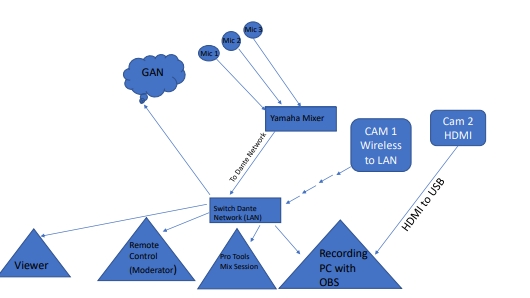
CONVERGING TECH OF AV

BASIC CONCEPT



Developing an AV system for live performance (film or theatre) which enables efficient and versatile transmission and control over audio and video data, whilst minimizing the requirement for post-production.

We to come together and decide on a project idea aswell as pitch the idea to our tutor. Our goal is to send audio data over a Yamaha mixer and the dante network and send video data over wireless and HDMI - USB inputs and compile them into an OBS session which can be viewed and controlled by another desktop. (full details of system in another post)

In class we tested Dante - Protools - Other desktop communications and also looked at teamviewer and vdo ninja for remote and viewer purpose.

AUDIO

Audio Sources would be sent to Yamaha O1V Mixer and then distributed via DANTE LAN network.

A Pro-Tools Session would be set up in order to receive the individual channels for mixing and mastering purposes.

Master output from pro-tols will be sent to OBS Computer via DANTE.

VIDEO

1st Camera set up and DATA Transmitted over HDMI to USB

2nd Camera to Set up wirelessly.

VDO Ninja for mixing received video.

OBS or another studio software to capture the mix.

Remote Control & Moderation

Potential Use of TeamViewer, or Touch OSC on another device to enable live control over the system

Tool Testing

This Week we progressed with looking into our system and troubleshooting some issues that popped up. We found that teamviewer wouldnt work for our remote monitoring purposes as it requires a computer that has it installed even when using the browser version. To resolve this we've started looking into using MAX/MSP on the remote monitoring computer interacting with a MAX/MSP patch on the computer with the Protools session.( <https://www.youtube.com/watch?v=MCXPpROr52k>) Good video example which we will be using as a starting point. Linking with this we also looked at remote controlling the mixer with connection to Protools and Max/MSP we also looked at other methods at controlling the mixer using Touch OSC as a controller and MAX/MSP as a bridge between Touch OSC and the Mixer/Protools. A combination of these different control ideas will be used to highlight the best of each program.

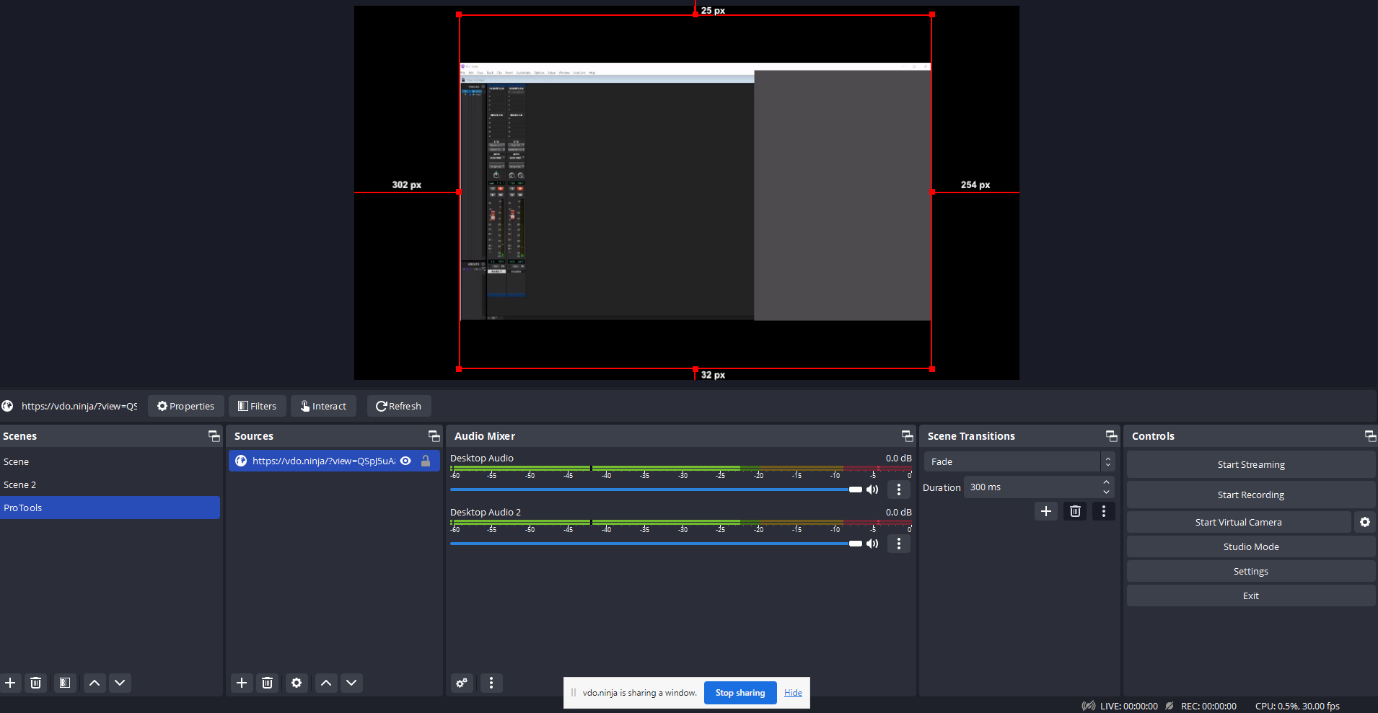
We also continued to look into OBS and other video alternatives VDO Ninja. We used video ninja to get a room setup on the mobile phone and computer webcam. Theres also a plugin for OBS called Touch Meter its used for exporting Audio and Video from OBS into Protools. Our goal for next week is to get more practise with using the hardware and to identify another potential pitfalls that our project could fall into.

This week we made progress on establishing a  bridge platform for our video and screenshare sources on VDO ninja Web application.

VDO. Ninja is a tool which enables remote video feeds into OBS software via WebRTC (Web Real-Time Communication)

We created a virtual room within the appication and allocated each video source camera and screen capture in indvidula scenes.

Each scene has a unique 'solo view' url link which can be copied and shared as  video source on OBS for monitoring and recording.

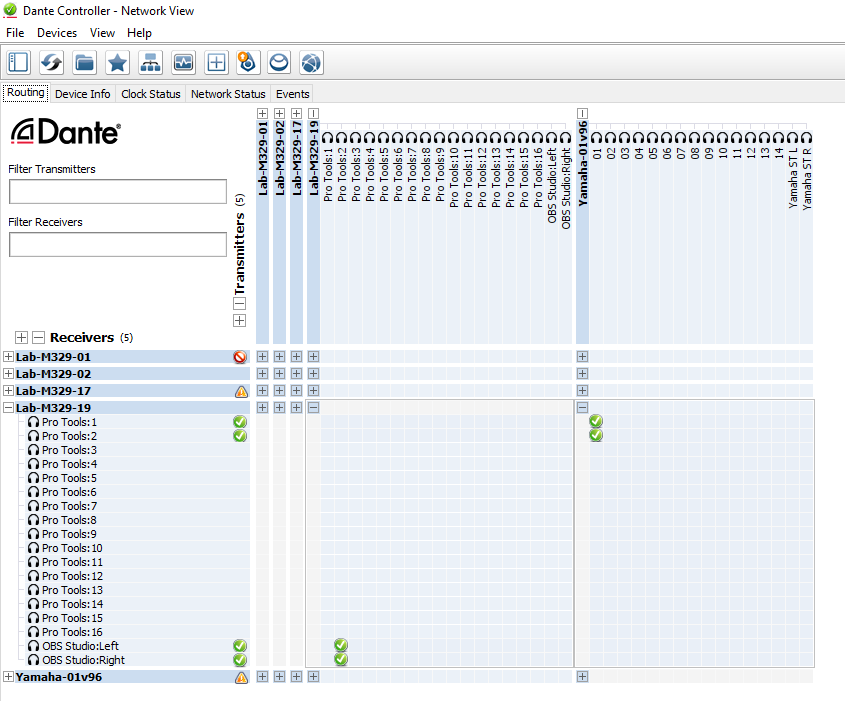


 We set up  3 scenes 1 for each camera and 1 for Pro-Tools mixing monitoring session screenshare.

For our audio sources we are using microphones directly plugged into Yamaha OV1 Mixer.

These Audio Signals are then distributed via Dante Network.

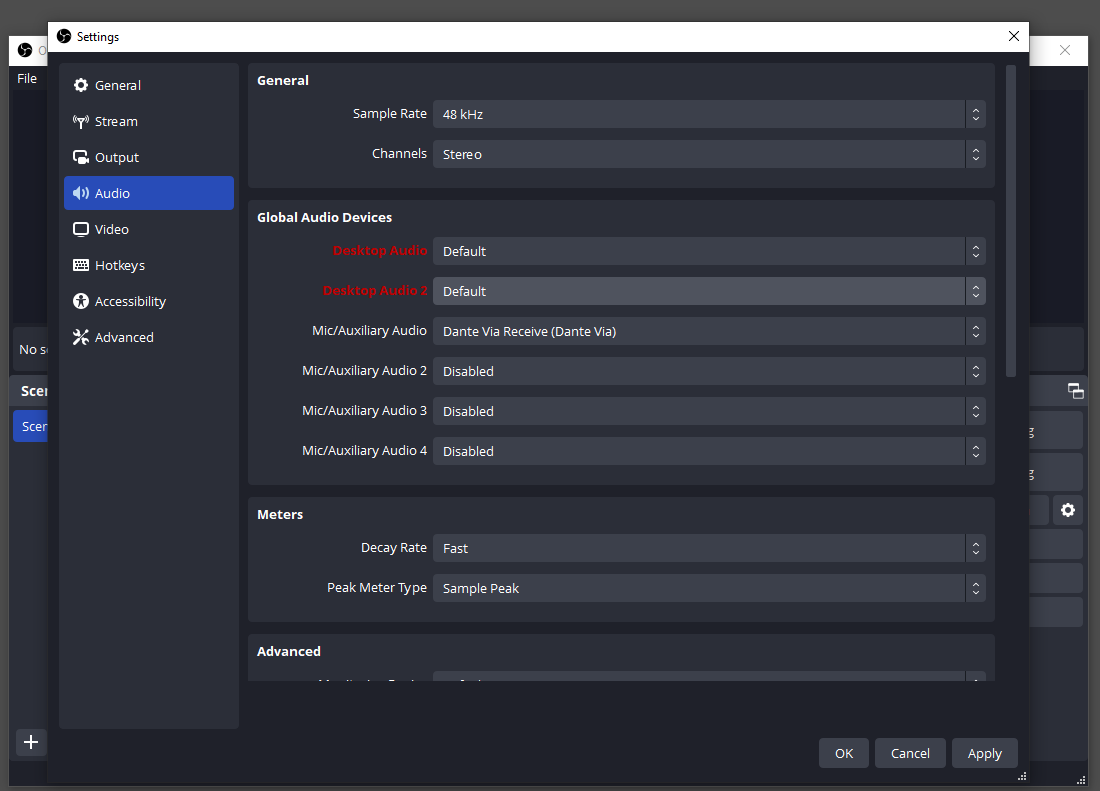
Please see Dante routing example below….



By using Dante controller, we are sending the raw audio to DAW for Mixing/Mastering. (in this scenario Pro-Tools which has to be configured accordingly with audio driver as Dante Via )

The initial problem we encountered with was the fact that OBS cannot receive more than 1 stereo channel from Dante. It has to be 1 Stereo (L/R)

Please refer to OBS Audio Setup Example bellow…



In addition, since Dante system cannot mix separate audio channels into 1 we are using an auxiliary stereo channel which uses buses to blends the audio in 1 stereo Chann

Integrating remote compatability !!!!!!!!!!!!!!

The Next Step of our Project is implementing a remote-control option for our system.

The Application used for remote control is TOUCH OSC

OSC Touch is a modular OSC and MIDI control surface for a mobile device which sends messages over UDP or TCP

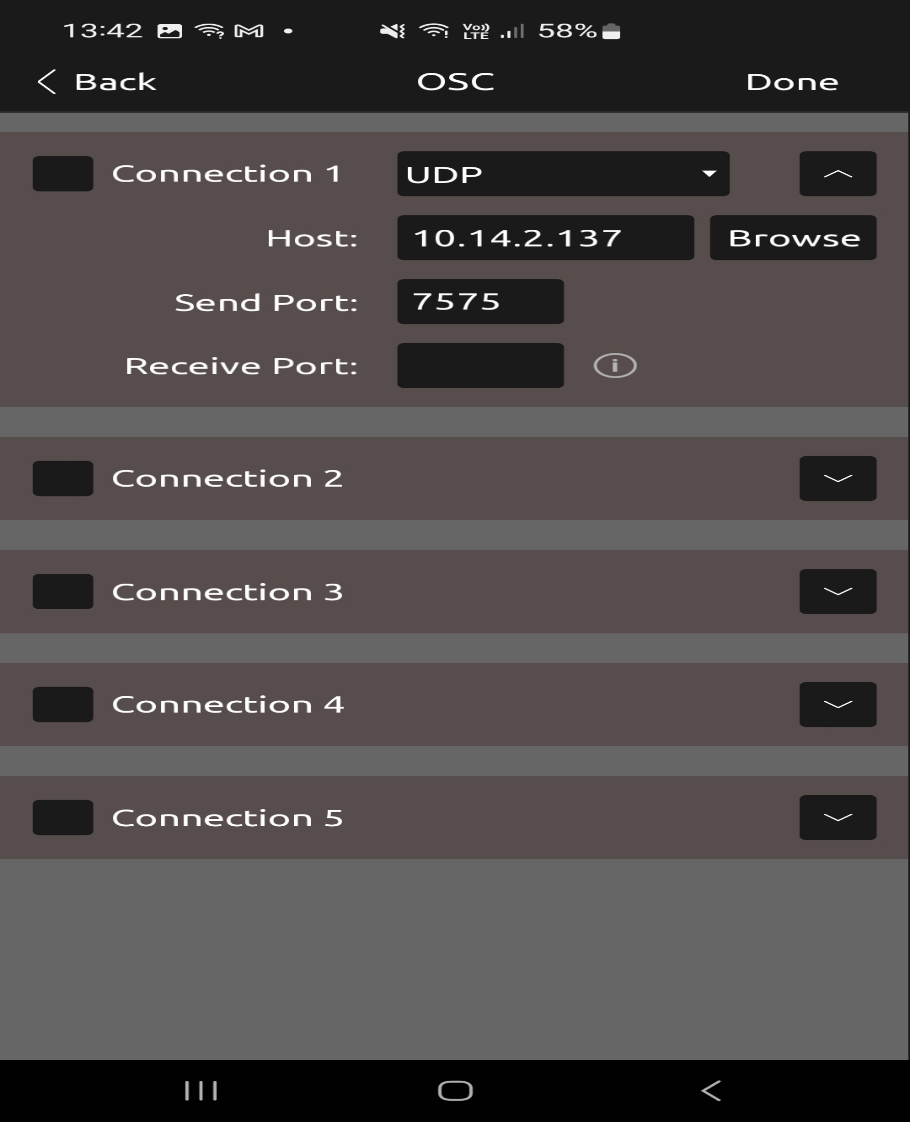
We created a custom control panel with 4 faders. Each Fader has specific address. And a dynamic range number depending on the position of the fader.

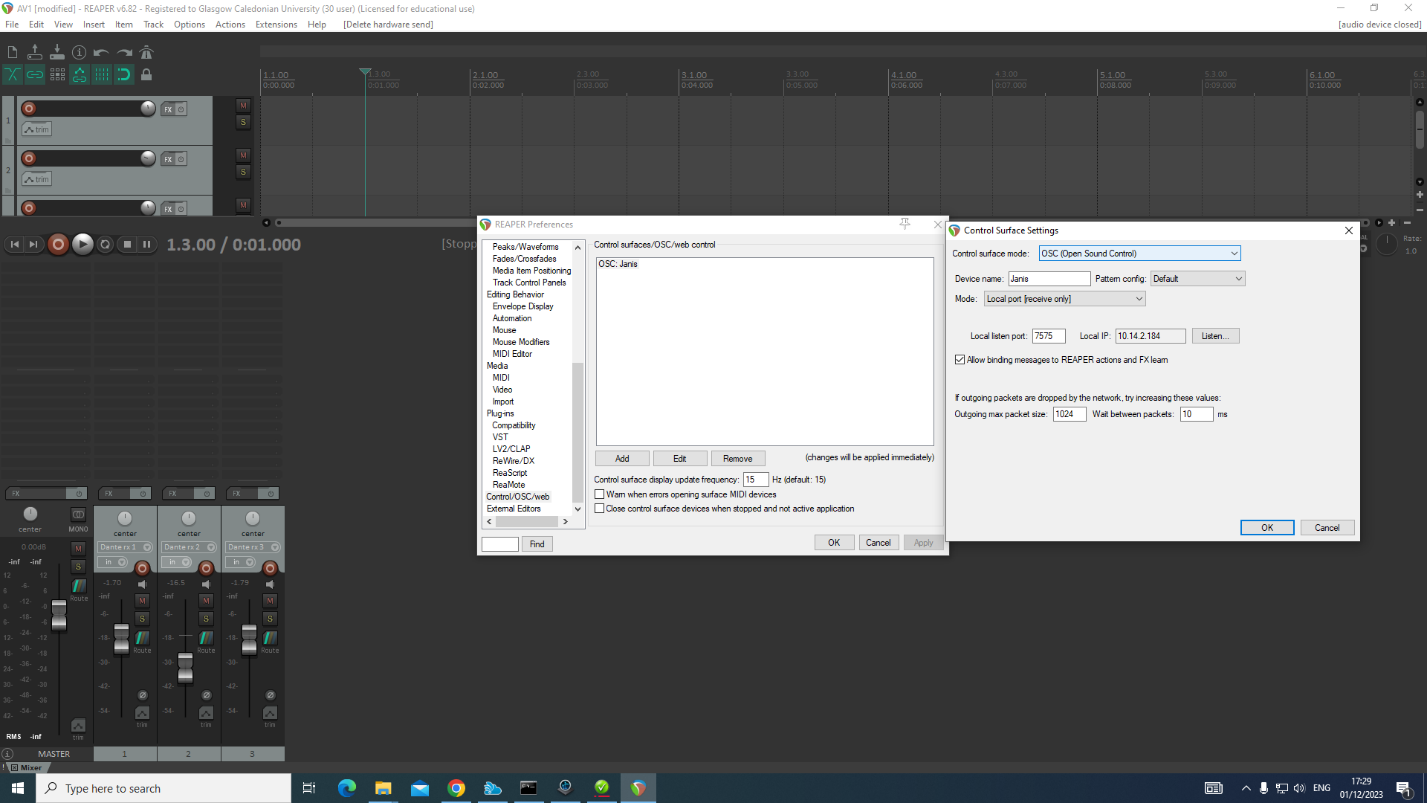
For sending the control data we used UDP Protocol. Since it prioritizes speed and efficiency.

In order to achieve this, we decided to use REAPER as our DAW instead Pro-Tools

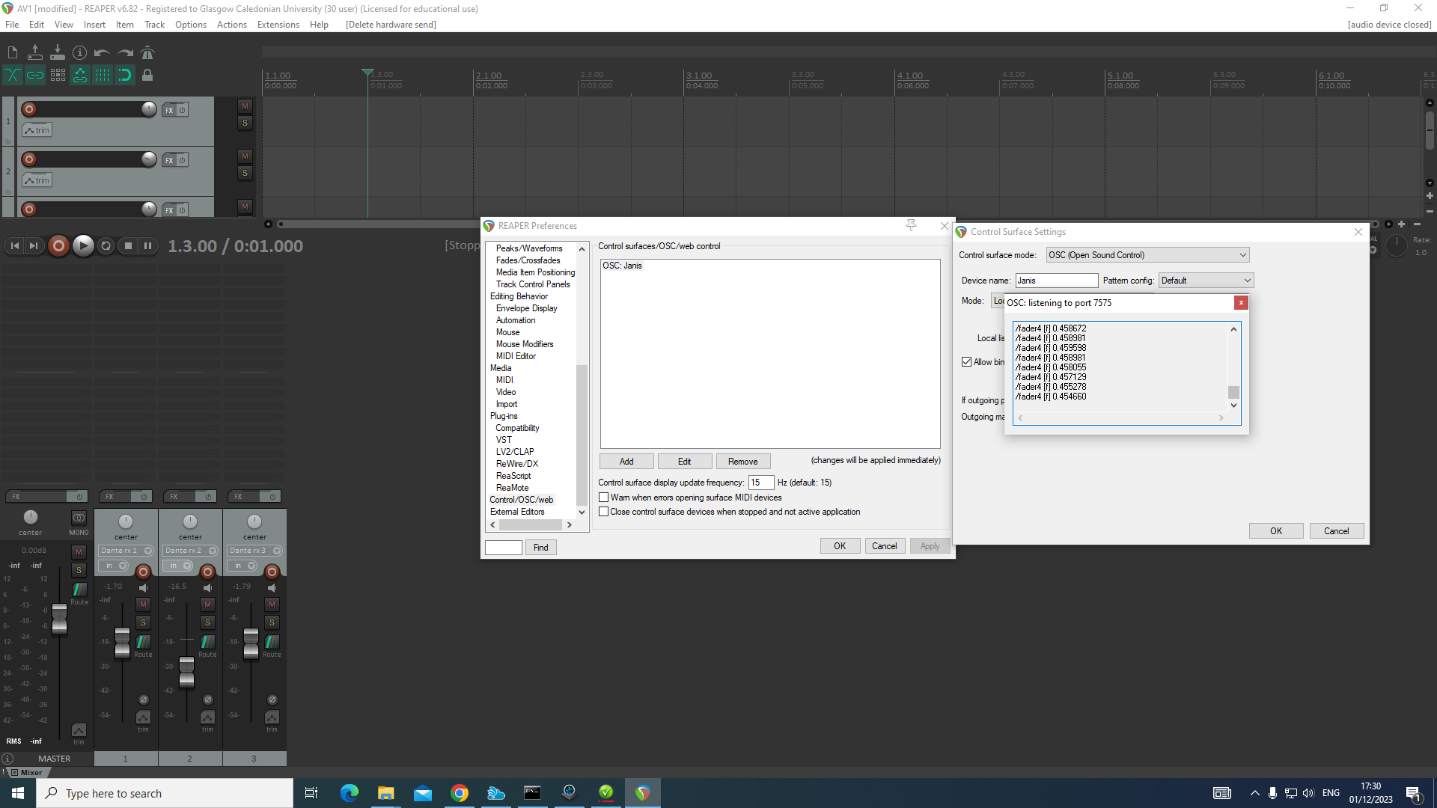
This decision alters the DAW was made due the flexibility Reaper offers in terms of channel routing and more ability to communicate directly with OSC messages via UDP over Local Area Network.

 First we established a UDP connection to The host ( i.e. the PC running REAPER)



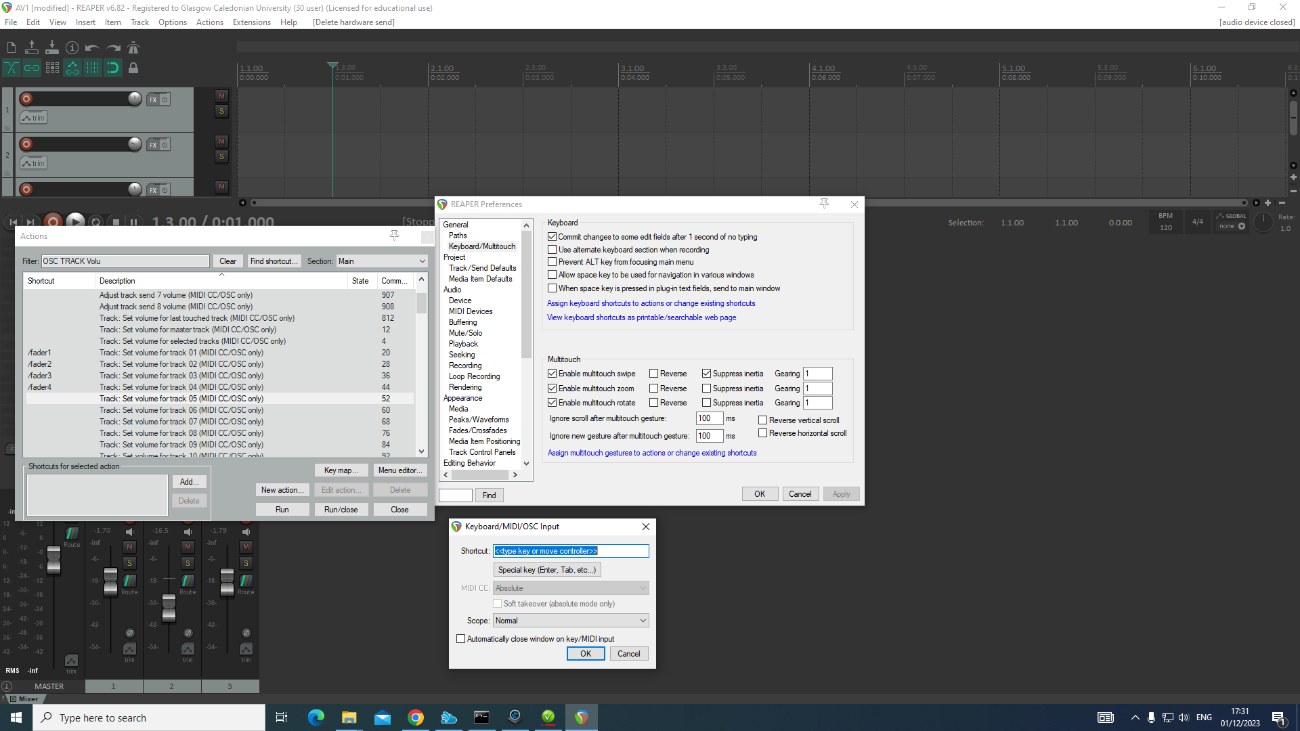
Following this we also set up an UDP receive point on Reaper session with corresponding Local IP and Specific Port.

Once this was done an option ‘’listen to port’’ was used to check if OSC messages are getting through and included appropriate address and a value as per screenshot below



Each Fader value from Touc OSC App was the n asigne to specific controls on Reaper.

Each Indivudual fader has to be mapped seperatley, in our case we mapped faders 1-4 for our 3 audio Tracks and the 4th for our Master Track.



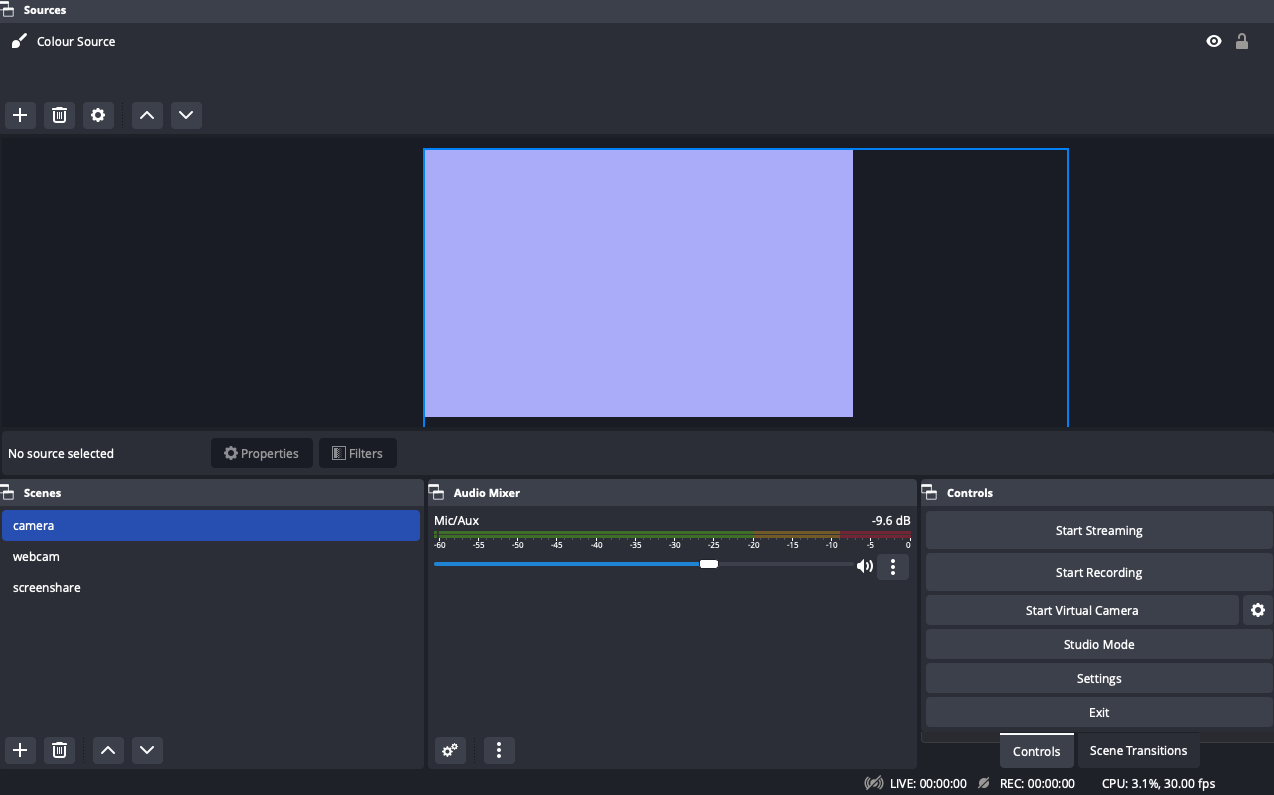
Routing Video and Streaming, !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Following on from implimenting the appropriate routing and remote control, of the audio from the Reaper session, it was time to further assess video routing and possible control options that are available.

As with the remote control of our audio sources, UDP protocols, and OSC data were actualised as an efficient means of sending and monitoring control data relating to our video sources across the network. Making use of the previously mentioned TouchOSC mobile application, there is possibility of interfacing remote OSC commands with our OBS project that could minimise the need for director to remain stationary while monitoring the ongoing project.

Below there is a basic OBS session set up that has one scene, of varied colours for test purposes, in practice three scenes will represent different cameras or incoming streams. Each scene within OBS can have independant audio and video sources assigned to it, either through hardware, software or browser connections.

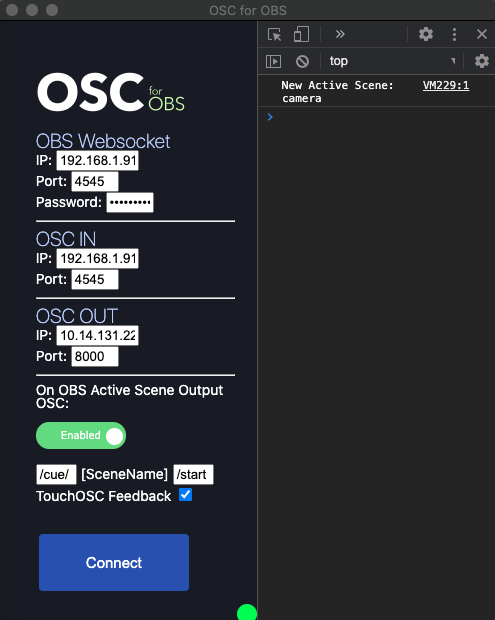
The browser functionality allows for mobile and desktop users to use platforms such as VDO Ninja to stream their audio and video outputs to an OBS session, by using their stream URL, this can then be implemented within the source tab, on the master OBS session, where it can be monitored and processed as required by the session director.

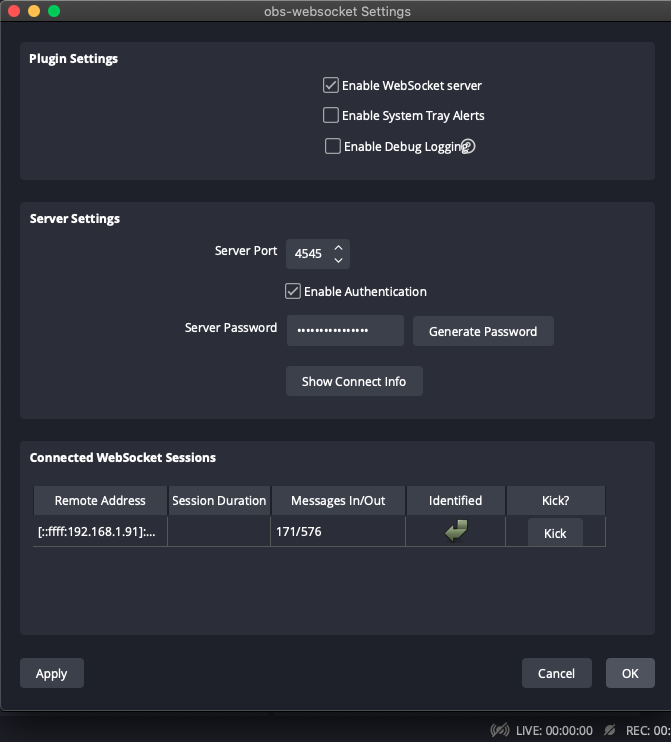


OBS comes with built in websocket functionality, allowing for increased accessibility across devices. This feature in conjunction with the OCS to OBS plugin allows for OSC data to be transmitted across our network, including the OBS session.

Below we can see the OSC to OBS interface on the left, and the details required for its proper connection, from its console panel we can also see that the active scene "camera" from the last image, is recognised as being active by OSC to OBS.

The "OBS Websocket" panel refers to the OBS project directors, IP address, password, and the server port which can be seen and set within the OBS "websocket settings" menu, displayed on the right where your WebSocket session connectivity can also be verified.





The "OSC IN" section refers to the directors IP and port, as this is the address where we are looking to receive our OSC data from any authorized devices.

"OSC OUT" is where we implement any remote devices, this would be set up to on command, issue data in the OSC syntax to our host 'OSC IN', which translates to specific operations within the directors OBS session. TouchOSC provides the perfect interface as buttons can be created and mapped to, on interaction send OSC data from desired ports to required IP addresses, allowing for real time remote control of audio, video and recording processes.

Proof of concept !!!!!!!!!!!!!!!!!!

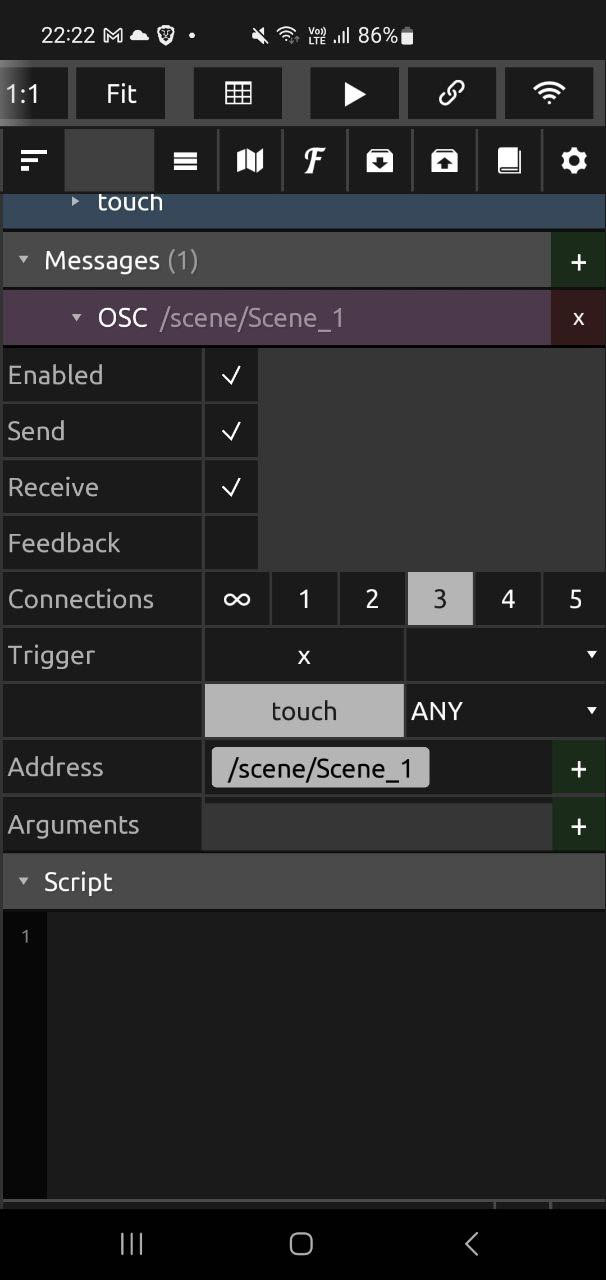
Below is a short video, demonstrating the OSC to OBS software sending OSC commands, and OBS showing it is receiving and understanding the sent data. Although this only proves that the websocket server connection is configured correctly, as it is that sending the change scene commands to OBS.

There is an extensive reference list available online, which set out the guidelines and preset commands that can be called to produce desired effects and proccesses within an OBS project. Showing that it is possible to integrate a mobile device using software such as TouchOSC, in order to allow remote control of the directors OBS session, overall increasing the flexibility of the system.

Test Video !!!!!!!!!!!!! in convergin av folder

To build on this further the Touch OSC app, allowed us to create a custom interface, including buttons and sliders that can send OSC data on interaction, to designated ports and IP addresses, for manipulating screen transitions, recording and volume controls.

Below you can see the settings for the button and interface mapping within TouchOSC, previous to this screen all that has to be configured is where the OSC data from the app has to be sent, i.e the directors IP. As you can see within the address setting window, scene/Scene\_1 would correspond to the scene names Scene\_1 within the master OBS session



This configuration was tested, and it functions the same way as the previous video did, but instead of the OCS tester, the TouchOSC app is responsible for sending the OSC command data. Which is be routed through the OSCtoOBS bridge to the OBS director session, where it can be decrypted and translated into real time processing.

[1] Control and listen to OBS via OSC / jshea2 - <https://github.com/jshea2/OSC-for-OBS>

[2] Touch OSC / Hexler - <https://hexler.net/touchosc>

Presentation !!!!!!!!!!!!!!!!!!!!!!!!!!

Demo VIDEEEO !!!!!!!!!!!!!!!!!!

References !!!!!!!!!!!!!!!!!!!!!!!

**Yamaha OV1 Mixer**

<https://jp.yamaha.com/files/download/other_assets/0/320870/01VE.pdf>

[](https://www.audinate.com/support/dante-via-quickstart)

**Working With Dante Network**

[Dante Via QuickStart | Audinate | Dante Pro AV NetworkingSoftware Downloads&nbsp; &nbsp; &nbsp;Dante-enabled Products&nbsp; &nbsp; &nbsp;Documentation&nbsp; &nbsp; &nbsp;FAQs&nbsp; &nbsp; Request Support Dante Via Quickstart Getting Started Watch this video for a quick product introduction and a tour through the&nbsp; interface and basics of using of [&hellip;]](https://www.audinate.com/support/dante-via-quickstart)

**Touch OSC**

[TouchOSC | hexler.net](https://hexler.net/touchosc)

**Touch OSC for Controlling Reaper Sesion**

[TouchOSC + Reaper. Information STASH - Cockos Incorporated ForumsTouchOSC + Reaper. Information STASH MIDI Hardware, Control Surfaces, and OSC](https://forum.cockos.com/showthread.php?t=261282#:~:text=Go%2520to%2520preferences%2520%253E%2520midi%2520devices,Configure%2520it.)

**Touch OSC for Controlling OBS**

[Free - OSC for OBS AppOSC for OBS An application for Mac and PC that controls and listens to OBS via OSC. Made for live events for triggering and automating cues from an external application (like QLab and TouchOSC) Requires: obs-websocket plugin OSC Commands...OBS Forums](https://obsproject.com/forum/resources/osc-for-obs-app.1222/)

Tocuh OSC Websocket for OBS and Syntax[](https://github.com/jshea2/OSC-for-OBS)

[GitHub - jshea2/OSC-for-OBS: Control and listen to OBS via OSCControl and listen to OBS via OSC. Contribute to jshea2/OSC-for-OBS development by creating an account on GitHub.GitHub](https://github.com/jshea2/OSC-for-OBS)

Further References

[1] OBS Studio / Open Broadcast Software - <https://github.com/obsproject/obs-studio>

[2] OBS Websocket / Open Broadcast Software - <https://github.com/obsproject/obs-websocket/tree/5.0.1>

[3] Touch OSC / Hexler - <https://hexler.net/touchosc>

[4] Control and listen to OBS via OSC / jshea2 - <https://github.com/jshea2/OSC-for-OBS>