

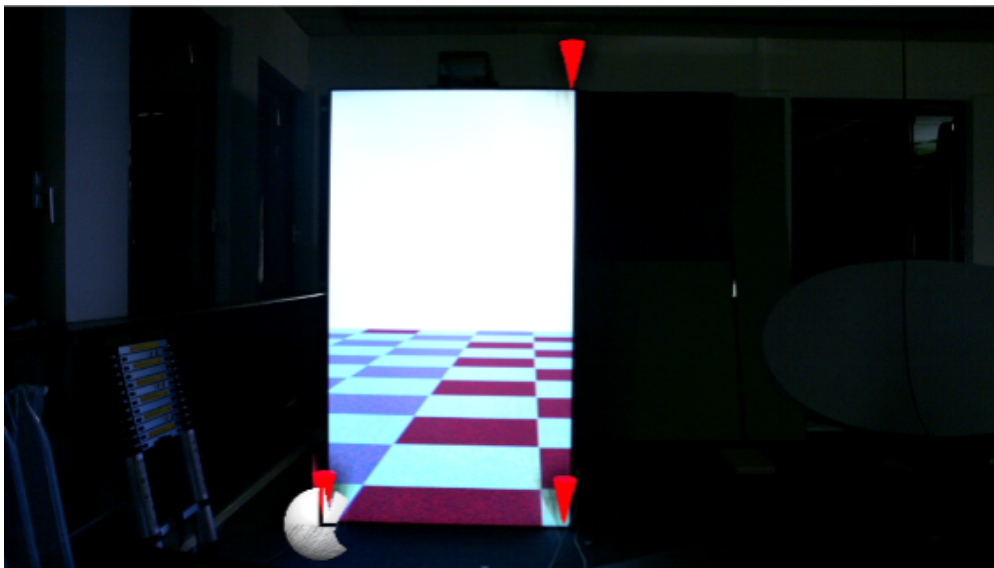
## VP Pro XR Manual - 5.3.2.2

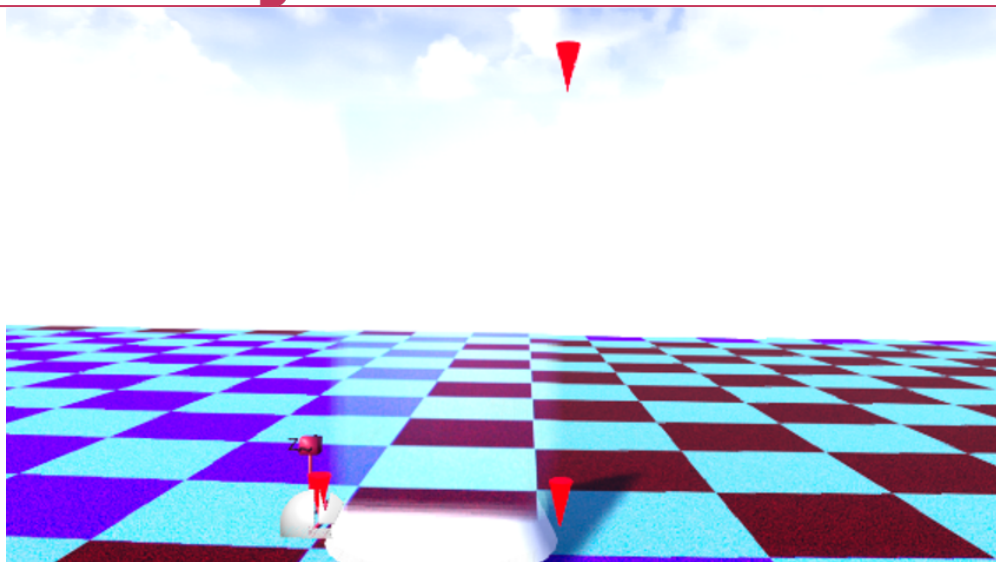
### 1. About the System

Mo-Sys VP Pro XR is a pre-configured, multi-node system, joining native Unreal Engine's nDisplay with Mo-Sys unique features like Cinematic XR focus and XR compositing tools.

The system comprises of:

- A number of rendering nodes driving the image on the LED background which create the depth/parallax effect, using Unreal's nDisplay plugin.
- An XR Engine which virtually extends the LED set from its boundaries. It blends the real image with a virtually rendered image. Once setup, the system automatically detects which area is the real LED set and which is to be replaced by a virtual camera output. Additionally, users can add AR objects to the composite and perform tasks such as recording tracking data or interfacing with selected cameras.
- A StarTracker for reliable and precise camera tracking, essential for XR to work.





## 1.1 Before Installation

In order to speed up the installation it is recommended to take the following steps:

- Measure the height of the LED volume (from the bottom to top row of pixels).
- Usually, the number of rendering nodes corresponds to the number of LED processors. The maximum resolution available from one node is 3840x2160 px. Divide the LED volume into equal-sized parts if possible. Every part is then driven by one LED processor.
- Prepare an FBX model of the volume in real scale. It should be split into meshes that correspond to the way the wall was split (from the previous point).
- Rendering nodes have Display Port outputs, so ensure the LED processor has the appropriate input or the conversion is available. If there is no DP input on LED processor, it is recommended to use a matrix switcher e.g. from DP to HDMI. It guarantees the video delays are consistent on all parts of the volume.



## 2. System Overview

### 2.1 System Structure

The system relies on Unreal nDisplay plugin to render the content on a LED wall. Thus, the setup is divided into 2 parts that must be completed in the following order:

1. LED wall setup with nDisplay.
2. XR set extensions setup.

### 2.2 Requirements

When shipped, the system is already pre-configured as follows:

XR Engine:

1. Listed in **VProManual**.
2. Remote Control Web Interface plugin requires installing Node.js from <https://nodejs.org/en/>.

Render nodes:

1. Unreal Engine 5.3.
2. Nvidia Quadro Sync Card II, if using hardware sync (included).

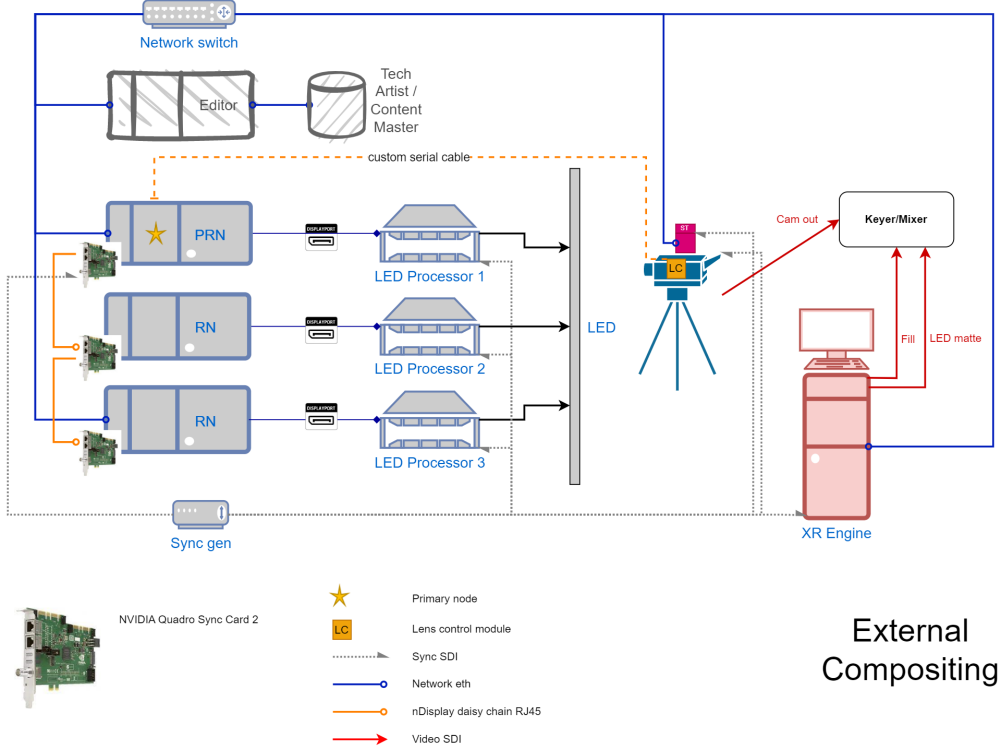
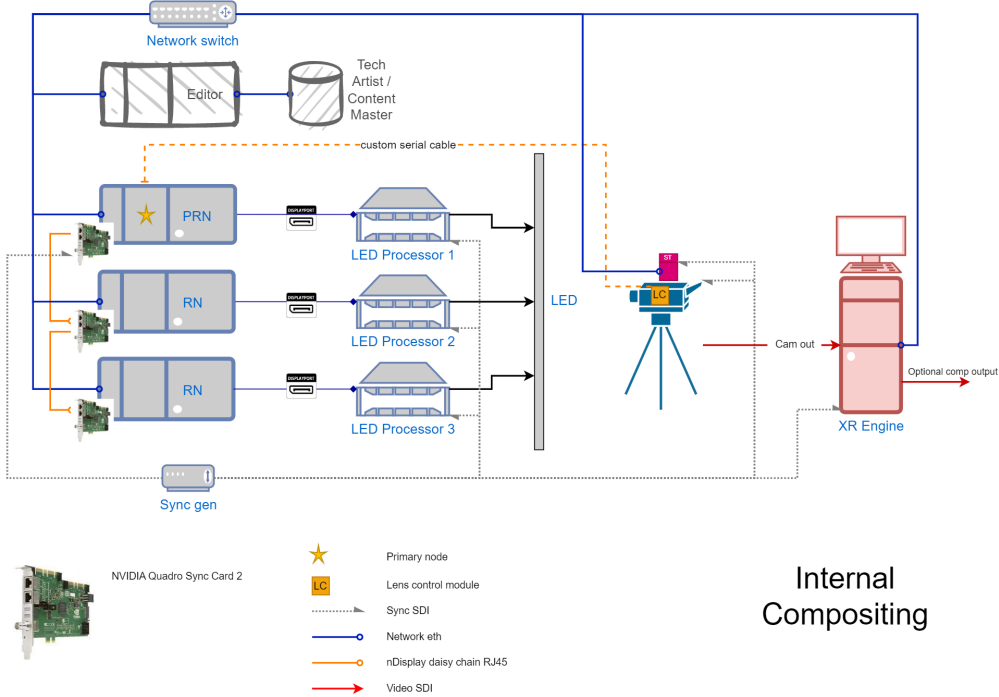
### 2.3 Connections Diagram

The diagram below represents a setup with three Render Nodes (RN) corresponding to the LED processors.

**Notes:**

1. Set the StarTracker to only send data to XR Engine and Primary Render Node (PRN). Tracking is passed automatically on other Render Nodes. The tracking won't be passed through Unreal's Multi-user if using in a configuration with a dedicated editor computer.
2. The network is solely used for tracking updates, triggering nDisplay and events. There is no image streaming over network.
3. Currently we support internal compositing and mixed compositing on external keyer/mixer, by providing a fill CG image and a matte for the LED wall.





### 3. Share and Map Drives

#### 3.1 Network Drives

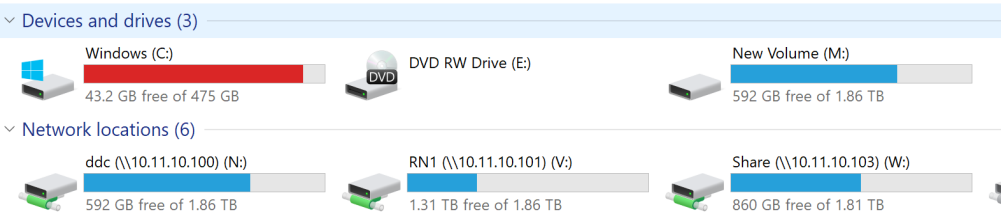
All the computers in the system are involved in presenting a believable and consistent graphics output, whether it's a Render Node rendering to the LED or XR engine compositing camera image with set extension.

That means the Unreal Project has to be maintained at the same state on all machines.

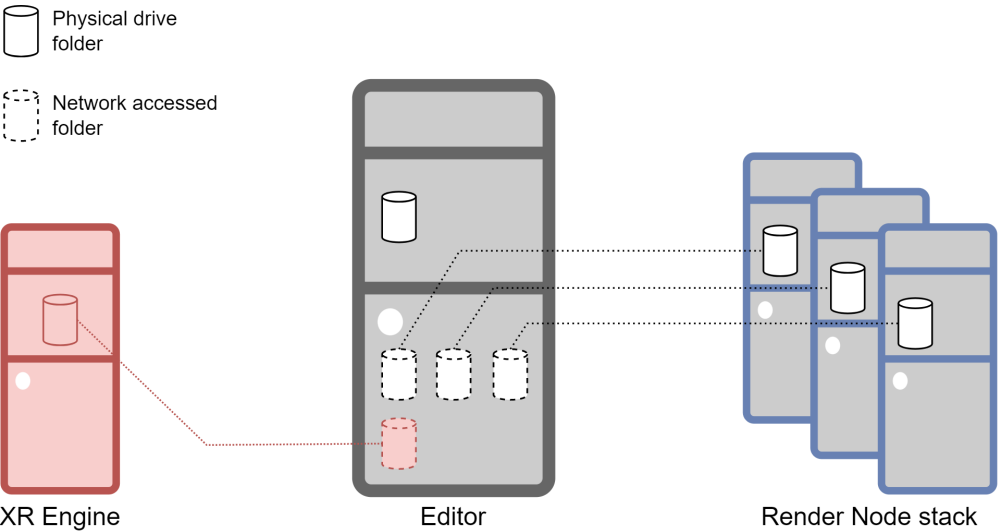
It is common to use an extra computer in In-Camera-VFX called Editor, which enables an operator or an artist to modify the environment. Additionally, those changes need to be applied to the XR engine.

The easiest way to do that is to map physical drives from Render Nodes and XR Engine on the Editor. Any modifications on the Editor can be then distributed to the other computers via network.

It is highly recommended to define a common directory on all computers. If possible, we recommend using a separate partition and changing the letter name to M. The usual path to the folder storing all available projects would then be: **M:\Share**.



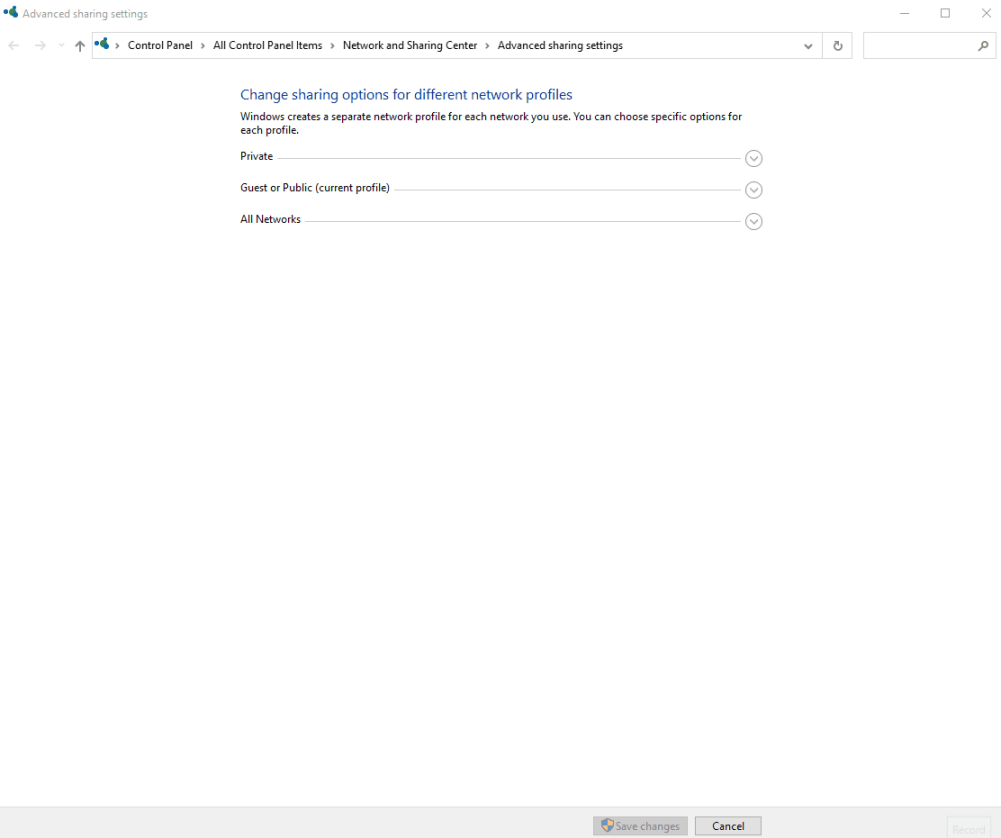
See to learn how to use the mapped drives in project updating.



### 3.2 Share a Folder on Local Network

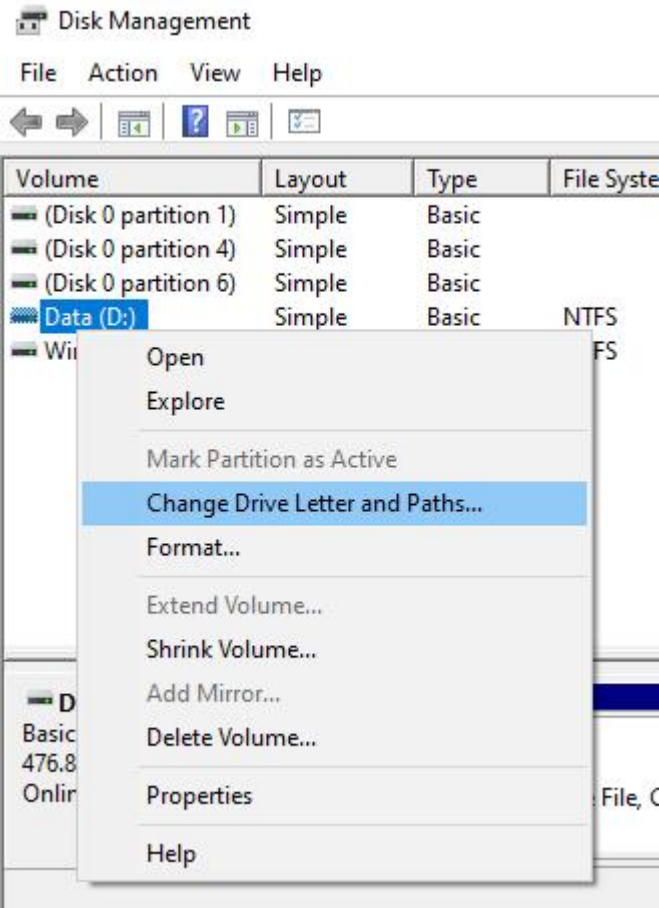
Advanced sharing settings:

- 1. Open Windows Advanced sharing settings.
- 2. Turn on network discovery on all the profiles.
- 3. Turn on sharing with anyone and turn off password protection.



Folder directories

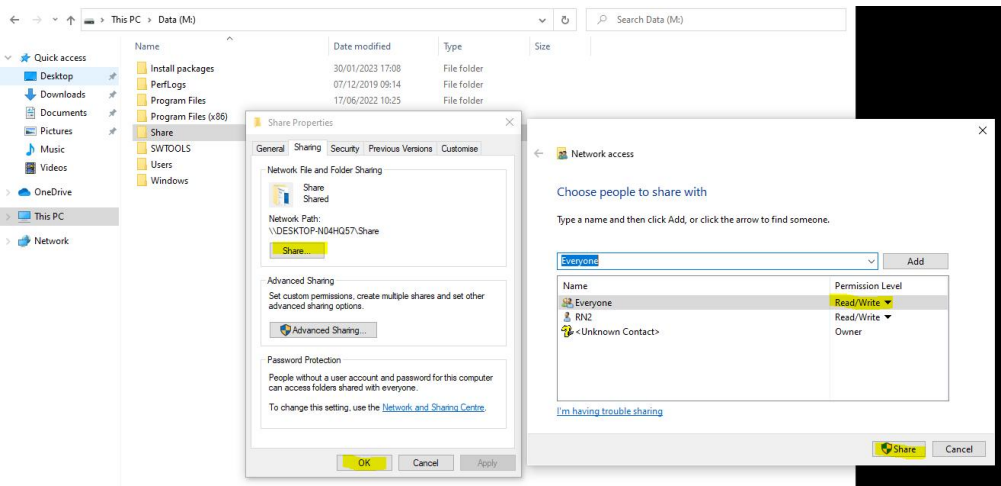
Project should be stored in the same local directory for all PCs. Make sure the directory is available on all computers. If possible, we recommend using a separate partition and changing the letter name to M. The usual path to shared folder is then: M:\Share. You can change the drive letter in Disk Mangement



Share folder

Right click on the selected folder and click Properties.

- 1. Go to Sharing tab.
- 2. Click Share....
- 3. Add Everyone from dropdown and assign Write/Read privileges under Permission Level.
- 4. Confirm with Share and OK.



## 3.3 Map Drives or Network Locations

### Methods:

There are 2 ways of connecting to other PCs' content:

1. Set up a network drive.
2. Set up a network location.

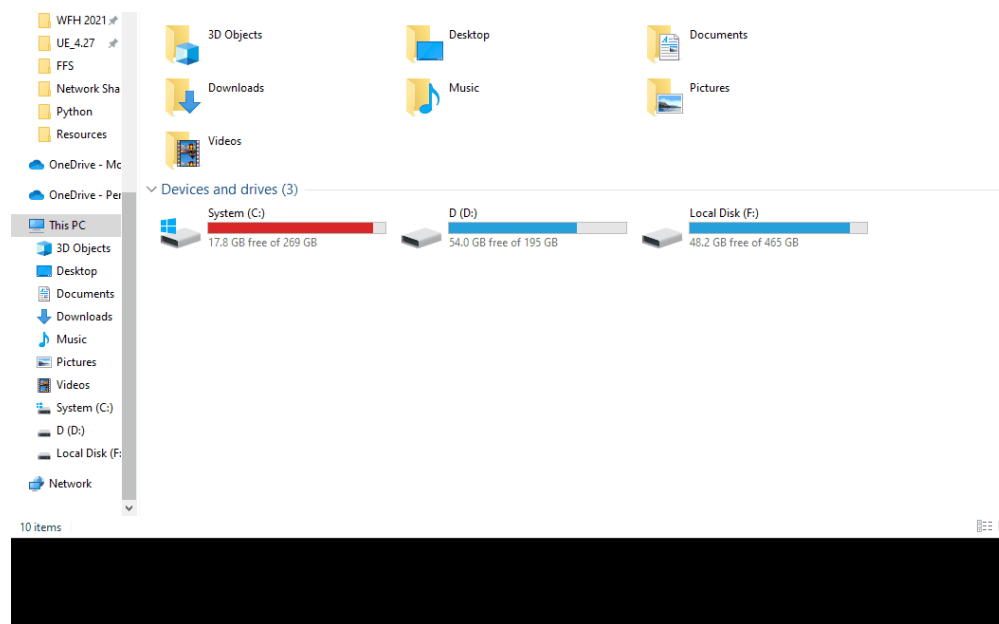
Both are valid for the purpose of maintaining the nDisplay and XR projects in sync.

### Using Mo-Sys File Sync tab in Switchboard

You can use the Mo-Sys File Sync tab to map all listed devices at once. Refer to Switchboard section in **VP Pro Manual** for the steps.

### Using Windows UI:

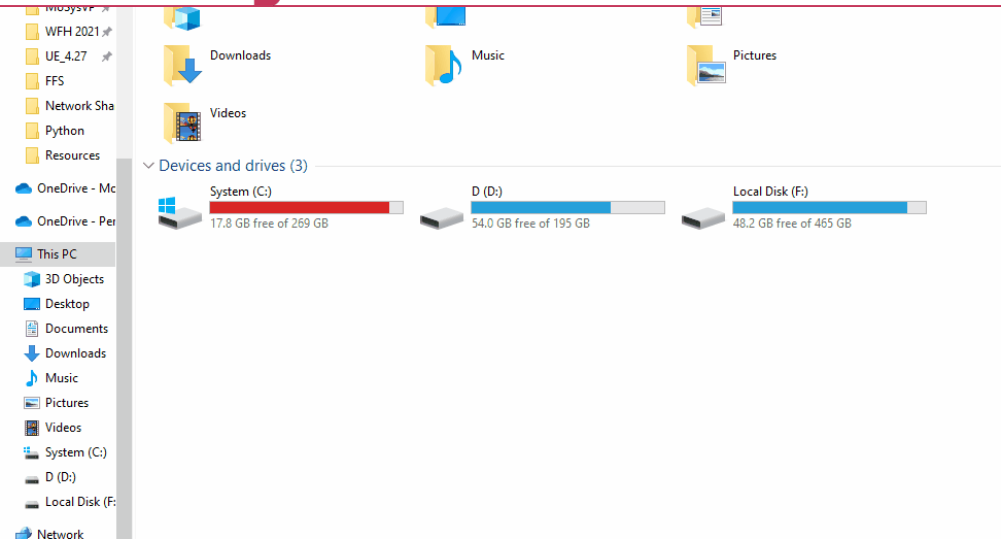
1. In This PC, right-click Network.
2. Type in, or paste, the shared folder directory, e.g. \\RN\Shared.
3. Type in Username and Password.
4. If having issues with access select Connect using different credentials.



### Network location

1. In This PC, right-click and select Add a network location.
2. Click Next and Next again.
3. Type in or paste the shared folder directory, e.g. \\RN\Shared.
4. Click Next and Finish.



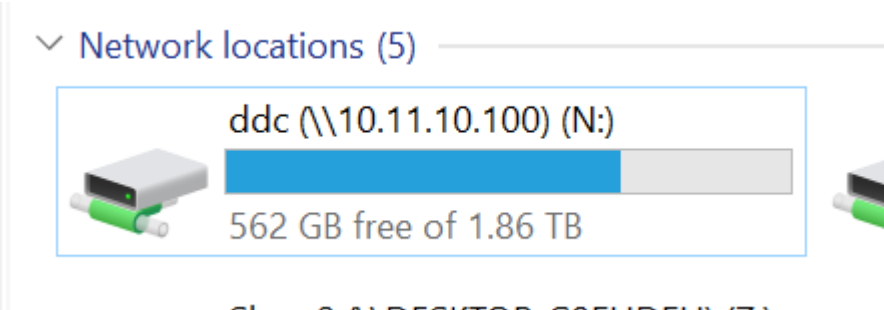


3.4 DDC on a Shared Drive

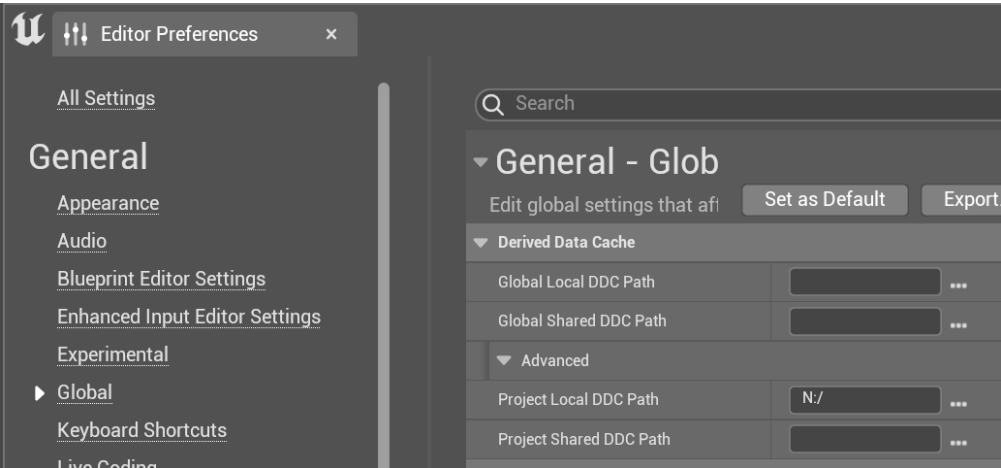
The mapped drive infrastructure can be used for a shared Derived Data Cache. These are the files that Unreal Engine generates when compiling shaders. It is possible to assign a location on one of the computers (Editor recommended) and share that location on the local network.

Example setup:

- 1. Share a folder on e.g. Editor PC and call it 'ddc'.
- 2. Map the shared folder on all computers.



3. Select the location for DDC in Editor Preferences - Project Local DDC Path. This will affect all the projects on that PC!



The PC that stores the DDC must be ON to take advantage of the setup.

### **3.5 Shared Saved folder**

If you wish to use a Shared project on a network drive or NAS, but have multiple machines with different licenses, please see the **VP Pro Manual** 4.1.1 section on Symlinking Saved Folder.

## 4. Common nDisplay Terminology

**nDisplay** is often used as a shorter name for the incamera LED VFX workflow. It is actually a plugin in Unreal Engine, that enables rendering a correct perspective on LED screens.

**Inner Frustum** - the area of the LED screen that the is visible through camera field of view. It is driven by tracking, so positional, rotational and lens change is reflected inside it.

**Outer Frustum** - area outside of inner frustum, it is usually static and is used as a source of realistic lighting.



**LED processor** - the hardware and software that combines multiple cabinets (sets of LED panels) into an array that displays a single image. There could be more LED processors driving an LED wall. The image rendered on render nodes is the input for LED processors.

**Primary Render Node (PRN)** - a computer that renders the image shown on LED volume, or part of it in bigger setups. Primary Render Node is the computer, which should receive tracking from StarTracker.

**Render Node (RN)** - a workstation that launches the Render Node(s) to put the image on the LED screen/volume and allows for editing.

### 4.1 Software nDisplay Related Terms

**Switchboard** - is a Python application for controlling multiple render nodes and relies on the companion application SwitchboardListener to communicate with them. Launch render on LED from here.

**Multi-User** - plugin in UE allowing for live level editing while content is displayed.

**Cluster** - set of render nodes.

## 5. StarTracker Installation in LED Volumes

### 5.1 Setting up the StarTracker System

For the full procedure of setting up a StarTracker system, please refer to the **StarTracker Manual**.

Additionally, Mo-Sys has a collection of helpful videos outlining the StarTracker installation process.

Those video can be found here: <https://vimeo.com/showcase/8529819> with access password: St@rTr4cker

This section of the manual discusses best practices of setting up the StarTracker system in an LED environment, as well as nDisplay-specific settings and setup options on the StarTracker system.

It is a crucial task to find a transform (position, rotation and scale) of the LED screen(s) in relation to the tracking coordinate system, as it is used for rendering the correct perspective on the screens. When working with set extension this is even more pronounced, as the transition between the graphics visible through a camera and the extended by the XR engine must be seamless.

The layout of screen(s) geometry will be used on both the XR Engine to mask the LEDs and the rendering nodes to define the nDisplay configuration.

### 5.2 Setup Considerations in the LED Environment

When setting up a StarTracker system in the LED environment, it is important to maintain a known geometric relationship between the StarTracker coordinate system and the UE5/nDisplay coordinate system. This manual will outline some common Autoalign patterns below - those patterns will help preserve a solid geometric relationship between real world coordinates, the StarTracker coordinate system, as well as the UE5 coordinate system.

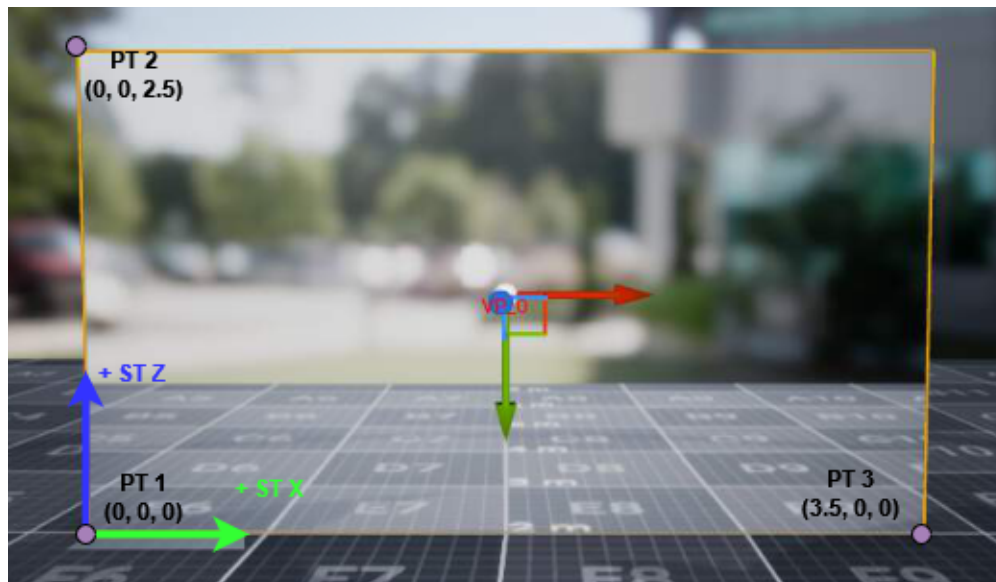
With regards to lens calibration, pure nDisplay / In-Camera VFX is one of the least challenging environments for tracking and lens calibration accuracy. Since the background plate would get distorted in camera, and field of view can be easily modified in the nDisplay config, the main lens parameter to get right is the focal distance.

### 5.3 Recommended AutoAlign Patterns

For all diagrams in this section, coordinates are represented as (X, Y, Z) in the StarTracker coordinate system.

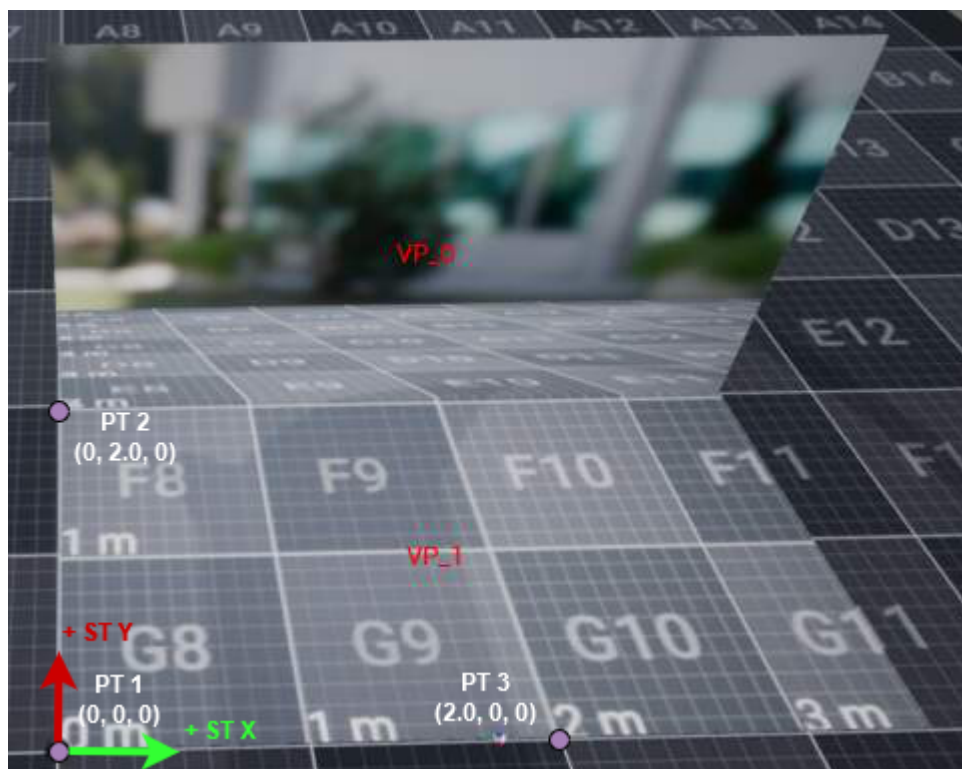
### Aligning to a flat plane of the LED wall

When working with a flat rectangular LED surface, one of the possible AutoAlign patterns includes aligning to the LED wall plane itself. In this arrangement, corners of the LED wall would serve as markers / AutoAlign points. In the example below it is top left, bottom left and bottom right corners. Set the height to correspond to studio floor.



### Aligning on the floor, relative to flat LED wall

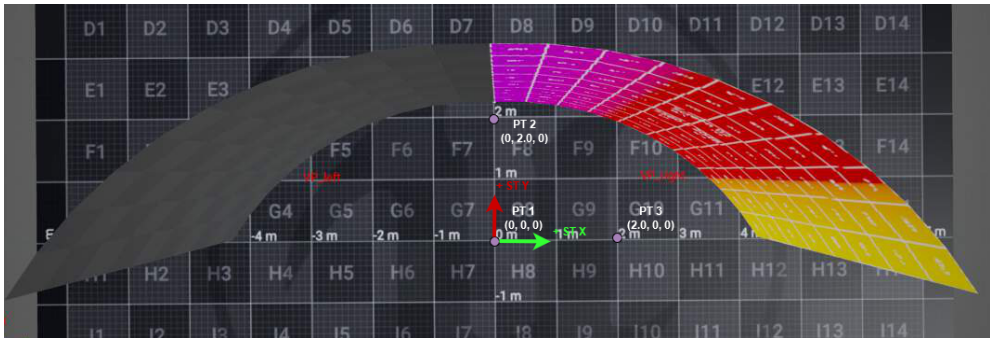
In some scenarios, such as when corners of the flat LED wall are obscured by the real stage floor, another possible AutoAlign pattern is to assign Auto-align points on the floor, but strictly relative (perpendicular and parallel) to the LED plane. This way there will be a known geometric translation offset between the StarTracker origin and the UE5 / nDisplay coordinate origin.





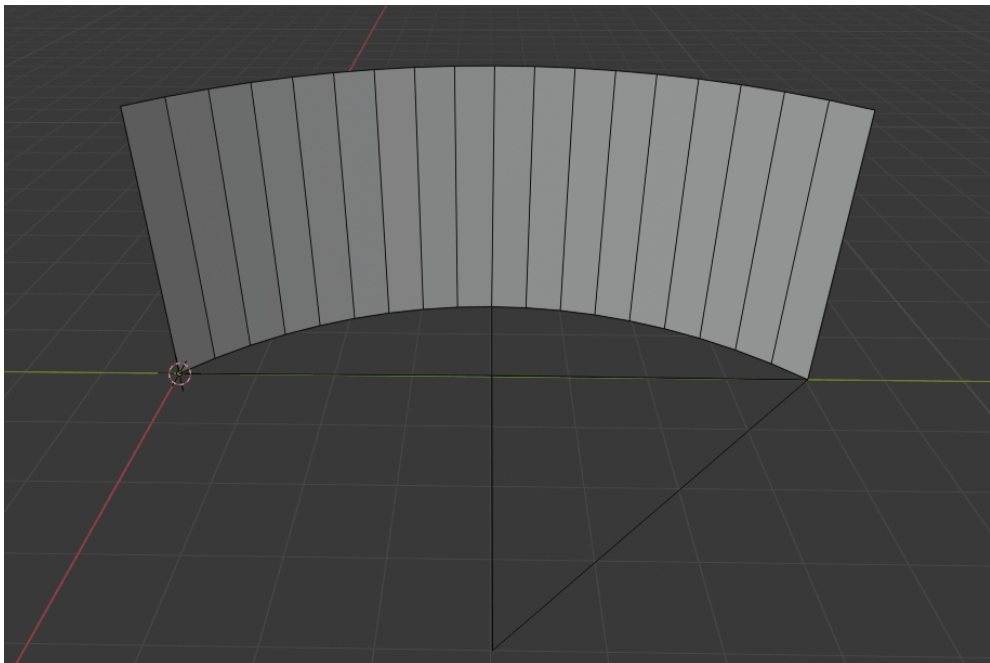
### Aligning to a curved LED volume

Start from drawing a line between bottom corners and mark a mid-point (0 point). Mark other 2 Auto-align points perpendicular to the drawn line.



### An example of a curved LED wall Auto-align

1. Mark a line between bottom corners and find the mid-point.
2. Use a 90 degrees square laser to find the perpendicular line starting from the mid point.
3. Make a right equilateral triangle for StarTracker auto-align with the side length of corner-midpoint.
4. The points in auto-align should be defined so the forward direction goes into the LED wall, like in the example below.



If there is a height offset (the LED is elevated from the floor) drop the points from the corners of the wall to the floor.

## 5.4 Recommended Tracking Data Output Configuration

Data should be sent to three computers: 1. The Editor PC (8001). 2. An extra PC. 3. To PRN (8001) > Live Link Preset 2.



## 5.5 LED latency minimisation

### Tuning > XYHPTR

Set to minimum possible, e.g. 0.

### Tuning > LED Wall

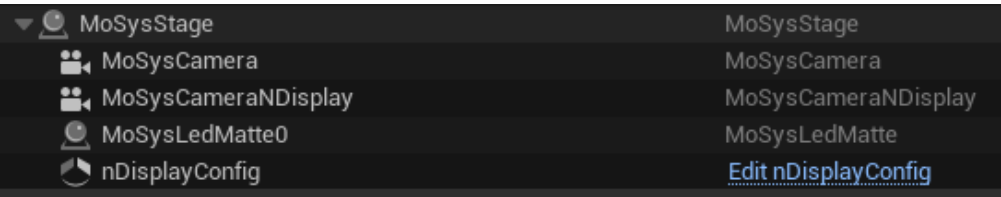
If set to negative values it will predict the camera pose [in fields], UDP output has to be set to LED On.

## 6. Unreal Engine - Starting Points

### 6.1 Level Structure

Both XR and Render nodes should adopt a specific level structure for easier synchronisation and maintenance of the project. It is recommended to use a separate, persistent level for MoSys actors and nDisplay configuration. This level can be attached to a master level or the other way round.

Use LMoSysXRExample.



Key points to take into account:

- 1. nDisplayConfig actor needs to be set to hiddenInGame.
- 2. nDisplay configuration mesh(es) should correspond 1:1 to the size of MoSysLEDMatte.
- 3. AR (foreground) objects must be tagged MoSysAR and set the custom stencil to 246. The objects get hidden if in nDisplay. Use XR Controller Panel to make actors AR with one click.
- 4. Multi-user on XR engine only updates the level when not in play. XR panels don't trigger multi-user changes.
- 5. MoSysCameraNDisplay acts as normal nDisplay-assigned CineCameraActor, but has the tracking already set-up.
- 6. Avoid copying DefaultGame.ini (game configuration in configuration folder) between XR engine and Render Nodes. Different settings must be set on both.

## 7. Mo-Sys nDisplay Integration - In-Camera VFX

### 7.1 nDisplay Project

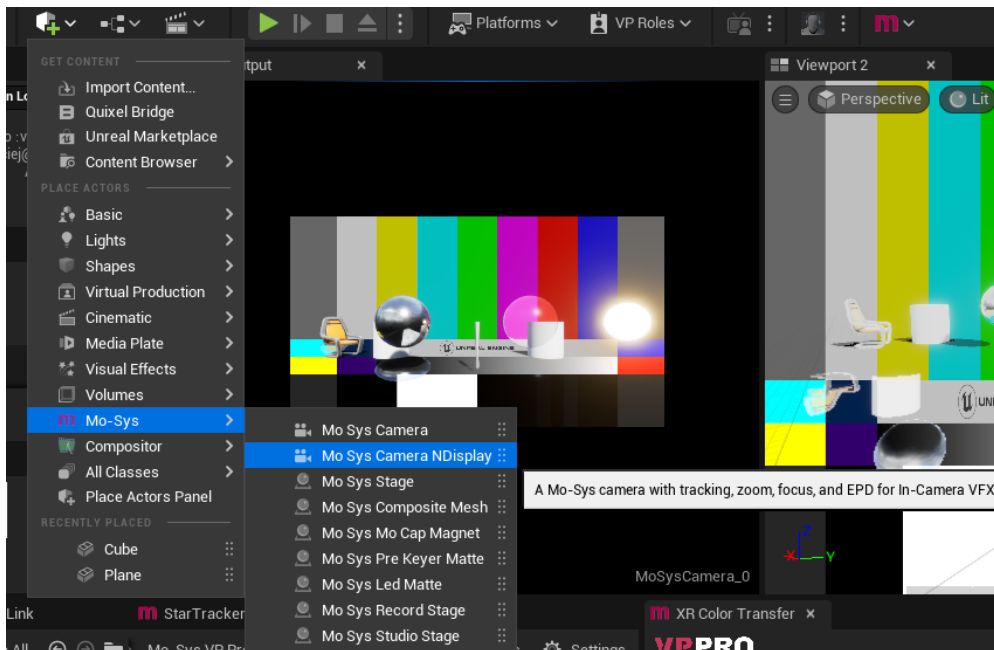
The following section only describes the extra setup on top of project prepared to render to the LED wall (nDisplay - ICVFX). For more information how to set it up from scratch view .

### 7.2 First Steps

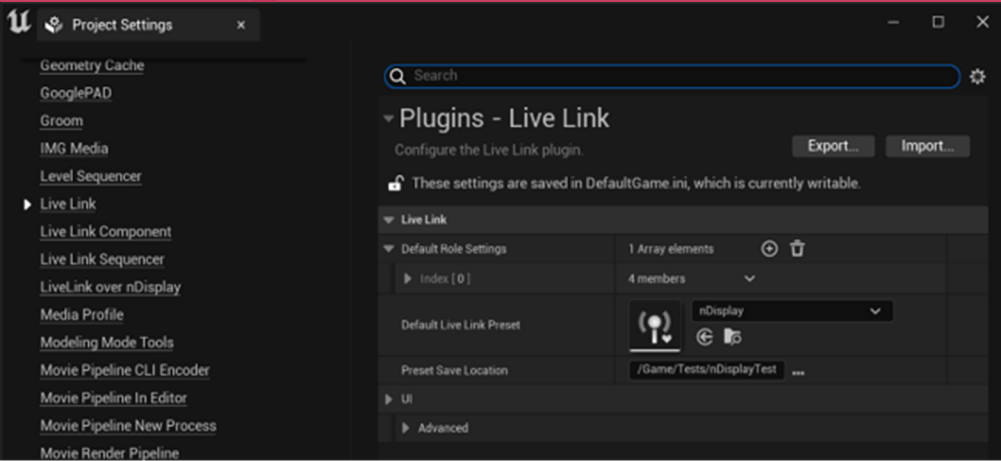
1. Install the plugin on XR Engine and the Primary Render Node (Chapter 3 in **VPPro Manual**).
2. Configure Live Link preset (Chapter 4.4 in **VPPro Manual**).
3. Set Artist Mode to true in Project Settings > Mo-Sys VP.

### 7.3 Unreal setup

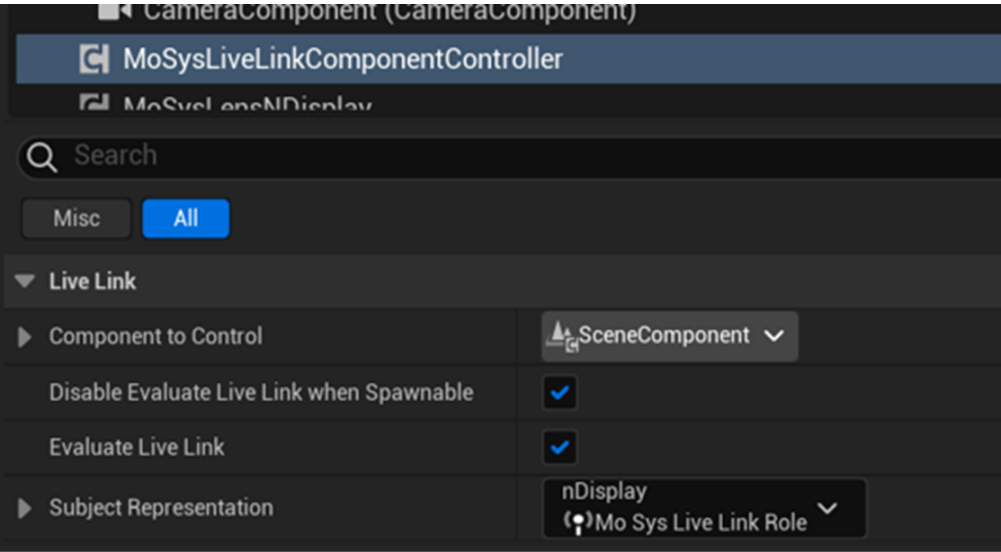
Spawn MoSysCameraNDisplay.



This is a replacement for CineCameraActor which is used in Unreal documentation. It is adding two components: MoSysLiveLink and MoSysLensNDisplay. Create and specify the Live Link preset in Project Settings > Live Link > DefaultLiveLinkPreset.

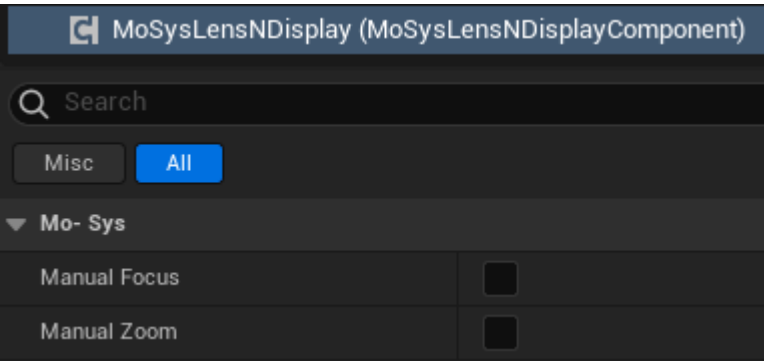


Select the Live Link subject on MoSysLiveLink component.



MoSysLensNDisplay component brings the following lens parameters:

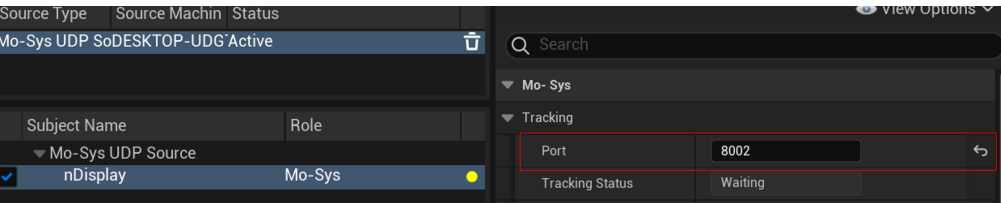
- Field of View.
- Focus Distance.



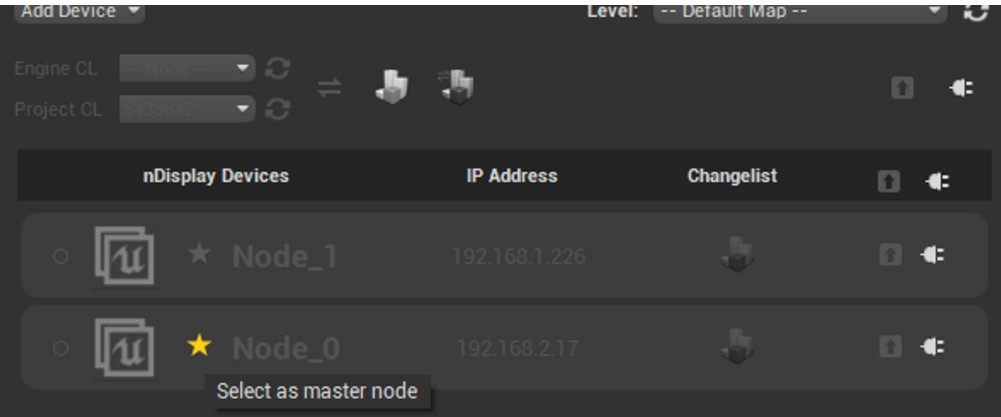
If the lens file is not available on the StarTracker or the data is not needed for a shoot, you can set the specific parameter to manual. This will give control to CineCameraComponent settings.



Finally, if editing and running nDisplay on a single PC, change the port (image below) on MoSys Live Link to free up the port for nDisplay Launched application (don't save the preset after the change).



For nDisplay cluster to work it is essential to send tracking to nDisplay master computer!



You can optionally designate another StarTracker IP endpoint to send data to editor's port, to see the camera movement there.

## 8. Render Nodes Hardware Setup

### 8.1 Network

All the rendering nodes as well as the XR Engine must be connected in a fast local network, preferably dedicated to them and the StarTracker.

All the devices should have a static IP address for easy maintenance. Prepare a list of the addresses, as it will be necessary for software setup. Install Tight VNC for remote access and set the project up in Perforce.

### 8.2 Connection

Daisy Chain the system by having an external sync source come into the master Sync Card via a BNC cable (Genlock) and spread the signal further down the chain using RJ45 cables. Ideally each render node should have only one display connected to it (the LED processor). If more than one are necessary configure them via Nvidia Mosaic.

### 8.3 Synchronisation

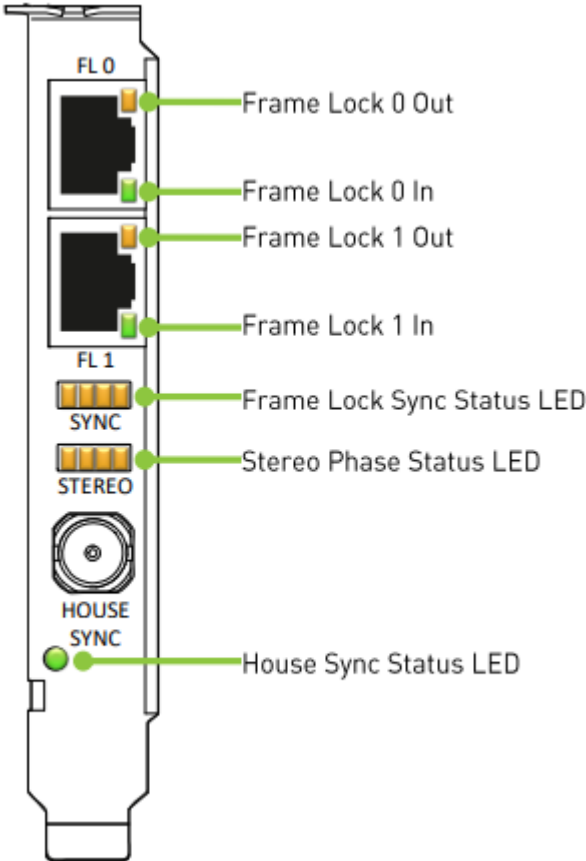
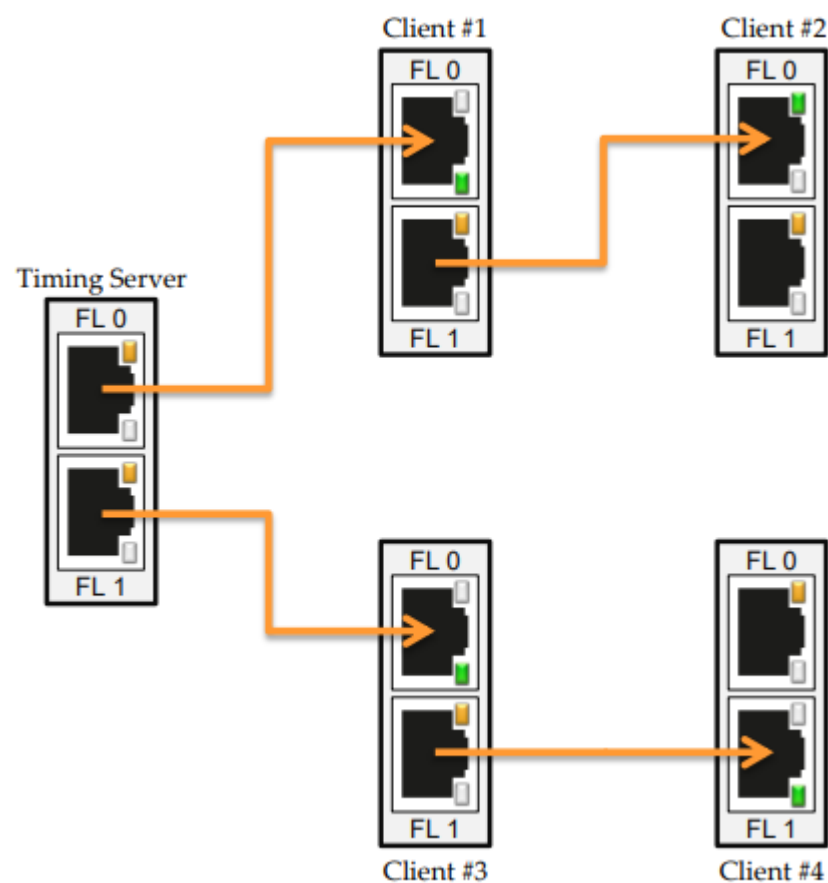
Synchronisation of the LED screen output is a key to a seamless shoot. It is crucial that all the sections of the volume are rendering at the same time, so there is no tearing between sections.

This is achieved by using an Nvidia Quadro Sync Card II. Every Render node has one of them connected to the GPU to guarantee synchronous rendering.

Refer to Frame Lock Configuration in [Nvidia's User Guide](#) for more information.

### 8.4 Quick Setup

1. Connect on-set genlock to the House Sync to the Primary Render Node.
2. Daisy chain the rest of the rendering nodes with CAT-5 Ethernet cables.
3. Enable VSync in NVIDIA control panel.
4. Configure the synchronisation in Synchronise Displays. Define whether it is a timing server (only the Primary Render Node).



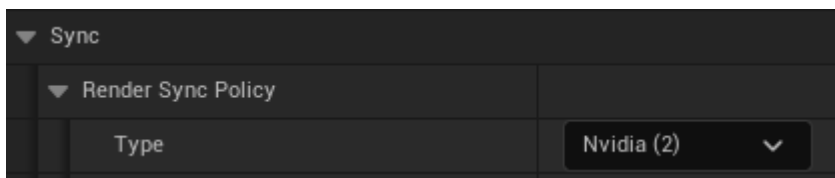
## 9. NVIDIA Software Synchronisation

### 9.1 Nvidia Driver Utility

Set a specific configuration on the Nvidia drivers via an Nvidia driver configuration utility, such as ConfigureDriver.exe - this can be obtained [here](#). Download and run ConfigureDriver.exe as administrator through Windows command prompt. When done, type 11 and press enter. This will enable the prePresentWait setting and improve performance without compromising sync.

### 9.2 nDisplay Configuration

In nDisplay configuration actor select Nvidia (2) in Cluster > Render Sync Policy > Type.



The Node > Window size should cover the entire desktop resolution. However, the viewport should only cover the resolution required by the LEDs.

In order for nDisplay to lock to the synchronisation from Nvidia sync it must run as a fullscreen foreground Windows application. No other application can sit on top of the nDisplay instance while it is running.

To manage this ensure that:

1. Nvidia control panel is not open.
2. No virtual desktop is running (such as teamviewer, zoom, etc).
3. Desktop based notifications and pop ups are disabled.
4. Windows desktop resolution scaling is set to 100%.
5. Fullscreen optimization is disabled on Unreal executable (using Fix ExeFlags from Switchboard).

You can verify that it is running in the correct mode by enabling option 8 in configureDriver.exe (Enable the SwapGroupPresentIndicator for Direct x).

### 9.3 Setting up EDID

Once you have everything setup in the Nvidia control panel with regards to sync, resolutions and colour space etc, it is useful to export the current EDID and then load it from a file. You can find the instructions [here](#). Alternatively, you can set your EDID through your switcher, such as a Lightware matrix.

It's important to note that incorrectly configured EDIDs can half the performance of nDisplay when using sync policy 2.

To avoid this, ensure you have an EDID which allows you to select a PC resolution at the frequency you wish to shoot at (i.e. 24hz) and is marked as native. Another solution to this is to create custom resolution based on the standard 3840 x 2160 60hz PC resolution and then set it to the appropriate frequency (i.e. 24hz, 25hz, 30hz etc).

9.4 Sync Validation

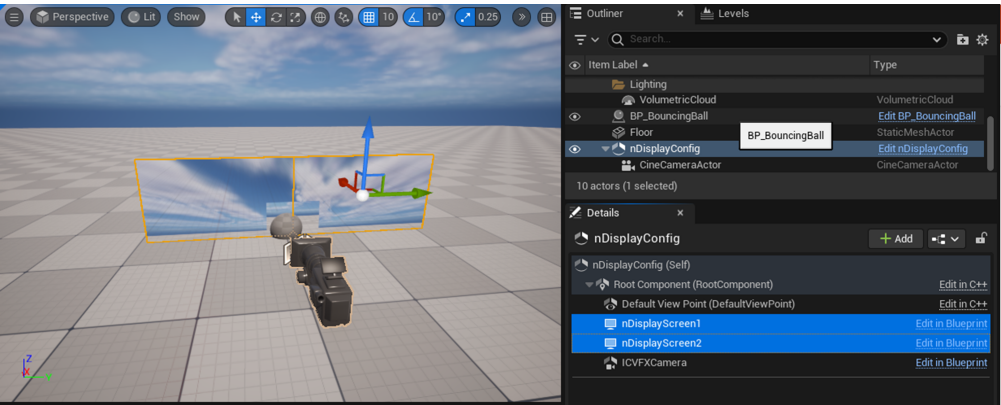
Use a bouncing ball test to validate the synchronisation. Place the ball on the edge between 2 parts of the LED wall, so it's visible on two separate segments of the wall. The ball will bounce up and down. If the ball is consistent it means the nDisplay setup is synced. If you see tearing of the ball then the synchronisation is failing.

The ball blueprint can be found under MoSysVPPro Content > nDisplay > Blueprints: BPBouncingBall.uasset\_.

Additionally, the Switchboard shows Nvidia driver and sync status of the nDisplay nodes. For a successful sync, the PresentMode column should indicate that each node is in Hardware Composed: Independent Flip. If it states Composed: Flip then you will want to check nothing is overlapping nDisplay as a full screen application on the nodes.

9.5 Bouncing Ball Test

Prepare the nDisplay level so the ball is positioned in between the nDisplay screens. After launching in Switchboard, the ball will bounce quickly up and down.





## 10. Render Nodes Setup - nDisplay

### 10.1 Live Link Setup

Start with the Primary Render Node and set up the Live Link connection with StarTracker.

### 10.2 nDisplay Level - Example

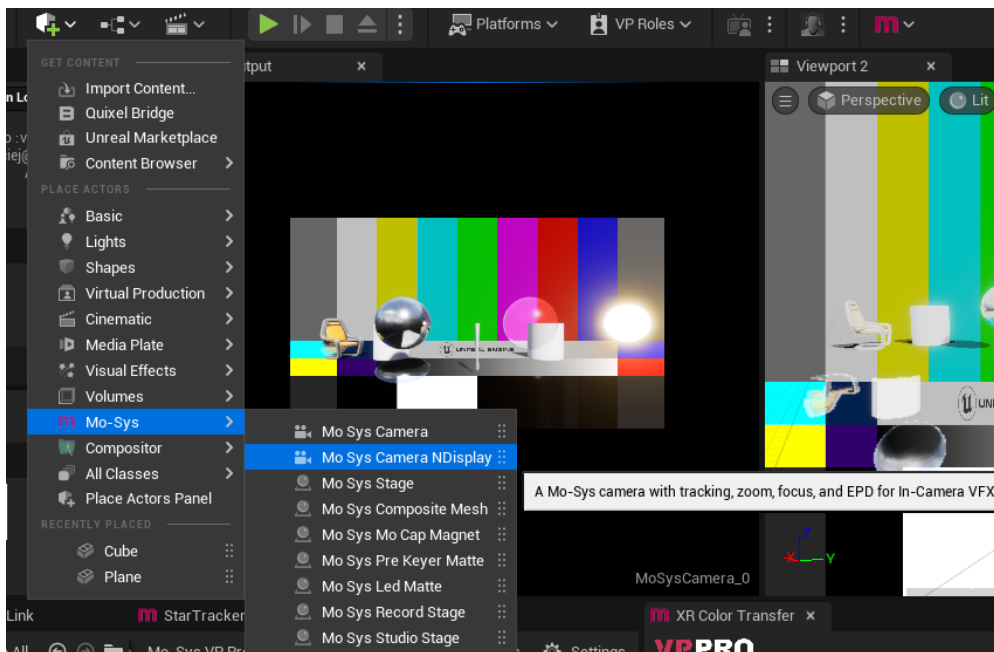
Open the nDisplayLevel level in MoSysVPPro's Content > nDisplay > Maps.

We will be following the [In-Camera VFX Template](#).

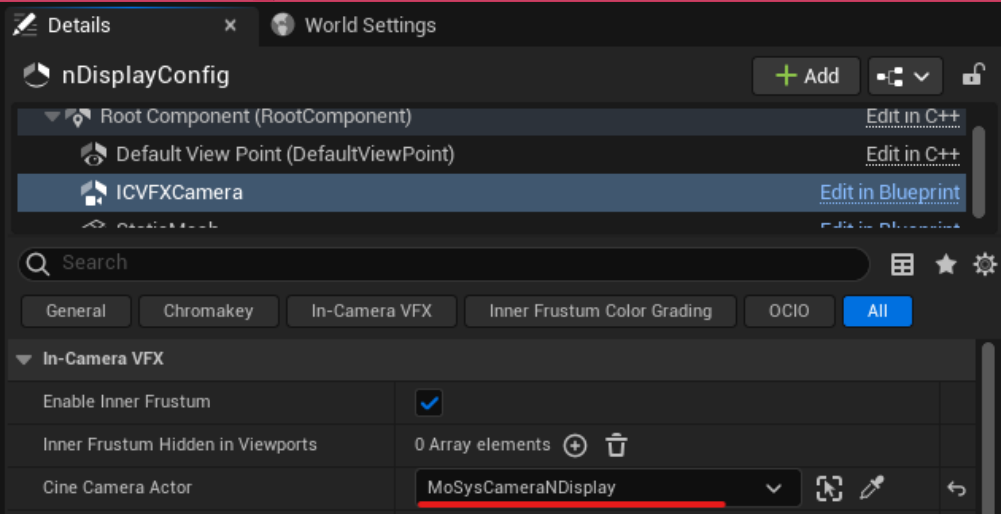
### 10.3 Setup

The setup requires two objects to render a correct perspective:

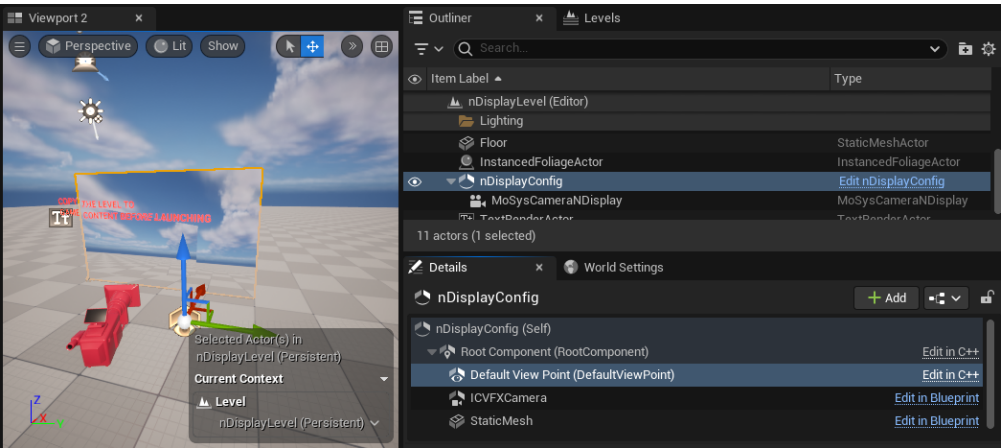
1. nDisplay Configuration - which defines the physical layout of the LED screen as well as the distribution of pixels rendered on the Render Nodes.
2. MoSys Camera NDisplay - this is used for rendering the inner frustum, which is tracked and visible in camera FOV.



The camera has to be specified in the ICFVFXCamera component in nDisplayConfig.

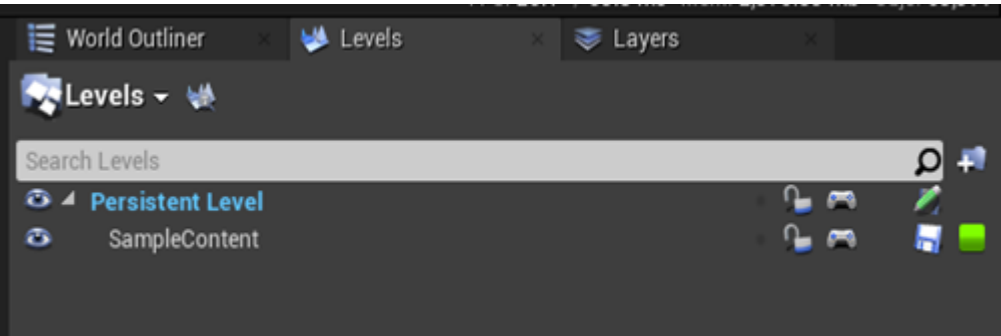


Default View Point is the origin of projection for all other areas of the LED that is not visible in-camera. It's not tracked and its purpose is to give the correct reflections on talent (actor), so it should be placed in front of the LED at a point where the actors will be, approximately.



10.4 Node Level Structure

It is recommended that the project is divided into sublevels, where at least one is for the content and one for the camera with a nDisplayConfig. That way, it is easy to transfer the setup to different content levels and projects. Remember: sublevels have to be always loaded, otherwise they will not be visible after launching.



## **10.5 Defining LED Screens in nDisplayConfig**

Depending on the number and shape of the LED volumes the virtual layout needs to be adjusted.

Top part of the nDisplay configuration actor defines the LED screen transform (where it is in space) and the bottom part describes the pixel mapping on the screen space. In other words, the image should be rendered in relation to the top left corner of the screen.

Edit nDisplayConfig to define the viewports.

## **10.6 Meshes - Geometry**

At the top of the nDisplayConfig, bring the meshes representing the screen/volume to 1:1 scale. For rectangular screens it is easy to use nDisplayScreen or a plane mesh, however, if dealing with more complex shapes like a curve it needs to be modelled externally.

Refer to [Step 2](#) - Create LED Panel Geometry in In-Camera VFX Quick Start guide for building a complex mesh.

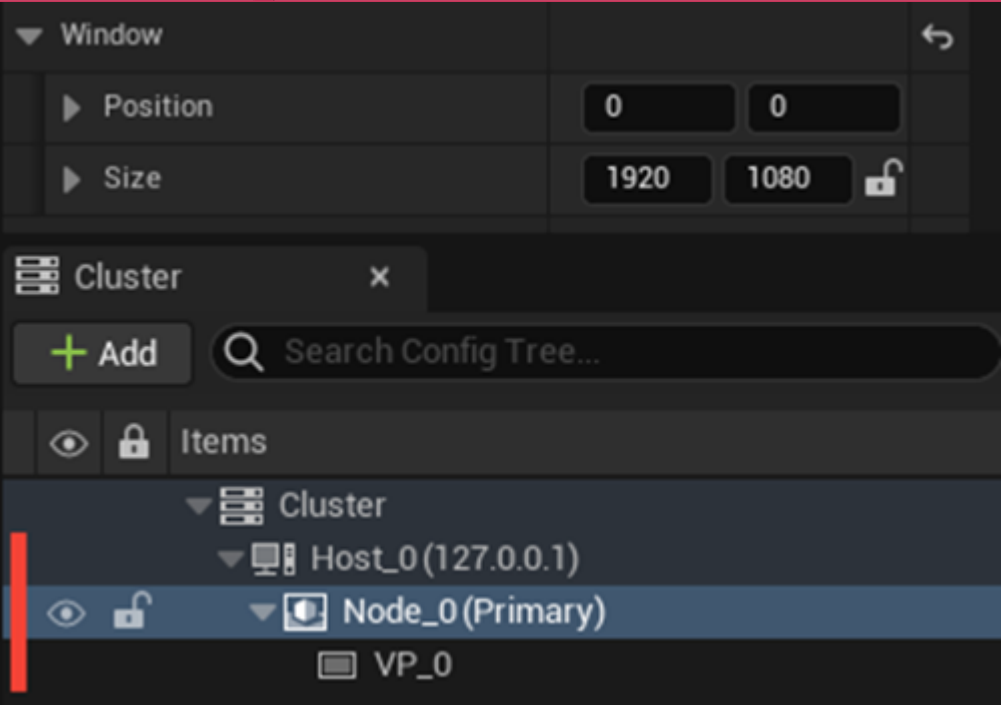
Every viewport responsible for rendering to a part of the LED volume is represented as a separate mesh in Unreal. The 3D model (mesh) should be split into the respective parts.

## **10.7 Output Mapping - Pixels**

Configure the resolution of the screen(s) at the bottom of nDisplayConfig.

Change the resolution of the viewport(s) to reflect the resolution of the screen or a segment of the volume driven by a Render Node. Usually one Node is responsible for one viewport.

Node (window) has to be big enough to fit viewport(s):



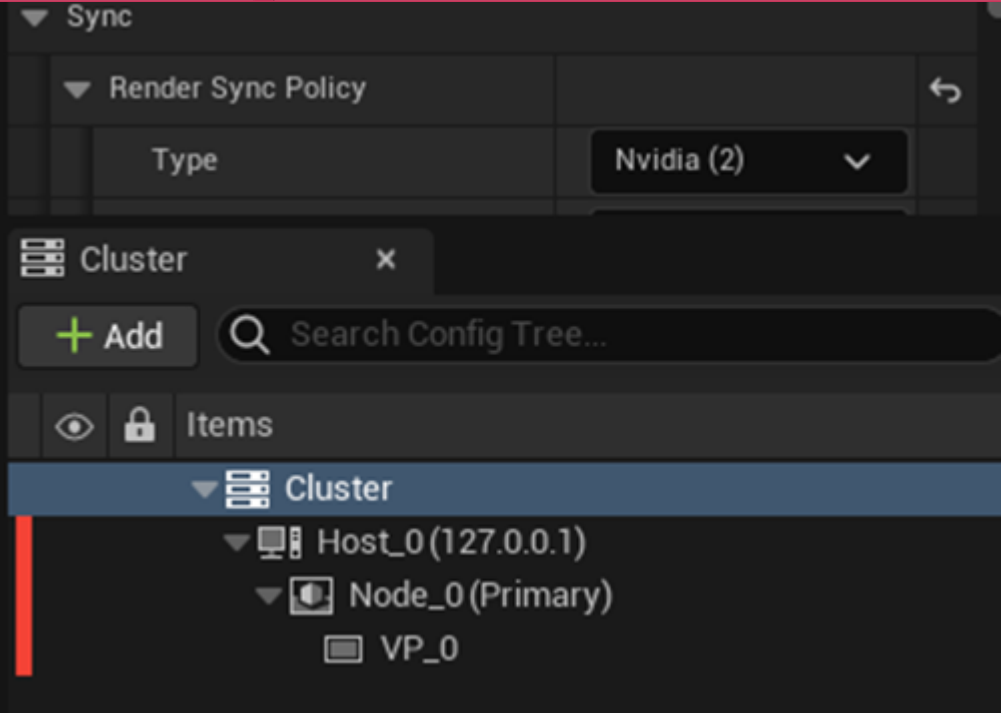
Refer to the next page images and, optionally, to [Step 3](#).

### 10.8 Render Nodes

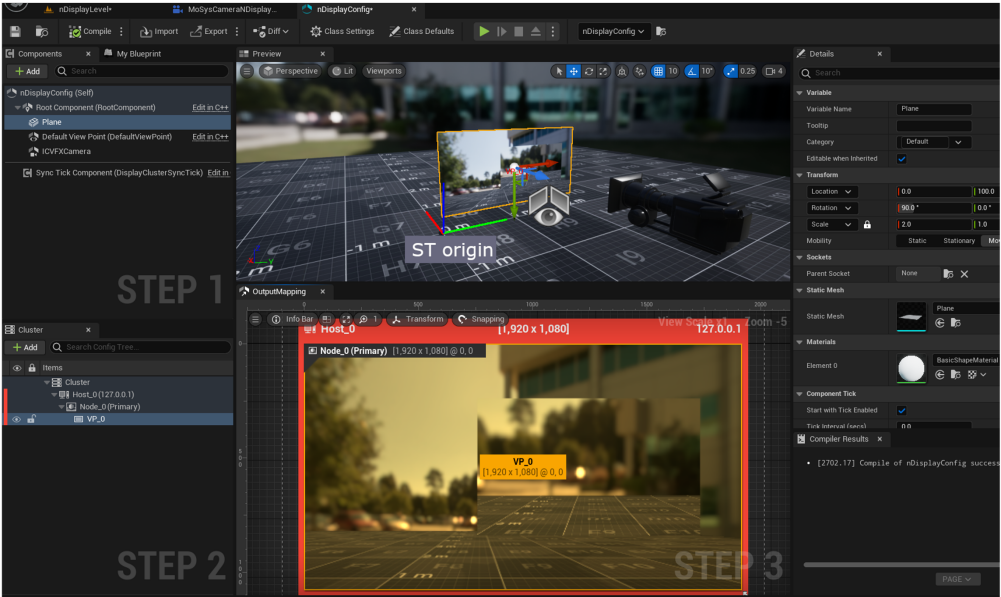
Every Render Node requires adding an extra cluster node and assigning the IP address to it on the Node component.

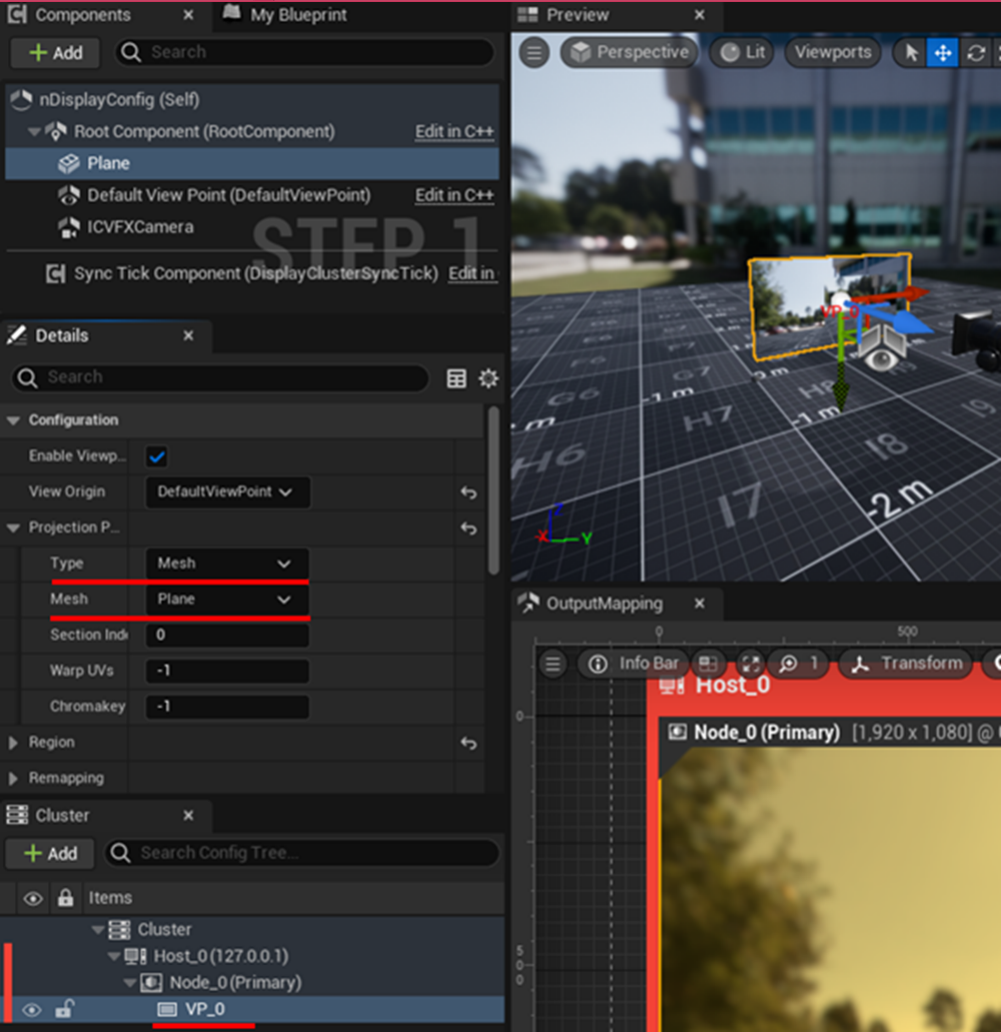
### 10.9 Sync Policy

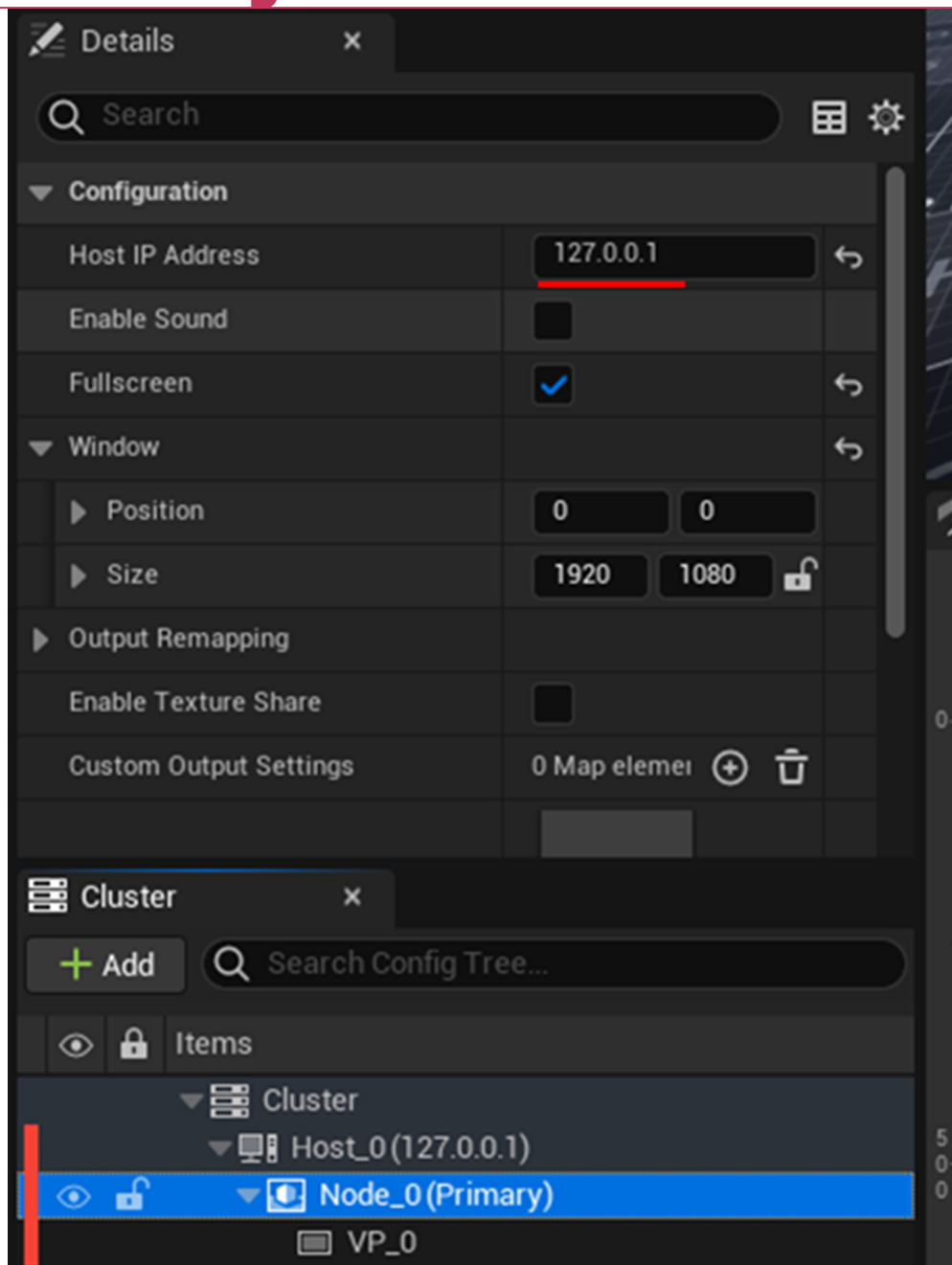
Select the sync policy. Nvidia (2) if using Quadro Sync Card II.



10.10 Example nDisplayConfig







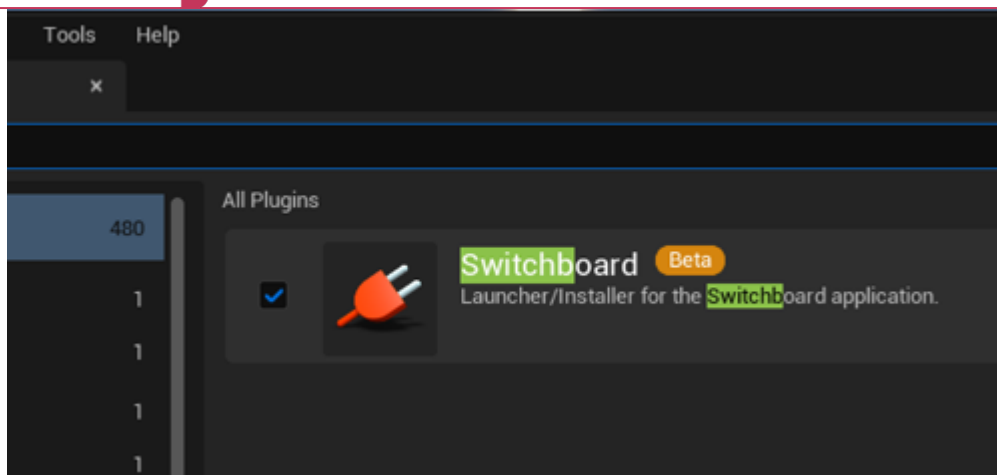
## 10.11 Launching

See [Step 4](#) for launching Your Project with nDisplay in In-Camera VFX Quick Start to launch the project on the LED screens.

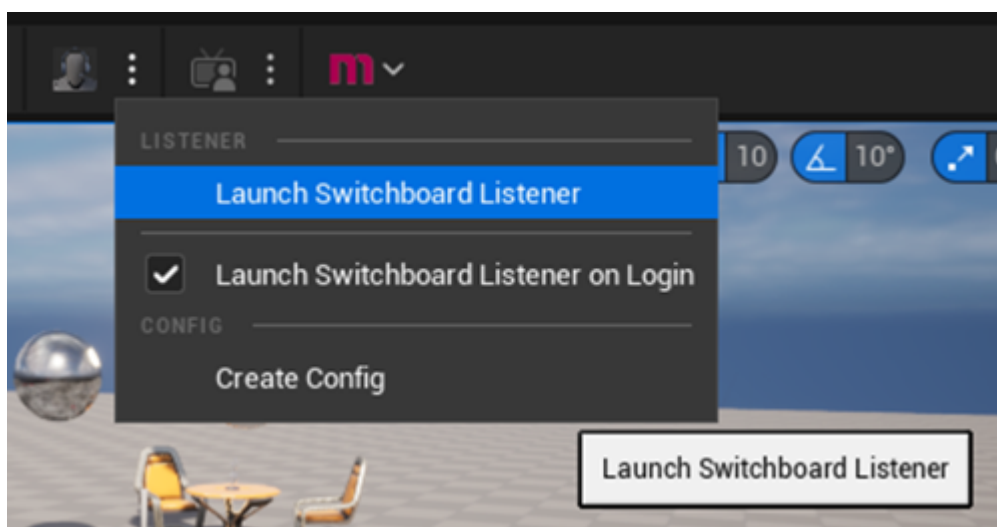
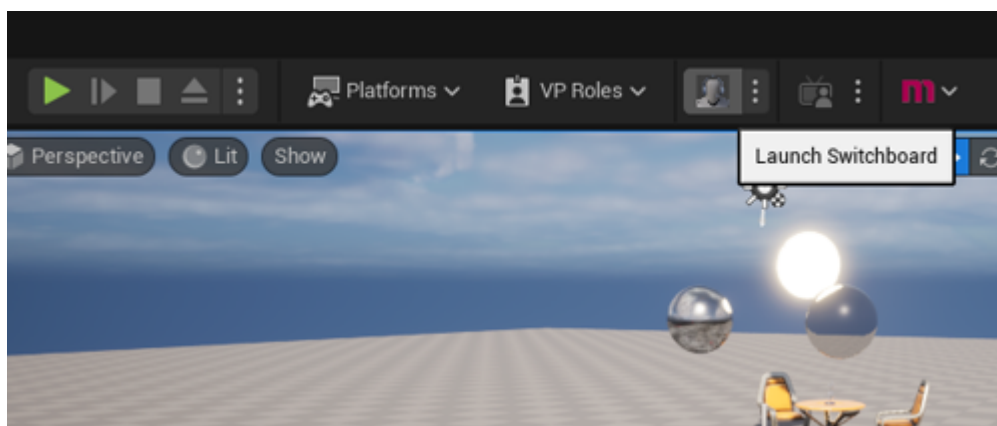
Use Switchboard plugin to launch:

1. Enable it in Edit > Plugins.

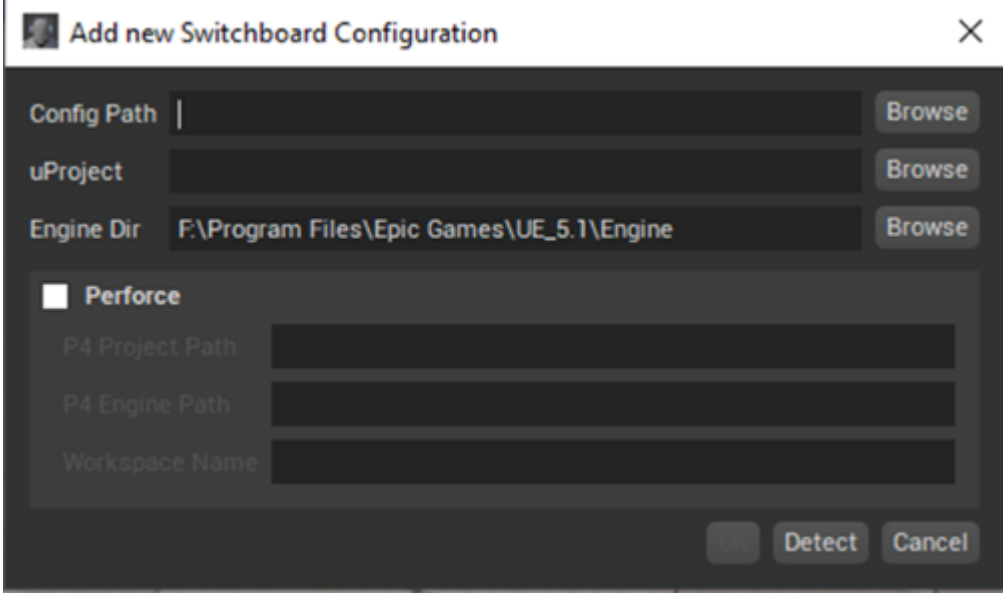




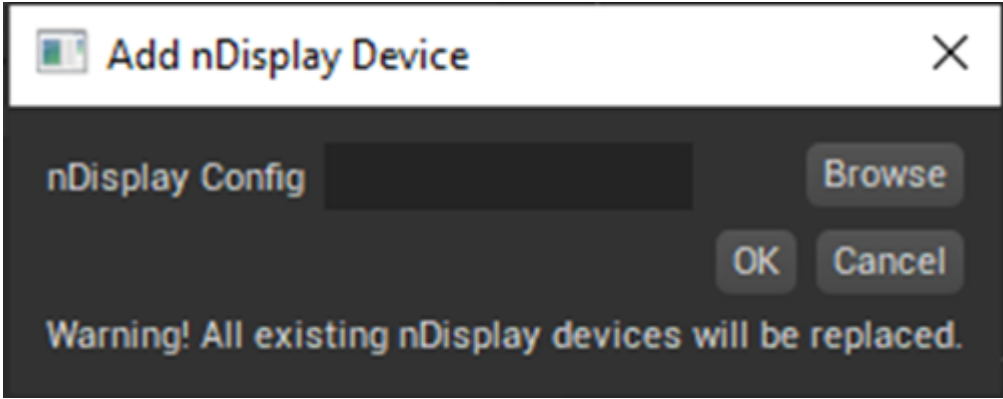
2. Install SwitchboardListener on all the rendering nodes (if not launched find it in ..\UE5.3\Engine\Binaries\Win64). Then start \_Switchboard on Editor or Primary Node. Launch it from UE or ..\UE5.3\Engine\Plugins\VirtualProduction\Switchboard\Source\Switchboard\_.



3. Configure the setup in Configuration > New configuration, by selecting the project and Engine folder.

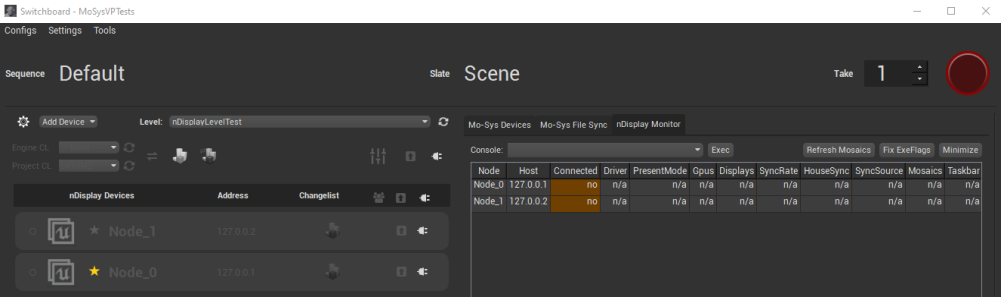


4. Add nDisplay Device from Add device dropdown and specify the config file.



5. Connect devices.
6. Start all connected devices.

After you launch in nDisplay launcher it should appear on the primary screen.



### 10.12 Project distribution

The Project must be copied to all of the Rendering Nodes to the same directory as the Primary Render Node. All the rendering nodes should store the same version of the project.

Various workflows can be used here:

1. It is recommended to use Mo-Sys File Sync panel in **Switchboard**. Refer to **VP Pro Manual**, Switchboard section (14.1) on how to install Mo-Sys Switchboard panels.
2. It is also possible to use a version control software like **Perforce**. The advantage being that the user can roll back to previous revisions.
3. Alternatively, especially for testing, a user can use a project that is located in the Windows shared folder, visible to all rendering nodes. That way, there is no need to distribute the project as it is loaded on start from the shared folder. This is not recommended for production, as the stability cannot be guaranteed and the start time is usually longer. This approach requires the user to assign a path to the network location in the Switchboard settings.

## 11. Cinematic Focus XR

### 11.1 System

Cinematic Focus XR provides a seamless focus control with LED stages. The system takes advantage of the existing geometry defined for nDisplay and camera pose given by the StarTracker. Allowing focus pullers to use it naturally regardless of whether an object is real or virtual. It is currently compatible with Preston and Teradek RT FIZ systems.

### 11.2 Prerequisites

Depending on the FIZ system the required hardware is either 1. or 2.

1. Preston FIZ system with:

- Preston MDR3.
- Hand unit 3 or 4.
- At least one motor for focus (all models are compatible).



MDR3 needs a firmware update to version 1.130 or higher.

1. Teradek RT with:

- RT MDR.X.
- RT CTRL.3.
- At least one motor for focus - Motor.X or MK3.1.



Custom firmware for both MDR and CTRL units is required and provided (TeradekRTfirmwarev1.2.15MoSys).

- Teradek lensmap copied from hand controller to the MDR (every time lensmap is changed).
- Use IN-1 port on MDR for connection.
- On hand controller - Menu > RX config > IN1 mode to wired and IN1 axis to focus.
- Current version of firmware doesn't allow for motor axis direction flip!
- Mo-Sys additionally provides cabling that connects to the MDR (cables vary between Teradek and Preston).
- Install the FTDI drivers, so a USB Serial Port appears in the Device Manager: <https://ftdichip.com/drivers/>.

## 11.3 Connection

Connect the serial cable to the Preston MDR 4 pin Lemo serial or 5 pin Lemo to Teradek MDR serial 1. On the other end plug it to the Primary Render Node USB port.

Configure the serial port in Device Manager and take note of the COM port number.

## 11.4 FIZ Setup

Setting up the FIZ system is out of the scope of this manual. Please refer to Preston's or Teradek's documents for more information. It can be found here: <https://prestoncinema.com/#!/downloads/firmware> <https://teradek.com/pages/downloads#teradekrt>

After powering up the system, it will try to find end stops of the lens. It is a part of the calibration.

Make sure to map focus to the lens reading distances. This can be done in both imperial and metric units.

## 11.5 VP Pro XR Software Setup

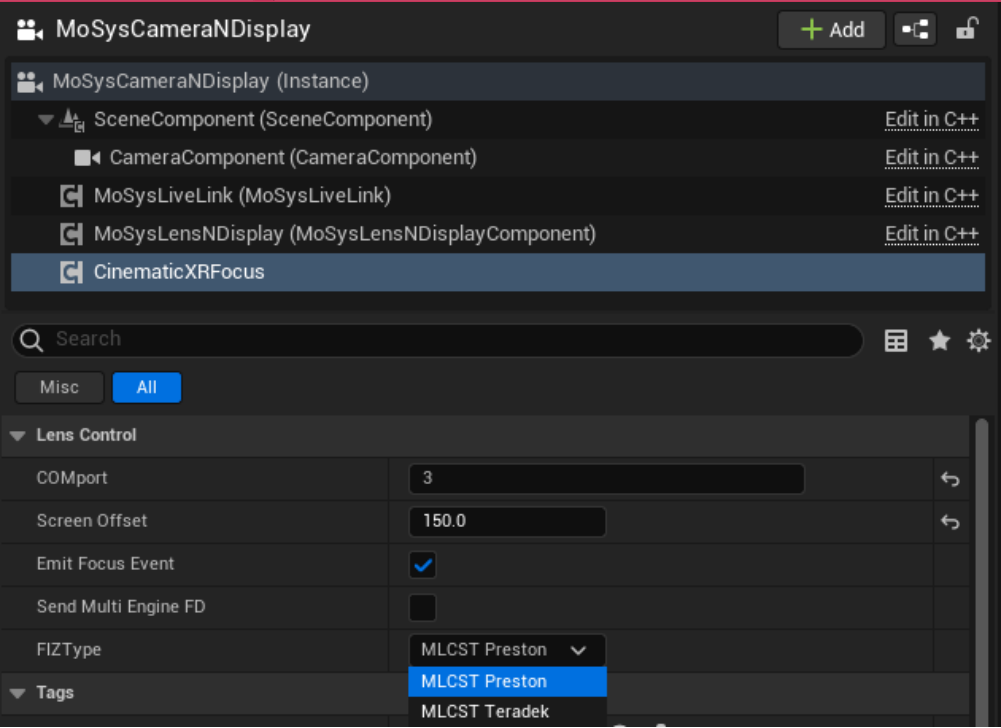
Open the Unreal project on Primary Render Node.

Verify a collision profile is setup in Project settings > Collision > Object Channels. There should be a custom Object Channel called LED. Create one if missing.

Add a new component on MoSysCameraNDisplay (used for ICVFX) called CinematicXRFocus.

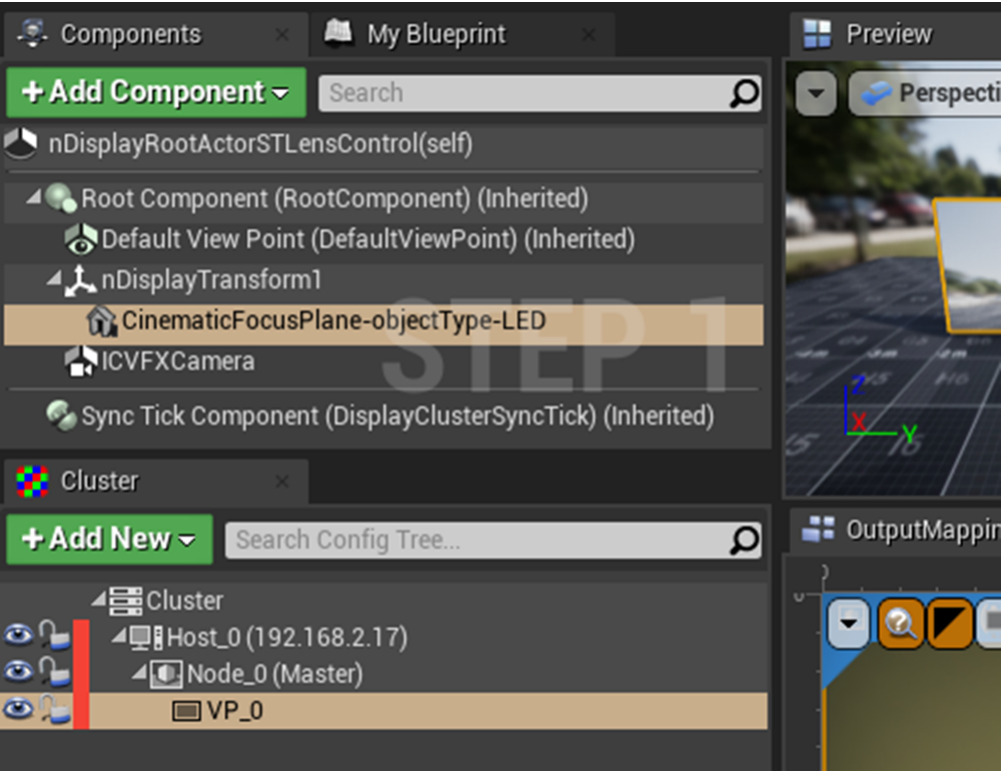
Configure the component:

1. Enter the com port used for communication.
2. Check Emit Focus Event if using multiple rendering nodes.
3. Specify a distance in [cm] from the LED screen. The physical focus should stop while pulling focus into the virtual scene, to mitigate the Moire Effect, which appears when LED array is in focus. The inserted distance should allow the system to keep it just outside of the depth of field.

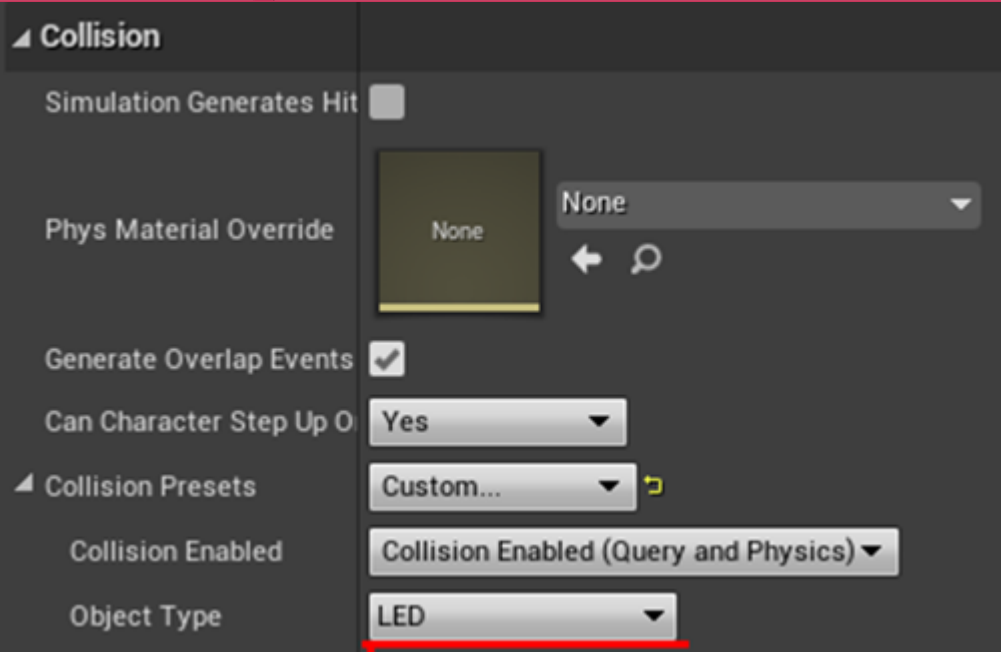


In nDisplay configuration blueprint, set the collision presets in collision section on the mesh(es) used with viewports as follows:

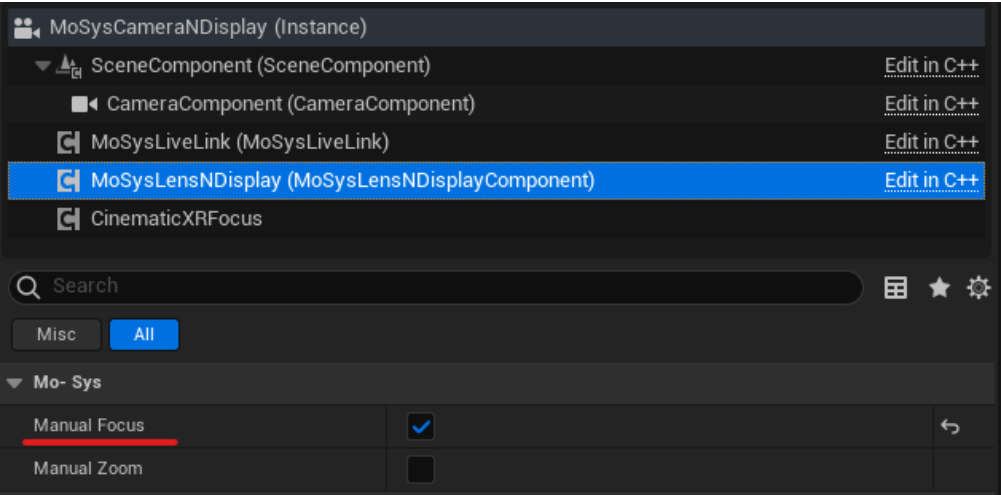
Collision Presets set to Custom and below Object Type is set to LED.







Finally, set Manual Focus to true on MoSysLensNDisplay component to enable Hand control to set the focus distance.

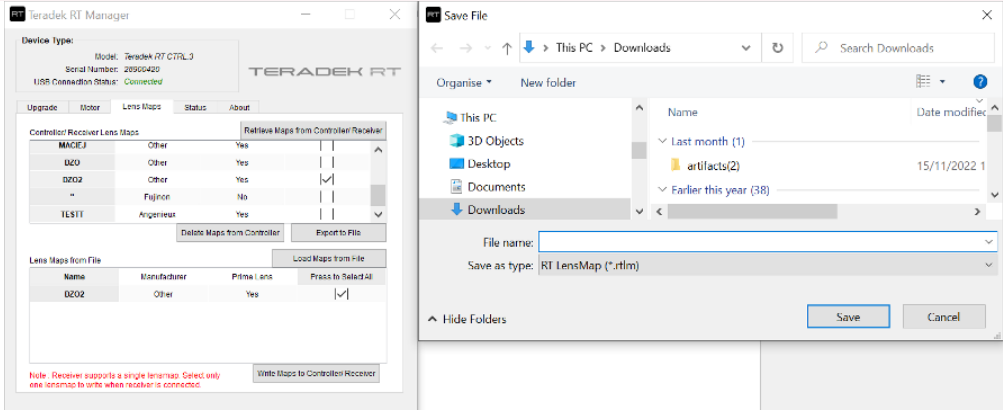


A sample level nDisplayLensControl with the component already on the camera can be found in: MoSysVPPro Content > nDisplay > LensControl.

## 11.6 Teradek RT Lensmap Setup

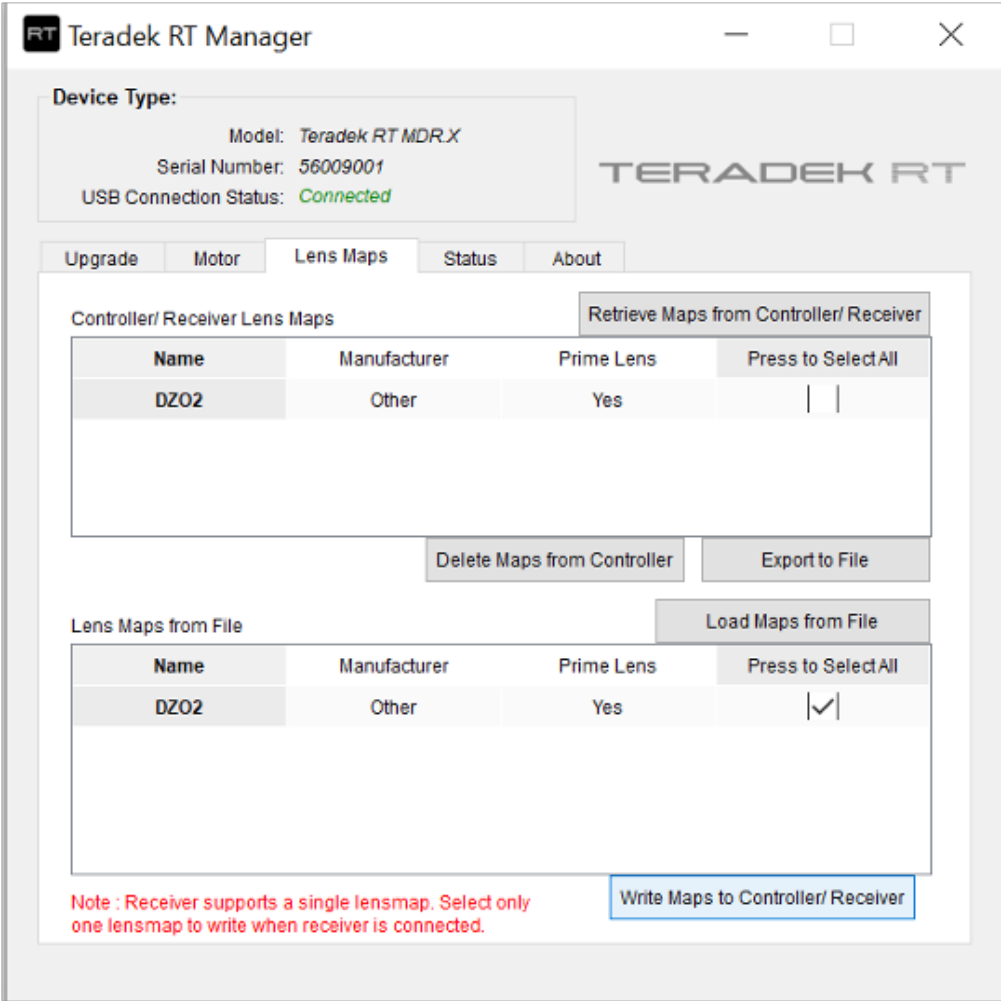
The lensmap created on the hand controller needs to be exported to the PC and then loaded to the MDR for the system to work correctly. At all times the Lensmap must be the same on both hand controller and the MDR.

1. Connect the hand controller with a mini USB cable to the Primary Render Node.
2. Open Teradek RT Manager and go to Lens Maps tab. Click Retrieve Maps From Controller/ Receiver. Select the Lensmap to be copied and export as file



- 3. Connect the MDR with the same mini USB cable to Primary Render Node. Select the Lensmap from file, by using Load Maps from File button. Then click Write Maps to Controller/Receiver.

Teradek RT Manager must be installed on the Primary Render Node to manage the file exchange (<https://teradek.com/pages/downloads#teradekrt>).



## 12. XR Engine Setup

### 12.1 Overview

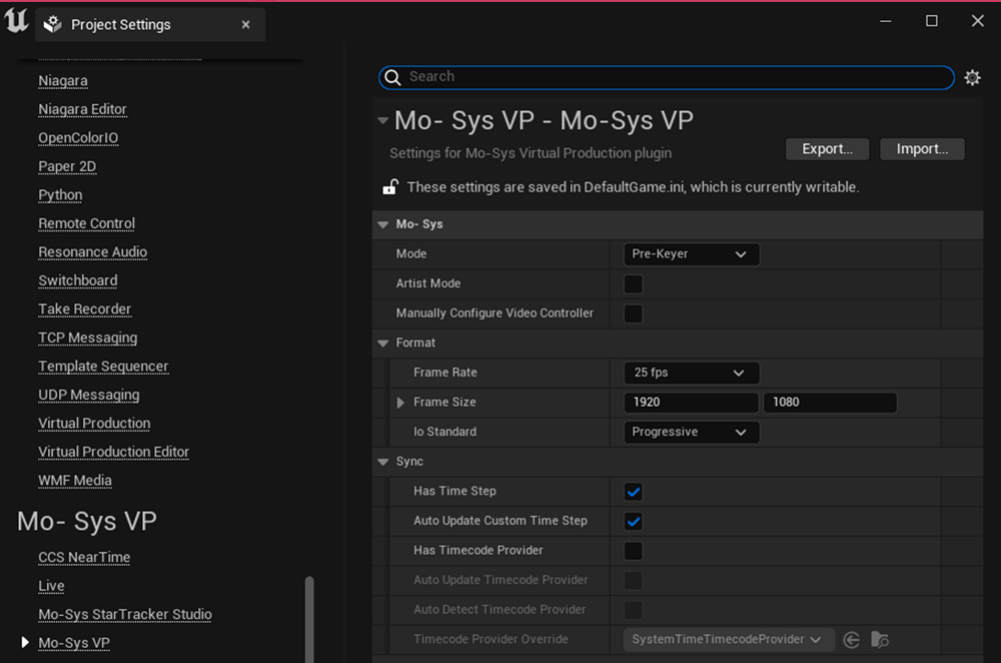
The XR Engine uses camera feed as an input and composites it with AR objects and virtually extends the set when camera points away from the LED screen. This means the talent is only visible if the screen is behind them.

**Currently we support both internal and external compositing using a mixer/keyer.**

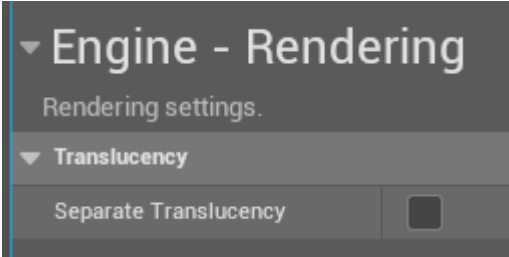


### 12.2 Settings

In Project Settings > Mo-Sys VP Pro: - Define the image framerate and resolution. - Uncheck Has Timecode Provider under Sync.



- If there are transparent actors, unset Separate Translucency.



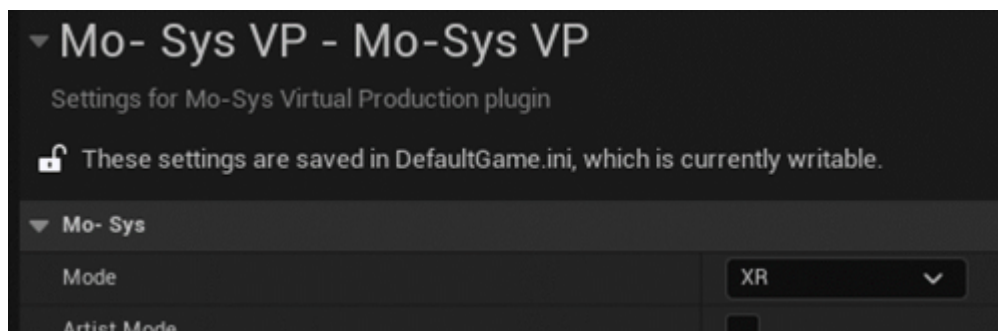
### 12.3 StarTracker

Setup one of the StarTracker's data output to send tracking data to XR Engine's IP address.

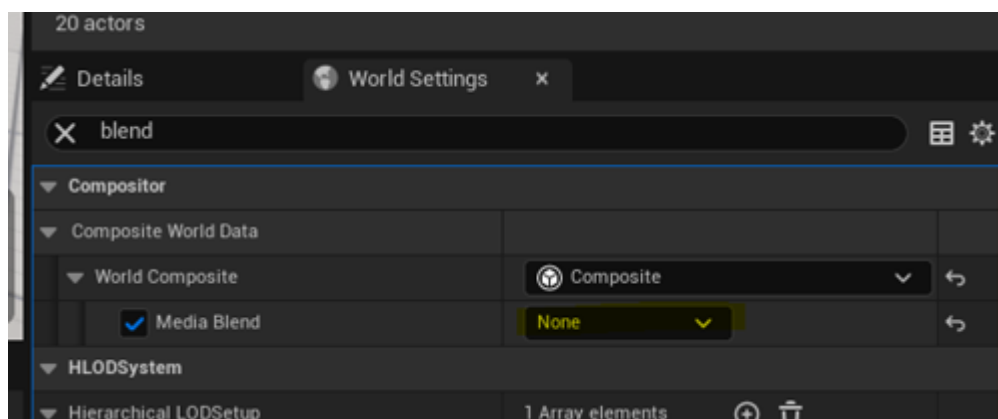
## 13. XR Engine Setup - Compositing

### 13.1 Internal Compositing

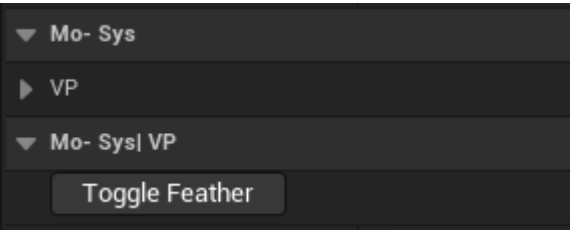
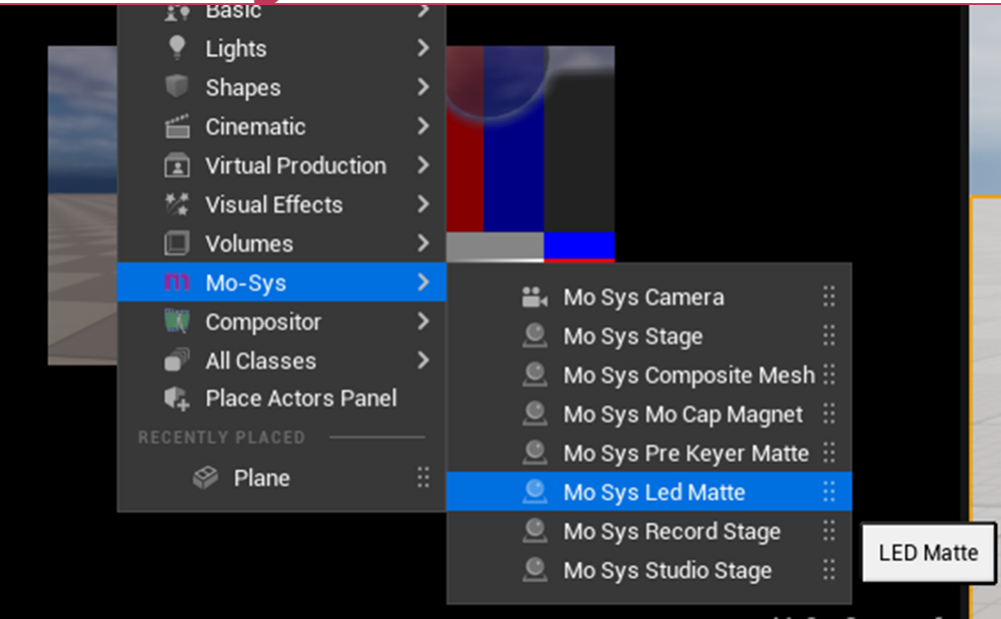
1. Use the same project as the nDisplay on Render Nodes one. The nDisplay configuration actor should be present in the level.
2. Set Mo-Sys > Mode > XR.



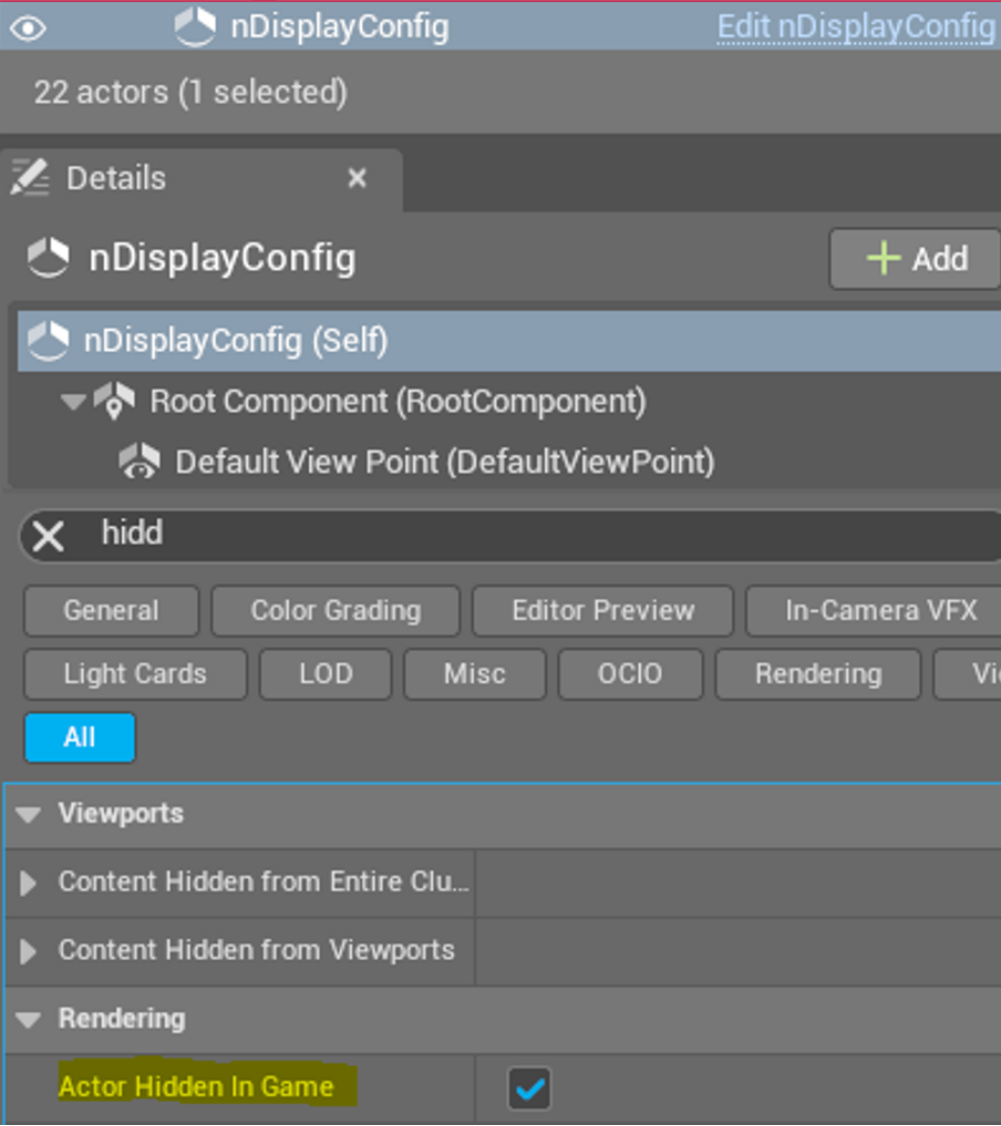
1. Set Composite World Data > World Composite > Media Blend to None.



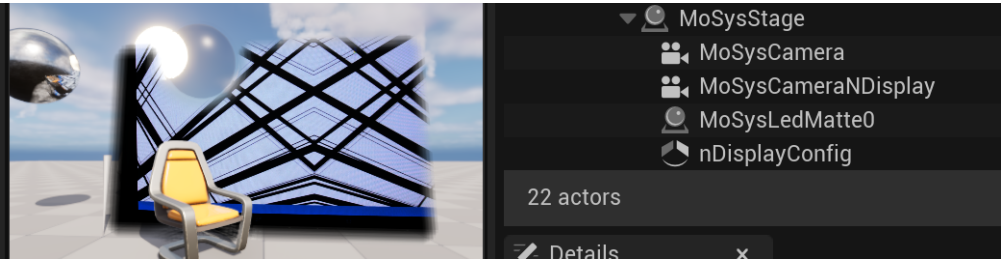
1. Follow **4.6.1 Enable Video In VPPro Manual** to set up video input and output.
2. To see the example setup for this compositing mode, open the LMoSysXRExample map from MoSysVPPro Content > StarterContent > Maps.
3. In the level you intend to use XR, spawn MoSysStage. This in turn should automatically spawn MoSysLedMatte to mask the volume, so the composite lets the video through that area. It should cover the nDisplayConfig mesh(es) size 1:1 (or scaled down to ~0.95 for extra margin). Extra MoSysLedMatte can be added if needed.
4. MoSysLedMatte has assigned custom stencil of 247 if feathering is to be used and 245 if feathering not set (Use Toggle Feather button on the matte to flip between the two). All the AR objects should be assigned 246 custom stencil and be tagged MoSysAR. The AR objects can be also assigned using XR Controller panel 13.6.



1. Set nDisplayConfig Actor Hidden In Game.



1. Final stage structure



13.2 Live Link Preset

Adapt the existing Live Link preset to enable receiving tracking on XR engine for MoSysCamera(s).

Change the IP address to correspond to the machine that is supposed to receive data. The setting is in: Mo-Sys UDP source > Tracking > Network interface IP Address.

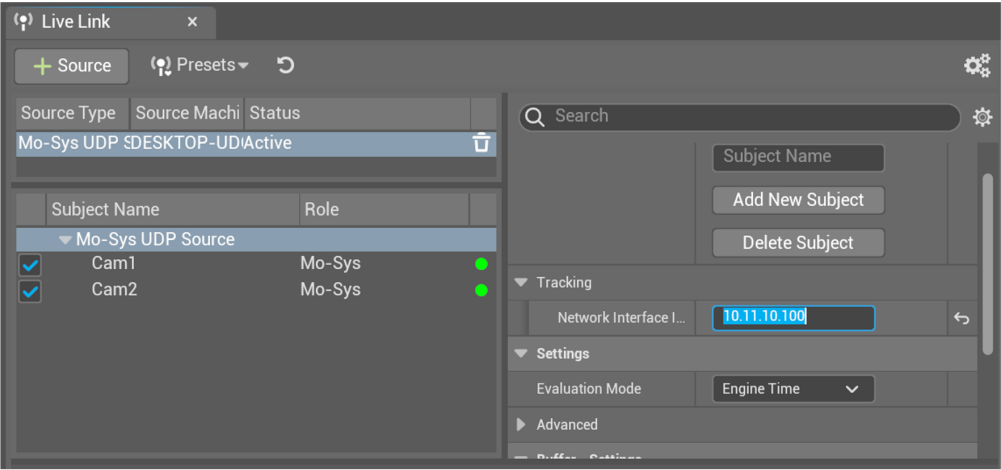
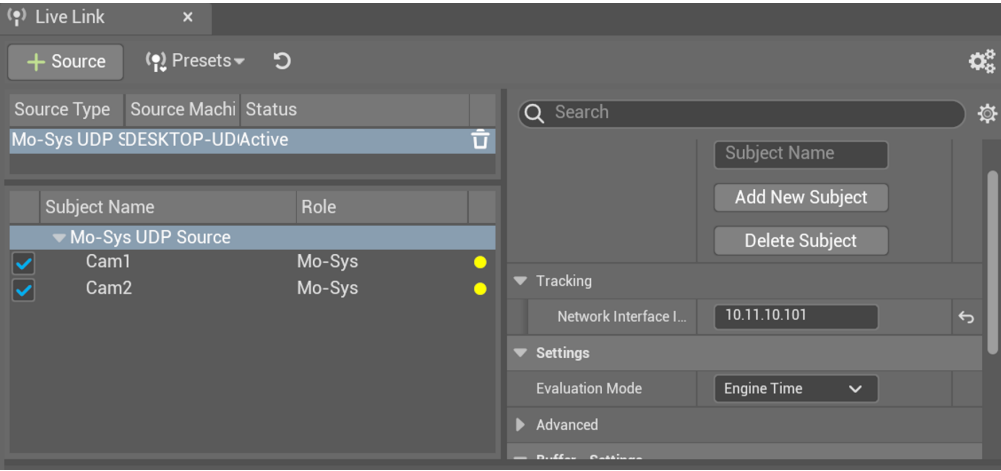


If XR PC has IP: 10.11.10.100, set it there and save as a separate Live Link preset. Select as the default Live Link Preset:



Apply Live Link subject only to MoSysCamera(s), as the nDisplay camera(s) should already have its own preset and subject assigned.

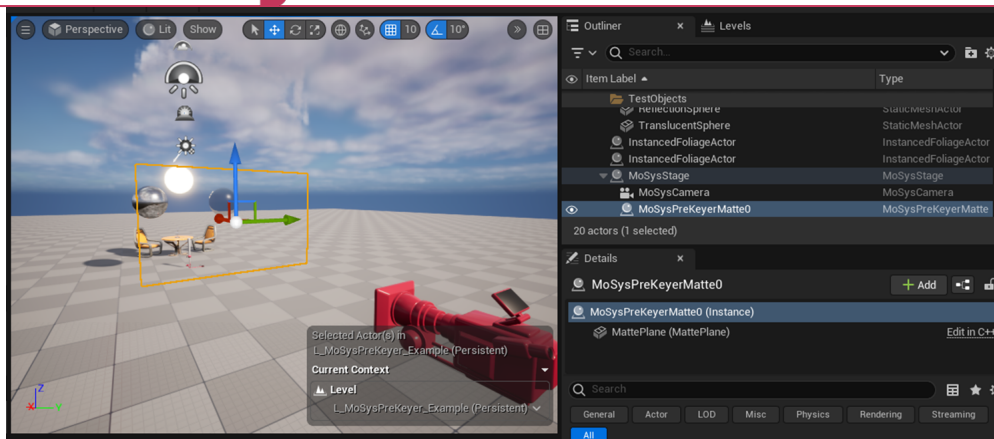
**Be careful of synchronising the configuration folder as the Render Nodes must store a different default Live Link preset! The default preset is stored in DefaultGame.ini.**



Img. XR preset

### 13.3 External Compositing

Set up the project to use pre-keyer mode (refer to **VPPro Manual - 4.9 Pre Keyer Mode**).

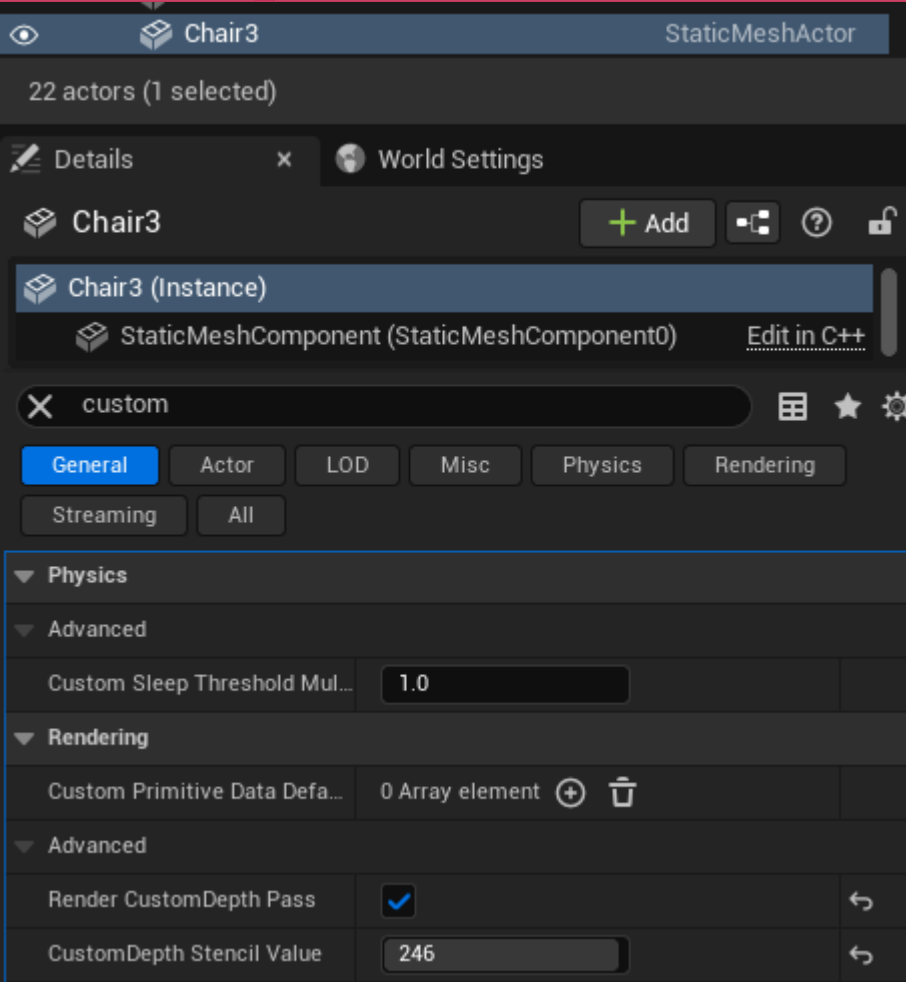


## Pre-keyer Matte Setup

Use pre-keyer matte to mask the LED wall. The mesh for the wall must be placed at the same position and rotation as in the nDisplayConfig. This will give the matte, that can be used to composite, the video with CG fill.

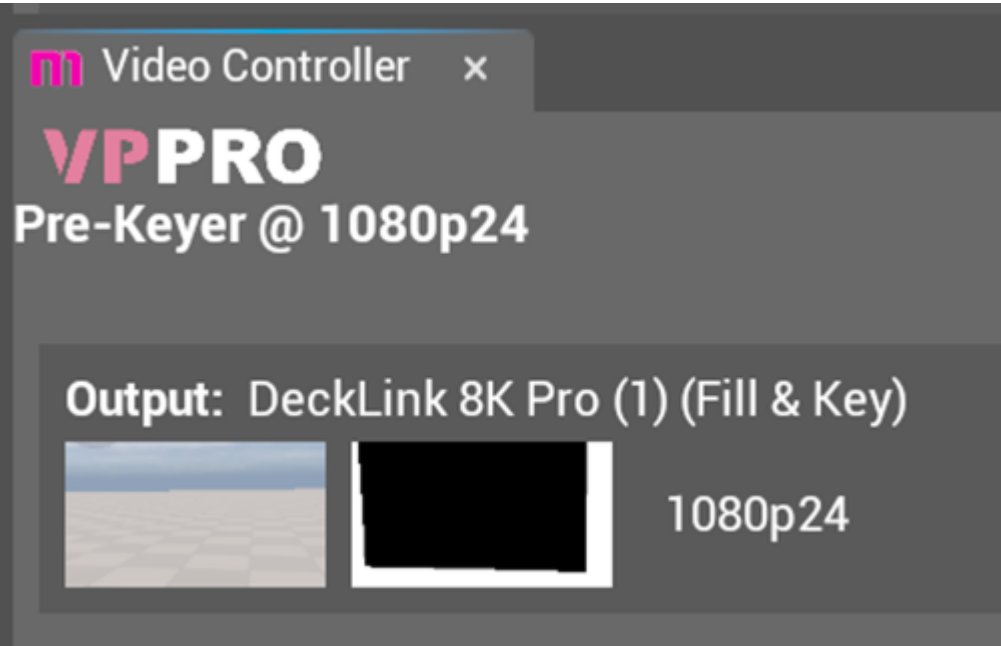
Pre-keyer matte has a custom stencil of 247.

Any object that is supposed to be in front of the LED wall (augmented - AR) needs to have custom stencil set to 246 and be tagged MoSysAR. You can use XR Controller panel to toggle actor to AR. Note that any objects that is not in front of the LED can't be changed to AR implicitly. All those objects are in front layer anyway (set extension).



Video Outputs

Pre-keyer mode provides 2 video outputs. One for the CG perspective of the camera (fill) and the LED matte for composition.

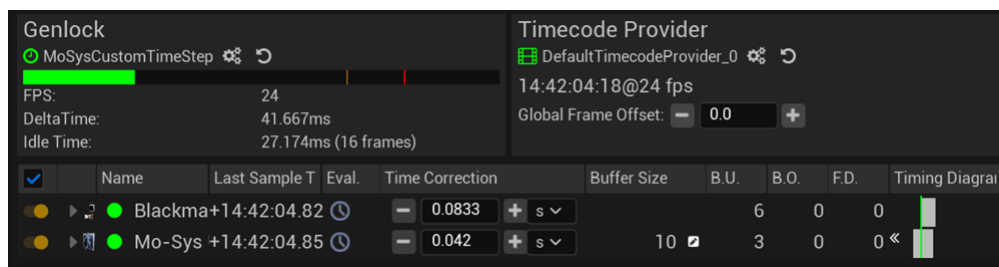


## 13.4 Delays Correction with Timed Data Monitor

Open the panel from Window/Developer Tools. If it's not present, enable the Timed Data Monitor plugin.

Use Time Correction value (in seconds) to delay the tracking. Watch Timing Diagram and make sure the vertical bar is green and safely inside the buffer. If it's not, set the Buffer Size to a higher value.

Use the augmented cones as reference to set the delays. Pan the camera in a series of short, sharp movements to see if the cones stay on the corners of the LED. Stop between the movements to distinguish whether the cones or the video moves first. Adjust the Time Correction. Time Correction for tracking is saved with Live Link preset, whereas the video delay is set in Blackmagic source file. If not saved there the values will default to 0 after the project is reloaded or video dropped.



## 13.5 Colour Calibration Procedure

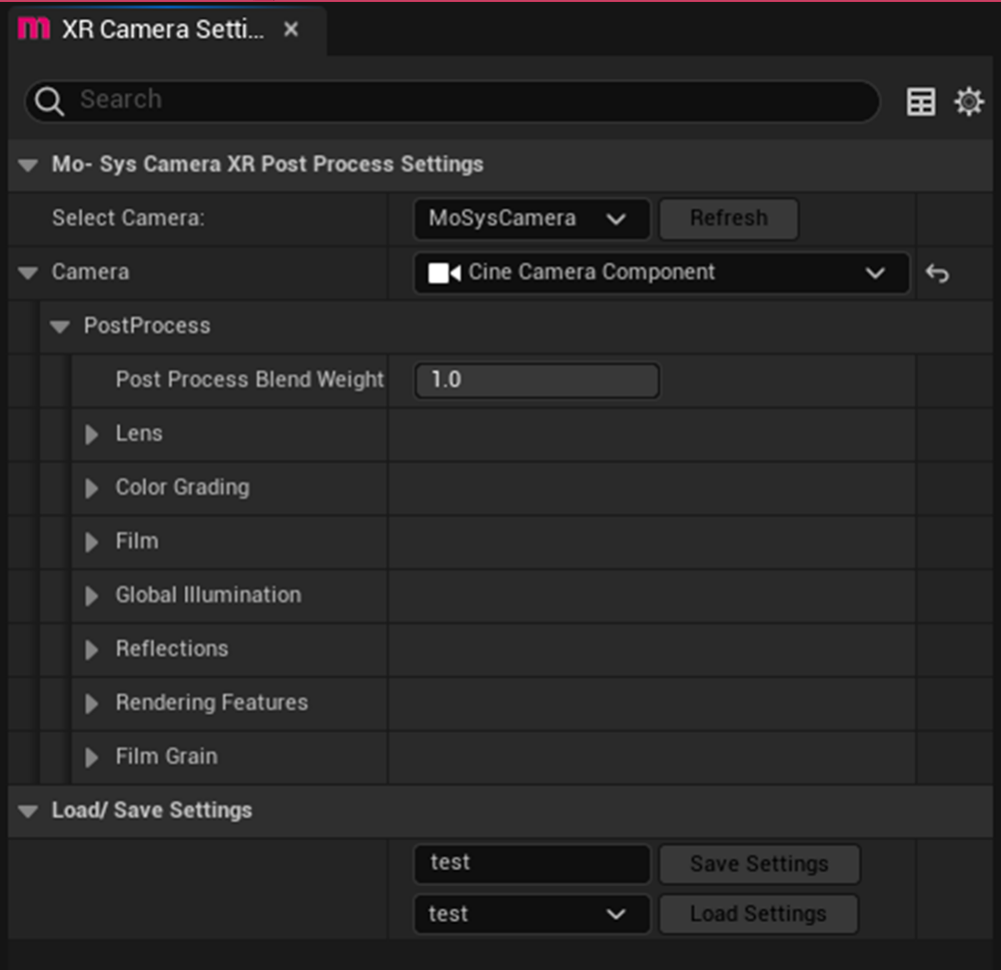
The standard workflow of colour adjustments using set extensions is as follows:

1. Colour adjustments on real camera and lighting.
2. Adjusting the exposure and colour grading on LED wall (nDisplay through multi-user session).
3. Adjusting the MoSysCamera to match the set extension to the image on the wall.

At this step we assume the first two steps are done. Use XR Camera Settings to adjust the set extensions.

Open the panel starting from the Mo-Sys icon.

1. Click Refresh.
2. Select the camera in dropdown menu.
3. Use PostProcess parameters to correct the CG fill.
4. Once the composite is matched, save the correction by giving the settings profile a name and clicking.
5. The profile can be loaded at any point by using the dropdown and Load Settings button.

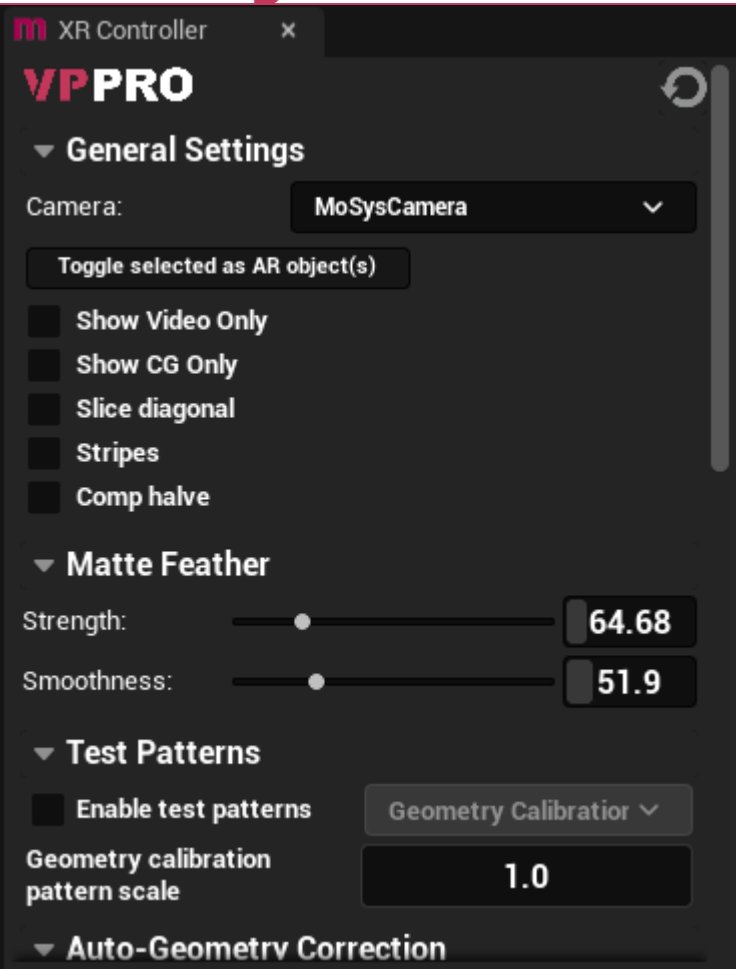


13.6 XR Controller Panel

XR Controller exposes some of the settings used to set extension. This includes:

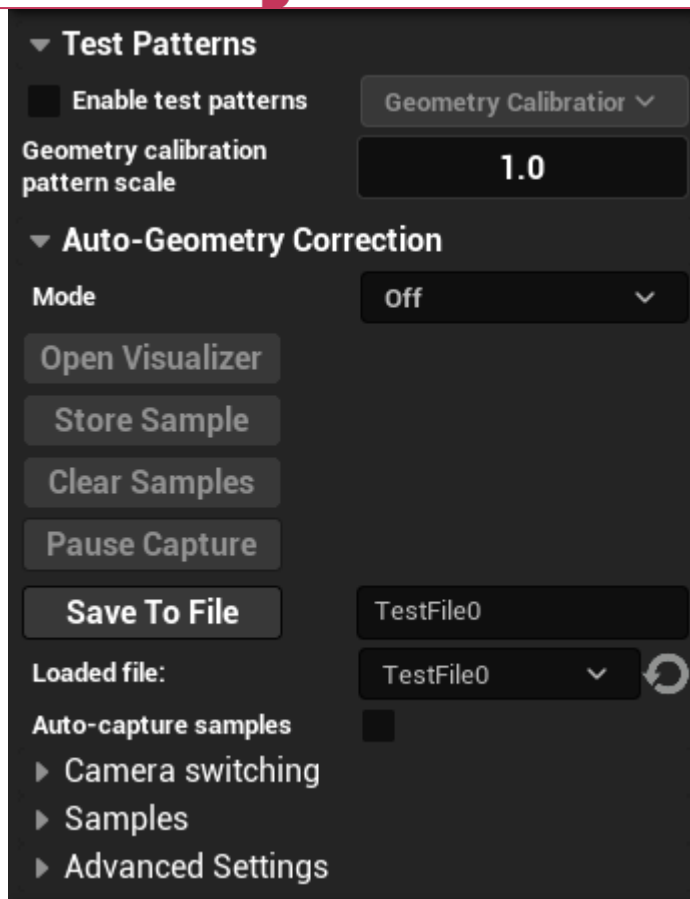
- Blend control between the LED screen and the extension (Feathering and Feather Quality).
- Toggle simple mesh objects to be AR (visible in front of the LED).
- Showing just the video input, CG, or combinations of both for troubleshooting.
- Auto-Geometry Correction settings used for better geometric match between LED and set extension (13.7)

If in Play Mode, click Refresh button to be able to change parameters dynamically.



### 13.7 Auto-Geometry Correction

Auto-Geometry Correction is a tool that allows for a seamless transition between the content in the set extensions and the content on the LED screen. It compares the CG-only image in the XR scene with the video image of the LED screen and then computes a homography correction that can be applied to the XR composite.

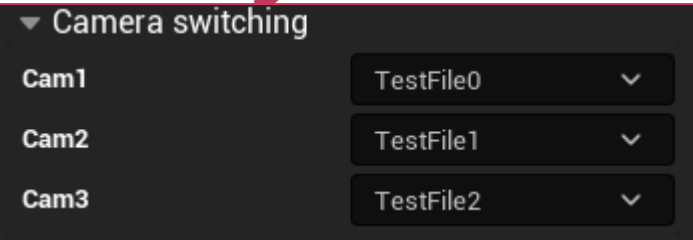


Steps on how to use it:

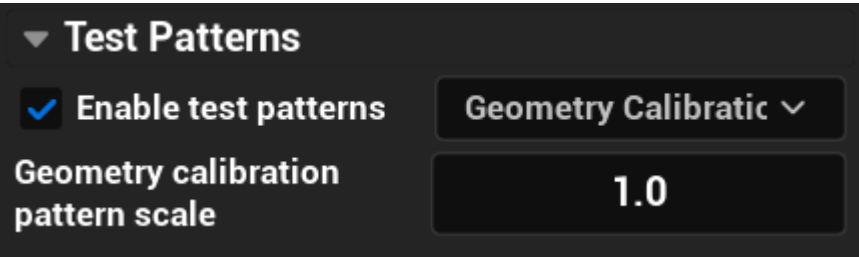
1. In Mode dropdown select Live Capture.
2. Check Auto-capture samples to capture samples automatically when camera stops moving.
3. Pan the camera so the LED screen is on the edge of the frame.
4. If tilt capture needed, lock the pan and change tilt so the LED screen is at the top or bottom of frame.
5. Change tilt to 3-4 more positions stopping after each one.
6. Pan the camera to 3-4 more positions and repeat steps 4-5 after each position.
7. If capture at different zooms needed, repeat steps 3-6 for 3-4 different zoom positions.
8. Move the camera to a different location and repeat steps 3-7.
9. Keep repeating step 8 depending on the size of the area used. The bigger the area the more samples are needed.
10. Enter the name of your save file and click Save To File.
11. Click the refresh button and select your save file.
12. Set Mode to Read From File.

Once you have the save file you can load it and use it with Read From File mode skipping the live capture steps. If you have multiple MoSysCameras in the scene, you can set a file for each of them in Camera Switching section.





**Note:** For better results if your Unreal scene doesn't have enough distinct features you can enable a Geometry Calibration pattern. For the calibration pattern to work, make sure to set up your Mo-Sys VP - Live settings correctly. Refer to points 4 and 5 from section .



In Test Patterns section, check Enable test patterns checkbox and select Geometry Calibration in the dropdown menu. Geometry calibration pattern scale can be adjusted to achieve better results for different camera field of views and distances to the LED screen. When finished with capturing samples, uncheck Enable test patterns checkbox to remove the calibration pattern.

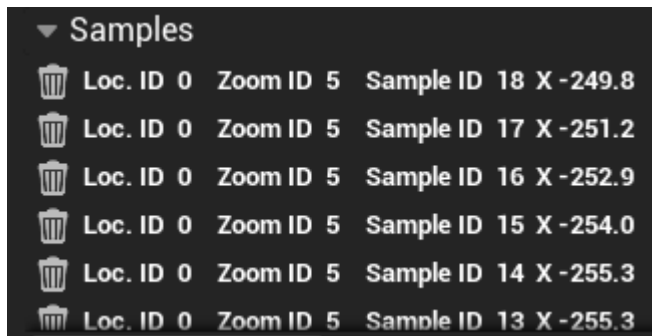
**Tip:** If geometry correction is unstable during live capture, here's a few things to try to improve feature matching: - Increase Geometry calibration pattern scale in Test Patterns section. - Lower Detection threshold in Advanced Settings section. - Open iris more on the lens. - Turn off lights that could be reflecting off the LED screen.

Other features of geometry match panel:

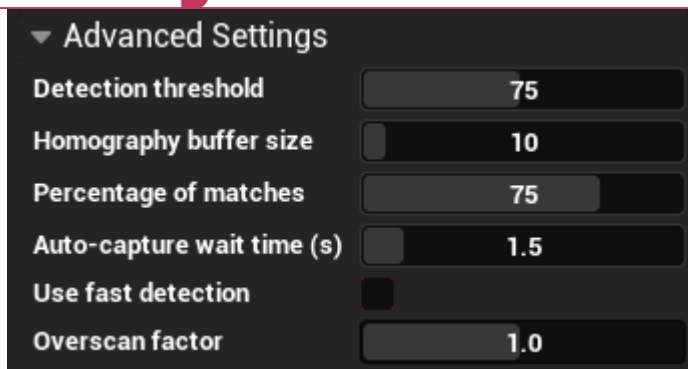
- Store Sample - can be used to store samples manually when Auto-capture samples is not set.
- Clear Samples - clears all currently stored samples.
- Visualizer - visualizes feature detection / matching for debug purposes. Can choose between Matches, Error and Overview.



- Pause Capture - holds the current homography in Live Capture mode. Can be used to evaluate correction with homography calibration features disabled.
- Samples - allows to view currently stored samples, delete specific ones.



- Detection threshold (Advanced Settings) - can be adjusted if there's not enough features detected.
- Homography buffer size (Advanced Settings) - increase to reduce noise, reduce to increase responsiveness.
- Percentage of matches (Advanced Settings) - percentage of all matches to use. Can be increased if limited amount of features detected. Lower amount for higher quality matches.
- Auto-capture wait time (s) (Advanced Settings) - time to wait in seconds after camera movement before capturing a sample.
- Use fast detection - enables faster detection but it can be noisier.
- Overscan factor (Advanced Settings) - can be used to manually set overscan factor.



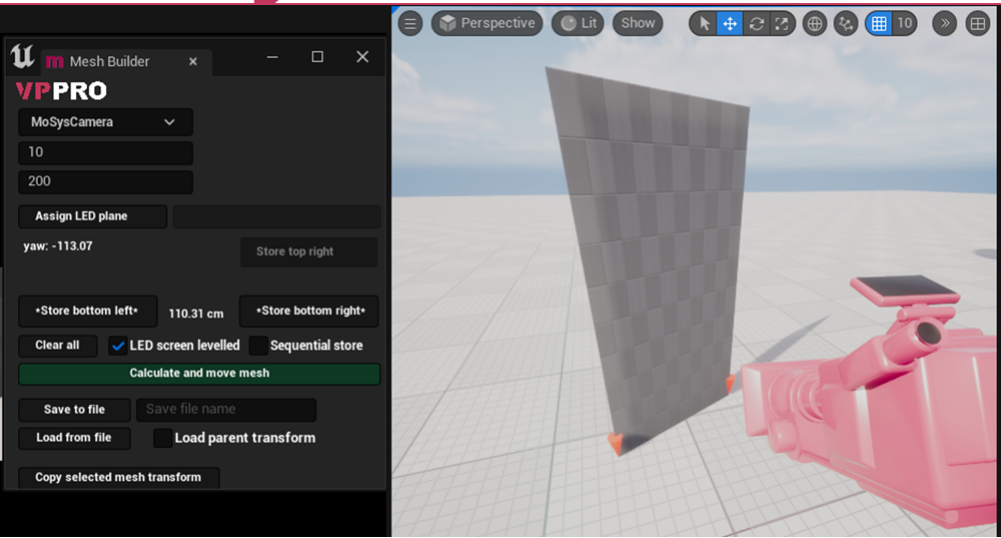
## 13.8 MeshBuilder

You can use Mesh Builder to generate the representation of the LED wall, by looking through a camera with the crosshair at the corners of the screen. Two or three observations are used for calculations.

The tool was designed for rectangular screen or a combination of rectangular screens. However, spawned corners can facilitate placing a more complex mesh as well.

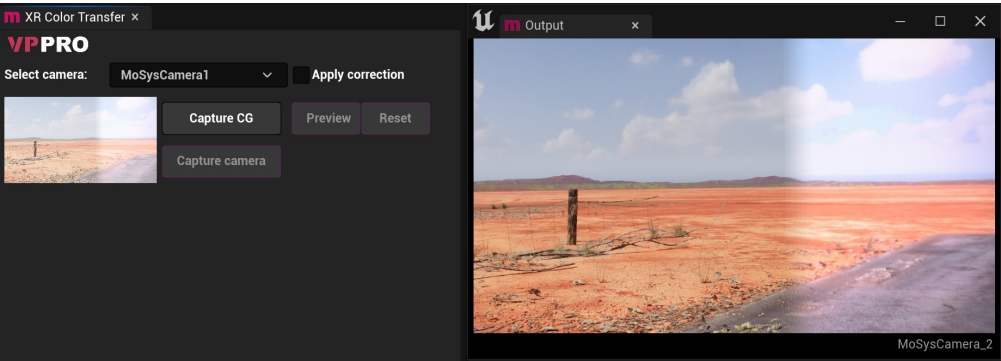
The procedure of finding the corners and generating a mesh is as follows:

1. Select a tracked camera used for observations.
2. Enter height of the bottom edge of the LED screen and the height of the screen from the bottom to the upper edge.
3. Select a plane to fit as the LED screen representation and click: Assign LED Plane. If there is no plane assigned, it is spawned automatically.
4. Position the camera so it is looking centrally at consecutive corners of the screen (through crosshair) and click on corresponding store buttons. Each time it will bring a cone indicating the corner position.
5. Click Calculate and move mesh to fit the plane in place.
6. Optionally, check LED screen levelled to only use 2 bottom corners. The calculation assumes the screen is perfectly vertical.
7. There is additional information presented as to the orientation of the plane (yaw: -113.07) and distance between bottom corners (110.31 cm).



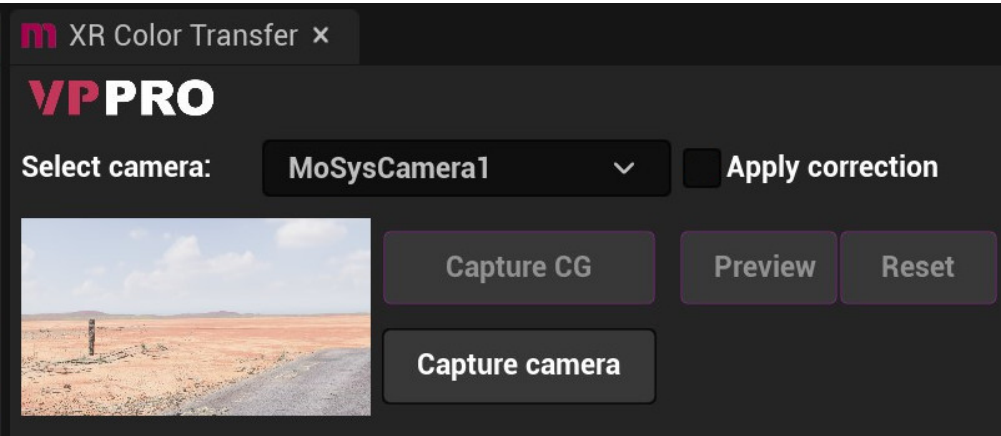
13.9 Color Transfer Panel

The panel can be used for a quick colour correction tool. It captures the incoming video and tries to adjust the set extension to match. It works per camera, if using multiple cameras select the one you wish to correct. Before starting, move your physical camera to a point where most of the image is filled with the LED. Make sure the MoSysCamera is tracked and the props or AR objects are not in the camera field of view.

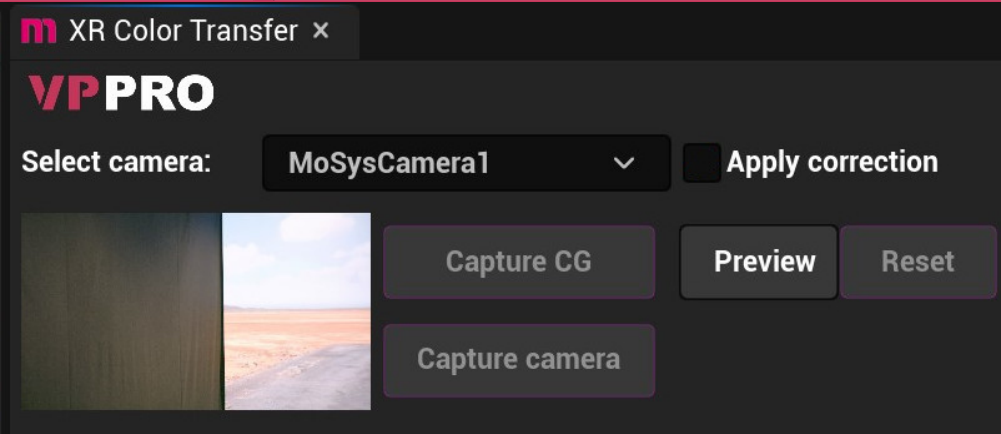


Follow the wizard:

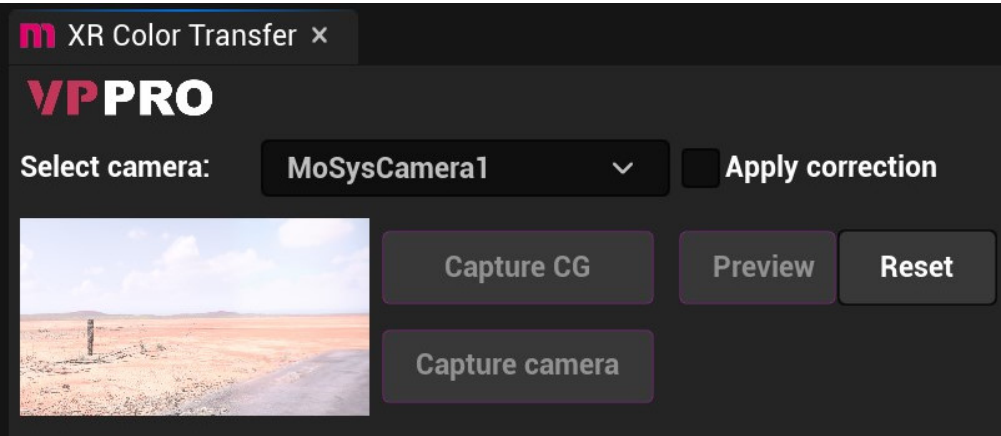
- 1. Click Capture CG.



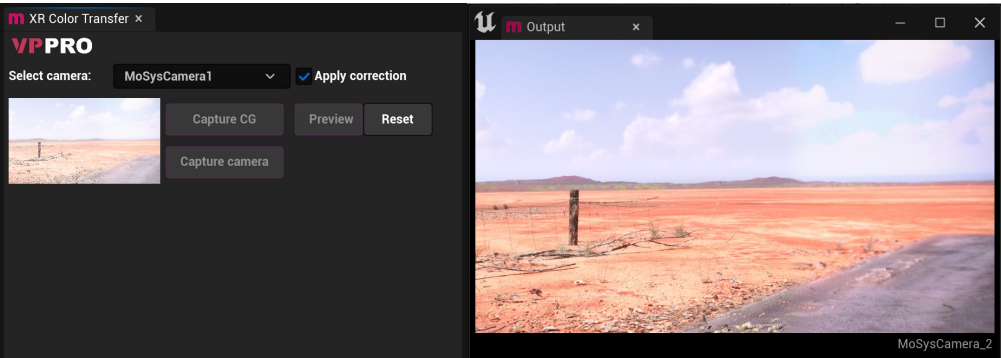
- 2. Click Capture Camera.



3. Click Preview.



The preview should show the colour corrected image. Apply it to Mo-Sys Output with Apply Correction checkbox.



You can start again with Reset.

## 14. XR Multi Cam

### 14.1 Overview

XR Multi Cam is a tool to seamlessly switch between cameras in Mo-Sys VP Pro XR. It sends a command to the Primary Render Node to update the perspective on the LED volume, and smooth transition can be achieved by adjusting the delays.

This can be seen in the sample level: LMoSysXRMultiCamExample.umap\_.

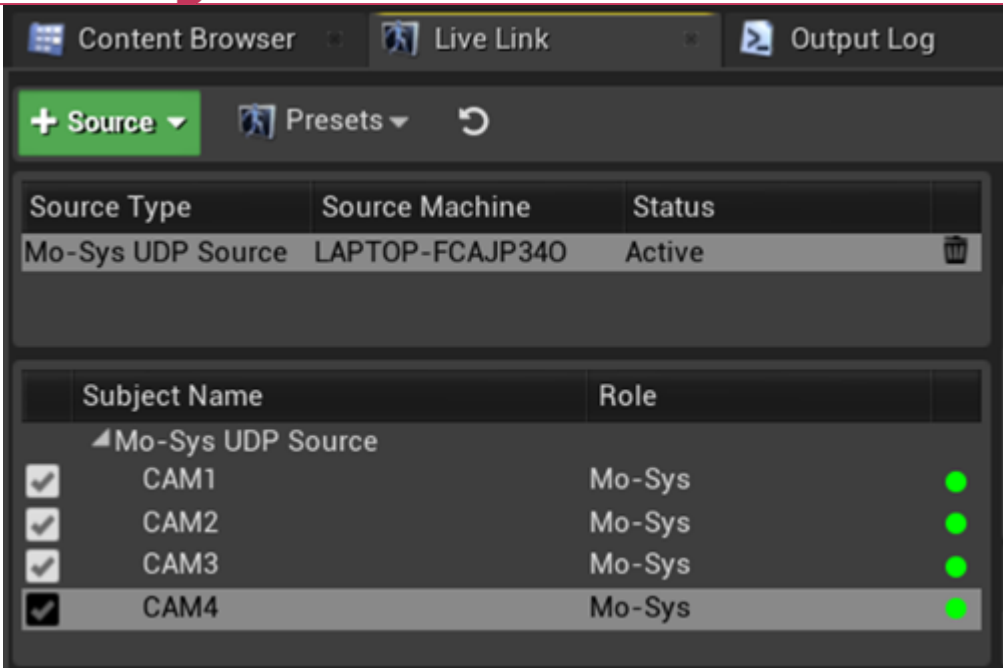
### 14.2 Hardware Prerequisites

- One of the integrated switchers: Blackmagic Design ATEM, Sony XVS Switcher or Grass Valley Switcher.
- All cameras need to be genlocked and set to the same FPS and Shutter speed, with identical output delay.
- The usual VP Pro XR setup previously specified in this manual.

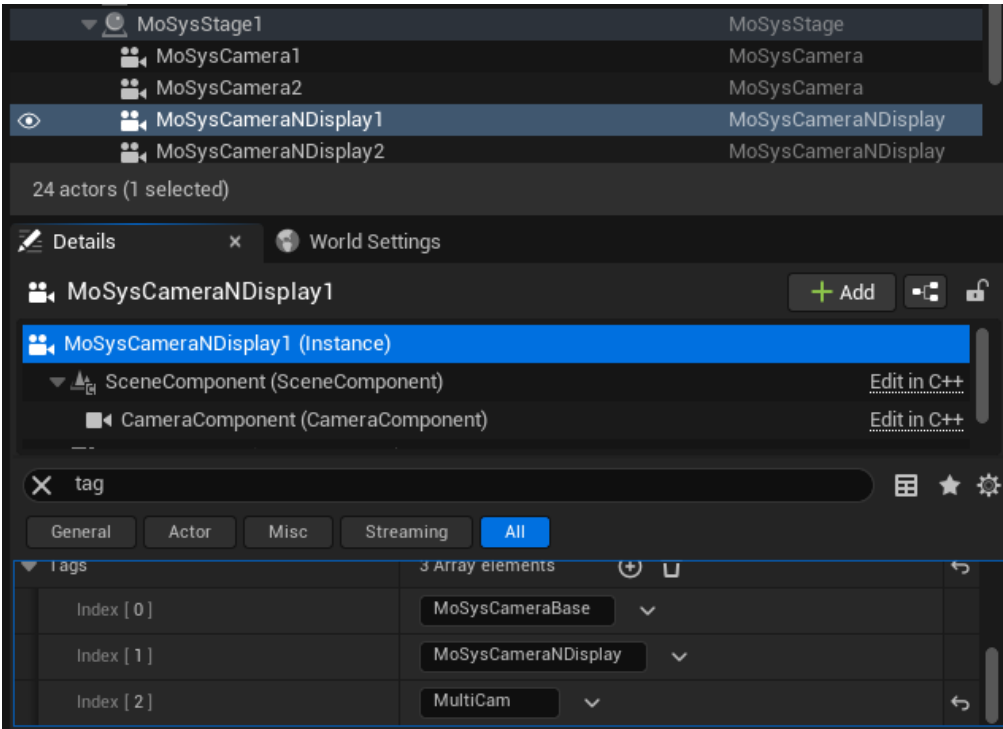
### 14.3 nDisplay Project Setup

- Add MoSysCameraNDisplay into the scene for each real camera to be used with XR Multi Cam. You can use the existing nDisplay scene with CineCameraActors, however, you need to add a MultiCam tag and two components: MoSysLiveLinkComponentController and MoSysLensNDisplay.
- Create a Live Link preset with a subject for each camera. Make sure the subject name has the camera index at the end, for example: CAM1 for camera one, CAM2 for camera two, etc.





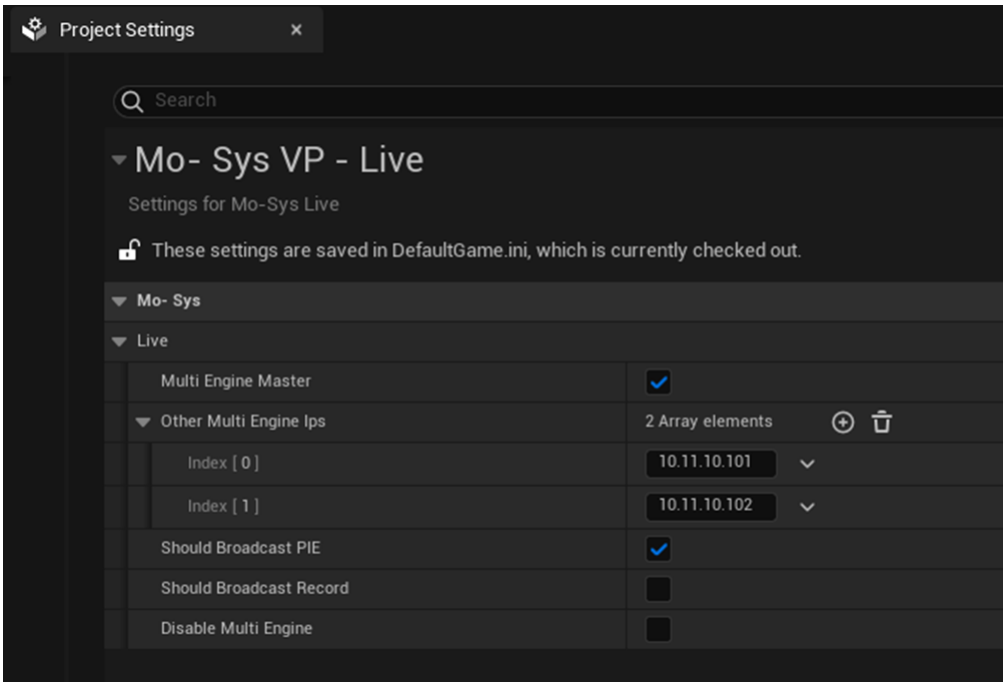
- Set this preset for all NDisplay cameras in the scene. Set the corresponding Live Link subject for each camera.
- Go to folder MoSysVPro Content > nDisplay > MultiCam in content browser.
- Find the asset called nDisplayMultiCamController and drop it anywhere in the scene. This is the blueprint that handles the perspective switch on the LED volume.



14.4 XR Engine Project Setup

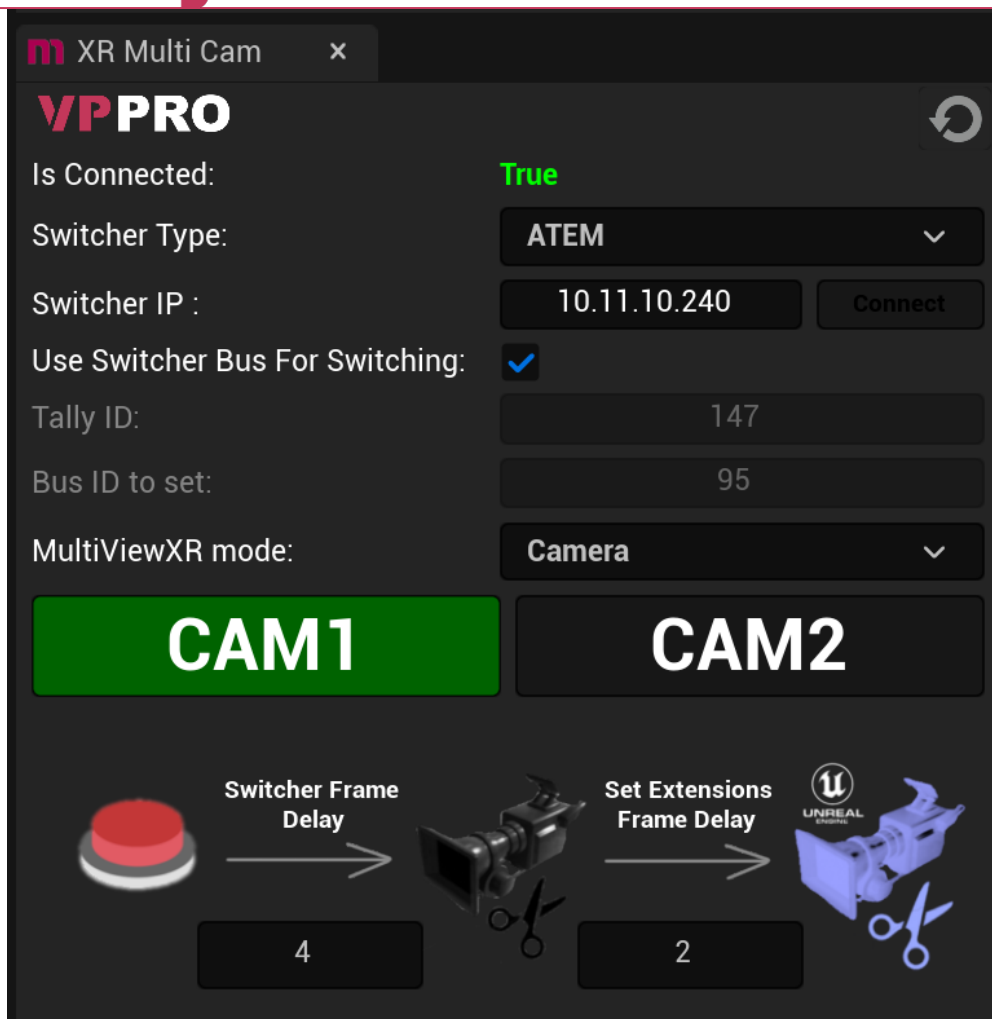
1. Add a MoSysCamera actor into the scene for each real camera to be used with XR Multi Cam. Camera indices will be assigned depending on actor names in alphabetic order.
2. Create a Live Link preset with a subject for each camera.

- 3. Set this preset for all Mo-Sys XR cameras in the scene. Set the corresponding Live Link subject for each camera.
- 4. Open Project settings > Mo-SysVP-Live.
- 5. Check on Multi Engine Master and add Other Multi Engine Ips corresponding to the nDisplay Render Nodes.



- 6. Open XR Multi Cam panel.





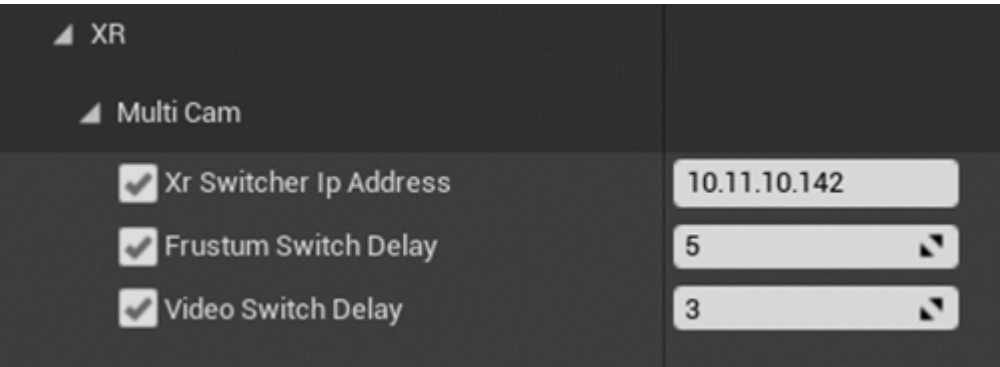
7. Enter the IP address of switcher and press enter key or click button Connect. If connection was successful, you should see Is Connected field show True in green. Otherwise you will see a False in red.
8. **'Frustum Switch Frame Delay'** specifies how many frames to wait before sending a request to the switcher to cut to another camera. This delay is needed to compensate for the extra time it takes for the perspective change to show up on the LED volume.
9. **'Video Switch Frame Delay'** specifies how many frames to wait before set extensions in the XR scene. This delay is needed to compensate for the time it takes for the switcher to cut to another camera. It starts to count after **Frustum Switch Frame Delay** transpires if positive, or before if negative value specified.
10. Refresh Cameras button needs to be used when cameras have been added/deleted in the scene.

## 14.5 If no XR PC - Experimental

Use an Editor PC and apply the settings from XR engine project setup. **Don't synchronise DefaultGame.ini in Config folder!**

14.6 Usage

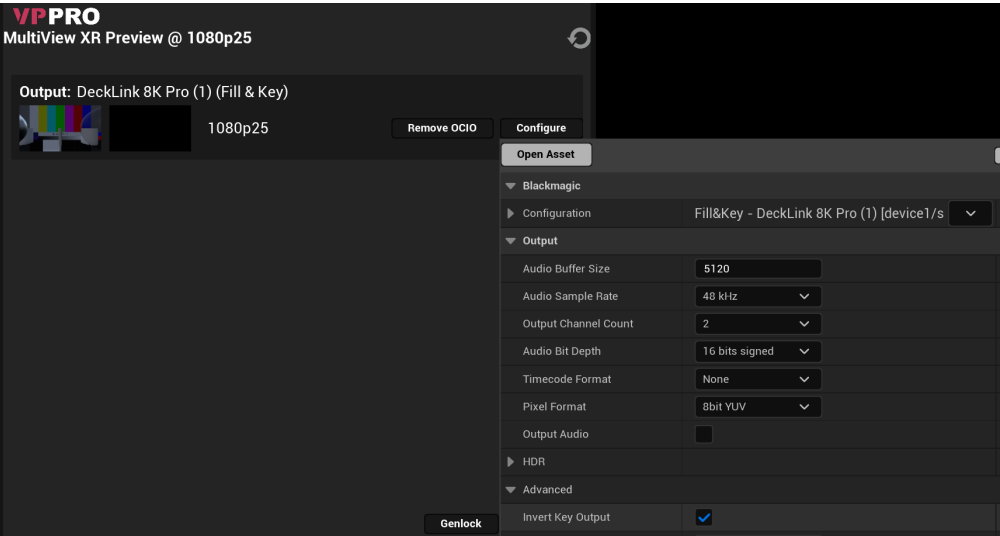
- Use a defined bus in the switcher to trigger the camera change.
- Alternatively, the cut can be triggered in the XR Multi Cam panel using buttons on the bottom of the panel. After clicking the button, camera switch will be initiated and the active camera button will turn green.
- Default values for XR Multi Cam can be set in the Mo-Sys VP Pro section of the Project Settings.



14.7 MultiViewXR director preview

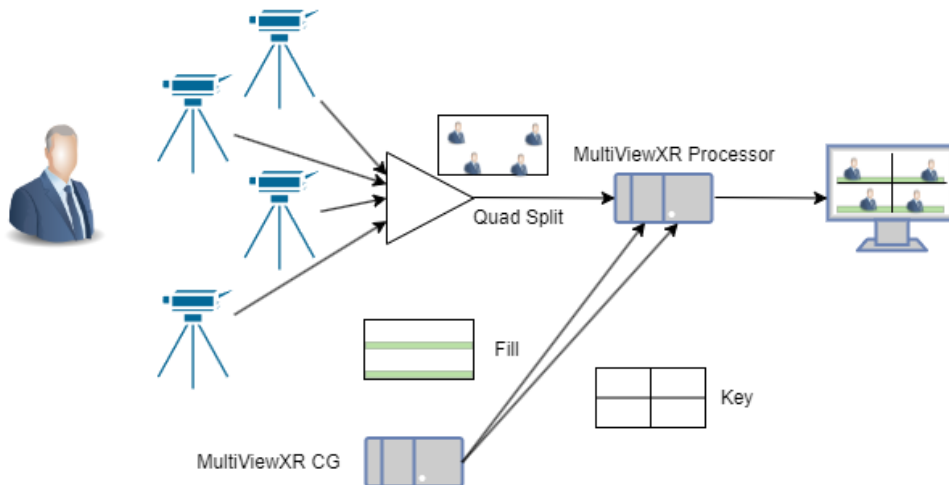
MultiViewXR can provide an AI-assisted director preview that removes the incorrect background from the non-live cameras and replaces them with the correct perspective background.

Two additional PCs are required. The first PC runs VP Pro in MultiViewXR Preview mode to generate the quad-split CG backgrounds. Project Settings > Mo-Sys VP > Mode. DeckLink SDI card configuration must match that of Pre-Keyer Mode. Note that sometimes inverting alpha is required on the output for the system to work correctly.



There is a sample scene in Plugins > MoSysVPPro > StarterContent > Maps > LMoSysXRMultiViewXRCGExample. This loads a sample overlay on the preview found in \_Plugins > MoSysVPPro > XR > \_ MultiViewXR\_ > MultiViewXROverlayWidget.

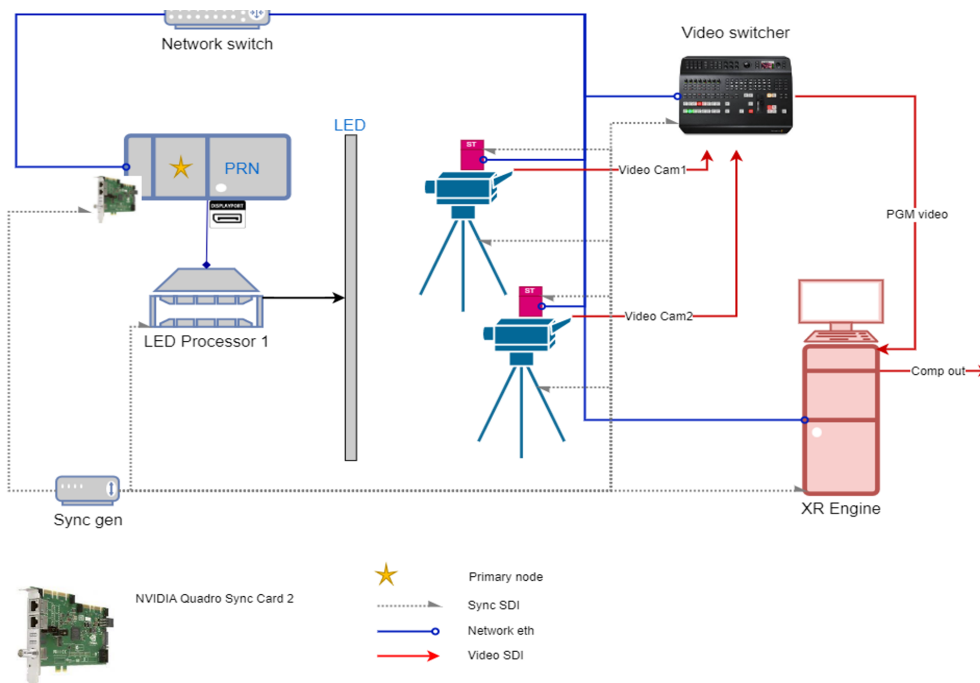
The fill and key outputs of this engine are fed - along with a quad split of the cameras - to the MultiViewXR processor as shown.



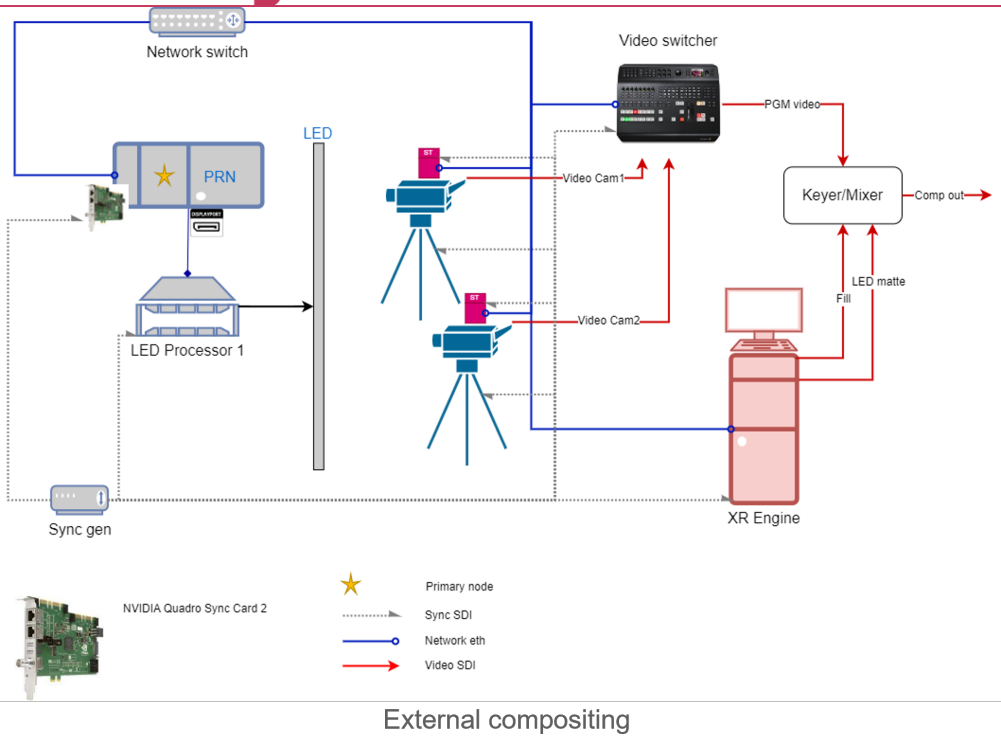
## 14.8 Tips

- To make it easier to set correct delays, it is recommended to set them one at a time. To help with this, the XR Controller panel can be used. If Show video only is set, only the video feed will be shown in the fill output without the set extensions. This way we can adjust Frustum Switch Delay until we get a clean video switch.
- After the correct value is given to Frustum Switch Delay, Show video can be unchecked again. Now, the Video Switch Delay can be adjusted by timing the set extension switch with the video switch.

## Internal compositing



## External compositing



## 15. FAQ and extra information

### 15.1 Mesh builder spawns a cone just in front of the camera or at another incorrect place

Go to Project Settings > Collision and verify it has a custom Object Channel: RefPlane.

### 15.2 Cinematic Focus XR doesn't stop the physical focus

Go to Project Settings > Collision and verify it has a custom Object Channel: LED.

Or

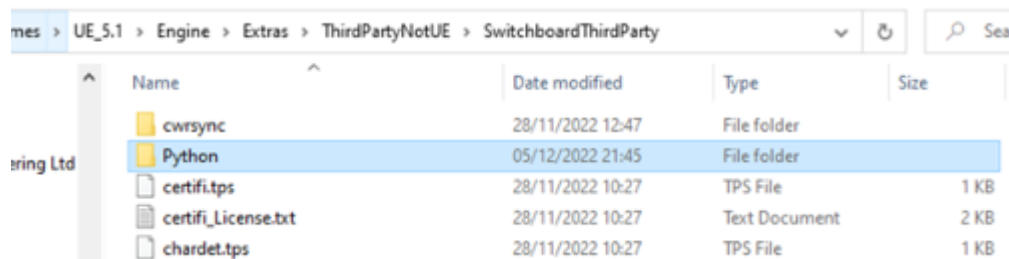
Make sure the Viewport's mesh is a Static mesh and not the nDisplayScreen. The mesh should have a collision preset set to LED.

### 15.3 How can I map a network drive?

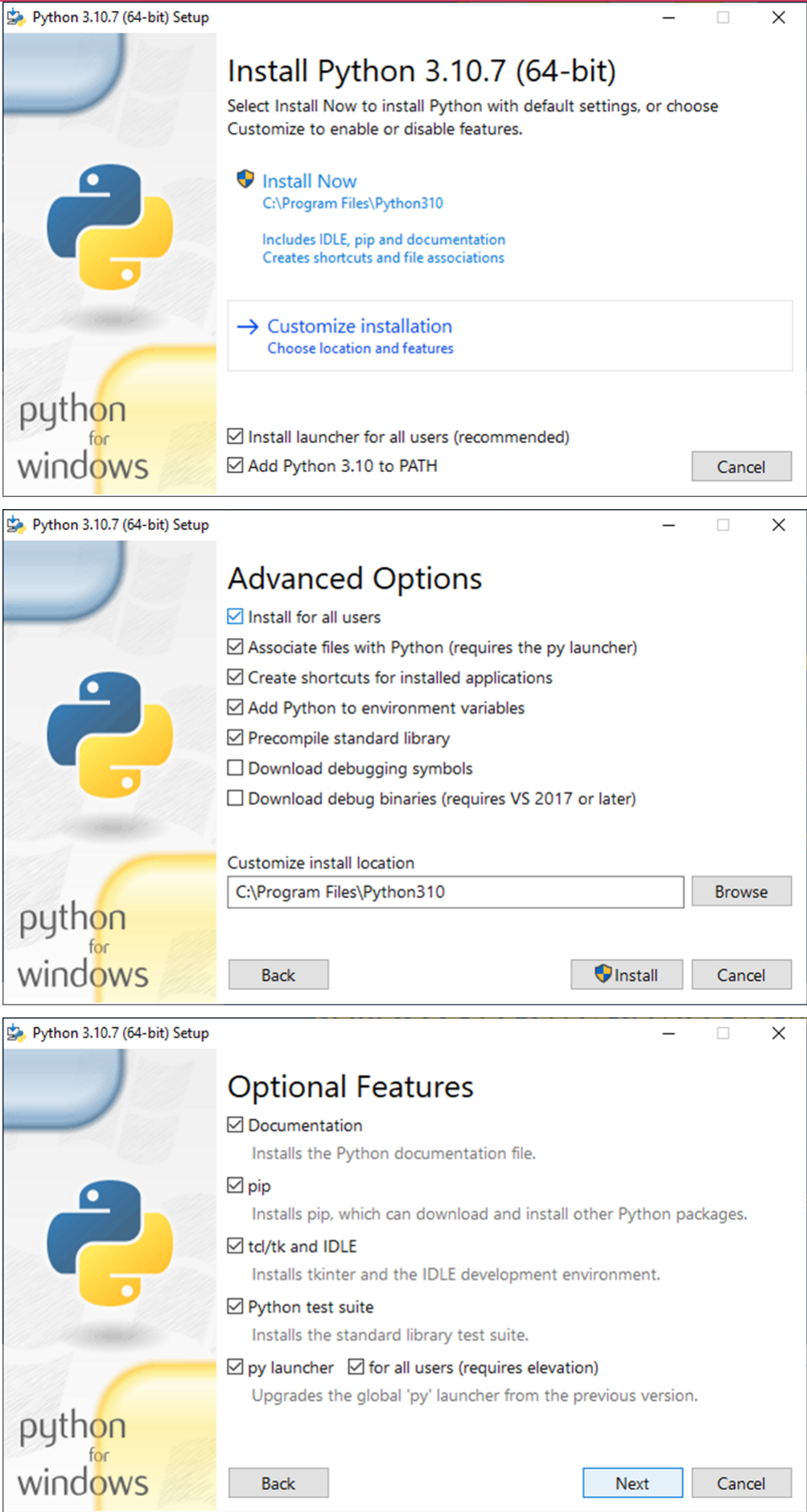
Refer to section .

### 15.4 Switchboard Launcher is crashing on start

1. Try running Switchboard from ..  
\\UE.x\\Engine\\Plugins\\VirtualProduction\\Switchboard\\Source\\Switchboard, \_Shift + right-click and open command there (Powershell) and run: ./switchboard.bat
2. If it's still crashing, regenerate third party folder for Switchboard. Delete Python folder and run the command from point 1 ..  
\\UE.x\\Engine\\Extras\\ThirdPartyNotUE\\SwitchboardThirdParty\\Python\_.



3. Sometimes the problem is that Unreal doesn't find Python's install path. In that case, reinstall Python and check install for all users.



4. Repeat point 2.

## **15.5 nDisplay not launching or crashes just after launch**

1. Check if Unreal is installed in the same directory on all computers / if the project is in the same directory.
2. It has been observed that if Render Node's user account name includes spaces it won't launch. The solution is to create a new Windows account user with no spaces in the name. Changing user name is not sufficient as it doesn't change the path to certain Unreal files.

## **15.6 XR no composite (only video or only CG)**

1. Verify the camera is looking at the LED and is not fully zoomed in.
2. Make sure the Mo-Sys Mode is set to XR.
3. Accept the initial settings when they pop up after Mo-Sys plugin installation.
4. Verify if in World Settings, under Composite World Data, the option WorldComposite Enabled is set to true.
5. Press Play.
6. Compare against a sample level in: MoSysVPProContent/StarterContent/Maps/LMoSysXRExample.umap.

## **15.7 Switchboard listener reports: Has been inactive for more than 5.0s - closing connection.**

This could be due to non standard (ASCII) symbols in project path. Try with simple, English alphabet only directory.

## **15.8 Persistent Float RGBA format in project settings (unable to change)**

To come back to 10 bit or 8 bit format, go to <project dir> > Config and edit DefaultEngine.ini. Change the r.DefaultBackBufferPixelFormat=4 to 3 and reopen Unreal Editor. This should be now settable to desired option in Project Settings.

## **15.9 Feathering on set extension on Mo-Sys Viewport is of very low quality (banding problem)**

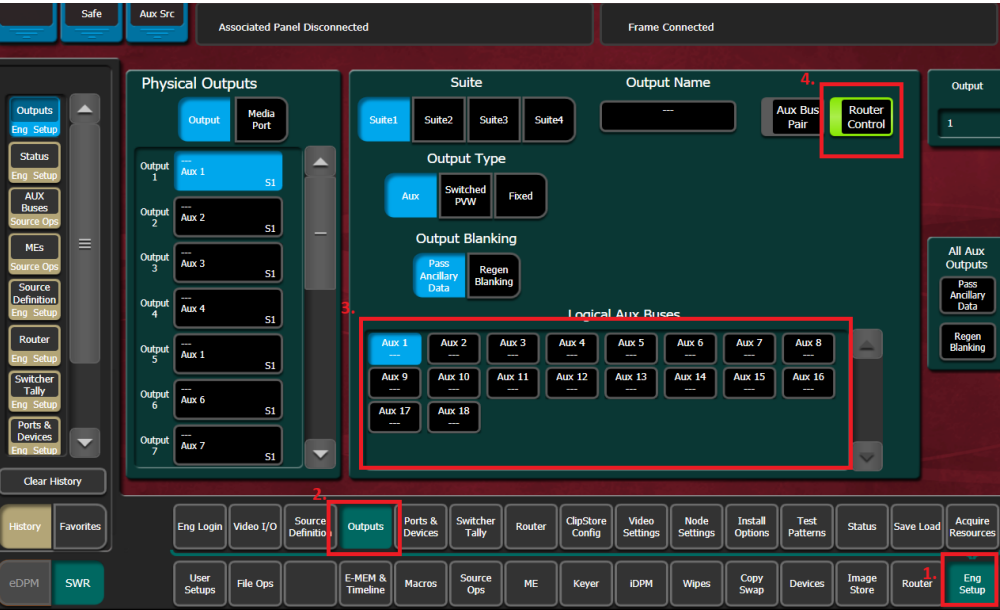
Apply OCIO in Video Controller output.

## **15.10 Grass Valley Switcher**

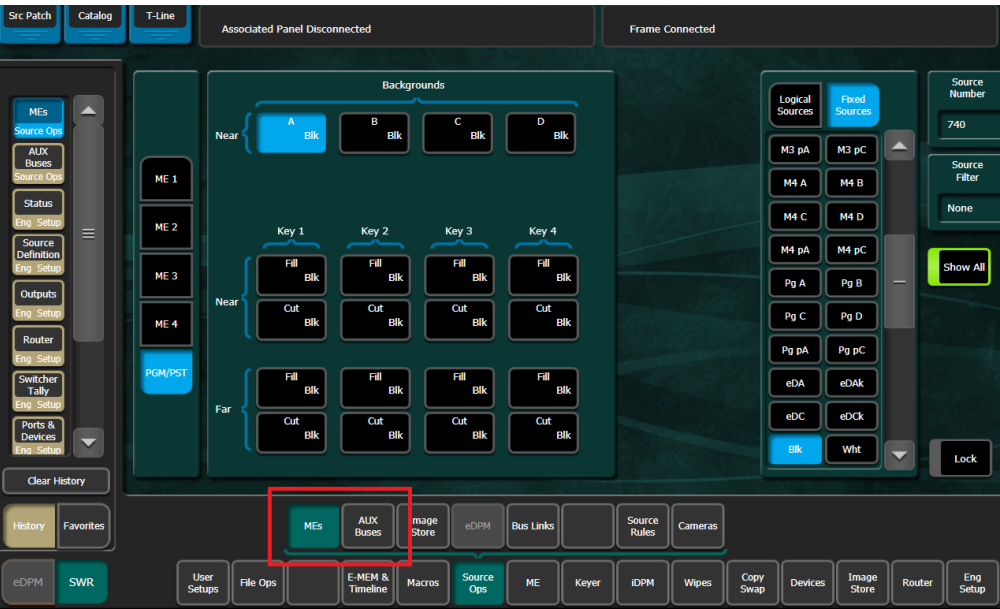
The current implementation makes use of the switcher's Aux buses for switching. As it stands, it uses Aux Bus 1 as Program Input, Aux Bus 2 as Preview Input and Mix Effect for all other Aux



buses. This requires some setup in the switcher itself to allow this. First, we need to allow Router Control of the Logical Aux Buses. To do so, in the Grass Valley Menu, go to Eng Setup > Outputs and select Aux 1. Now, click on Router Control on the top right section of the window, to highlight it. Do this for Aux 2 and all other Logical Aux Buses you intend to control.



The next step requires you to do the mapping of these logical buses and the input/mix effect cutting. This can be done in the Menu application as well, under Source Ops > MEs and Source Ops > AUX Buses.



### 15.11 nDisplay instance getting stuck on black after launch from Switchboard

1. Make sure the correct level is selected in Switchboard, or if using multiple levels, that all the levels in Levels tab are set to Change Streaming Method -> Always loaded.



2. This could be related to the sync policy selected. Try changing the sync policy to None (it can be either set in the nDisplay configuration file or selected in Switchboard settings in Render sync policy, which overrides the nDisplay configuration file). If successful, consider recreating the nDisplay configuration file and try the desired policy again. We recommend using Nvidia (2) as described in .