1 endTime: 2019-01-04T01:59:32Z
*r BookingsListResult Meeting 1 creatorEmail: room@kanovia.com
*r BookingsListResult Meeting 1 creatorName:
*r BookingsListResult Meeting 1 isPrivate: off
*r BookingsListResult Meeting 1 checkedIn: on
*r BookingsListResult Meeting 1 isInstantMeeting: on
*r BookingsListResult Meeting 1 calendarID: onrccc4oufoo2q68r15hphtr94
*r BookingsListResult Meeting 1 calendarChangeKey:
*r BookingsListResult Meeting 1 accessRole: owner
*r BookingsListResult Meeting 1 location: ZachMacMini, HQ-1-E-TestRoom
(8)
*r BookingsListResult Meeting 1 scheduledFrom: InstantMeeting
*r BookingsListResult Meeting 1 thirdparty service_provider: -1
*r BookingsListResult Meeting 1 thirdparty meeting_number:
*r BookingsListResult Meeting 1 thirdparty sip_address:
*r BookingsListResult Meeting 1 thirdparty h323_address:
*r BookingsListResult Meeting 2 meetingName: New Meeting
*r BookingsListResult Meeting 2 meetingNumber: 204690242
*r BookingsListResult Meeting 2 hostName:
*r BookingsListResult Meeting 2 startTime: 2019-01-04T02:30:00Z
```

```
*r BookingsListResult Meeting 2 endTime: 2019-01-04T03:00:00Z
*r BookingsListResult Meeting 2 creatorEmail: room@kanovia.com
*r BookingsListResult Meeting 2 creatorName:
*r BookingsListResult Meeting 2 isPrivate: off
*r BookingsListResult Meeting 2 checkedIn: off
*r BookingsListResult Meeting 2 isInstantMeeting: off
*r BookingsListResult Meeting 2 calendarID: 83f18p0uh083tpcf78vu25in8k
*r BookingsListResult Meeting 2 calendarChangeKey:
*r BookingsListResult Meeting 2 accessRole: owner
*r BookingsListResult Meeting 2 location: ZachMacMini, HQ-1-E-TestRoom
(8)
*r BookingsListResult Meeting 2 scheduledFrom: ZRP
*r BookingsListResult Meeting 2 thirdparty service_provider: -1
*r BookingsListResult Meeting 2 thirdparty meeting_number:
*r BookingsListResult Meeting 2 thirdparty sip_address:
*r BookingsListResult Meeting 2 thirdparty h323_address:
** end
```

## zCommand Phonebook List

Get a list of phonebook entries, starting with **Offset**, and retrieve up to **Limit** entries.

**Note:** The avatarURL field of a contact will be empty until the Zoom Room has been in a meeting with that contact, once the Zoom Room has been in a meeting with that participant, the Zoom Room will cache the avatarURL of that participant. The cache is located here:

**Mac:** /Users/xxx/Library/Application Support/ZoomPresence

**Windows:** %APPDATA%\ZoomRooms\data

Schema:

```
zCommand Phonebook List Offset: <int> Limit: <int>
OK
*r PhonebookListResult (status=OK):
*r PhonebookListResult resultInfo Offset: <int> // original offset passed
*r PhonebookListResult resultInfo Limit: <int> // original limit passed
*r PhonebookListResult resultInfo TotalRows: N
*r PhonebookListResult Contact n jid: <string> // the Join Id of the
contact.
Use this ID to invite a person to a meeting.
*r PhonebookListResult Contact n screenName: <string> // the Zoom IM
ScreenName of the contact
*r PhonebookListResult Contact n firstName: <string> // the First name of
the contact.
*r PhonebookListResult Contact n lastName: <string> // the Last name of
the contact.
*r PhonebookListResult Contact n phoneNumber: <string> // the phone
number of the contact
```

```
*r PhonebookListResult Contact n email: <string> // the email of the
contact
*r PhonebookListResult Contact n avatarURL: <string> // a URL pointing to
the avatar of the contact, if available
*r PhonebookListResult Contact n presence: <PRESENCE_OFFLINE,
PRESENCE_ONLINE, PRESENCE_AWAY, PRESENCE_BUSY, PRESENCE_DND>
*r PhonebookListResult Contact n onDesktop: <bool> // Set to on if the
user is currently using a desktop Zoom client. Only valid if the presence
status is PRESENCE_ONLINE.
*r PhonebookListResult Contact n onMobile: <bool> // Set to on if the
user is currently using a mobile Zoom client. Only valid if the presence
status is PRESENCE_ONLINE.
*r PhonebookListResult Contact n isZoomRoom: <on/off> // Set to on if the
user is a zoom room.
*r PhonebookListResult Contact n isLegacy: <on/off> // Set to on if the
user is currently using a legacy platform (EX. SIP/H323).
*r PhonebookListResult Contact n presence_status: <int>
*r PhonebookListResult Contact n sip_phone_number: <string> //
sip_phone_number if applicable
*r PhonebookListResult Contact n cloud_pbx_info isValid: <on/off> //
Whether cloud_pbx is on/off
*r PhonebookListResult Contact n cloud_pbx_info extension: <string> //
Only if cloud_pbx_info isValid is on
*r PhonebookListResult Contact n cloud_pbx_info company_number:
<string> //Only if cloud_pbx_info isValid is on
*r PhonebookListResult Contact n cloud_pbx_info direct_number_list: m
direct_number: <string> //Only if cloud_pbx_info isValid is on
...
** end
```

Example:

```
zCommand Phonebook List Offset: 0 Limit: 1
OK
*r PhonebookListResult (status=OK):
*r PhonebookListResult resultInfo Offset: 0
*r PhonebookListResult resultInfo Limit: 5
*r PhonebookListResult resultInfo TotalRows: 5
*r PhonebookListResult Contact 1 jid: x1ozr9jbrew8ahgvsmkzww@xmpp.zoom.us
*r PhonebookListResult Contact 1 screenName: Alex Marmer
*r PhonebookListResult Contact 1 firstName: Alex
*r PhonebookListResult Contact 1 lastName: Marmer
*r PhonebookListResult Contact 1 phoneNumber:
*r PhonebookListResult Contact 1 email:
*r PhonebookListResult Contact 1 avatarURL:
*r PhonebookListResult Contact 1 presence: PRESENCE_OFFLINE
*r PhonebookListResult Contact 1 onDesktop: off
*r PhonebookListResult Contact 1 onMobile: off
```

```
*r PhonebookListResult Contact 1 isZoomRoom: off
*r PhonebookListResult Contact 1 isLegacy: off
*r PhonebookListResult Contact 1 presence_status: 0
*r PhonebookListResult Contact 1 sip_phone_number:
*r PhonebookListResult Contact 1 cloud_pbx_info isValid: off
** end
```

**Offset**: Offset into the phonebook list (0 = no offset)

**Limit**: number of entries to return, maximum.

If a contact is a Zoom Room, then the convention is to refer to the Zoom Room as “Available” if it has a status if **PRESENCE_ONLINE**.

## zCommand Run

Run a test script. The test script must be in the proper directory:

- Mac: ~/Library/Logs/zoom.us/zaapi
- Windows: %APPDATA%/ZoomRooms/logs Provide the name of the file, without the path.

Schema:

```
zCommand Run File: <string>
```

Example:

```
zCommand Run File: tst2
```

**File**: Name of the script file to run.

## zCommand Comment

Used in scripts only: Issue a comment, to be echoed back on the CLI.

Schema:

```
zCommand Comment text: <string>
```

Example:

```
zCommand Comment Text: Try this test.
```

**Text**: Text of the comment.

## zCommand Wait

Wait for a given number of seconds. Can be used only in script files. It is used to wait for a reply to come back before executing the next command in the script file, to ensure that commands and replies are displayed in order, to make test verification easier.

Schema:

```
zCommand Wait Sec: <int>
```

Example

```
zCommand Wait Sec: 4
```

**Sec**: Number of seconds to delay

## zCommand Call Leave - In-Meeting

Leave the meeting, without ending the meeting for everyone: applies if the Zoom Room is a host or non-host.

Schema:

```
zCommand Call Leave
```

Example:

```
zcommand call leave
OK
*r CallDisconnectResult (status=OK):
** end
...
*e CallDisconnect success : on
** end
...
*s Call Status: NOT_IN_MEETING
** end
```

## zCommand Call Invite - In-Meeting

Invite a contact to a meeting, while in the meeting.

Schema

```
zCommand Call invite user: <string> user: <string> ...
```

Example

```
zcommand call invite user: thqrasdqs4wzdnch8kcbiw@xmpp.zoom.us
OK
*r InviteCallResult (status=OK):
** end
```

**user**: jID of user to invite. After the user is invited to the call, the presence status of the user will change to **PRESENCE_BUSY**.

## zCommand Call InviteH323Room - In-Meeting

Invite a H.323 Room to the meeting.

**Address**: Address of the H.323 endpoint. **Cancel**: **On** to cancel a sent request, **Off** to send a new request. Schema

```
zCommand Call InviteH323Room Address: <string> cancel: <on/off>
```

Example

```
zCommand Call InviteH323Room Address: xxx@xxx.xx.xxx.xxx Cancel: Off
OK
*r CallInviteH323Room (status=OK):
** end

*e H323SIPRoomCallingStatus status: CallingStatus_Accepted
** end

*e H323SIPRoomCallingUserInfo user_id: 16779265
*e H323SIPRoomCallingUserInfo user_name: Name
** end
```

The command returns a common response, then when the foreign room responds, two notifications are returned. The call status, and the call info. Since they are separate notifications, and the status notification does not have any identification information, it is recommended to only send 1 invitation at a time and limit the scope of when you are listening to the status notifications. That way you can easily match a status notification to an info notification.

If you don't get a reply from the room in a given amount of time, the status notification will return a timeout with the value **CallingStatus_TimeOut**.

## zCommand Call InviteSipRoom - In-Meeting

Invite a SIP Room to the meeting.

**Address**: Address of the SIP endpoint. **Cancel**: **On** to cancel a sent request, **Off** to send a new request. Schema

```
zCommand Call InviteSipRoom Address: <string> cancel: <on/off>
```

Example

```
zCommand Call InviteSipRoom Address: xxx@xxx.xx.xxx.xxx Cancel: Off
OK
*r CallInviteSipRoom (status=OK):
** end

*e H323SIPRoomCallingStatus status: CallingStatus_Accepted
** end

*e H323SIPRoomCallingUserInfo user_id: 16779265
*e H323SIPRoomCallingUserInfo user_name: Name
** end
```

The command returns a common response, then when the foreign room responds, two notifications are returned. The call status, and the call info. Since they are separate notifications, and the status notification does not have any identification information, it is recommended to only send 1 invitation at a time and limit the scope of when you are listening to the status notifications. That way you can easily match a status notification to an info notification.

If you don't get a reply from the room in a given amount of time, the status notification will return a timeout with the value **CallingStatus_TimeOut**.

## zCommand Call MuteParticipantVideo - In-Meeting

If the Zoom Room is the meeting host, mute or unmute the video of a meeting participant. The remote client will always ask the user for permission to unmute video, but not to mute video.

Schema

```
zCommand Call MuteParticipantVideo Mute: <bool> Id: <int32>
```

Example

```
zcommand call muteparticipantvideo mute: off id: 33555457
OK
*r CallMuteParticipantVideoResult (status=OK):
** end
```

If the participant agrees to unmute video, you get a notification:

```
...
*r ListParticipantsResult video_status is_sending: on
...
```

**Mute**: Set to On to mute the participant. Set to Off to unmute.

**Id**: Participant ID. Retrive the participant ID from the user_id parameter returned from **zCommand Call ListParticipants**.

## zCommand Bookings update

Download the current meeting list for this Zoom Room, for all meetings starting today, between 12:00 AM and midnight. Users may use this command to manually force a meeting list download after a change to the meetings.

Schema:

```
zCommand Bookings Update
```

Example:

```
zCommand bookings update
OK
*r BookingsUpdateResult (status=OK):
** end
```

If there were new bookings to update, you get a zEvent notification:

```
*e Bookings Updated
** end
```

## zCommand Dial Sharing

Start a presentation-only meeting. Corresponds to the ZRC Presentation Tab -> Desktop, iPhone/iPad/Mac buttons:

Schema:

```
zCommand Dial Sharing Duration: <int> Displaystate: <None | Laptop | IOS>
Password: <string>
OK
*r DialSharingResult (status=OK):
** end
```

Example:

```
zcommand dial sharing duration: 30 displaystate: Laptop password:
OK
*r DialSharingResult (status=OK):
** end

*e StartLocalPresentMeeting success: on
** end

*s Call Status: CONNECTING_MEETING
** end
...
*s Call Status: IN_MEETING
** end
```

**Duration**: After presentation mode starts, the Zoom Room will automatically book a meeting for the zoom Room, and on the integrated calendar, for this many minutes. BUG: The Zoom Room currently does not book a meeting, when using any combination of ZR / ZRC / ZR-CSAPI. It is likely that this bug will not be fixed; see the bug list at the beginning of this doc. **Displaystate**: Display instructions for sharing with a Laptop or IOS, or None to display no instructions. **Password**: Enter only white space after the : to specify no password, otherwise enter a string.

## zCommand Call ShareCamera - In-Meeting

Start sharing with the camera. This command only works in a regular meeting, not a sharing meeting. Corresponds to the “Share content or Camera” button on the ZRC. First, invoke: **zStatus video camera line** To get a list of camera devices. Then use the Id from a device to choose that device for sharing.

schema

```
zCommand Call ShareCamera id: <string> Status: <on/off>
```

Example 1:

```
zCommand Call ShareCamera id: 0x14200000046d082d Status: on
OK
*r ShareCameraResult (status=OK):
** end
...
*e SharingState state: Sending
*e SharingState paused: off
** end
...
*s CameraShare is_Sharing: on
```

```
*s CameraShare id: 0x14200000046d082d
*s CameraShare can_Control_Camera: on
*s CameraShare is_Mirrored: off
** end
```

Example 2: Error scenario, such as if another participant is already sharing:

```
zCommand Call ShareCamera id: CamTwist Status: on
OK

*e CallConnectError error_code: 0
*e CallConnectError error_message: Unable to start camera sharing. Try
again later.
** end
```

## zCommand Call SetInstructions

Change sharing instructions. After sharing is launched, and the Zoom Room is waiting for a participant to start sharing, this command provides the ability to change the instructions that are shown on the ZR:

Schema

```
zCommand Call SetInstructions Show: on Type: <Laptop | IOS | None>
```

The OK response is “CallSharingToggle” which is also used for **zCommand Call Sharing ToNormal**.

Example:

```
zcommand call setinstructions show: on Type: IOS
OK
*r CallSharingToggle (status=OK):
** end
```

## zCommand Call Sharing ToNormal - In-Meeting

While actively sharing using the sharing key, convert the meeting to a normal meeting.

Schema

```
zCommand Call Sharing ToNormal
```

The OK response is “CallSharingToggle” which is also used for **zCommand Call SetInstructions**.

Example:

```
zcommand call sharing tonormal
*e SharingState state: Sending
*e SharingState paused: off
** end

OK
*r CallSharingToggle (status=OK):
```

```
** end
```

Once you are in a normal meeting, to exit, you must use either **zCommand Call Disconnect**or **zCommand call leave**.

## zCommand Call Sharing Disconnect - In-Meeting

While actively sharing using the sharing key, use this command to stop the sharing, but stay in presentation mode, and go back to showing the sharing instructions:

Schema

```
zCommand Call Sharing Disconnect
```

Example:

```
zCommand Call Sharing Disconnect
OK
*r CallSharingDisconnect (status=OK)
** end
```

Invoking this command puts the ZR back into presentation-only mode, waiting for someone to start sharing. The instructions for either None, Laptop, or IOS will appear on the screen.

## zCommand Call Sharing HDMI Start

Start sharing the HDMI input. Zoom Rooms are often configured to automatically start sharing when an HDMI signal is connected to an HDMI capture device. However, if that automatic sharing is not enabled, use this command to start sharing the HDMI input. Be aware that only certain white-listed HDMI capture cards may be used with content sharing. See restriction here

Schema

```
zCommand Call Sharing HDMI Start
```

Example:

```
zCommand call sharing hdmi start
*s Sharing wifiName:
*s Sharing serverName: ZR-ScottFirestone2
*s Sharing password: 5182
*s Sharing isAirHostClientConnected: off
*s Sharing isBlackMagicConnected: on
*s Sharing isBlackMagicDataAvailable: on
*s Sharing isSharingBlackMagic: on
*s Sharing directPresentationPairingCode: 224425
*s Sharing directPresentationSharingKey: EWOOVF
*s Sharing isDirectPresentationConnected: off
*s Sharing dispState: Laptop
** end
OK

*r CallSharingHDMI (status=OK):
```

```
** end
```

## zCommand Call Sharing HDMI Stop

Stop sharing the HDMI input.

Schema

```
zCommand Call Sharing HDMI Stop
```

Example:

```
zCommand call sharing hdmi stop
*s Sharing wifiName:
*s Sharing serverName: ZR-ScottFirestone2
*s Sharing password: 5182
*s Sharing isAirHostClientConnected: off
*s Sharing isBlackMagicConnected: on
*s Sharing isBlackMagicDataAvailable: on
*s Sharing isSharingBlackMagic: off
*s Sharing directPresentationPairingCode: 224425
*s Sharing directPresentationSharingKey: EWOOVF
*s Sharing isDirectPresentationConnected: off
*s Sharing dispState: Laptop
** end
OK

*r CallSharingHDMI (status=OK):
** end
```

## zCommand Call Layout TurnPage - In-Meeting

In Gallery view, the zoom Room can show multiple participants at the same time, each in a small video thumbnail. The zoom Room can also show multiple pages of tiled thumbnails, and provides a way to page flip forward and back. In Strip view, up to about 8 video thumbnails are shown at the bottom, and the user can flip forward and back to different sets of 8 videos.

Schema:

```
zCommand Call Layout TurnPage Forward: <bool>
```

Example:

```
zCommand Call Layout TurnPage Forward: On
OK

*r TurnPageResult (status=OK)
** end
```

**Forward**: Set to **On**, to flip forward. Set to **Off** to flip backward.

## zCommand Call Expel - In-Meeting

Corresponds to the ZRC per-user “Remove” button. If the Zoom Room is a host, the Zoom Room can expel participants. Once a participant is expelled, that participant cannot re-join the meeting, unless the meeting is ended for all and re-launched. First invoke **zCommand Call ListParticipants** To get a list of participants, which will include the Id for each participant. Then use the Id to expel a user.

Schema:

```
zCommand Call Expel Id: <int32>
```

Example:

```
zCommand Call Expel Id: 16780288
OK

*r ExpelResult (status=OK):
** end
```

## zCommand Test Microphone Start

Start Testing Microphone. Corresponds to launching the Microphone settings dialog, which shows the microphone volume level in real time as a volume indicator. When this mode is activated, the API will send a message as often as once every 1/2 second, with an updated volume level from the microphone, if the microphone volume level has changed. First, invoke **zStatus audio input line** To get a list of microphone devices. Then use the id from a device to choose that device for the volume check. The device id can have spaces.

Schema:

```
zCommand Test Microphone Start Id: <string>
```

Example:

```
zCommand Test Microphone Start Id: Sennheiser SC70 USB CTRL
...
*e TestMicrophone Volume: <int>
** end
```

## zCommand Test Microphone Stop

Stop Testing the Microphone

Schema:

```
zCommand Test Microphone Stop
```

Example:

```
zCommand Test Microphone Stop
OK

*r TestMicrophoneResult (status=OK):
** end
```

## zCommand Test Speaker Start

Start Testing the Speaker. Corresponds to the “Test Speaker” link on the ZRC, under Settings. First, invoke **zStatus audio output line** To get a list of speaker devices. Then use the id from a device to choose that device for the volume check. The device id can have spaces. The speaker test will start: the test has a fixed duration of a few seconds, but you can end it sooner using the stop command.

Schema:

```
zCommand Test Speaker Start Id: <string>
```

Example:

```
zCommand Test Speaker Start Id: Sennheiser SC70 USB CTRL
OK
*r TestSpeakerResult (status=OK):
** end
...
*e TestSpeakerFinished Volume: <int>
*e TestSpeakerFinished Ended: <bool>
```

## zCommand Test Speaker Stop

Stop Testing Speaker. You do not get additional notifications if you end the test using this method.

Schema:

```
zCommand Test Speaker Stop
```

Example:

```
zCommand Test Speaker Stop
OK
*r TestSpeakerResult (status=OK):
** end
```

## zCommand Call HostChange - In-Meeting

Change Host. When a meeting is scheduled for a Zoom Room, the first person to join will have host status.

If a participant (including a Zoom Room) has host status, it’s possible for that participant to pass host status to any another participant.

If the Zoom Room is currently the host, this command re-assigns the host of the meeting to a different participant. First check to verify that you have host status, by invoking:**zCommand Call ListParticipants** and check to see if the participant with **is_myself**set to **On** also has **is_host** set to **On**. Then look for the participant Id of the person who you want to be the host.

**Note**: Once a participant is host, the Zoom Room no longer tracks its **can_record** and **is_recording** status.

Schema:

```
zCommand Call HostChange Id: <int>
```

Example:

```
zCommand Call HostChange id: 33555457
OK
*r CallHostChange (status=OK):
** end
```

This command will trigger a participant List notification message, with updated values for all participants, including an update **is_host** parameter.

## zCommand Call HostClaim - In-Meeting

Any participant without host status can try to claim host status. In order to claim host status, the participant must enter either:

- The host key of the Zoom Room
- The host key of the person who scheduled the meeting

You can check to see if you have host status, by invoking: **zCommand Call ListParticipants**and check to see if the participant with **is_myself** set to **On** also has **is_host** set to **On**.

If the Zoom Room is currently not the host, this command allows the Zoom Room to attempt to claim host status.

**Note**: Once a participant is host, the Zoom Room no longer tracks its **can_record** and **is_recording** status.

Schema:

```
zCommand Call HostClaim Key: <int>
```

Example:

```
zCommand Call HostClaim Key: 620501
OK
*r CallHostClaim (status=OK):
** end
...
*e CallHostClaim result: Success
** end
```

Followed by a participant list update notification. Or, if you entered the wrong host key, you get an error notification:

```
*e CallHostClaim result: InvalidHostKey
** end
```

## zCommand Call Record - In-Meeting

If the Zoom Room has recording privilage (either by being host or being granted privilage by host), this command will start recording the meeting.

*Note: Currently, we only support recording of meetings scheduled through calendar apps such as Google Calendar. Instant Meeting recording and recording of meetings scheduled through ZRC*

*itself is currently disabled due to needing the user to enter an email address. However, for those meetings, the Zoom Room can still give permission for other participants to record locally*

***Bug***: ***zCommand Call ListParticipants***'s **can_record** and **is_recording** might be wrong for the Zoom Room itself (is right for non-host participants). It is recommended to not rely on the participant list's **can_record** and **is_recording** for the Zoom Room itself. Instead, use the **UpdatedCallRecordInfo zEvent** documented below and in the **zEvent** section.

Schema:

```
zCommand Call Record Enable: <on/off>
```

Example (successful response):

```
zCommand Call Record Enable: on
OK
*r CallRecord (status=OK):
** end
```

Example (no recording privilege)

```
zCommand Call Record Enable: on
*r CallRecord(status=Error):
*r Result reason: "No recording privilege"
** end
ERROR
```

Example (need email address/instant meeting)

```
zCommand Call Record Enable: on
*r CallRecord(status=Error):
*r Result reason: "Need email address"
** end
ERROR
```

You will get this notification whenever the Zoom Room's recording privilege is updated:

```
*e UpdateCallRecordInfo canRecord: true
*e UpdateCallRecordInfo emailRequired: true
*e UpdateCallRecordInfo amIRecording: false
*e UpdateCallRecordInfo meetingIsBeingRecorded: false
** end
```

## zCommand Call Pin - In-Meeting

Allows the Zoom Room to pin participants to a screen. The Zoom Room does not have to be a host to do this.

You can see meeting participants by invoking: **zCommand Call ListParticipants**

Schema:

```
Turning on: zCommand Call Pin Id: <int> Enable: On Screen: <int>
```

```
For the screen input, the integer can between 0-3 depending on how many
screens your Zoom Room has.
```

```
Turning off: zCommand Call Pin Id: <int> Enable: Off
```

Example:

```
zCommand Call Pin Id: 33555457 Enable: On Screen: 0
OK
*r PinVideoOnScreen (status=OK):
** end
```

You will get this notification whenever the pin status of one of the Zoom Room is updated:

```
In this example the user 33555457 is pinned to screen 0:

*e PinStatusOfScreenNotification screen_index: 0
*e PinStatusOfScreenNotification can_be_pinned: true
*e PinStatusOfScreenNotification pinned_user_id: 33555457
*e PinStatusOfScreenNotification screen_layout: PinnedVideo
*e PinStatusOfScreenNotification pinned_share_source_id: -1
*e PinStatusOfScreenNotification share_source_type: None
*e PinStatusOfScreenNotification can_pin_share: false
*e PinStatusOfScreenNotification why_cannot_pin_share: None
** end
In this example, nobody is pinned to screen 0:

*e PinStatusOfScreenNotification screen_index: 0
*e PinStatusOfScreenNotification can_be_pinned: true
*e PinStatusOfScreenNotification pinned_user_id: -1
*e PinStatusOfScreenNotification screen_layout: Gallery
*e PinStatusOfScreenNotification pinned_share_source_id: -1
*e PinStatusOfScreenNotification share_source_type: None
*e PinStatusOfScreenNotification can_pin_share: false
*e PinStatusOfScreenNotification why_cannot_pin_share: None
** end
```

## zCommand Call Spotlight - In-Meeting

The Zoom Room can spotlight a user, which means that the spotlightted user will be the primary active speaker for all participants. The requirements to do this are that the Zoom Room is a host and that there are at least 3 participants in the meeting.

You can see meeting participants by invoking: **zCommand Call ListParticipants**

Schema:

```
zCommand Call Spotlight Id: <int> Enable: <on/off>
```

Example:

```
zCommand Call Spotlight Id: 33555457 Enable: On
OK
*r SetSpotlight (status=OK):
```

```
** end
```

You will get this notification whenever the spotlight status of the Zoom Room is updated:

```
In this example the spotlight is on 33555456:

*e SpotlightStatusNotification user_id: 33555456
*e SpotlightStatusNotification present: on

In this example the spotlight is off:

*e SpotlightStatusNotification user_id: -1
*e SpotlightStatusNotification present: off
```

## zCommand Call AllowRecord - In-Meeting

If the Zoom Room is the host, it can allow other participants to record meetings.

You can see meeting participants by invoking: **zCommand Call ListParticipants**

**Note**: Once a participant is host, the Zoom Room no longer tracks its **can_record** and **is_recording** status in **zCommand Call ListParticipants**

Schema:

```
zCommand Call AllowRecord Id: <int> Enable: <On/Off>
```

Example:

```
zCommand Call Allowrecord Id: 33555457 Enable: On
OK
*r AllowRecord (status=OK):
** end
```

You will get a ListParticipantsResults notification, in that notification, check to make sure that the can_record field for the user you want is set to the value that you want:

```
zcommand call allowrecord id: 33555457 enable: on
*r ListParticipantsResult user_name: Scott Firestone Testing ZR
*r ListParticipantsResult user_id: 33555457
*r ListParticipantsResult is_host: off
*r ListParticipantsResult is_myself: off
*r ListParticipantsResult can_record: on
*r ListParticipantsResult is_recording: off
*r ListParticipantsResult avatar_url: https://zoom.us/p/
ThqraSdQS4WzDNCH8kCBiw/bc48fef7-66a7-49c3-a388-076df886b6e4-2983
*r ListParticipantsResult local_recording_disabled: off
*r ListParticipantsResult is_video_can_mute_byHost: on
*r ListParticipantsResult is_video_can_unmute_byHost: on
*r ListParticipantsResult isCohost: off
*r ListParticipantsResult user_type: NORMAL
*r ListParticipantsResult audio_status type: AUDIO_VOIP
*r ListParticipantsResult audio_status state: AUDIO_MUTED
```

```
*r ListParticipantsResult video_status has_source: on
*r ListParticipantsResult video_status is_sending: on
*r ListParticipantsResult video_status is_receiving: off
*r ListParticipantsResult camera_status can_i_request_control: on
*r ListParticipantsResult camera_status am_i_controlling: off
*r ListParticipantsResult camera_status can_switch_camera: on
*r ListParticipantsResult camera_status can_move_camera: off
*r ListParticipantsResult camera_status can_zoom_camera: off
*r ListParticipantsResult event: ZRCUserChangedEventUserInfoUpdated
** end
```

## zCommand Call CameraControl - In-Meeting

The Zoom Room can control its own camera, as well as controlling remote camera (far end camera control).

To control a remote camera, both the Zoom Room and the participant must have far end camera control settings turned on: See here to learn more

You can see meeting participants by invoking: **zCommand Call ListParticipants**

Schema:

```
There are two versions of the commands, their usage will be explained.

zCommand Call CameraControl Id: <int> State: <Start|Continue|Stop|
RequestRemote|GiveupRemote|RequestedByFarEnd> Action: <Left|Right|Up|
Down|In|Out>

zCommand Call CameraControl Id: <int> Speed: <int>
```

For both versions of the commands:

```
Id: 0 to control own camera, Id of participant to control remote camera
```

For first version of command:

```
State: Describes what you intend to do. For example, start a new movement
or maintain an existing movement.

Action: How you want your state executed. For example, move left, or
maintain left.

Below are the list of States:
Start: Start a new movement
Continue: Continue an existing movement
Stop: Stop an existing movement
RequestRemote: Request far end camera for control
GiveupRemote: Give up control of far end camera
RequestedByFarEnd: Let participant control Zoom Room's camera
```

```
Below are the list of Actions:
Left: Move camera left
Right: Move camera right
Up: Move camera up
Down: Move camera down
In: Zoom camera in
Out: Zoom camera out
```

For second version of command:

```
Speed: Speed of the camera while controlling (1-100)
```

The second version of the command is used to control the speed of some models of cameras, however, not all camera models support speed control. In fact, the majority of cameras do not.

The first version of the command is used for the majority of camera control functionalities. Note that there are two ways to use the second version of this command. The way that it is used will depend on the model of camera you are trying to control. For some camera models (referred to as type 1 from now on), issuing a start move command will keep moving the camera until the camera's limit is reached or a stop command is issued. For other models (referred to as type 2 from now on), issuing a move command will move the camera a little bit then wait for a continue command.

Below is a sequence of commands to turn a type 1 camera to the left:

```
State: Start, Action: Left
Then when camera is in position that you want:
State: Stop, Action: Left
```

Below is a sequence of commands to turn a type 2 camera to the left:

```
State: Start, Action: Left
Wait 10ms
State: Continue, Action: Left
Wait 10ms
State: Continue, Action: Left
Wait 10ms
State: Continue, Action: Left
Once camera is in position that you want, stop issuing continue.
Note: If you wait more than 10ms between Continue or Start, you have to
issue Start again.
```

As you can see the two types of cameras are controlled radically differently. Type 1 uses **Start State** and **Stop State** commands while type 1 uses **Start State** and **Continue State**. So what should you do if you want your controls to work for both camera types? You have to issue both commands.

Below is an example of a sequence of commands that will turn both camera types to the left:

```
State: Start, Action: Left
Wait 10ms
State: Continue, Action: Left
Wait 10ms
State: Continue, Action: Left
Wait 10ms
State: Continue, Action: Left
```

```
Wait 10ms
State: Continue, Action: Left
State: Stop, Action: Left
```

```
The Stop command will stop type 1 cameras, while the Continue command
will continue to move type 2 cameras.
```

As for the other **States** such as **RequestRemote**, **GiveupRemote**, and **RequestedByFarEnd**, they are the same between type 1 and type 2 cameras. Also these **States** do not use a corresponding **Action**, however since the **zCommand** requires an action as an input no matter what, you can simply enter whatever **Action** you want when using these **States**, the **Action** will simply have no impact.

Example 1 Giving far end control of camera:

```
Note: In this case, the Action can be anything, it will not affect the
command
```

```
zCommand Call CameraControl Id: 16778240 State: RequestedByFarEnd Action:
Left
OK
*r CameraControl (status=OK):
** end
```

Example 2 Controlling Zoom Room's own camera type 1:

```
Note: Id of 0 means own camera
```

```
zCommand Call CameraControl Id: 0 State: Start Action: Left
OK
*r CameraControl (status=OK):
** end
```

```
Then when camera is in position we want:
```

```
zCommand Call CameraControl Id: 0 State: Stop Action: Left
OK
*r CameraControl (status=OK):
** end
```

Example 3 Controlling far end camera type 2:

```
zCommand Call CameraControl Id: 16778240 State: Start Action: Left
OK
*r CameraControl (status=OK):
** end
```

```
In 10ms:
```

```
zCommand Call CameraControl Id: 16778240 State: Continue Action: Left
OK
```

```
*r CameraControl (status=OK):
** end
```

```
Now camera in position that we want, stop issuing continue.
```

Example 4 Controlling far end camera both types:

```
zCommand Call CameraControl Id: 16778240 State: Start Action: Left
OK
*r CameraControl (status=OK):
** end
```

```
In 10ms:
```

```
zCommand Call CameraControl Id: 16778240 State: Continue Action: Left
OK
*r CameraControl (status=OK):
** end
```

```
When camera in position we want:
```

```
zCommand Call CameraControl Id: 16778240 State: Stop Action: Left
OK
*r CameraControl (status=OK):
** end
```

When a far end participant requests to control the Zoom Room's camera, this notification will be sent:

```
*e CameraControlNotification user_name: Zach
*e CameraControlNotification user_id: 16778241
*e CameraControlNotification state:
ZRCCameraControlStateRequestedByFarEnd
** end
```

When a far end participant gives up the Zoom Room's camera controls, this notification will be sent:

```
*e CameraControlNotification user_name: Zach
*e CameraControlNotification user_id: 16778241
*e CameraControlNotification state: ZRCCameraControlStateGaveUpByFarEnd
** end
```

# zCommand Dial CheckIn

Allows the user to check in to a meeting without actually joining. If check in feature is enabled in the web portal, this command will prevent the Zoom Room from releasing a meeting. You can see meetings currently booked by invoking: **zCommand Bookings List**

This command will return a common response, then if check in was successful, a ***Bookings Updated*** event notification will also be returned.

Schema:

```
zCommand Dial CheckIn MeetingNumber: <string>
```

Example:

```
zCommand Dial CheckIn MeetingNumber: 890741258
OK
*r CheckIn (status=OK):
** end

*e Bookings Updated
** end
```

## zCommand Schedule Add

Schedule a meeting.

This command will return a common response, then if schedule was successful, a ***Bookings Updated*** event notification will also be returned.

Schema:

```
zCommand Schedule Add MeetingName: <string> Start: <string> End: <string>
private: <on/off>
```

***MeetingName***: Name of meeting

***Start***: Start time in YYYYMMDDHHMM format, date must before ***End***, HHMM are in 24 hour clock

***End***: End time in YYYYMMDDHHMM format, date must be after ***Start***, HHMM are in 24 hour clock

***Private***: Is meeting private <on/off>

Example:

```
zCommand Schedule Add MeetingName: Example Start: 201812061800 End:
201812061830 Private: off
OK
*r ScheduleAdd (status=OK):
** end

*e Bookings Updated
** end
```

Example (Invalid date, scheduled for 32nd day of December)

```
zCommand Schedule Add MeetingName: Example Start: 201812321800 End:
201812321830 Private: off
*r ScheduleAdd(status=Error):
*r Result reason: "Invalid date"
** end
ERROR
```

Example (Invalid time, start time is after end time)

```
zCommand Schedule Add MeetingName: Example Start: 201812061830 End:
201812061800 Private: off
*r ScheduleAdd(status=Error):
*r Result reason: "Invalid time"
** end
ERROR
```

## zCommand Schedule Delete

Delete a booked meeting. You can see meetings currently booked by invoking: **zCommand Bookings List**

This command will return a common response, then if check in was successful, a ***Bookings Updated*** event notification will also be returned.

Schema:

```
zCommand Schedule Delete MeetingNumber: <string>
```

Example:

```
zCommand Schedule Delete MeetingNumber: 890741258
OK
*r ScheduleDelete (status=OK):
** end

*e Bookings Updated
** end
```

## zCommand Dial PhoneCallOut

Call a telephone using Zoom Room. You must first enable this feature by following the instructions here

This command will return a common response, then if successful, an event displaying the phone call will be returned.

*Please note that at this time, the Zoom Room can only support one phone call dial out at a time. Trying to call out to multiple calls at a time may lead to undocumented behavior.*

Schema:

```
zCommand Dial PhoneCallOut Number: <string>
```

Example:

```
zCommand Dial PhoneCallOut Number: 5555555555
OK
*r DialPhoneCallOut (status=OK):
** end
```

```
*e PhoneCallStatus callID: oCo7mCnH2o8fEIxl5A-h9w..
1fb31480@16692714630.zoom.us
*e PhoneCallStatus peerDisplayName:
*e PhoneCallStatus peerNumber: 5555555555
*e PhoneCallStatus peerUri: 5555555555
*e PhoneCallStatus isIncomingCall: false
*e PhoneCallStatus status: PhoneCallStatus_Ringing
** end
```

## zCommand Dial PhoneHangUp

Hang up a telephone call. You must first enable this feature by following the instructions here

This command will return a common response, then if successful, an event displaying the phone call that was terminated will be returned.

Please note that at this time, the Zoom Room can only support one phone call dial out at a time.

The **CallId** can be gotten by calling **zCommand PhoneCall List**.

Schema:

```
zCommand Dial PhoneHangUp CallId: <string>
```

Example:

```
zCommand Dial PhoneHangUp CallId: oCo7mCnH2o8fEIxl5A-h9w..
1fb31480@16692714630.zoom.us
OK
*r DialPhoneHangUp (status=OK):
** end

*e PhoneCallTerminated callID: oCo7mCnH2o8fEIxl5A-h9w..
1fb31480@16692714630.zoom.us
*e PhoneCallTerminated peerDisplayName:
*e PhoneCallTerminated peerNumber: 5555555555
*e PhoneCallTerminated peerUri: 5555555555
*e PhoneCallTerminated isIncomingCall: false
*e PhoneCallTerminated status: PhoneCallTerminateReason_ByLocal
** end
```

## zCommand PhoneCall List

Displays the current list of telephone calls that the Zoom Room is in. Currently, the Zoom Room can only dial out to one call at a time, however, this command returns a list of calls to support future features.

Schema:

```
zCommand PhoneCall List
```

Example:

```
zCommand PhoneCall List
OK
*r PhoneCallListResult (status=OK):
*r PhoneCallListResult start
*r PhoneCallListResult Call 0 callID:  yKdds2buO8crJDrsYlLgdw..
1fb39430@16692714630.zoom.us
*r PhoneCallListResult Call 0 peerDisplayName:
*r PhoneCallListResult Call 0 peerNumber: 5555555555
*r PhoneCallListResult Call 0 peerUri: 5555555555
*r PhoneCallListResult Call 0 isIncomingCall: false
** end
```

## zCommand DTMF Support

**Syntax:** zCommand SendSIPDTMF CallID: <string> Key: <string>

```
zCommand SendSIPDTMF CallID : f26863c2-0c06-1238-0599-000c29324577 Key :
1
```

Description of the parameter:

For parameter CallID： The CallID is a unique ID representing the ongoing call, which should be found in the previous control system output. There are two situations.

1. When receiving an incoming sip call. The control system's output is as below:

```
*e PhoneCallStatus callID: f26863c2-0c06-1238-0599-000c29324577
*e PhoneCallStatus peerDisplayName: 4133
*e PhoneCallStatus peerNumber: 4133
*e PhoneCallStatus peerUri: 4133@10.100.56.198
*e PhoneCallStatus isIncomingCall: true
*e PhoneCallStatus status: PhoneCallStatus_Incoming

end
```

2. When making a call. The control system's input and output is as below:

```
input:zCommand Dial PhoneCallOut Number:4133
output:OK*r DialPhoneCallOut (status=OK):

end
*e PhoneCallStatus callID: 8LoH6D8QMYa4yye9IZtk_Q..53cdda0@4134
*e PhoneCallStatus peerDisplayName:
*e PhoneCallStatus peerNumber: 4133
*e PhoneCallStatus peerUri: 4133
*e PhoneCallStatus isIncomingCall: false
*e PhoneCallStatus status: PhoneCallStatus_Ringing

end
```

For parameter Key: Key is a character among 0-9 or * #

# zConfiguration

zConfigurations represent read/write configuration parameters: you can get the current value of the parameter, by specifying the parameter name but leaving out the value. Or, you can set the value by specifying the parameter name with a value.

## zConfiguration Call Sharing optimize_video_sharing - In-Meeting

Turn on or off video optimization for screen sharing.

Schema

```
zConfiguration Call Sharing optimize_video_sharing[: <on/off>]
```

Example

```
zConfiguration Call Sharing optimize_video_sharing
*c zConfiguration Call Sharing optimize_video_sharing: off
** end
OK
```

**optimize_video_sharing**: Set to **On** to optimize screen sharing when playing videos.

## zConfiguration Call Microphone Mute - In-Meeting

Mute/unmute the Zoom Room microphone.

Schema

```
zConfiguration Call Microphone Mute[: <on/off>]
```

Example

```
zConfiguration Call Microphone Mute
*c zConfiguration Call Microphone Mute: off
** end
OK
```

**Mute**: Set to **On** to mute the currently selected microphone

## zConfiguration Call Camera Mute - In-Meeting

Mute/unmute the Zoom Room camera.

Schema

```
zConfiguration Call Camera Mute[: <on/off>]
```

Example

```
zConfiguration Call Camera Mute
*c zConfiguration Call Camera Mute: off
** end
```

```
OK
```

**Mute**: Set to **On** to mute the currently selected camera

## zConfiguration Audio Input SelectedId

Get the ID of the currently selected audio input device. Or, switch to an audio input device by setting its ID here. Get the list of device IDs using **zStatus Audio Input Line**.

Schema

```
zConfiguration Audio Input selectedId[: <string>]
```

Example

```
zConfiguration Audio Input SelectedId
*c zConfiguration Audio Input selectedId: Sennheiser SC70 USB CTRL
** end
OK
```

**selectedId**: The Id of the device

## zConfiguration Audio Input is_sap_disabled

Turn on or off software audio processing (SAP)

Schema

```
zConfiguration Audio Input is_sap_disabled[: <on/off>]
```

Example

```
zConfiguration Audio Input is_sap_disabled
*c zConfiguration Audio Input is_sap_disabled: off
** end
OK
```

**is_sap_disabled**: Set to **On** to disable software audio processing (SAP), which includes automatic gain control (AGC) for the microphone and acoustic echo cancellation (AEC). You'll want to disable SAP if you have a speakerphone that implements echo cancellation.

## zConfiguration Audio Input reduce_reverb

Turn on or off reduce reverb feature.

Schema

```
zConfiguration Audio Input reduce_reverb[: <on/off>]
```

Example

```
zConfiguration Audio Input reduce_reverb
*c zConfiguration Audio Input reduce_reverb: off
** end
OK
```

**reduce_reverb**: Set to **On** to reduce echo coming from the Zoom Room. This feature will only work if **is_sap_disabled** is set to **Off**.

## zConfiguration Audio Input volume

Get/Set the volume for the currently selected microphone.

Schema

```
zConfiguration Audio Input volume[: <int>]
```

Example

```
zConfiguration Audio Input volume
*c zConfiguration Audio Input volume: 0
** end
OK
```

**volume**: Range is 0 to 100. Volume settings are stored per-device, and automatically change to a new value when the selected device changes, in which case the ZR-CSAPI issues a notification.

## zConfiguration Audio Output SelectedId

Get/Set the ID of the currently selected audio output device. Or, switch to an audio output device by setting its ID here. Get the list of device IDs using **zStatus Audio Output Line**.

Schema

```
zConfiguration Audio Output selectedId[: <string>]
```

Example

```
zConfiguration Audio Output selectedId
*c zConfiguration Audio Output selectedId: Sennheiser SC70 USB CTRL
** end
OK
```

**selectedId**: The Id of the device

## zConfiguration Audio Output volume

Get/Set the volume for the currently selected speaker.

Schema

```
zConfiguration Audio Output volume[: <int>]
```

Example

```
zConfiguration Audio Output volume
*c zConfiguration Audio Output volume: 60
** end
OK
```

**volume**: Range is 0 to 100. Volume settings are stored per-device, and automatically change to a new value when the selected device changes, in which case the ZR-CSAPI issues a notification.

## zConfiguration Video hide_conf_self_video

Show or hide the self view video while in a meeting.

Schema

```
zConfiguration Video hide_conf_self_video[: <on/off>]
```

Example

```
zConfiguration Video hide_conf_self_video
*c zConfiguration Video hide_conf_self_video: on
** end
OK
```

**hide_conf_self_video**: Set to **On** to hide the self-view video, which appears either full-screen (if no other video participants have joined), or in a thumbnail window in active speaker mode.

## zConfiguration Video Camera selectedId

Get/Set the ID of the currently selected camera. Or, switch to a camera by setting its ID here. Get the list of device IDs using **zStatus Camera Line**.

Schema

```
zConfiguration Video Camera selectedId[: <string>]
```

Example

```
zConfiguration Video Camera selectedId
*c zConfiguration Video Camera selectedId: CDC85FD0-E73A-4FC2-B3A8-
EA237D6990E0
** end
OK
```

**selectedId**: The Id of the device

## zConfiguration Video Camera Mirror

Get/Set Mirror mode for the ZR self view video window. This setting does not affect what other participants see.

Schema

```
zConfiguration Video Camera Mirror[: <on/off>]
```

Example

```
zConfiguration Video Camera Mirror
*c zConfiguration Video Camera Mirror: on
** end
OK
```

**Mirror**: Set to **On** to mirror the camera

## zConfiguration Client appVersion

Get/Set the appVersion value. This string will appear on the Web Portal under the ZR Devices list for this Zoom Room. It can be used to report the version of software running in an automation controller.

Schema

```
zConfiguration Client appVersion[: <string>]
```

Example

```
zConfiguration Client appVersion
*c zConfiguration Client appVersion: 1.0.9
** end
OK
```

**appVersion**: The app version text that will appear on the Zoom Web Portal

## zConfiguration Client deviceSystem

Get/Set the deviceSystem value. This string will appear on the Web Portal under the ZR Devices list for this Zoom Room. It can be used to report the type of automation controller connected to the ZR.

Schema

```
zConfiguration Client deviceSystem[: <string>]
```

Example

```
zConfiguration Client deviceSystem
*c zConfiguration Client deviceSystem: Command Line 2
** end
OK
```

**deviceSystem**: The system device text that will appear on the Zoom Web Portal

## zConfiguration Call Layout ShareThumb - In-Meeting

Swaps the screen share image to/from a thumbnail: Corresponds to the “Swap Content” button in the Layout options on the ZRC. Normally, when sharing screen content, the screen capture content is shown full-screen, and other videos are shown in a thumbnail. Use this command to put the sharing content into a thumbnail video.

Schema:

```
zConfiguration Call Layout ShareThumb[: <on/off>]
```

Example:

```
zConfiguration Call Layout ShareThumb
*c zConfiguration Call Layout ShareThumb: off
```

```
** end
OK
```

**ShareThumb**: If allowed by **can_Switch_Floating_Share_Content**, set to **On** to show sharing content in a thumbnail, instead of full-screen. If set to **Off**, shows sharing content full screen, which is the normal default operation. Applies when viewing the Speaker video layout, which shows a large video that takes the entire screen, plus one thumbnail showing other content.

## zConfiguration Call Layout Style - In-Meeting

Configure the layout options: Corresponds to the “Change view” button on the ZRC. Layout options include: **Gallery**, **Speaker**, **Strip**, or **ShareAll** to show sharing on all screens. First invoke **zStatus Call Layout** To determine which layouts are possible.

schema:

```
zConfiguration Call Layout Style[: <Gallery, Speaker, Strip, ShareAll>]
```

Example:

```
zConfiguration Call Layout Style
*c zConfiguration Call Layout Style: Speaker
** end
OK
```

**Style**: The style of video layout to show

## zConfiguration Call Layout Size - In-Meeting

Set size of the thumbnails: In Speaker mode only, change the size of the thumbnail video, which can either be a self-view video, or a remote view, or a sharing video.

Schema:

```
zConfiguration Call Layout Size[: <Off | Size1 | Size2 | Size3 | Strip>]
```

Example:

```
zConfiguration Call Layout Size
*c zConfiguration Call Layout Size: Size1
** end
OK
```

**Size**: For the Speaker view mode, allowed value are: **Off**, **Size1**, **Size2**, **Size3**. **Strip** is ignored. For filmstrip mode view, only the **Strip** value is allowed.

## zConfiguration Call Layout Position - In-Meeting

Set the position of the thumbnails: In Speaker mode only, change the position of the thumbnail video, which can either be a self-view video, a remote view, or a sharing view.

Schema:

```
zConfiguration Call Layout Position[: <Center | Up | Right | UpRight |
Down | DownRight | Left | UpLeft | DownLeft>]
```

Example:

```
zConfiguration Call Layout Position
*c zConfiguration Call Layout Position: UpRight
** end
OK
```

**Position**: For Speaker view, currently supported values include: UpRight, DownRight, UpLeft, DownLeft. Other values are reserved for future use.

## zConfiguration Call Lock Enable - In-Meeting

Corresponds to “Lock Meeting” button in the user management dialog of the iPad ZRC. After a meeting is locked, no new participants are allowed to join.

Schema:

```
zConfiguration Call Lock Enable[: <bool>]
```

Example:

```
zConfiguration Call Lock Enable
*c zConfiguration Call Lock Enable: off
** end
OK
```

**Enable**: Set to **On** to prevent other participants from joining the meeting

## zConfiguration Call MuteUserOnEntry Enable - In-Meeting

Turn on/off Mute on Entry: Corresponds to the “Mute Participants on Entry” button in the ZRC user management dialog. When enabled, the audio of participants who join the meeting will be automatically muted upon joining.

Schema:

```
zConfiguration Call MuteUSerOnEntry Enable[: <bool>]
```

Example:

```
zConfiguration Call MuteUSerOnEntry Enable
*c zConfiguration Call MuteUserOnEntry Enable: off
** end
OK
```

**Enable**: Set to **On** to Mute all new users on entry.

## zConfiguration Call ClosedCaption Visible - In-Meeting

Turn on/off closed captioning: Turning this one will display closed captioning if it is available. To see if closed captioning is available, refer to **zStatus Call ClosedCaption Available**

Schema:

```
zConfiguration Call ClosedCaption Visible[: <bool>]
```

Example:

```
zConfiguration Call ClosedCaption Visible
*c zConfiguration Call ClosedCaption Visible: off
** end
OK
```

**Enable**: Set to **On** to see available closed captions. Resets after every meeting. Default is **off**at the beginning of a meeting.

## zConfiguration Call ClosedCaption FontSize - In-Meeting

Sets closed captioning size. To see if closed captioning is available, refer to **zStatus Call ClosedCaption Available**

Schema:

```
zConfiguration Call ClosedCaption FontSize[: <0-2>]
```

Example:

```
zConfiguration Call ClosedCaption FontSize
*c zConfiguration Call ClosedCaption FontSize: 1
** end
OK
```

**FontSize**: Resets after every meeting. Default is **1** at the beginning of a meeting.

# zEvents

Most zEvent Notifications correspond to zStatus (read only) or zConfiguration (read-write) changes, and return data structures that are identical to the data structures returned by zStatus and zConfiguration.

However, pure zEvents are events that return structures with a schema that doesn't correspond to the zStatus or zConfiguration reply structures.

## zEvent Pipe OtherControllerLogin

Currently, it is not possible to have both the ZR-CSAPI, and the iPad Zoom Room Controller controlling the Zoom Room. When an iPad Zoom Room Controller logs in, it takes over control of the Zoom room from the ZR-CSAPI. This zEvent is issued, and the ZR-CSAPI loses the pipe connection to the ZR. In order to regain control, isse any ZR-CSAPI command; this action causes the ZR-CSAPI to reconnect, but not execute the command you entered. Then, you can re-enter the command you wish to run.

Schema:

```
*e Pipe OtherControllerLogin
** end
```

Example:

```
*e Pipe OtherControllerLogin
** end
```

## zEvent Pipe Reconnecting

If the ZR-CSAPI loses the pipe connection to the ZR, then when any command is issued, the command will not be executed; instead, this zEvent will be issued, and the ZR-CSAPI will re-establish the pipe connection to the ZR. The desired command must be issued again.

Schema:

```
*e Pipe Reconnecting
** end
```

Example:

```
*e Pipe Reconnecting
** end
```

## zEvent IncomingCallIndication

Issued when an incoming call arrives at the ZR. Typically, another Zoom client will call the Zoom Room by selecting the Zoom Room from the Contact list.

Schema

```
*e IncomingCallIndication callerJID: <string> // Join ID of the caller. Use this value to Accept the caller using zCommand Call Accept, if the Zoom Room is set to not automatically accept incoming calls.
```

```
*e IncomingCallIndication calleeJID: <string> // Typically empty. Join ID
of the host of the meeting being called
*e IncomingCallIndication meetingID: <string> // An internal meeting Id
used for setting up recording.
*e IncomingCallIndication password: <string> // The password entered by
the participant who intends to join.
*e IncomingCallIndication meetingOption: <string> // DEPRECATED
*e IncomingCallIndication meetingNumber: <int> // Meeting number for this
meeting
*e IncomingCallIndication callerName: <string> // The name of the caller
*e IncomingCallIndication avatarURL: <string> // Avatar image of the
person joining
*e IncomingCallIndication lifeTime: <int> // Scheduled duration of the
meeting
** end
```

Example

```
*e IncomingCallIndication callerJID: thqrasdqs4wzdnch8kcbiw@xmpp.zoom.us
*e IncomingCallIndication calleeJID:
*e IncomingCallIndication meetingID: E
*e IncomingCallIndication password:
*e IncomingCallIndication meetingOption: 6
*e IncomingCallIndication meetingNumber: 766315077
*e IncomingCallIndication callerName: Scott Firestone Testing ZR
*e IncomingCallIndication avatarURL:
*e IncomingCallIndication lifeTime: 60
** end
```

## zEvent TreatedIncomingCallIndication

When another user directly calls the Zoom Room, and the Zoom Room has been set to not automatically accept incoming calls, the Zoom Room will get an IncomingCallIndication. However, it is possible for the call to end before the Zoom Room accepts or rejects the call via the ZR-CSAPI. In particular:

- The caller might end the meeting before the Zoom Room accepts or rejects the call.
- If you hook up a RevoLabs UV 500 speakerphone to the ZR, users can accept/reject incoming calls by pressing buttons on the RevoLabs Speakerphone, rather than through the ZR-CSAPI.

When the call is connected or disconnected before the ZR-CSAPI accepts or rejects the call, you get notification that the call has been accepted or rejected:

Schema:

```
*e TreatedIncomingCallIndication callerJID: <string>
*e TreatedIncomingCallIndication calleeJID: <string>
*e TreatedIncomingCallIndication meetingID: <string>
*e TreatedIncomingCallIndication password: <string>
*e TreatedIncomingCallIndication meetingOption: <int>
```

```
*e TreatedIncomingCallIndication meetingNumber: <int>
*e TreatedIncomingCallIndication callerName: <string>
*e TreatedIncomingCallIndication avatarURL: <string>
*e TreatedIncomingCallIndication lifeTime: <int>
*e TreatedIncomingCallIndication accepted: <on/off>
** end
```

Example:

```
*e TreatedIncomingCallIndication callerJID:
thqrasdqs4wzdnch8kcbiw@xmpp.zoom.us
*e TreatedIncomingCallIndication calleeJID:
*e TreatedIncomingCallIndication meetingID: ?
*e TreatedIncomingCallIndication password:
*e TreatedIncomingCallIndication meetingOption: 6
*e TreatedIncomingCallIndication meetingNumber: 284756735
*e TreatedIncomingCallIndication callerName: Scott Firestone Testing ZR
*e TreatedIncomingCallIndication avatarURL:
*e TreatedIncomingCallIndication lifeTime: 45
*e TreatedIncomingCallIndication accepted: off
** end
```

The entries are the same as for the IncomingCallIndication notification, with the addition of the accepted parameter: it is off if the call was rejected, and on if the call was accepted.

## zEvent CallDisconnect

Notification that the room exited the meeting

Schema

```
*e CallDisconnect success : <on/off>
** end
```

Example

```
*e CallDisconnect success : on
** end
```

## zEvent CallConnectError

The Zoom Room has encountered an error when connecting

Schema

```
*e CallConnectError error_code: <int> // An integer error code; It will
be 0 if there is no error.
*e CallConnectError error_message: <on/off> // A plain text error message
** end
```

Example

```
*e CallConnectError error_code: 0
```

```
*e CallConnectError error_message: This meeting has been ended by host.
** end
```

## ZAAPIconnectFail

An internal error in the ZR-CSAPI

Schema

```
*e ZAAPIconnectFail
** end
```

Example

```
*e ZAAPIconnectFail
** end
ERROR
```

## zEvent Bookings Updated

There has been a change to the list of scheduled meetings. Download the list to see the difference.

Schema

```
*e Bookings Updated
** end
```

Example

```
*e Bookings Updated
** end
```

## zEvent Connection rejected

**This limitation will be removed in the January release**

Encryption for the link between the ZR and ZRC can be turned on via the web portal, under **Account settings -> Zoom Rooms -> Secure connection channel**. In this case, the ZR-CSAPI must use encryption for the connection to the ZR; but the ZR-CSAPI has not been upgraded to support encryption. You get the following event notification:

```
*e Connection rejected
** end
```

## zEvent VideoUnMuteReqeust

This scenario happens if a Zoom Room connects to a meeting, and it is not the host of that meeting, then the meeting host asks the Zoom Room to unmute the ZR video. In this case, the Zoom Room will not automatically unmute its video; instead, the Zoom Room will issue a notification on the CLI that the meeting host has requested the ZR to unmute video:

```
*e VideoUnMuteRequest ID: 16778241
*e VideoUnMuteRequest isHost: on
```

```
*e VideoUnMuteRequest isCoHost: off
** end
```

The automation controller can react by asking the user if the request for video unmute is granted. After the user grants the unmute video request, the automation controller can unmute the Zoom Room's video by invoking **zConfiguration Call Camera Mute: off**

## zEvent MeetingNeedsPassword

If you do not enter a password for a meeting that needs it, you get this message after attempting to join:

```
*e MeetingNeedsPassword needsPassword: true
*e MeetingNeedsPassword wrongAndRetry: false
** end
```

If you enter the wrong password for a meeting that needs a password, you get this notification:

```
*e MeetingNeedsPassword needsPassword: true
*e MeetingNeedsPassword wrongAndRetry: true
** end
```

In some cases, if a meeting does not need a password, you get this notification:

```
*e MeetingNeedsPassword needsPassword: false
*e MeetingNeedsPassword wrongAndRetry: false
** end
```

If a password is required, you must re-issue the join command, and specify a password. See the **zCommand Dial Join command** for details.

## zEvent SharingState

If someone starts a sharing session, you get the following notification:

```
*e SharingState state: <None | Connecting | Sending | Receiving |
Send_Receiving>
*e SharingState paused: <on/off>
** end
```

You also get this notification if someone is running the Zoom Room on a Windows Touch device, and running the white board, and the person launches a sharing meeting from the whiteboard.

## zEvent : device change

If someone plugs in, or unplugs, a camera, speaker, or microphone, you get a notification, with a complete list of the current devices. This notification is related to three status commands:

```
zStatus Audio Input Line
zStatus Audio Output Line
zStatus Video Camera Line
```

For instance, if someone plugs in a Logitech HD Pro Webcam C920, that adds a new camera and a new microphone, you get an updated list of cameras and microphones:

```
*s Video Camera Line 1 id: CDC85FD0-E73A-4FC2-B3A8-EA237D6990E0
*s Video Camera Line 1 Name: CamTwist
*s Video Camera Line 1 Alias: usb:7D69:90E0
*s Video Camera Line 2 id: 0x14200000046d082d
*s Video Camera Line 2 Name: HD Pro Webcam C920
*s Video Camera Line 2 Alias: usb:046D:082D
*s Video Camera Line 3 id: CDC85FD0-E73A-4FC2-B3A8-EA237D6990E1
*s Video Camera Line 3 Name: CamTwist (2VUY)
*s Video Camera Line 3 Alias: usb:7D69:90E1
** end
```

```
*s Audio Input Line 1 id: Sennheiser SC70 USB CTRL
*s Audio Input Line 1 Name: Sennheiser SC70 USB CTRL
*s Audio Input Line 1 Alias:
*s Audio Input Line 2 id: HD Pro Webcam C920
*s Audio Input Line 2 Name: HD Pro Webcam C920
*s Audio Input Line 2 Alias:
*s Audio Input Line 3 id: Built-in Input
*s Audio Input Line 3 Name: Built-in Input (Line In)
*s Audio Input Line 3 Alias:
*s Audio Input Line 4 id: Soundflower (2ch)
*s Audio Input Line 4 Name: Soundflower (2ch)
*s Audio Input Line 4 Alias:
*s Audio Input Line 5 id: Soundflower (64ch)
*s Audio Input Line 5 Alias:
** end
```

## zEvent NeedWaitForHost

when you join a pre-scheduled meeting by invoking **zCommand Dial Join MeetingNumber:**You will immediately get a notification to indicate whether you need to wait for the host to join the meeting.

Schema:

```
*e NeedWaitForHost Wait: <bool>
** end
```

If the host has not enabled JBH (join before host) for the meeting, then you must wait. In the case that you need to wait, you get:

```
*e NeedWaitForHost Wait: on
** end
```

Followed by another notification, once the host joins:

```
*e NeedWaitForHost Wait: off
```

```
** end
```

Or, if you join a meeting that allows JBH, then you get a notification immediately, indicating that you don't have to wait:

```
*e NeedWaitForHost Wait: off
** end
```

## zEvent OpenVideoFailForHostStop

When in a meeting, the host is able to turn off the video coming from any participant, including Zoom Rooms. If the host turns off the video coming from a Zoom Room, then the Zoom Room will not be able to turn the video back on, until the host allows it. If the video from the Zoom Room is currently disabled, and the Zoom Room tries to unmute its camera, you will first get an error for the unmute operation, then you will get an event indicating that the host has disabled video from the Zoom Room:

Schema:

```
*e Message Event: <Unknown | OpenVideoFailForHostStop>
** end
```

Example:

```
zConfiguration Call Camera Mute: off
*r Configuration (status=Error):
** end
ERROR
*e Message Event: OpenVideoFailForHostStop
** end
```

The host can ask the Zoom Room to re-enable its camera video by sending a request to unmute. The ZR-CSAPI will provide a notification when this request happens:

```
*e VideoUnMuteRequest ID: 16779265
*e VideoUnMuteRequest isHost: on
*e VideoUnMuteRequest isCoHost: off
** end
```

Then, the Zoom room can successfully unmute its video:

```
zConfiguration Call Camera Mute: off
** end
OK
```

## zEvent AddedContact

Notification when a new contact is added to the Zoom Room's phonebook.

Schema:

```
*e Phonebook Added Contact n jid: <string> / the Join Id of the contact.
Use this ID to invite a person to a meeting.
```

```
*e Phonebook Added Contact n screenName: <string> // the Zoom IM
ScreenName of the contact
*e Phonebook Added Contact n firstName: <string>
*e Phonebook Added Contact n lastName: <string>
*e Phonebook Added Contact n phoneNumber: <string> // the phone number of
the contact
*e Phonebook Added Contact n email: <string> // the email of the contact
*e Phonebook Added Contact n avatarURL: <string> / a URL pointing to the
avatar of the contact, if available
*e Phonebook Added Contact n presence: <PRESENCE_OFFLINE,
PRESENCE_ONLINE, PRESENCE_AWAY, PRESENCE_BUSY, PRESENCE_DND>
*e Phonebook Added Contact n onDesktop: <on/off> // Set to on if the user
is currently using a desktop client. Only valid if the presence status is
PRESENCE_ONLINE.
*e Phonebook Added Contact n onMobile: <on/off> // Set to on if the user
is currently using a mobile Zoom client. Only valid if the presence
status is PRESENCE_ONLINE.
*e Phonebook Added Contact n isZoomRoom: <on/off> / Set to on if the user
is a zoom room.
*e Phonebook Added Contact n isLegacy: <on/off> / Set to on if the user
is currently using a legacy platform (Ex. SIP/H323).
*e Phonebook Added Contact n presence_status: <int>
*e Phonebook Added Contact n sip_phone_number: <string> //
sip_phone_number if applicable
*e Phonebook Added Contact n cloud_pbx_info isValid: <on/off> // Whether
cloud_pbx is on/off
*e PhoneBook Added Contact n cloud_pbx_info extension: <string> //Only if
cloud_pbx_info isValid is on
*e PhoneBook Added Contact n cloud_pbx_info company_number: <string> //
Only if cloud_pbx_info isValid is on
*e PhoneBook Added Contact n cloud_pbx_info direct_number_list: m
direct_number: <string> //Only if cloud_pbx_info isValid is on
** end
```

Example:

```
*e Phonebook Added Contact 22 jid: H323_SIP_00000005
*e Phonebook Added Contact 22 screenName: testcontact
*e Phonebook Added Contact 22 firstName:
*e Phonebook Added Contact 22 lastName:
*e Phonebook Added Contact 22 phoneNumber:
*e Phonebook Added Contact 22 email:
*e Phonebook Added Contact 22 avatarURL:
*e Phonebook Added Contact 22 presence: PRESENCE_OFFLINE
*e Phonebook Added Contact 22 onDesktop: off
*e Phonebook Added Contact 22 onMobile: off
*e Phonebook Added Contact 22 isZoomRoom: off
*e Phonebook Added Contact 22 isLegacy: on
```

```
*e Phonebook Added Contact 22 presence_status: 0
*e Phonebook Added Contact 22 sip_phone_number:
*e Phonebook Added Contact 22 cloud_pbx_info isValid: off
** end
```

## zEvent UpdatedContact

Notification when a contact in the Zoom Room's phonebook is updated.

Schema:

```
*e Phonebook Updated Contact n jid: <string> / the Join Id of the
contact.
Use this ID to invite a person to a meeting.
*e Phonebook Updated Contact n screenName: <string> // the Zoom IM
ScreenName of the contact
*e Phonebook Updated Contact n firstName: <string>
*e Phonebook Updated Contact n lastName: <string>
*e Phonebook Updated Contact n phoneNumber: <string> // the phone number
of the contact
*e Phonebook Updated Contact n email: <string> // the email of the
contact
*e Phonebook Updated Contact n avatarURL: <string> / a URL pointing to
the avatar of the contact, if available
*e Phonebook Updated Contact n presence: <PRESENCE_OFFLINE,
PRESENCE_ONLINE, PRESENCE_AWAY, PRESENCE_BUSY, PRESENCE_DND>
*e Phonebook Updated Contact n onDesktop: <on/off> // Set to on if the
user is currently using a desktop client. Only valid if the presence
status is PRESENCE_ONLINE.
*e Phonebook Updated Contact n onMobile: <on/off> // Set to on if the
user is currently using a mobile Zoom client. Only valid if the presence
status is PRESENCE_ONLINE.
*e Phonebook Updated Contact n isZoomRoom: <on/off> // Set to on if the
user is a zoom room.
*e Phonebook Updated Contact n isLegacy: <on/off> / Set to on if the user
is currently using a legacy platform (Ex. SIP/H323).
*e Phonebook Updated Contact n presence_status: <int>
*e Phonebook Updated Contact n sip_phone_number: <string> //
sip_phone_number if applicable
*e Phonebook Updated Contact n cloud_pbx_info isValid: <on/off> //
Whether cloud_pbx is on/off
*e PhoneBook Updated Contact n cloud_pbx_info extension: <string> //Only
if cloud_pbx_info isValid is on
*e PhoneBook Updated Contact n cloud_pbx_info company_number: <string> //
Only if cloud_pbx_info Updated is on
*e PhoneBook Updated Contact n cloud_pbx_info direct_number_list: m
direct_number: <string> //Only if cloud_pbx_info isValid is on
```

```
** end
```

Example:

```
*e Phonebook Updated Contact 22 jid: H323_SIP_00000005
*e Phonebook Updated Contact 22 screenName: testcontact
*e Phonebook Updated Contact 22 firstName:
*e Phonebook Updated Contact 22 lastName:
*e Phonebook Updated Contact 22 phoneNumber:
*e Phonebook Updated Contact 22 email:
*e Phonebook Updated Contact 22 avatarURL:
*e Phonebook Updated Contact 22 presence: PRESENCE_OFFLINE
*e Phonebook Updated Contact 22 onDesktop: off
*e Phonebook Updated Contact 22 onMobile: off
*e Phonebook Updated Contact 22 isZoomRoom: off
*e Phonebook Updated Contact 22 isLegacy: on
*e Phonebook Updated Contact 22 presence_status: 0
*e Phonebook Updated Contact 22 sip_phone_number:
*e Phonebook Updated Contact 22 cloud_pbx_info isValid: off
** end
```

## zEvent UpdatedCallRecordInfo

Notification when the recording status of the Zoom Room changes.

Schema:

```
*e UpdateCallRecordInfo canRecord: <true/false> // Can Zoom Room record
*e UpdateCallRecordInfo emailRequired: <true/false> // Is email required?
Or email already stored in meeting?
*e UpdateCallRecordInfo amIRecording: <true/false> // Is Zoom Room
currently recording
*e UpdateCallRecordInfo meetingIsBeingRecorded: <true/false> //Is the
meeting currently being recorded by anyone, NOTE: This is not completely
reliable yet
** end
```

Example:

```
*e UpdateCallRecordInfo canRecord: true
*e UpdateCallRecordInfo emailRequired: true
*e UpdateCallRecordInfo amIRecording: false
*e UpdateCallRecordInfo meetingIsBeingRecorded: false
** end
```

# zFeedback

Use the zFeedback mechanism to subscribe or unsubscribe from zEvent notifications, so that you receive a subset of notifications from the ZR-CSAPI.

You determine which events you receive by adding filters to two different filter sets:

- an inclusion filter set
- an exclusion filter set Each filter consists of a path node in the command hierarchy. A zEvent matches a filter if the zEvent is at or below the path node. The filtering is a 2-phase process: First, an incoming zEvent is compared to the filters in the incluion set. If the zEvent does not match any filter, the ZR-CSAPI drops the zEvent. Otherwise, the ZR-CSAPI compares the zEvent to the filters in the exclusion set. If the zEvent matches any filter, it will be dropped; otherwise, it will become a notification from the ZR-CSAPI.

The idea is to add a few filters to the **inclusion** set to establish the broad categories of zEvents that you wish to receive; you typically specify a few path nodes that are higher up in the hierarchy. Then, add filters to the **exclusion** set that specify path nodes further down in the hierarchy, to carve out small subsets of zEvents near the leaves of the hierarchy that you don't want to receive.

The top of the hierarchy consists of three path nodes: /Status /Configuration /Events

You can get a list of filters in both sets using this command:

zFeedback List

By default, the ZR-CSAPI launches with these settings:

Inclusion set:

- /Status
- /Configuration
- /Event

Exclusion set: empty.

For example, it can be annoying to receive the very large callin_country_list when you get the **zEvent InfoResult notification** message:

Example:

```
zcommand call info
OK
*r InfoResult (status=OK):
*r InfoResult Info real_meeting_id: euBIzGp3S5i7qjrifE87vQ==
*r InfoResult Info meeting_id: 5526136251
*r InfoResult Info participant_id: 34
*r InfoResult Info my_userid: 16778240
*r InfoResult Info am_i_original_host: on
*r InfoResult Info is_webinar: off
*r InfoResult Info is_view_only: off
*r InfoResult Info meeting_type: NORMAL
*r InfoResult Info meeting_password:
*r InfoResult Info dialIn: +1 669 900 6833;+1 646 558 8656
*r InfoResult Info toll_free_number:
*r InfoResult Info international_url: /zoomconference
```

```
*r InfoResult Info is_toll_free_callin_list_available: on
*r InfoResult Info is_callin_country_list_available: on
*r InfoResult Info is_calling_room_system_enabled: on
*r InfoResult Info support_callout_type: NONE
*r InfoResult Info user_type: PRO
*r InfoResult Info callin_country_list 1 id: AR
*r InfoResult Info callin_country_list 1 name: Argentina
*r InfoResult Info callin_country_list 1 code: 54
*r InfoResult Info callin_country_list 1 number: 54 3415122188
*r InfoResult Info callin_country_list 1 display_number: +54 341 512 2188
*r InfoResult Info callin_country_list 2 id: AR
*r InfoResult Info callin_country_list 2 name: Argentina
*r InfoResult Info callin_country_list 2 code: 54
*r InfoResult Info callin_country_list 2 number: 54 3434145986
*r InfoResult Info callin_country_list 2 display_number: +54 343 414 5968
*r InfoResult Info callin_country_list 3 id: AU
*r InfoResult Info callin_country_list 3 name: Australia
*r InfoResult Info callin_country_list 3 code: 61
*r InfoResult Info callin_country_list 3 number: 61 280152088
*r InfoResult Info callin_country_list 3 display_number: +61 (0) 2 8015
2088
*r InfoResult Info callin_country_list 4 id: AU
...
```

The path node of the hierarchy for the callin_country_list section of the notification is:

/Events/InfoResult/info/callin_country_list

In order to avoid receiving this list, while still receiving the other fields in the InfoResult notification, use the **zFeedback Register command**, with the Operation set to **ex**, to add this path node to the exclusion set:

zFeedback Register Op: ex Path: /Event/InfoResult/info/callin_country_list

The path is case-insensitive.

Currently, the zFeeback mechanim applies only to /Event/InfoResult/info/callin_country_list, and only for the CLI responses, not the JSON reponses. The idea is to make the CLI more usable. The zFeedback mechanism will be extended later to other notifications, and to both the CLI and JSON responses.

## zFeedback Register

Add a filter to either the inclusion or exclusion filter set

Schema

```
zFeedback Register Op: <in | ex> Path: <string>
** end
OK
```

Example

```
zFeedback Register Op: ex Path: /Event/InfoResult/info/
callin_country_list
** end
OK
```

**Op**: The operation: either add the filter to the inclusion list, or the exclusion list.

**Path**: The path node in the hierarchy. a zEvent will match the filter if it it located at the same level, or below, this path node.

## zFeedback Deregister

Schema

```
zFeedback Deregister Path: <string>
** end
OK
```

Example

```
zFeedback Deregister Path: /Event/InfoResult/info/callin_country_list
** end
OK
```

**Path**: The path node in the hierarchy to remove from both the inclusion set and the exclusion set.

## zFeedback List

List all filters in the inclusion set and the exclusion set.

Schema

```
zFeedback list
*f in: <string>
...
*f ex: <string>
** end
OK
```

Example

```
zFeedback list
*f in: /Configuration
*f in: /Event
*f in: /Status
** end
OK
```

## zFeedback Deregisterall

Remove all filters from all sets

Schema

```
zFeedback Deregisterall
** end
OK
```

Example

```
zFeedback Deregisterall
** end
OK
```

# zStatus

zStatus provides access to read-only status information. These values can change based on external events. When a value changes, the CLI will issue an asynchronous notification.

The status parameters are arranged in a path hierarchy, and you can access the hierarchy at various points. In some cases, you cannot go all the way down to a leaf in the hierarchy; you can specify a path down to an intermediate node, and you get all values at that point, and lower, in the hierarchy.

## zStatus Call Status

Get current call status

Schema

```
zstatus call status
*s Call Status: <NOT_IN_MEETING | CONNECTING_MEETING | IN_MEETING |
LOGGED_OUT |UNKNOWN> // The state of the call
** end
OK
```

Example

```
zstatus call status
*s Call Status: IN_MEETING
** end
OK
```

## zStatus Audio Input Line

Get info on an audio input

Schema

```
zStatus Audio Input Line
*s Audio Input Line n id: <string> // Id of the device
*s Audio Input Line n Name: <string> // Name of the device
*s Audio Input Line n Alias: <string> // Alias of the device
*s Audio Input Line n Selected: <on/off> // Whether the device is
selected
*s Audio Input Line n manuallySelected: <on/off> // Whether device is
manually selected
*s Audio Input Line n combinedDevice: <on/off> //Whether device is
actually multiple devices
*s Audio Input Line n numberOfCombinedDevices: <int> //Number of combined
devices
*s Audio Input Line n ptzComId: <int> // PanTiltZoom Comm ID if
applicable
...
```

```
** end
OK
```

Example

```
zStatus Audio Input Line
*s Audio Input Line 1 id: HD Pro Webcam C920
*s Audio Input Line 1 Name: HD Pro Webcam C920
*s Audio Input Line 1 Alias:
*s Audio Input Line 1 Selected: on
*s Audio Input Line 1 manuallySelected: on
*s Audio Input Line 1 combinedDevice: off
*s Audio Input Line 1 numberOfCombinedDevices: 0
*s Audio Input Line 1 ptzComId: -1
** end
OK
```

## zStatus Audio Output Line

Get info on an audio Output

Schema

```
zStatus Audio Output Line
*s Audio Output Line n id: <string> // Id of the device
*s Audio Output Line n Name: <string> // Name of the device
*s Audio Output Line n Alias: <string> // Alias of the device
*s Audio Output Line n Selected: <on/off> // Whether the device is
selected
*s Audio Output Line n manuallySelected: <on/off> // Whether device is
manually selected
*s Audio Output Line n combinedDevice: <on/off> //Whether device is
actually multiple devices
*s Audio Output Line n numberOfCombinedDevices: <int> //Number of
combined devices
*s Audio Output Line n ptzComId: <int> // PanTiltZoom Comm ID if
applicable
...
** end
OK
```

Example

```
zStatus Audio Output Line
*s Audio Output Line 1 id: Built-in Output
*s Audio Output Line 1 Name: Built-in Output (Internal Speakers)
*s Audio Output Line 1 Alias:
*s Audio Output Line 1 Selected: on
*s Audio Output Line 1 manuallySelected: on
*s Audio Output Line 1 combinedDevice: off
```

```
*s Audio Output Line 1 numberOfCombinedDevices: 0
*s Audio Output Line 1 ptzComId: -1
** end
OK
```

## zStatus Video Camera Line

Get info on Camera input

Schema

```
zStatus Video Camera Line
*s Video Camera Line n id: <string> // Id of the device
*s Video Camera Line n Name: <string> // Name of the device
*s Video Camera Line n Alias: <string> // Alias of the device
*s Video Camera Line n Selected: <on/off> // Whether the device is
selected
*s Video Camera Line n manuallySelected: <on/off> // Whether device is
manually selected
*s Video Camera Line n combinedDevice: <on/off> //Whether device is
actually multiple devices
*s Video Camera Line n numberOfCombinedDevices: <int> //Number of
combined devices
*s Video Camera Line n ptzComId: <int> // PanTiltZoom Comm ID if
applicable
...
** end
OK
```

Example

```
zStatus Video Camera Line
*s Video Camera Line 1 id: 0x14100000046d082d
*s Video Camera Line 1 Name: HD Pro Webcam C920
*s Video Camera Line 1 Alias: usb:046D:082D
*s Video Camera Line 1 Selected: off
*s Video Camera Line 1 manuallySelected: off
*s Video Camera Line 1 combinedDevice: off
*s Video Camera Line 1 numberOfCombinedDevices: 0
*s Video Camera Line 1 ptzComId: -1
*s Video Camera Line 2 id: 0x14a00000095d9204
*s Video Camera Line 2 Name: Polycom EagleEye IV USB Camera
*s Video Camera Line 2 Alias: usb:095D:9204
*s Video Camera Line 2 Selected: on
*s Video Camera Line 2 manuallySelected: on
*s Video Camera Line 2 combinedDevice: off
*s Video Camera Line 2 numberOfCombinedDevices: 0
*s Video Camera Line 2 ptzComId: -1
```

```
** end
OK
```

## zStatus Video Optimizable

Whether screen sharing is optimizable for video playback. If so, then you can issue the command **zConfiguration Call Sharing optimize_video_sharing**

Schema

```
zStatus Video Optimizable
*s Video Optimizable: <on/off> // Set to On if it is possible to invoke the **zConfiguration Call Sharing optimize_video_sharing** command.
** end
OK
```

Example

```
zStatus Video Optimizable
*s Video Optimizable: on
** end
OK
```

## zStatus SystemUnit

Get System information

Schema

```
zStatus SystemUnit
*s SystemUnit login_type: <google | work_email> // Users log into the Zoom room using either a Google email or a work email.
*s SystemUnit email: <string> // The auto-generated email assigned to this Zoom Room: It defines a virtual "Zoom user" for this Zoom Room. This virtual user is listed in the account user list on the Web Portal.
*s SystemUnit meeting_number: <string> // The Zoom Room PMI meeting number. It is an int, but this value is a string because in the future it may be a vanity number.
*s SystemUnit platform: <string> // Zoom Room platform type
*s SystemUnit room_version: <string> // Zoom Room release version
*s SystemUnit room_info room_name: <string> // Room name
*s SystemUnit room_info account_email: <string> // The email of the Zoom user who is administrating this Zoom Room. This is the email that is used to log into the Zoom Room.
*s SystemUnit room_info is_auto_answer_enabled: <on/off> // Whether auto answer is available on this Zoom Room
*s SystemUnit room_info is_auto_answer_selected: <on/off> // Whether auto answer is turned on for this Zoom Room
```

```
*s SystemUnit room_info display_version: <string> //Displays version
number
** end
OK
```

Example

```
zStatus SystemUnit
*s SystemUnit login_type: work_email
*s SystemUnit email: scott.firestone_U-ygqt3HStSmbY1JfYW-OA@parasync.com
*s SystemUnit meeting_number: 5526136251
*s SystemUnit platform: Mac OS X, 10.13.6
*s SystemUnit room_version: 4.1.33237.0924
*s SystemUnit room_info room_name: ZR-ScottFirestone2
*s SystemUnit room_info account_email: room@kanovia.com
*s SystemUnit room_info is_auto_answer_enabled: on
*s SystemUnit room_info is_auto_answer_selected: off
*s SystemUnit room_info display_version: 4.2.0 (416.1123)
** end
OK
```

## zStatus Capabilities

Get System Capabilities

Schema

```
zStatus Capabilities
zStatus Capabilities is_Airhost_Disabled: <on/off>
zStatus Capabilities can_Dtmf_For_Invite_By_Phone <on/off>
zStatus Capabilities pstn_Call_In_Local_Presentation <on/off>
zStatus Capabilities can_Ringing_In_Pstn_Call <on/off>
zStatus Capabilities supports_Sip_Call_out <on/off> // The ZR allows SIP
callout. The Zoom Room must be configured to have the SIP Phone
Integration add-on license.
zStatus Capabilities aec_Setting_Stored_In_ZR <on/off>
zStatus Capabilities supports_Web_Settings_Push <on/off>
zStatus Capabilities support_Pin_And_Spotlight <on/off> // Whether the
zoom Room can Pin other participants. When a user is pinned, that user
will always appear full-screen on the Zoom Room (does not affect remote
endpoints), and always show up in one of the connected monitors, even if
speaker selection mode is turned on. For spotlight: applies to meetings
with 3 or more participants only. The host, and all remote participants,
will see that person full screen: speaker selection mode, and pin mode,
is overridden on all endpoints. The spotlighted person is effectively
pinned on all endpoints.
zStatus Capabilities can_Mute_On_Entry <on/off> // Whether this ZR is
capable of muting all participants when they enter the meeting.
```

```
zStatus Capabilities support_Claim_Host <on/off> // Whether the Zoom Room can be instructed to take over the host privilege for a meeting.
zStatus Capabilities support_Out_Room_Display <on/off> // Whether the Zoom Room can host a scheduling display, typically mounted outside the conference room.
zStatus Capabilities can_Switch_To_Specific_Camera <on/off>
zStatus Capabilities supports_Software_Audio_Processing: <on/off> // Whether the Zoom Room supports SAP.
zStatus Capabilities supports_Hdmi_Cec_Control: <on/off> // Whether the ZR can send a CeC command via HDMI to wake up a video monitor.
zStatus Capabilities supports_Highly_Reverberant_Room <on/off> // Whether the Zoom Room can be configured to compensate for high reverb within a conference room.
zStatus Capabilities supports_Share_For_Floating_And_Content_only: <on/off> // Whether the zoom Room allows the shared content to be switch to a video thumbnail in active speaker mode.
*s Capabilities supports_Loading_Contacts_Dynamically: <on/off> // Whether the Zoom Room supports loading Phonebook contacts in a scalable manner, to accommodate a very large directory.
*s Capabilities supports_ShareCamera: <on/off> // Whether it is possible to share an HDMI capture signal
*s Capabilities supports_Audio_Checkup: <on/off> // Whether audio troubleshooting mode is supported: Not implemented in the ZR-CSAPI.
*s Capabilities supports_Loading_Participants_Dynamically: <on/off> // Whether the Zoom Room supports loading meeting participants in a scalable manner, to acccommodate large webinars of > 1000 participants
*s Capabilities supports_CheckIn: <on/off> // whether or not the Zoom Room can be configured to require users to check in to avoid having a meeting cancelled. Not supported in the ZR-CSAPI.
*s Capabilities supports_Expel_User_Permanently: <on/off> // Whether Expelled users are removed from meetings permanently, until the meeting is ended for all.
*s Capabilities supports_Multi_Share: <on/off> //Whether multi_share is supported
*s Capabilities supports_Mic_Record_Test: <on/off> //Whether recording mic test is supported
*s Capabilities supports_Encrypted_Connection: <on/off> //Whether encrypted connection is supported
*s Capabilities supports_H323_DTMF: <on/off> //Whether H323 DTMF is supported
*s Capabilities supports_Cloud_PBX: <on/off> //Whether cloud PBX is supported
** end
OK
```

Example

```
zStatus Capabilities
```

```
*s Capabilities is_Airhost_Disabled: off
*s Capabilities can_Dtmf_For_Invite_By_Phone: on
*s Capabilities pstn_Call_In_Local_resentation: on
*s Capabilities can_Ringing_In_Pstn_Call: on
*s Capabilities supports_Sip_Call_out: on
*s Capabilities aec_Setting_Stored_In_ZR: on
*s Capabilities supports_Web_Settings_Push: on
*s Capabilities support_Pin_And_Spotlight: on
*s Capabilities can_Mute_On_Entry: on
*s Capabilities support_Claim_Host: on
*s Capabilities support_Out_Room_Display: on
*s Capabilities can_Switch_To_Specific_Camera: on
*s Capabilities supports_Software_Audio_Processing: on
*s Capabilities supports_Hdmi_Cec_Control: on
*s Capabilities supports_Highly_Reverberant_Room: on
*s Capabilities supports_Share_For_Floating_And_Content_Only: on
*s Capabilities supports_Loading_Contacts_Dynamically: on
*s Capabilities supports_ShareCamera: on
*s Capabilities supports_Audio_Checkup: on
*s Capabilities supports_Loading_Participants_Dynamically: on
*s Capabilities supports_CheckIn: on
*s Capabilities supports_Expel_User_Permanently: on
*s Capabilities supports_Multi_Share: on
*s Capabilities supports_Mic_Record_Test: on
*s Capabilities supports_Encrypted_Connection: on
*s Capabilities supports_H323_DTMF: on
*s Capabilities supports_Cloud_PBX: off
** end
OK
```

## zStatus Sharing

Get status of sharing

Schema:

```
zStatus Sharing
*s Sharing wifiName: <string> // If the ZR uses a WiFi access point, then
the name of that WiFi hot spot appears here. It will be blank if the ZR
has a wired connection.  You need to display that WiFi access point name
to the user, so the user can connect the laptop to that WiFi network. If
the ZR is on a wired connection, this entry will be empty.
*s Sharing serverName: <string> // Name of Zoom Room.
*s Sharing password: <string> // The airplay password, for sharing your
iOS device. It is not the sharing key or the meeting password.
```

```
*s Sharing isAirHostClientConnected: <on/off> // Used in the
notification. When this entry is set to true, you know that you are now
in a sharing meeting, and you can remove the airplay instructions.
*s Sharing isBlackMagicConnected: <on/off> // Set to on if a compatible
HDMI capture device is connected via USB to the Zoom Room. We support
Magewell and INOGENI.
*s Sharing isBlackMagicDataAvailable: <on/off> // Set to on, if the user
has connected an HDMI cable from a laptop to the HDMI capture card, and
the HDMI capture card sees HDMI video coming from the Laptop.
*s Sharing isSharingBlackMagic: <on/off> // Whether HDMI is currently
actively sharing.
*s Sharing directPresentationPairingCode: <string> // This is the paring
code that is broadcast via an ultrasonic signal from the ZRC. It is
different than the user-supplied paring code. The ZRC uses a Zoom-
proprietary method of advertizing the ultrasonic pairing code, so it's
not possible to advertize it using commonly available libraries.
*s Sharing directPresentationSharingKey: <string> // The alpha-only
sharing key that users type into a laptop client to share with the Zoom
Room.
*s Sharing isDirectPresentationConnected: <on/off> // The laptop has
connected to the ZR, either via HDMI, or via network sharing.
*s Sharing dispState: <None | Laptop | IOS> // Gets the instructions the
ZR is displaying on the monitor
** end
OK
```

Example:

```
zStatus Sharing
*s Sharing wifiName:
*s Sharing serverName: ZR-ScottFirestone2
*s Sharing password: 4621
*s Sharing isAirHostClientConnected: off
*s Sharing isBlackMagicConnected: off
*s Sharing isBlackMagicDataAvailable: off
*s Sharing isSharingBlackMagic: off
*s Sharing directPresentationPairingCode: 450031
*s Sharing directPresentationSharingKey: JTEXBA
*s Sharing isDirectPresentationConnected: off
*s Sharing dispState: None
** end
OK
```

## zStatus CameraShare - In-Meeting

Gets status of camera sharing

***Bug***: It seems pan_Tilt_Speed is always 0

Schema:

```
zStatus CameraShare
*s zStatus CameraShare is_Sharing: <bool> // Whether the camera is
currently being shared
*s zStatus CameraShare id: <string> // If sharing, the id for the camera
that is sharing
*s zStatus CameraShare can_Control_Camera: <bool> // Whether it is
possible to use the PTZ controls to control the camera.
*s zStatus CameraShare is_Mirrored: <bool> // Whether the camera is
currently mirrored.
*s CameraShare pan_Tilt_Speed: <int> // Movement speed of camera when
sharing
** end
OK
```

Example:

```
zStatus CameraShare
*s CameraShare is_Sharing: off
*s CameraShare id:
*s CameraShare can_Control_Camera: off
*s CameraShare is_Mirrored: off
*s CameraShare pan_Tilt_Speed: 0
** end
OK
```

## zStatus Call Layout - In-Meeting

Video layout Capabilities and status. Lists which video layout state transitions are allowed at this moment. The logic in the ZR to determine what layouts are allowed is complex. The possible allowed layouts depend on:

- The number of monitors attached (1, 2, 3).
- The Web portal settings for this Zoom Room under the Display Settings tab.
- Whether sharing is active.
- Whether the Zoom Room camera video is muted or unmuted.

***Known Bug***: This command does not currently display whether you can switch to strip view (thumbnail view on iPad ZRC), to do this, use ***zStatus NumberOfScreens***, strip view can be used if number of screens is 1.

Schema:

```
zStatus Call Layout
*s Layout can_Switch_Wall_View: <on/off> // On if it is possible to
invoke zConfiguration Call Layout Style: Gallery, to switch to the
Gallery mode, showing video participants in tiled windows: The Zoom Room
shows up to a 5x5 array of tiled windows per page.
```

```
*s Layout can_Adjust_Floating_Video: <on/off> // On if it is possible to change the position and size of the thumbnail shown in Speaker view, by invoking zConfiguration Call Layout Position/Size.
*s Layout can_Switch_Floating_Share_Content: <on/off> // on if it is possible to invoke zConfiguration Call Layout ShareThumb: on, to put the sharing content into a small video thumbnail.
*s Layout can_Switch_Share_On_All_Screens: <on/off> // Set to On if it is possible to invoke zConfiguration Call Layout Style: ShareAll, to switch to the ShareAll mode, where the content sharing is shown full screen on all monitors.
*s Layout can_Switch_Speaker_View: <on/off> // Set to On if it is possible to switch to Speaker view by invoking zConfiguration Call Layout Style: Speaker. The active speaker is shown full screen, and other video streams, like self-view, are shown in thumbnails.

These are valid only if the Layout Style is currently Gallery or Strip:

*s Layout is_supported: <on/off> //Set to On if it is possible to change the parameters
*s Layout is_In_First_Page: <on/off> // Set to On if the ZR is showing the first page of possibly multiple pages of videos. The UI can show left/right arrow button to flip between pages: When On, gray out the left arrow button.
*s Layout is_In_Last_Page: <on/off> // Set to On if the ZR is showing the last page of possibly multiple pages of videos. On the UI, gray out the right arrow button.
*s Layout video_Type: <Gallery | Strip> // Indicates which mode applies: Strip or Gallery.
*s Layout video_Count_In_Current_Page: <int> // For the filmstrip only, specifies the number of thumbnails shown.
```

Example:

```
zStatus Call Layout
*s Layout can_Switch_Wall_View: off
*s Layout can_Adjust_Floating_Video: off
*s Layout can_Switch_Floating_Share_Content: off
*s Layout can_Switch_Share_On_All_Screens: off
*s Layout can_Switch_Speaker_View: off
*s Layout is_supported: on
*s Layout is_In_First_Page: on
*s Layout is_In_Last_Page: on
*s Layout video_Type: Strip
*s Layout video_Count_In_Current_Page: 0
** end
OK
```

## zStatus Call ClosedCaption Available - In-Meeting

Gets whether closed captioning is currently available. This can either be queried, or it will return an asynchronous notification when availability changes.

Schema:

```
zStatus Call ClosedCaption Available
```

Example (Querying):

```
zStatus Call ClosedCaption Available
*s Call ClosedCaption Available: off
** end
OK
```

Example (Asynchronous):

```
*s Call ClosedCaption Available: off
** end
```

## zStatus NumberOfScreens

Get number of screens attached to the Zoom Room. Can be queried or asynchronous when number of screen changes.

Schema

```
zstatus NumberOfScreens
*s NumberOfScreens: <int> //Total number of Screens
*s NumberOfCECScreens: <int> //Number of CEC connected screens
** end
OK
```

Example (Querying):

```
zstatus NumberOfScreens
*s NumberOfScreens: 3
*s NumberOfCECScreens: 0
** end
OK
```

Example (Asynchronous):

```
zstatus NumberOfScreens
*s NumberOfScreens: 3
*s NumberOfCECScreens: 0
** end
```