

ZOOMOSC VERSION 3.0

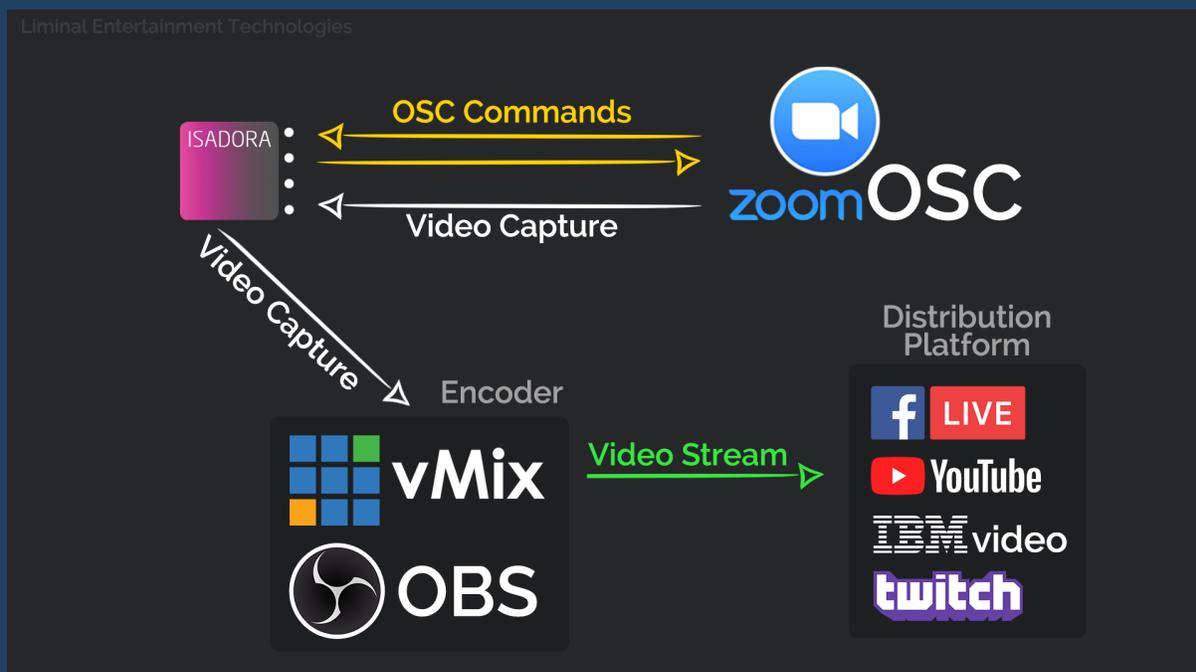
# ZOOMOSC SETUP GUIDE

*For Windows 10*

## OVERVIEW

ZoomOSC is an incredibly powerful tool for online performance. At a technical level, ZoomOSC is a modified version of Zoom that adds a bidirectional OSC engine which allows you to send OSC commands to control different functions in a Zoom meeting (such as spotlighting or muting) while also receiving information about participants (such as the order of people in Gallery View).

ZoomOSC can be a linchpin in a larger production workflow that requires a media server and/or broadcast encoder. Video designers with experience in media servers have been able to create advanced show programming.



# WORKFLOW

ZoomOSC can be used with both serialized (cued) shows or non-linear control surfaces. From the perspective of video flow, a suggested methodology is to have ZoomOSC in communication with a media server. That way, ZoomOSC can send and receive data and the ZoomOSC window/s can be screen-captured.

We recommend using the multi-display mode of the Zoom interface because you can see a gallery view and a full-screen view simultaneously.

After applying video overlays and effects using the media server, the aggregate AV feed can be sent to an encoder (ex: VMix), potentially using NDI, and then the show's "program output" can be sent to a remote audience on platforms like YouTube and Vimeo via RTMP, or it can be looped back into the Zoom meeting for a program monitor within the call.

## GETTING STARTED

### 1) Run the ZoomOSC Installer

Install the program to a location on your computer, such as Documents, that does not require administrator privileges.

### 2) When you run Zoomosc.exe, you will see a console and Zoom interface

The console contains information about the operation of the program that you can use to debug. The Zoom interface operates much like the regular Zoom platform, with a few modifications. When you launch ZoomOSC, click SET DOMAIN (without changing the text field). Most users can use the default domain when prompted. Leave "Customize UI" unchecked, as this feature at present is experimental and may lead to crashes.

### 3) Find the SDK Key and Secret (required for unofficial Zoom integrations)

Authenticate ZoomOSC via the Key and Secret found in the text file, "ZoomOSC License Key.txt." Copy and paste them into the fields for ease;. After that, the authentication should lead you to a screen from which you can associate your Zoom account or directly join a meeting.

### 4) Log into your Zoom account

The simplest way to accomplish this is to sign in with an account that is not associated with a "Log in with..." interface (e.g. "Log in with Google").

Please create a free Zoom account to use for running this application that does not require that sign in system. You can still use your organization's premium account, if you have one, to create the meetings and get access to those features, but you should log into the application with this new account. When joining a meeting, always name yourself ZoomOSC, with case sensitivity.

# STORING PARTICIPANTS

To use ZoomOSC, you have to store participants in the program's memory. Once you do this, you can call functions (**ex: /zoom/spot 1**) on participants using their ZoomOSC ID.

At any time, you can see the list of participants you've previously stored with the command, **/zoom/list**.

**If your talent changes their username** - you will not have control over them again unless you call update or modify your "performance\_config.txt" file and call load.

**If you don't know the names of the Zoom participants ahead of time** - Calling update and saving to the program's memory is an easy way to externalize the participant data to other applications.

**If someone disconnects from the meeting and then rejoins** - As an operator, you don't have to do anything to regain control of the participant for the purposes of calling commands. ZoomOSC automatically tracks information about users in the background.

*If you want to get creative, you can have your talent swap usernames (or you can do it for them) as part of an improvisational programming scheme. There are enough building blocks to build very powerful control patches!*

## 2 OPTIONS TO STORE PARTICIPANTS

### Option 1: Live Update

ZoomOSC will look at all the participants in the Zoom meeting and store them in memory, associating their usernames with a ZoomOSC ID, a placement in a zero-indexed list of participants

**/zoom/update**

### Option 2: Save and Load Preset from Disk

ZoomOSC will look for performance\_config.txt and load its contents into memory. This file is created by calling /zoom/save, which takes the contents of ZoomOSC's participant memory and stores them into the text file - (If you don't see the file, try calling update, then save) The idea of this text file is that you can just write in the Zoom usernames of the participants you want to have control of, and you can order the file so that you can have persistent bindings between ZoomOSC IDs and Zoom participants that you can use for multiple rehearsals or performances.

**/zoom/save** (Take the contents of memory and save to disk)

**/zoom/load** (Take the contents of "performance\_config.txt" on disk and load it into memory)

# THE RETURN CHANNEL

Applications can now receive OSC back from ZoomOSC. Presently, ZoomOSC sends OSC packets to port 1234 on the software loopback, 127.0.0.1. It is set up like this to work with Isadora out of the box, and you can use Isadora to forward or split the OSC data to other devices.

The most powerful feature of the return channel is the ability to track participants throughout the Gallery View. You can build upon this framework to create an automatic switching engine to simulate ISO feeds of the Zoom participants.

To use the "galtrack" system:

- Have all participants turn off video (if any were on)
- If you are using a program monitor, "Hide Self View" and turn off ZoomOSC's video feed
- Load participants into memory with load or update
- Arm galtrack with /zoom/galtrack 1
- Have participants turn on video one at a time
- Listen to /zoomosc/vid/galorder to receive the ZoomOSC IDs of everyone in the gallery view

You can also build reactive programming off of things like /zoomosc/sound/off to fire events in your Isadora patch, for example, when such an action occurs.

# ZOOMOSC 3.0 API

## FOR WINDOWS

(where x is the floating point index of a user in the zero-indexed list)

### BEFORE YOU START:

To send commands to ZoomOSC, have your OSC packets target port 8000 on the software loopback device (127.0.0.1) or via the host PC's private IP address if you are connecting from an external device, like QLab running on a Mac on the network.

Please remember to send floats from your control surface!

## COMMANDS GOING INTO ZOOMOSC ON PORT 8000

### GLOBAL ACTIONS

Spot User x	/zoom/spot x
Unspot User x	/zoom/unspot x
Request to Turn on Video for User x	/zoom/on x
Turn Off Video for User x	/zoom/off x
Mute Audio for User x	/zoom/mute x
Request to Ummute Audio for User x	/zoom/unmute x

### SELF ACTIONS

Pin User x	/zoom/pin x
Unpin User x	/zoom/unpin x
Turn Off ZoomOSC Video	/zoom/me/off 1.0
Turn On ZoomOSC Video	/zoom/me/on 1.0
Switch to Gallery View	/zoom/gallery 1.0
Switch to Speaker View	/zoom/gallery 0.0

### GALLERY TRACKING CONTROLS

Request a reply with the count of People in call with Video On	/zoom/galcount 1.0
Arm Gallery Placement Tracker and/or Request Gallery Layout	/zoom/galtrack 1.0
Disarm Gallery Placement Tracker and Delete Tracking	/zoom/galtrack 0.0

### MEMORY FUNCTIONS

Create a list of Current Participants and store it in Memory	/zoom/update 1.0
Create a list in Memory from the contents of performance_config.txt	/zoom/load 1.0
Save the contents of Memory to performance_config.txt	/zoom/save 1.0

### CONSOLE DEBUG FUNCTIONS

Print Status Information	/zoom/survey 1.0
Print the list in Memory to the Console	/zoom/list 1.0

### ZOOM ROLE KEY

0 USERROLE\_NONE  
For Initialization

1 USERROLE\_HOST  
Role of the Host

2 USERROLE\_COHOST  
Role of Co-Host

3 USERROLE\_PANELIST  
Role of the Panelist, Only in Webinar Mode

4 USERROLE\_BREAKOUTROOM\_MODERATOR  
Host role in Breakout Room

5 USERROLE\_ATTENDEE  
Role of Attendee

## COMMANDS COMING OUT OF ZOOMOSC ON PORT 1234

### OUTPUTS

Provides ID of User when they turn on their video	/zoomosc/vid/on x
Provides ID of User when they turn off their video	/zoomosc/vid/off x
Provides the Gallery View Order by sending a list of IDs mapping from top left to bottom right	/zoomosc/vid/galorder [x1,x2,...]
Provides the Quantity of Participants with video on	/zoomosc/gallery/count y
Provides ID of User when they turn on their sound	/zoomosc/sound/on x
Provides ID of User when they turn off their sound	/zoomosc/sound/off x