OBS + ZOOM

NAVIGATING AN UNHAPPY MARRIAGE

+

ZOOM

CACHE_MISS

INTRO FLUFF

This guide goes over advanced steps to improve OBS & Zoom integration. In particular, it demonstrates a technique for organizing your Zoom setup that lets you quickly switch where people appear on-screen during a stream.

Zoom + OBS are a weird sort of pair. On the one hand - Zoom is the industry leader in terms of call quality, background noise filtering, consumer accessibility, scale, stability, and more. It basically became a household name during the pandemic, and it remains - by far - the easiest and most ubiquitous solution for video conferencing as of writing.

OBS, meanwhile, is a great platform for creating broadcasts online. Because it's free, widely supported, and well-documented - it's easy to pick it up and use it to create a broadcast that looks as-good as anything you'd see from people with budgets running into the thousands (assuming you have the technical and artistic skills necessary).

But combine them together, and things get… messy. Zoom doesn't export its video participants in any easily-consumable format to other applications, so OBS is stuck capturing the Zoom chat window to bring the callers' video feeds into a stream. But this creates issues - the broadcast operator needs to crop all the individual windows by-hand - creating duplicate screen grabs and manually cutting it down to just the right video feed, doing resizing manually, and so on. If a caller drops, everything moves around physically on the original screen grab - videos shift into a different layout and potentially grow to take up more space. If a new caller comes in - same problem but in reverse..

This results in many streams, especially early in production, having obvious minor technical issues. Screen crops that put participants in weird frames, subtle errors of a few pixels due to manual adjustments of cropings from stream to stream, pre-stream confusion or delays as the operator needs to adjust all the window captures, and thick tangled nests of sources that just kinda get tossed into the scene whenever the operator can't find what they need. And of course - if a caller drops permanently, it's usually a 3-4 minute drop to a holding screen while the operator changes everything for the new participant layout.

Fortunately, there's a better solution. With up-front investment, a broadcast operator can create a system of croppings, nested scenes, and macros that allow the thole thing to operate in a relatively turn-key fashion, which preserves previous work, and facilitates making quick adjustments or new scenes while maintaining the same look, cropping, and flexibility they had previously, without having to redo any work.

Goals

By the end of this guide, you should understand how to add Zoom to OBS in a way that doesn't suck and minimizes unprofessional-looking scene adjustments mid-stream resulting from dropped callers. To adjust scenes, you'll press one button to set the number of callers, and then press 3 buttons for each caller to assign them to a character scene.Prereqs

This guide tries to assume very little. It will walk through most of the steps required, although a baseline understanding of "what OBS is" and "why you would want to use it" are of course required.

You need to use OBS or OBS.live for most of the main parts to make sense. You can use StreamLabs OBS (SLOBS) but…. Don't. It's a slow, buggy mess - more-so than normal OBS, and it doesn't work well with broadcasts as complex as those created by this guide. The small niceties (undo, numeric control of crops) don't pay for themselves after accounting for the significant amount of lag and operational hassle introduced by SLOBS.

You need Zoom installed, and setup to show participants in gallery mode. You'll also need the installed version, rather than the "App" verison if you're running in Windows (this guide hasn't been texted on Mac or Linux). This guide demon-

Fig S-1: An example overlay

Protip

Add frames to your image - avoid just 'cutting holes in a pretty background'. If it helps, you can separate your "overlay" into two components: one that goes on top (containing the frames and what not) and one that goes in back (containing the things that are pretty to look at)

Add some subtle animation to the background to create a more engaging visual experience. The name of the game is 'understatement' - avoid visually distracting animation

Figure out in advance where you'll put things like "alerts" (if you have any) and make sure to clear up space for that - we put ours over the bottom left window, which was otherwise used for displaying chat, toggle-able visual aides, and other misc info.

strates how to tell the difference later on. Note that you can use other video call software (e.g. Discord) - although you're on your own for adopting the guide accordingly.

Finally, you need to use the Streamdeck software - either via the physical device or with your phone. The software saves a *ton* of hassle, so it's not hard to justify - if you can't, or don't want to, buy a full hardware streamdeck, the phone app is $3/mo after a free trial at time of writing, which basically pays for itself if you're doing anything even modestly complex.

Note that you can still squeeze value out of the ideas provided in this guide by skipping the Stream Deck platform and building your own macros (e.g. with Auto Hotkey) - this just won't help push you over the finish line.

Setup

First thing's first. If you haven't already, get yourself a Stream Overlay. This is basically "A pretty image you place over the cameras on the stream layout". Fig S-1 shows an example from How Not to End the Word.

The cameras go in the wooden frames and the opaque rock texture (I have the two layers as separate PNG sources) and we use a background movie of some clouds looping behind the rock texture.

From here, we'll walkthrough adding the bedrock layers of the scenes that eventually get stuffed into our camera frames. We have one scene for each character - the goal is to fill the entire scene with "visually valid" material, as well as a "backdrop" for when the person isn't present. This will, down the line, minimize the amount of work you need to do to fix a dropped caller.

I'll be referring to the people on the zoom call as "characters" from here out, since I acquired these techniques while working on TTRPG streams.

1. 'Character' Scenes

The first part is simple: Create a scene for every character you expect to regularly appear on your stream, yourself included, until you have something that looks like Fig 1-2. Make sure you use descriptive names for your own benefit.

Next, add a simple "color source" to each scene that fills the

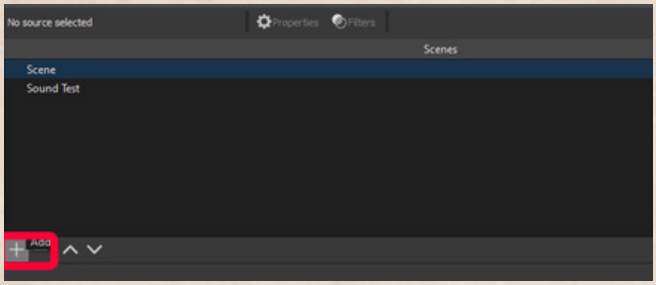


Fig 1-1: Add a scene by clicking the "plus" sign under the "scenes" view at the bottom of OBS

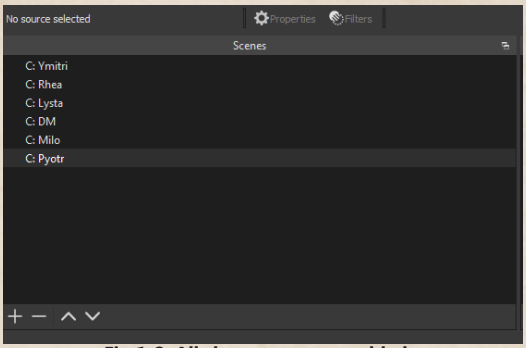


Fig 1-2: All character scenes added

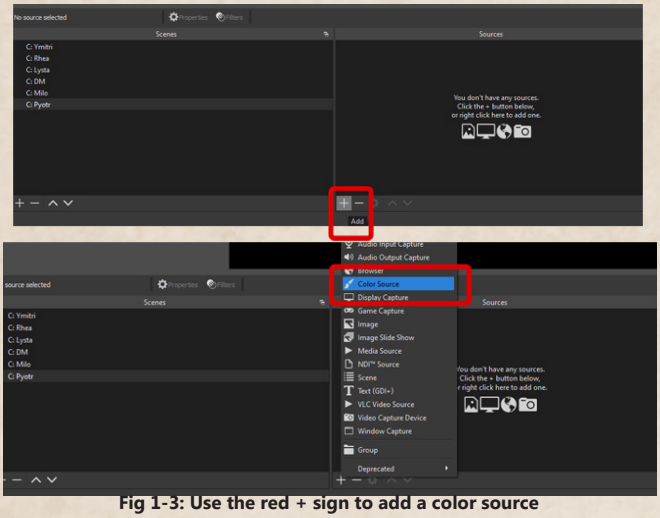


Fig 1-3: Use the red + sign to add a color source

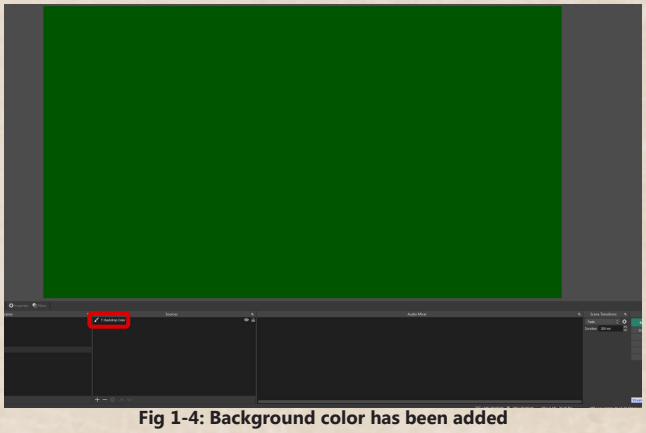


Fig 1-4: Background color has been added

PROTIP

Use prefixes in your scene & Layer names to keep things organized, and to ensure you know what that scene's for without having to look. This guide uses "C" for characters.

whole scene as in Fig 1-3. We use this to ensure that, no matter how awkward the Zoom crop, you will fill the frame on the "main" scene we'll be constructing.

When you're done, you should have something that looks like Fig 1-4.

Step 2: "Local Player" scene

This **optional** step can help make things a *little* nicer if you have a local player - e.g. someone whose voice and video are easiest to capture through the computer running the broadcast. If you export this camera to Zoom, and then pick it up from there, you'll often end up with a lower-quality video stream due to the downscaling Zoom performs. Not a big deal in 99% of situations, but if you want to make the camera big, front, and center - it'll help out significantly.

Create another scene (prefixed again with "Z") This will be our first "source" scene. If you have a camera from which multiple programs can read at the same time (e.g. an HDMI capture card) add a "Video Capture Camera" as in Figure 2-1. Otherwise, you may want to acquire software like ManyCam, which facilitates streaming the same webcam footage to both Zoom and OBS.

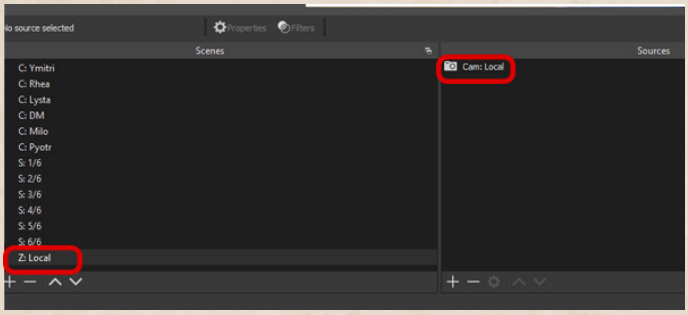


Fig 2-1 - added "video capture camera"

Step 3: Zoom Crop Scenes

For this step, we create the other Zoom scenes. These are the things that actually crop out the Zoom call. You'll create one Zoom cropping for each potential caller position in the gallery view. If you're not using Zoom, you can do a similar thing - but you'll need to ensure that the caller positions all match up for your given software. The key is that, from call to call - if you have the same number of participants, the video feeds show up in the same locations and same sizes (not necessarily the same order).

Anyway, we're going to create one scene for each zoom video stream. In other words - if we plan to support a maximum 6 callers, we'll make 6 scenes: "Z: Zoom 1" through "Z: Zoom 6". These scenes each correlate to a particular video feed, which I personally number from left to right, top to bottom - no matter how many callers or whichever layout (you can use whatever numbering system, as long as it makes sense to you). See Fig 3-1 for an example numbering of a 5 person call.

In each of these scenes, we'll include window captures of Zoom for each set of total participants. So if you're dealing with the scene for character 3, you'll cover the bottom middle of a 3 person call, the bottom left of a 4 and 5 person call, and the far bottom left of a 6 person call.

I'll include a quick table here for 6 callers fullscreened to a standard 16:9 monitor - larger caller counts or different window sizes will need adjustments.

Fig 3-1 - Example zoom numbering

Scene/Caller	2 Callers	3 Callers	4 Callers	5 Callers	6 Callers
Zoom 1	Left	Top Left	Top Left	Top Far Left	Top Far Left
Zoom 2	Right	Top Right	Top Right	Top Middle	Top Middle
Zoom 3	N/A	Bottom	Bottom Left	Top Far Right	Top Far Right
Zoom 4	N/A	N/A	Bottom Right	Bottom Left	Bottom Far Left
Zoom 5	N/A	N/A	N/A	Bottom Right	Bottom Middle
Zoom 6	N/A	N/A	N/A	N/A	Bottom Far Right

Table 3-1 - Location of each video feed for a given number of callers

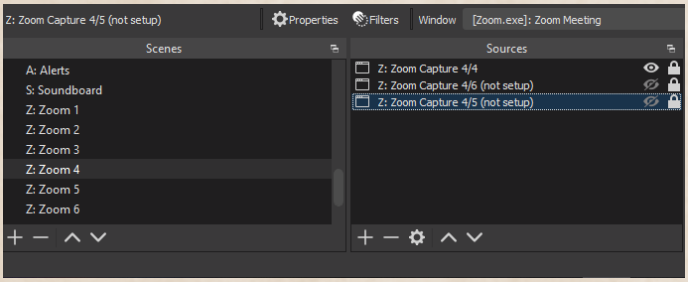


Fig 3-2 - the 6 Zoom scenes you should have (don't worry about the sources yet)

Don't worry about actually adding any captures or performing the cropping now, but table 3-1 sketches out the location of a video feed of interest for each zoom caller count & scene pair for a call on a 16:9 full-screen window. We use "top/bottom, left/right" when there are two rows/columns video feeds to choose from, and "far left/right, middle" when there are three columns (there aren't ever 3 rows with only 6 callers)

OOF, that's a lot of croppings. Fortunately - you do not need to crop all of these before your stream - just enough to where you feel comfortable with what's going to be displayed. I personally left most of them (esp. 2 & 3 caller captures since we had a 6 caller stream) for the first time they became relevant. Likewise, some more astute readers might have noticed that some scenes have the same video in multiple locations - e.g. "Zoom 1" for 5 & 6 callers. These only need one source, which saves some time. When you're done, your scene list should look like Fig 3-2.

Step 4: Filling out Character Scenes

This step focuses on making it so we have a "scene" that we *know* later on will contain the video feed for a specific character when we place it on the top level.

When thinking about how this operates, we have 3 layers:

- **Zoom scenes**: These focus on zoom croppings
- **Character scenes**: These provide the video feed or image representation of a specific character
- **The "Main" Scene[s]**: The top-level scene or scenes which OBS displays directly

In this step, we hook layer 1 up to layer 2. This will, once we setup our streamdeck, let us quickly move which cropping maps to what player off camera, and quickly shift the number of croppings we expect when a player drops.

To do this, we take all of the Zoom scenes and toss each of them, wholly into each of the character scenes. We show you the buttons to click in Fig. 4-1 and 4-2, and give an example of what this looks like when finished in Fig 4-3.

Do just one scene first by hand, then copy-paste all of these "Zoom Scene" sources into the other Character scenes. Be sure of the following:

- **DO NOT**: Change any of the Zoom scenes. You crop at the bottom level (inside the zoom scene) to ensure the changes make their way to all other scenes. Scenes should be placed unmodified into character scenes to minimize the risk of mistakes.

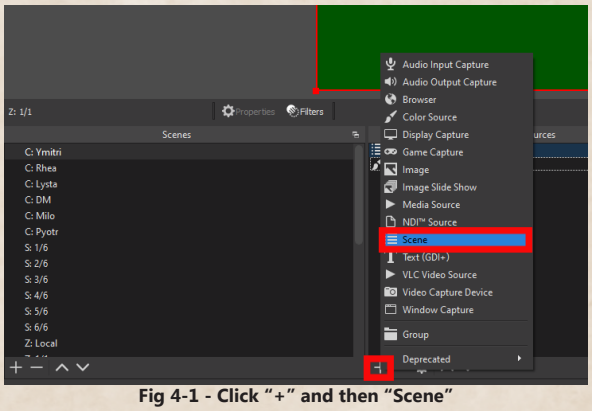


Fig 4-1 - Click "+" and then "Scene"

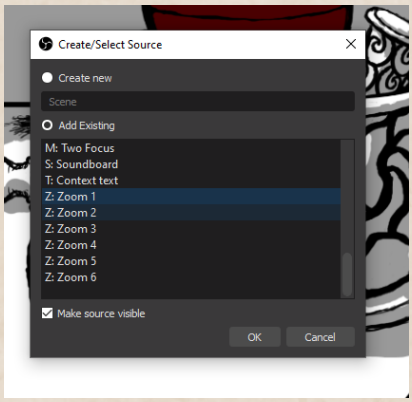


Fig 4-2 - Adding a Zoom scene to a character scene

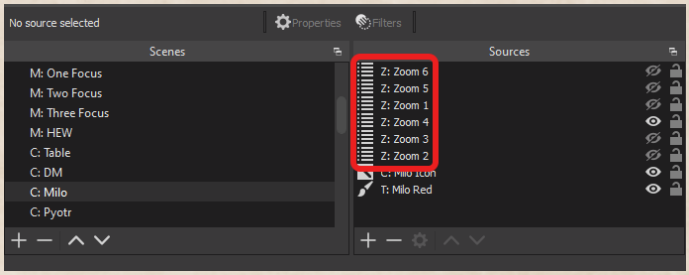


Fig 4-3 - All of the zoom scenes added to a Character scene

- **DO**: Count your scenes before copy-pasting them to make sure the number you made is equal to the number in your scene list. It's easy to forget to add one or two zoom scenes when entering them in manually.
- **DO**: Add in the "Z: Local" scene here if you have one
- **DO**: Ensure your "Zoom" scenes all show up above the "Color" source you added way back in step 1 in the source list. Otherwise, your color layer will be painted over everything else. This is a quick drag-and-drop to fix

Step 5: Adding the "Zoom captures"

This step adds all the appropriate screen grabs - or, at a minimum, adds a screen grab for when you can use it. In this guide, I'm screen capturing off of a full-screen Firefox window displaying an example Zoom screen, but you'll want to capture from Zoom directly for anything you'll use in the real world. I suggest, at this point, you get a call going by yourself so you can set the window target appropriately.

Figures 5-1 through 5-3 walk you through the first screen grab, where we make "Z: Zoom Capture 1/2" for Zoom: 1. Of partic-

ular note - you'll want to make sure the Window name shows "Zoom.exe" rather than "Firefox.exe".

When you make the screen capture, make sure to tick "capture cursor" OFF in order to prevent OBS from showing your cursor traveling over the window. This can happen even when you have Zoom hidden and unfocused behind several other windows, so it really will save you some weird moments if you turn it off here.

Once you're done with that, Figure 5-4 shows how to have the source "fit to screen". You'll want to do this to avoid any centering or scaling issues.

Next, we need to do this for each of the remaining potential video feed locations for a given caller. Table 5-1 shows the sources you'll need for each scene, based on table 4-1. Each column represents a source you'll need to make, and denotes the caller count & locations it can cover (e.g. caller 1 in a 3 person call (1/3) shows up in the same spot as caller 1 in a 4 person call

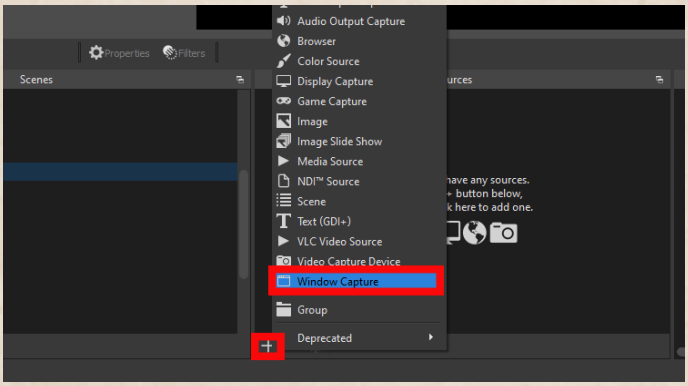


Fig 5-1 - Add a Window Capture object to your sources

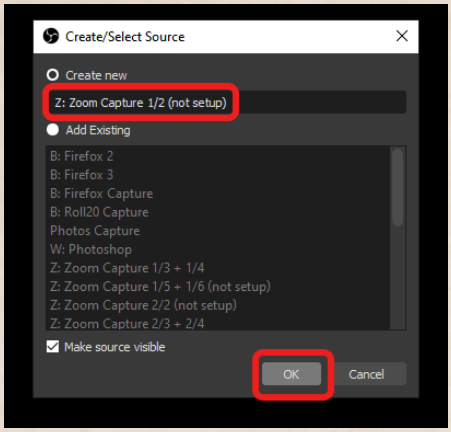


Fig 5-2 - Adding a new zoom source. I use '(not setup)' to mark whichsources still need cropping

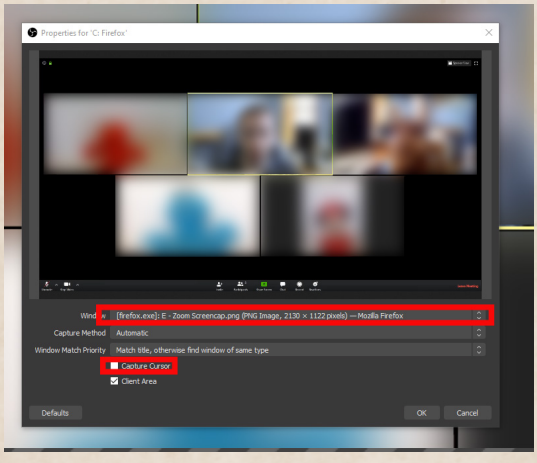


Fig 5-3 - Configuring the screen capture. Make sure "Capture Cursor" is off!

(1/4)

Step 6: Test Cropping

We're going to do some manual cropping here to give ourselves something more interesting to look at in the main scene. You do not need to do this step now if you don't mind having boring things to look at for your screen test.

For this, we'll need to get a Zoom call going with the number of callers we expect to run with most-often. There are a few options for doing this:

Get a bunch of computers you have control over on the same call. This could be laptops, tablets, phones, whatever - If you have a few computers scattered around, you can use "Google Remote Desktop" to control computers remotely. In any case - start a meeting, grab the "invite" link, send it to each device, and set each thing's camera to show something. I personally suggest stupid reaction gifs or memes using something like ManyCam or SnapCam or whatever.

Actually start a call with your group. I hate dealing with other people's schedules and availability, but you do you.

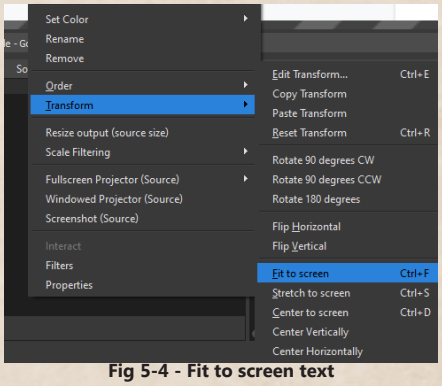


Fig 5-4 - Fit to screen text

For this demonstration, I use secret option #3: Using a pre-blurred, full-screened image of a Zoom call with the correct number of participants thrown up with Firefox. Don't do this unless you want to re-crop everything later - I

FAQ: Zoom Problems

Having problems with Zoom screen capture (e.g. blank screens)? Ensure you're using the Zoom desktop program, not the Windows 10 app. If you're using the right thing, it'll use the window in Fig 5-5.

Assuming that's the case, you can follow up in Appendix A for the next debugging steps.

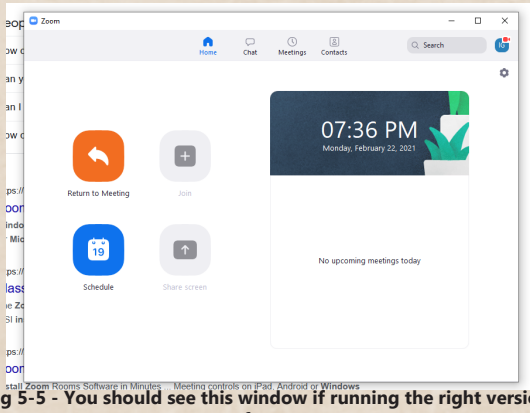


Fig 5-5 - You should see this window if running the right version of zoom

Caller Count	Expected Croppings (Equivalent Crops share the same cell)		
Zoom 1	1/2	1/3 + 1/4	1/5 + 1/6
Zoom 2	2/2	2/3 + 2/4	2/5 + 2/6
Zoom 3	3/3	3/4	3/5 + 3/6
Zoom 4	4/4	4/5	4/6
Zoom 5	5/5	5/6	
Zoom 6	6/6		

Table 5-1 - Croppings needed for a given caller number

can only get away with this because I won't be using the final product in a stream.

Once you have all that going, it's time to start cropping!

Switch to a Zoom scene for a given caller (e.g. Z: Zoom 1 for the first caller)

Since you already picked "Fit to Screen" in step 6, this process is easy. Click on the source (either in the sources list or on the preview) to bring focus - you should see a red outline with four "boxes" dotted around. Hold alt, and click and drag the bottom middle box up. This should begin cropping upwards. Do that until a bit of the bottom of the camera you want to crop to gets chopped off. **Don't aim to be perfectly on the edge** - you need to crop a bit in anyway to remove the yellow "current speaker" border anyway, and it's better to crop too far in than not crop enough! Instead **Aim to be a few pixels inside the camera**.

Do this for the left middle box, then the right middle box, then the top box. As you do this, the camera should dynamically resize as the proportions change, and once you're done dragging the top box down, the video should *mostly* fill the scene.

I say mostly, because there's a 99% chance the crop is *slightly* off the exact 16:9 proportions. To fix this - hold shift and drag the bottom right corner down a bit to scale the source proportionally until the black bar (either on the bottom or the right) falls off the scene.

Do the same for the top left box for the other black bar. Once you're done, you should have something that looks like Figure 6-1.

Do the same for the appropriate capture in "Z: Zoom 2", "Z: Zoom 3", etc until you have finished. If you screw up during the cropping and drag the frame a bit off-center, no worries! Just ctrl+f that bad boy and it'll re-center and re-size itself.

And of course, if you're following my convention of tagging all Zoom window captures with "(not setup)" until you've cropped them, now is the time to update the name.

Fig 6-1 - Fully cropped Zoom window capture - showing a blurred out camera from the screen shot earlier

Step 7: Macros! Part 1

Next up we have the thing that will make this *really* streamlined - macros. Specifically - streamdeck macros.

Now - you don't need to use the streamdeck app to accomplish this task - you can do similar with AutoHotKey (not explained in this guide as of v1.0 - suffice to say, "google it") or some heavy-handed macro programming… But you'll have the easiest time with Streamdeck.

Caution: If you skip this step

Make sure each "Character" scene has it's background color or image exposed by clicking the "eye" icon on the source list next to each "Z: Zoom x" scene.

This will make sure you have something stable and sensible to crop for the next step.

Anyway, Get your streamdeck and accompanying software installed, plugged in, and ready to go. Once you have the windows app up, setup is simple, if labor-intensive. It consists of two sets of macros:

- A set that turns window capture sources on and off at the "Zoom" level, so only valid crops show up given a specific number of callers
- A layer that lets you toggle zoom scenes per-character, so you can just turn off the one they're not on, and turn on the one they are on.

Lets say you have 4 callers, vs 6 last stream. You'd press the "4 caller" button to deactivate all the crops that assume 6 callers, then activate all the crops that assume 4 callers. Then, you'd go through and turn off the "Z: Zoom x" source that contained their feed last time, and activate the "Z: Zoom y" feed that contains their feed for the new call.

FAQ: Source Problems

What if you open Stream Deck, create your OBS Scene action, and it doesn't display any sources for you to disable?

The Stream Deck program needs OBS open so it can read the available Scenes, Scene Collection, and Sources of each scene. Make sure to open OBS before continuing with Step 7.

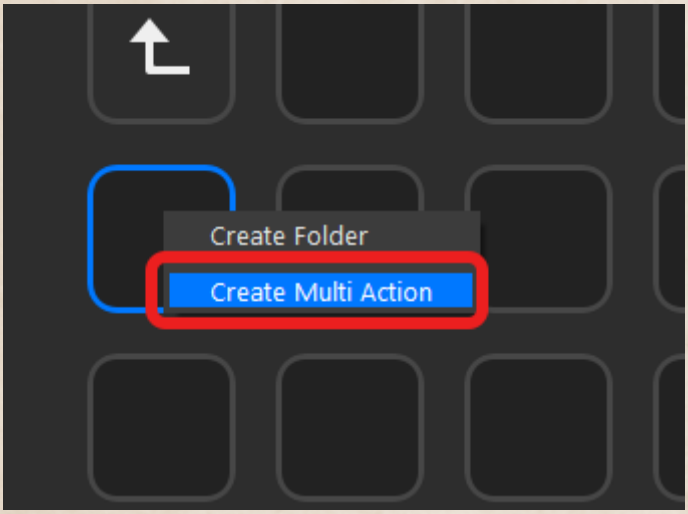


Fig 7-1 - Right click to create a Multi Action

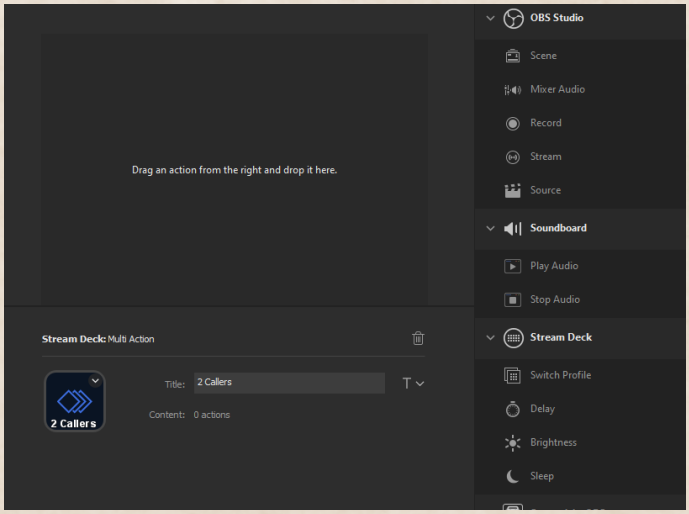


Fig 7-2 - the Multi Action setup screen

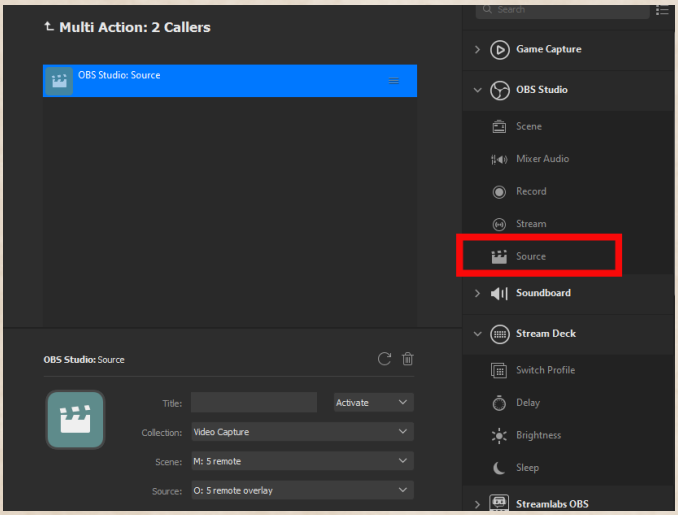


Fig 7-3 - Adding a new source to the Multiaction

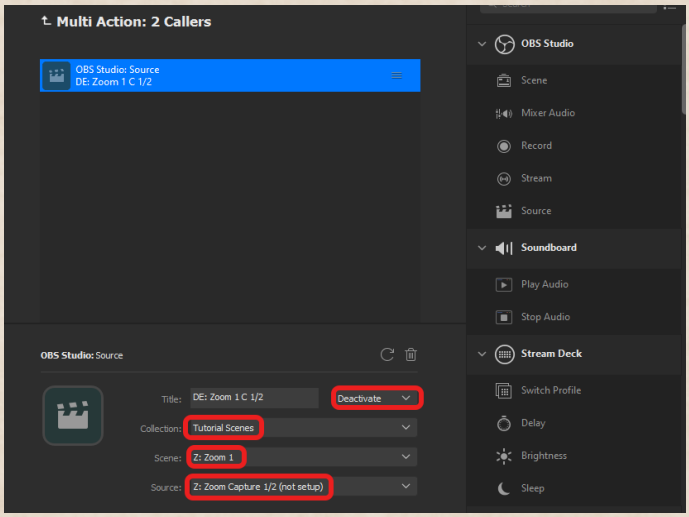


Fig 7-4 - Configuring the source macro

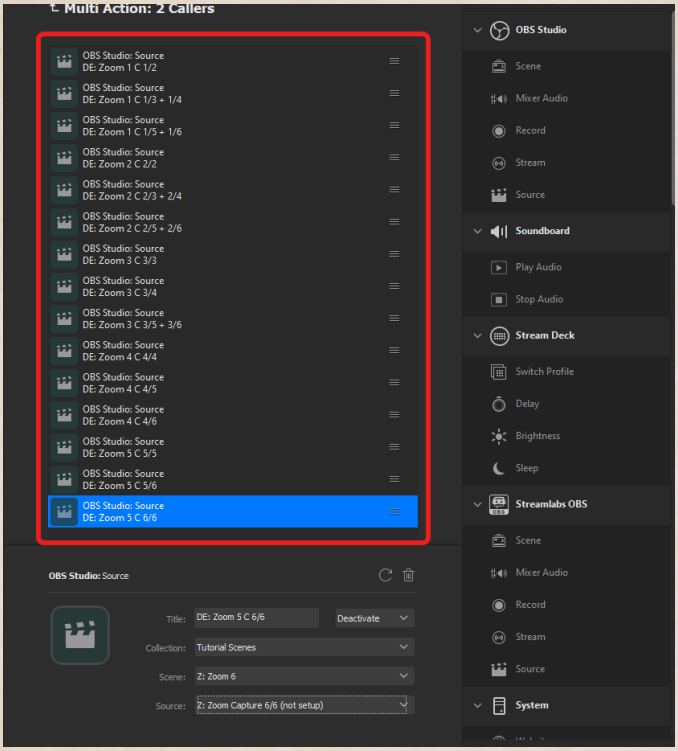


Fig 7-5 - All of the Zoom sources setup correctly

Lets start on the first layer. Open up the streamdeck software from Elgato's site on your computer, create a new folder (right click an empty box & click "create folder"), and hop into it.

Create a new "multiaction" (Fig 7-1), title it "2 callers", and double click to select it (Fig 7-2). Twirl open the "OBS Studio" layer (if you haven't already, make sure you've installed any relevant OBS plugins for streamdecks) and drag a Source command onto the list (Fig 7-3)

We're aiming to turn off all Zoom screen captures that don't pertain to "2 visible callers", so make sure the "Collection" is set to your current collection, "Scene" is set to the first Zoom scene ("Z: Zoom 1"), and the Source is set to the first capture source ("Z: Zoom Capture 1/2"). Finally, switch "Activate" to "Deactivate". Fig 7-4 shows a source with all the field filled in correctly (note again that I tag captures I haven't finished cropping with "(not setup)" so I know they'll need adjustment before using them on-stream)

Go ahead and use copy-paste to create actions for ALL ZOOM SOURCES in ALL ZOOM SCENES - including the ones you want turned on at the end of the day. Deactivating all Zoom sources is a little slopp, but makes it much easier to safely copy-paste this group later without risk of mistakes or oversights. I've gone ahead and titled my actions in 7-5 to reflect what each action does ("DE" == Deactivate, then the Zoom scene number).

Ok, so you have your macro list of *deactivations* taken care of. Now, click out of that macro and copy paste the action 4 times - 1 for each other caller count you want to support (3 callers, 4 callers, …) as seen in Fig 8-6

Now for the "turning stuff on" part. Open up the "2 Caller" macro again (double click) and add another "OBS Source" - this

CAUTION: Check Your Work

The Deactivation macros here and later on in the guide are most annoying yet easy step to get wrong, due how you need to repeat something same-y looking 15+ times. STOP and double check your work using the following strategies:

- **Count the deactivation actions**. Does the number match up with the number of sources on OBS? (15 if you set it up according to table 6-1)
- **Check through scenes** in each action. Are any repeated?
- **Manually turn on all Zoom sources**, then run the macro. Are they all off now?

This will make your life MUCH EASIER later down the line. The problem with this setup is that, if you miss a single source or get something *slightly* wrong when turning sources off, you may not notice for a very, very long time due to coincidence and happenstance (e.g. that source is always hid behind another, you never need to use that particular macro, etc). Since you copy-paste the macro you just created multiple times, if you screwed up somewhere, not only will you need to find and fix it - but you'll need to remember to find and fix it across all the macros you make, or redo this step entirely.

Doing due diligence now will save you a huge amount of hassle later when the problem inevitably comes up mid-stream and you need to debug live, 3 months removed from this guide, with only half an idea of what might be wrong.

Fortunately, the "activation" side of each macro is really easy to test, catch, and correct - and thus warrant less caution.

time, to ACTIVATE a source in the "Z: Zoom 1" scene corresponding to the expected number of callers (in this case: "Z: Zoom Capture 1/2"). Figure 7-7 demonstrates what this should look like.

Repeat this for each macro - the '2 callers' macro should activate "Z: Zoom Capture 1/2" in "Z: Zoom 1" and "Z: Zoom Capture 2/2" in "Z: Zoom 2", the '3 callers' macro should activate sources for "Z: Zoom 1" through "Z: Zoom 3", and so on. I've provided a screenshot of '4 callers' for clarity (Fig 7-8)

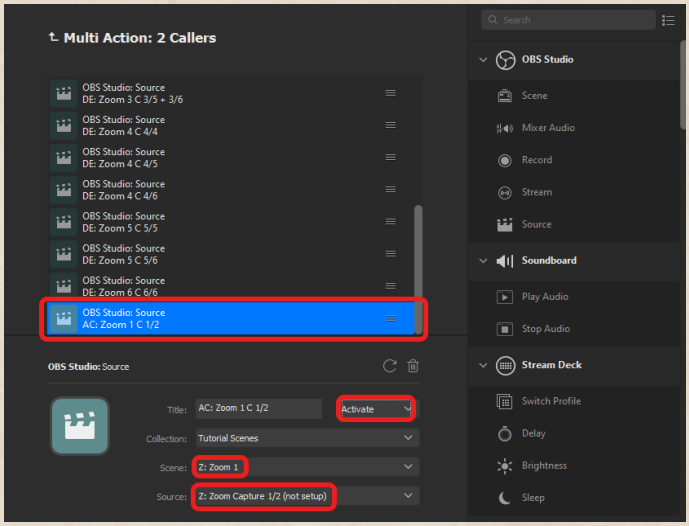


Fig 7-7 - Create a new source activating the '1/2' crop on Zoom 1

Step 8: The Other Boring Macro Part

Ugh, more macros? Yep, more macros. Don't worry - after this, we get to see payoff.

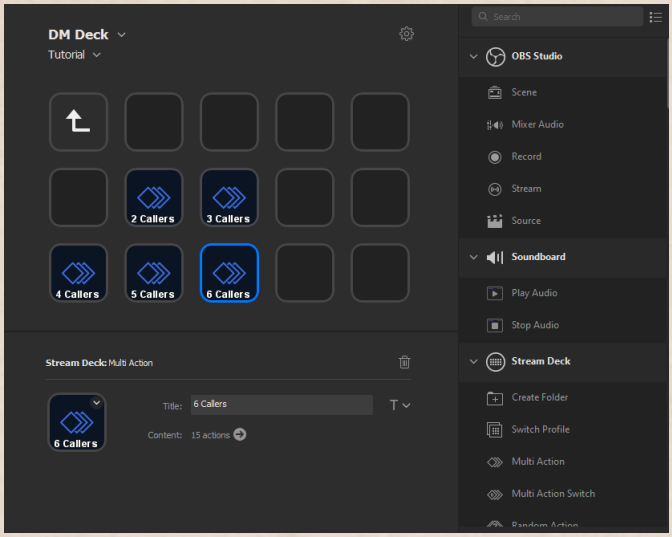


Fig 7-6 - The duplicated caller multi-actions

Caution: Macro Order

Be careful to ensure all "activate" macros come after the "deactivate" macros. Putting them further up in the list is a great way to mess up (although, it'll be a pretty easy issue to catch and fix).

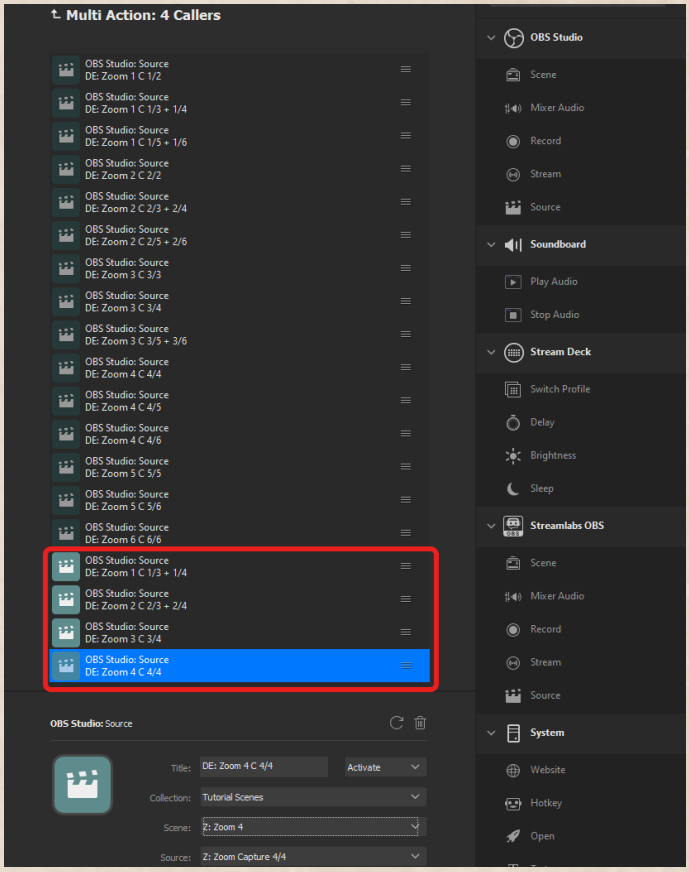


Fig 7-8 - The 4 caller macro, fully setup

We're going to be doing something like what we did in step 7, but for that second set of macros we talked about.These will be a bit more tedious, a bit less copy-paste-able, but at the end of the day it'll be worth it.

STOP

You just did a lot, it's testing time again!

- **Click the "2 callers" macro** (to make sure everything on needs to be turned off), **then the "6 callers" macro**. Are only "Z: Zoom Capture x/6" sources turned on?
- **Repeat for 5 callers down to 1.**
- Did everything that needed to be on get turned on?
- Did everything that needed to be off get turned off?

If you do this right, you should never have to worry about accidentally leaving in weird croppings on-stream (e.g. whatever's in the location of a 6th caller's video after pressing the "3 callers" button). Likewise, getting it down now minimizes the risk of problems on air - so it's doubly worth being thorough.

First, create a folder for your first character (Fig 8-1). Then, create a button that controls a Zoom scene-source for that character (e.g. the "Z: Zoom 1" source for the "C: Pyotr" scene - Fig 8-2). Repeat for each of the "Z: Zoom" sources in that scene in a 2x3 grid. Be sure to include "Z: Local" if you made such a scene. Once you're done, you should have something that looks like Fig 8-3.

Rinse and repeat for each of the other character scenes you've made. You can copy the "character macro" folder at the top level for each other character, but you'll have to go through and manually correct each switch so that it's pointed to the right character scene. For instance, if you copy Pyotr's macros for Milo, you'll have to make sure to change "Source: 'C: Pyotr'" to 'C: Milo' for each button in the folder. This will likely cause the action to point at a different source in the scene as well, so you'll want to double check and manually set each action's Source and Scene when copy-and-pasting folders like this.

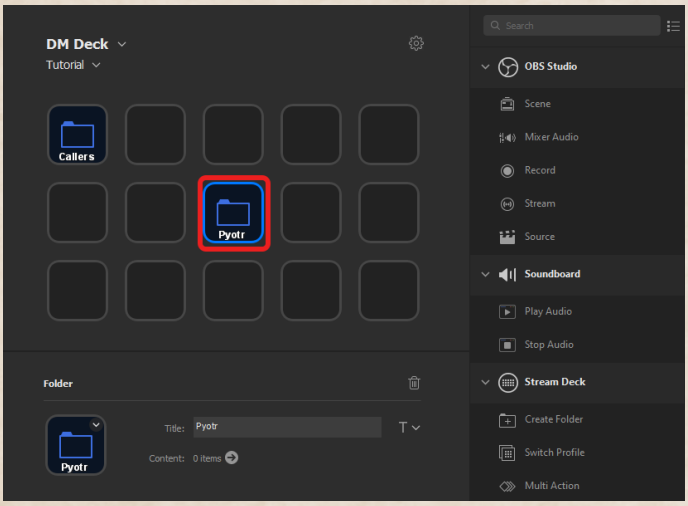


Fig 8-1 - Create a new folder for the character macros

Fortunately - this is much less hazardous than the previous step, and any mistakes you make should be pretty obvious and quick to fix, so double and triple checking your work is less-necessary.

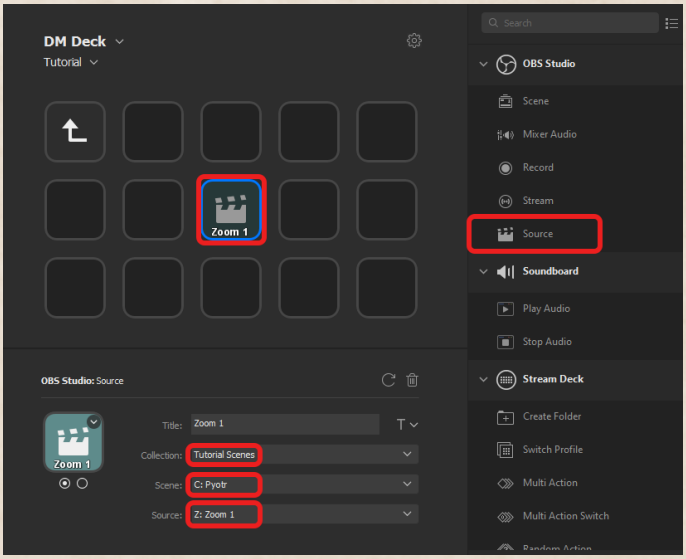


Fig 8-2 - Add an OBS Source action to the grid. Set it to toggle "Z: Zoom 1" on one of your character scenes

Step 10: Filling in the Main Scene

Here's the fun part! Or at least, as fun as "rote stream layout setup" gets, anyway. We're going to add the Character scenes to the top level overlay, which means we finally have something we could stream with!

This is pretty straight forward: For each Character, decide where in the main overlay they should appear, add their "C" scene to the main scene, crop, and move to the next one - for this guide, we'll be cropping the DM scene. Because I'm suffering from a

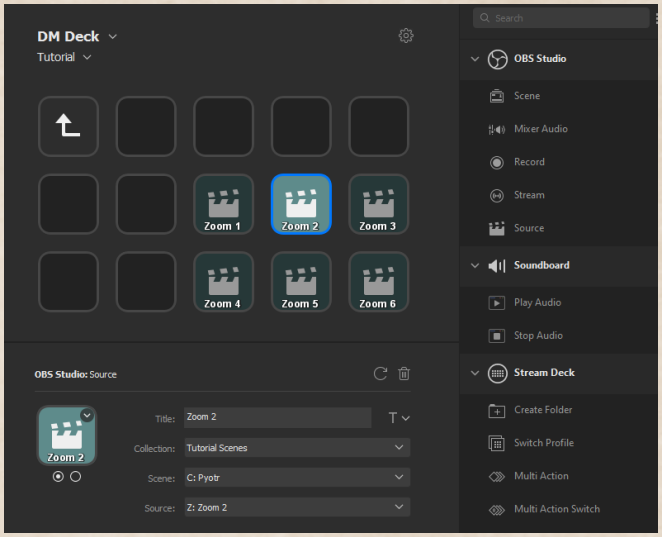


Fig 8-3 - All of the Source toggles added. Note how Zoom 2 lights up when the source is "on" in OBS

Protip

If you figured out symbols or art for your characters (or, maybe you got Emoji for frequent on-stream personalities) you can make your streamdeck more visually interesting, and easier to navigate at a glance, by switching the icons of the folders to those images.

At the top level (e.g. Figure 8-4) - click on a folder, and on the bottom left, right click on the icon in the box. Click "set from file" and then open up the image you used and - voilà - interesting-looking streamdeck layout.

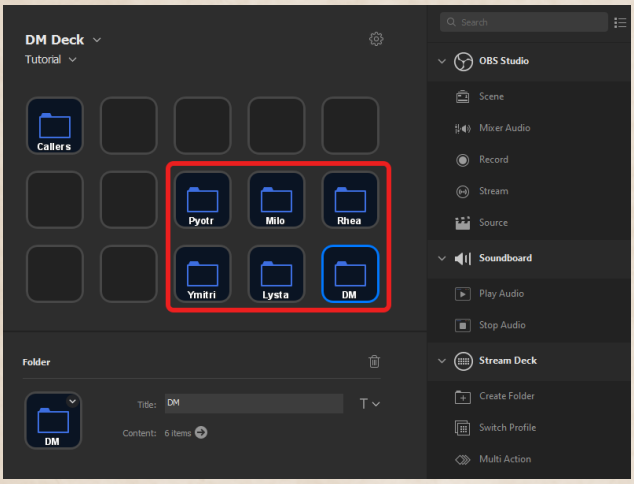


Fig 8-4 - All character folders created and setup

chronic case of "pandemic laziness", I've skipped the part where we get a zoom call going, and instead we're going to be adjusting the croppings just our Character backgrounds.

Doing it with Zoom to work off of is similar - it just has the additional steps of "calling people", "pressing the macro to set the number of callers", "going through each character you're setting up to make sure they have a valid Zoom call", and "not qualifying as an anti-social shut-in". (That last one's the hardest part)

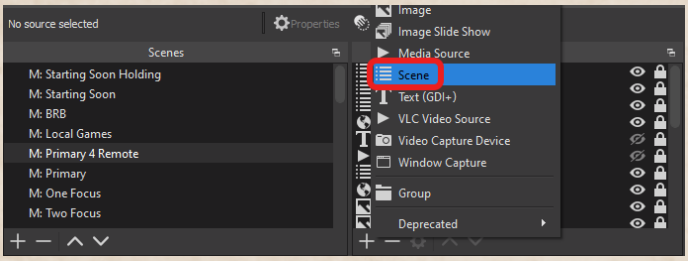


Fig 9-1 - Add another scene in the usual way

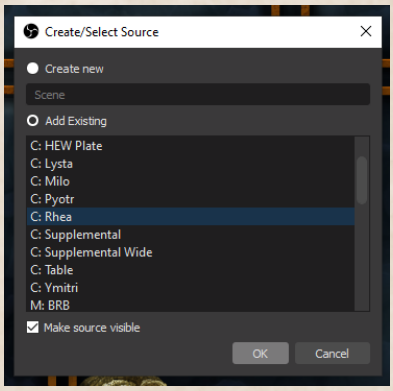


Fig 9-2 - Add a Character scene. We're setting up "Rhea"'s character this time, who'symbol is a crashing fist-wave

To do this - just add the Character scene from previous steps to your main scene (fig 9-1, 9-2). Ensure it's near the top of the "Sources" list so it appears above everything else (this will make cropping easier) To crop so they fit in a frame, you can follow this 4 step process:

Protip

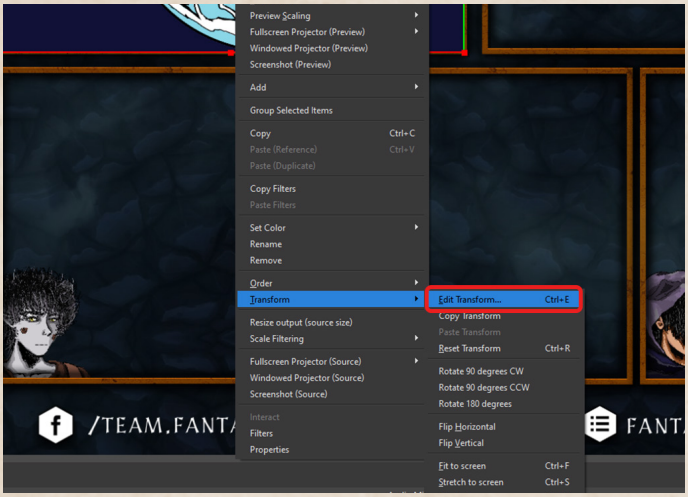
Think of your characters' frames like their cubicle - largely the same from person to person, but it's fun to decorate a little so each is uniquely "theirs". Here are some ideas:

- Adding in little banners or lower-thirds with names
- Adding character art, a stream logo, an icon, or another iconographic representation of the person
- Stats, counters, health, or other information from the current game (you'll want to look into editing these with text documents or other tools, rather than OBS directly)

The key is really just to be uniformly different - everyone should get the same "sort" of decorations, styled to their person

Fig 9-3 - After scaling the Rhea scene from full portrait. We make sure it's aligned with the top left of the portrait we're sticking it in, and then scaling it in by dragging the bottom left "red box" until the height or the width matches our target size (in this case, the height)

Fig 9-4 - With alt held, click on the red box in the center of the overhand, and drag it in to crop to the appropriate size of the frame



9-5 - Right click on the source, and click on Transform > Edit Transform (or just press ctrl+e on windows)

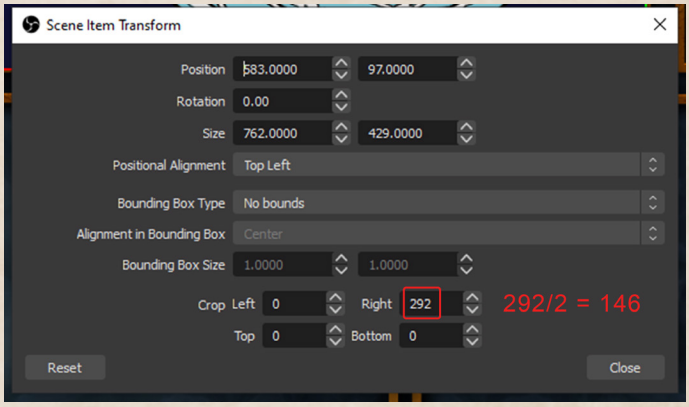


Fig 9-6 - In the edit transform dialog, find the number for the edge you cropped and divide it by two

- Fig 9-3: Fit 3 adjacent borders to the frame (I start from the top left corner, then shrink in from the bottom right until it's the right width or height)
- Fig 9-4: Hold alt, and click on red box centered on the source boundary's overhanging edge. Scale inward until all edges match the frame. Make sure you can see the edges of the frame you're cropping to, but not the insides
- Fig 9-5, 9-6, 9-7: Go into the numeric adjustments, divide the only "crop" number in two, and split between that and the other crop direction (e.g. "left and right" or "top and bottom"). You should now have something that looks like Fig 9-8.

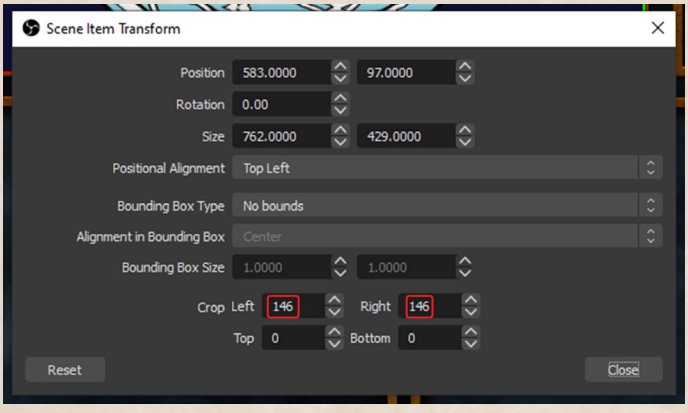


Fig 9-7 - Take the result of 9-6 and set as the number for the left & right crop (or top and bottom if you're working on a frame that's squatter than your source)

Protip: Zoom & Alerts

When dealing with Zoom & OBS, your screen gets cluttered quickly with different camera feeds. Many amateur streams try to blow up these feeds as large as possible, taking up all or almost all of the screen realestate.

Resist the temptation, however, to plaster big ugly alerts ontop of everything! Set aside null space in your overlay to handle alerts et al. This might be your upper/lower third (for instance, roll-over text when someone subscribes) - or it could be an entirely separate area.

For HNEW, we carved out a window on the bottom left - it contained visual aids (maps, pictures, etc), then Chat above that, then alerts above that. This made sure nobody was unduely covered up by a sub event, but we still gave space for contributor recognition.

Fig 9-8 - A fully cropped scene, before we've moved it to the back of the source list

- Last thing's last, drag the source under the overlay (whatever frame you're cropping against) - I like to keep all of my character sources in a row on the sources list

Do this for each of the scenes, and then behold your creation! (fig 9-8).

Once you're done, you should take a sec to play around. Both to check and make sure everything works, and because... well, I personally find playing with the result of hard work gratifying :)

Conclusion

If you've been following along until now, you should finally have a Stream-Ready layout to play with. When you're getting ready to go live, you'll need to do a few things:

- Press the "caller" button corresponding to the number of people you expect this stream
- Go to each caller, and set their "Zoom" source (1 through 6, corresponding to their caller count)

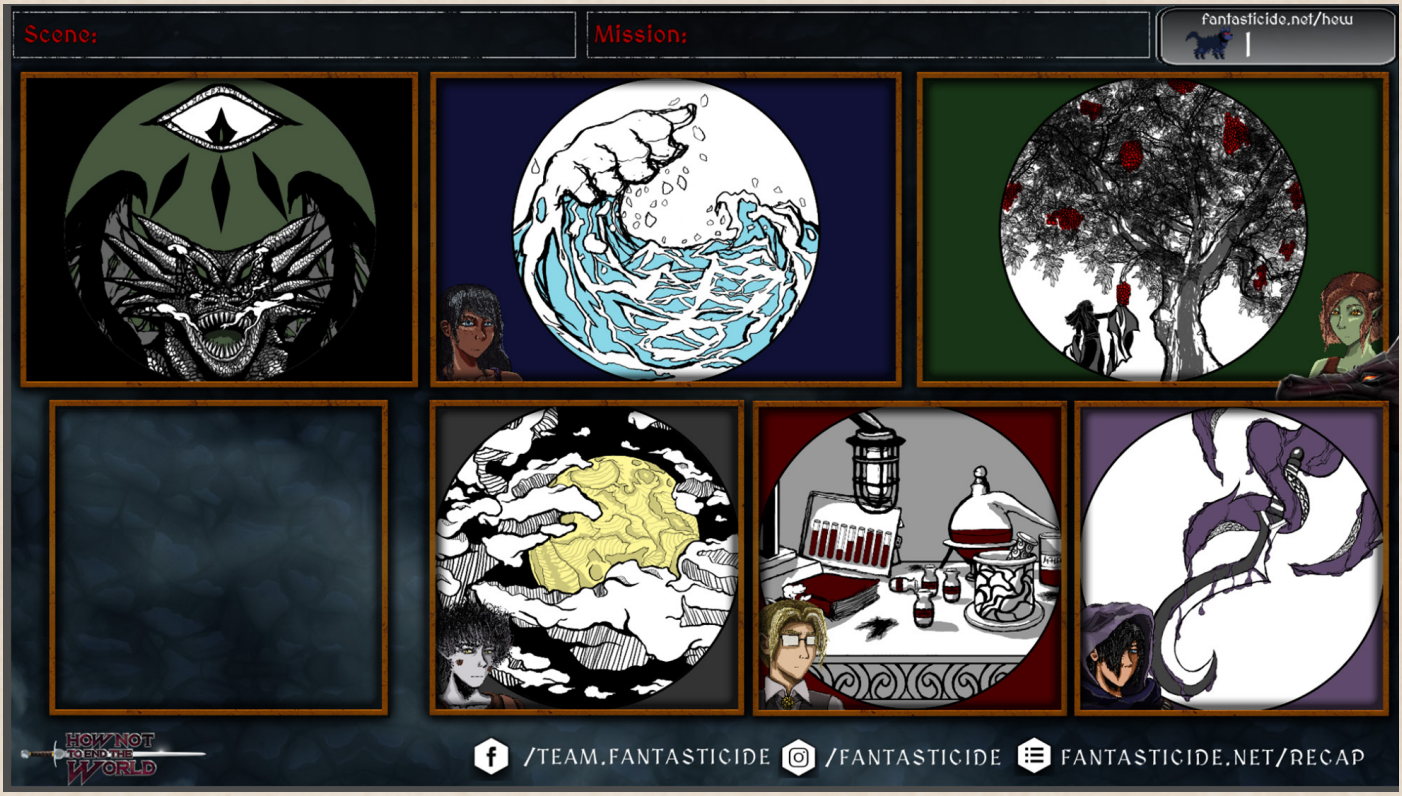


Fig 9-9 - A fully setup scene for all the characters

- Verify that the video feeds you're using for that caller count (the sources in the "Z: Zoom x" scenes) have been previously setup
- Do the cropping on the "Z: Zoom x" scenes if they aren't

If someone drops during the call, you should be able to quickly hit the "caller count" button on your streamdeck to fix all the croppings, and then remap the remaining callers - ensuring that the person who dropped has their "official background" showing in case they come back. (You can have players rejoin with video-off if you want to avoid further disruption, and just update crops during a break).

I hope this guide helped you through the process of making OBS & Zoom play together a bit nicely. If you have questions or feedback, please let me know via email: igorham000@gmail.com, or discord: segphault#0901 (Yes, I have too many screen names, deal with it).

In the mean time, happy streaming!

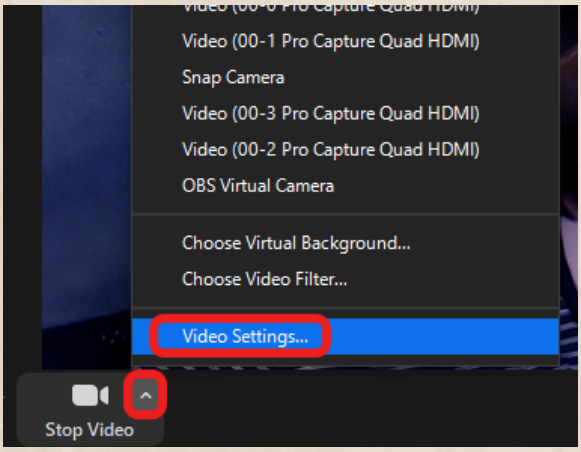


Fig A-1 - How to get to the video settings from a meeting

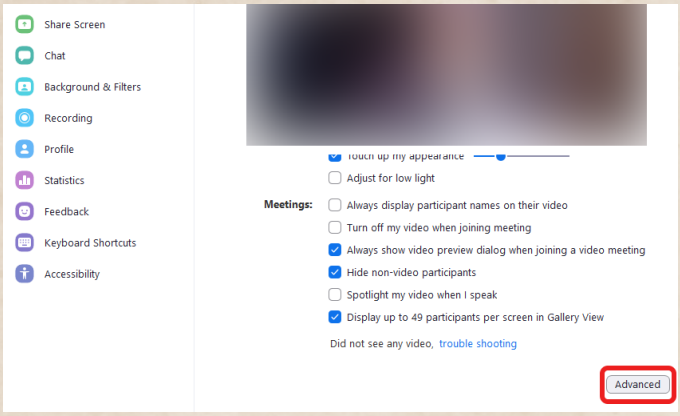


Fig A-2 - The 'advanced' button is at the bottom of the Video settings

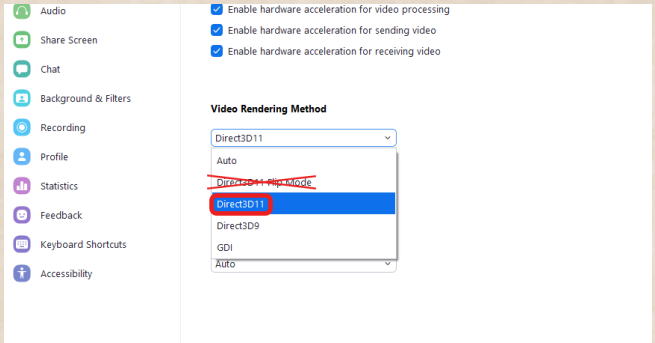


Fig A-3 - Pick Direct3D11 or similar

Appendix A: Zoom Capture Problems

Ok, so you did all the steps, but you saved testing with Zoom for game day. Everyone's on the call, you start in 10, you fire up OBS for cropping but... OH NO! OBS Doesn't pick up the Zoom window - it maybe picks up the cursor (if you didn't follow my instructions) but otherwise it just shows a transparent void where pretty faces and sparkling personalities should be. What's a streamer to do!?

Well, a streamer is to follow these instructions[1]:

- First: Make sure you're using the exe-installed Zoom desktop app (see Fig 5-5 for a quick sanity check). The Windows 10 app thing (I think it's compatible with tablets and runs through Microsofts app store?) is garbage. Just like all the other "could be an EXE but is instead an app" apps (looking at you Skype). The other tell is that, if two "zoom" options pop up in your start menu, it has a square-ish icon instead of a round-ish icon
- Next: you'll want to open video settings. There are two ways to do this - you can either click on the "gear" icon on the welcome screen, then click on "video" - or, if you're mid-meeting, you can pull up the arrow next to the video icon and click "Video Settings" (Fig A-1).
- In the video settings, near the bottom, click "Advanced" (Fig A-2)
- In the box that comes up, make sure whatever it's set to does **NOT** have "Flip" in the name (Fig A-3). Switch to the remaining "Direct3D" option with the highest number (e.g. "Direct 3D11")
- Restart Zoom. If OBS still doesn't pick it up, restart your computer.

And that's it! If OBS still isn't capturing your Zoom video chat, you'll want to look in other places - make sure the title of the window capture is correct, make sure the source is visible, try creating a new scene collection and capturing Zoom from there.

Legal Stuff & License BS

1 Assuming you're running something that looks like version 5.2.1 (the latest at time of writing)