

ZOOMOSC VERSION 4.5.0



# ZOOMOSC COMMAND SYNTAX

ZoomOSC adds a bi-directional Open Sound Control (OSC) interface to Zoom. Commands can be sent to the ZoomOSC app from third party hardware and software devices to control Zoom, and ZoomOSC can communicate back to those products with information about the Zoom meeting.

This document describes the syntax for these OSC commands.

The application's OSC messages can be divided into **3 major categories**

- **User Commands** require targets, which can be either users/participants or groups.
- **Global Commands** do not use targets, and control general meeting or application settings.
- **Outputs** are the OSC messages that come out from ZoomOSC containing useful information. Some outputs are triggered by events of subscribed participants, others are triggered by a User or Global command and thus provide a response.



# REFRESHER ON OSC PROTOCOL

**ZoomOSC interacts with Open Sound Control as UDP.**

**Here is an example ZoomOSC message:**

**`/zoom/userName/videoOn "Andy Carluccio"`**

The address is delimited with "/" (in this case, the address is `/zoom/userName/videoOn`).

The payloads follow the address and are delimited with whitespace. ZoomOSC payloads can be ints, floats, or strings.

Multi word strings can be enclosed in quotes to pass as a single argument.

Please reference the user manual for the control software you are using to determine how that program constructs and/or parses OSC.

## Every ZoomOSC Input starts with `/zoom`

### **Constructing User Command to send to ZoomOSC**

To send a message to ZoomOSC, you must construct an address. The sections below are appended to `/zoom` to specify who to call the actions on.

# HOW DO YOU TARGET A USER?

After /zoom, you may select a "target type" for the user you want to act on.

The available target types are below.

## .../targetID

A single user who is logged by index in the target list. NOTE: For this to succeed, you must have loaded or updated users into target memory before referencing.

{int targetIndex}

## .../userName

A single zoom username, case sensitive and including spaces. Multiple participants with the same name will revert the target to the first participant it finds with that name

{str userName}

## .../galIndex

The zero-indexed position of the participant in the gallery.

{int galleryIndex}

## .../zoomID

The zoomID is randomly assigned when a participant joins, and will change if they leave and rejoin.

{int zoomID}

## .../galleryPosition [Mac Required]

A single user at a specified gallery position identified by row and column (zero indexed).

{str position in form "row, col" ie "0, 2"}

## .../me

Passes self as the target

N/A

# TARGET A GROUP OF USERS WITH A MACRO

Instead of calling a target type, you can pass one of the below "group macros" as an alternative

<b>.../all</b>	Everyone in the Zoom call
<b>.../targets</b>	Everyone in the target list
<b>.../panelists</b>	All webinar panelists
<b>.../attendees</b>	All webinar attendees

## EXCLUDE USERS FROM A GROUP MACRO

Another type of group macro is "exclusionary" and must be combined with a target type. For example, **/zoom/allExcept/me/...** would call the user action on everyone in the meeting except for the self user.

<b>.../allExcept/[TARGET TYPE]</b>
<b>.../targetsExcept/[TARGET TYPE]</b>
<b>.../panelistsExcept/[TARGET TYPE]</b>
<b>.../attendeesExcept/[TARGET TYPE]</b>

## MAKE A CUSTOM GROUP "FROM SCRATCH"

Finally, it is also possible to define a custom group using the "users" keyword followed by the target type that will be used to designate the members of the group.

**.../users/[TARGET TYPE]**



# ZOOMOSC COMMANDS

EVERY ZOOMOSC INPUT STARTS WITH **/ZOOM**

## EXAMPLES

**/zoom/userName/chat "Andy" "Hello!"**

Send a "Hello" chat to user Andy

**/zoom/targetsExcept/userName/videoOff "Joe"**

Turn off video for everyone on the target list except for Joe

**/zoom/users/galleryIndex/addPin 2 6 8 3**

Create a multipin from the persons of gallery indices 2, 6, 8, and 3

**/zoom/targetID/videoOn 7**

Request that the user with targetID 7 turn on their video

**/zoom/users/zoomID/addSpot 58387672 48394821 48271291**

Create a multi spotlight from the persons using the raw zoomID system

**/zoom/all/lowerHand**

Lower the hands of all users in the Zoom call

## How to read the commands:

### Command Name

This is a description of the command

Here is an additional note which may give a warning or important information

### ../commandAddress

{int arg1 | str arg2}

Mac Only **MAC**

Windows Only **WIN**

Paid, Pro Feature **PRO**

Experimental (may have bugs) **BETA**

Please remember that the first arguments of any user action will be the designator for that user. The arguments listed in the spreadsheet are in addition to the user specification arguments and must be attached at the end of the payload list.

# USER COMMANDS

## VIDEO/MIC COMMANDS

<b>Request Video On</b> Requests to turn on participant video	<b>../videoOn</b>
<b>Set Video Off</b> Turns off participant video	<b>../videoOff</b>
<b>Toggle Video</b> Toggles participant video on or off	<b>../toggleVideo</b>
<b>Mute Mic</b> Mutes participant mic	<b>../mute</b>
<b>Unmute Mic</b> Requests to unmute participant mic, unless pre-approved consent is given in Zoom settings. ZoomOSC will automatically accept requests to unmute.	<b>../unMute</b>
<b>Toggle Mic</b> Toggles participant mic on or off	<b>../toggleMute</b>

## SPOTLIGHT COMMANDS (see global commands for clearSpot command)

<b>Spotlight</b> Replaces current spotlight with a new participant spotlight  If called on a group, it will only spotlight the last user in the list.	<b>../spot</b>	
<b>Add Spotlight</b> Adds a new participant spotlight to the mutli-spot group	<b>../addSpot</b>	<b>PRO</b>
<b>Un-Spotlight</b> Removes a spotlight from a participant	<b>../unSpot</b>	
<b>Toggle Spotlight</b> Toggles spotlight on or off from a participant	<b>../toggleSpot</b>	<b>PRO</b>

## HAND RAISING COMMANDS (see global commands for lowerAllHands command)

<b>Raise Hand</b> Raises Hand	<b>../raiseHand</b>	
<b>Lower Hand</b> Lowers Hand	<b>../lowerHand</b>	
<b>Toggle Hand</b> Toggles the raising or lowering of the hand	<b>../toggleHand</b>	<b>WIN</b>

# USER COMMANDS

## PIN COMMANDS

(see page 13 for remote pin controls)

<b>Pin Participant</b> Pins a participant	<b>../pin</b>	
<b>Add Pin</b> Adds a new participant pin to the multi-pin group  Only Co-Hosts/Hosts. Other participants must be given permission by the host to multi-pin.	<b>../addPin</b>	<b>PRO</b>
<b>Un-Pin Participant</b> Removes a pin from a participant	<b>../unPin</b>	
<b>Pin to Second Screen</b> Pins participant to the second screen	<b>../pin2</b>	
<b>Un-Pin from Second Screen</b> Unpins participant from the second screen	<b>../unPin2</b>	
<b>Toggle Pin First Screen</b> Toggles pin on or off from the first screen	<b>../togglePin</b>	<b>PRO</b>
<b>Toggle Pin Second Screen</b> Toggles pin on or off from the second screen	<b>../togglePin2</b>	<b>PRO</b>
<b>Clear all Pins</b> Clear all pins (from the first display, Zoom itself does not support multipinning on the second display at this time).	<b>../clearPin</b>	<b>PRO</b>

## VIEW COMMANDS

<b>Set Gallery View</b> Changes screen to gallery view	<b>../setGalleryView</b>
<b>Set Speaker View</b> Changes screen to speaker view	<b>../setSpeakerView</b>
<b>Next Gallery Page</b> Changes screen to next page of Gallery View	<b>../galleryPageNext</b>
<b>Previous Gallery Page</b> Changes screen to previous page of Gallery View	<b>../galleryPagePrev</b>

# USER COMMANDS

## USER ROLES AND ACTION COMMANDS

<b>Make Host</b> Makes the defined participant the meeting host	<b>../makeHost</b>	
<b>Make Co-Host</b> Makes the defined participant a meeting co-host	<b>../makeCoHost</b>	
<b>Reclaim Host</b> Reclaims host role if originally the creator of the meeting  Only works if you are the original host of the meeting; must be called on self	<b>../reclaimHost</b>	
<b>Revoke Co-Host</b> Revokes co-host permission from participant	<b>../revokeCoHost</b>	
<b>Make Panelist</b> Promotes a webinar attendee to a panelist	<b>../makePanelist</b>	
<b>Make Attendee</b> Demotes a webinar panelist to an attendee	<b>../makeAttendee</b>	
<b>Eject from Meeting</b> Kicks participant from meeting	<b>../eject</b>	
<b>Rename</b> Change the name of a user	<b>../rename</b>	<b>PRO</b>
<b>Allow to Record</b> Permit an attendee to record	<b>../allowToRecord</b>	<b>PRO</b>
<b>Disallow to Record</b> Remove permission for an attendee to record	<b>../disallowToRecord</b>	<b>PRO</b>

## CHAT COMMANDS

<b>Send Chat</b> Sends chat to a participant  Zoom chat is limited to 1024 characters.  This command will send via a direct message to the selected user(s)	<b>../chat</b> {str chat message}
---	--------------------------------------

## WEBINARS

<b>Allow Attendee to Speak</b> Allows a webinar attendee to speak	<b>../allowToSpeak</b>	<b>PRO</b>
<b>Disallow Attendee to Speak</b> Disallows a webinar attendee to speak	<b>../disallowToSpeak</b>	<b>PRO</b>



# USER COMMANDS

## BREAKOUT ROOMS\*

<b>Send User to Breakout Room</b> Moves a specific user to a room, or moves that user between rooms	<b>../sendToBreakout</b> {str breakout name   int breakout index}	<b>PRO BETA</b>
<b>Remove User from Breakout Room</b> Removes a specific user from a room	<b>../removeFromBreakout</b> {str breakout name   int breakout index}	<b>PRO BETA</b>
<b>Assign User to Breakout Room</b> Assigns a specific user to a breakout room before opening. To move an already assigned user, un-assign first before re-assigning.	<b>../assignToBreakout</b> {str breakout name   int breakout index}	<b>PRO BETA</b>
<b>Un-assign User from Breakout Room</b> Un-assigns a specific user to a breakout room before opening.	<b>../unassignFromBreakout</b> {str breakout name   int breakout index}	<b>PRO BETA</b>

## \*INFORMATION ON BETA BREAKOUT CONTROLS

The current implementation of breakouts in ZoomOSC is limited by the absence of Global UUID in the Zoom Meeting SDK. However, both internal and external teams have thoroughly tested and seen stable performance. Because there are significant "corner cases" arising from the complexity of breakout rooms, please use with caution and avoid managing breakout rooms both through ZoomOSC and through the UI simultaneously. We strongly encourage complete testing of your exact command flow before production use.

### Order of Breakout Room operations:

1. **CREATE** breakout rooms
2. **CONFIGURE** breakout rooms with OSC command
3. **ASSIGN** participants to rooms
4. **OPEN** breakout rooms
5. **SEND/REMOVE** participants from open rooms

## WAITING ROOMS

<b>Send User to Waiting Room</b> Sends a specific user to the waiting room if enabled in the meeting	<b>../sendToWaitingRoom</b>	<b>PRO</b>
<b>Admit User from Waiting Room</b> Admits a specific user from the waiting room into the main meeting	<b>../admit</b>	<b>PRO</b>

# USER COMMANDS

## SCREENSHARE COMMANDS

<b>List Screens</b> Lists all screens available for screenshare	<b>../listScreens</b>	<b>PRO</b>
<b>List Windows</b> Lists all windows available for windowshare	<b>../listWindows</b>	<b>PRO MAC</b>
<b>Start Screenshare</b> Starts a screenshare on defined screen Use the listScreen command first to identify monitor ID or names	<b>../startScreenShare</b> {int screenID   str screenName}	<b>PRO</b>
<b>Start Windowshare</b> Starts a windowshare on defined window Use the listWindows command first to identify monitor ID or names	<b>../startWindowShare</b> {int windowID   str windowName}	<b>PRO MAC</b>
<b>Stop Share</b> Stops sharing screen, window, camera, or audio	<b>../stopShare</b>	
<b>Start Screenshare (Primary Display)</b> Starts screenshare on primary display only	<b>../startScreenSharePrimary</b>	
<b>Start Sharing Computer Audio (Only)</b> Start "Share Computer Sound" (audio only)	<b>../startAudioShare</b>	<b>PRO WIN</b>
<b>Enable Computer Sound (for Sharing)</b> Enables the computer audio share setting when sharing visual content	<b>../enableComputerSoundWhenSharing</b>	<b>PRO</b>
<b>Disable Computer Sound (for Sharing)</b> Disables the computer audio share setting when sharing visual content	<b>../disableComputerSoundWhenSharing</b>	<b>PRO</b>
<b>Start Sharing Camera Source</b> Starts sharing computer camera source	<b>../startCameraShare</b>	<b>PRO</b>
<b>Advance to Next Camera Share Source</b> Cycle to the next shareable camera source while inside of camera sharing mode	<b>../shareNextCamera</b>	<b>PRO</b>
<b>Enable Video Share Optimization</b> Sets screen share to prefer frame rate over resolution	<b>../enableOptimizeVideo</b>	<b>PRO</b>
<b>Disable Video Share Optimization</b> Sets screen share to prefer resolution over frame rate	<b>../disableOptimizeVideo</b>	<b>PRO</b>
<b>Set Primary Window Size</b> Adjust the primary window to a specific size	<b>../setWindowSize</b> {int width   int height}	<b>PRO MAC BETA</b>
<b>Set Primary Window Position</b> Adjust the primary window's position on screen	<b>../setWindowPosition</b> {int x  int y}	<b>PRO MAC BETA</b>

# USER COMMANDS

## SETTINGS COMMANDS

<b>Display Usernames on Videos</b> Displays usernames in the bottom left of participants	<b>../showUserNames</b>	
<b>Hide Usernames on Videos</b> Hides usernames in the bottom left of participants	<b>../hideUserNames</b>	
<b>Show Non-Video Participants</b> Shows non-video participants in gallery	<b>../showNonVideoParticipants</b>	
<b>Hide Non-Video Participants</b> Hides non-video participants in gallery	<b>../hideNonVideoParticipants</b>	
<b>Enable “Original Sound”</b> Enables "original sound" for audio  The command will work, but it helps to use the "show original audio" setting in meeting"	<b>../enableOriginalSound</b>	
<b>Disable “Original Sound”</b> Disables "original sound" for audio	<b>../disableOriginalSound</b>	
<b>Hide Self View</b> Hides your own video tile in the gallery	<b>../hideSelfView</b>	<b>PRO MAC</b>
<b>Show Self View</b> Show your own video tile in the gallery	<b>../showSelfView</b>	<b>PRO MAC</b>
<b>List All Camera Devices</b> Lists all available camera devices	<b>../listCameraDevices</b>	<b>PRO</b>
<b>List All Mic Devices</b> Lists all available mic devices	<b>../listMicDevices</b>	<b>PRO</b>
<b>List All Speaker Devices</b> Lists all available speaker devices	<b>../listSpeakerDevices</b>	<b>PRO</b>
<b>Set Camera Device</b> Changes camera device  Use the listCameraDevices or getCameraDevice command to get the deviceId or index	<b>../setCameraDevice</b> {string deviceId   int index}	<b>PRO</b>
<b>Set Mic Device</b> Changes mic device  Use the listMicDevices or getMicDevice command to get the deviceId or index	<b>../setMicDevice</b> {string deviceId   int index}	<b>PRO</b>
<b>Set Speaker Device</b> Changes speaker device  Use the listSpeakerDevices or getSpeakerDevice command to get the deviceId or index	<b>../setSpeakerDevice</b> {string deviceId   int index}	<b>PRO</b>

# USER COMMANDS

## SETTINGS COMMANDS CONTINUED

<b>Get Current Camera Device</b> Lists current camera device in use	<b>../getCameraDevice</b>	<b>PRO</b>
<b>Get Current Mic Device</b> Lists current mic device in use	<b>../getMicDevice</b>	<b>PRO</b>
<b>Get Current Speaker Device</b> Lists current speaker device in use	<b>../getSpeakerDevice</b>	<b>PRO</b>
<b>List Virtual Backgrounds</b> Lists all available virtual backgrounds	<b>../listBackgrounds</b>	<b>PRO MAC</b>
<b>Change Virtual Background</b> Changes virtual background  Use the listBackgrounds or the getBackground commands first to identify index or bgName	<b>../setBackground</b> {int index   str bgName}	<b>PRO MAC BETA</b>
<b>Get Current Virtual Background</b> Lists current virtual background in use	<b>../getBackground</b>	<b>PRO MAC BETA</b>
<b>Get Mic Level</b> Returns the mic level	<b>../getMicLevel</b>	<b>PRO</b>
<b>Set Mic Level</b> Sets the mic level to a value	<b>../setMicLevel</b> {int 0> 100}	<b>PRO</b>
<b>Get Speaker Volume</b> Returns the system speaker volume	<b>../getSpeakerVolume</b>	<b>PRO</b>
<b>Set Speaker Volume</b> Sets the system speaker volume to a value	<b>../setSpeakerVolume</b> {int 0> 100}	<b>PRO</b>
<b>Enable Mirror Video</b> Mirrors self video	<b>../enableMirrorVideo</b>	<b>PRO</b>
<b>Disable Mirror Video</b> Mirrors self video	<b>../disableMirrorVideo</b>	<b>PRO</b>
<b>Enable HD Video (Setting)</b> Enable the HD Video Setting	<b>../enableHDVideo</b>	<b>PRO</b>
<b>Disable HD Video (Setting)</b> Disable the HD Video Setting	<b>../disableHDVideo</b>	<b>PRO</b>
<b>Set Video Filter</b> Set video filter	<b>../setVideoFilter</b>	<b>PRO MAC BETA</b>

# GLOBAL COMMANDS

## GLOBAL COMMANDS

<b>Enable Users Unmuting</b> Enables the meeting setting which allows participants to unmute	<b>/zoom/enableUsersUnmute</b>	
<b>Disable User Unmuting</b> Disables the meeting setting which allows participants to unmute	<b>/zoom/disableUsersUnmute</b>	
<b>Mute All</b> mutes all participants except host	<b>/zoom/all/mute</b>	
<b>Unmute All</b> Unmutes all participants	<b>/zoom/all/unMute</b>	
<b>Lower All Raised Hands</b> Lowers all raised hands in the meeting	<b>/zoom/lowerAllHands</b>	
<b>Clear Spotlight</b> Clears all spotlights from meeting	<b>/zoom/clearSpot</b>	<b>PRO</b>
<b>Ping ZoomOSC</b> Sends a ping to ZoomOSC and review a pong response	<b>/zoom/ping</b> {optional any argument to reply}	
<b>Join a Meeting</b> Joins a meeting with OSC arguments	<b>/zoom/joinMeeting</b> {str meetingID} {str meetingPass} {str userName}	<b>PRO</b>
<b>Leave Meeting</b> Leaves current meeting	<b>/zoom/leaveMeeting</b>	<b>PRO</b>
<b>End Meeting</b> Ends current meeting for all	<b>/zoom/endMeeting</b>	<b>PRO</b>
<b>Send Chat to Everyone</b> Sends a chat message to all participants	<b>/zoom/chatAll</b> {str message}	
<b>Eject All Webinar Attendees</b> Ejects all attendees from the webinar	<b>/zoom/ejectAttendees</b>	
<b>Get Webinar Reaction Count</b> Outputs the number of each emoji reaction for the webinar	<b>/zoom/getWebinarReactionCounts</b>	<b>PRO</b>
<b>Reset Webinar Reaction Counters</b> Resets all webinar reaction counters to 0	<b>/zoom/resetWebinarReactionCounts</b>	<b>PRO</b>

# GLOBAL COMMANDS

## BREAKOUT ROOMS\*

<b>Request Breakout Rooms List</b> Commands ZoomOSC to respond with the breakout list Please note: the output OSC message is currently different on mac and win	<b>/zoom/listBreakouts</b>	<b>PRO BETA</b>
<b>Create Breakout Room</b> Creates a breakout room while rooms are closed	<b>/zoom/createBreakout</b> {string breakoutName   int bo_index}	<b>PRO BETA</b>
<b>Delete Breakout Room</b> Deletes a breakout room while rooms are closed	<b>/zoom/deleteBreakout</b> {string breakoutName   int bo_index}	<b>PRO BETA</b>
<b>Delete All Breakout Rooms</b> Deletes all breakout Rooms	<b>/zoom/deleteAllBreakouts</b>	<b>PRO BETA</b>
<b>Open Breakout Rooms</b> Opens breakout rooms	<b>/zoom/openBreakouts</b>	<b>PRO BETA</b>
<b>Close Breakout Rooms</b> Closes breakout rooms	<b>/zoom/closeBreakouts</b>	<b>PRO BETA</b>
<b>Configure Breakout Rooms</b> Configures breakout room settings, such as timers and duration	<b>/zoom/configureBreakouts</b> {int postCloseSeconds} {int allowChooseBreakout (0=false, 1=true)} {int allowReturnAtWill} {int autoMoveParticipants} {int useTimer} {int closeWithTimer} {int breakoutDurationSeconds}	<b>PRO BETA</b>
<b>Broadcast Message to Breakout Rooms</b> Send a message to all breakout rooms which appears at the top of the attendees screens	<b>/zoom/broadcastToBreakouts</b> {string message}	<b>PRO BETA</b>

## \*INFORMATION ON BETA BREAKOUT CONTROLS

Because there are significant "corner cases" arising from the complexity of breakout rooms, please use with caution and avoid managing breakout rooms both through ZoomOSC and through the UI simultaneously. We strongly encourage complete testing of your exact command flow before production use.

### Order of Breakout Room operations:

1. **CREATE** breakout rooms
2. **CONFIGURE** breakout rooms with OSC command
3. **ASSIGN** participants to rooms
4. **OPEN** breakout rooms
5. **SEND/REMOVE** participants from open rooms

# GLOBAL COMMANDS

## MEETING AND WEBINAR Q&A\*

<b>Answer Question with Text</b> ZoomOSC will post the supplied text as the answer to the designated question	<b>/zoom/answerQuestionText</b> {string answer, string question_id}	<b>PRO</b>
<b>Mark Question as Answering Live</b> The designated question will be marked as answering live	<b>/zoom/answerQuestionLive</b> {string question_id}	<b>PRO</b>
<b>Finish Answering Question</b> The designated question will be marked as having been completed	<b>/zoom/answerQuestionDone</b> {string question_id}	<b>PRO</b>
<b>Dismiss Question</b> The designated question will be dismissed	<b>/zoom/dismissQuestion</b> {string question_id}	<b>PRO</b>
<b>Delete Question</b> The designated question will be deleted	<b>/zoom/deleteQuestion</b> {string question_id}	<b>PRO</b>
<b>Reopen Question</b> The designated question will be reopened	<b>/zoom/reopenQuestion</b> {string question_id}	<b>PRO</b>
<b>Get Question List</b> Request a list of all questions	<b>/zoom/getQuestionList</b>	<b>PRO</b>
<b>Get Question Info</b> Get the details of a specific question	<b>/zoom/getQuestionInfo</b> {string question_id}	<b>PRO</b>
<b>Get Answer List</b> Request a list of all answers to a single question or across all questions	<b>/zoom/getAnswerList</b> {(optional) string question_id}	<b>PRO</b>
<b>Get Answer Info</b> Get the details of a specific answer	<b>/zoom/getAnswerInfo</b> {string answer_id}	<b>PRO</b>

## \*INFORMATION ON QUESTION / ANSWER INTERFACE

These commands require the unique ID of each question and answer, which is provided in the OSC Outputs for the Q&A system. Those IDs should be recalled so that you can take actions on questions and answers that were captured by your listening to the Q&A OSC outputs.

# GLOBAL COMMANDS

---

## POLLS

<b>Start Poll</b> Start a poll	<b>/zoom/startPoll</b> {string poll_ID}
<b>Stop Poll</b> Stop a poll	<b>/zoom/stopPoll</b> {string poll_ID}
<b>Relaunch Poll</b> Relaunch a previously-stopped poll	<b>/zoom/relaunchPoll</b> {string poll_ID}
<b>Get Poll Answers</b> Request the answer information for a poll	<b>/zoom/getPollAnswers</b> {string poll_ID}
<b>Get Poll Info</b> Request the metadata of the poll	<b>/zoom/getPollInfo</b> {string poll_ID}
<b>Get List of Polls</b> Request a list of all the available polls	<b>/zoom/getPollList</b>
<b>Get Poll Questions</b> Request the list of questions within a poll	<b>/zoom/getPollQuestions</b> {string poll_ID}



# GLOBAL COMMANDS

## RECORDING

<b>Start Local Recording</b> Starts a local recording	<b>/zoom/startLocalRecording</b>
<b>Pause Local Recording</b> Pauses local recording	<b>/zoom/pauseLocalRecording</b>
<b>Resume Local Recording</b> Resumes a local recording after paused	<b>/zoom/resumeLocalRecording</b>
<b>Stop Local Recording</b> Stops local recording	<b>/zoom/stopLocalRecording</b>
<b>Start Cloud Recording</b> Starts a cloud recording	<b>/zoom/startCloudRecording</b>
<b>Pause Cloud Recording</b> Pauses cloud recording	<b>/zoom/pauseCloudRecording</b>
<b>Resume Cloud Recording</b> Resumes a cloud recording after paused	<b>/zoom/resumeCloudRecording</b>
<b>Stop Cloud Recording</b> Stops cloud recording	<b>/zoom/stopCloudRecording</b>

## WAITING ROOMS AND ZAK JOIN

<b>Enable Waiting Room</b> Enables waiting room for the meeting	<b>/zoom/enableWaitingRoom</b>	<b>PRO</b>
<b>Disable Waiting Room</b> Disables waiting room for the meeting	<b>/zoom/disableWaitingRoom</b>	<b>PRO</b>
<b>Message Waiting Room</b> Sends a message to the waiting room	<b>/zoom/messageWaitingRoom</b> {str message}	<b>PRO</b>
<b>Admit All from Waiting Room</b> Admits all users from the waiting room	<b>/zoom/admitAll</b>	<b>PRO</b>
<b>ZAK Join Meeting</b> Joins a meeting using a ZAK token	<b>/zoom/zakJoin</b> {str zak} {str meetingID} {str name} [optional] {str password}	<b>PRO</b>
<b>ZAK Start Meeting</b> Starts a meeting using a ZAK token. Use 0 for meeting ID to use PMID	<b>/zoom/zakStart</b> {str zak} {str meetingID} {str name} [optional] {str password}	<b>PRO</b>

# GLOBAL COMMANDS

## MEMORY MANAGEMENT

<b>Update TargetIDs</b> Assigns targetIDs to participants in memory	<b>/zoom/update</b>
<b>Include</b> Appends new participants to memory	<b>/zoom/include</b>
<b>Load from Target List</b> Loads participants from target list file	<b>/zoom/load</b>
<b>Load via OSC</b> Loads participants from OSC arguments	<b>/zoom/load</b> {string userName1} .. {string userNameN}
<b>Save to Target List</b> Saves participants to target list	<b>/zoom/save</b>
<b>Reset</b> Resets application memory containers	<b>/zoom/reset</b>
<b>List*</b> Prints memory to console  *Returns list via OSC if subscribed and PRO	<b>/zoom/list</b>

## GALLERY TRACKING AND DATA REQUESTS

<b>Gallery Track Mode</b> Changes gallery tracking mode to participantID or zoomID	<b>/zoom/galTrackMode</b> {int participantID = 0, zoomID = 1}
<b>Subscription Level</b> Sets subscription level for user output data. None = 0, TargetList = 1, All = 2, Panelists = 3, OnlyGallery = 4	<b>/zoom/subscribe</b> {int mode}
<b>Request Order of Gallery View</b> Outputs the order of the gallery view	<b>/zoom/getGalleryOrder</b>
<b>Request Order of Spotlights</b> Outputs the order of the participants in a multi-spot group	<b>/zoom/getSpotOrder</b>
<b>Request Gallery Count</b> Outputs the number of participants in the gallery view	<b>/zoom/galCount</b>

# OUTPUTS

(uses **/zoomosc/me/..** if originates from own self, **/zoomosc/user/..** otherwise)

User Prefix messages (4) output for all as:  
{int targetIndex}, {str userName}, {int galleryIndex}, {int zoomID}

Additional arguments are listed below

<b>Participant Video Turned On</b> Output message when subscribed participant turns video on	<b>/zoomosc/user me/videoOn</b> usual user messages (4)
<b>Participant Video Turned Off</b> Output message when subscribed participant turns video off	<b>/zoomosc/user me/videoOff</b> usual user messages (4)
<b>Participant Muted</b> Output message when subscribed participant mutes audio	<b>/zoomosc/user me/mute</b> usual user messages (4)
<b>Participant Unmuted</b> Output message when subscribed participant unmutes audio	<b>/zoomosc/user me/unMute</b> usual user messages (4)
<b>Active Speaker Changed</b> Output message whenever active speaker changes *on macOS, the user identity is not sent	<b>/zoomosc/user me/activeSpeaker</b> <b>WIN*</b> usual user messages (4)
<b>Spotlight Started</b> Output message when subscribed participant is spotlit	<b>/zoomosc/user me/spotlightOn</b> usual user messages (4)
<b>Spotlight Stopped</b> Output message when subscribed participant is no longer spotlit	<b>/zoomosc/user me/spotlightOff</b> usual user messages (4)
<b>User Role Updates*</b> Output message when user roles have been changed 0 - USERROLE_NONE 1 - USERROLE_HOST 2 - USERROLE_COHOST 3 - UERROLE_PANELIST 4 - USERROLE_BREAKOUTROOM_MODERATOR 5 - USERROLE_ATTENDEE	<b>/zoomosc/user me/roleChanged</b> usual user messages (4), {int role}
<b>User Online</b> Output message when subscribed participant becomes online	<b>/zoomosc/user me/online</b> usual user messages (4)
<b>User Offline</b> Output message when subscribed participant becomes offline	<b>/zoomosc/user me/offline</b> usual user messages (4)
<b>User's Hand Raised</b> Output message when subscribed participant raises hand	<b>/zoomosc/user me/handRaised</b> usual user messages (4), {str message}
<b>User's Hand Lowered</b> Output message when subscribed participant lowers hand	<b>/zoomosc/user me/handLowered</b> usual user messages (4)
<b>Chat Message Received (From)</b> Output message when chat message is received 0 - NONE 1 - ALL 2 - ALL PANELISTS 3 - INDIVIDUAL PANELIST 4 - INDIVIDUAL 5 - WAITING ROOM	<b>/zoomosc/user me/chat</b> usual user messages (4), {str message} {str message_id}, {int message_type}

# OUTPUTS

<b>User Rename Event</b> Output message when a rename event occurs (note that the new username will be contained in the user prefix)	<b>/zoomosc/user/me/userNameChanged</b> usual user messages (4) {str oldUserName}	
<b>List of Breakout Rooms Output</b> Output message containing a list of all breakout rooms. Please note that at this time mac and windows send different formatted outputs	<b>/zoomosc/listBreakouts</b> {int index} {str bo_name}	<b>BETA</b>
<b>List Output</b> Output message containing all user data as arguments <ul style="list-style-type: none"> <li>• <b>int targetCount:</b> The number of users in the target list</li> <li>• <b>int listCount:</b> The number of users in the sent list</li> <li>• <b>int userRole:</b> The role of the user</li> <li>• <b>int onlineStatus:</b> Whether the user is online (0 - offline, 1 = online)</li> <li>• <b>int videoStatus:</b> Whether the user's video is on (0 = off, 1 = on)</li> <li>• <b>int audioStatus:</b> Whether the user's mic is on (0 = off, 1 = on)</li> <li>• <b>int handRaised:</b> Whether the user's hand is raised (0 = off, 1 = on)</li> </ul>	<b>/zoomosc/user/me/list</b> usual user messages (4)  {int targetCount} {int listCount} {int userRole} {int onlineStatus} {in videoStatus} {int audioStatus} {int handRaised}	<b>PRO</b>
<b>Meeting Status Changed</b> Output message when join or leave event occurs <a href="#">Status Codes</a> <a href="#">Error Codes</a> <a href="#">Exit Codes</a>	<b>/zoomosc/meetingStatusChanged</b> {int status_code, int error_code, int exit_code}	
<b>Ping Reply (Pong)</b> Output reply to a ping command containing information about the ZoomOSC application version and status	<b>/zoomosc/pong</b> {any pingArg (zero if none sent)} {str zoomOSCversion} {int subscribeMode} {int galTrackMode} {int inCallStatus 0 or 1} {int number of targets} {int number of users in call} {int isPro (1=true, 0=false)}	
<b>Gallery Order</b> Output message with order of gallery view by participant, starting from top left	<b>/zoomosc/galleryOrder</b> {int item0} ... {int itemN}	
<b>Gallery Count</b> Output message with number of participants in the gallery with video on	<b>/zoomosc/galleryCount</b> {int count}	
<b>Gallery Shape</b> Output message with the shape of the gallery in rows and columns	<b>/zoomosc/galleryShape</b> {int rows} {int cols}	<b>MAC</b>
<b>Spotlight Order</b> Output message with order of spotlight participants, starting from top left	<b>/zoomosc/spotOrder</b> {int item0} ... {int itemN}	

# OUTPUTS

<b>User Emoji Changed</b> Output message when a participant changed their emoji reaction. 1 - Clapping , 2 - Thumbs Up, 3 - Heart, 4 - Joy, 5 - Open Mouth, 6 - Tada	<b>/zoomosc/user/me/emoji changed</b> usual user messages (4) {int emojiID}
<b>User is Speaking</b> Output message with a subscribed participant is speaking	<b>/zoomosc/user/me/isSpeaking</b> usual user messages (4)
<b>User Stopped Speaking</b> Output message when a subscribed participant stops speaking	<b>/zoomosc/user/me/stopppedSpeaking</b> usual user messages (4)
<b>User Joined Waiting Room</b> Output message when a user joins the waiting room	<b>/zoomosc/user/me/joinedWaitingRoom</b> usual user messages (4)
<b>User Leaves Waiting Room</b> Output message when a user leaves the waiting room	<b>/zoomosc/user/me/leftWaitingRoom</b> usual user messages (4)
<b>Webinar Emoji Reaction Count</b> Output number of emoji reactions by type, listed 1 - Clapping , 2 - Thumbs Up, 3 - Heart, 4 - Joy, 5 - Open Mouth, 6 - Tada	<b>/zoomosc/webinarReactionCounts</b> <b>PRO</b> {int clapping, int thumbsup, int heart, int joy, int openmouth, int tada}
<b>User started sharing audio</b> Output message when a user starts sharing audio	<b>/zoomosc/user/me/audioShareStarted</b> usual user messages (4)
<b>User stopped sharing audio</b> Output message when a user stops sharing audio	<b>/zoomosc/user/me/audioShareStopped</b> usual user messages (4)
<b>User started sharing video</b> Output message when a user starts sharing video	<b>/zoomosc/user/me/videoShareStarted</b> usual user messages (4)
<b>User stopped sharing video</b> Output message when a user stops sharing video	<b>/zoomosc/user/me/videoShareStopped</b> usual user messages (4)
<b>Live Transcription Data</b> Output message containing the live transcription in a given language, with codes for operation changes 0 - None 1 - Add 2 - Update 3 - Delete 4 - Complete 5 - Not Supported  For language enums, please manually check the output as these are not pre-determined	<b>/zoomosc/user/me/transcriptMessage</b> usual user messages (4) for <u>SPEAKER</u> {str transcribed_text, int operation_type, str enum_language}

# OUTPUTS

<b>Webinar Question Deleted</b> Output message when someone deletes a question in a webinar.	<b>/zoomosc/user/me/deletedQuestion</b> usual user messages (4) for <u>ASKER</u> {str question, str question_id}
<b>Webinar Question Asked</b> Output message when an attendee asks a question in a webinar.	<b>/zoomosc/user/me/askedQuestion</b> usual user messages (4) for <u>ASKER</u> {str question, str question_id}
<b>Answer Deleted for Question</b> Output message when someone deletes an answer to a webinar question	<b>/zoomosc/user/me/deletedAnswer</b> usual user messages (4) for <u>ANSWERER</u> {str answer, int votes, str question_id, str answer_id}
<b>Answered Question with Text</b> Output message when a question has been answered via text	<b>/zoomosc/user/me/answeredQuestionText</b> usual user messages (4) for <u>ANSWERER</u> {str answer, int votes, str question_id, str answer_id}
<b>Question Dismissed</b> Output message when someone dismisses a webinar question	<b>/zoomosc/user/me/questionDismissed</b> usual user messages (4) for <u>ASKER</u> {str question, int votes, str question_id}
<b>Question Answered Live</b> Output message when a question has been marked as answering live	<b>/zoomosc/user/me/questionAnsweredLive</b> usual user messages (4) for <u>ASKER</u> {str question, int votes, str question_id}
<b>Completed Live Answer to Question</b> Output message when a question that was marked for live answering is marked as having been answered	<b>/zoomosc/user/me/questionAnsweredDone</b> usual user messages (4) for <u>ASKER</u> {str question, int votes, str question_id}
<b>Question Answered Live</b> Output message when a question has been marked as answering live	<b>/zoomosc/user/me/questionAnsweredLive</b> usual user messages (4) for <u>ASKER</u> {str question, int votes, str question_id}
<b>List of Questions</b> Outputs a series of messages corresponding to the questions asked in a meeting or webinar	<b>/zoomosc/questionList</b> {int list_index, int list_size, str question_id, str question_content}
<b>List of Answers</b> Outputs a series of messages corresponding to the answers to an individual or all questions asked in a meeting or webinar	<b>/zoomosc/answerList</b> {int list_index, int list_size, str answer_id, str answer_content}
<b>Question Information</b> Output all information on a given question	<b>/zoomosc/questionInfo</b> {int q_id, str sender_name, str q_content, bool isSenderMyself, int timestamp, bool isAnonymous, bool isMarkedAnswered, bool isMarkedDismissed, int votes, bool hasLiveAnswers, bool hasTextAnswers, bool isSelfUpVoiting, bool isSelfAnswering}

If a question is asked as anonymous, the USER PREFIX is {-1, "Anonymous Attendee", -1, -1}

# OUTPUTS

## List of Polls

Output message series for each available poll.

### Poll Status Codes

ZoomSDKPollingStatus\_Initial = 0

ZoomSDKPollingStatus\_Started = 1

ZoomSDKPollingStatus\_ShareResult = 2

ZoomSDKPollingStatus\_Stopped = 3

### Poll Type Codes

ZoomSDKPollingType\_Unknown = 0

ZoomSDKPollingType\_Poll = 1

ZoomSDKPollingType\_Quiz = 2

## /zoomosc/polls

```
{int list_index,  
int list_size,  
str poll_id,  
str poll_name,  
int pollStatus,  
int pollType,  
int isPollLibrary,  
int numQuestions}
```

## List of Poll Answers

Output message series for answer statistics on the requested poll

## /zoomosc/pollAnswers

```
{int list_index,  
int list_size,  
str poll_id,  
str poll_question_id,  
str poll_sub_question_id,  
str poll_answer_id,  
str poll_answer_name,  
int numChosen}
```

## List of Poll Question Info

Output message series for questions on the requested poll

### Question Type Codes

ZoomSDKPollingQuestionType\_Unknown = 0

ZoomSDKPollingQuestionType\_Single = 1

ZoomSDKPollingQuestionType\_Multi = 2

ZoomSDKPollingQuestionType\_Matching = 3

ZoomSDKPollingQuestionType\_RankOrder = 4

ZoomSDKPollingQuestionType\_ShortAnswer = 5

ZoomSDKPollingQuestionType\_LongAnswer = 6

ZoomSDKPollingQuestionType\_FillBlank = 7

ZoomSDKPollingQuestionType\_NPS = 8

ZoomSDKPollingQuestionType\_Dropdown = 9

## /zoomosc/pollQuestions

```
{int list_index,  
int list_size,  
str poll_id,  
str poll_question_id,  
str poll_question_name,  
int question_type}
```

## Poll Metadata Info

Output message containing the metadata of a poll

### Poll Status Codes

ZoomSDKPollingStatus\_Initial = 0

ZoomSDKPollingStatus\_Started = 1

ZoomSDKPollingStatus\_ShareResult = 2

ZoomSDKPollingStatus\_Stopped = 3

### Poll Type Codes

ZoomSDKPollingType\_Unknown = 0

ZoomSDKPollingType\_Poll = 1

ZoomSDKPollingType\_Quiz = 2

## /zoomosc/pollInfo

```
{str poll_id,  
str poll_name,  
int pollStatus,  
int pollType,  
int isPollLibrary,  
int numQuestions}
```

# OUTPUTS

<b>Mic Devices List</b> Output message containing mic devices information	<b>/zoomosc/user/me/micDevices</b> {int itemIndex}, {int listSize}, {str deviceId}, {str deviceName}, {int isSelected}	<b>PRO</b>
<b>Speaker Devices List</b> Output message containing speaker devices information	<b>/zoomosc/user/me/speakerDevices</b> {int itemIndex}, {int listSize}, {str deviceId}, {str deviceName}, {int isSelected}	<b>PRO</b>
<b>Camera Devices List</b> Output message containing camera devices information	<b>/zoomosc/user/me/cameraDevices</b> {int itemIndex}, {int listSize}, {str deviceId}, {str deviceName}, {int isSelected}	<b>PRO</b>
<b>Virtual Background List</b> Output message containing virtual background information	<b>/zoomosc/user/me/backgrounds</b> {int itemIndex}, {int listSize}, {str backgroundName}	<b>PRO MAC</b>
<b>Windows List</b> Output message containing a list of windows	<b>/zoomosc/user/me/windows</b> {str windowID} {str windowName}	<b>PRO MAC</b>
<b>Screens List</b> Output message containing a list of screens for screenshare	<b>/zoomosc/user/me/screens</b> {str screenID} {str screenName}	<b>PRO</b>
<b>Current Mic Device</b> Output message containing current mic device	<b>/zoomosc/user/me/micDevice</b> {str deviceId} {str deviceName}	<b>PRO</b>
<b>Current Speaker Device</b> Output message containing current speaker device	<b>/zoomosc/user/me/speakerDevice</b> {str deviceId} {str deviceName}	<b>PRO</b>
<b>Current Camera Device</b> Output message containing current camera device	<b>/zoomosc/user/me/cameraDevice</b> {str deviceId} {str deviceName}	<b>PRO</b>
<b>Current Virtual Background</b> Output message containing current virtual background	<b>/zoomosc/user/me/background</b> {str bgID} {str bgName}	<b>PRO MAC</b>
<b>Current Video Filter</b> Output message containing current video filter	<b>/zoomosc/user/me/filter</b> {str filterID} {str filterName}	<b>PRO MAC</b>
<b>Mic Level</b> Output message with mic level	<b>/zoomosc/micLevel</b> {int 0..100 level}	<b>PRO</b>
<b>Speaker Volume</b> Output message with speaker volume	<b>/zoomosc/speakerVolume</b> {int 0..100 volume}	<b>PRO</b>
<b>Waiting Room User List</b> Output message containing a list of participants in the waiting room	<b>/zoomosc/waitingRoomUserList</b> usual user messages (4)	





# RESOURCES

SIGN UP FOR LIMINAL'S NEWSLETTER TO STAY  
UP TO DATE ON UPDATES AND NEW RELEASES

SUBSCRIBE TO LIMINAL ON YOUTUBE FOR  
TRAININGS AND TUTORIALS

