PRIME VSAR 2.2.0

User Guide

September 2025

SINCE DAY ONE.

Chyron PRIME VSAR User Guide • 2.2.0 • September 2025 • This document is distributed by Chyron in online (electronic) form only, and is not available for purchase in printed form.

Table of Contents

About this Document

This document describes how to use features specific to VSAR.

For documentation pertaining to the installation and configuration of VSAR, please refer to VSAR 2.2.0 Setup Guide

For general documentation about the Unreal Engine, please refer to:
https://docs.unrealengine.com/

Help and support

For contact information or our online helpdesk, please visit our **Support page**.

Disclaimer

Our products are subject to continual development and improvement. Therefore, while the information in this document was complete and accurate when it was written, additions or modifications to the products may cause changes to the technical and functional specifications. No rights can be derived from this document.

Third-Party Designers - Licensing

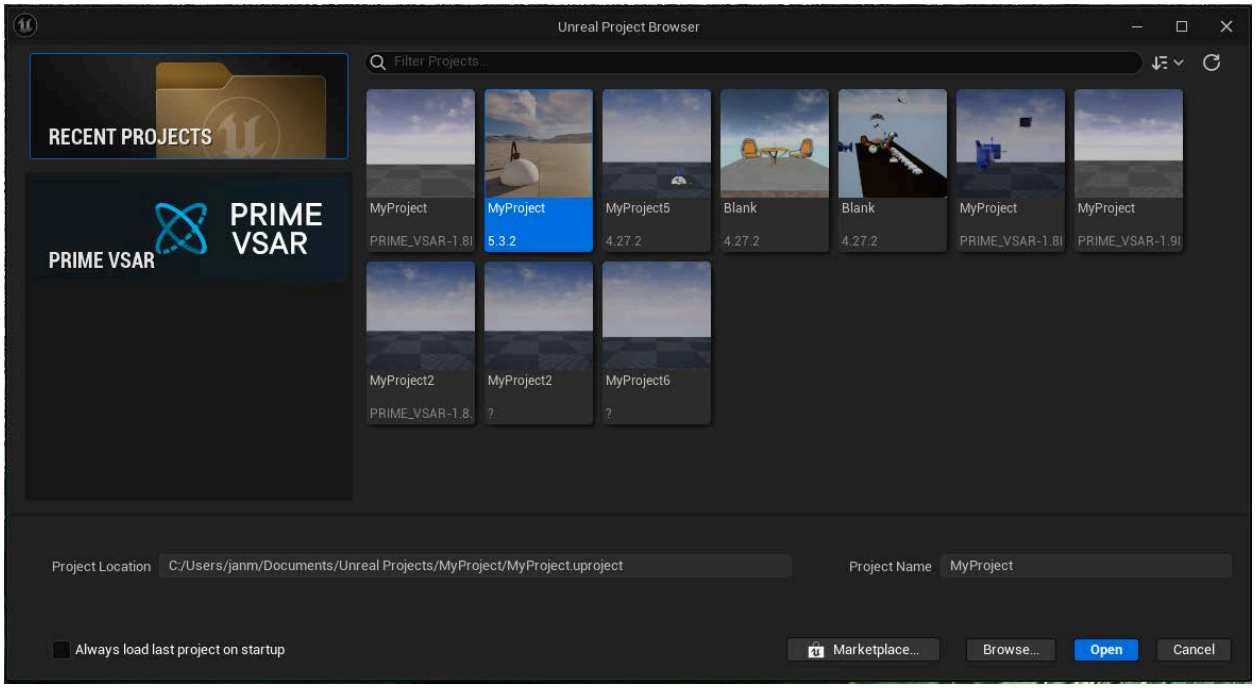
The Primitive Library is part of the PRIME VSAR plugin, but it is licensed independently of PRIME VSAR.

In order for VSAR Primitives to work, you need to have a license for PRIME VSAR and a license for VSAR Primitives.

My First VSAR Blank Project

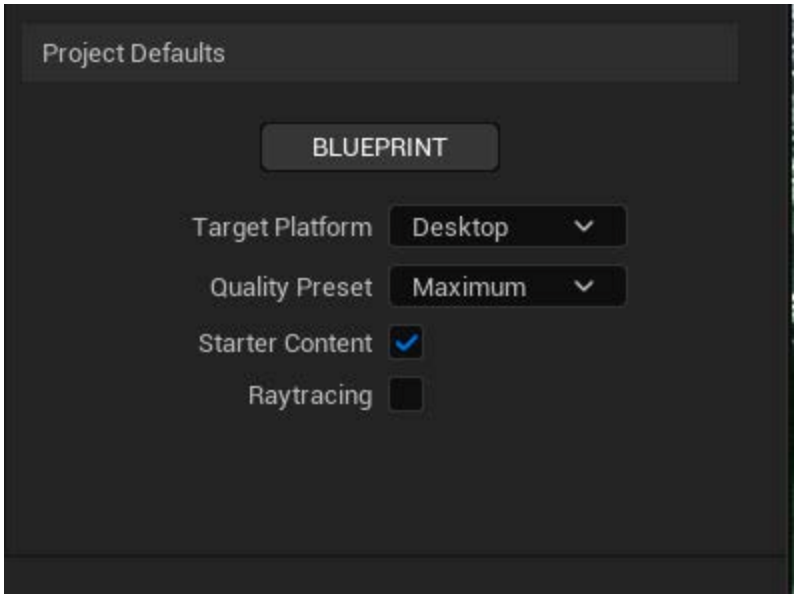
Create New Project

1. Start PRIME VSAR.
2. Create a new project by selecting the PRIME VSAR example project and press Next.



3. PRIME VSAR comes with a couple of Virtual Studio samples originating from ***Unreal*** marketplace. Select the Blank project and press Next.

4. As for any ***Unreal*** projects, settings may be set up before creating it, but may be later changed. It is recommended to have Maximum Quality and Desktop/Console by default. Specify the Project path where the related files will be saved, its name and then press Create Project.

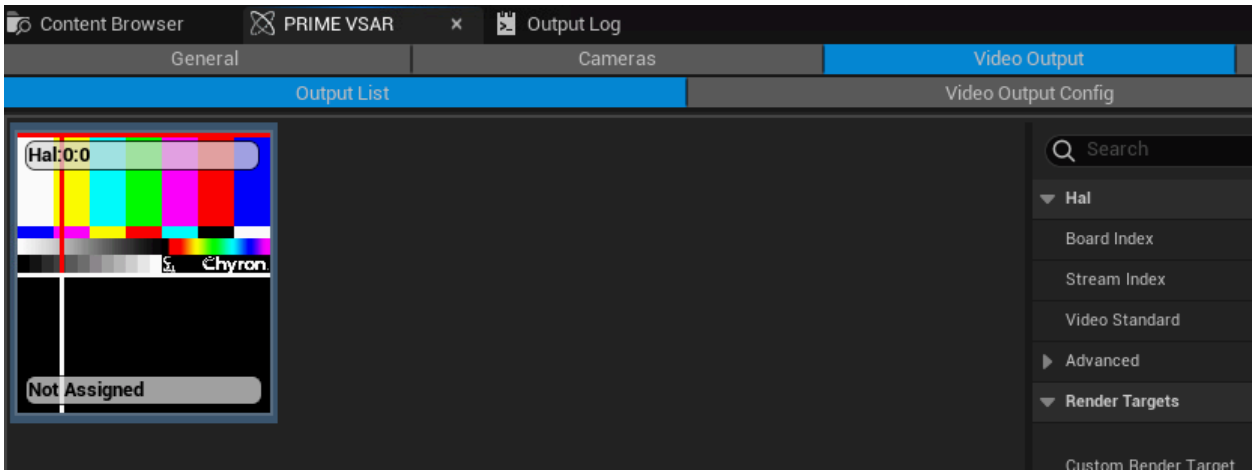


PRIME VSAR will automatically restart and load the new Project.

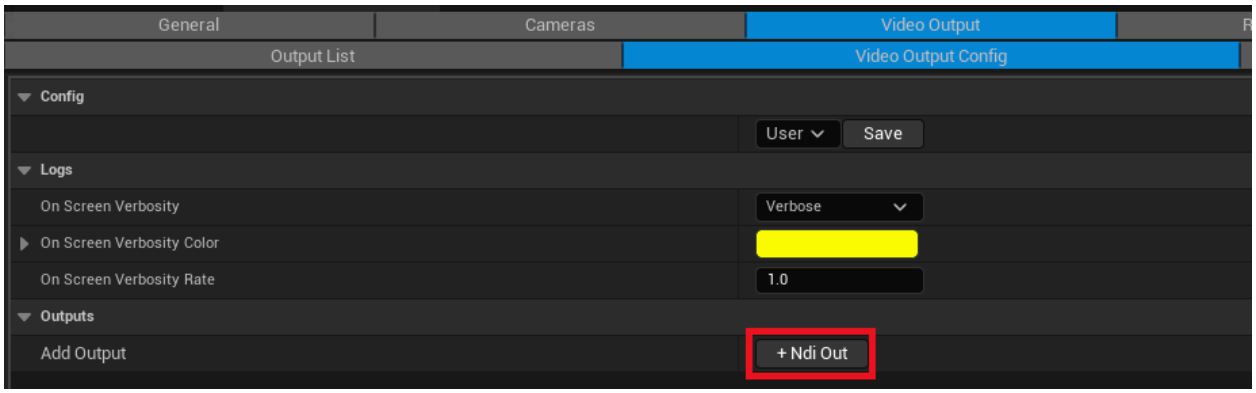
The Project will load an empty Level (or Map) in the main window. It is possible to create other Levels and save them as Assets than can be found in the Content Browser. Within the Content Browser you can organize your content by creating folders.

- Create a 'Levels' folder and save the empty Level in it.

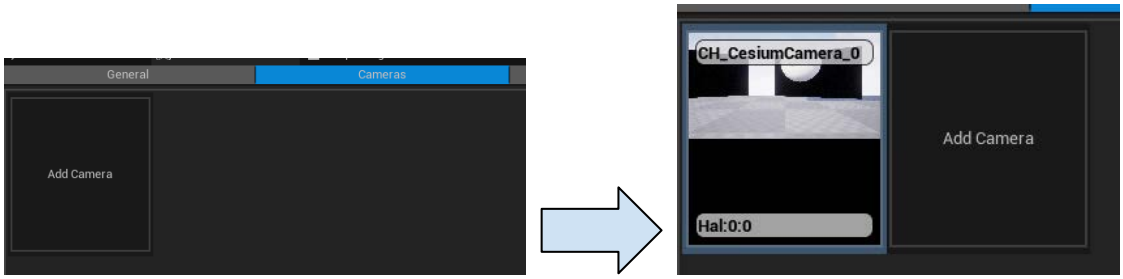
If the system has a Matrox Card, by default there should be an existing Video Output using the first playout port of the card (VSAR Config > Video Output > Outputs).

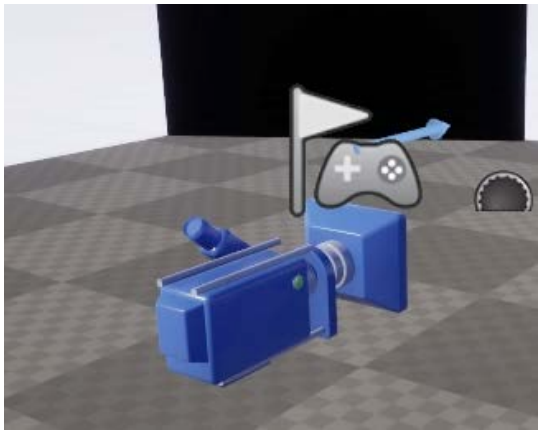


- If not, create a Video Output by adding a NDI output (VSAR Config > Video Output > Output Configuration).

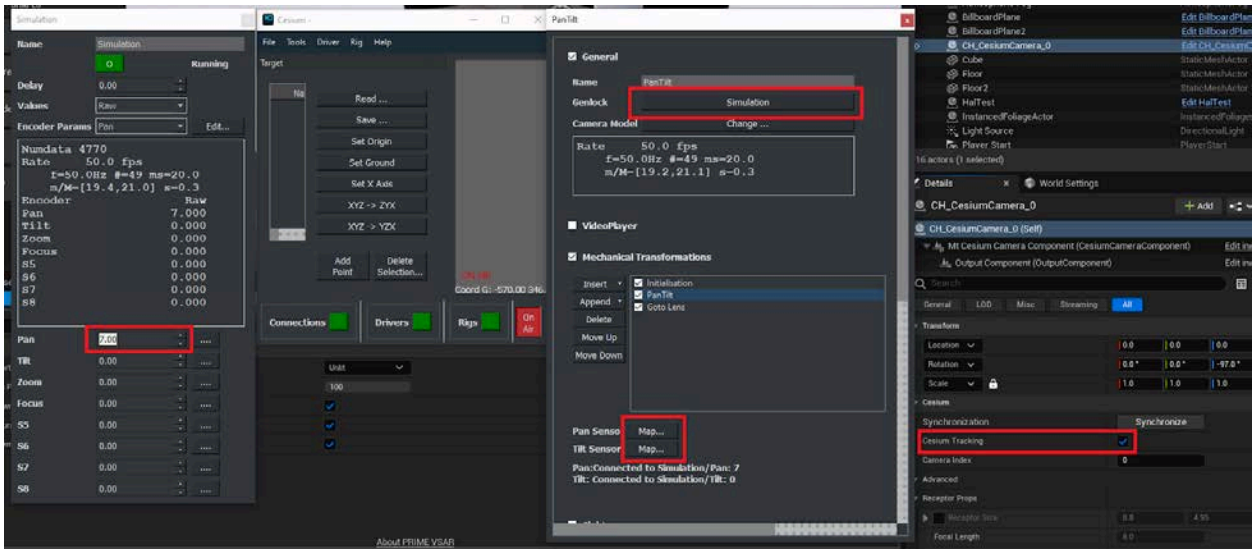


- Add a MT Cesium Camera (VSAR Config > Cameras).

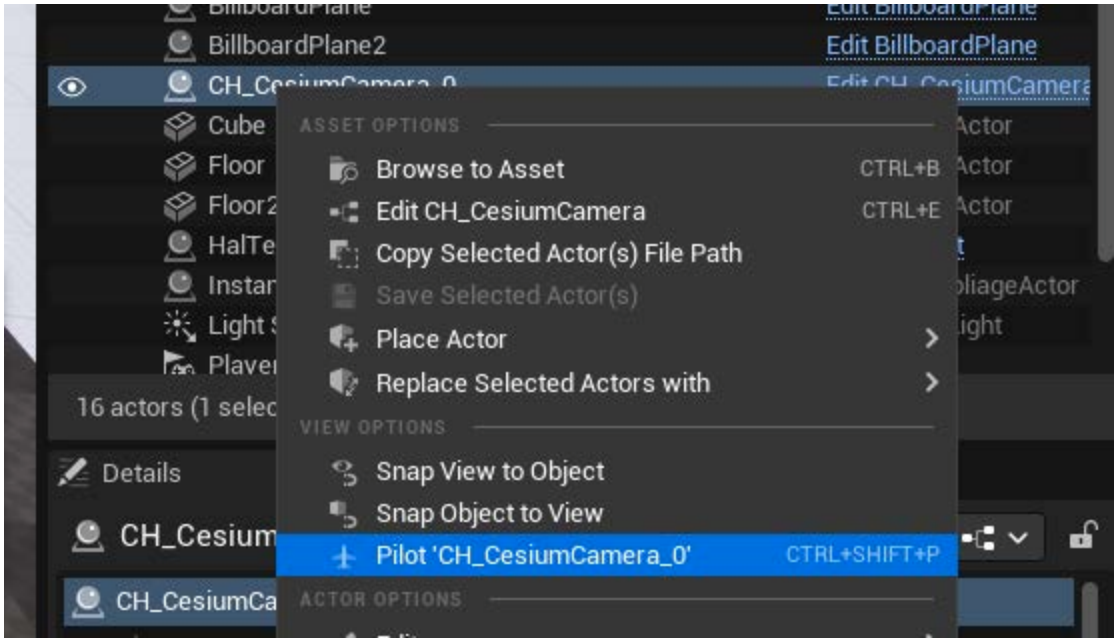




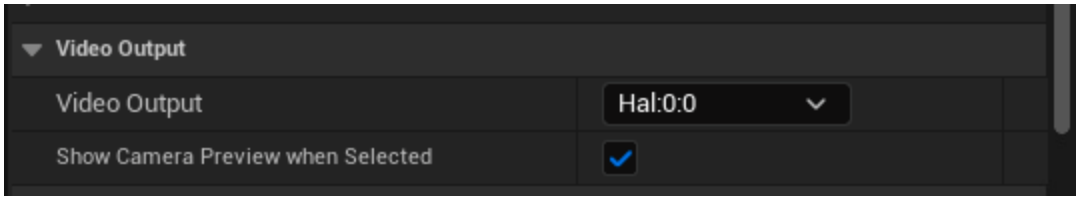
Usually the Camera is controlled by Cesium, you can test it with Cesium and the 'Simulation' driver and the PanTilt rig (mapping the PanTilt in the mechanical Transform to Pan/Tilt) and enabling the Cesium Tracking on Cesium Camera, by changing the Pan value or others.



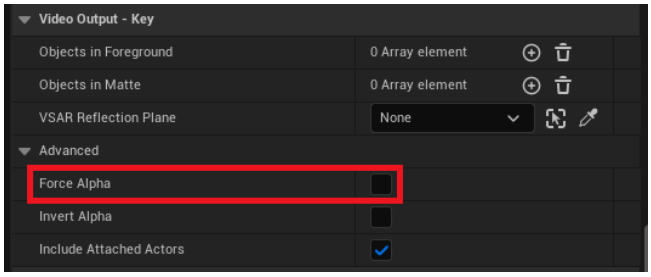
Tip: to easily manipulate the Camera you can use the Pilot option as follows:

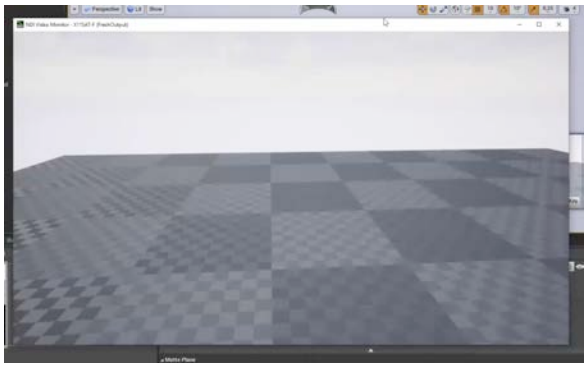


The camera output is as well connected to the first output port of the Matrox Card through HAL:

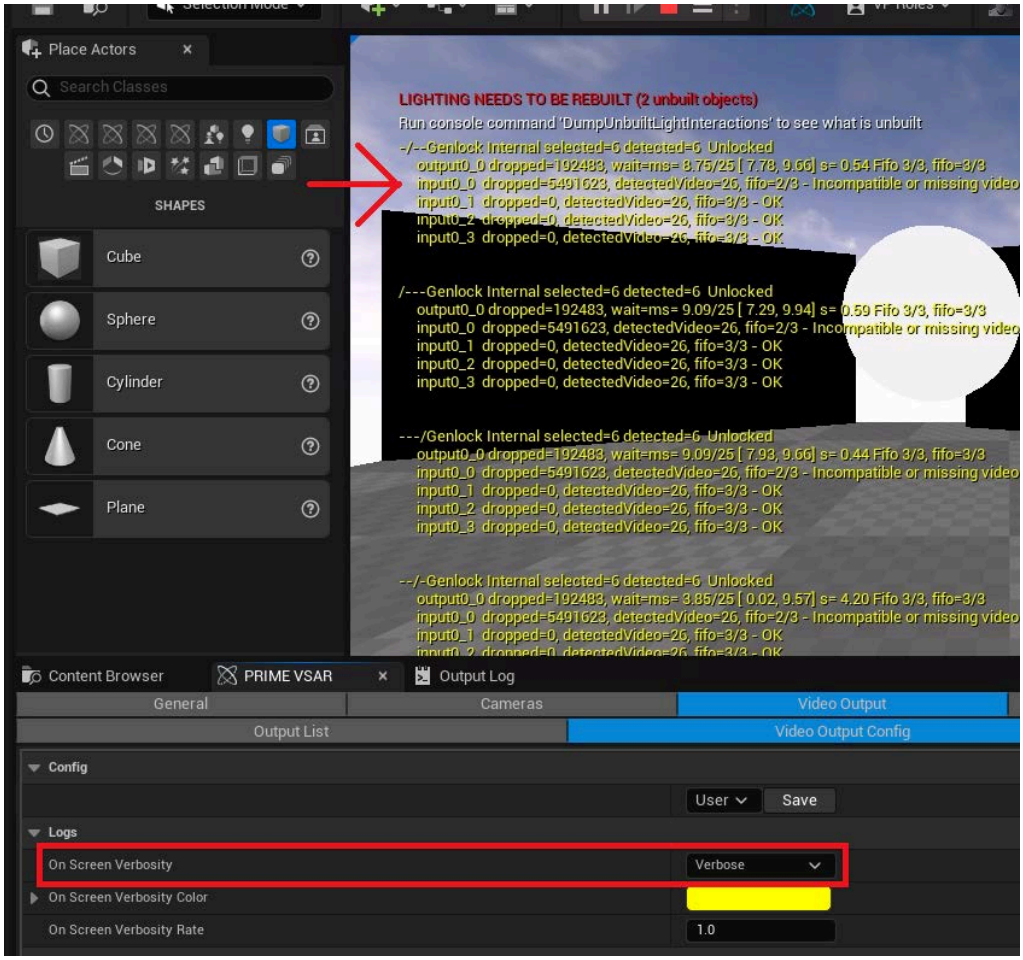


- If there is no Matrox card present in your rig, change this setting to NDI, enable the 'Force alpha' checkbox (in the Camera properties under Video Output - Key) and open the NDI Video Monitor application to get the Camera Video output.

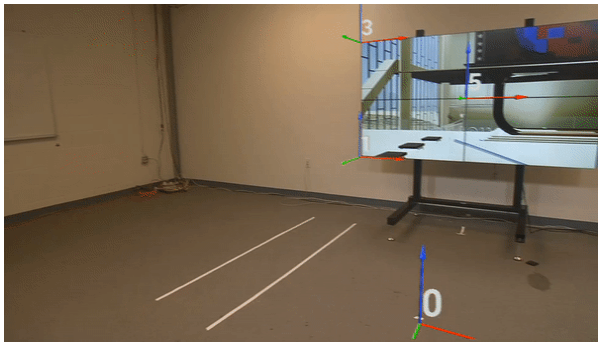


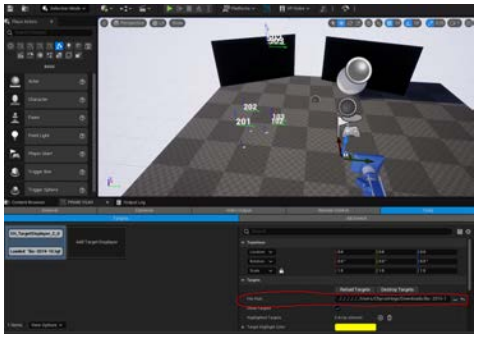


- The Matrox card status can be displayed as real-time logs on the main window. To enable/disable or change the logs level, use the 'On Screen Verbosity' (Video Output > Video Output Configuration).



If you plan to work in Tracking mode, or facilitate the work of designers by setting up special references of the real studio, you can load a tracking file in Tools > Targets:

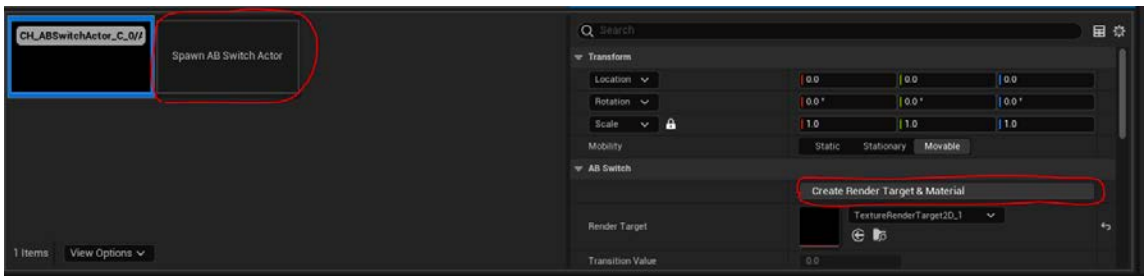




Adding a Virtual Screen controlled by an AB Switch

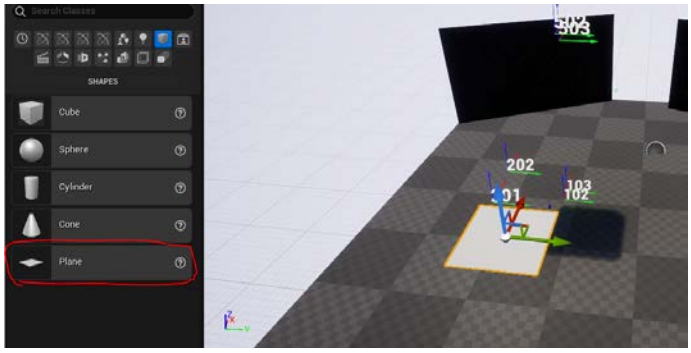
The AB Switch is used to provide video transitions within virtual screens or between Scenes.

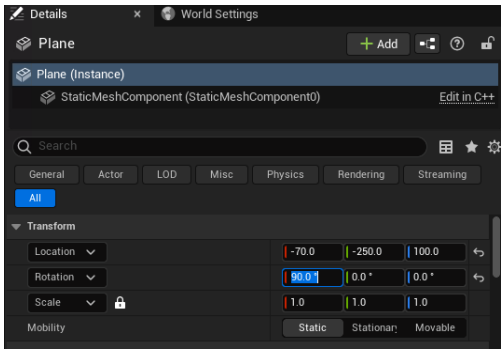
- Start by creating an AB Switch (VSAR Config > Tools > AB Switch), select the created AB Switch and click the 'Create Render Target & Material' button. This will create a new Material object in the Content Browser.



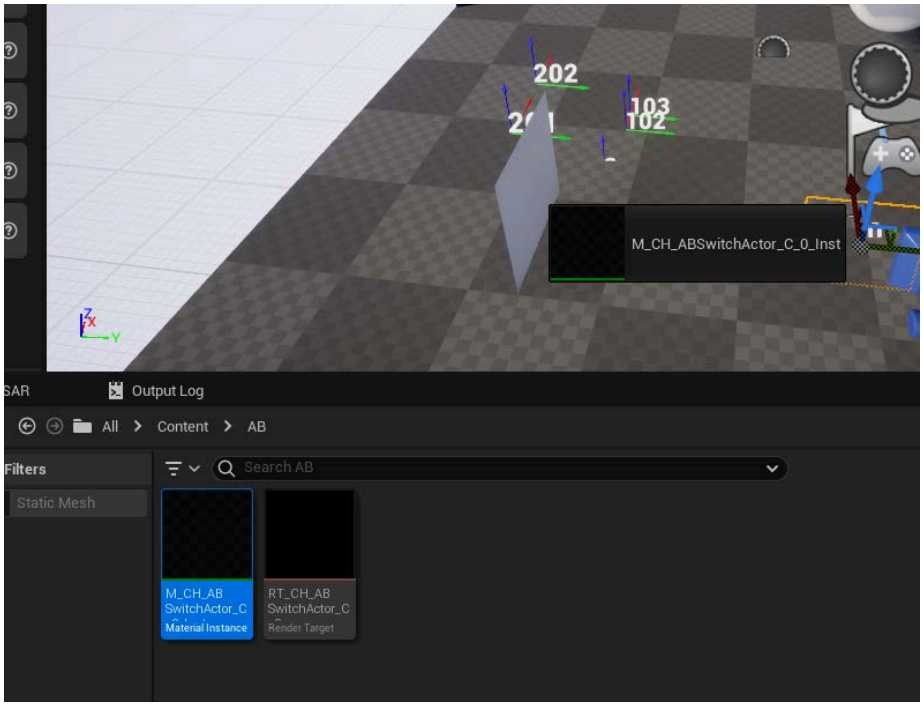
- Add a Plan Actor from the left side menu. Then change the Plan position from the right menu to be vertical (Transform > Rotation > X=90), upper on the set (Location settings), its size to be 16/9 (Scale settings), etc.

Tip: press the F shortcut to frame the object.

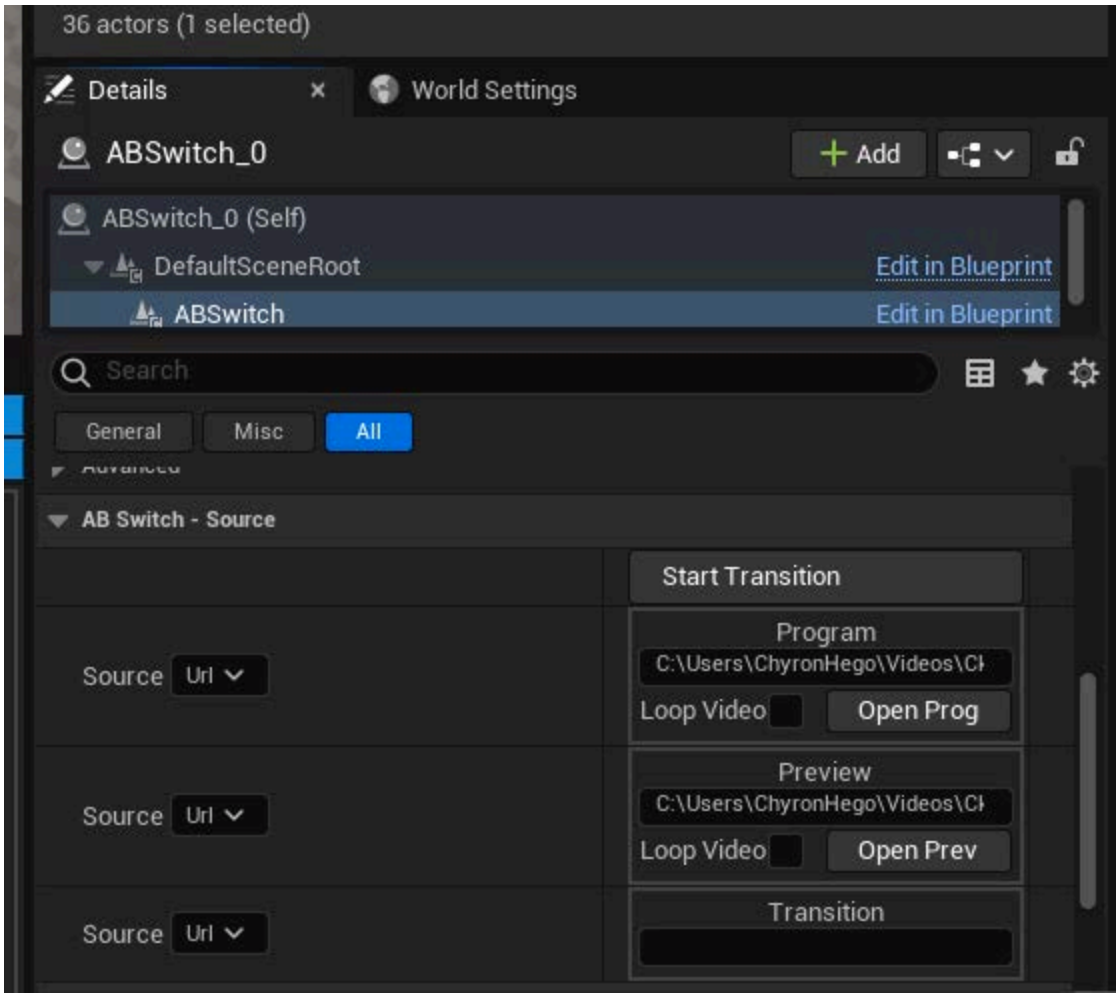




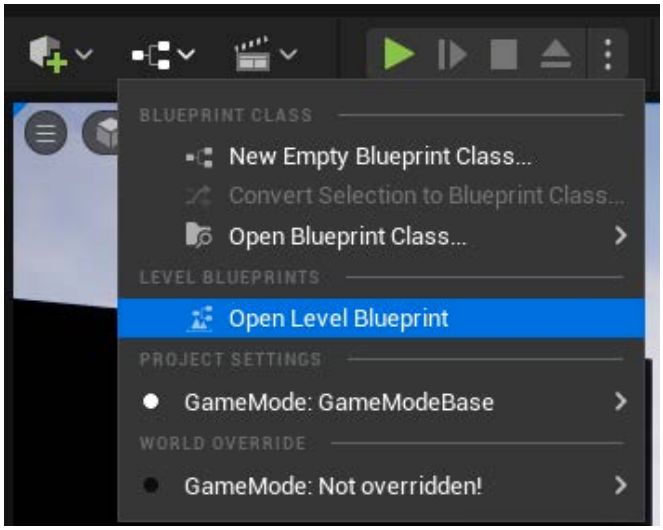
- At this stage this Plan is just an opaque object to interconnect the Plan and the AB Switch and change the Plan surface, select the previously created Material Object (Content Browser > ABSwitch) and drop it onto the plan.



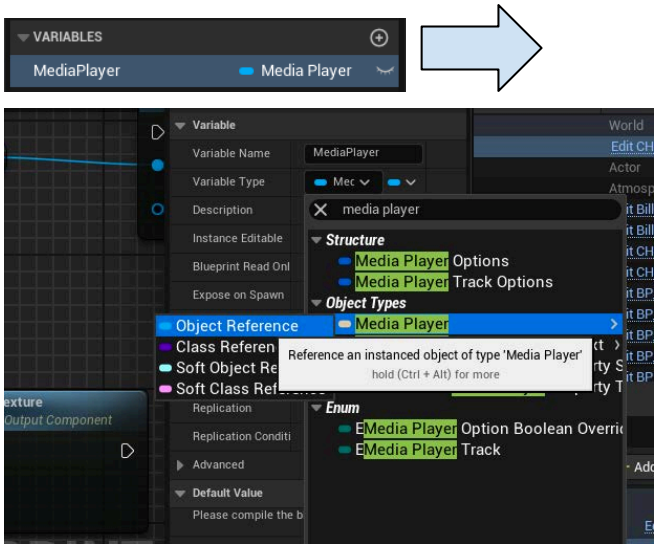
- To test it use the Effects and Simulate settings of the switch.



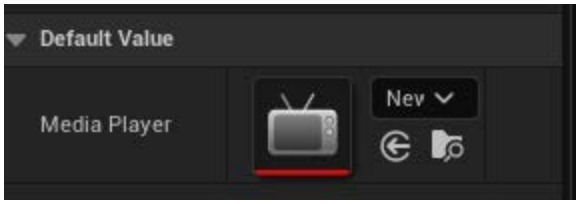
- To autoplay media files or Matrox video inputs with the AB Switch, open the Level Blueprint.



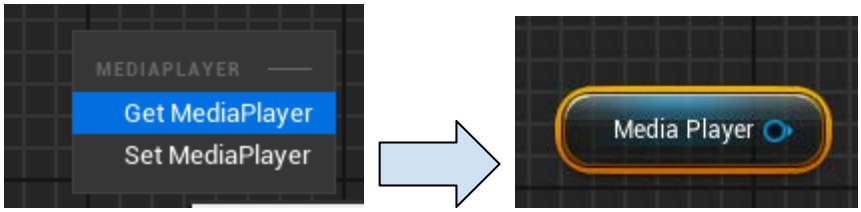
- Create a new Variable with a name such as 'MediaPlayer', of Type 'Media Player' and subType 'Object Reference' to allow to drag and drop it.



- Compile and add the target Media Player object as value of the variable.

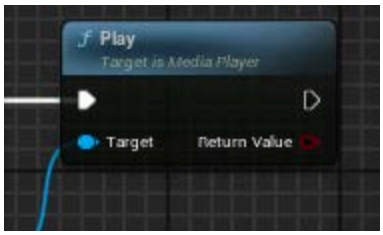


- Drop the Variable onto the Event Graph window and select the displayed option 'Get MediaPlayer'.



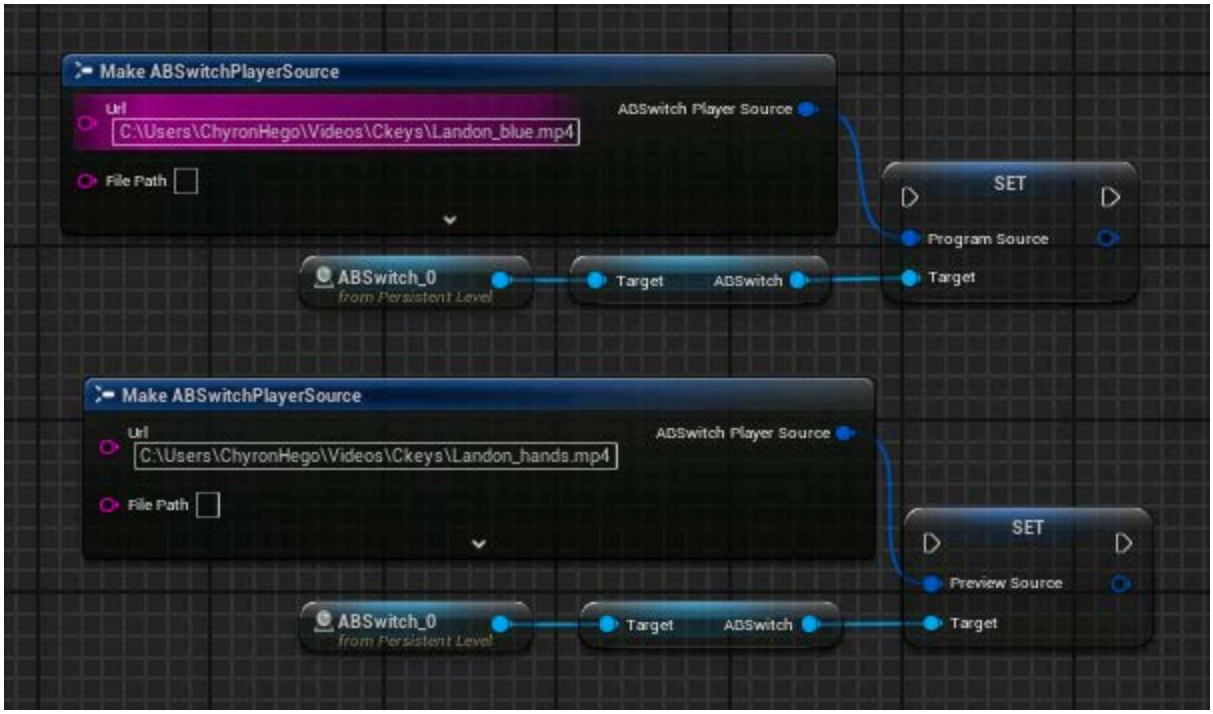
- Add a 'Open Url' element and fill the Url setting with the source media file or stream.

- Add a Play element.



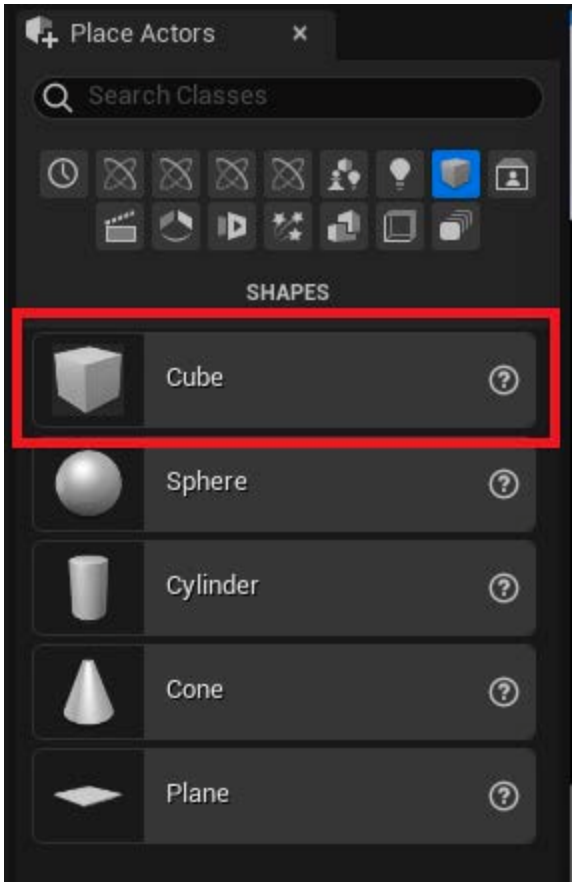
- Link the elements as follows and press Compile.

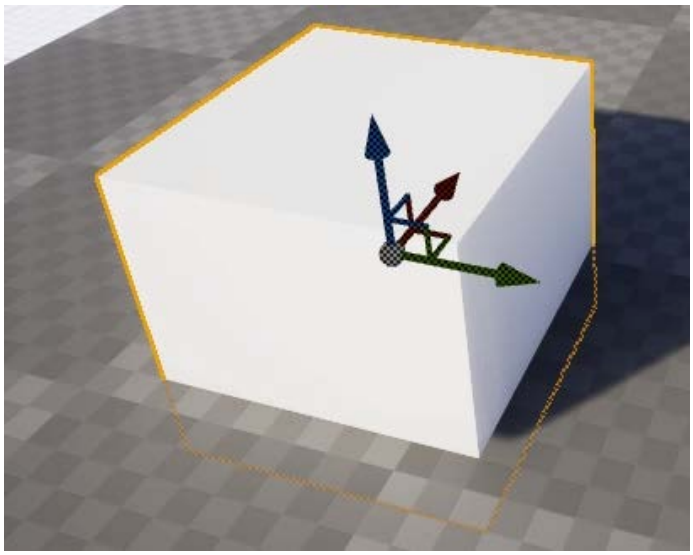
- we can also set the sources

- 

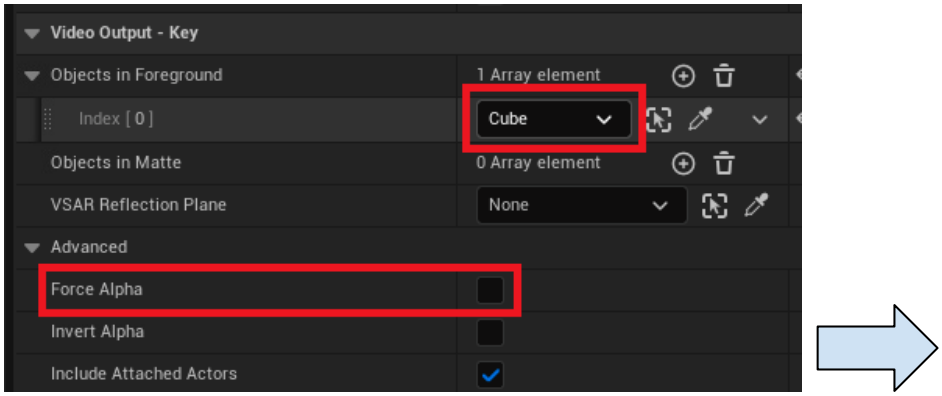
Reflections and Shadows

- Add a Cube (Using the Place Actors Window) to the main Window.

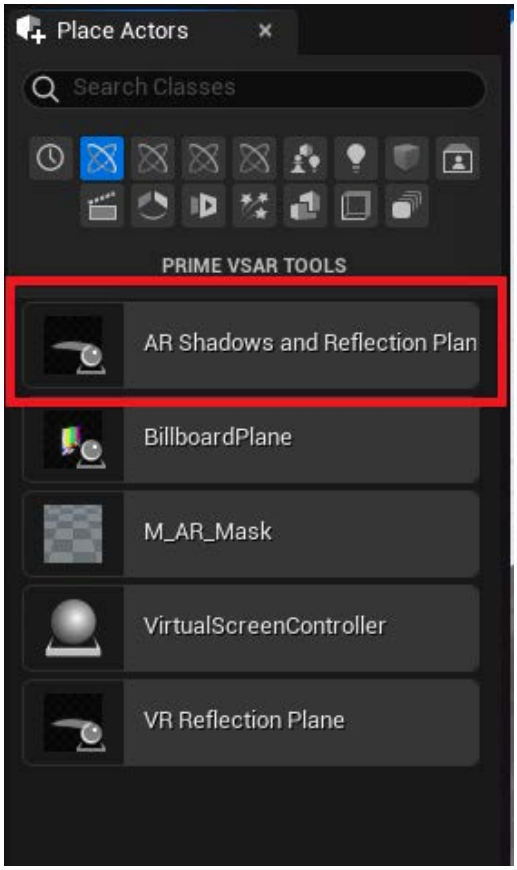




- In the Camera Settings, deactivate the Force Alpha option and set the Cube as a Foreground Object:

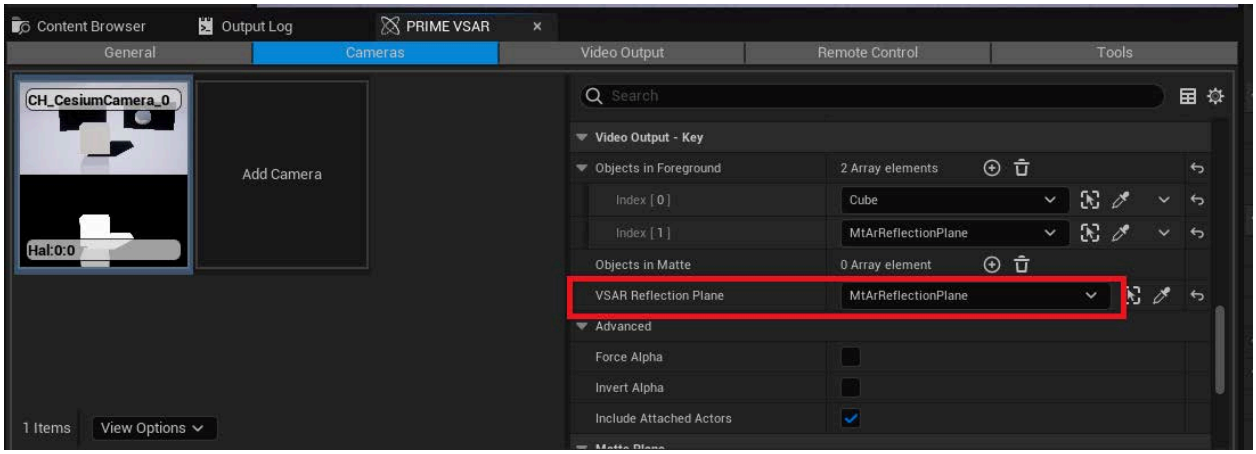


- Add a “AR Shadows and Reflection Plane” Actor to substitute the default to one allowing shadows and reflections in Alpha and place it to roughly cover the existing one:



- Hide the default Floor by deselecting the Visible checkbox.

- Go to the Camera settings and set the ARPlane Object setting to the created Mt ARPlane object. This should correctly enable the display of the reflexion and shadow to the Camera:



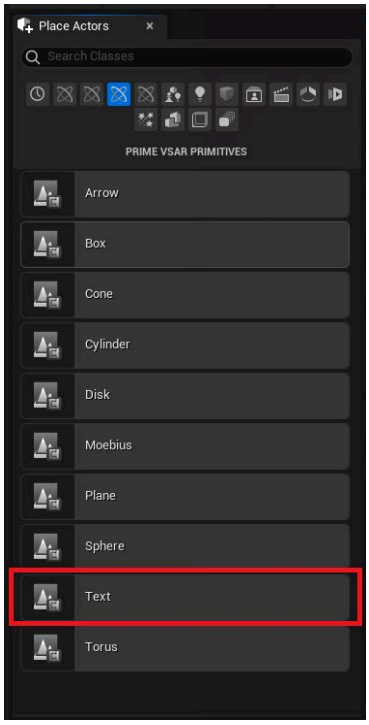
- To add the reflection and shadow of the screen panel, add it in the Camera's Object in Foreground list.

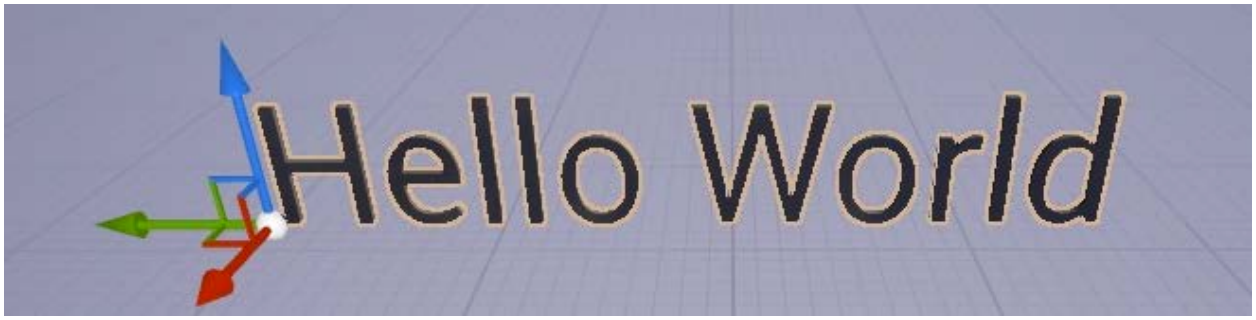
TIP1: to change the light origin, change the Rotation Z value of the Light Source object.

TIP2: the Reflection and Shadow opacity settings are under the MtARPlane settings.

Text

- Add a Mt TextPrimitive object to the main window.

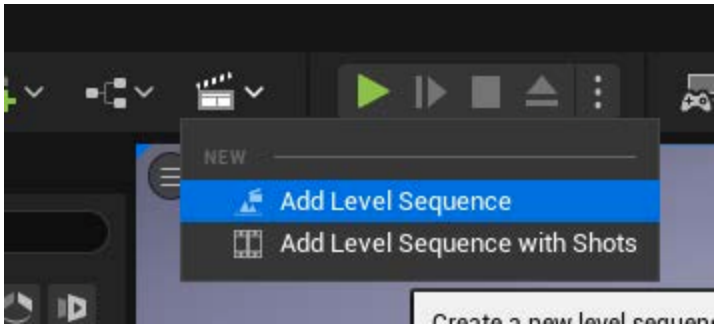




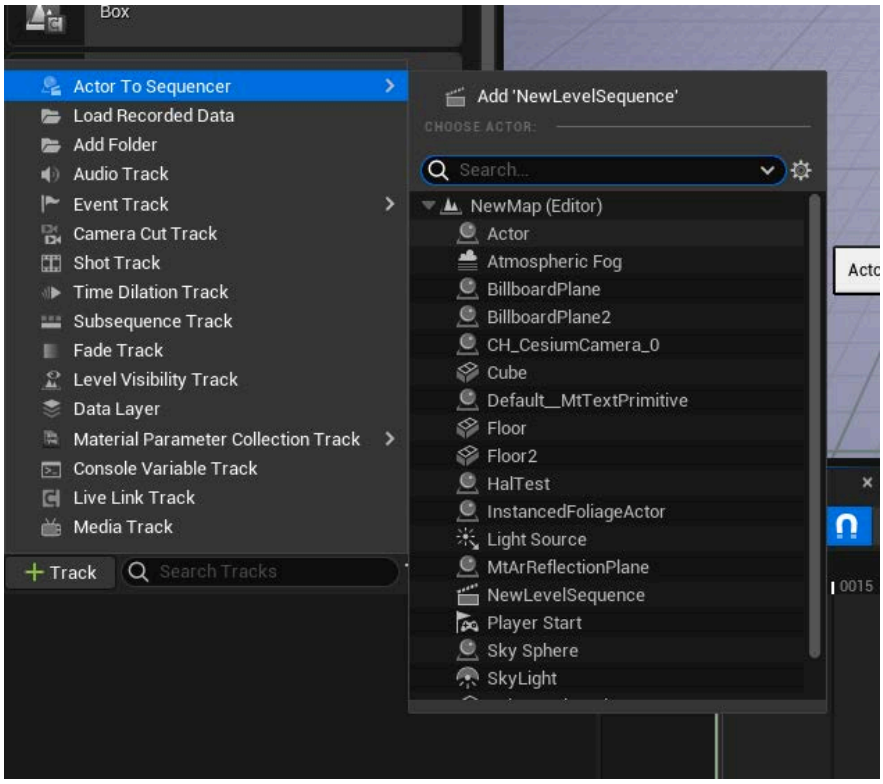
Animations

In this scenario we just want our object to be moved between 2 Keyframes.

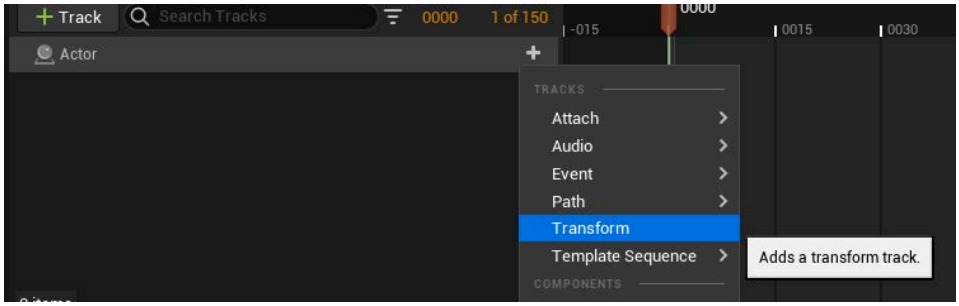
- A simple way to add Animations (or Actions in Prime) is to launch from the top menu Cinematics > Add Level Sequence.



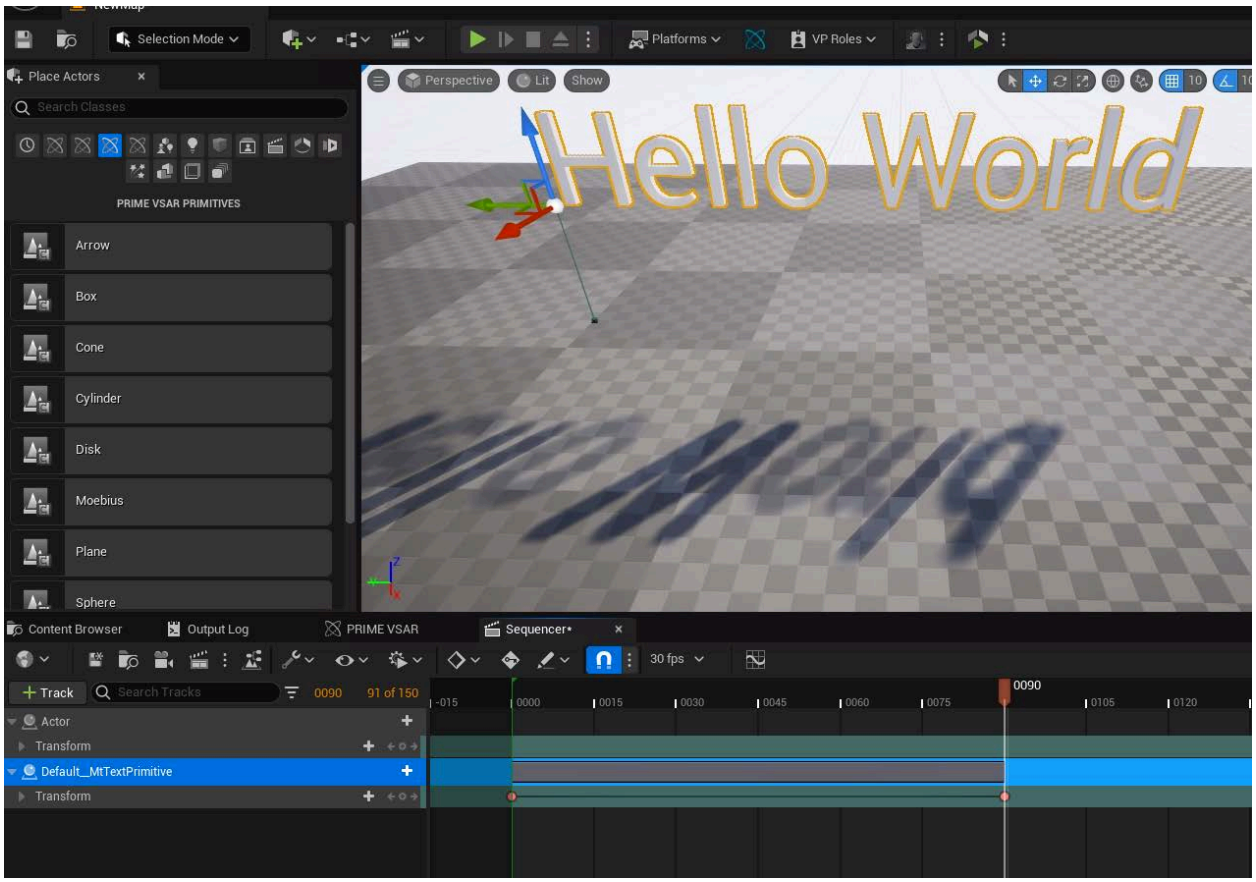
- Name the new Sequence and save it in the Content Browser in a folder or your choice (e.g. Sequences). This will open a Timeline window similar to that in PRIME.
- Click + Track and add an Actor (Actor to Sequencer > Actor).



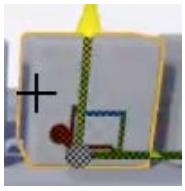
- On the Actor line, click +Track and select Transform.

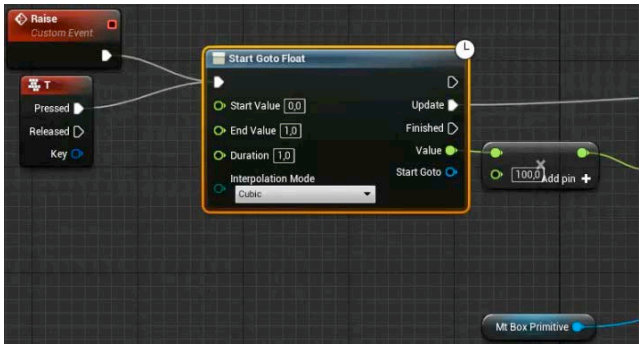


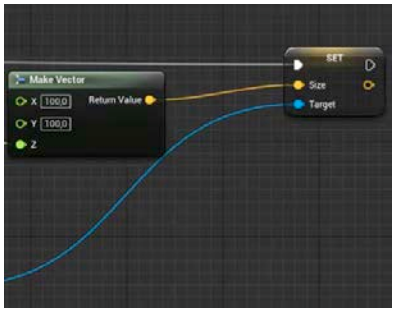
From there the same way as in PRIME, you can create Keyframes and change the XYZ values at each Keyframe.



NOTE: Animations can also be created using Blueprint functions. Example here with a growing Box Primitive triggered by a T shortcut:




Raise
Custom Event
T
Pressed
Released
Key
Start Goto Float
Start Value 0,0
End Value 1,0
Duration 1,0
Interpolation Mode
Cubic
Update
Finished
Value
Start Goto
100,0
Add pin
Mt Box Primitive


Make Vector
X 100,0
Y 100,0
Z
Return Value
SET
Size
Target

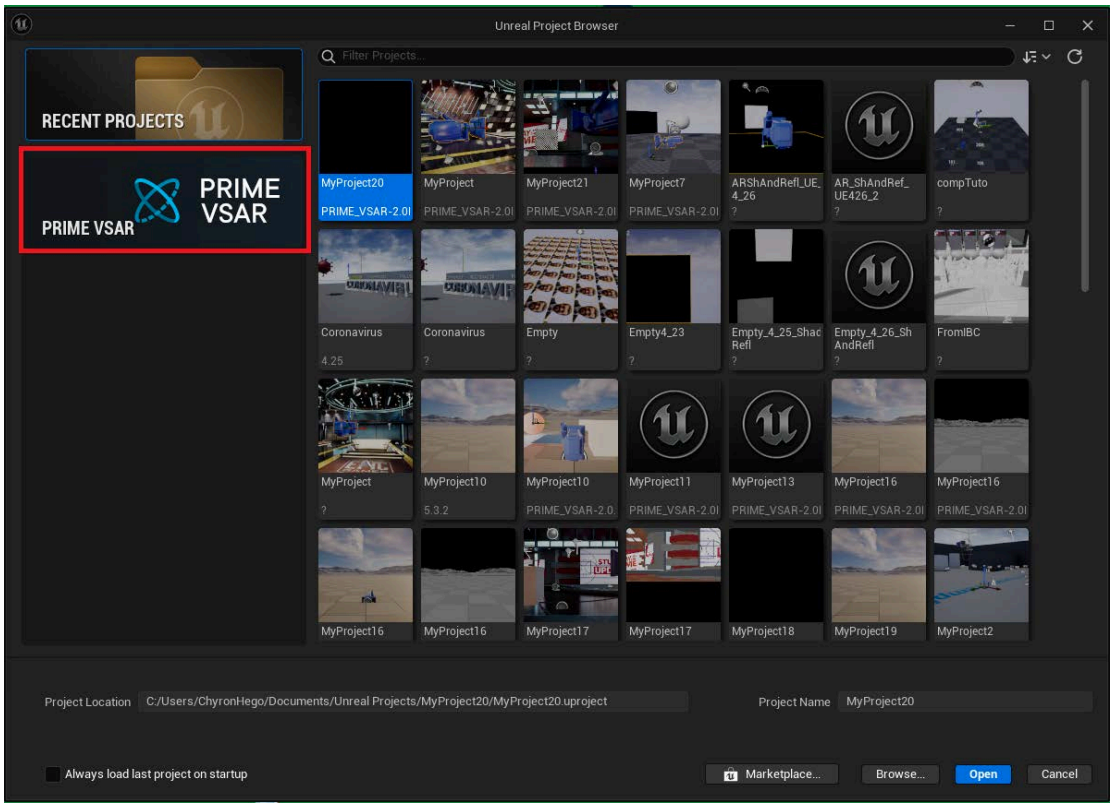
My First VSAR Virtual Studio Project

Overview

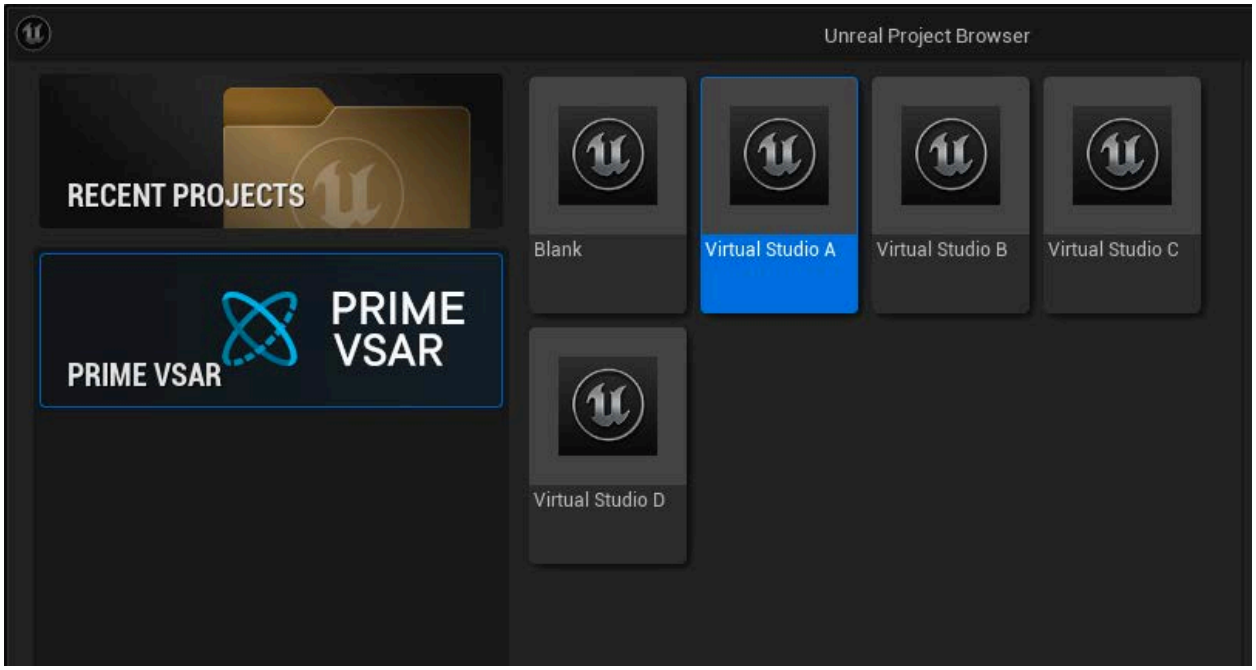
In this section we will use a Virtual Studio template project and insert video signal in it.

Create New Project

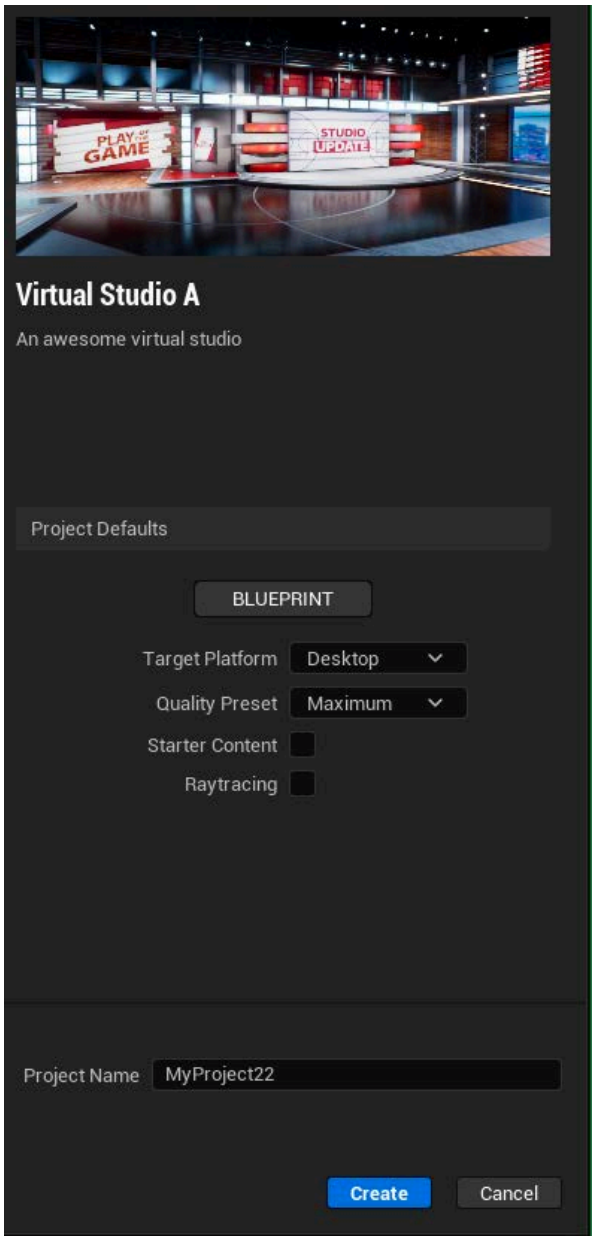
5. Start PRIME VSAR.
6. Create a new project by selecting the PRIME VSAR example project and press Next.



7. PRIME VSAR comes with a couple of Virtual Studio samples provided at Unreal marketplace. Select the one of the Virtual Studios (e.g. Studio A) and press Next.



- As for any Unreal projects, settings may be set up before creating it, but may be later changed. It is recommended to have Maximum Quality and Desktop/Console by default. Specify the Project path where the related files will be saved, its name and then press Create Project.

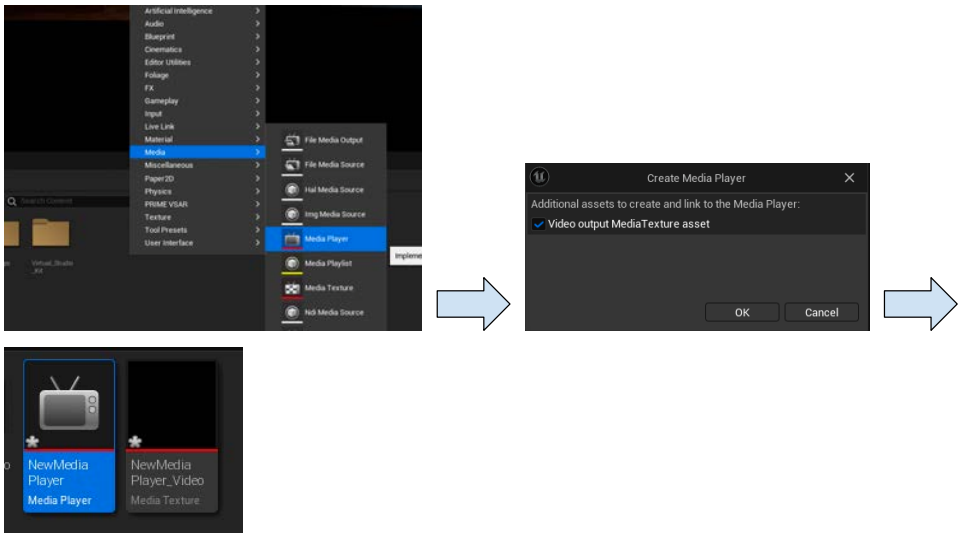


PRIME VSAR will automatically restart and load the new Project.

Create Video Board and File Inputs

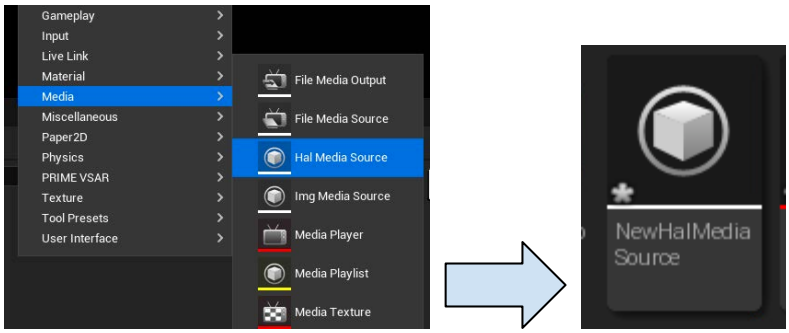
- In the Content Browser, create a new Folder, e.g. 'Media' or Composure (right-click > 'New Folder').
- From that folder, create a new Media Player (right-click > 'Media > Media Player'). A pop-up window will be displayed, check the 'Video output Media Texture asset' checkbox in order to

allow the Media Player to be inserted into Objects within the Virtual Studio. This will create 2 objects, a MediaPlayer and a MediaPlayer_Video.

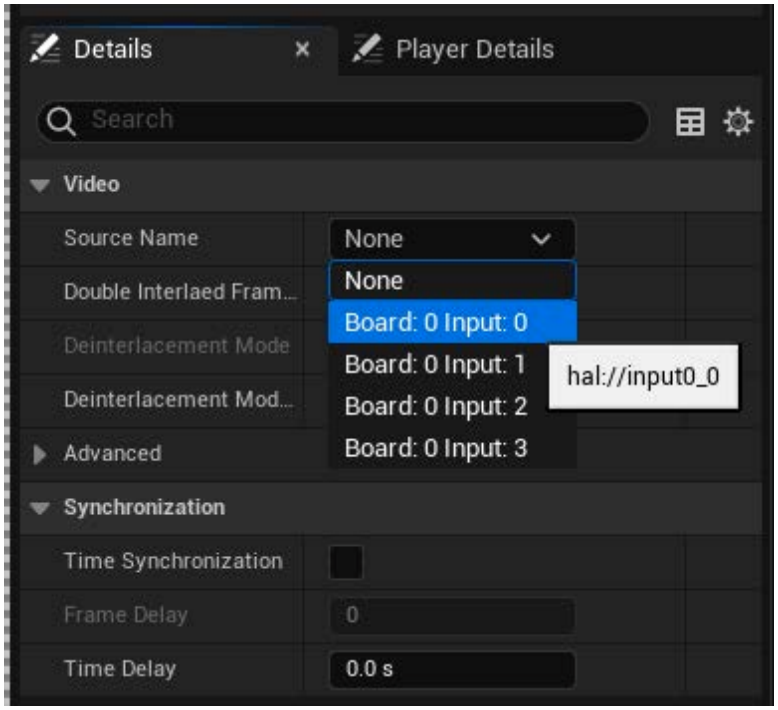


1- In the case an incoming Video Signal onto a Matrox board is available, follow the procedure below, otherwise jump to 2.

- Create a new 'Hal Media Source' (right-click > 'Media > Hal Media Source').

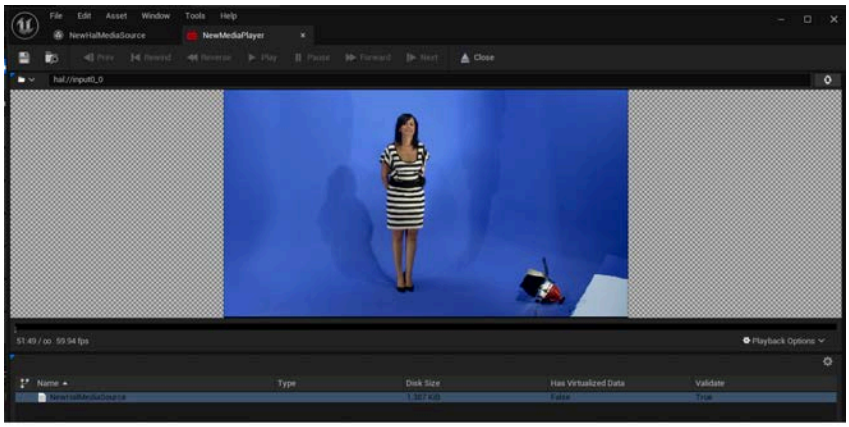


- Double-click the HalMediaSource object and select the input from the dropdown



⚠When no inputs are available, possible causes: Cesium consuming HAL inputs - to solve this in Cesium go to Tools → Preferences... → Video → Enable Matrox and disable it (this will free HAL inputs but Cesium will not see video inputs for calibration, recommended after calibration), **HAL.xml configuration**, license

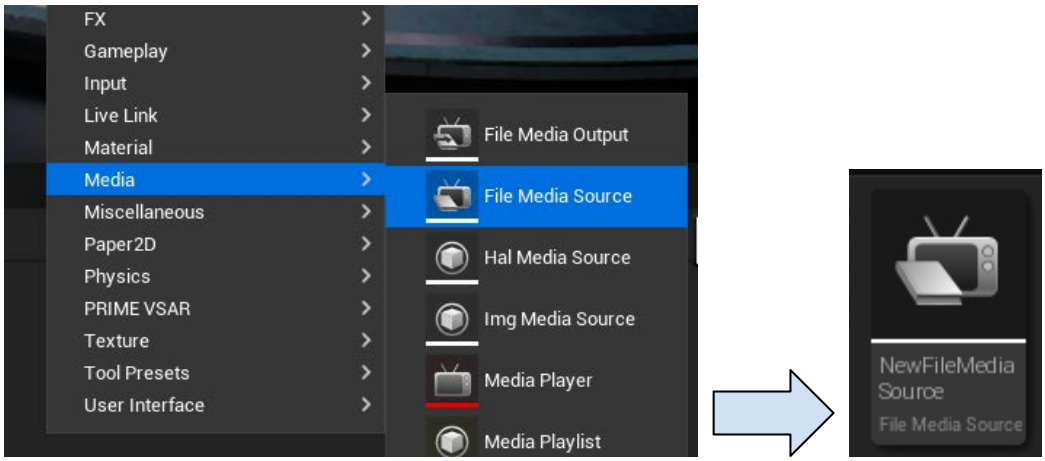
- Double-click the MediaPlayer object, select the HalMediaSource object to preview it. Then Save.



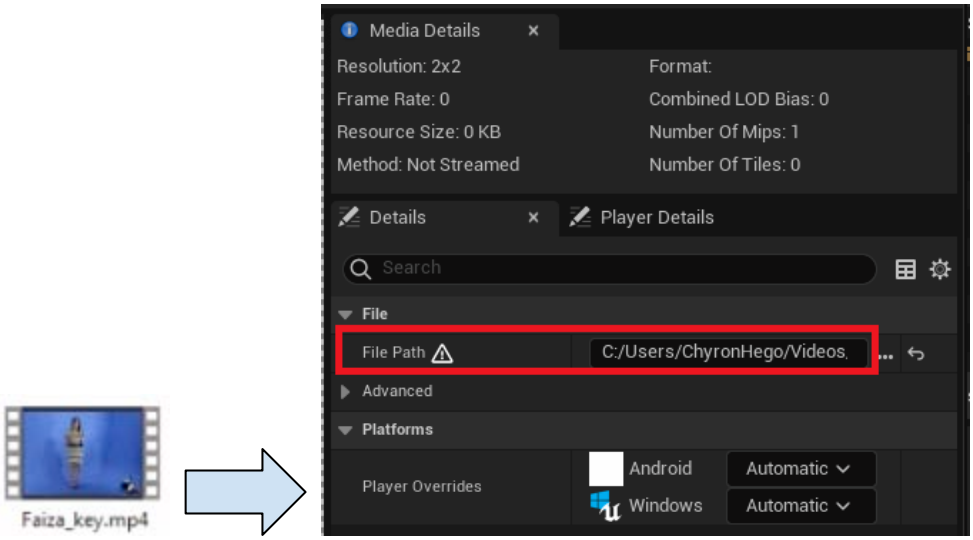
END 1

2 - To use a input Video File within the Virtual Studio:

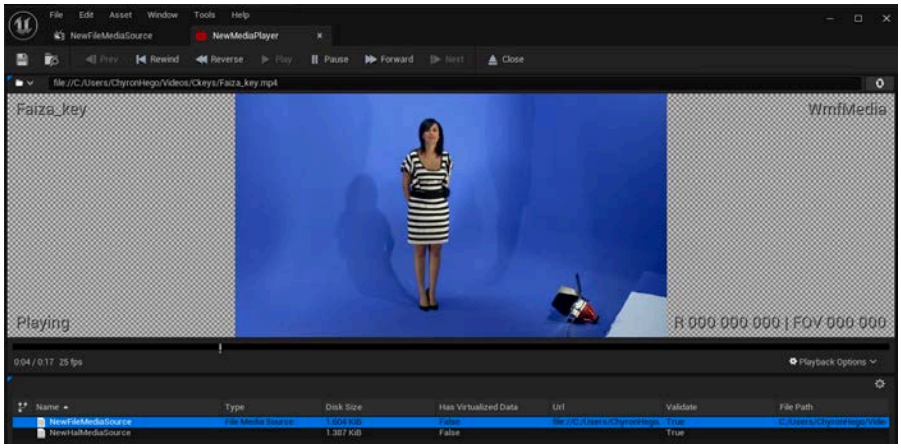
- Create a new ‘File Media Source’ (right-click > ‘Media > File Media Source’).



- Double-click the FileMediaSource object and add a video with a unicolor (blue/green) background and Save.



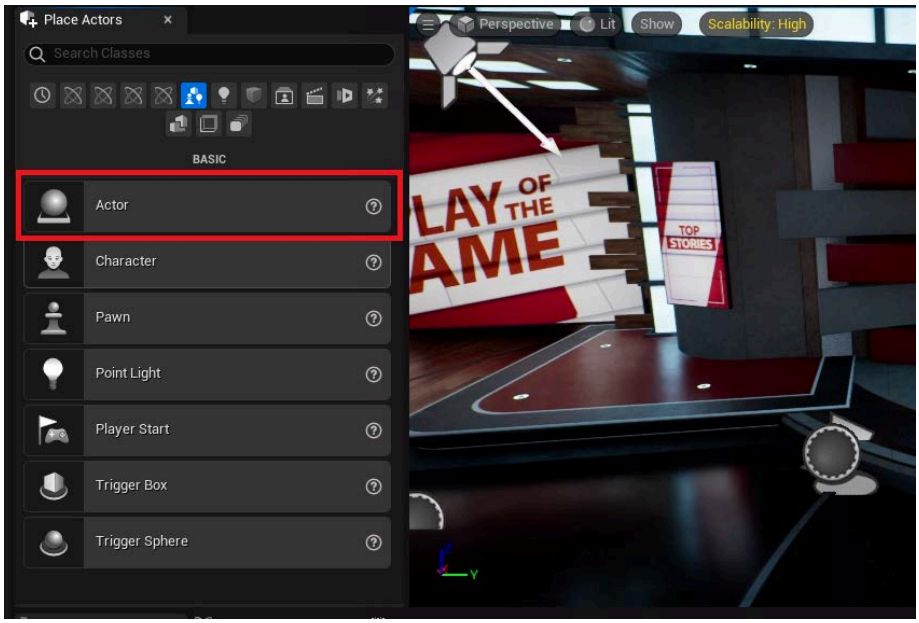
- Double-click the MediaPlayer object, select the FileMediaSource object to preview it. Enable the Loop option and Save.



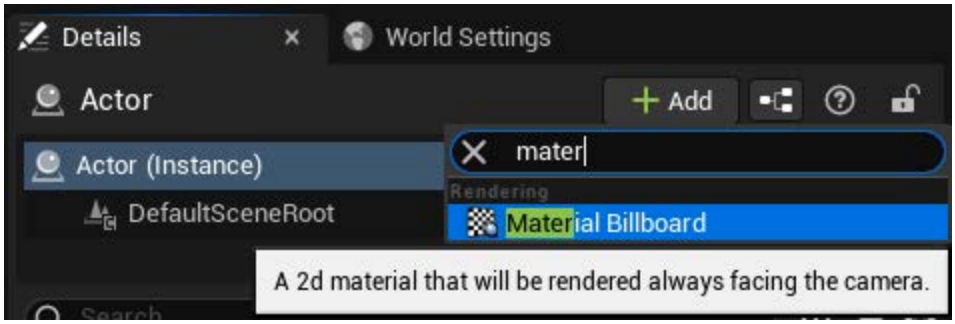
END 2

Create Billboard Actor

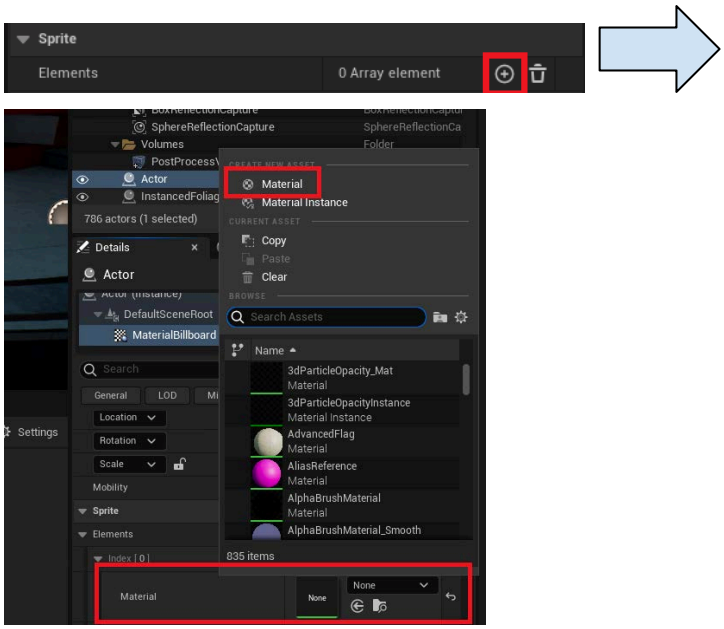
- Add a ‘Empty Actor’ object by dragging and dropping it into the virtual studio.



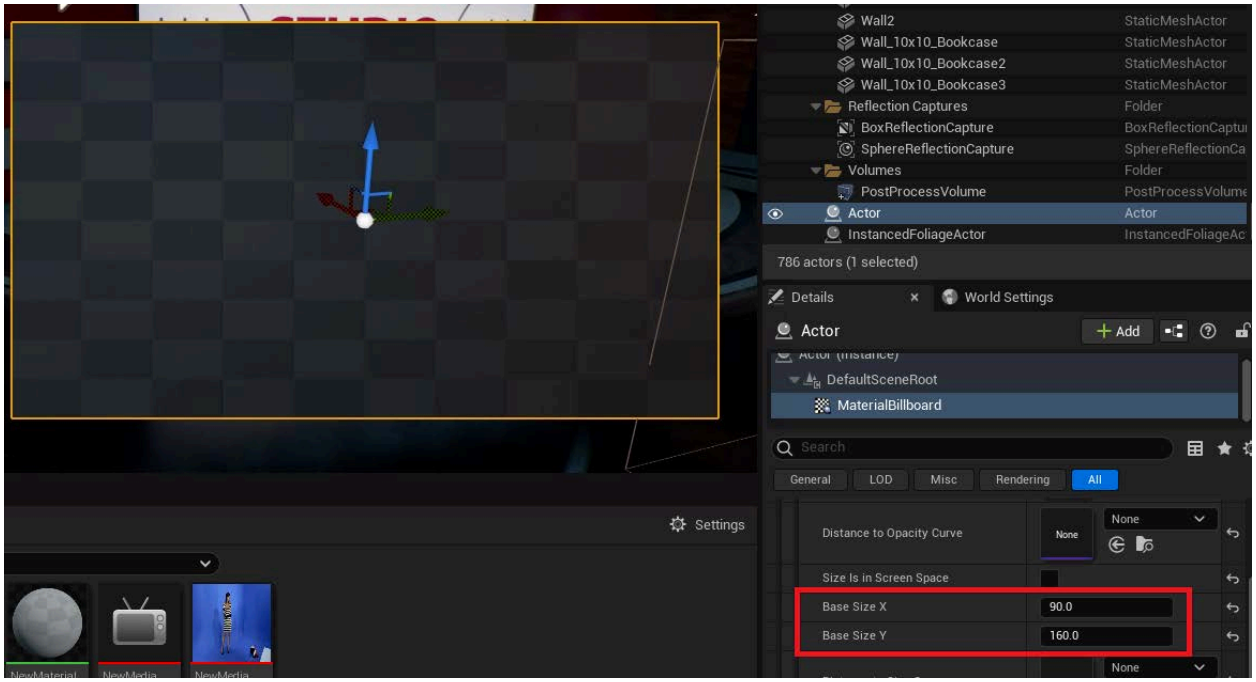
- Add a ‘MaterialBillboard’ Component. A Billboard is like a Plane object, excepting that it is always facing the Camera.



- From the MaterialBillboard settings, add a Sprite element, select the Material drop-down menu and create a new Material asset.

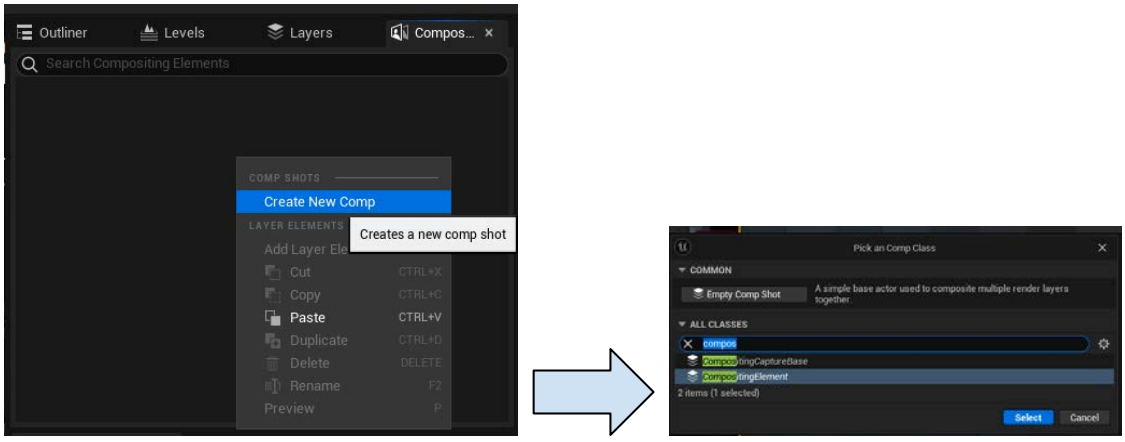


- Still in the MaterialBillboard settings, under Sprite > Elements > 0, change the Base Size X and Y values to 90 and 160. Change the Z position of the Billboard to be coinciding with the floor.

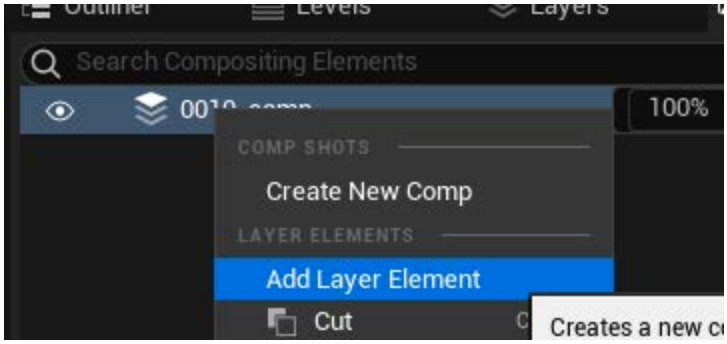


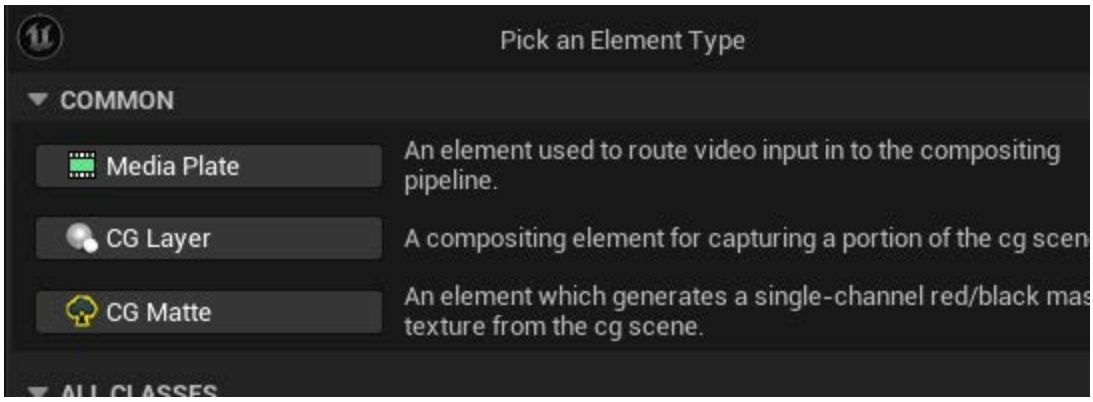
Create Chroma Keyer Compositing

- From the top right ‘Composure Compositing’ window, right-click and select ‘Create New Comp’. On the pop-up window select ‘CompositingElement’.

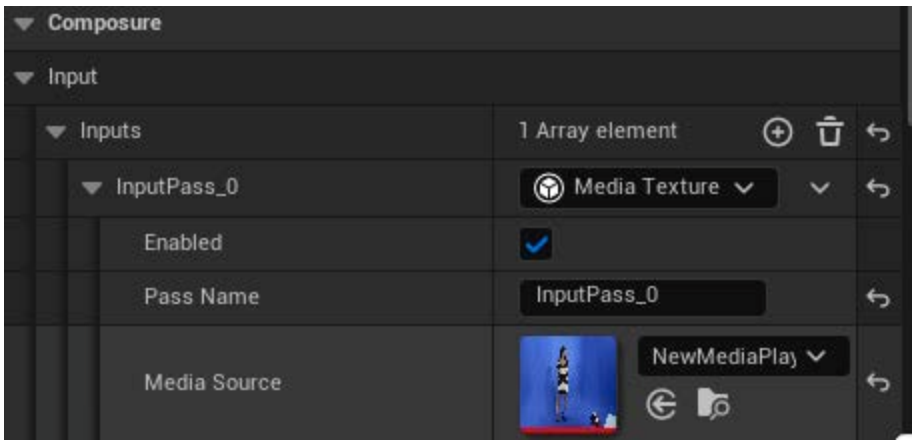


- Select the Compositing Element, right-click and select ‘Add Layer Element’. On the pop-up window select ‘Media Plate’.

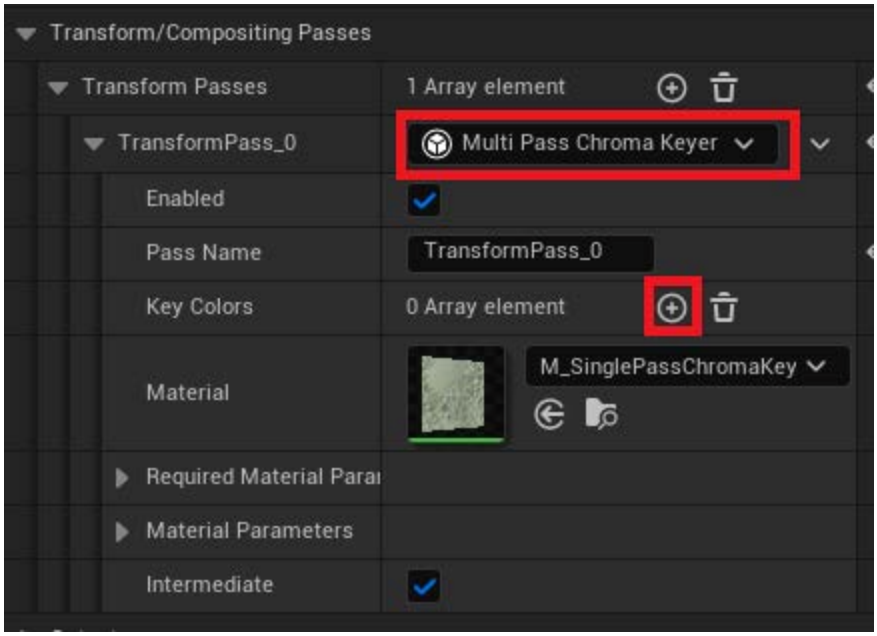




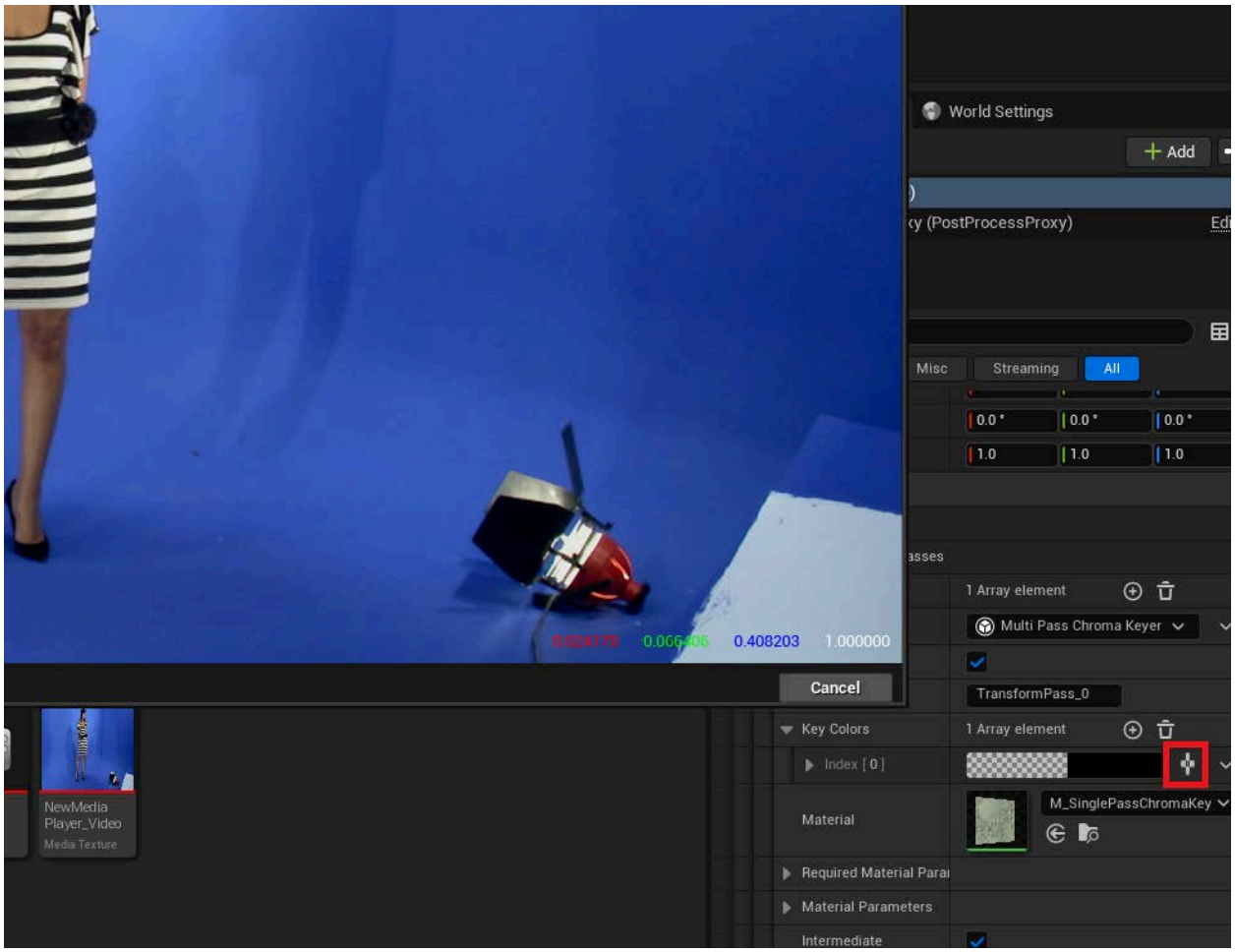
- In the Media Plate settings, go to 'Composure > Input > Inputs > MediaSource' and drop the 'MediaPlayer_Video' object into the 'Media Source' field.



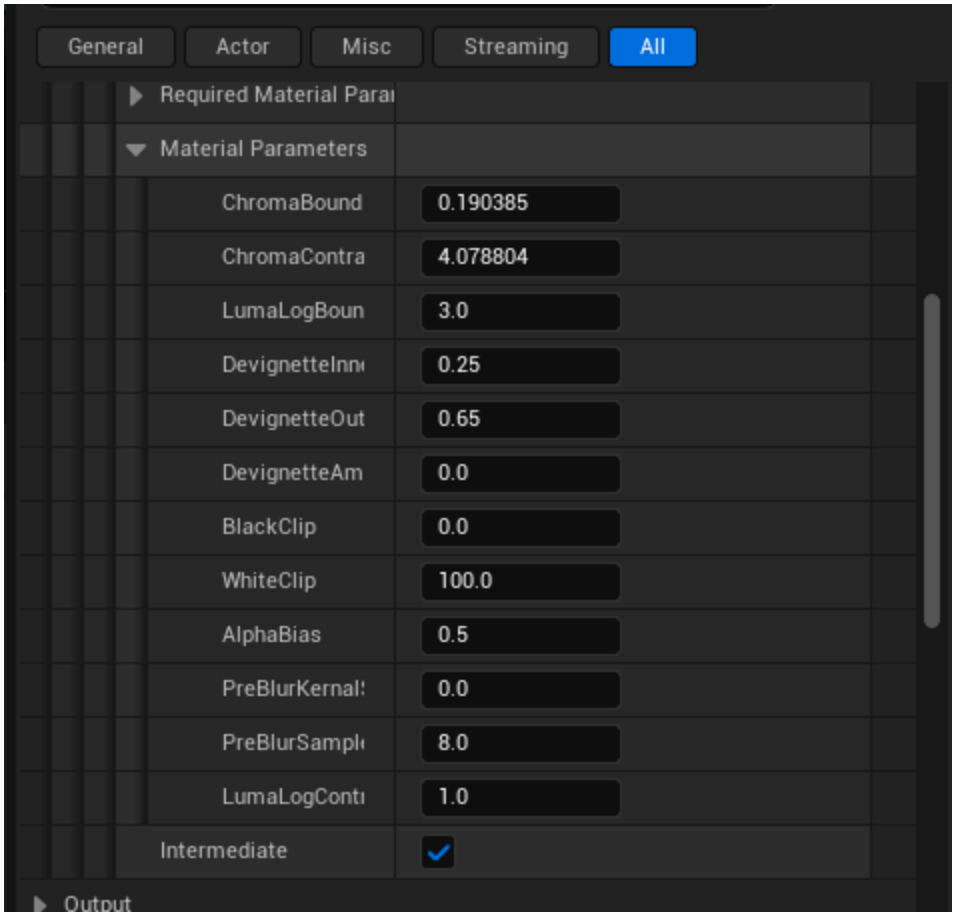
- Go to 'Transform Passes > Chroma Keying' and add a 'Key Colors' Element.



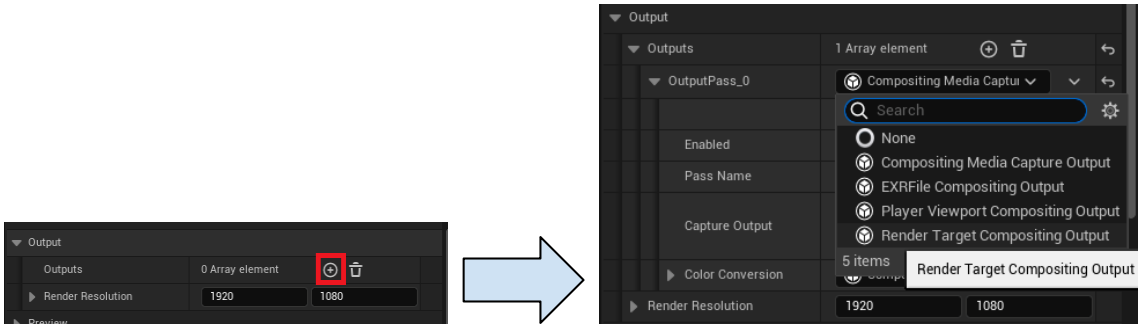
- Click the Color Picker and select a color representative pixel for the transparency.



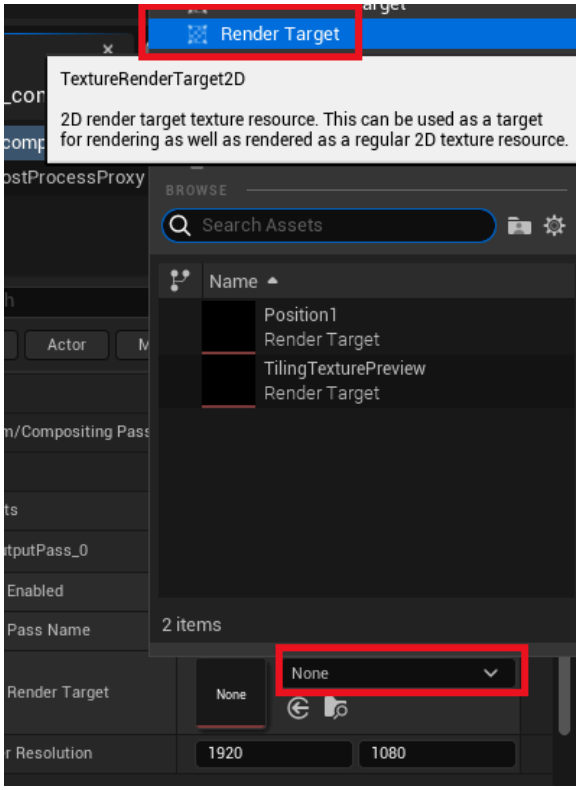
- Under ‘Transform Passes > Material Parameters’ you can refine the Chroma sensitivity using e.g. with the ChromaBound and the DevignetteOuter parameters.



- Go to ‘Output’ and add a ‘Outputs’ Element. Set the ‘OutputPass_0’ to ‘Render Target Asset’.

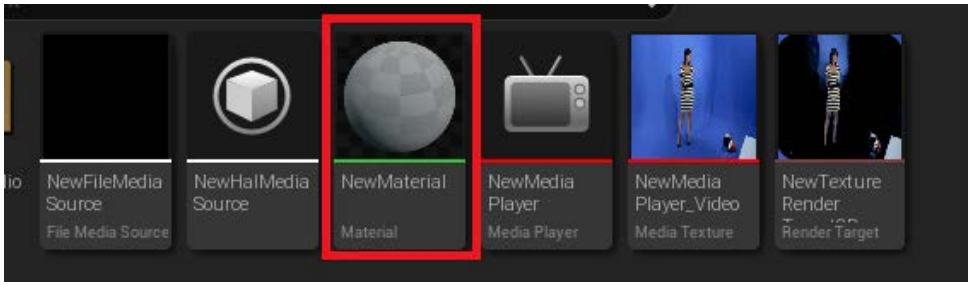


- Under ‘OuputPass_0 > Render Target’ click the drop-down menu and create a new ‘Render Target’ Asset.

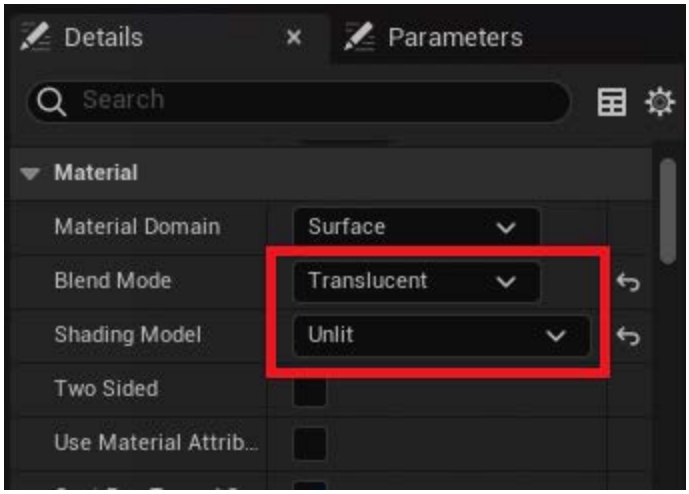


Create material and resize texture for the Billboard

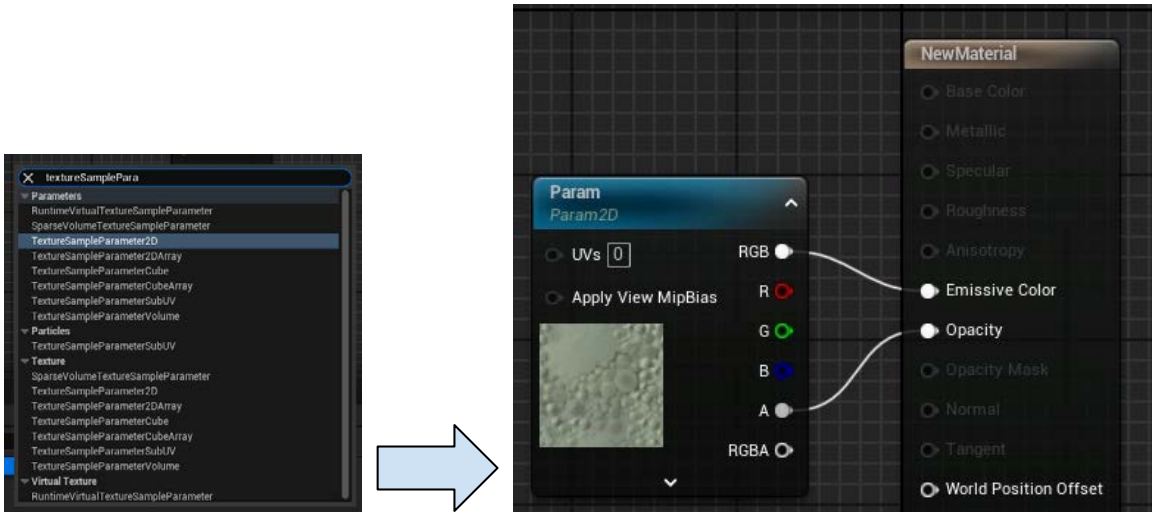
- From the Content Browser, double-click the Material Asset to edit it.



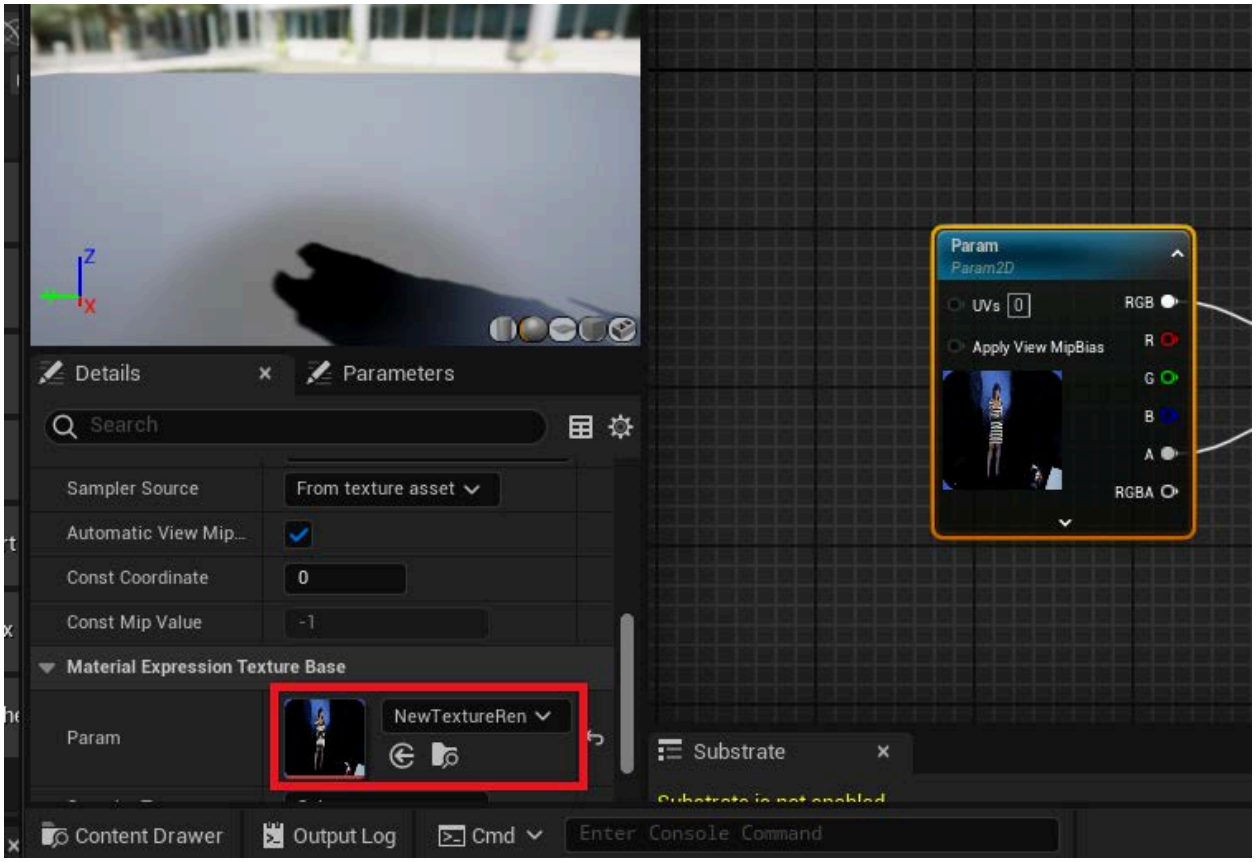
- From the left panel, set the Blend Mode to ‘Translucent’ and the Shading Model to ‘Unlit’.



- In the main window, right-click and add a 'TextureSampleParameter2D. Link then the created object' 'RGB' output to the Material's 'Emissive Color' input and the 'Alpha' output to the 'Opacity' input.



- In the 'Param' object's parameter in the left panel, go to 'Material Expression Texture Base > Param' and select the 'TextureRenderTarget2D' Asset. Apply and Save.



This last operation should display the input media into the Virtual Studio, well done!

AB Switch Component

What is an AB Switch?

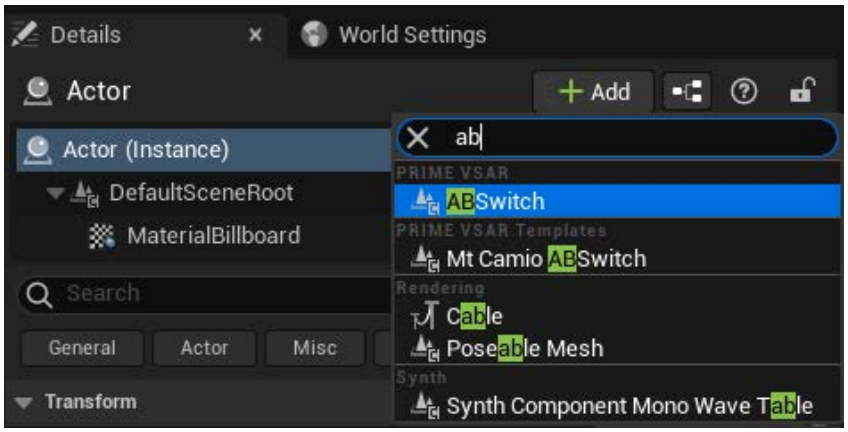
AB Switch is a tool that features texture transitions mapped onto a 3D object, for example, a virtual screen in scenes in which transitions occur. It behaves similarly to a standard A/B (or Program/Preview) transition effect in a switcher/vision mixer.

Setup

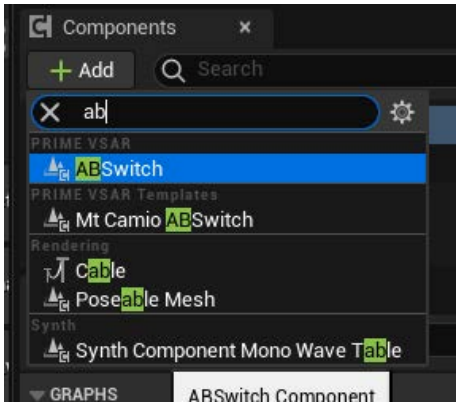
There are two ways to create an AB Switch:

- Add the AB Switch component to the actor of your choice by using the Add Component button.

From the Details panel:

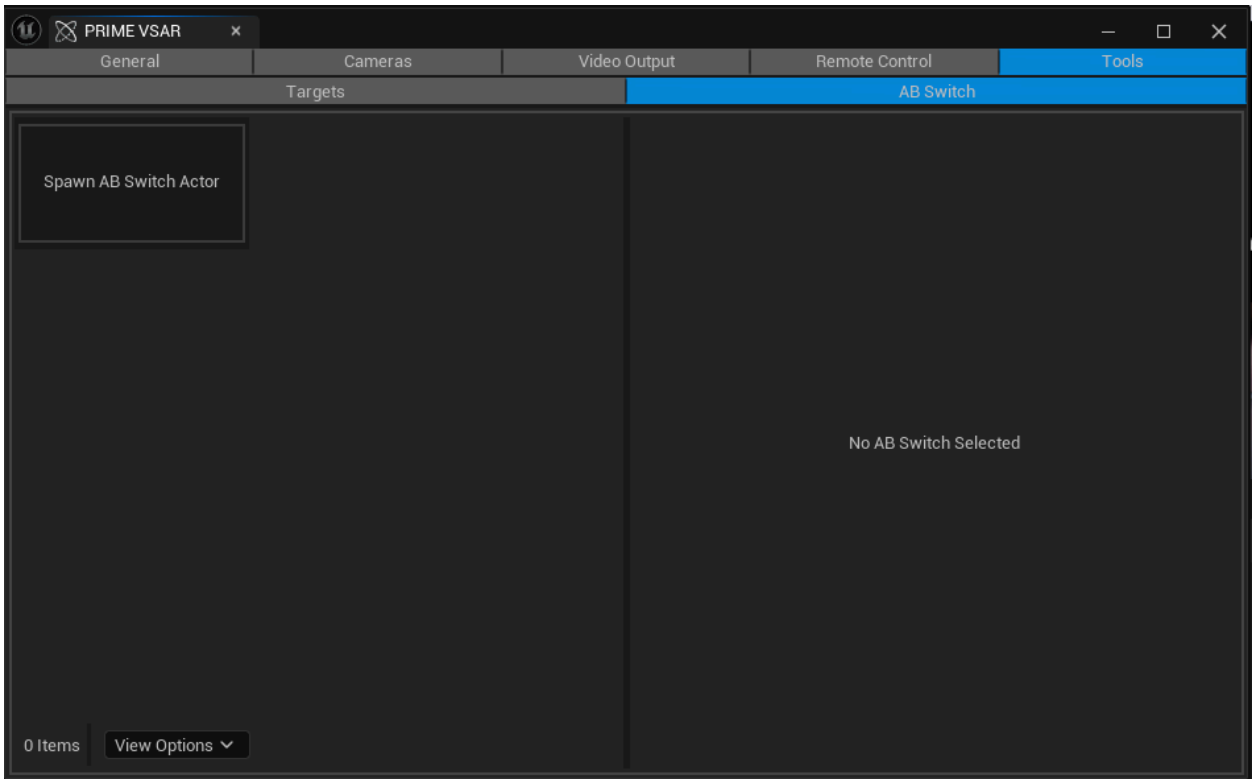


From a blueprint:

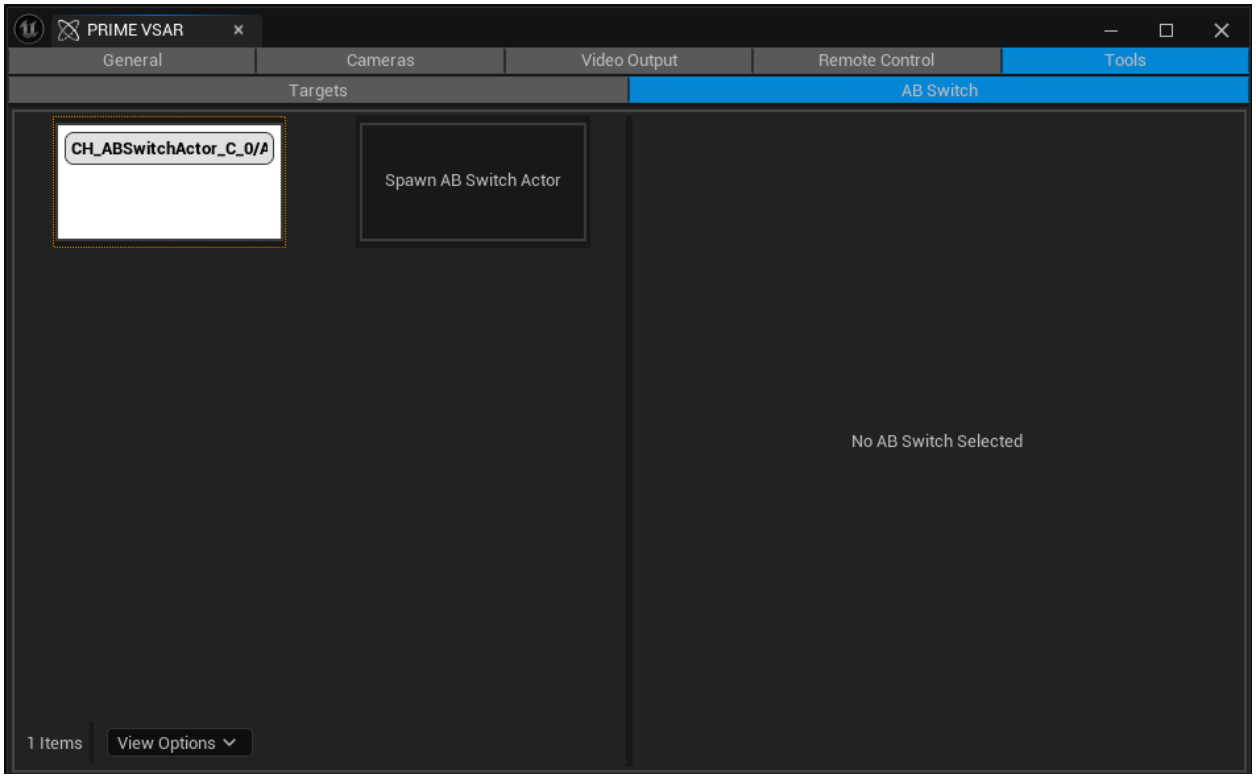


Note: all the AB Switch will show up in the config panel (components and the AB Switch Actor).

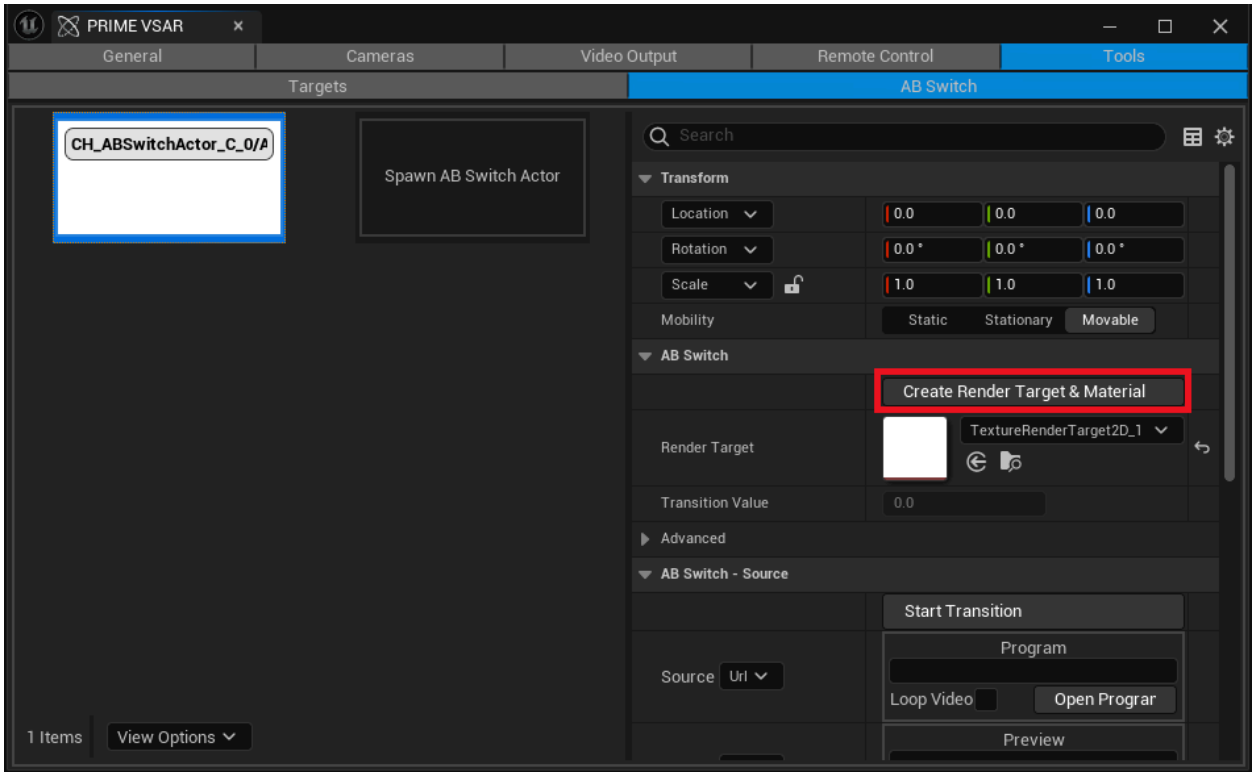
- Spawn the actor from the config panel:
    a. Go to the PRIME VSAR Config panel.
    b. Choose the AB Switch category under Tools and click Spawn AB Switch Actor.



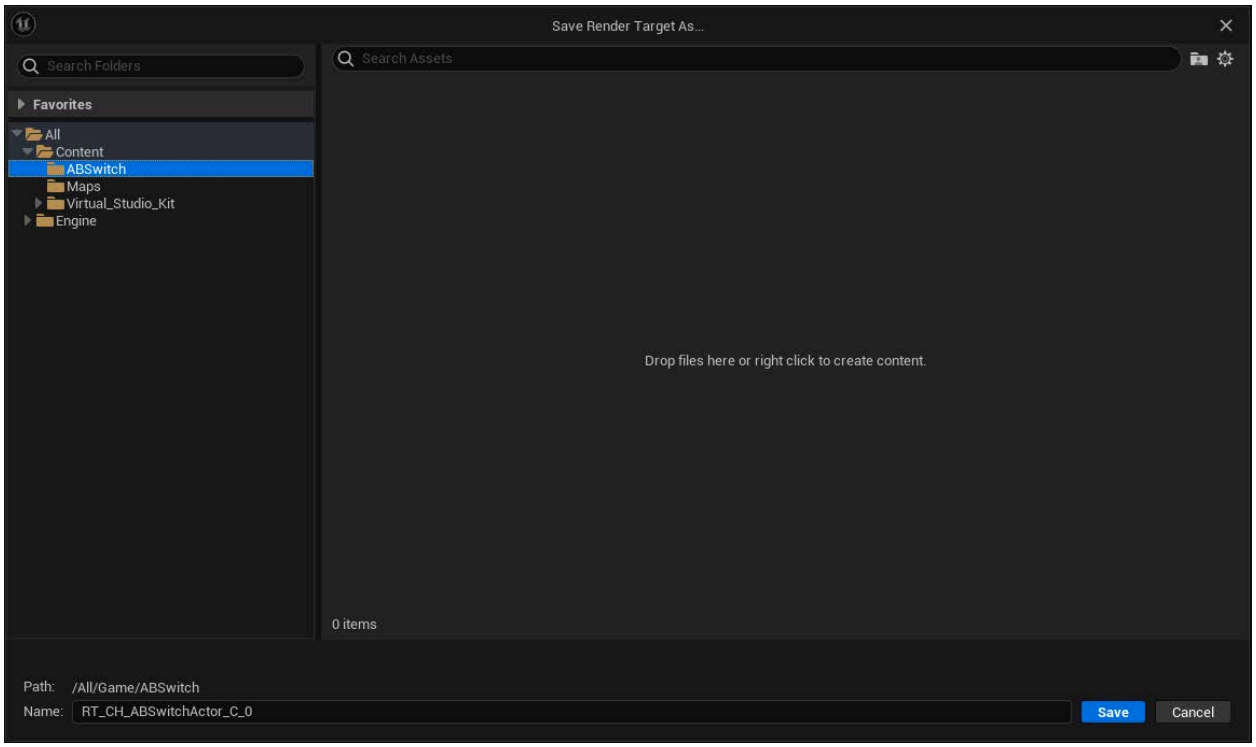
- Click the newly created AB Switch icon.



- In order to assign the AB Switch to an object, you must create the Render Target and the Material and assign it to your object(s). Click Create Render Target & Material.



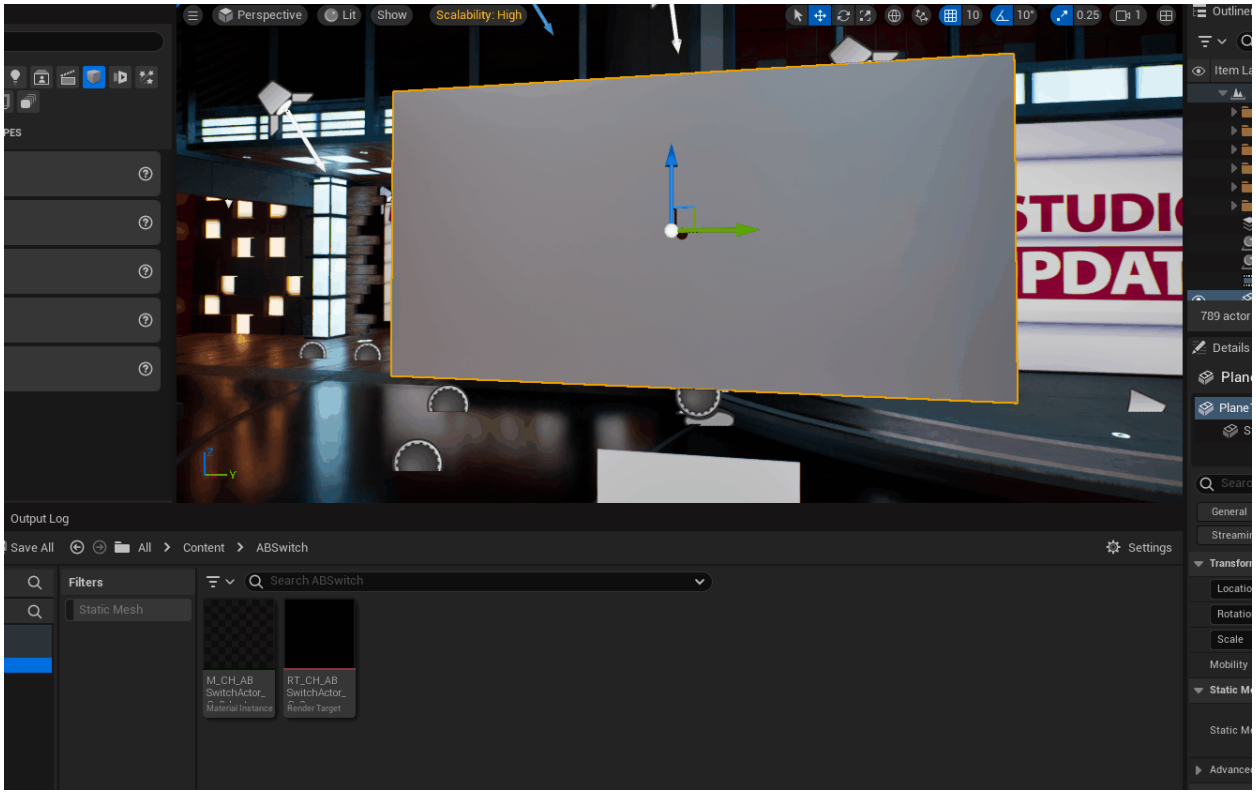
- Two Save Prompts display, one to save the Render Target, the other to save the Material.



Here we created an ABSwitch folder to save these assets.

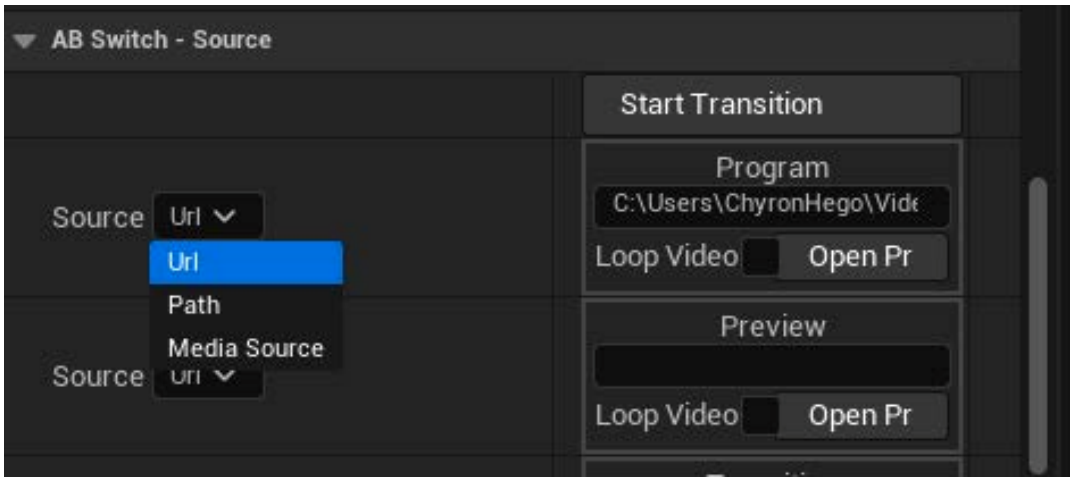
- Drag and drop the Material Instance onto one or multiple objects.

The object will become transparent, because no source is open/loaded.



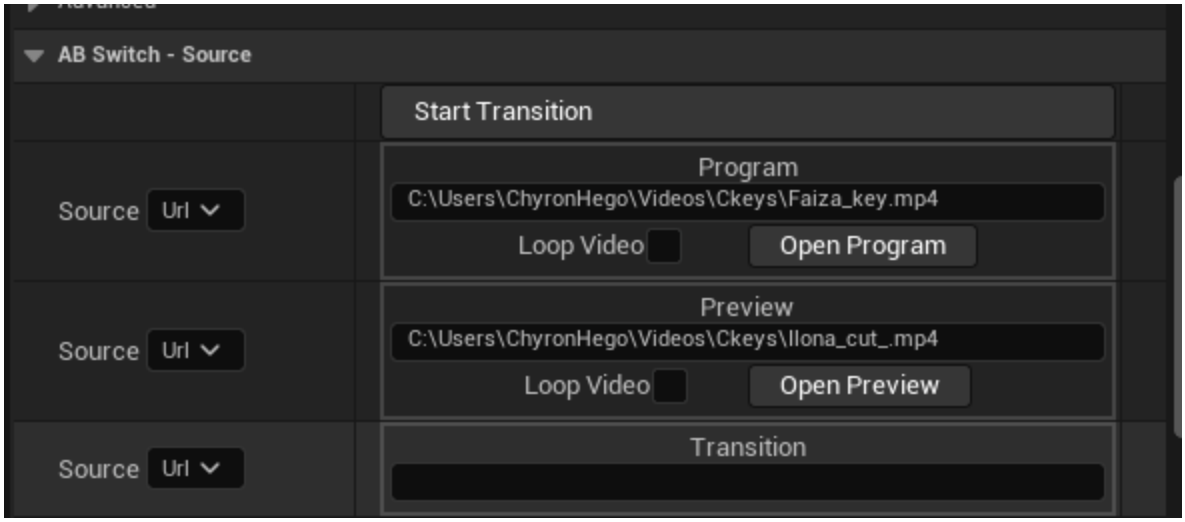
AB Switch - Source

You can preview the effects from the AB Switch - Source area.



Multiple source types can be used:

- Url
- File Path (Path)
- Media Source



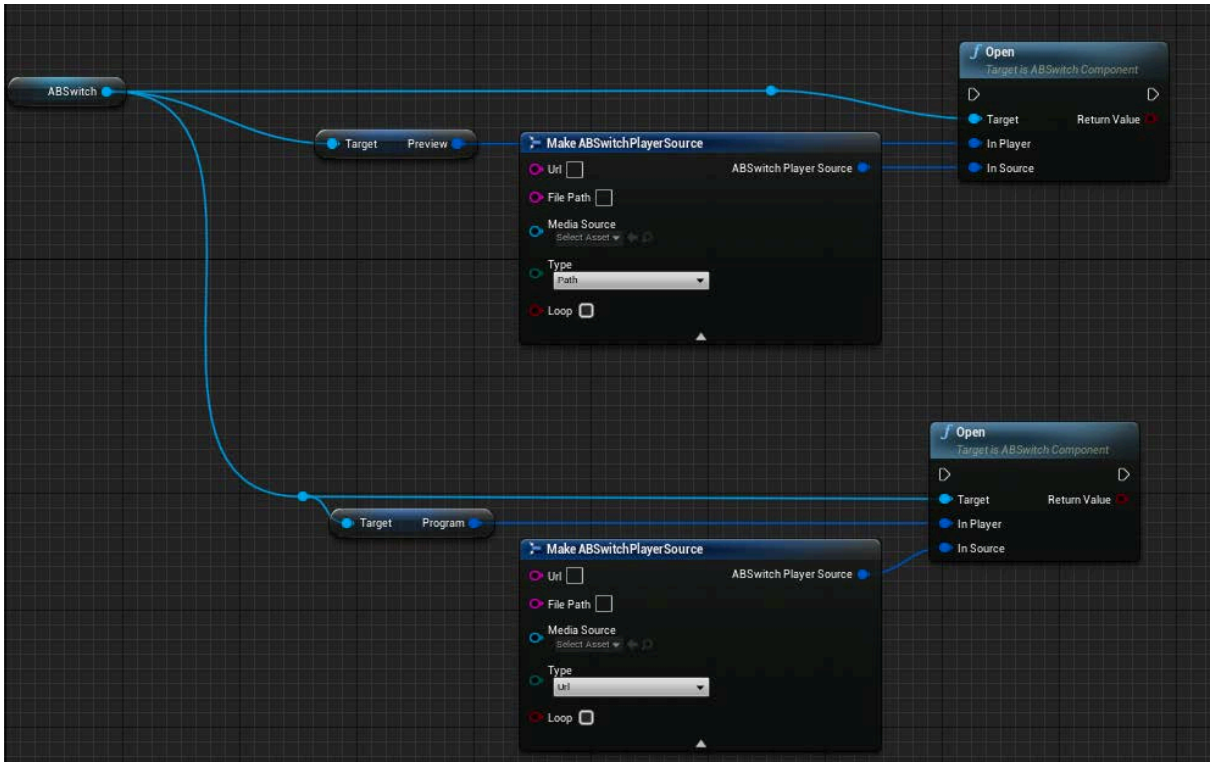
Url are formatted as follows:

- Files (video, images): File://C:\Users\User\Clips\2019-03-05_13-07-24.mp4
- External input: hal://input0_0

“Loop Video” the source loops forever after it has been opened. Normal behavior is that after the source is opened it plays from start to finish and then it is closed, enabling loop video prevents this.

“Open Program/Preview” opens the source.

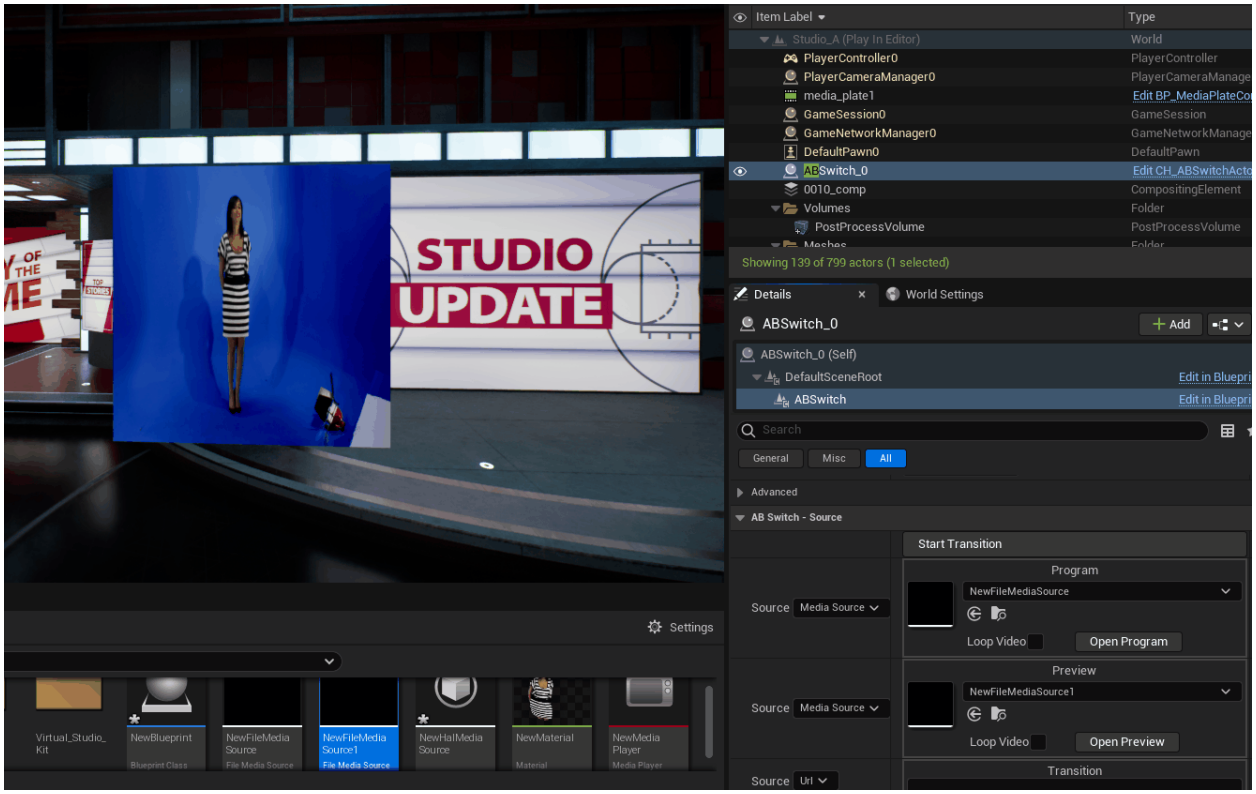
📝Open Program/Preview is for preview purposes only, do not use in production. Use open function instead



📝 AB Switch supports sources with alpha channel (color channel), making the material transparent when the source alpha is at 0 value.

To preview the transition:

- Click Start Transition.



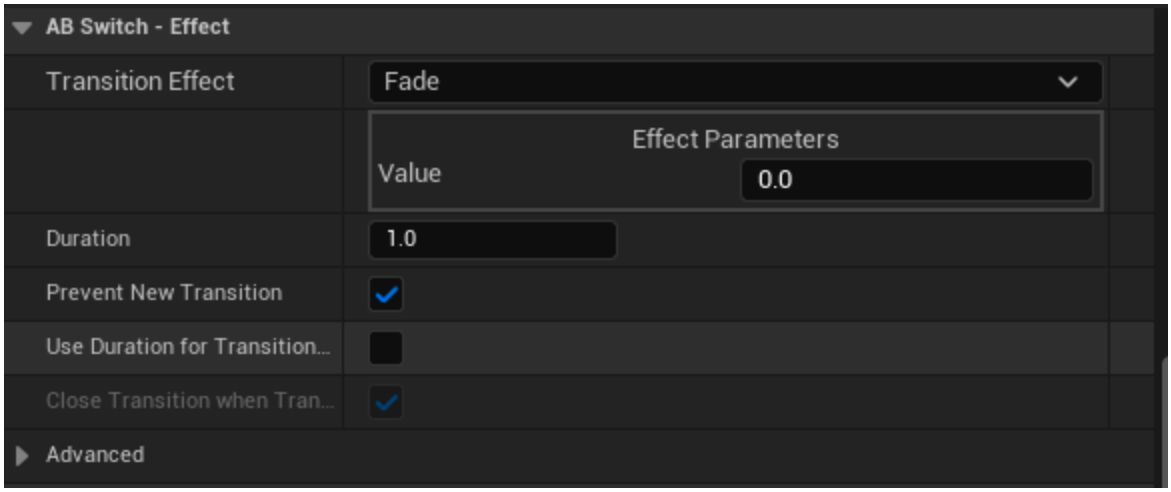
Note:

The third channel source is the “Transition” channel.

It’s used for effects that are using a third source to do the transition (more details in the **AB Switch - Effect**).

AB Switch - Effects

You can select your transition effects option under the AB Switch - Effect area.



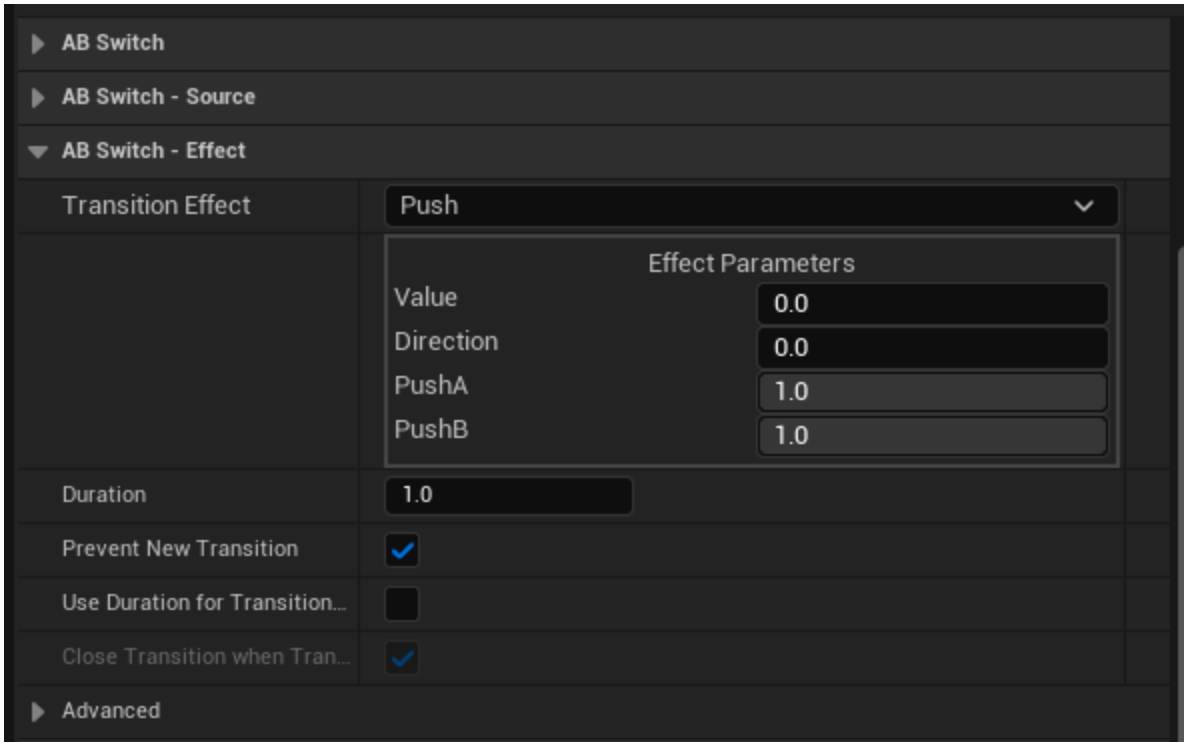
Transition Effect: The selected effect.

Effect Parameters: Each effect parameters are automatically generated from exposed parameters in the effect material.

Value: The value of the effect. Range: 0 - 1. 0 is no effect, 1 is the full effect, 0.5 is half effect.

This value can be changed for testing, but it will be overwritten by the AB Switch when a transition is happening

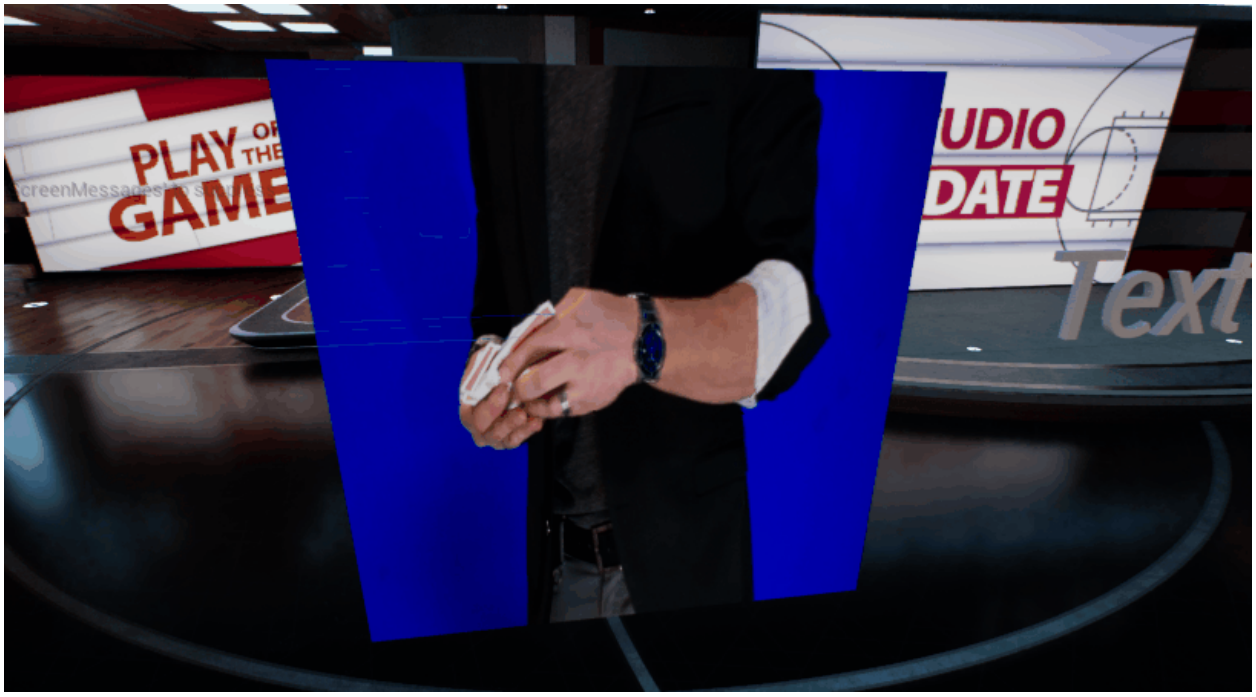
Note: Some effects have whole number (i.e., no decimal) requirements, you can hover your mouse to view a description of the values.



Effect List: Currently implemented effect in the list. Available effects (name and individual properties) are as follows:

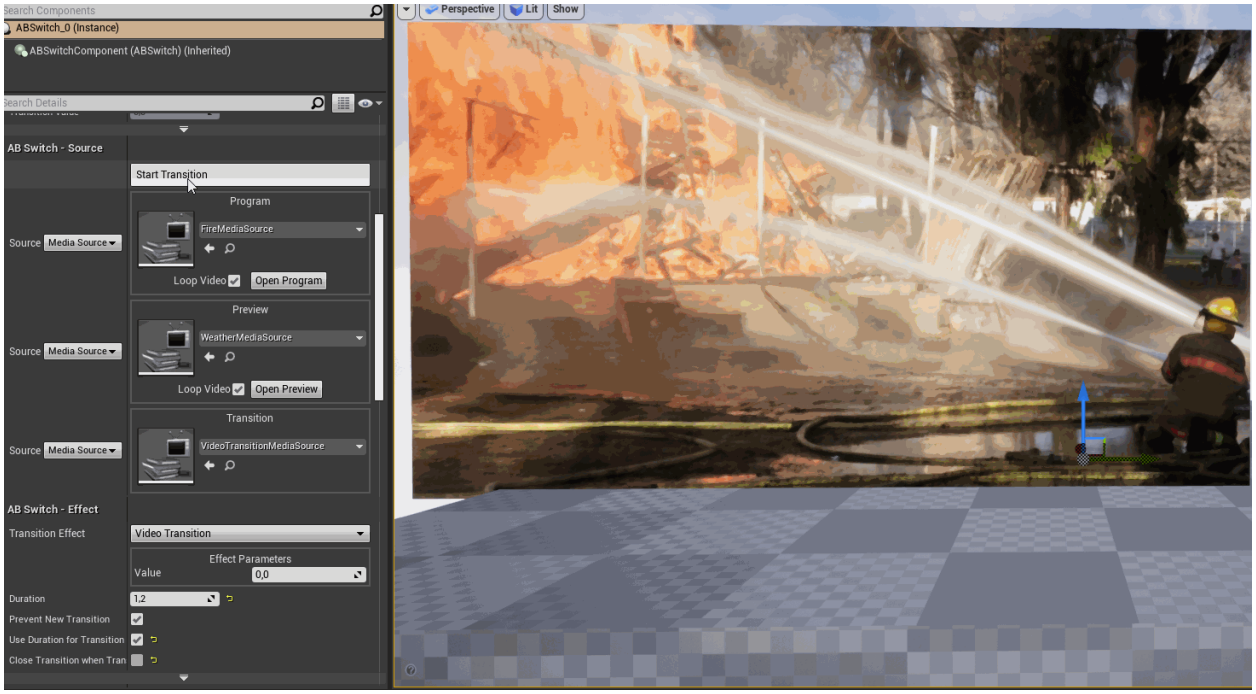
- **Fade**
- **Disc**
- **Push**
- **Wipe**
- **Alpha Transition**
- **Video Transition**

Alpha Transition: Allows the use of Alpha Channel (color channel) value as the transition value taken from the “Transition” Source Channel.



Video Transition: Video Transition allows you to use the video set in the “Transition” Source Channel, in order to do a “hover” transition.

⚠ Images are not supported as “Transition” Source Channel for Alpha/Video Transition



Duration: The time it takes the Alpha/Video Transition to to play from start to finish.

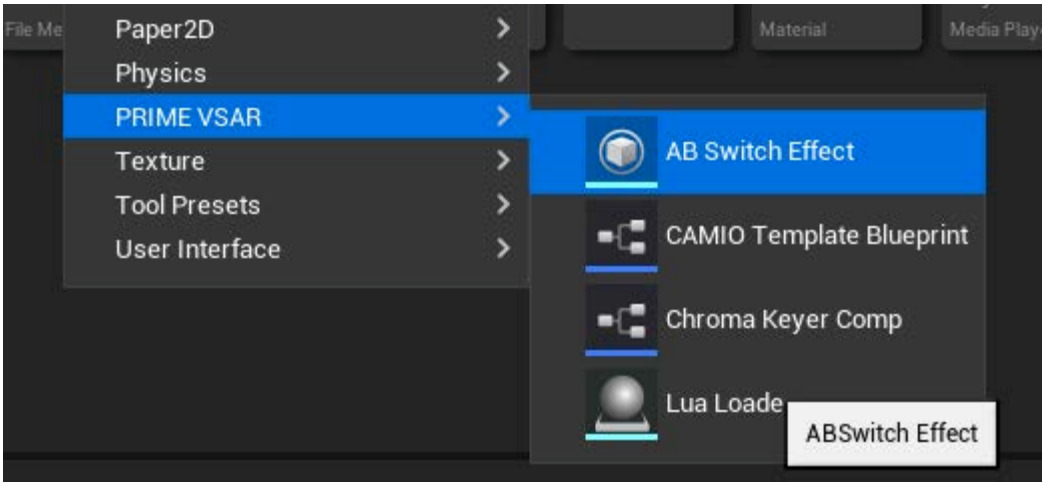
Prevent New Transition: Prevents another transition to start before the current one is finished

Use Duration for Transition Videos: Instead of using the length of the video for the transition the Duration parameter is used.

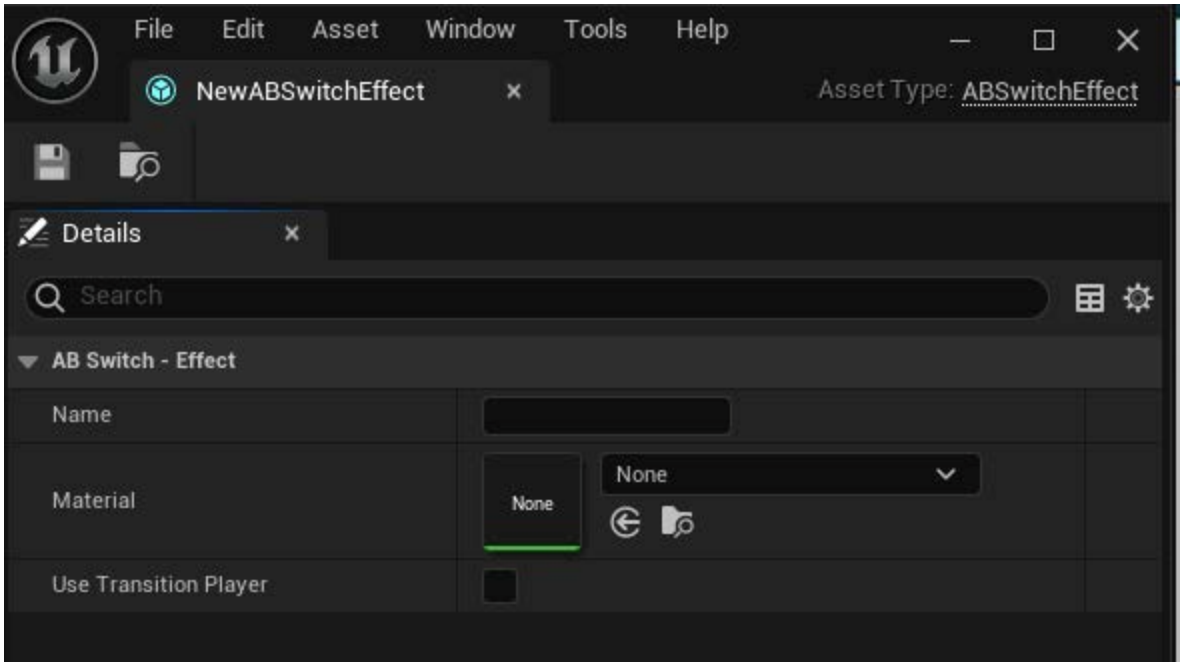
Close Transition When Transition Finished: Should the transition player immediately be closed when transition is finished (only valid when “Use Duration for Transition Videos” is enabled)

Custom Effects

Effects are Material based Structure. so we will need two thing to create custom one, Material and the AB Switch Effect Structure that can be added through the content browser Add/Import → PRIME VSAR → AB Switch effect:

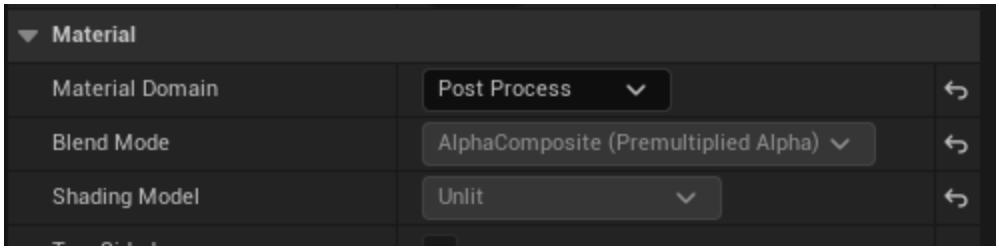


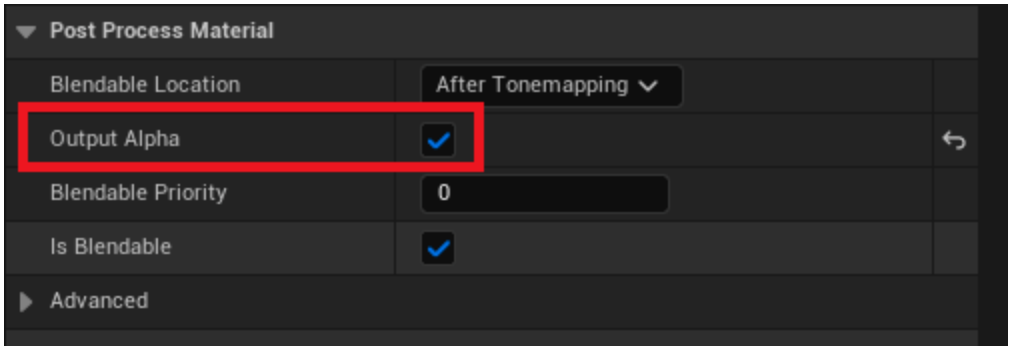
Wen opening the newly create AB Switch Effect we can see:



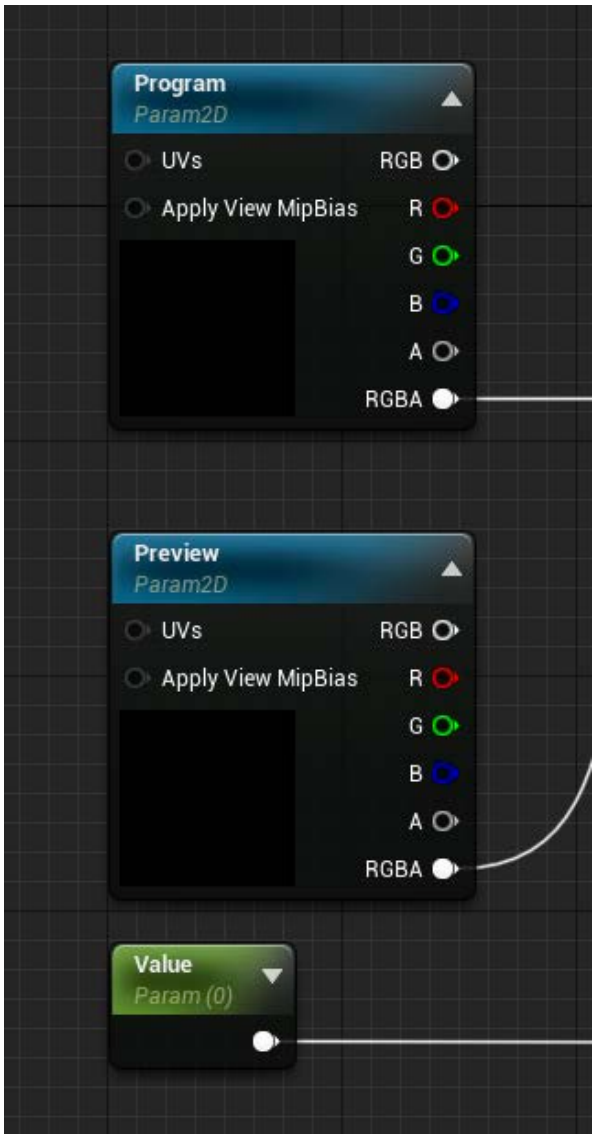
- Name - the name that will be visible in the Transition dropdown
- Material - material used for the transition
- Use Transition Player - this is when we want to use the “Transition” Source Channel inside the material

Material used for AB Switch Effect is expected to be post process domain, the blend mode to be alpha composite (this can be set after Output Alpha is enabled), the shading Model to be Unlit and Output Alpha enabled:

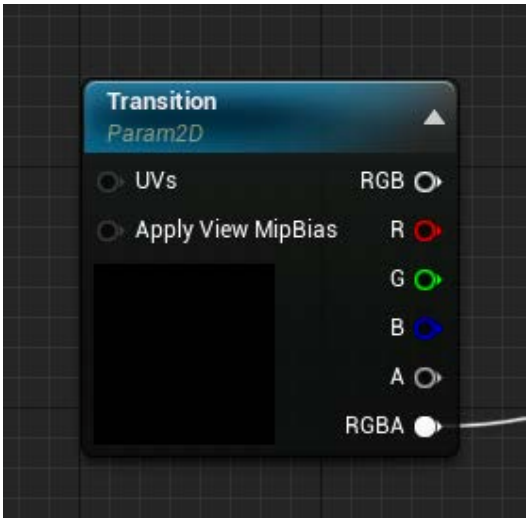




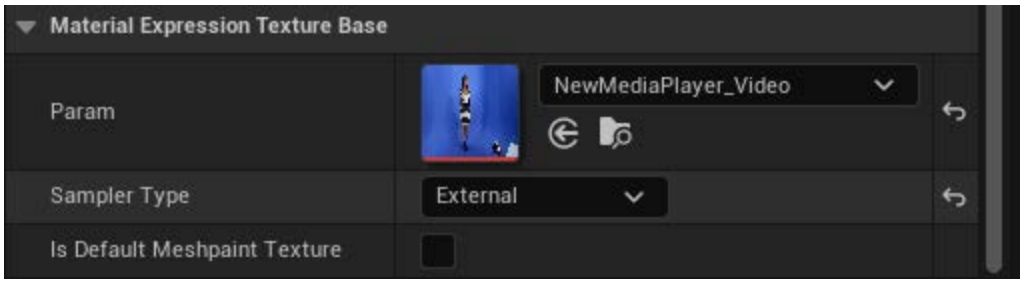
Material gets provided Two Textures Parameters and Value Parameter (with specific names that have to match):



When using the Use Transition Player the Transition texture parameter becomes available to the material:



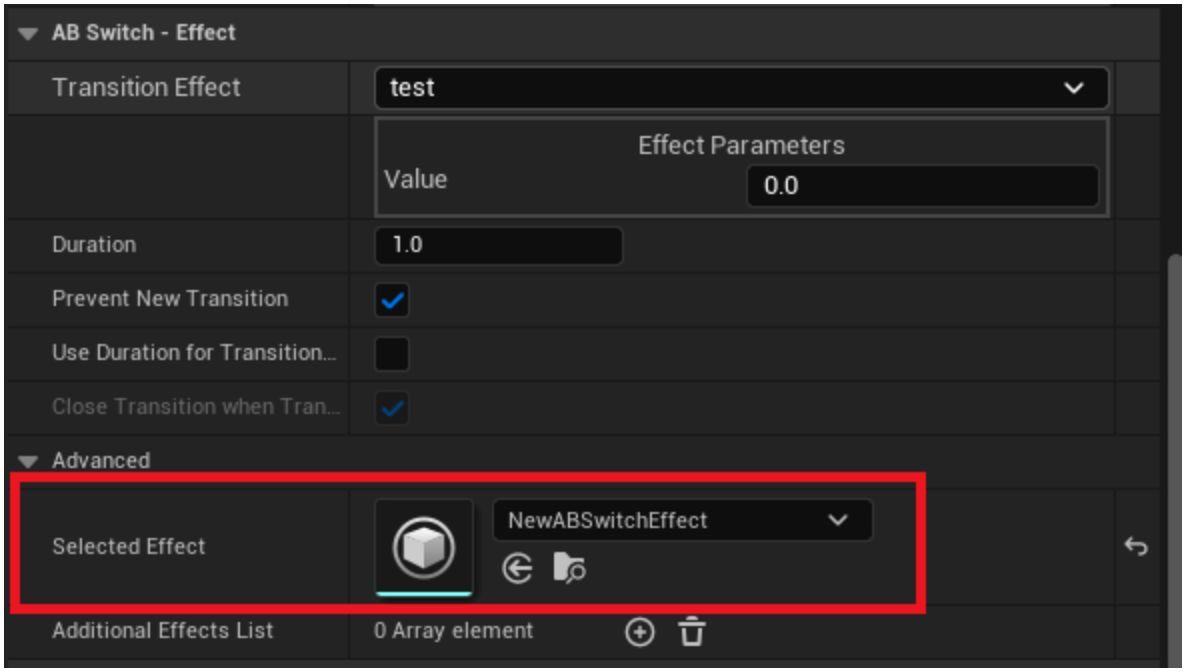
All Textures Parameter needs to have their sampler type set to be external:



Value Parameter is used by the AB Switch to drive the transition, When using Use Transition Player, Transition Texture Parameter is used instead (of Value Parameter) to drive transition.

To use your material:

- Assign it to the AB Switch effect in the material section
- Assign AB Switch effect to the Selected effect in the AB Switch


AB Switch - Effect
Transition Effect
test
Effect Parameters
Value
0.0
Duration
1.0
Prevent New Transition
Use Duration for Transition...
Close Transition when Tran...
Advanced
Selected Effect
NewABSwitchEffect
Additional Effects List
0 Array element

Primitives

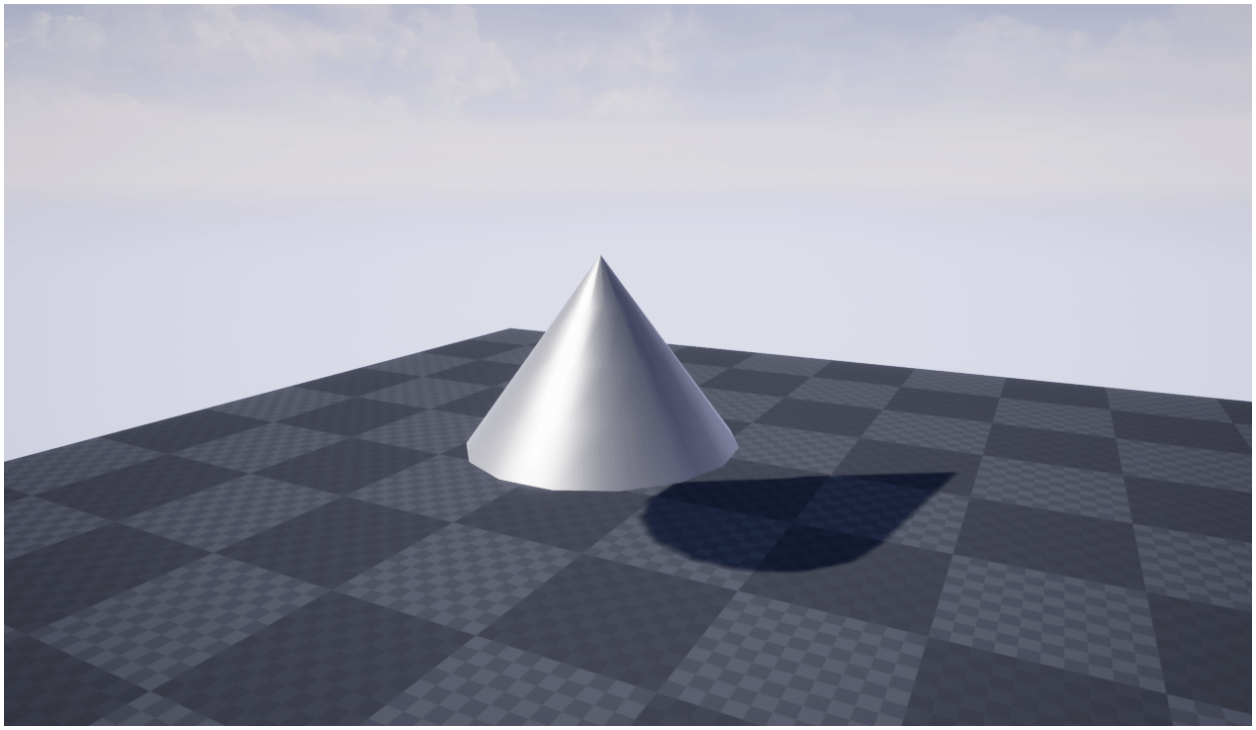
About Primitives

VSAR provides a set of procedural primitives. Procedural primitives, as opposed to static elements, are customizable and can be modified at runtime, which can be useful for dynamic geometries such as bar charts, pie charts and so on.

List of Primitives

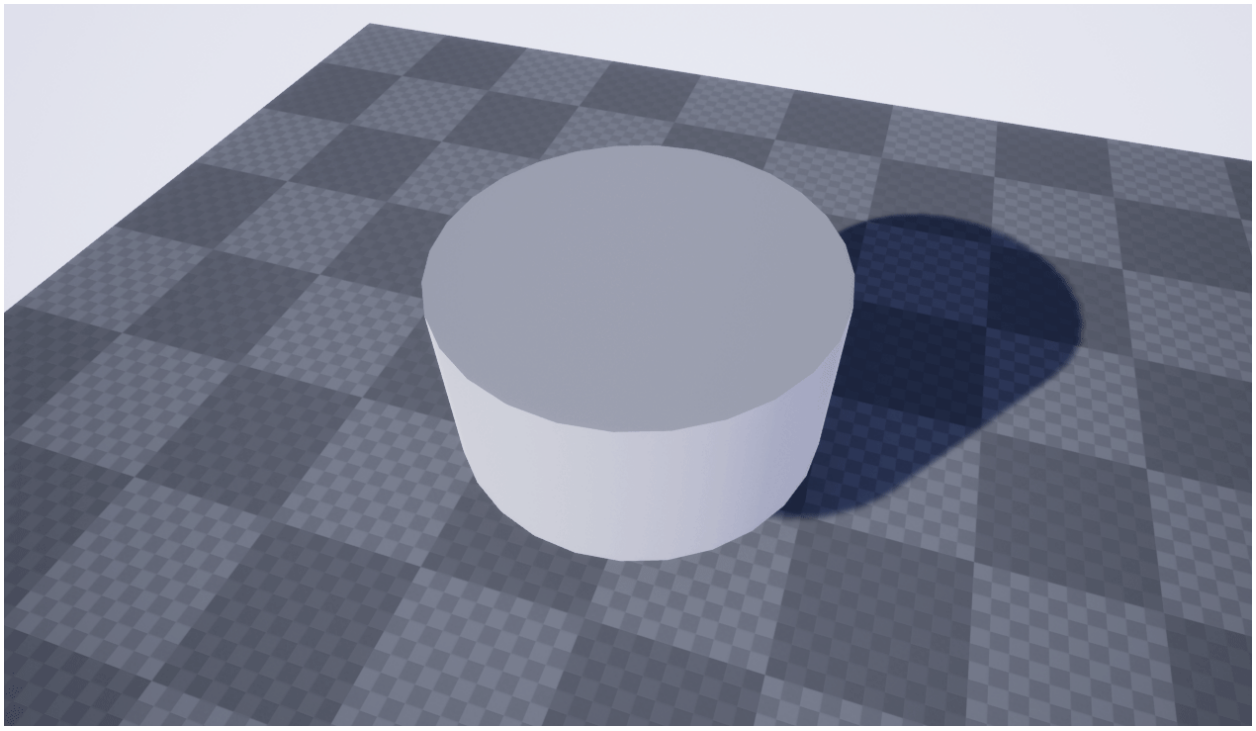
Cone

A beveled cone. Top vertex can be beveled as well. It is a component of the arrow.



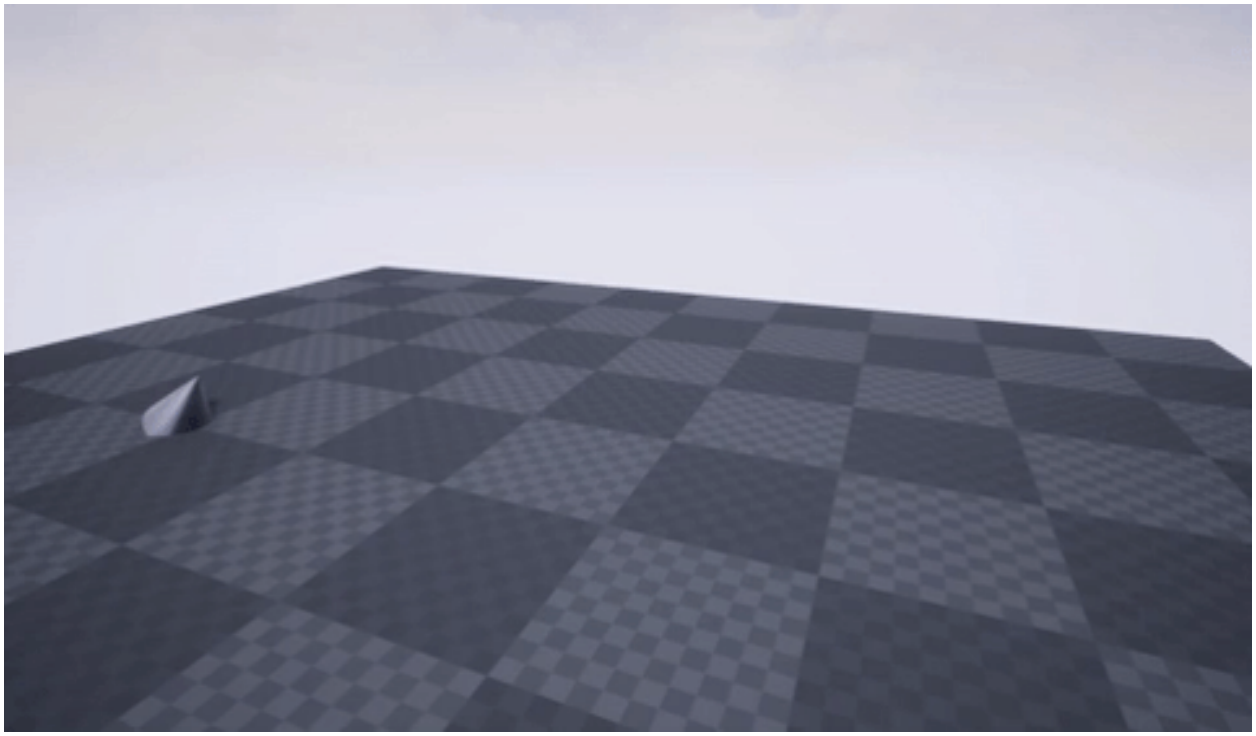
Cylinder

This is the base for Pie charts. Many configurations are possible, including cylinders with holes.



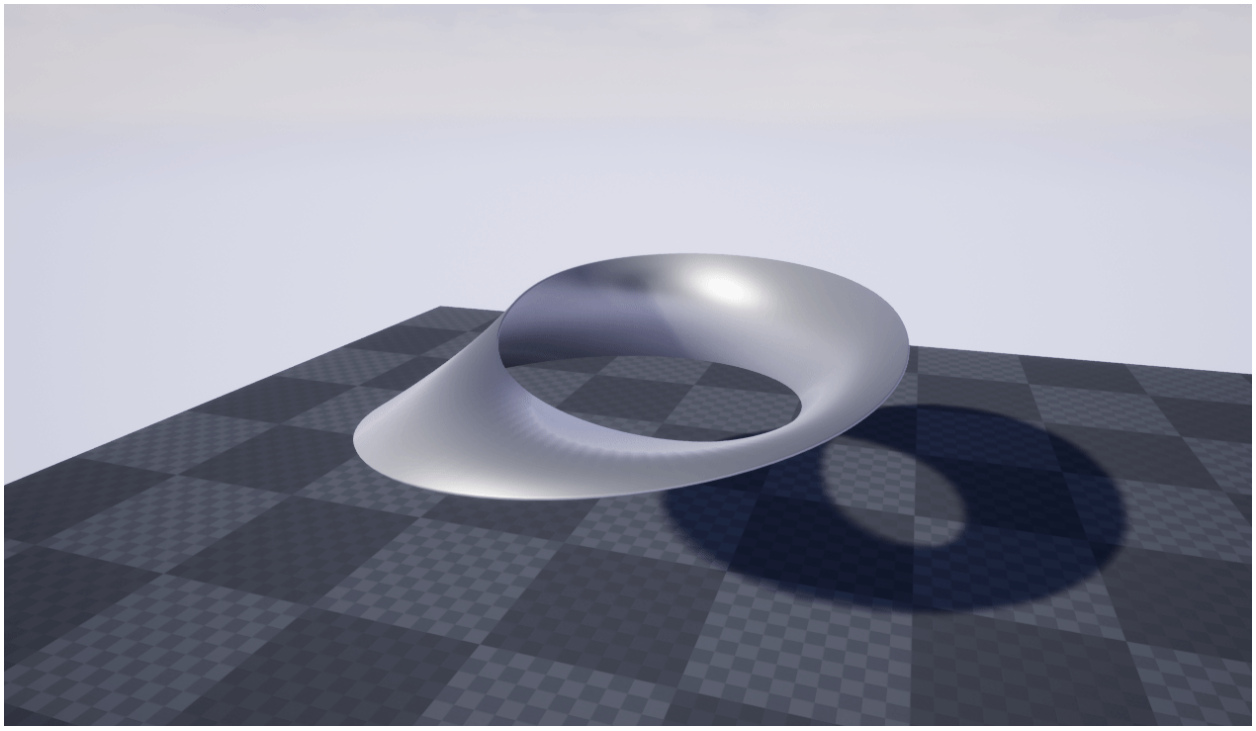
Arrow

An beveled Arrow composed of a Cone and a Parametric Curve 2D (parabola).



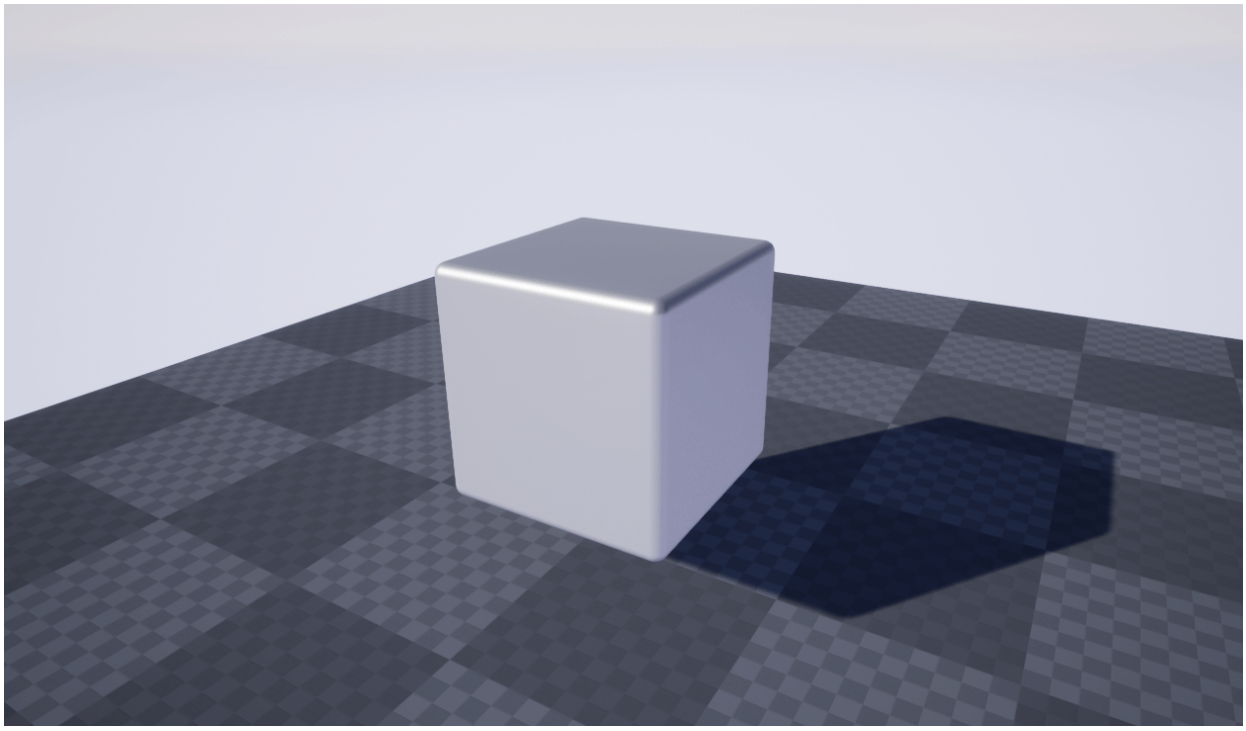
Moebius Strip

Möbius loop is a surface that can be formed by attaching the ends of a strip of paper together with a half-twist.



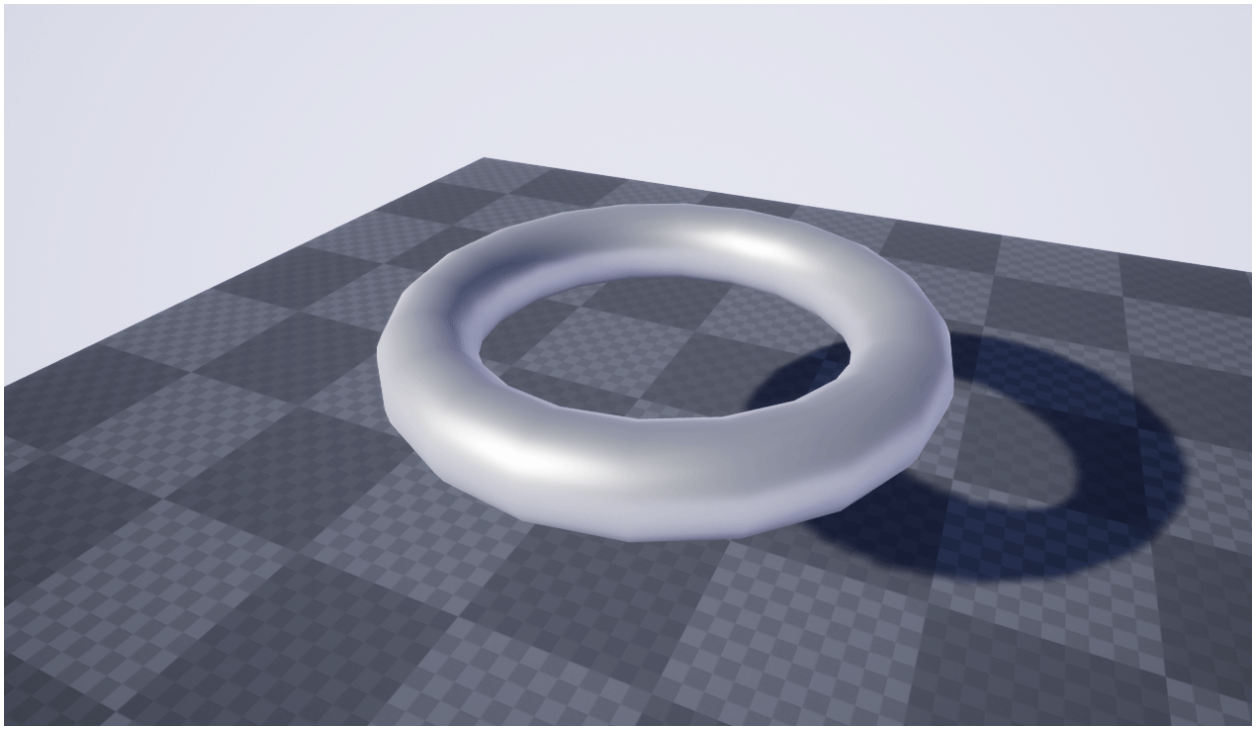
Box

A simple beveled box.



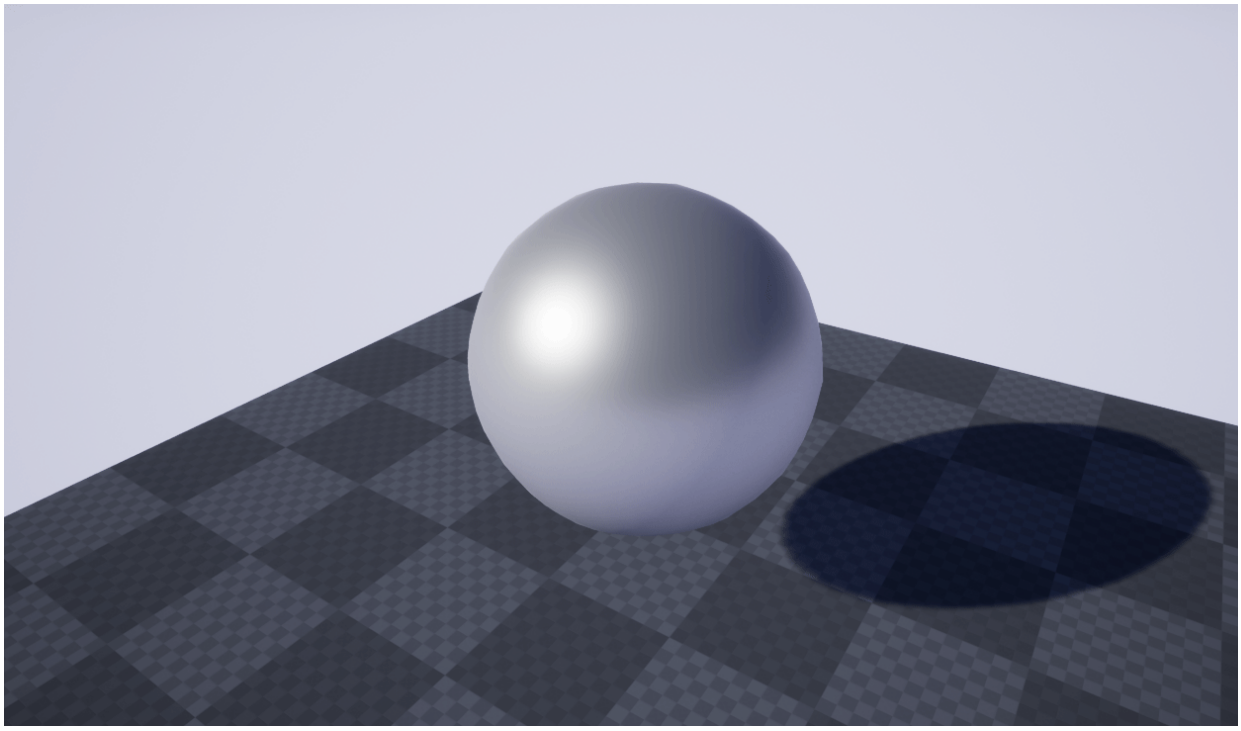
Torus

A Torus.



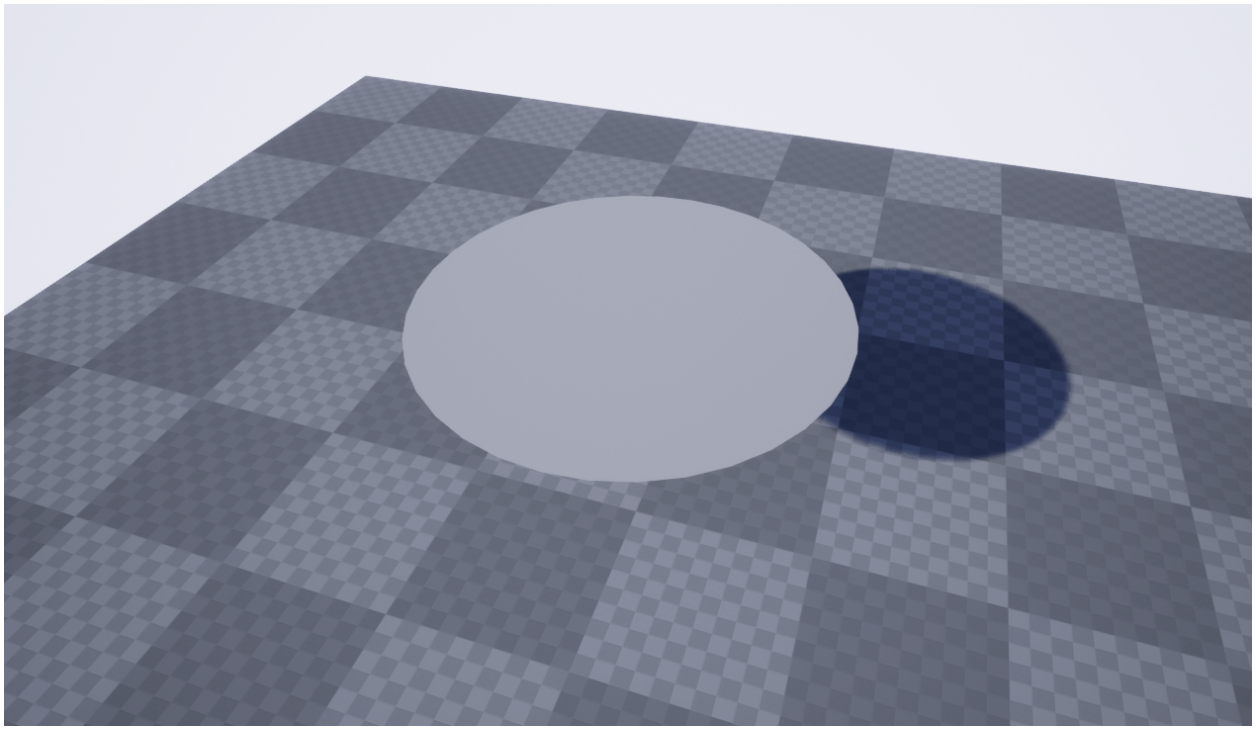
Sphere

An ellipsoid displayed with triangles (aka icosphere).



Disk

A flat, simple disk.



Text

Text is considered deprecated please use **Camio 2D/3D Text Component** or Text3D

📝 Alternative is provided by Unreal Engine called “Text3D” (it is considered experimental), it is component that can be added to actors.

📝 if text transition is needed, refer to **Camio 2D/3D Text Component** section for text with transition out of the box.

Usage

A primitive is represented by a Component object in the Unreal Editor. Multiple Components can be attached to an Actor object (e.g., a pie chart would be composed of multiple MtCylinder components). The Component inherits the parent Actor’s transformation in addition to its own transformation (Location, Rotation, Scale).

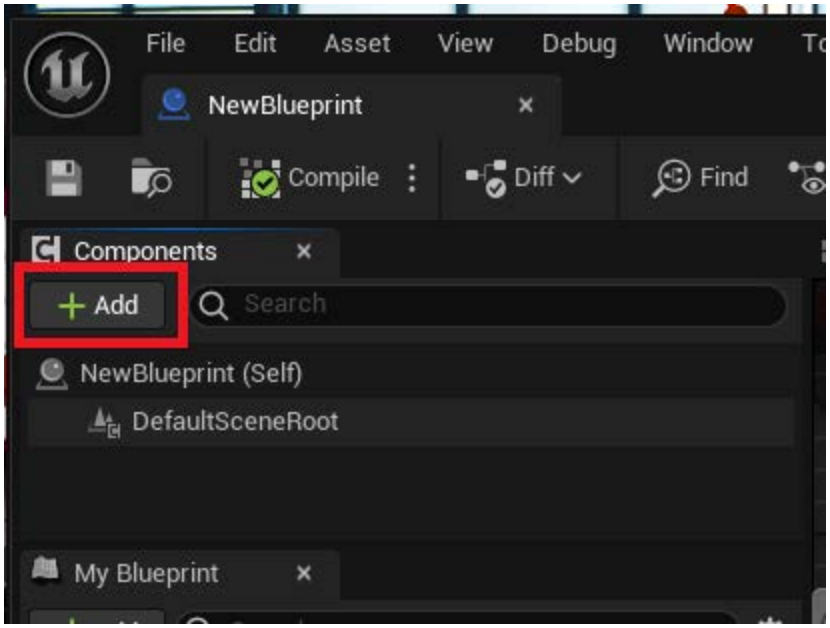
Each primitive exposes a set of properties that the user can get/set in real time with the Editor UI or Lua scripts.

The property center location is common to all primitives and defines the normalized coordinates of the primitive’s anchor point in 3D space.

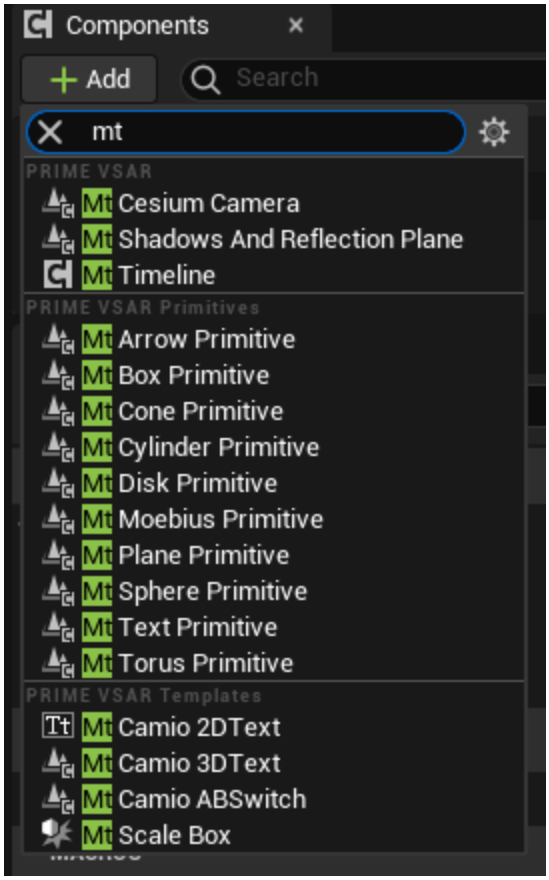
For example, a “bottom-center location” anchor point for a box would be equal to a value of (0,0,-0.5).

Unreal Editor

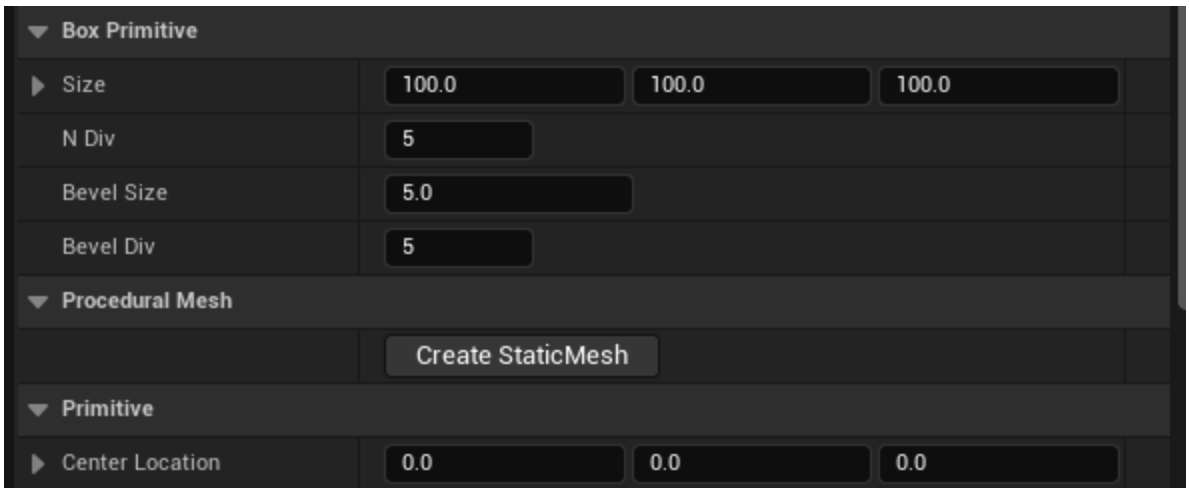
Add a **Component** to an **Actor**.



Tip: the user can search for the Component in the search bar (e.g., MtBox, MtCylinder, etc.).



Edit the properties in the Details Panel.



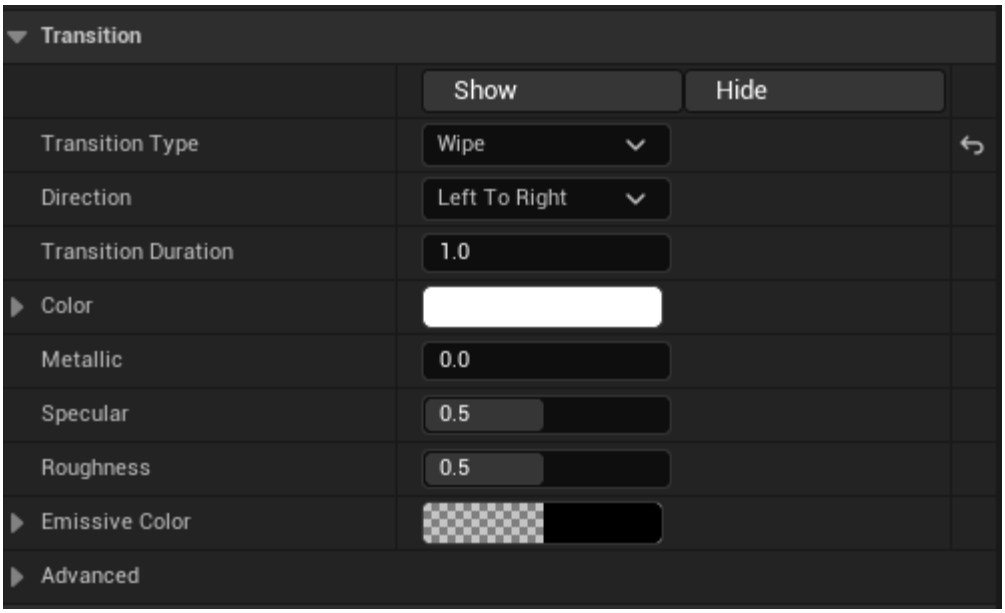
Lua Scripting

For documentation about LUA commands, please refer to PRIME VSAR API Guide

Transitions

⚠ This feature is still in Beta, testing was not finished !

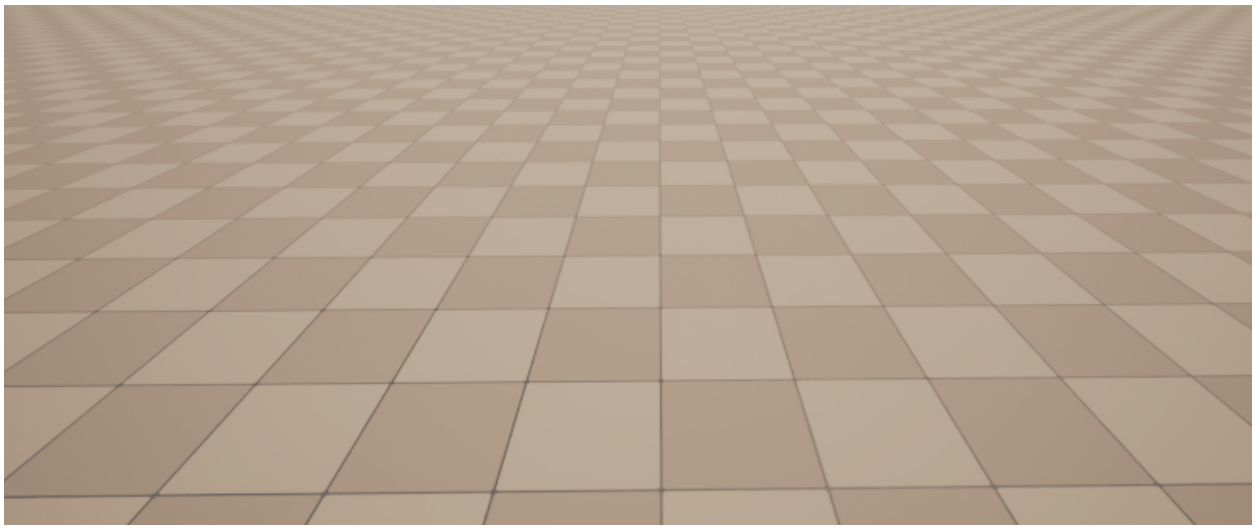
All primitive components (except MtTextPrimitive) support show/hide transitions. Users can choose from multiple types of transitions, directions and customize Transition Duration time. Transitions can be started from the details panel (by buttons), or call them from blueprints.



Color, Metallic, Specular, Roughness, Emissive Color are material parameters. Can be set by users.

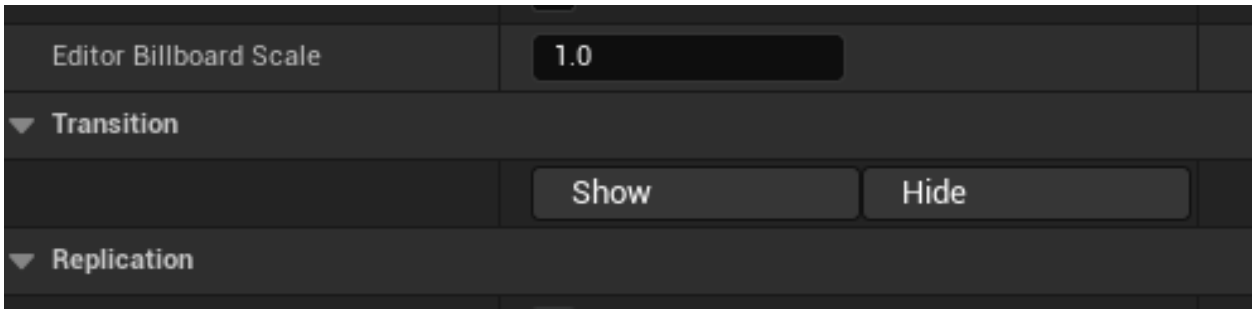
Transition types

Push *Wipe* *Fade* *Radial Fade* *Radial Wipe*



Primitive Actor

MtPrimitiveActor supports transitions. There are Show and Hide buttons in details panel, which execute show or hide transition on all its components (in case, the component supports primitive transitions)



When creating a new blueprint actor in blueprint editor, go to Class Settings and in the details menu, select Parent Class as MtPrimitiveActor. This enables usage of Shoiw/Hide buttons from details and usage of actor functions in blueprints.

Blueprints

PlayShow - starts show transition

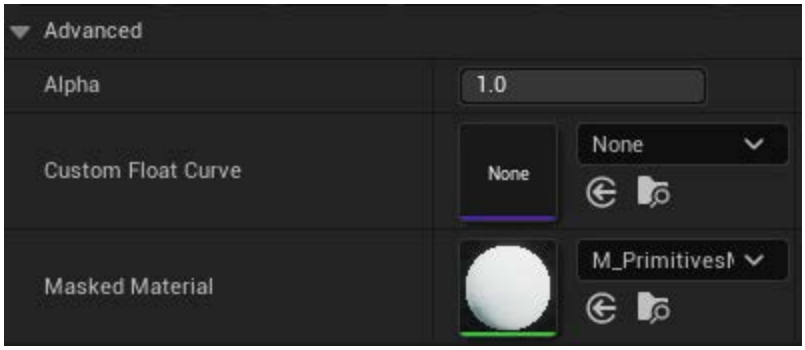
PlayHide - starts hide transition

PlayNext - starts transition according to current state, if object is shown, hide will be played, if object is hidden, show will be played. When next is called upon the actor, each component decides for itself, which transition will be played. (so if one component is hidden and second is shown, they will both play show/hide accordingly)

TakeShow, TakeHide, TakeNext - performs show/hide/next transition. This is latent action, so blueprint execution will be paused until transition finished. With an actor, execution is paused until all components have finished transitioning.

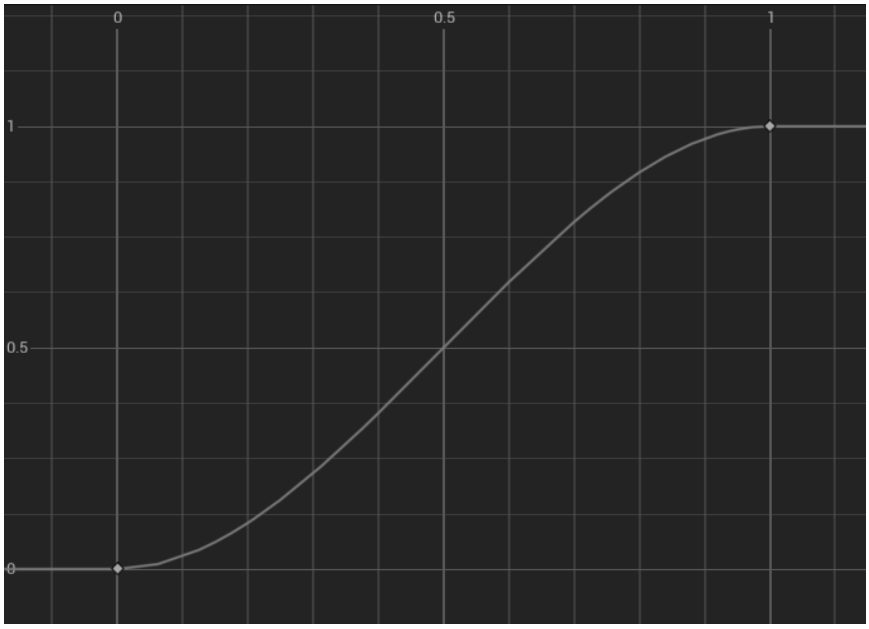
Advanced

In the advanced panel, users can change Alpha parameters. Alpha determines the stage of transition. Its values are in range 0-1, 0 for hidden and 1 for shown. Beware, that playing transition changes this parameter.

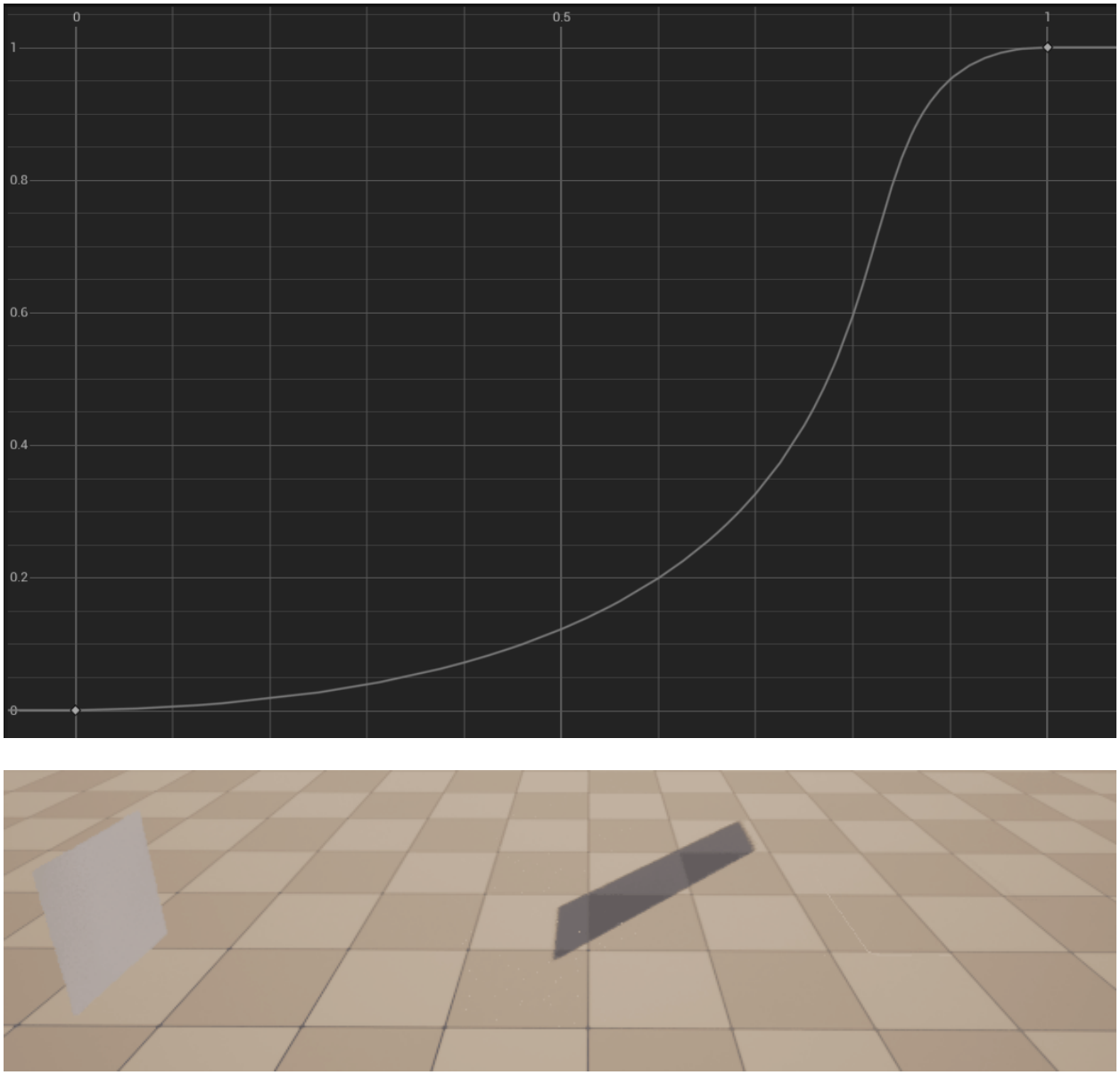


Custom Float Curve

User can specify own Custom Float Curve, changing speed of the transition. This curve should map 0-1 (time from start to finish - 0 is start and 1 is finished) to 0-1 (Alpha values - transition stage). That means, it should cross points [0,0] and [1,1]. For Show, left to right pass is made, for hide the direction is opposite (right to left).



Default Float Curve (when Custom Float Curve is not set - or set to None)



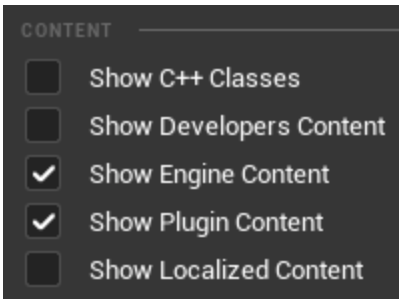
Custom float curve.

With this curve transition is visibly slower around hidden objects. Using custom curves with customizing transition duration gives users a lot of options for controlling transition speed.

Material Editing

Users can even edit material. For that, the user should copy the material asset M_PrimitivesMaterialMaskedWithTransitions.

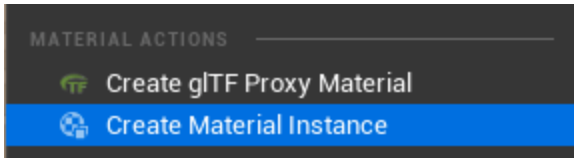
In the Content Browser in the right corner, in settings, check Show Engine Content and Show Plugin Content.



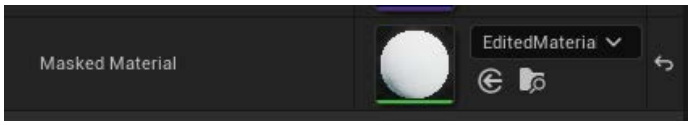
Material can be found in All -> Engine -> Plugins -> PRIME VSAR Content -> Primitives -> Materials. User should not under any circumstances edit this material (it will change it in Engine globally) directly, Please Right-click and Duplicate it.

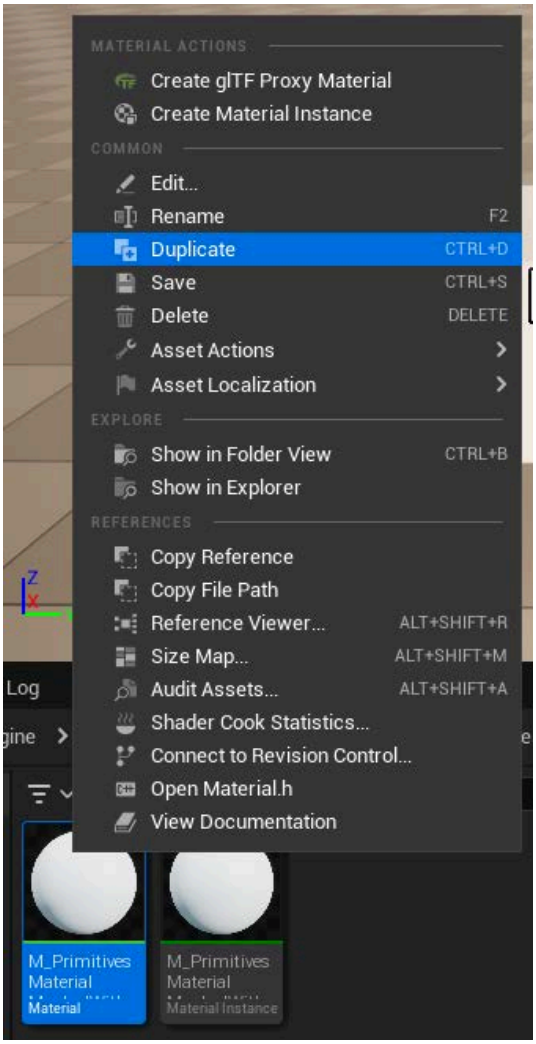
It is recommended to name duplicated material properly, so it would not be mistaken for the original material.

User can then do any changes on the copied material, then create material instance (Right-click edited material and Create Material Instance)



and then select it (the instance) as Masked Material in details. (drag and drop)





In order for transitions to work, users should not edit anything changing the opacity mask and world position offset. Adding textures or other material properties should be fine.

AR Reflection and Shadow

This section describes how to create planar reflection and shadow for objects visible in Augmented Reality, (i.e. rendered in the Foreground pass).

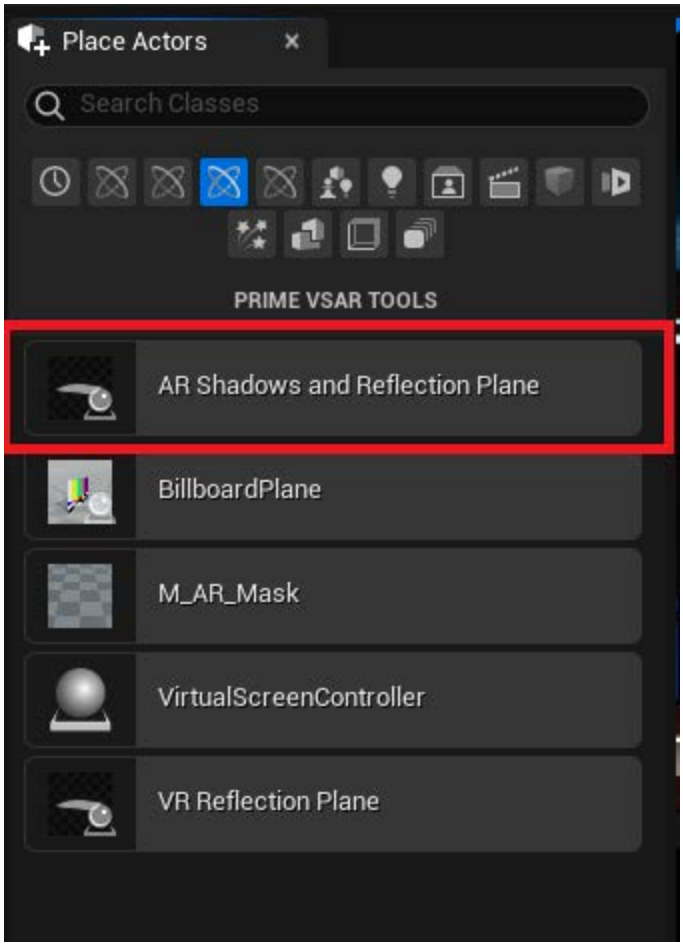
⚠ This section assumes that the level is already lit with dynamic lights, objects are in the Foreground pass and casting shadows.

The objects casting shadows should be set as Stationary or Movable.

For more information about lighting in Unreal Engine, please refer to unreal documentation - **Lighting Tools and Plugins**:

Setup

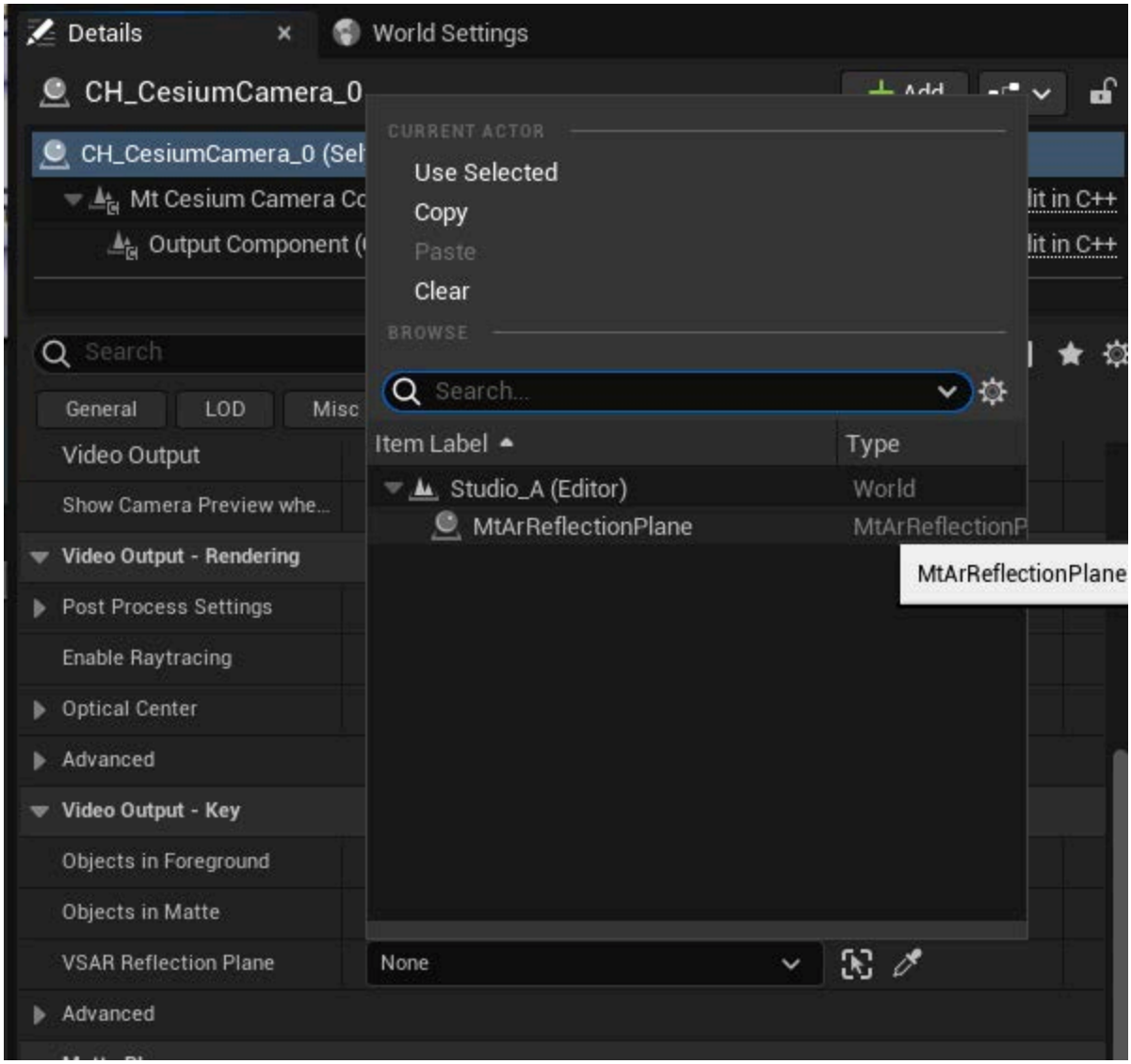
- Create an “AR Shadows and Reflection Plane” object, its purpose is to receive the reflection and shadow. Search for “AR Shadows and Reflection Plane” and drag and drop the object in the level.

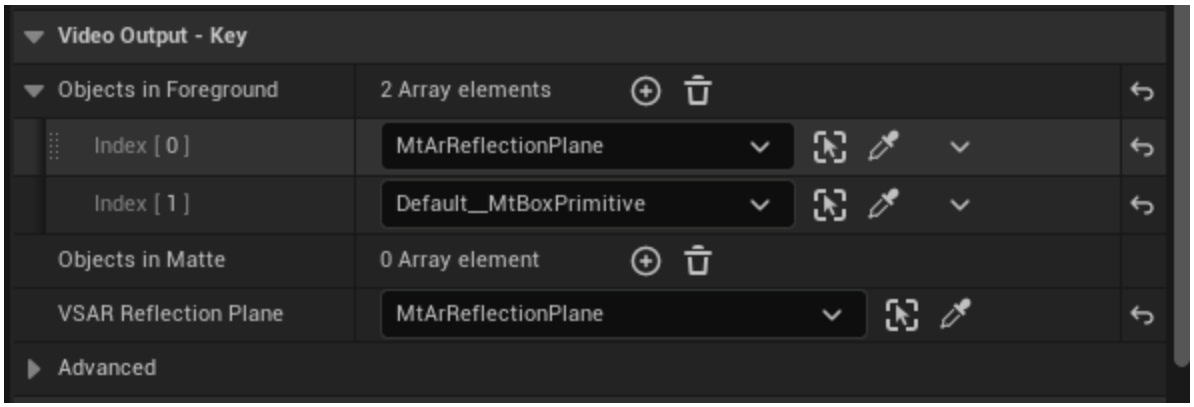


- Adjust the height (Z axis) of the “AR Shadows and Reflection Plane” in order to match the real floor’s height. The Target Displayer might be helpful to locate where the real floor is.

The following step should be done for each Cesium Camera where the effect is visible:

- Select the Cesium Camera (create one if it is not already done) and bind the ARPlane object via the Details Panel at the section Video Output Key:


Details
World Settings
CH_CesiumCamera_0
CH_CesiumCamera_0 (Sel
Mt Cesium Camera Co
Output Component (
CURRENT ACTOR
Use Selected
Copy
Paste
Clear
BROWSE
Search...
Item Label
Type
Studio_A (Editor)
World
MtArReflectionPlane
MtArReflectionP
MtArReflectionPlane
Search
General
LOD
Misc
Video Output
Show Camera Preview whe...
Video Output - Rendering
Post Process Settings
Enable Raytracing
Optical Center
Advanced
Video Output - Key
Objects in Foreground
Objects in Matte
VSAR Reflection Plane
None
Advanced

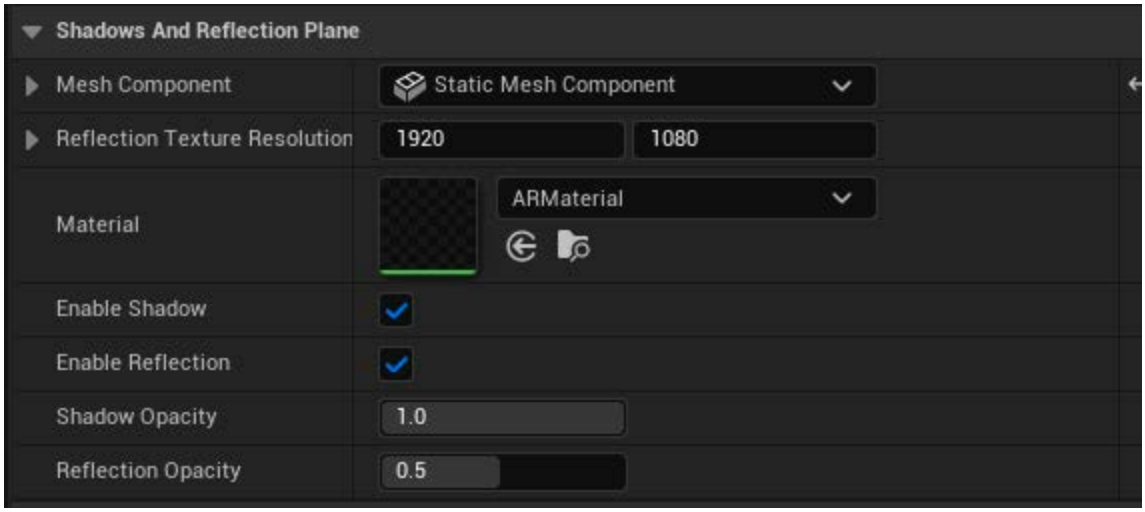

Video Output - Key
Objects in Foreground
2 Array elements
Index [0]
MtArReflectionPlane
Index [1]
Default__MtBoxPrimitive
Objects in Matte
0 Array element
VSAR Reflection Plane
MtArReflectionPlane
Advanced

This will automatically add the “AR Shadows and Reflection Plane” object to the list of Foreground objects.

📝 The reflection and shadow are not displayed according to the Editor viewport’s camera. In order to visualize the final result, check the Cesium camera output in the VSAR Config panel or the video output.

Parameters

ARPlane object has the following parameters to alter the rendering.



- Mesh Component: The default Static Mesh is a plane. This property is accessible in case of custom shape (e.g. reflection on a rounded surface).
- Reflection Texture Resolution: The resolution of the texture used for the reflection pass.

Default: 1920x1080

- Material: For advanced usage with custom Materials (e.g. blur, alpha mask...),
- Enable Shadow: Enable/Disable shadow, default value is Enabled.
- Enable Reflection: Enable/Disable reflection, default value is Enabled.
- Shadow Opacity: Affects the shadow opacity, 1 is fully opaque and 0 is fully transparent.

Default: 1

- Reflection Opacity: Affects the reflection opacity, 1 is fully opaque and 0 is fully transparent.

Default: 0.5

Troubleshooting

The shadow is not visible or disappears after building the lighting.

Verify that the following parameters are correctly set:

- The level is lit and the light is casting shadows.
- Objects casting shadows are set as Stationary or Movable.
- Shadow opacity is different from 0.
- The object is in the Foreground objects list.

AR Foreground Object Masking

About Foreground Object Masking

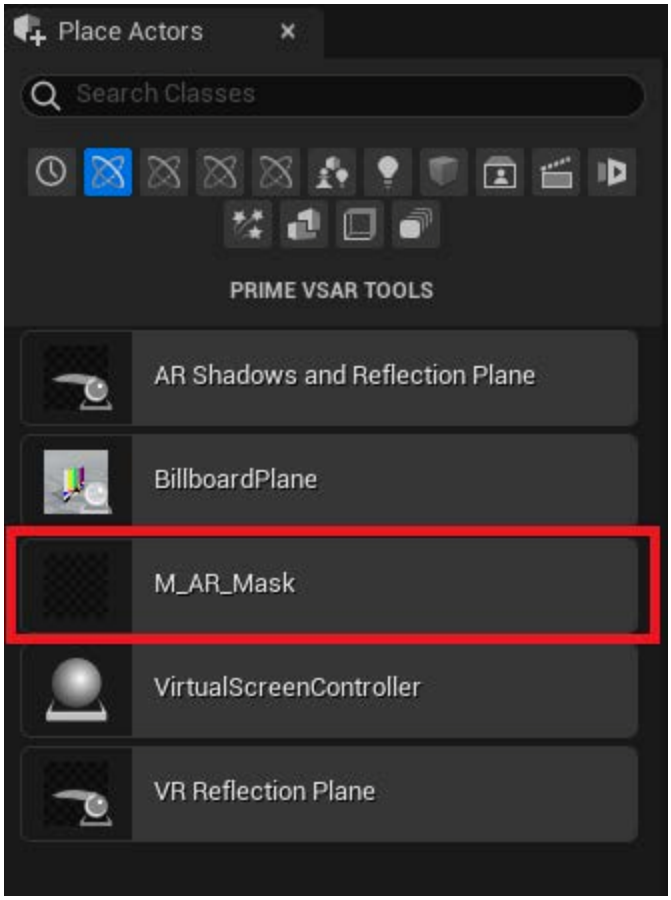
This section describes how to mask objects in the foreground in Augmented Reality (i.e. rendered in the Foreground pass), also known as Alpha Holdout. This is useful when part of a virtual object should be hidden by a real life object such as objects appearing out of the floor.

Setup

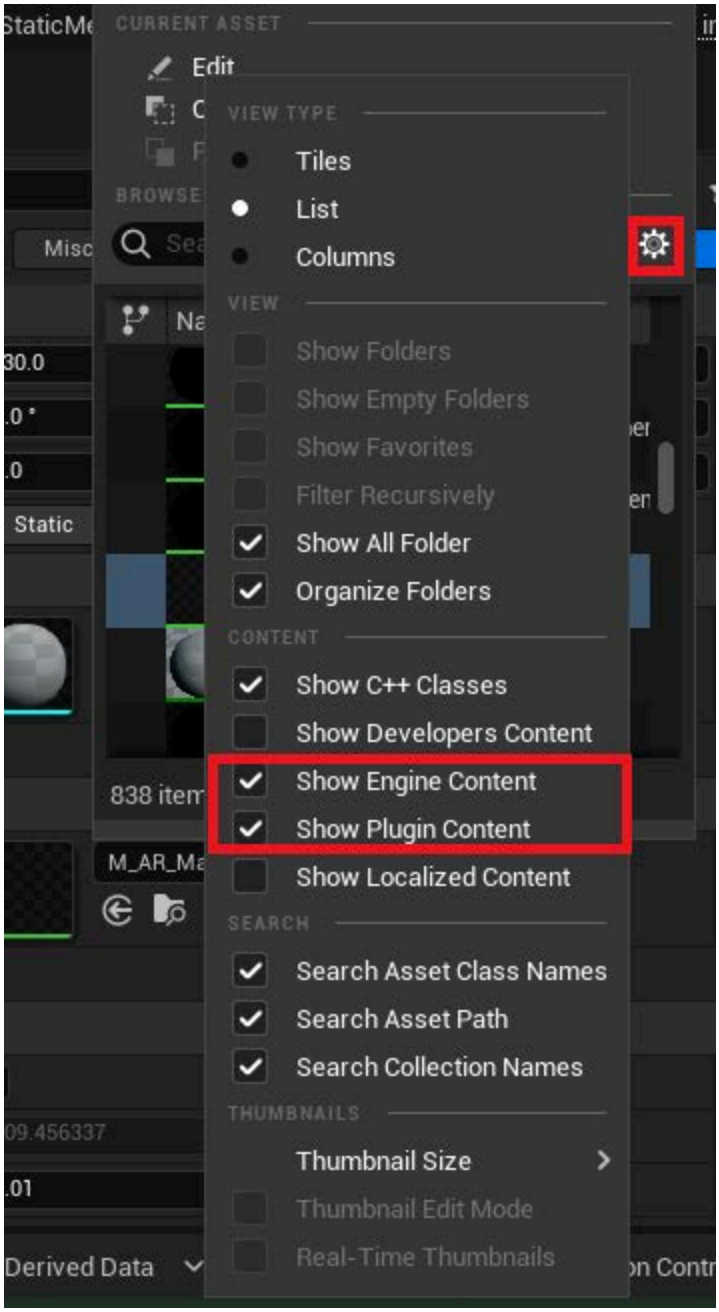
- For each Cesium Camera add the objects that should mask AR objects (e.g. the floor or a wall) to the list of Foreground objects. They should now appear in the key channel.

In this example we want the sphere to hide the cube. The cube and sphere are in the Foreground. The next step will set the sphere as mask.

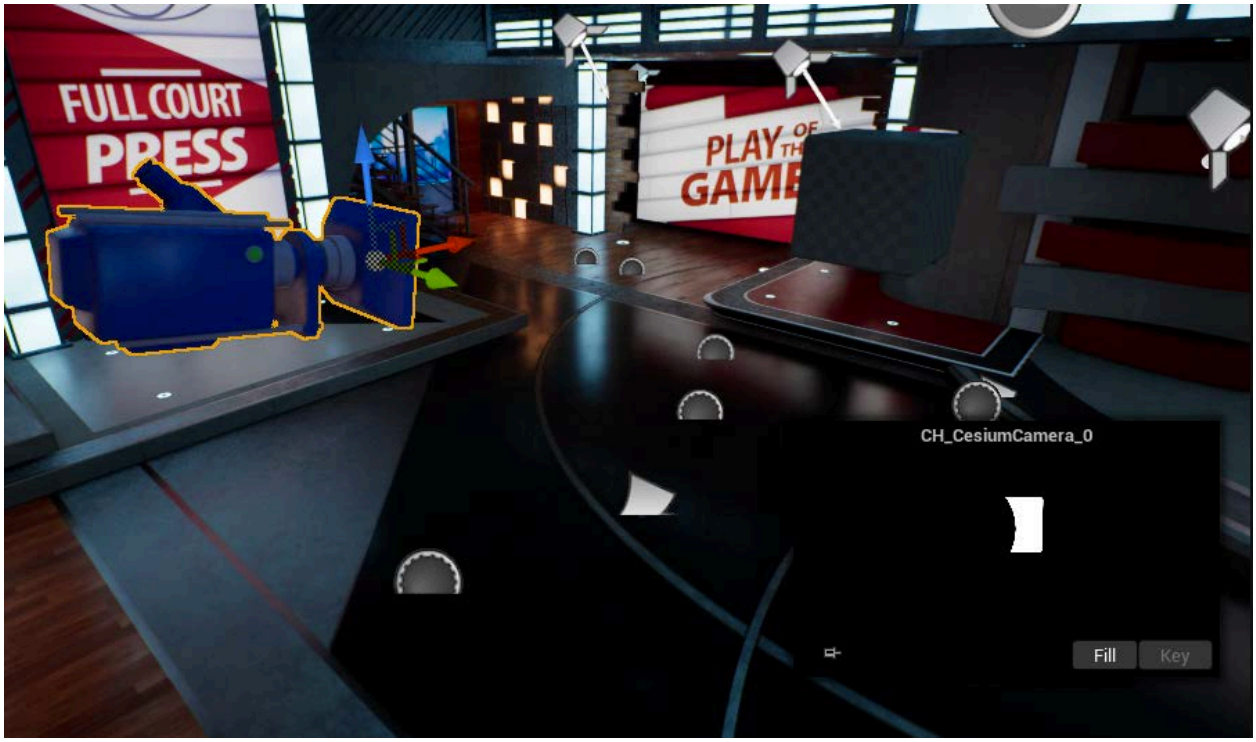
- Select the material “M_AR_Mask” in the “PRIME VSAR Tools” menu and drag and drop it on the masking objects.



⚠ In order to find “M_AR_Mask” when using the assets browser, “Show Engine Content” and “Show Plugin Content” must be enabled in “View Options”.



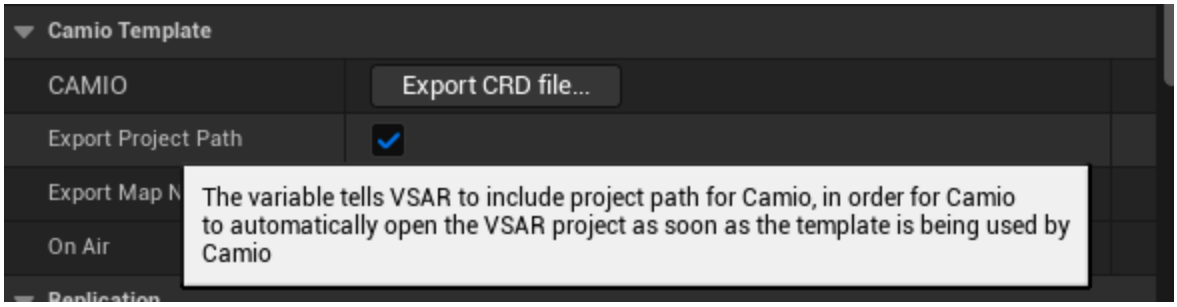
- The masking objects are now invisible on the Fill channel and masking in the Key channel (alpha holdout).



The sphere is invisible in the Fill channel and masking the cube in the Key channel.

Open Project Feature

When exporting CAMIO/LUCI .CRD template (available on every CAMIO Template blueprint), there is a default checkbox, checked in called Export Project Path. If this checkbox is checked, it tells VSAR to automatically open this project when CAMIO/LUCI tries to play the exported template.

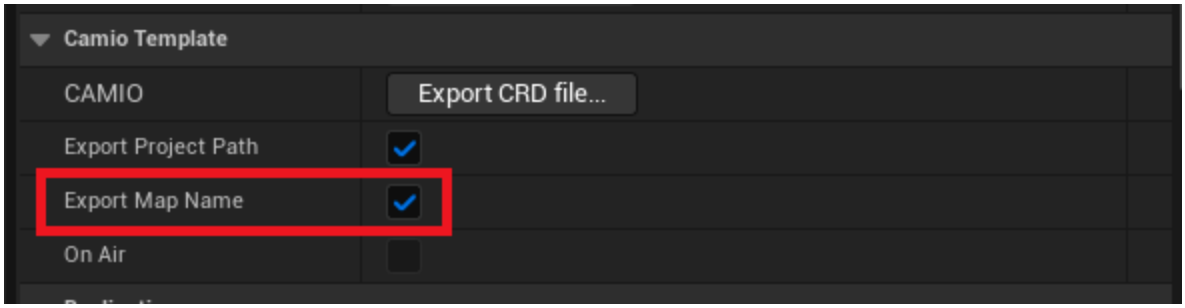


This functionality has been implemented in order to prevent confusion when having a template from a different project opened up in CAMIO and a different project in VSAR.

Part of this functionality is also that when you hit Render preview in CAMIO/LUCI, it automatically puts Unreal Editor - VSAR into play mode.

⚠ Path to project can not contain white spaces, Example of a path with space: “C:\Program Files\MyProject”.

Open project also saves map to the .CRD while the “Export Map Name” is checked



It exports the map name every time while exporting from instance in level but this is not the case while exporting from Content browser or Template.

Exporting from Content browser or Template does not export the map.

From PRIME VSAR version 1.8.2+ the currently opened map gets added if the corresponding asset has an instance in it.

Weather Template

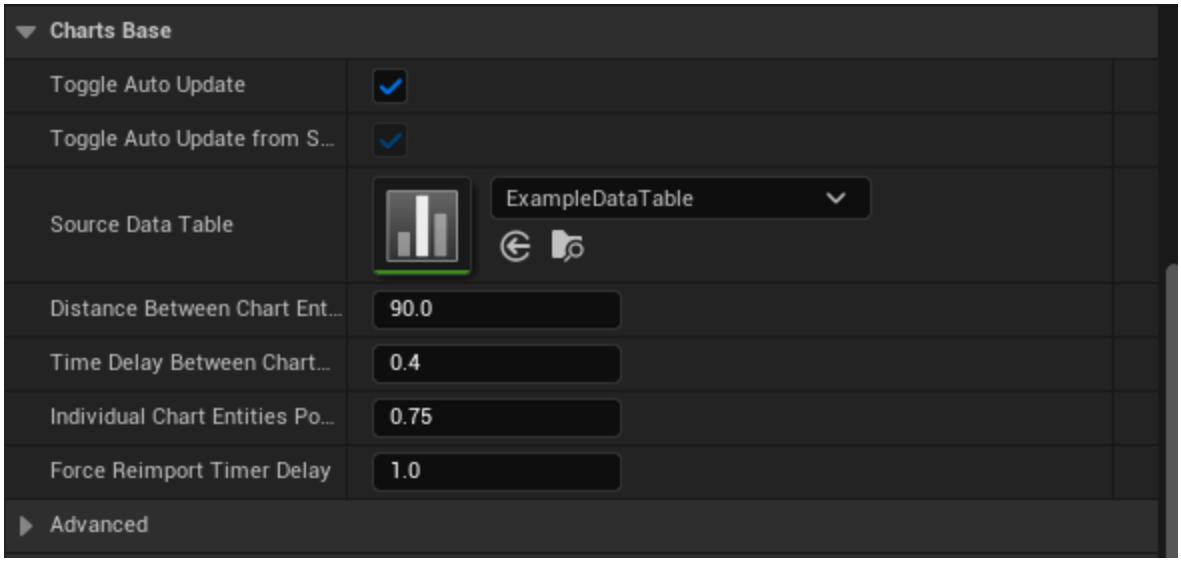
Setup

- Drag and drop weather template from PRIME VSAR Templates ->WeatherTemplate_7Days into the scene from Place Actors menu.

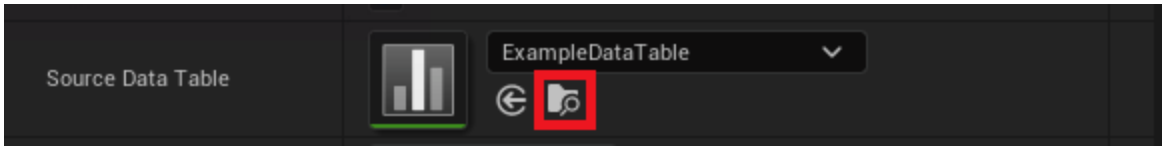
- This immediately creates weather pillars preview based on dummy data provided with the template.

⚠ using the dummy data DataTable will result in saving your values per VSAR installation. It is important to create new DataTable described in next step

- The source DataTable can be found next to the template itself. And inside of the blueprint itself in Details panels - Weather Template tab (Charts base in 1.8.1) - Weather Data.

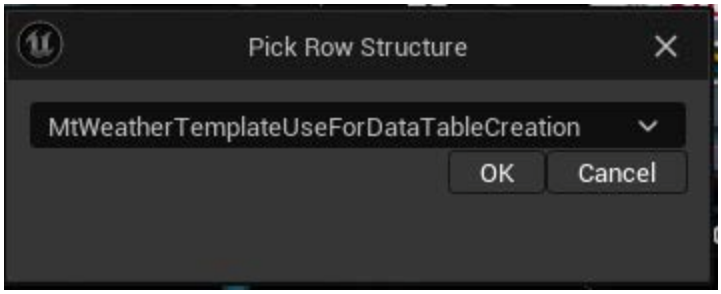


- To modify the data, locate the data table in the Content Browser, or simply by clicking on a magnifying glass next to it in the Detail panel.



This will bring you to DataTable in the Content Browser.

- To modify the data right-click the DataTable and choose Open in External Editor option.
- If this option is grayed out, re-export the table into .CSV.
    - Right-click the DataTable Export as .CSV - save it into the same folder as the original DataTable was saved.
        - To find out where it was saved:
            - Right-click the DataTable and choose - Open Source Location.
    - After reexporting the DataTable to .CSV - a prompt from unreal engine should appear, click on import.
        - After this you will get an option to automatically create a new DataTable from your newly exported .CSV file.
            - From a dropdown menu in the following table choose the following data structure as shown on the picture: MtWeatherTemplateUseForDataTableCreation.



This will create a new DataTable.

- Drag and drop it onto the WeatherTemplate - Weather Data Table - in the Detail panel inside the Weather Template tab, shown on the first image in this section.
- After this, you can put the editor into play mode.
- To raise up the pillars in play mode simply press the “+” button on Numpad or “-” to lower the weather template down.
- While the Weather template is raised up, you can modify the .CSV source file which serves as a source for the Template and the weather template should update accordingly.
- **Please note that you have to keep the format of the .CSV file as is (Adding any other rows or columns will cause undefined behavior.).**
    - Do not change the first column - days description while in playmode.
    - You can change the temperatures at will.
    - You can change the weather description to be one of the following weather types:
        - Sunny
        - Mostly_Sunny
        - Rainy
        - Snowy
        - Mostly_Cloudy
        - Heavy_Rain
        - Thunder
        - Haily
    - These weather descriptions have to be written precisely as presented here, otherwise, any other weather type will cause undefined behavior.

CAMIO/LUCI Use

To use the template with CAMIO/LUCI.

- Put the editor into playmode.

Note that the template in default settings is set to not to rise on itself.

To export a CAMIO template, find the Weather Template in the scene, either in play mode inside of World Outliner Window, or outside of Play in Editor simply by clicking on the Template gizmo in the scene, which looks like a white ball. After that:

- While having the template selected in the scene or in Content Browser - In CAMIO Template Tab, in the Detail panel locate the Export CRD file… button and press it.
- This will export all the necessary data into a .CRD file importable to CAMIO/LUCI.
- Import this template into CAMIO/LUCI.
- From LUCI, you can change the data, and as soon as the Unreal Engine is in play mode - click the Render Preview Button in LUCI. VSAR will change the data to desired ones - from LUCI and will render a preview for LUCI.

Barchart Template

Displays values visualized as 3D bars

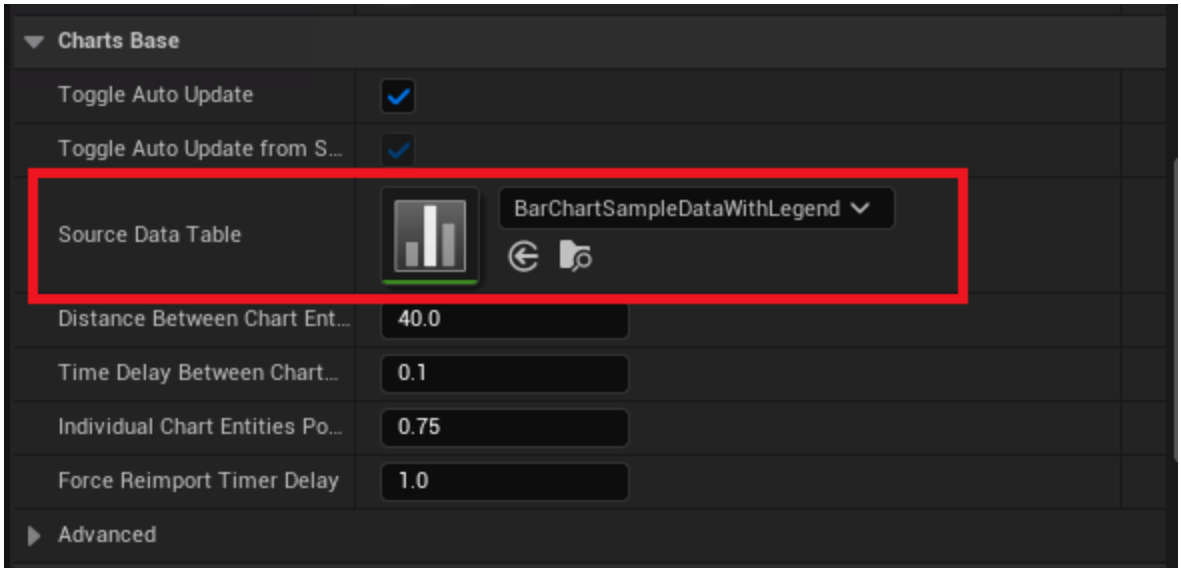
Setup

- Drag and drop weather template from PRIME VSAR Templates -> BarChartTemplate into the scene from Place Actors menu.

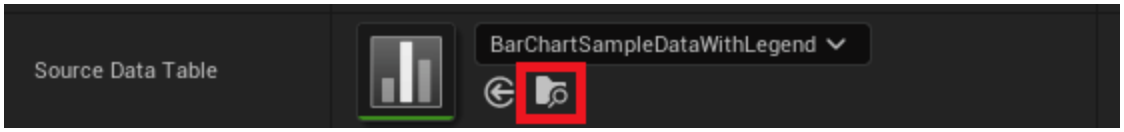
- This immediately creates weather pillars preview based on dummy data provided with the template.

  ⚠ using the dummy data DataTable will result in saving your values per VSAR installation. It is important to create a new DataTable described in the next step.

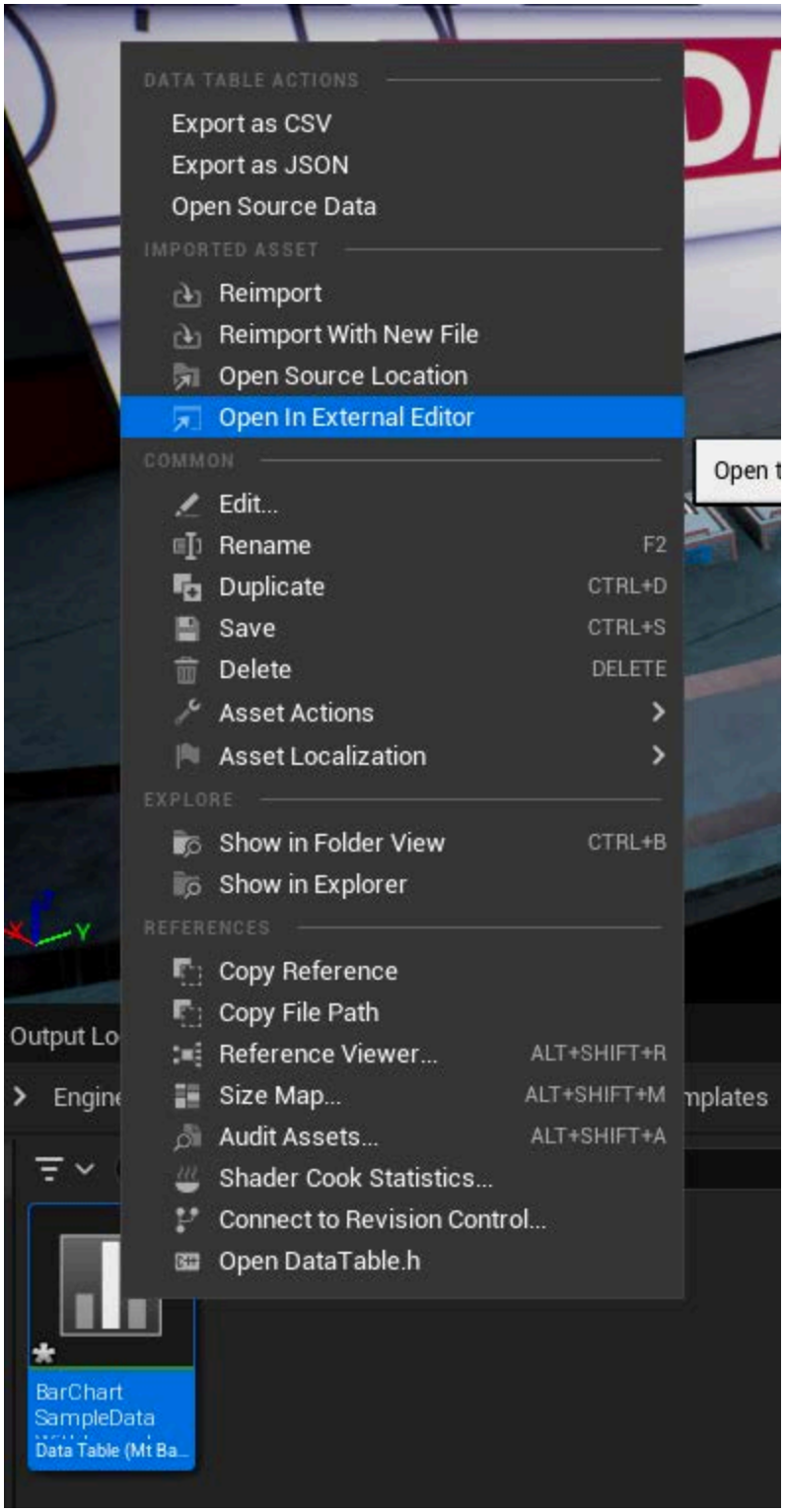
- The source DataTable can be found inside of the blueprint Details panels - Charts base section - Source Data Table.



- To modify the data, locate the data table in the Content Browser, or simply by clicking on a magnifying glass next to it in the Detail panel.

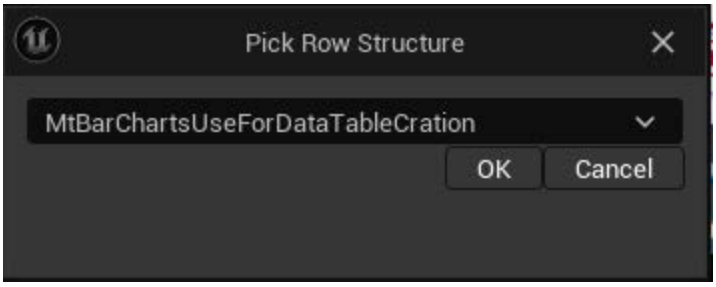


- To modify the data right-click the DataTable and choose Open in External Editor option.



- If this option is grayed out, re-export the table into .CSV or to create a new DataTable.

- Right-click the DataTable Export as .CSV - save it into the same folder as the original DataTable was saved.
    - To find out where it was saved:
        - Right-click the DataTable and choose - Open Source Location.
- After reexporting the DataTable to .CSV Drag and drop the .CSV from file explorer (folder) to content browser (to the same folder with the DataTable or to a different content folder to create a new DataTable).
- make sure to select correct DataTable Row Type shown below (if this window appears):



- Drag and drop the DataTable from content browser back to the source DataTable that can be found inside of the blueprint Details panels - Charts base section - Source Data Table

- After this, you can put the editor into play mode.
- To raise up the pillars in play mode simply press the “+” button on Numpad or “-” to lower the weather template down.
- While the barchart template is raised up, you can modify the .CSV source file which serves as a source for the Template and the weather template should update accordingly.
- Adding/Removing rows in .CSV adds or removes the number of bars

Parameters

- Size to bar scale value - When active bars get scaled to Bar scale value (to the value below) so that the highest value is scaled to the Bar scale value as if it was the highest value. Example: highest value is 10 and Bar scale value is 150, children bars will be scaled 15 times to get the highest value to the desired height.

- Bar scale value - Value in unreal units (cm by default)
- Bar minimum value - Restricts the minimum value of the bar (after scale), if the value of the bar is below this value it will be bound to the minimum value. Example: bar value is 0 and minimum value is 14 then the bar height will be as if the bar would have the value of 14.

CAMIO/LUCI Use

To use the template with CAMIO/LUCI, refer to the chapter **CAMIO/LUCI use** for Weather Template as the use is very similar.

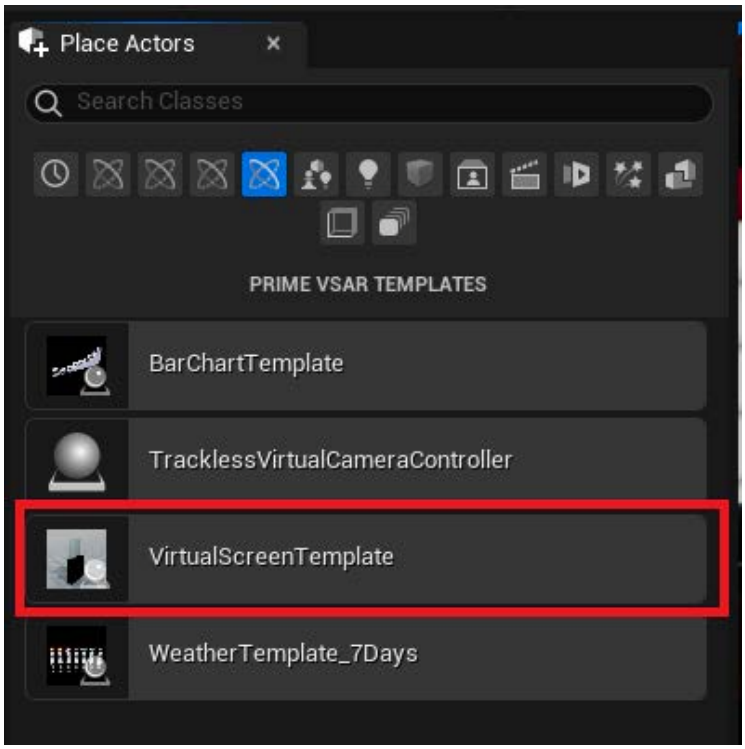
Exported template in LUCI supports up to 15 bars

A showcase of this functionality can be seen in this video for **LUCI** and for **CAMIO/ENPS**.

Virtual Screen Template

Setup

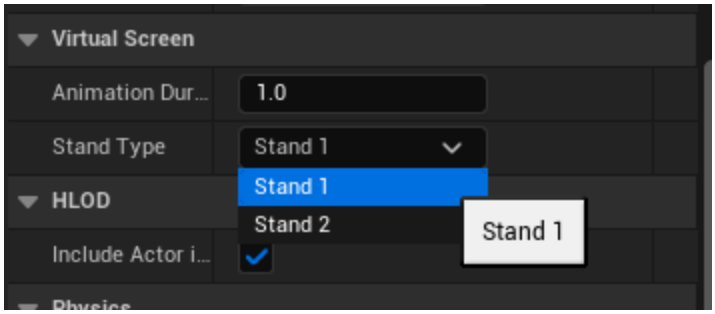
In the category PRIME VSAR Templates, drag and drop the Blueprint VirtualScreenTemplate.



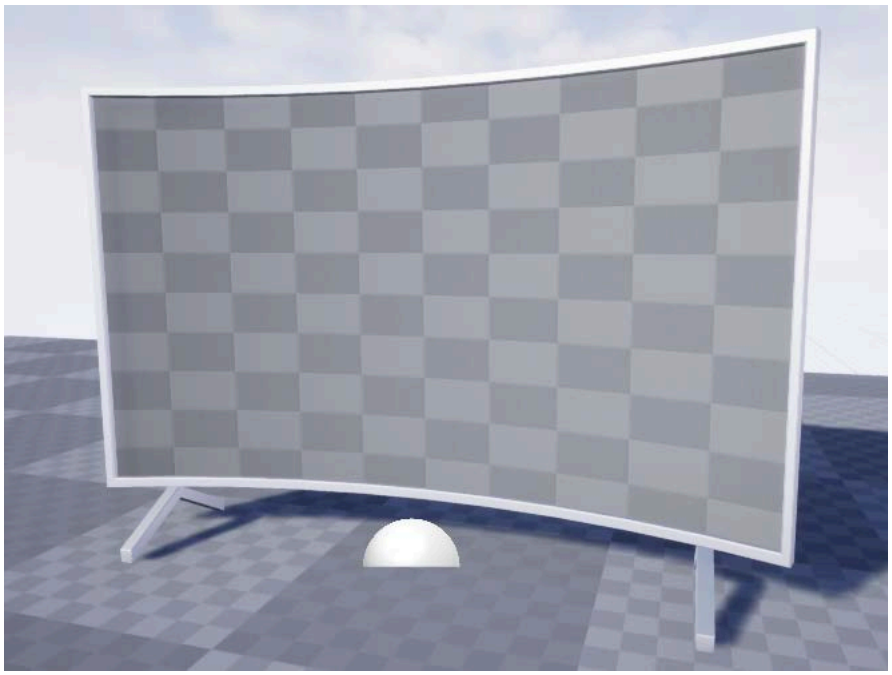
PRIME VSAR Templates

Parameters

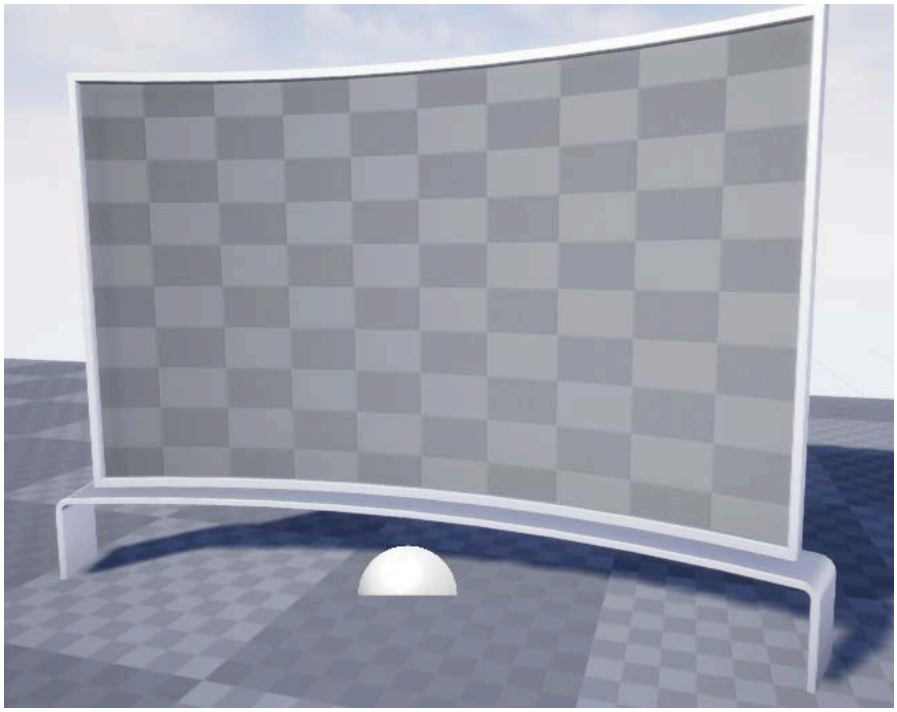
In the Details panel, the virtual screen stand asset visual is customizable:



Stand 1 and 2 selection

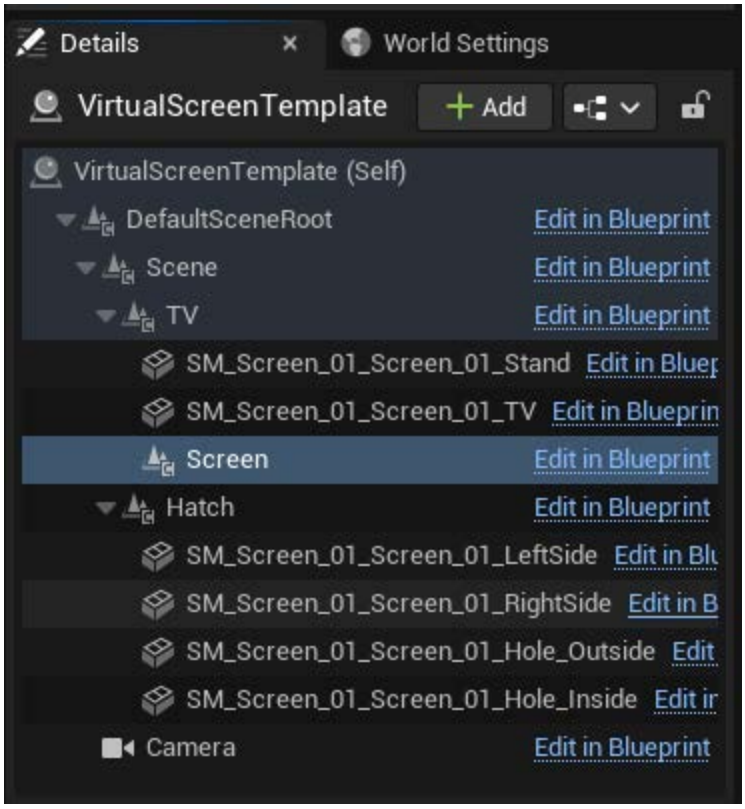


Stand 1



Stand 2

The template is based on the AB Switch component, select the component named “Screen” for more customization options:

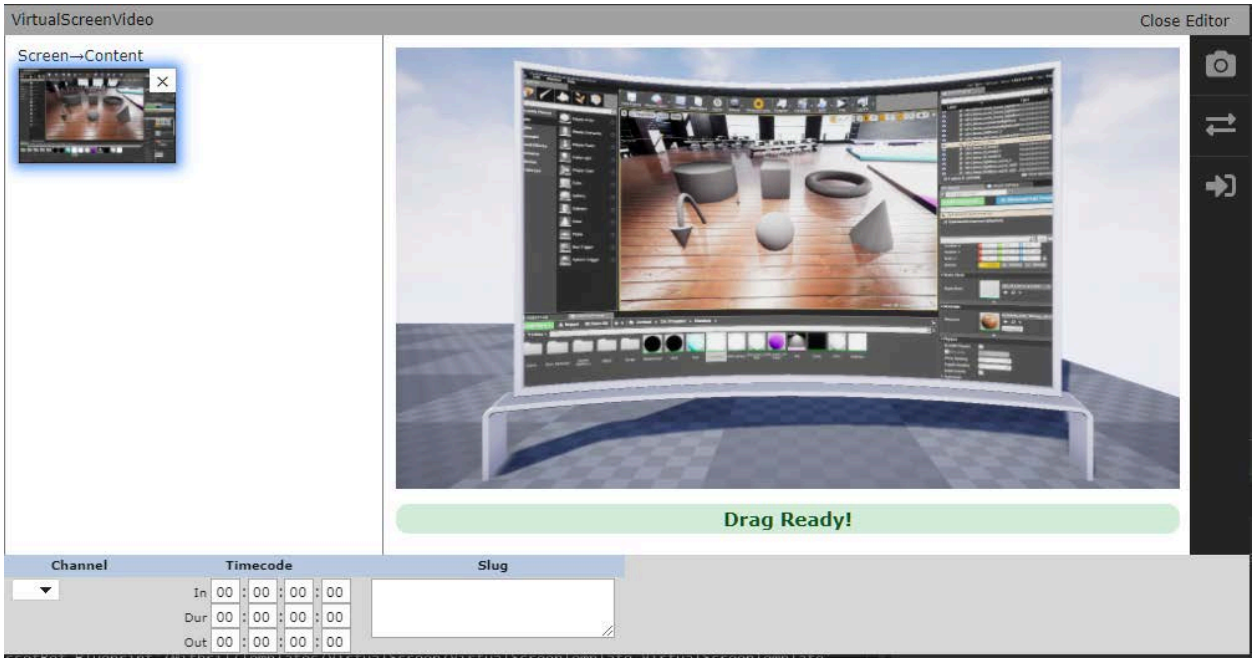


The AB Switch component is named “Screen”

Please refer to the section **AB Switch - Effects** for more information about the AB Switch parameters.

Usage

The template is driven by the CAMIO workflow:

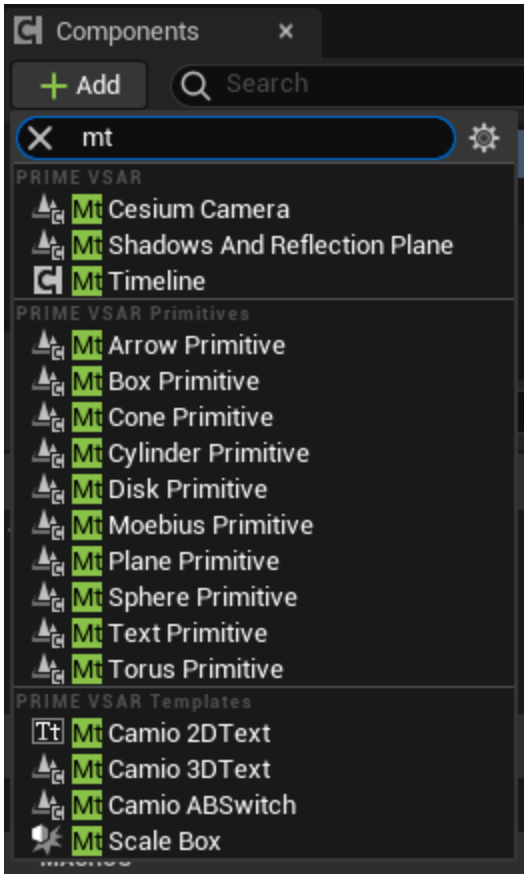


LUCI 5 Preview

Video tutorial showing the workflow with PRIME VSAR, CAMIO/iSQ and ENPS.

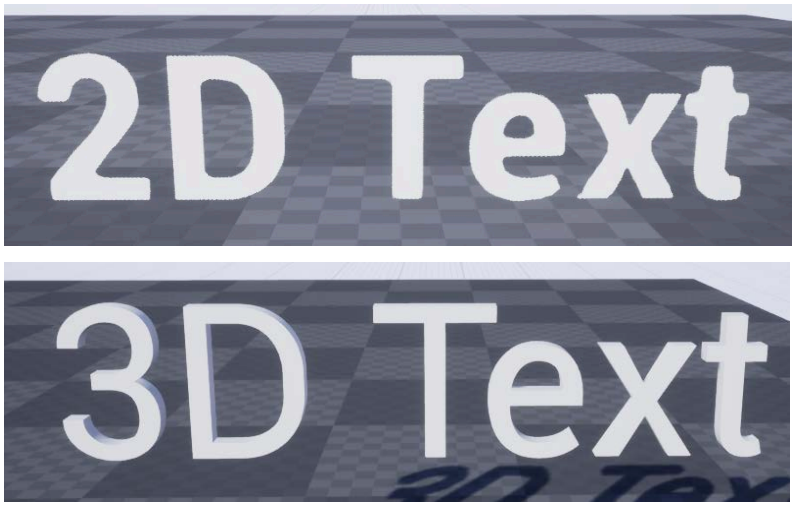
Meta Templates

Components that are aimed to make custom template creation easier. They can be added with the Add Component button in the blueprint editor.



Camio 2D/3D Text Component

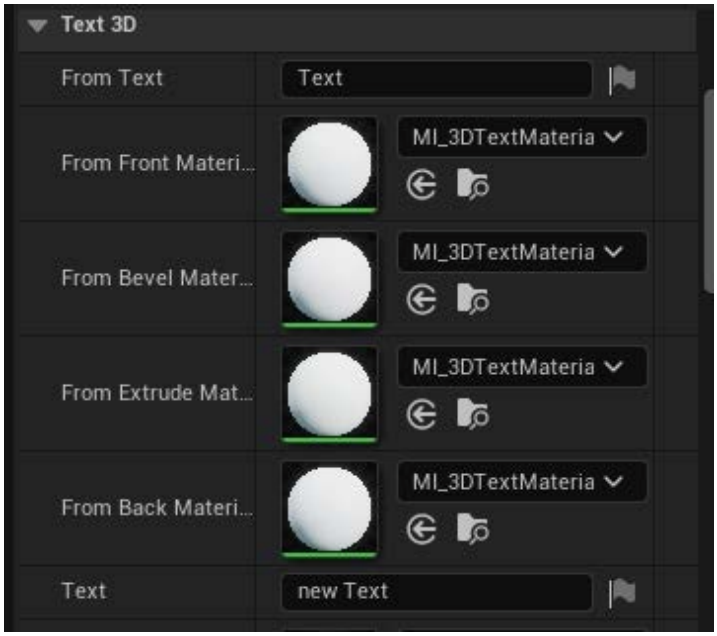
“CAMIO 2D Text Component” is based on Text Render Component and “CAMIO 3D Text Component” is based on 3D Text Component, Both can be placed into 3D world, but the 2D Text Component has a flat billboard like look to it. Both components can transition between two texts and have built-in support for CAMIO workflow.

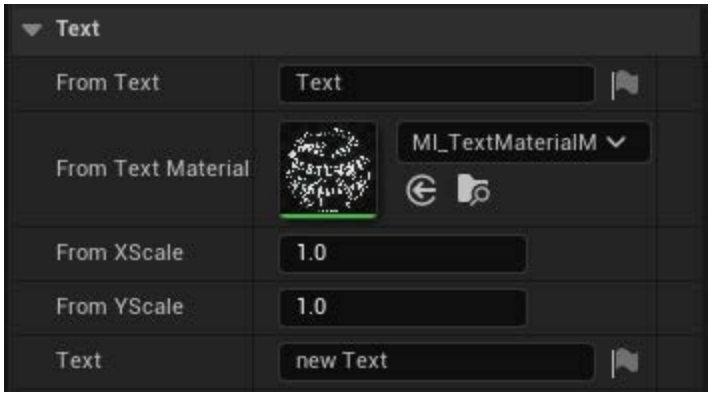


Parameters

Text Inputs

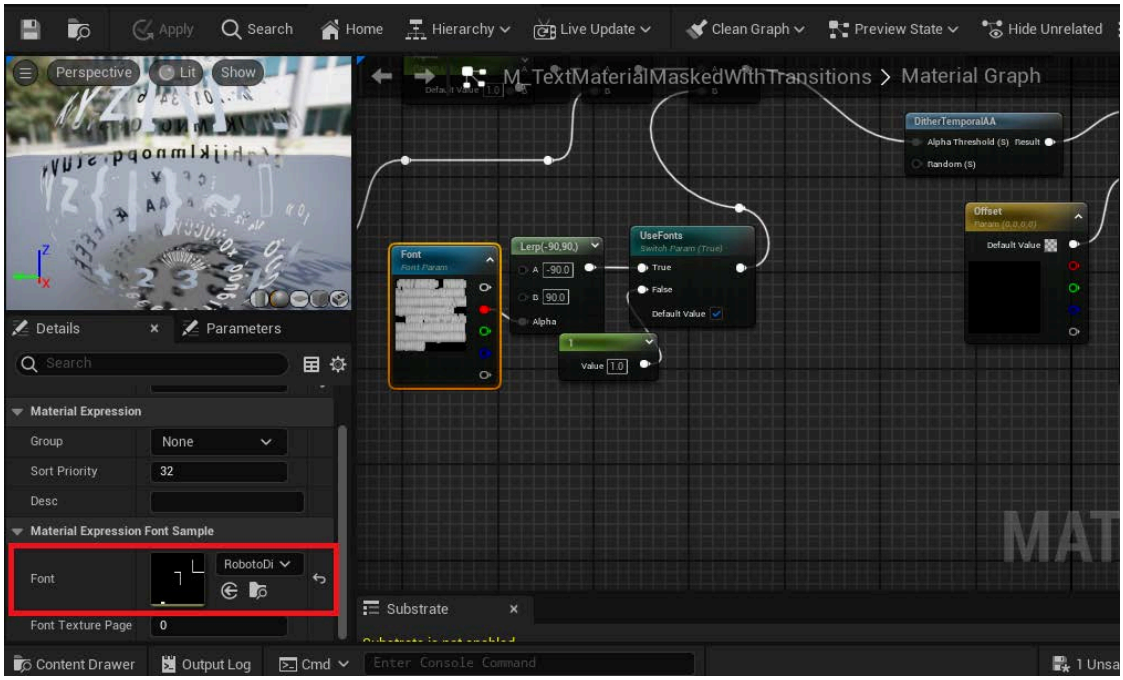
In the Text (3D) section in the details panel text components have two text inputs: From Text that is the current text if the transition didn’t start yet and Text (To Text) that appears after transition finishes.





Fonts

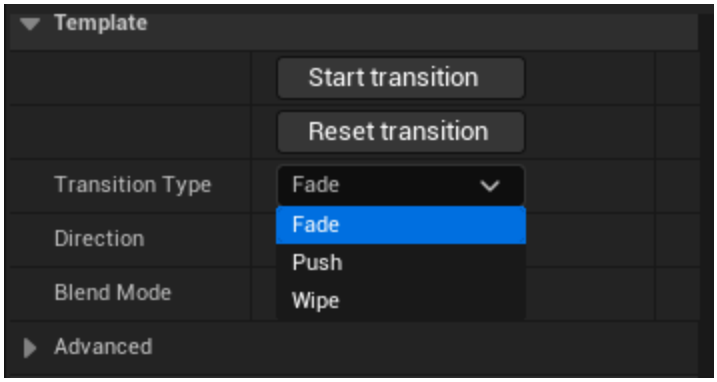
CAMIO 2D Text Component supports fonts only in Offline mode with Use Distance Field Alpha turned on, if there is need to include a foreign characters for this font, these characters need to be included in the font's Characters array ahead of production. Changing what Characters array contains can be changed by setting Unicode Range in Import Options of the font for example: "0001-017F" including unicode ranges for desired character symbols, then reimporting font to apply these settings. Duplicate Material in "Mithril Content/Templates/Text2D" (you can find it by clicking on the magnifying glass next to the material) and move the copy to your content folder, Edit the material copy and change the corresponding font in the material inside parameter Font. Assign the new font material to the CAMIO 2D Text Component with custom font.



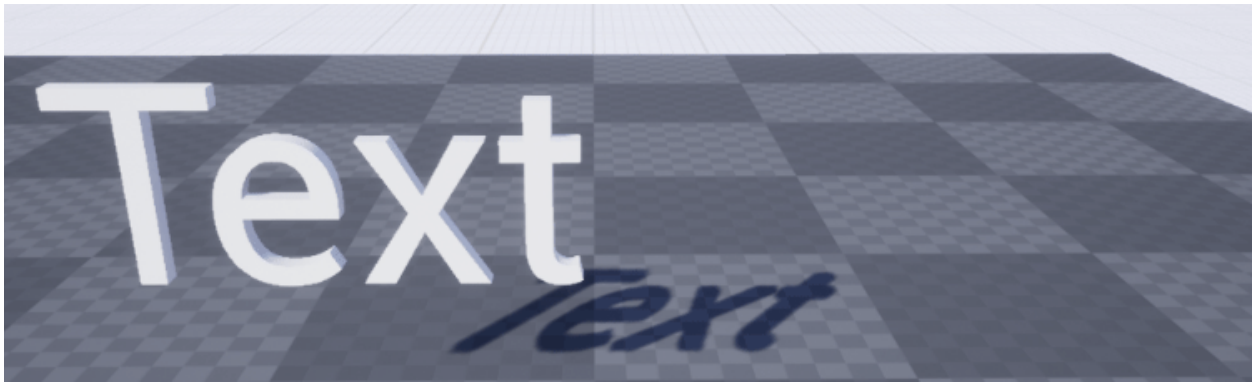
CAMIO 3D Text Component supports fonts only in Runtime mode, so there is no need to include foreign characters but these characters need to be present in the font file.

Transition Type

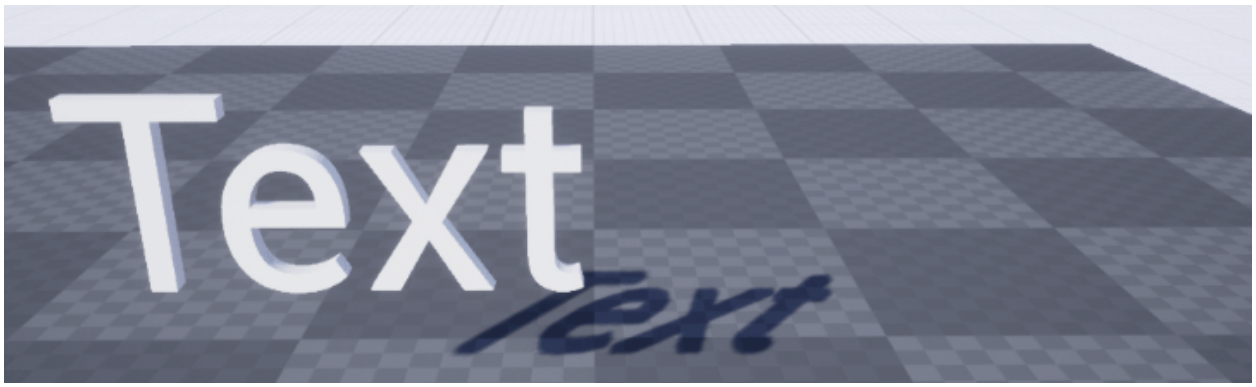
Transition Type as well as transition Direction can be changed under the Template section in the details panel.



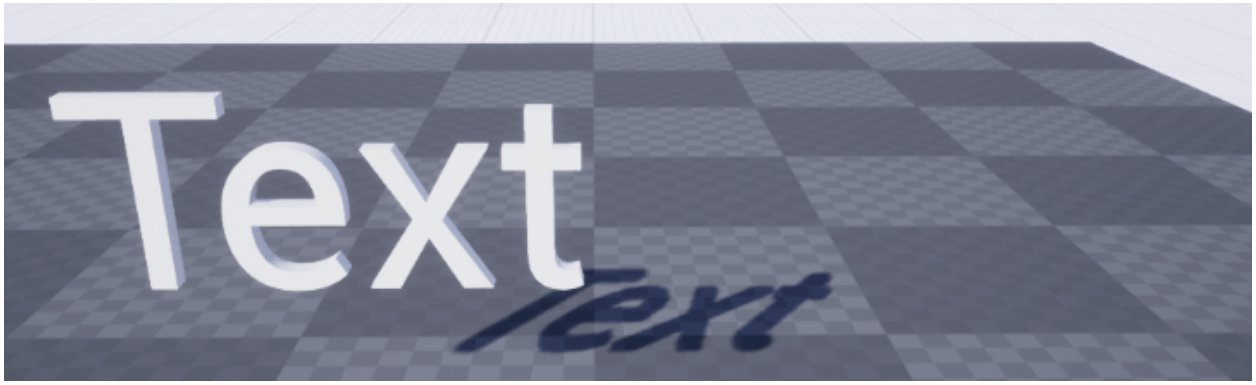
Premade transition types are: Fade - a simple dissolve transition, Push where the From Text moves away and the To Text comes in to replace it and Wipe where sweeping line changes From Text to the To Text along this line.



Fade



Push

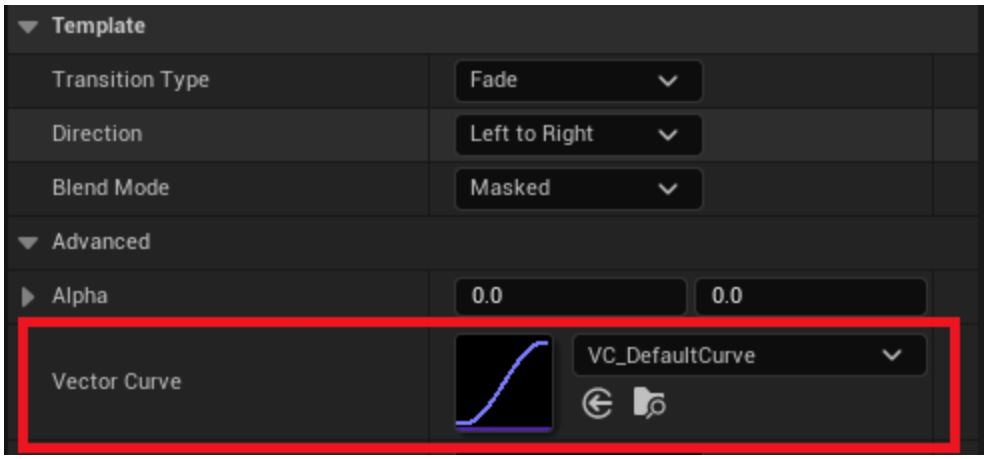


Wipe

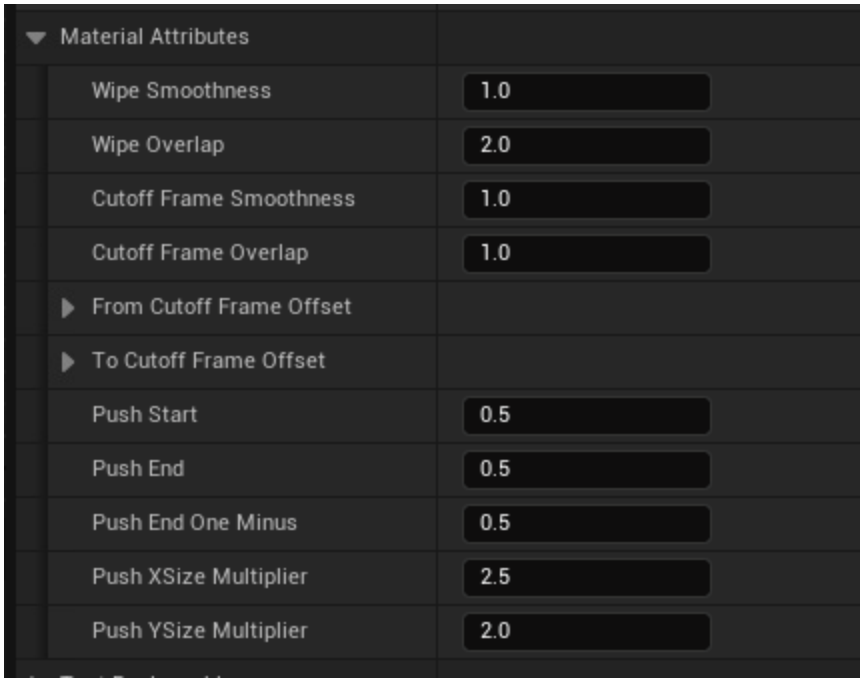
Advanced

The duration and smoothness of the transition can be changed with a Vector Curve asset, X axis represents the From Text transition and Y axis Represents the Text (To Text) transition (Z is unused).

A custom Vector Curve asset reference can be set in the Vector Curve parameter under the Template section (Advanced section) in the details panel.



Transitions can be further customized with Material Attributes parameters:

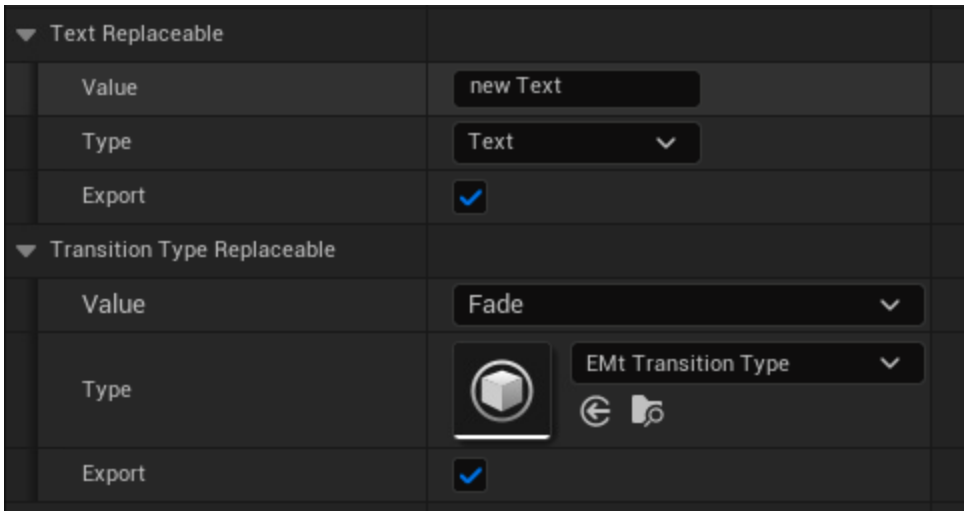


Blueprint functions

Setting text through “SetFromText” and “SetToText” is highly recommended. Setting variables directly or using the function “SetText” from the parent component may result in the text visual not being properly displayed.

Usage

Texts include automatically pre-populated CAMIO replaceables prepared for the CAMIO workflow. They can be found under the Template section (Advanced section) in the details panel.



If you do not wish to export default replaceables to use your custom replaceables instead, you can do so by unchecking the Export bool value.

There is a **video tutorial showing how to work with texts and use them in the CAMIO workflow**.

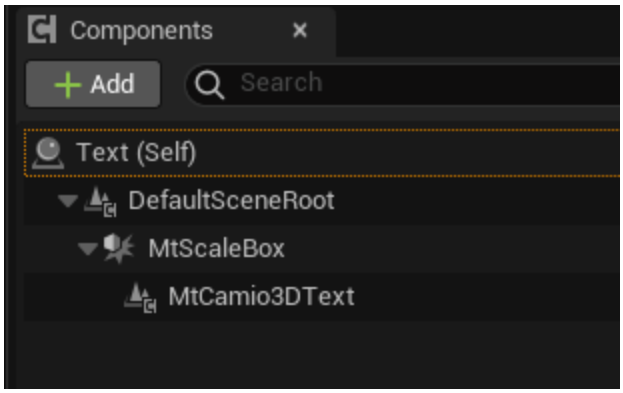
Scale Box

About Scale Box

A Scale Box is a component that scales attached content depending on the scale type selected.

Scale Box Setup

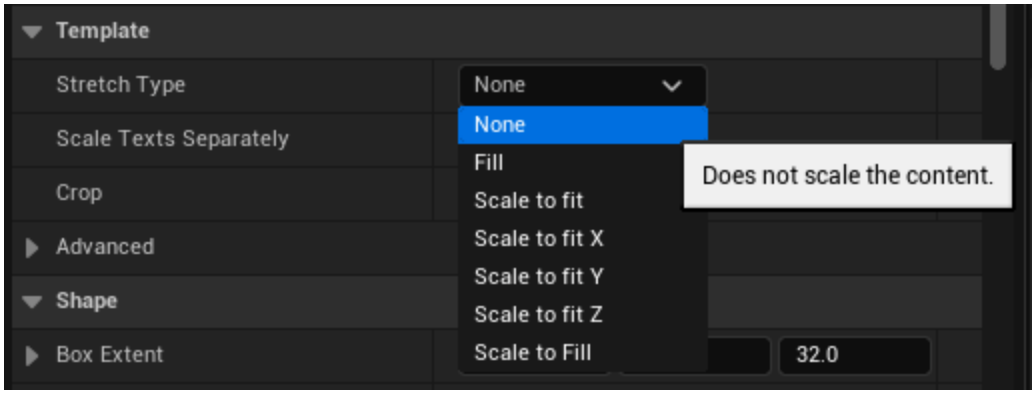
- Add Scale Box by Add Component button.
- Drag and drop desired component to be scaled on the Scale Box Component to be attached to it as a child, resulting in this hierarchy:



Parameters

Stretch Type

Scale is determined by Stretch Type that can be changed under the Template section in detail panel.



Predefined scratch types are: None doesn't scale, Fill Scales to fill out the box non-uniformly, Scale to fit scales uniformly to fit inside the box, Scale to fit X/Y/Z scales uniformly to fit into the box's specified dimension and Scale to Fill scales uniformly to the largest box dimension.

Scale Texts Separately

Makes text sizes the same by stretching smaller text non-uniformly. This also applies for multiline texts.

Crop

If enabled, overflow out of the box is cut for supported components: CAMIO 2D Text Component, CAMIO 3D Text Component.

Usage

Here is a video tutorial explaining how to work with scale box.

⚠ When using ScaleBox, any translation, rotation, and additional scaling should be done to the ScaleBox and not its children components as this may result in unpredictable behavior.

📝 **MtTextPrimitive** is currently not supported by Scale Box.

Internal Chroma Keyer

Composure Compositing

Internal Chroma Key Overview

Unreal provides two methods for internal chroma keying:

- A single pass chroma key material.
- A multi pass chroma key with the plugin Composure.

The recommended method for Prime VSAR is the chroma key based on Composure.

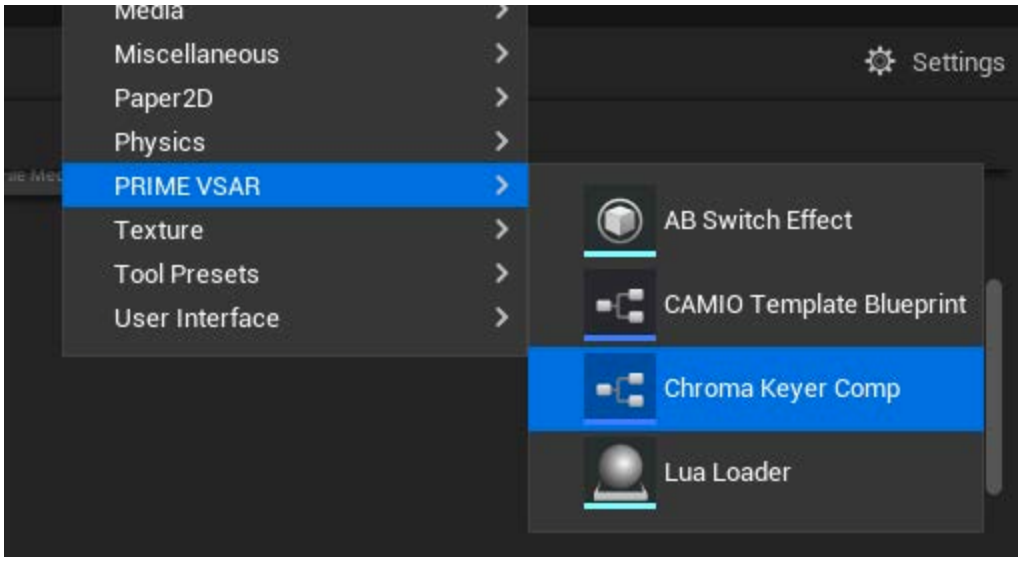
For more information about Composure take a look into **Unreal Engine documentation**.

For more information about the **Chroma Key algorithm developed by Unreal**.

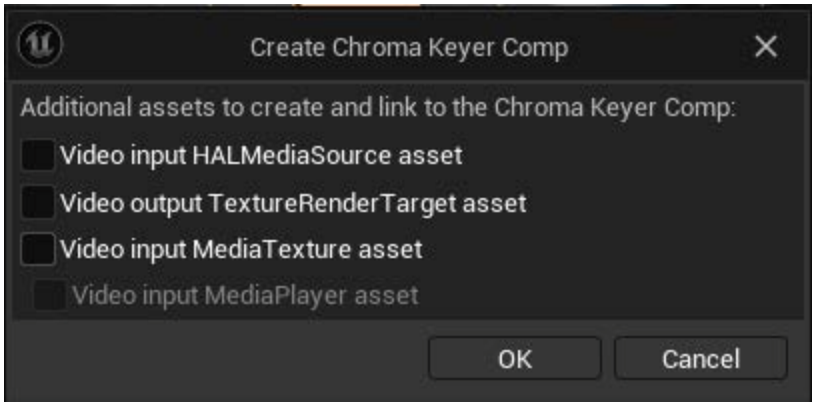
Setup

There is a dedicated UI in order to create all the necessary assets needed.

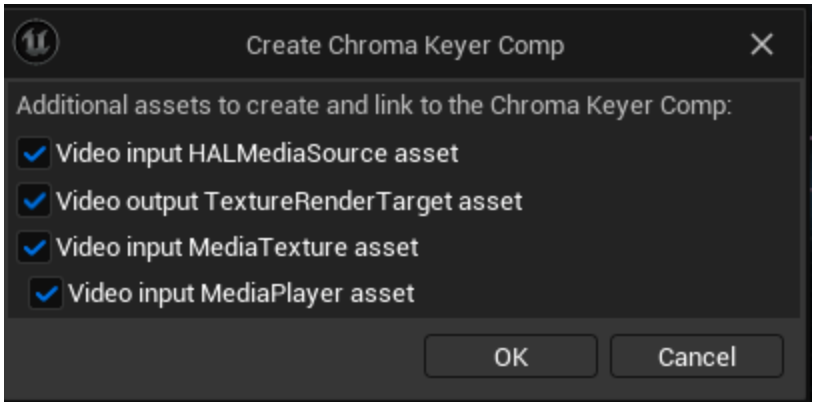
In the Content Browser, right-click to show the contextual menu, go to “PRIME VSAR” category and select “Chroma Keyer Comp”.

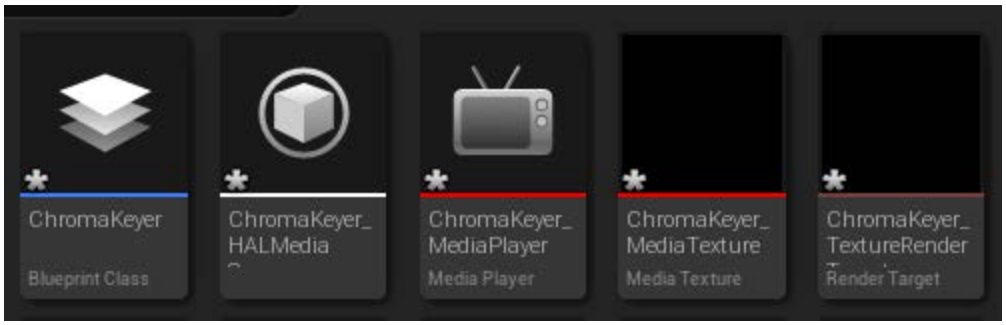


You can optionally select assets to create along and link them to the comp.



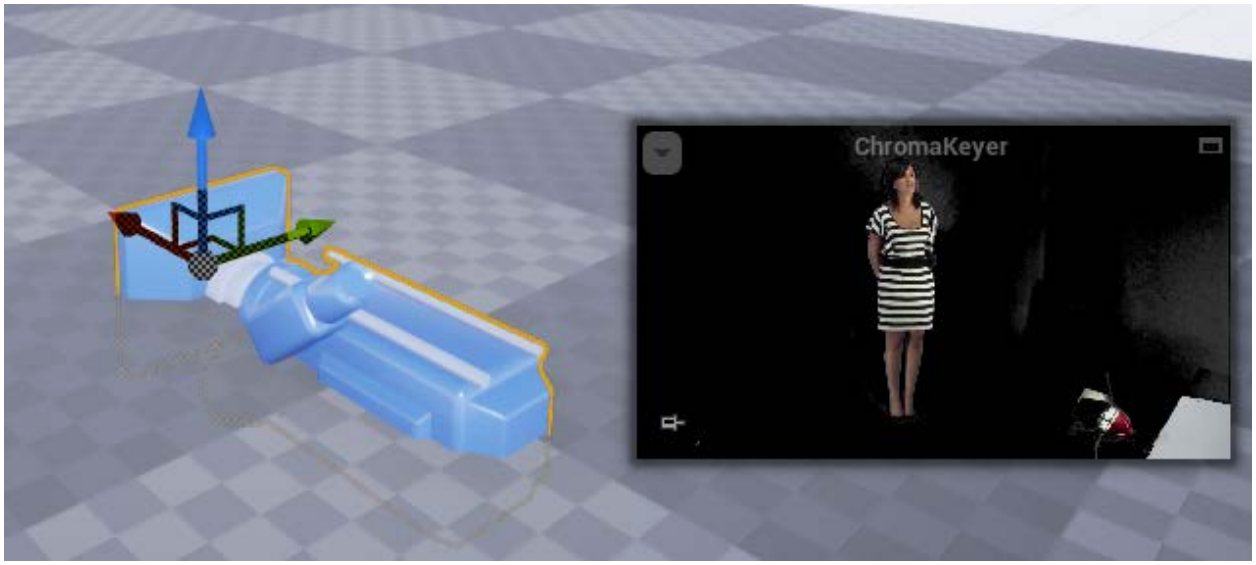
If it is the first Chroma Keyer Comp created, It is recommended to check all options in order to also create and link all necessary assets. The MediaPlayer will also automatically play the MediaSource in this case.

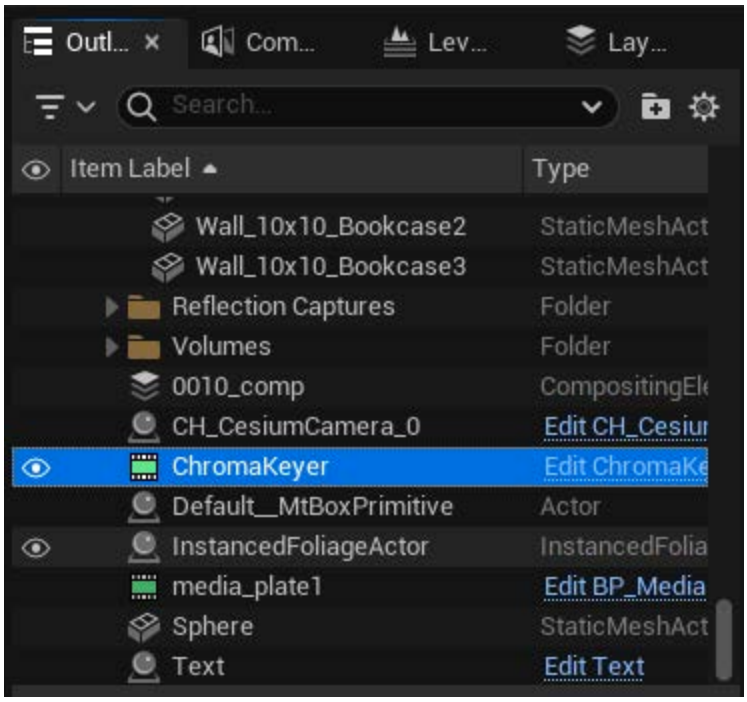




Assets created when all options are checked, the main asset named “ChromaKeyer”.

Drag and drop the asset “ChromaKeyer” in the level.

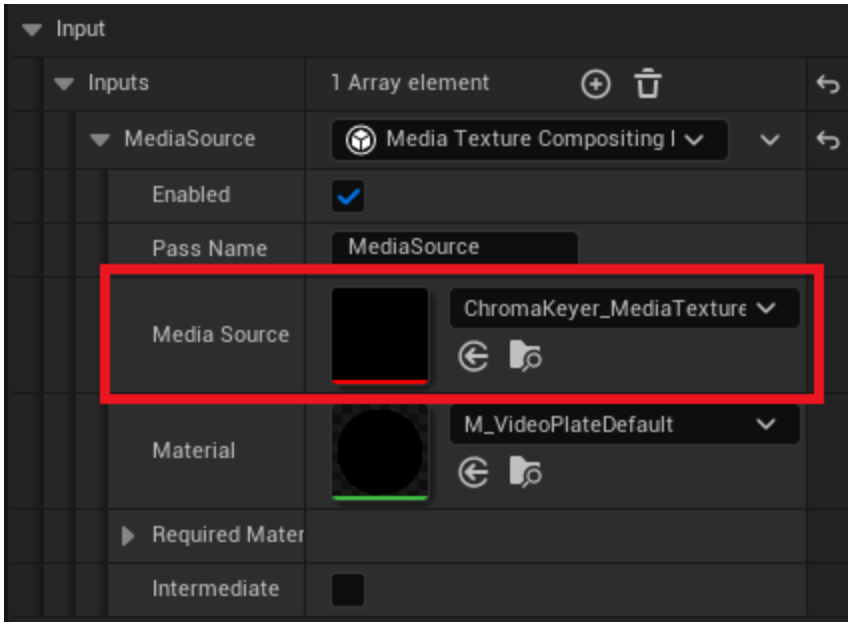




Parameters

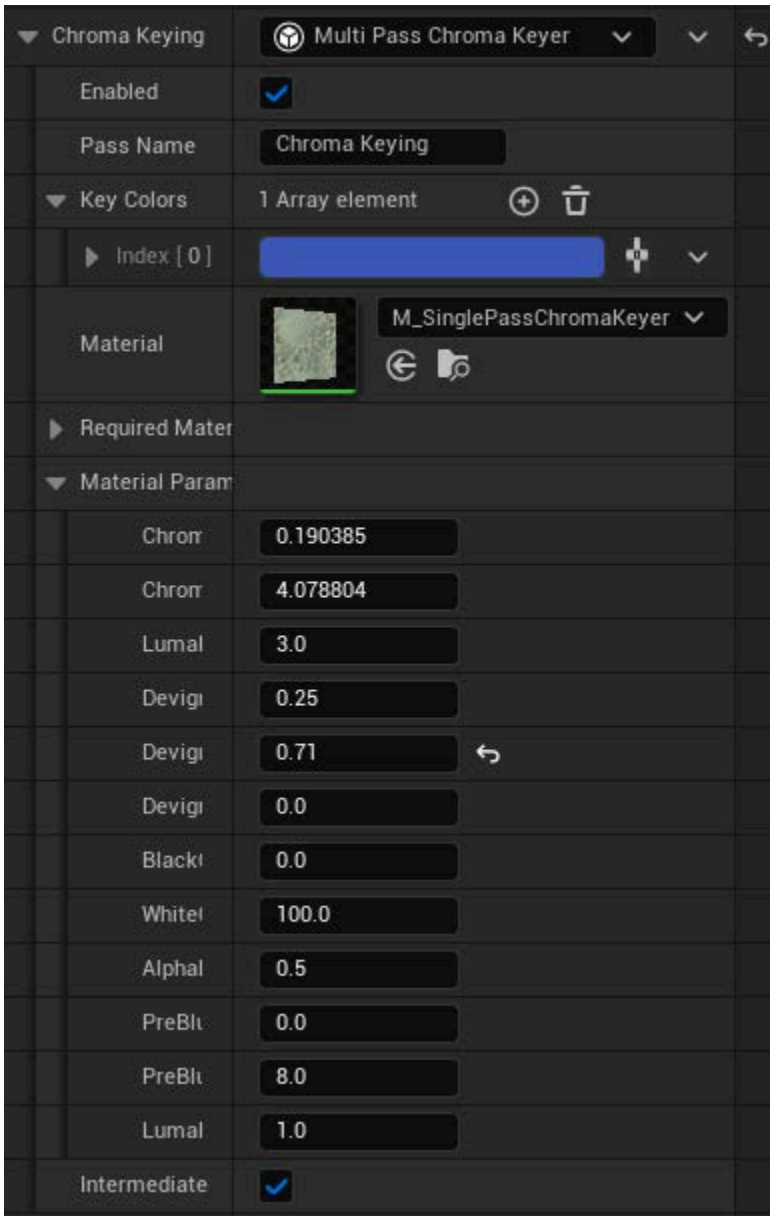
Input

Select a Media Texture as the input (it is already set if the option to create a Media Texture was checked previously).



Then go under Composure > Transform/Compositing Passes > Transform Passes.

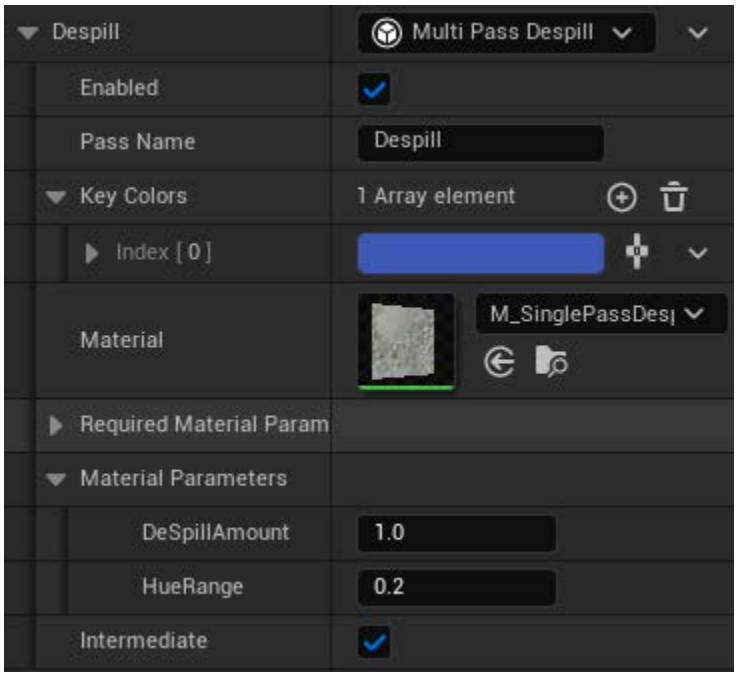
Multi Pass Chroma Keyer



Select the key colors with the color picker .

Multiple key colors can be added but be aware that each color picked adds an additional rendering pass.

Despill



Select the key colors with the color picker  .

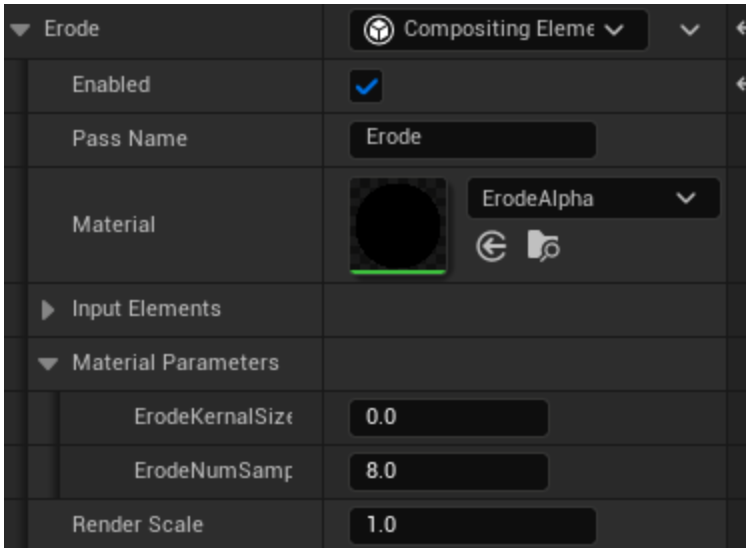
Multiple key colors can be added but be aware that each color picked adds an additional rendering pass and thus increasing demands on performance.

Despill off

Despill on

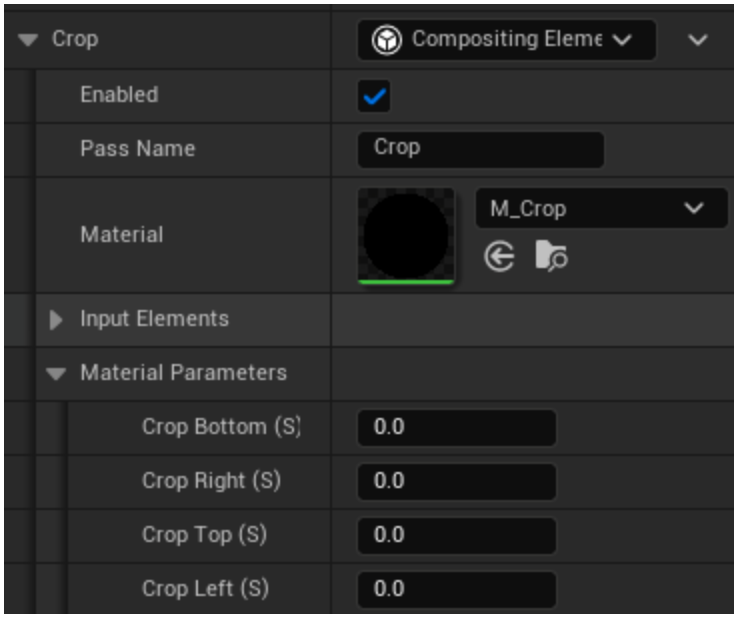
Erode

Erodes outline of the keyed subject, useful for removing spilled color on hair outline.



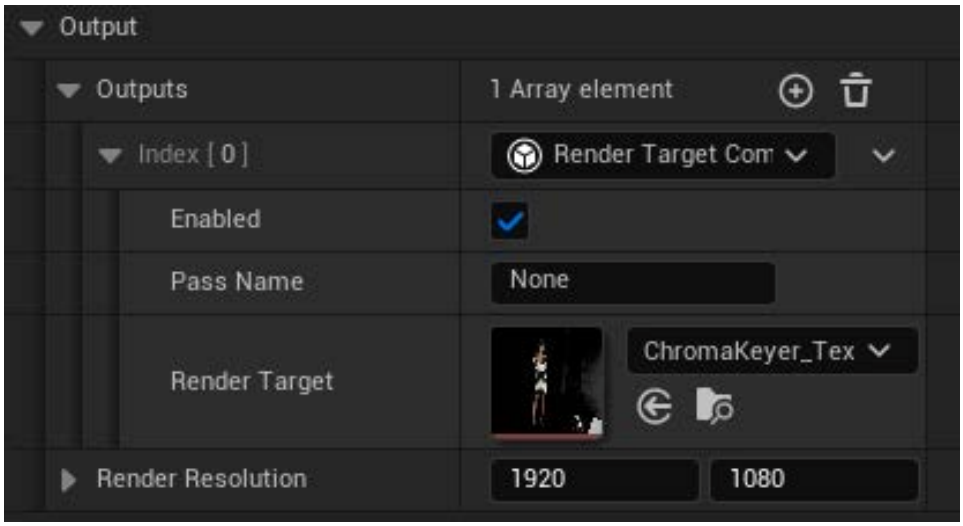
Crop

Normalized crop from each side in range [0-1]



The alpha channel can be cropped as a garbage matte.

Output



The output resolution is adjustable, it should match the input's resolution (FHD by default).

Billboard for Chroma Key

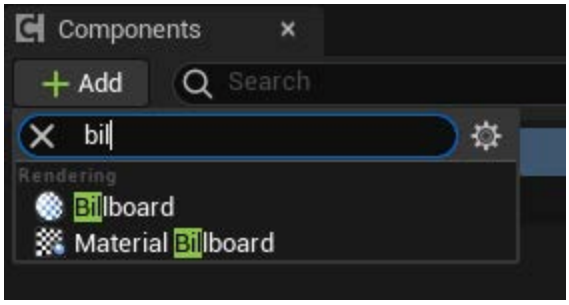
The result of the chroma key is rendered to a Target Texture. This texture can be used in Material to apply on geometries. For the Trackless solution, we are displaying the texture on a billboard (i.e. a plane always facing the camera).

There are multiple ways to create a billboard in PRIME VSAR.

- Unreal Engine's native Component "Material Billboard"
- PRIME VSAR's Billboard Actor
- Matte plane

	"Material Billboard" Component	PRIME VSAR Billboard Actor	Matte plane
Multiple Camera	Yes	No (single camera only)	No (single camera only)
Cast shadows	No	Yes	Yes (needs to be setup in material)
Planar Reflection	No	Yes (AR Plane)	Yes (AR Plane)
Billboard mode	Based on Camera's rotation	Based on Camera's location	Attached to camera (copies location and rotation)

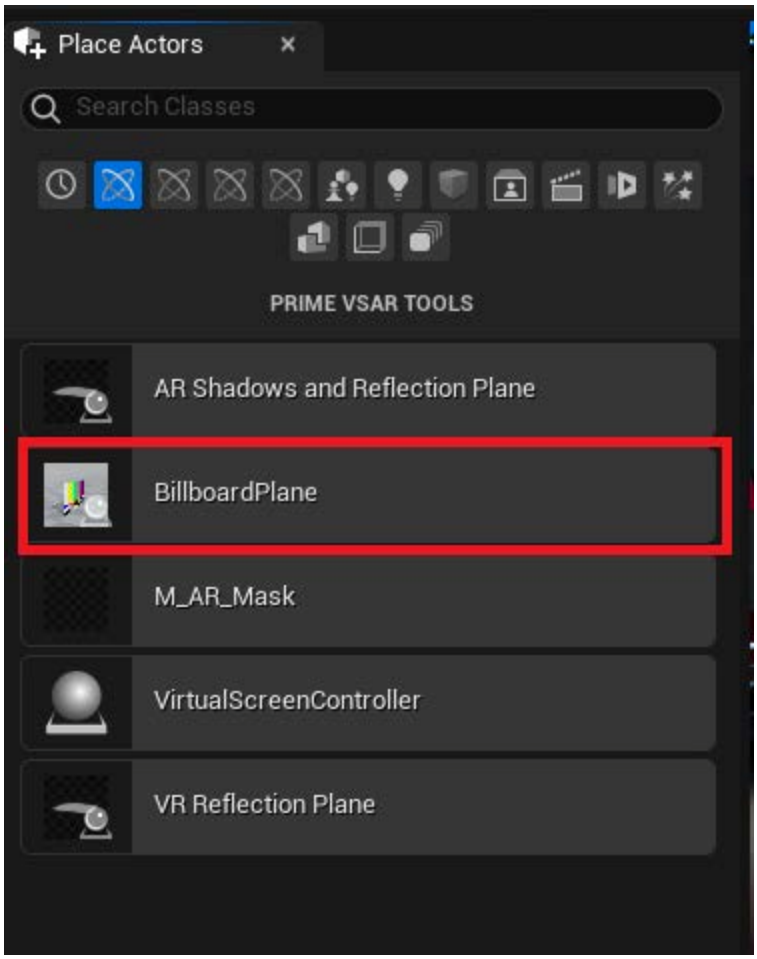
"Material Billboard" Component



Unreal Engine's "Material Billboard" Component added to an Actor

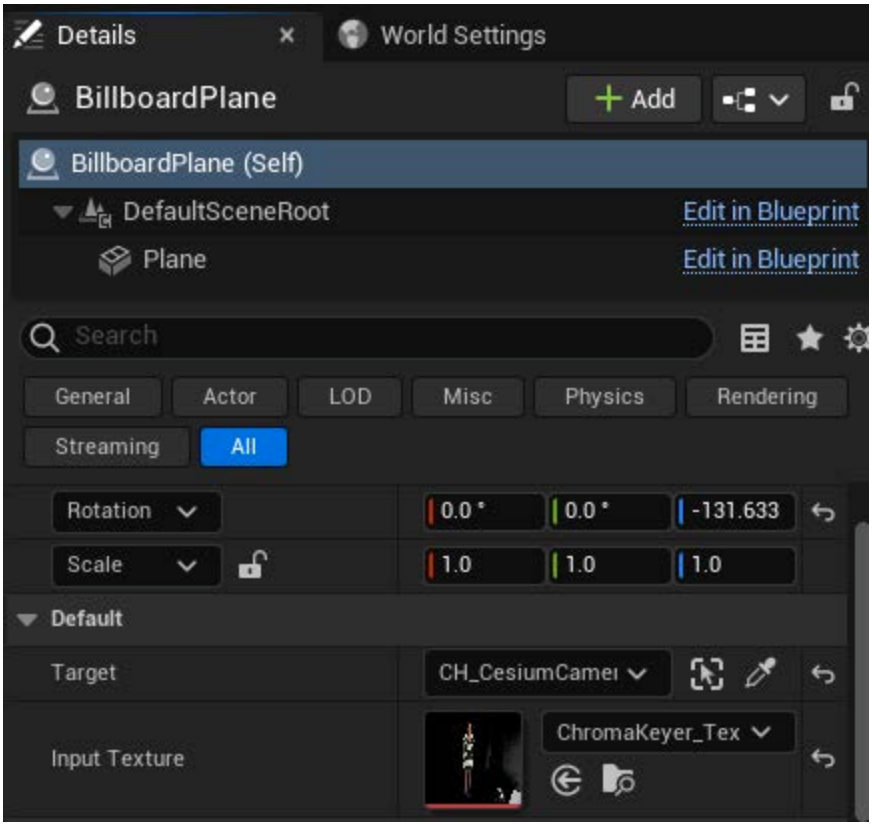
PRIME VSAR's Billboard Actor

From the menu "PRIME VSAR Tools", drag and drop the Actor "BillboardPlane" in the level.



PRIME VSAR Tools menu

Select the Actor and choose a "Target", i.e. the Object the plane should "look at" typically the Cesium Camera and choose the Texture from the Chroma Key as the "Input Texture"



📝 Billboard Plane does only rotate towards camera on the Z axis (Yaw), It does not Pitch towards the camera.

Clean Mode

Billboard's **Clean Mode** allows to capture the Talent's video feed cleanly, without applying any texture filters or tonemapping.

Original

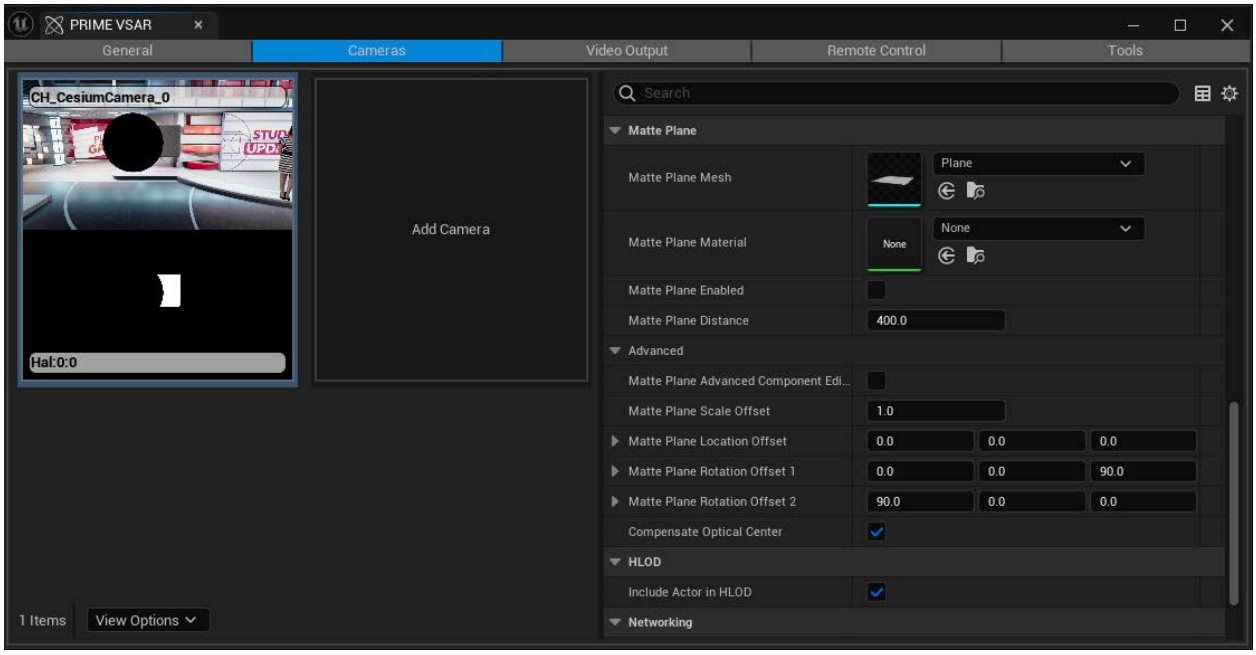
Without Clean Mode Enabled

With Clean Mode Enabled

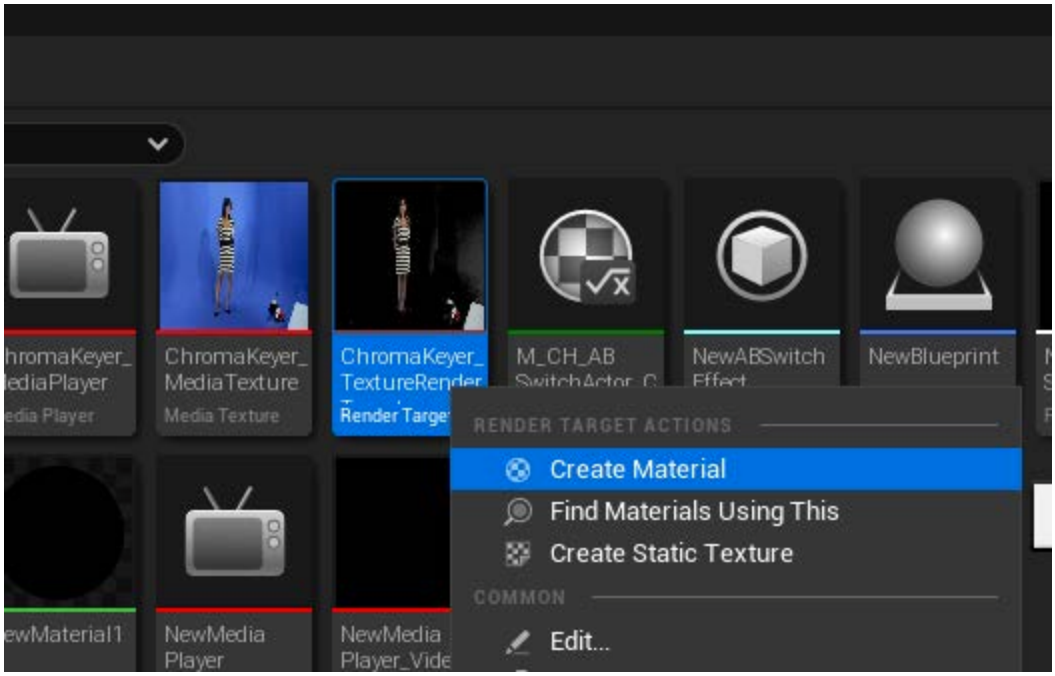
Matte plane

Plane directly attached to the Cesium camera, VSAR 2.2.0 Setup Guide has information about matte plane parameters.

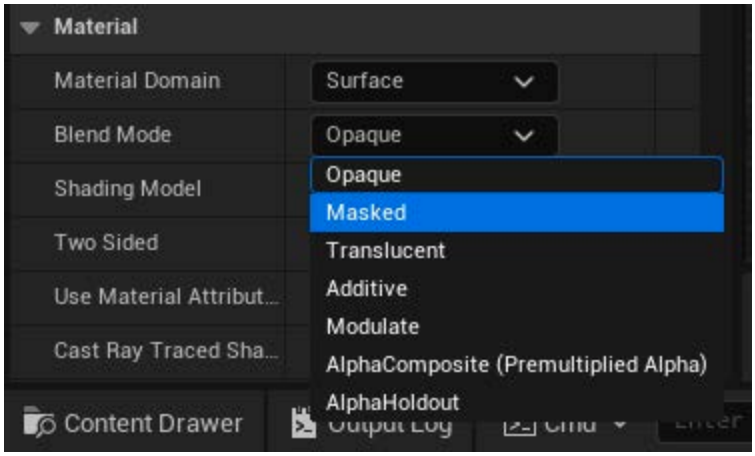
Setting for Matte plane can be changed in when clicking on cesium camera in the Matte plane section or in the VSAR Config → Cameras section



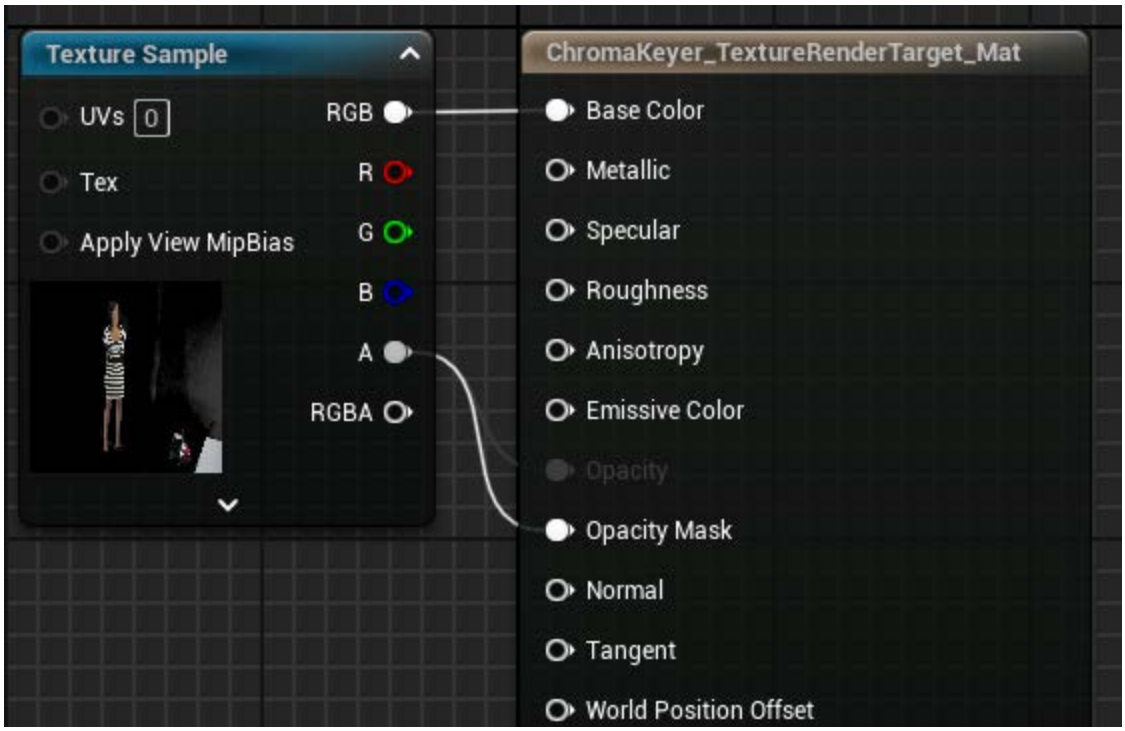
- Enable with the checker box “Matter Plane Enabled”
- Create material from the RenderTarget
  right click on the RenderTarget for the chroma key and click “Create Material”



- Open the material and set the blend mode to Translucent | Masked

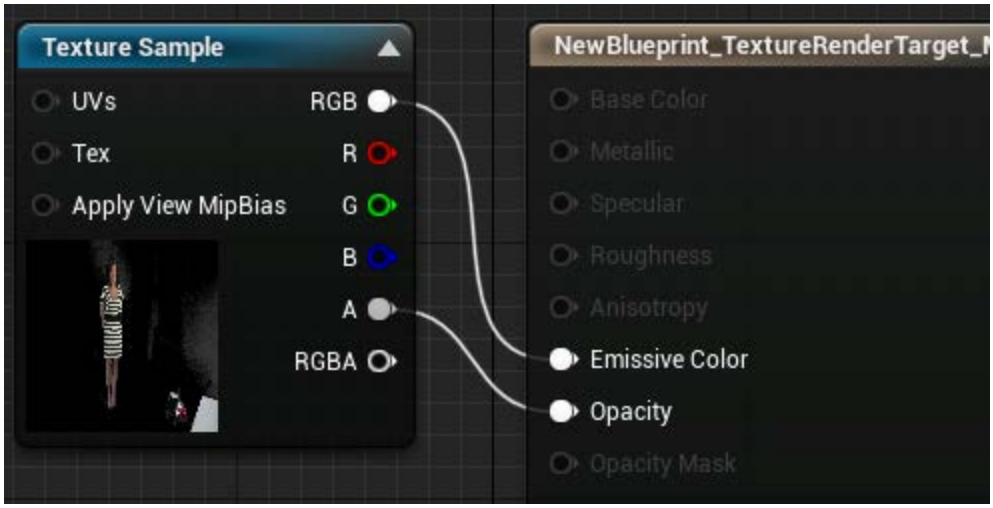


- Plug the A (alpha value) from the Texture sample to the Opacity | Opacity Mask

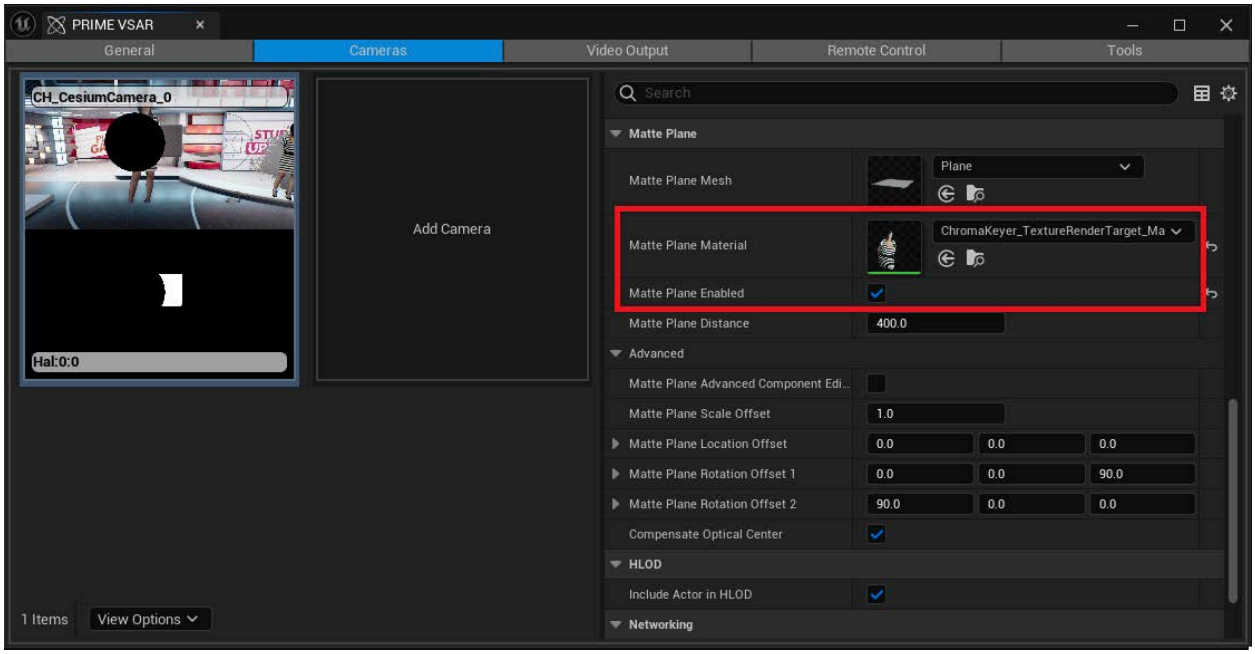


- (Optional step) set the “Shading Mode” to Unlit and plug the RGB from the Texture sample to Emissive color - this might be something that you want to make the Matter Plane not get

affected by the lighting in level.

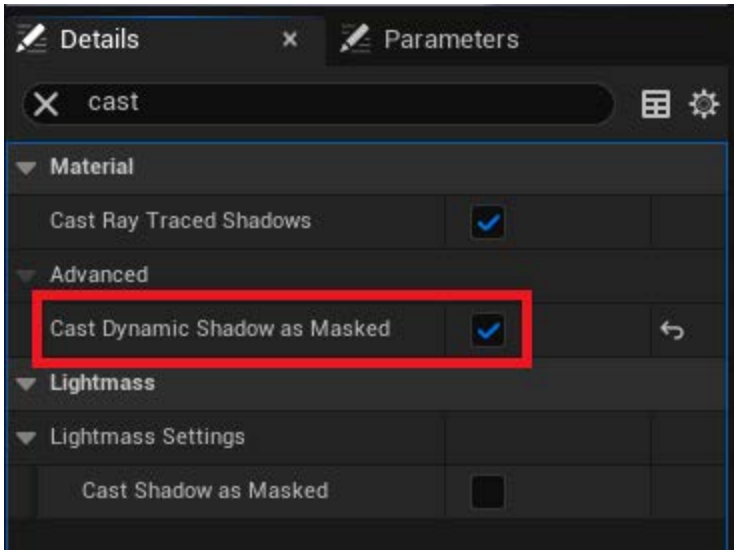


- Assign this new material to “Matte Plane Material”

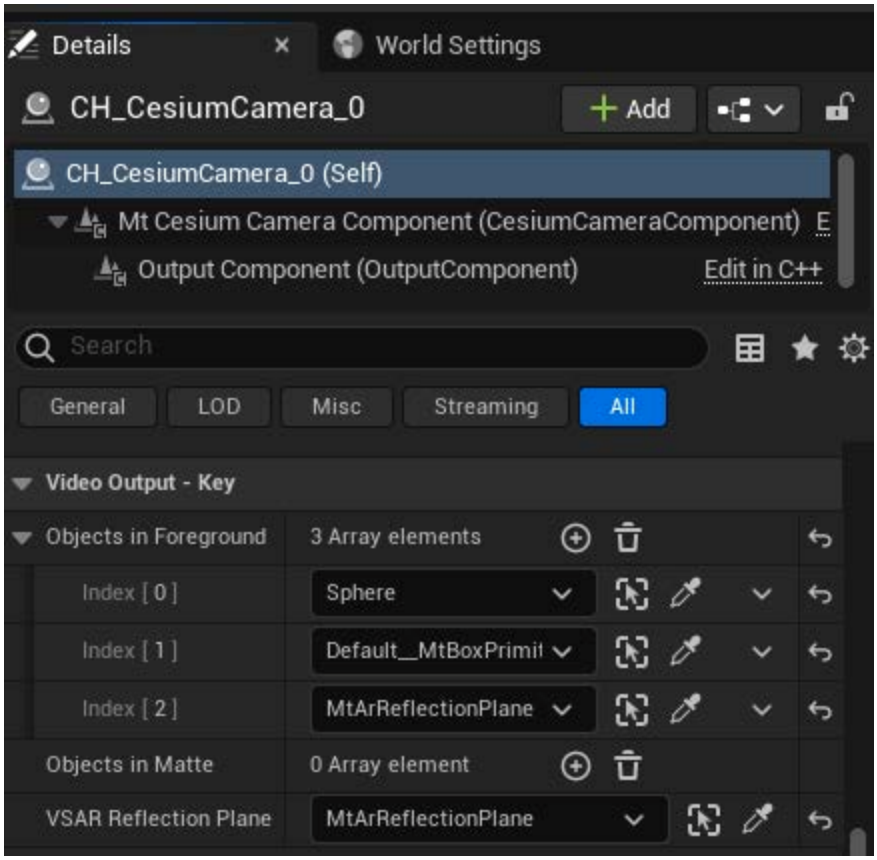


To set up shadow in the Matte Plane material when using translucent blend mode:

- In the advanced section of the material section, enable “Cast Dynamic shadow as Masked”



To set up reflection for matte plane with AR Plane:
Add Cesium camera actor itself into “Objects in Foreground” in Cesium camera details.

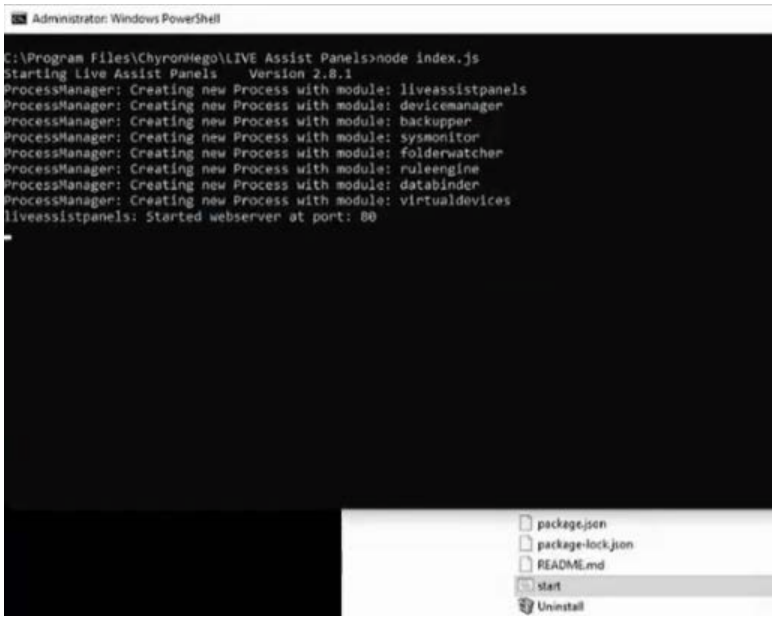


Live Assist Panels

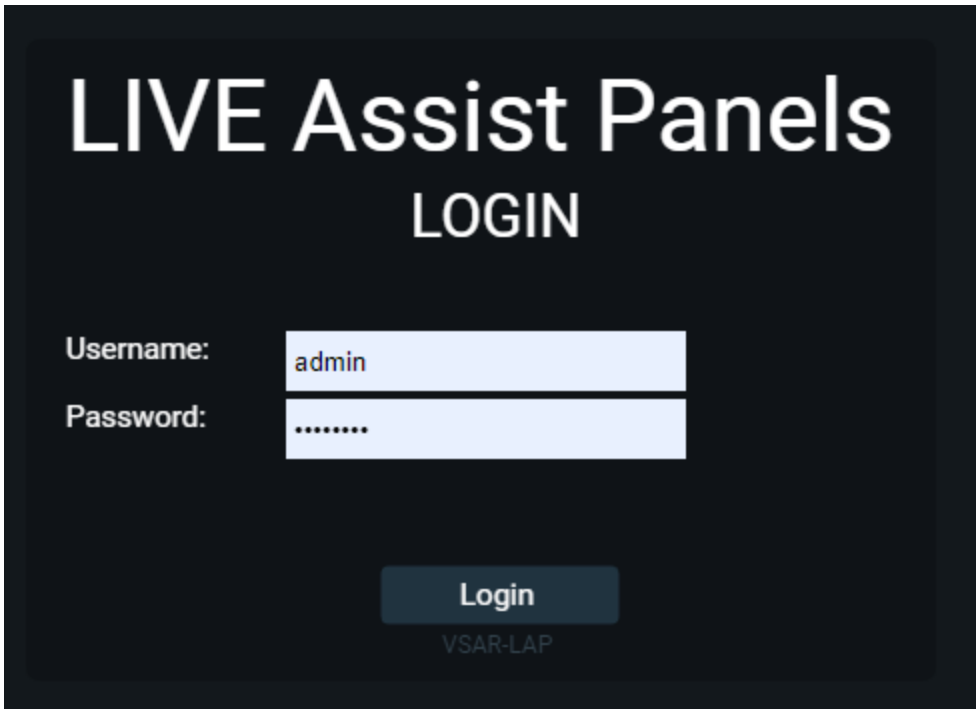
About Live Assist Panels

The Live Assist Panels allow you to design web based user panels to control elements of the Unreal projects, such as Cameras, lights, objects positions etc.

- The Live Assist Panel web server needs to be launched manually using the dedicated desktop shortcut. Run it to start the web server. You should get the following output:

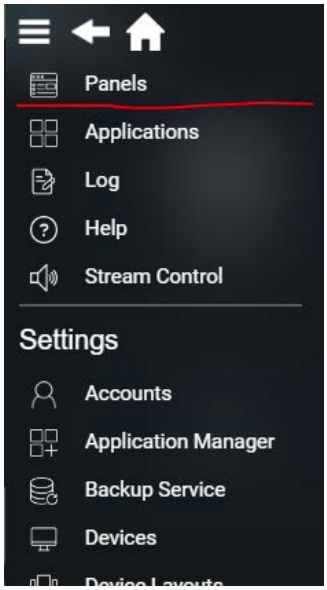


- Open Chrome and type in the following URL to open the Panel web page: **http://localhost**. Login default credentials are admin/adminLAP.

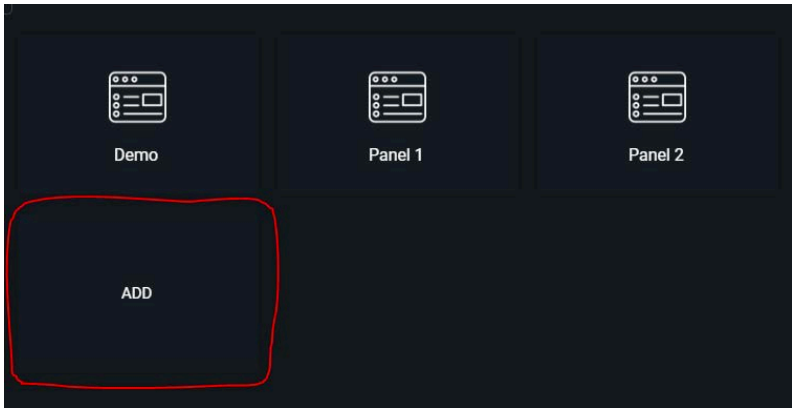


First custom panel

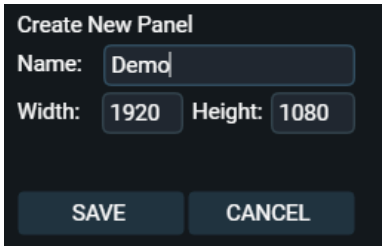
We start by selecting Panels in Live Assist Panels


Panels
Applications
Log
Help
Stream Control
Settings
Accounts
Application Manager
Backup Service
Devices

Then we add new Panel

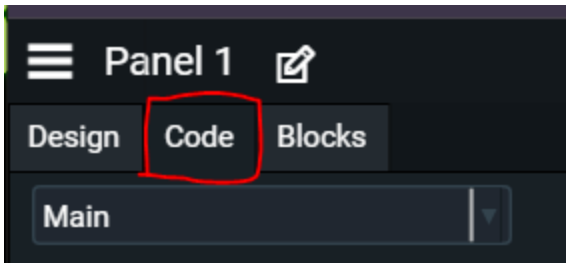


Set the Size



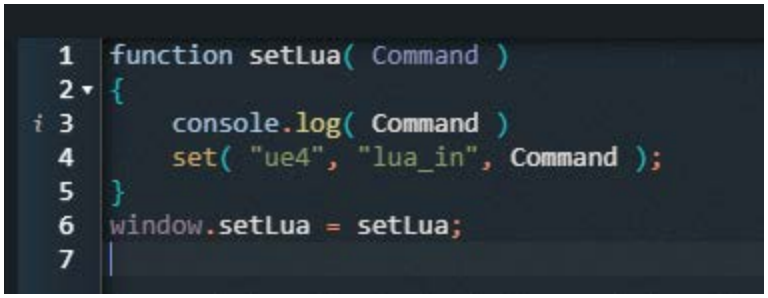
To Start Editing the Panel we need to click the Edit Icon in the Top Left Corner

Select the Code tab



In the Code -> Main we add General command to help us send Commands to VSAR Data Engine

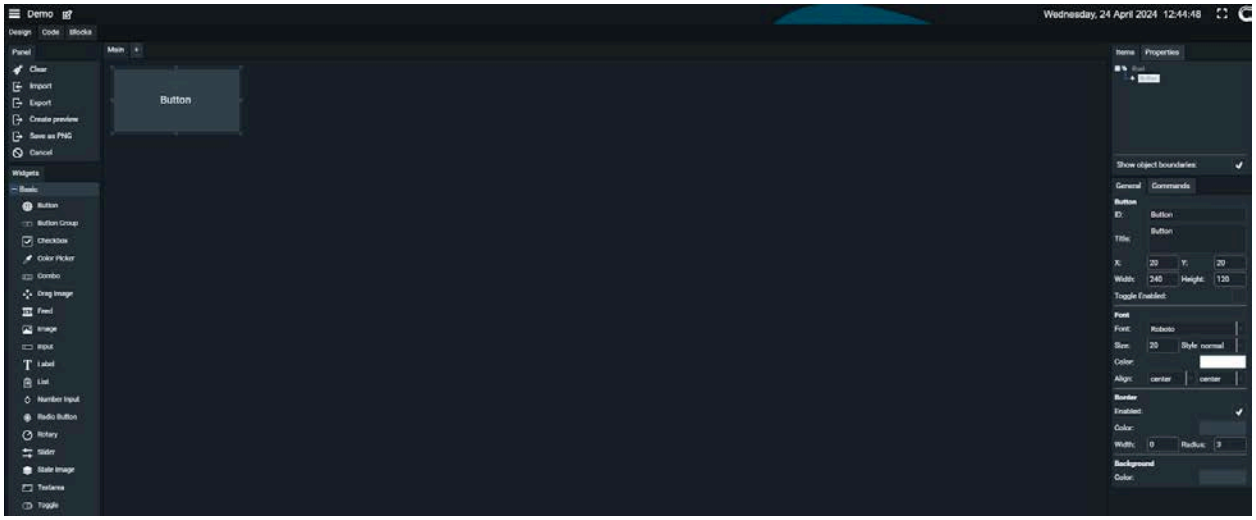
```
function setLua( Command )
{
   console.log( Command )
   set( "ue4", "lua_in", Command );
}
window.setLua = setLua;
```



We define function called setLua that accepts LUA Command, prints the command to log and sets the command in the ue4 bucket and the lua_in key

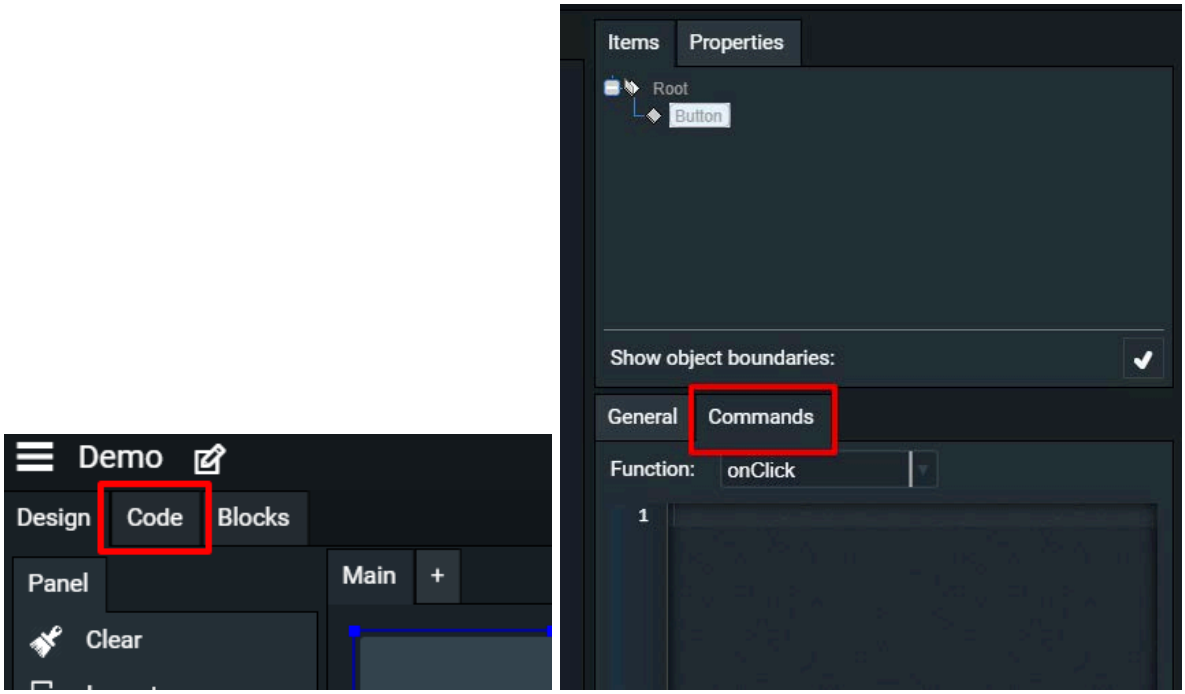
this essentially forwards the command to VSAR

On the bottom left one you will find commands that can be added to the layout by clicking on them. Start by adding a button and resize it manually to a decent size.



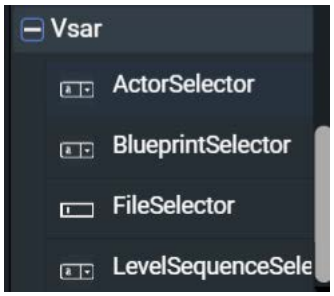
Commands and Actions that are triggered when pressing that button are written as Functions in Lua scripting language within the Code tab at the top left.

Those Functions are then called by the Button from the Commands tab at the bottom right:



VSAR Widgets in Live Assist Panels

Starting with Live Assist Panels 2.9.0, we've introduced a series of widgets specifically dedicated to VSAR control. These widgets are located under the "VSAR" category in the LAP Design view.



- Actor Selector: Used to select any Actor from the open Level in VSAR. In the VSAR tab of the Actor Selector, users can associate panel controls with different Actor properties (Position, Scale, Rotation, Visibility).
    - Base: Base value.
    - Coef.: Coefficient of change (x1, x2, x0.5, etc.). If empty defaults to 1.
    - Widget: Panel control that will provide the new values. Specially designed for association with sliders.

- Blueprint Selector: Used to select any Blueprint from the open Level in VSAR. The VSAR tab specifies:
    - Blueprint: The selected Blueprint
    - Function: The selected Function/Event
    - Widget: The control that will be used to execute the Blueprint Function/Event. Specially designed for Buttons.
    - Parameters: The list of parameters associated with the selected Function/Event, and a drop-down list to associate each Parameter value with a control on the panel.

- File Selector: Used to open an Explorer window and select a file. The General tab specifies:
    - Images Only: Filter only images.
    - Default folder: User must specify the default folder where to search for files.
    - Include file extension: Enable/Disable file extension as part of the return value

File Selector can output images directly to the Image widget.

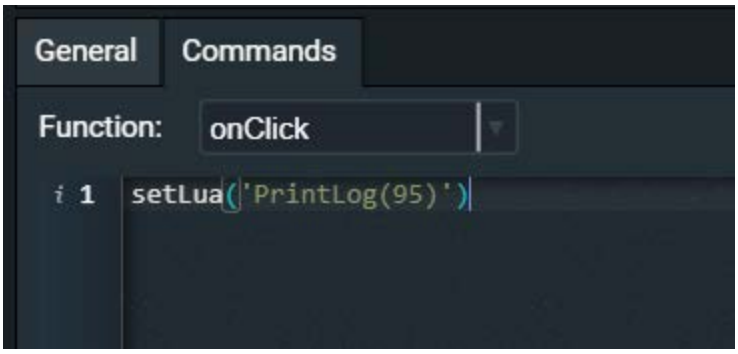
1. Add an Image widget to your panel,
2. Go to the VSAR tab and specify to which File Selector in your panel the Image widget will be associated with.

- Level Sequence Selector: Used to select any Level Sequence from the open Level in VSAR. The VSAR tab specifies a list of standard Level Sequence commands (Play, Pause, Stop, etc.) and the associated control to each one of them.

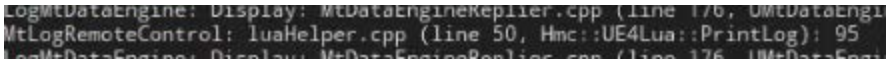
Lua script examples - Message to Unreal Output Logs

Command (onClick):

setLua('PrintLog(95)')



Output:



More on Lua API

Mercury Panel

About the Mercury Panel

The Mercury Panel is an Application inside the Live Assist Panel that allows to easily store and replay virtual camera movements. It is mostly used in trackless mode.

General Presentation

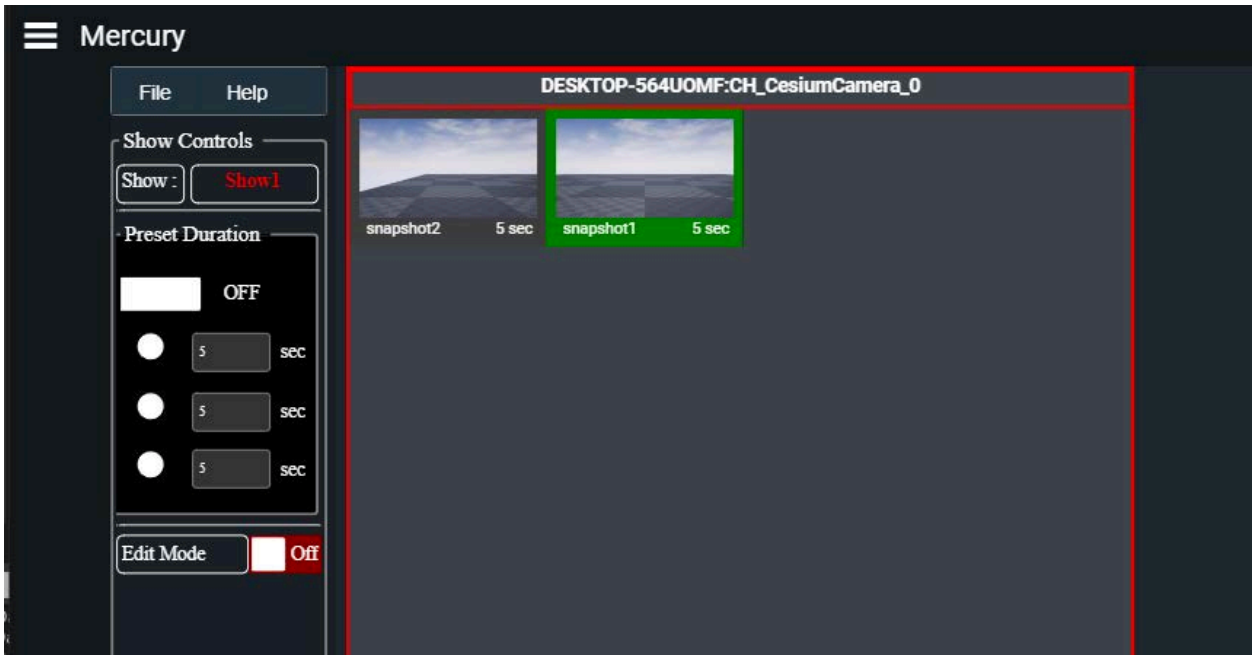
Mercury Panel runs on LAP 2.8.1. It is strongly recommended to run it on Chrome.

Mercury and Prime VSAR are communicating through the DataEngine.

- The first column deals with menus and tools.
- The other columns are dedicated to the cameras. Each configured camera has its own column. Each camera column contains snapshots of camera positions. In the following example, there are 2 configured cameras, each of them having two snapshots.

Access the Mercury Application

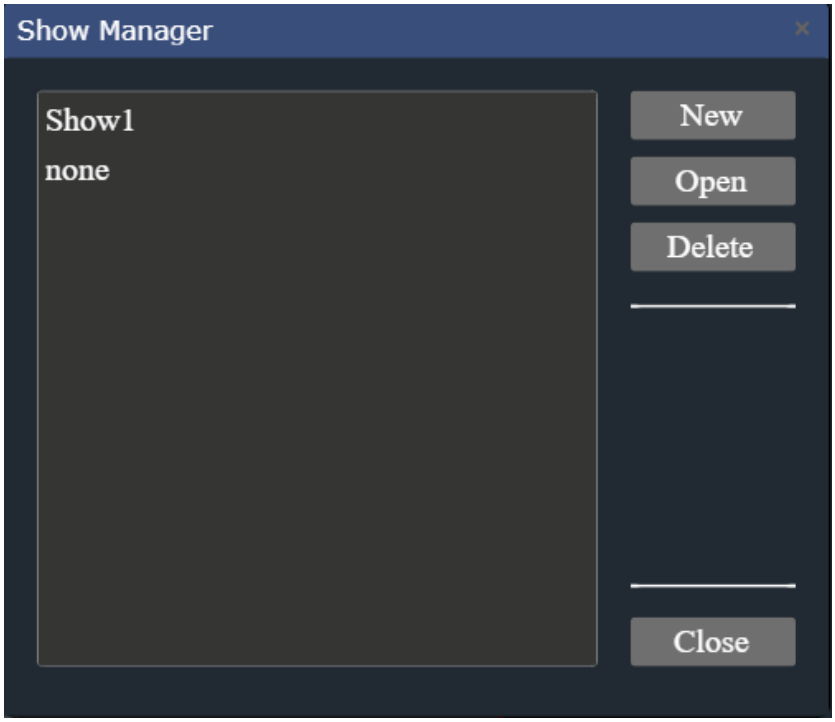
From the Live Assist Panel web interface, go to Applications → Mercury



📝 Mercury Application should be installed with VSAR, if this is not the case it can be installed manually. It is located in “[VSAR installation]\Thirdparty\Panels\lap.mercury-master.lapp”, and can be added within Live Assist Panel → Settings → Application Manager → Install Application.

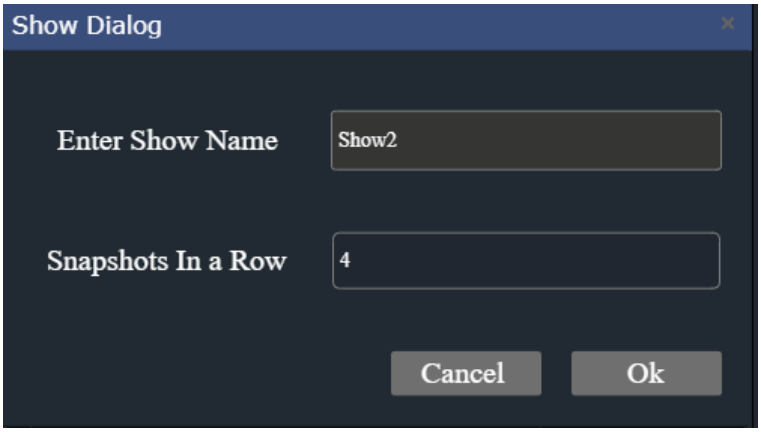
The File Menu

File > Show Manager



A show is a full Mercury project, with all the cameras and corresponding snapshots. Here you can:

New: Create a new empty show.



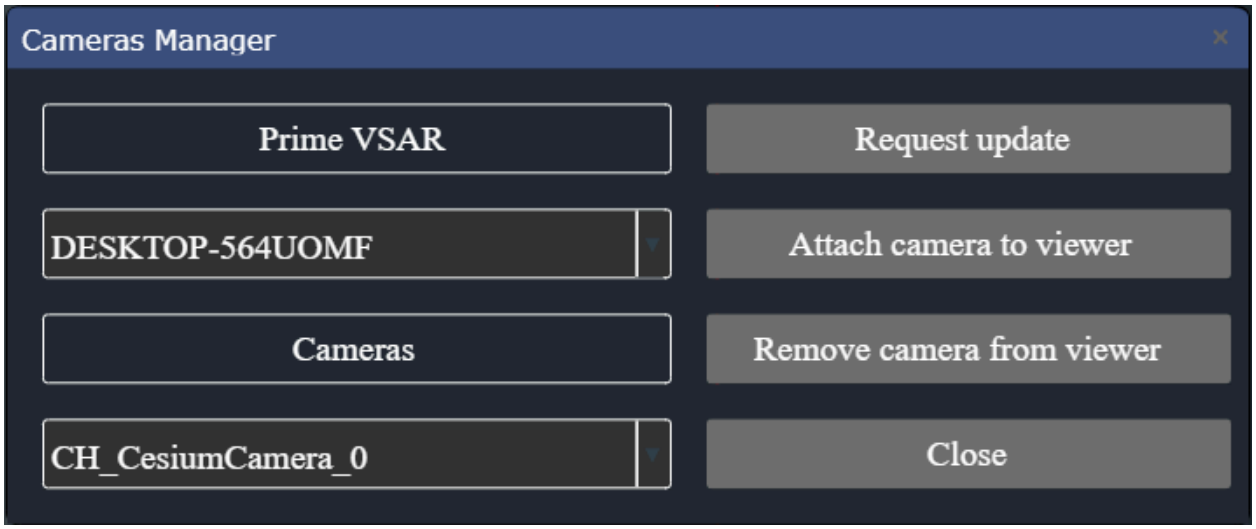
.

You have to enter the name of the show you want to create and the number of snapshots per row (default is 4). When the show is created you automatically switch to the “File>Camera Manager” (see below).

Open: open an already existing show.

Delete: delete an existing show. Acknowledgement will be requested on Show deletion.

File > Cameras Manager



- “PRIME VSAR” The first dropdown lists the Prime VSAR (computers) connected to Mercury. Here the selected system name is “DESKTOP-564UOMF”.
- “Cameras” The second dropdown list, shows the CH Cameras present in the “CH_CesiumCamera_0” Prime VSAR.
- “Request Update” will refresh the 2 dropdown lists. This may be useful if you experience connection issues with Prime VSAR.
- “Attach camera to viewer” will create a new column in Mercury to handle the selected camera. Repeat this operation as many times as needed. You are not required to create a column for each camera.
- “Remove camera from viewer” removes the corresponding camera column.
- “Close” closes the dialog.

File > Quick Save Show

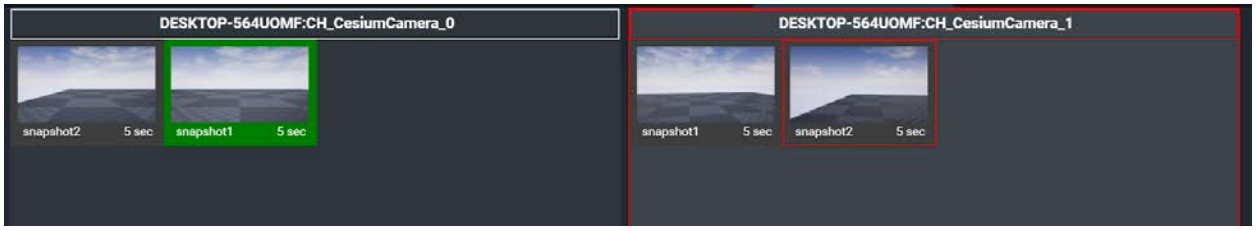
Under normal operation, the show configuration is automatically saved when exiting “Edit mode”. If for some reason you want to save the state of the show during “Edit mode”, select this menu.

The Toolbox

- “Show” displays the name of the current show (here “Show1”).
- “Preset Duration” when On (green) the selected time will be used to start camera transition. When Off (red) the time given in the snapshot will be used. There are 3 available preset times that you can set.
- “Edit Mode” when On (green) edition can be done on snapshots (see below) and camera transitions are forbidden, when Off (red) edition is disabled and camera transitions can be run.
- Editing only: “Store Position”, when you click this button a new snapshot is immediately created for the currently selected camera. A default name and duration are set.

- Editing only: “Delete Position”, when you click this button, the current selected snapshot is removed. An acknowledgement is required.
- Editing only: “Rename Position” posts a dialog to change the name of the snapshot.
- Editing only: “Edit Position Duration” posts a dialog to change the duration of the snapshot.

Cameras and Snapshots



This example displays a show made of 2 cameras. The first one has 4 snapshots, the second has 2 snapshots.

The currently selected camera has a red frame around it(in the top part, you can see a name of the selected camera in a red frame). Here the second(CH_CesiumCamera_1) camera is selected.

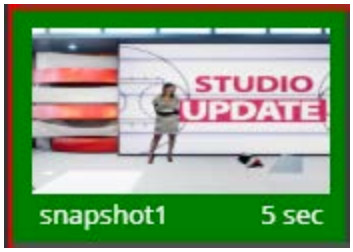
The currently selected snapshot has a green frame around it. Here is the 2th snapshot of camera 1 (CH_CesiumCamera_0).

It is possible to deselect a snapshot by selecting another snapshot or by clicking outside of the snapshot area.

Trigger a camera move: When a snapshot is selected (red frame), and if “Edit Mode” is Off, clicking again on the same snapshot will trigger the camera movement. The used duration will be:

- The selected preset duration, if “Preset Duration” is checked.
- The snapshot duration, if “Preset Duration” is unchecked.

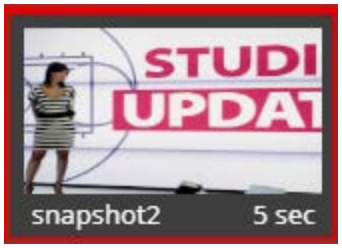
During the camera move a thick green frame will blink around the snapshot and the duration will countdown.



When the actual position is reached the green frame will still be displayed around the snapshot.

Move snapshots: to move snapshot:

- Check select “Edit Mode”.
- Select a snapshot. A red frame will surround the image.

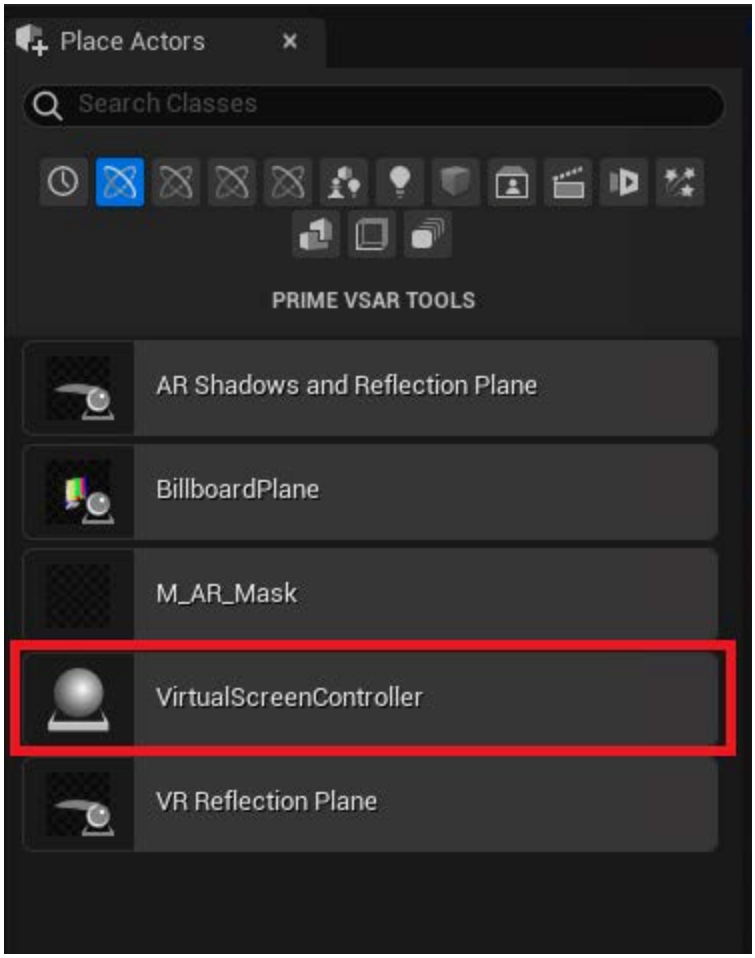


- Select either:
    - An empty location: the snapshot will move to this location.
    - An already created snapshot: the positions of the two snapshots will be exchanged.

Virtual Trackless Controller with CAMIO

Camera snapshots can also be recalled with the CAMIO Template “VirtualTracklessController”. The camera snapshots used by VirtualTracklessController needs to be premade in Mercury.

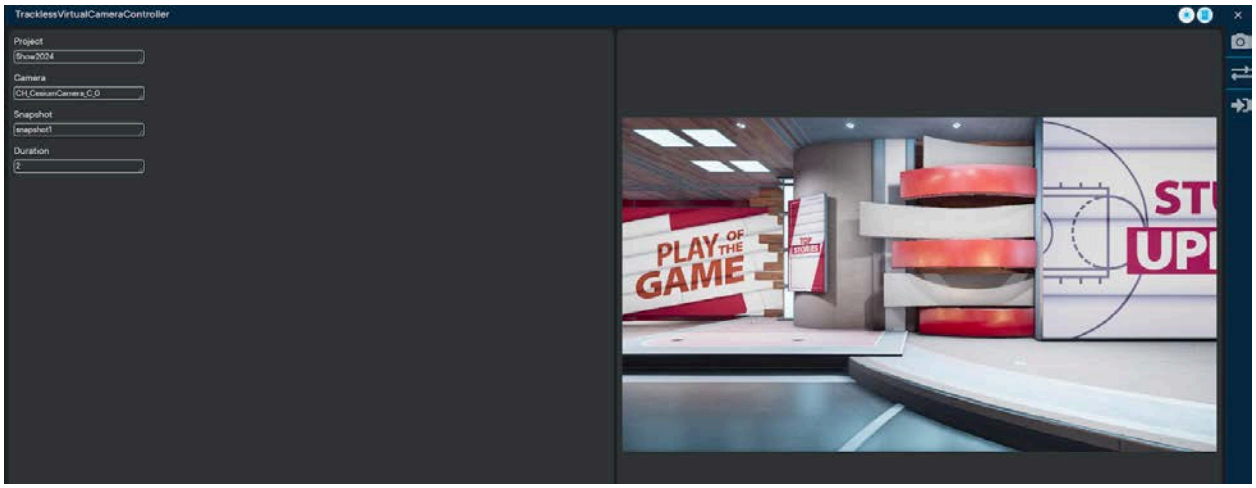
Drag and drop the VirtualTracklessController form the Place actors panel:



it will spawn several cesium cameras, this is normal behavior. Select the VirtualTracklessController in level and in the details panel select “Export CRD..” and import it to CAMIO.

📝 LUCI Render is using spawned camera not the actual camera to take images and while rendering the duration is 0

⚠ while using LUCI Render with VirtualTracklessController it does not support AR Shadow and Reflections plane



Template editor in LUCI5

The template as the following replaceables:

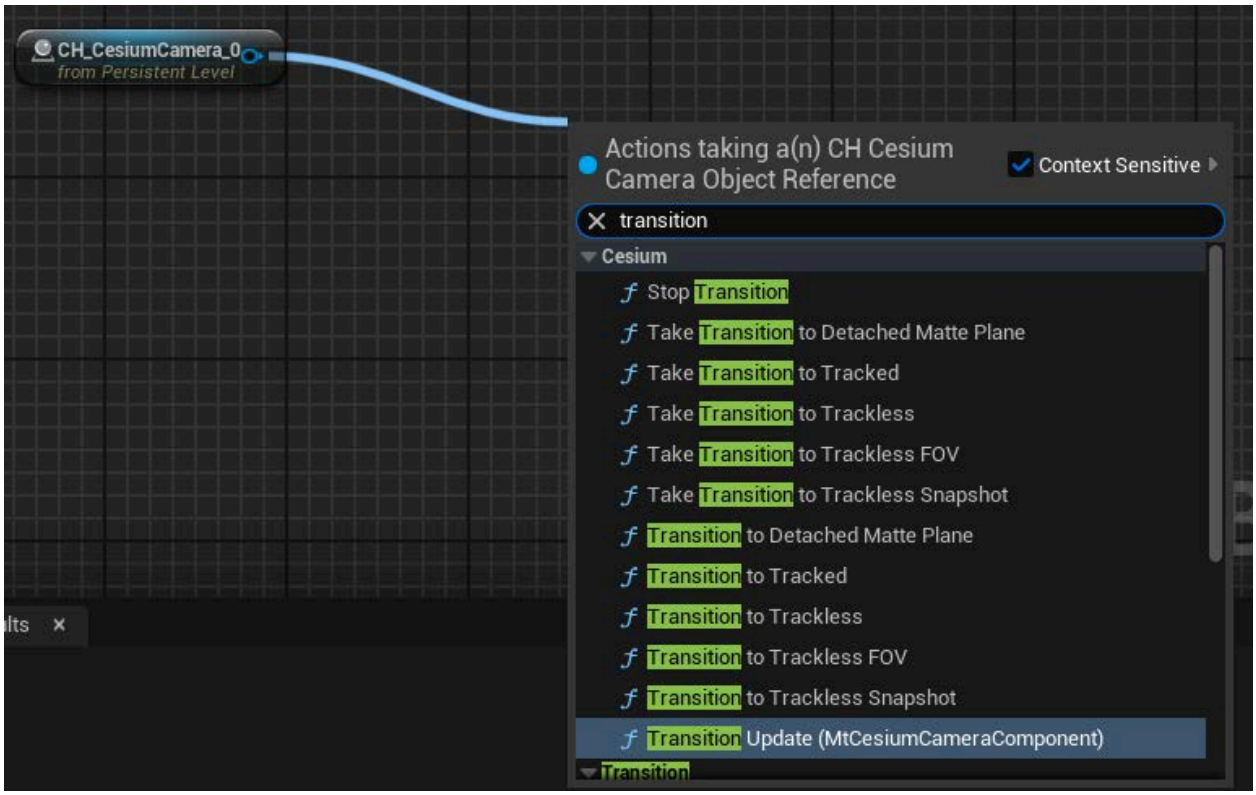
- Project: The name of the “Show” in Mercury Panel (Note: Not to be confused with VSAR Project name)
- Camera: The name of the camera actor in the VSAR level (eg. CH_CesiumCamera_0)
- Snapshot: The name of the snapshot to recall
- Duration: The duration in seconds of the transition from the current camera position to the snapshot position

Cesium Camera Transitions

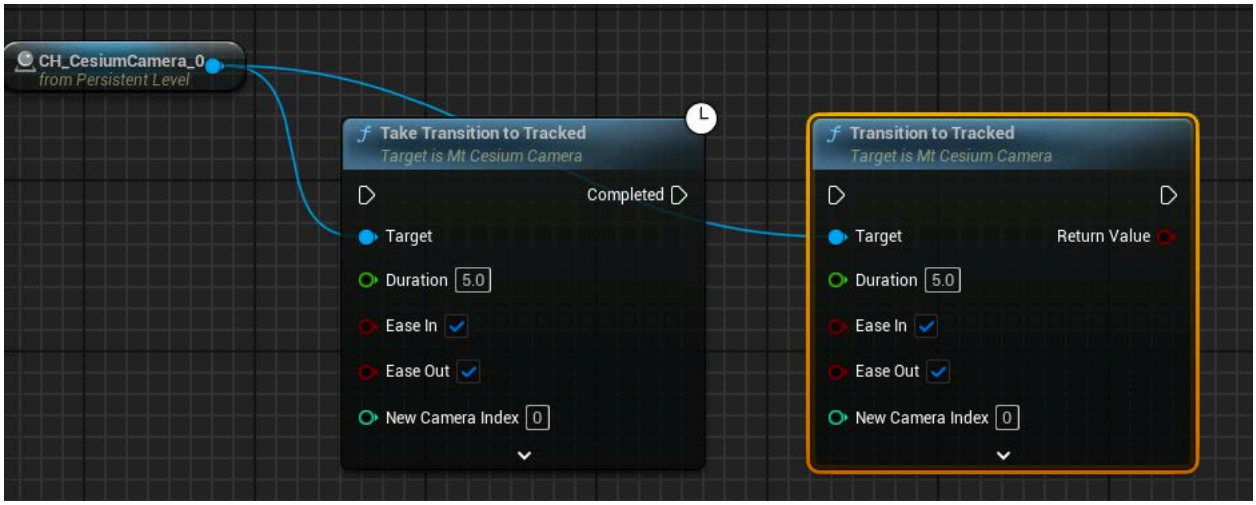
The Cesium camera transitions enable the transition of the camera from Trackless to tracked and back. you can also transition from one trackless snapshot to different trackless snapshot or transition from one tracked camera to another tracked camera.

you can use several Blueprint functions from Cesium Camera

- Transition to tracked
- Transition to trackless
- Transition to trackless FOV
- Transition to trackless snapshot
- Transition to detached Matte Plane
- Stop transition



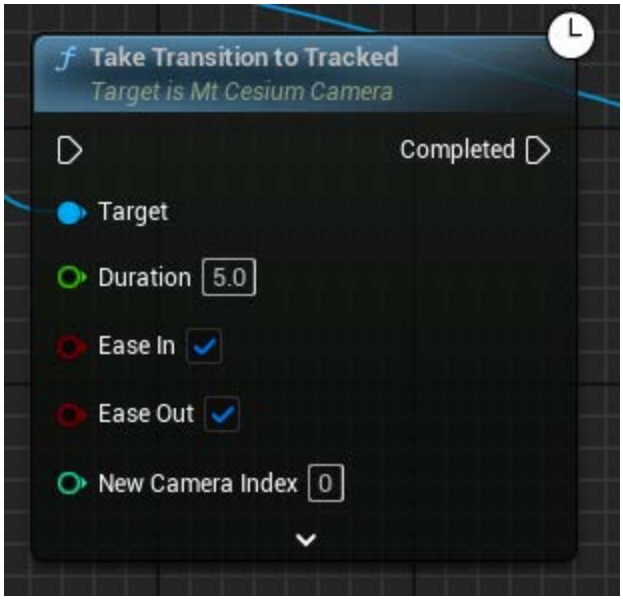
We have two variants of blueprint functions Take and non-take



Take functions execute next node after the transition is finished

Transition to tracked

This function tries to transition to Cesium Tracked Camera, it expects that Cesium is connected and the Camera index is valid.

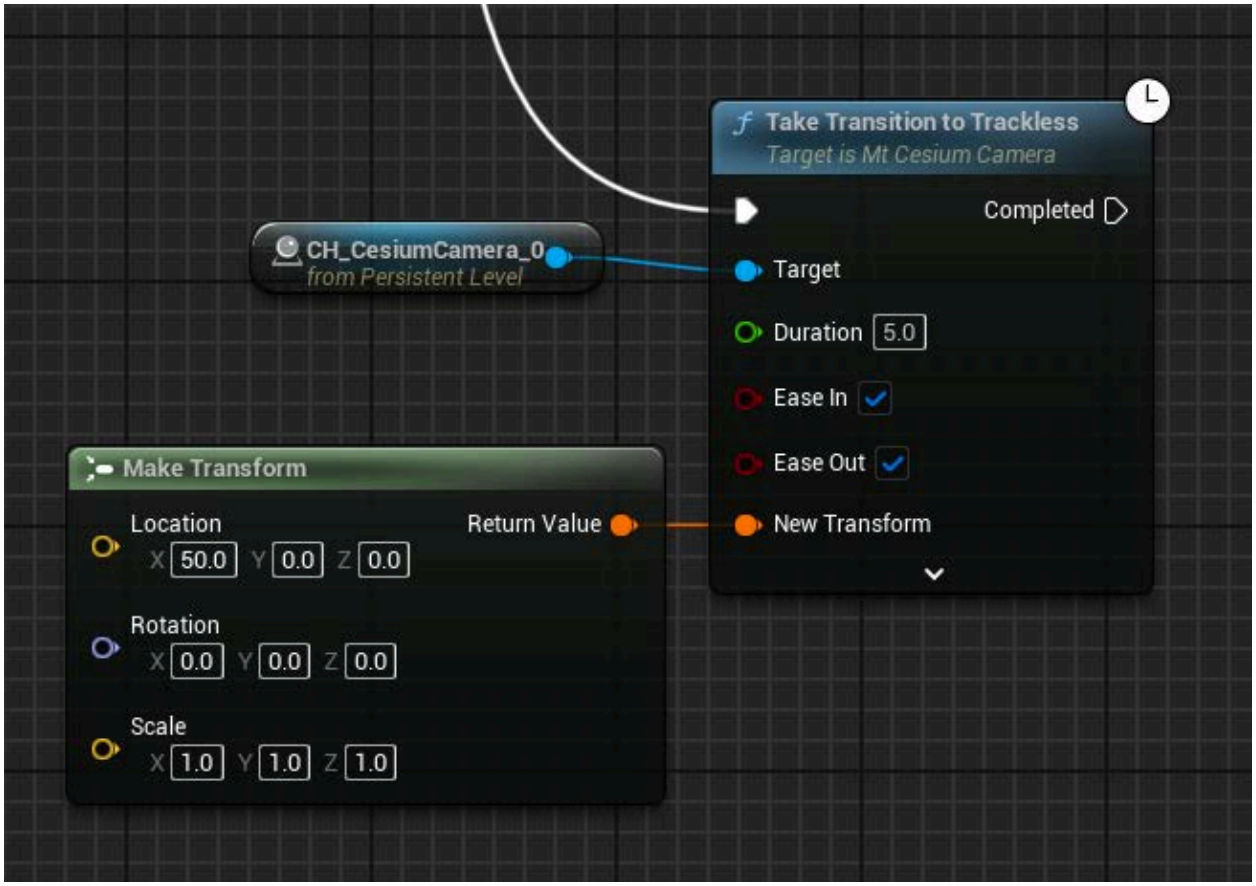


- Duration - duration that the transition takes
- Ease In - if we want to slow down on the end of the transition

- Ease Out - if we want to slow down when starting the transition
- New Camera Index - the camera index refers to the Camera Index in Cesium Camera that refers to the rig index in Cesium
- Custom Float Curve - replaces the default Curve with the provided one, note that Duration, Ease In, Ease Out is ignored
- Detach Matte Plane - detaches matte plane if it is enabled
- Reattach Matte Plane - tries to reattach detached matte plane if it exist
- Track Matte Plane - should the detached plane keep tracking as if it would be attached to Cesium Camera, if detaching from Tracked Camera
- Matte plane look at camera - should the detached plane keep rotating (Yaw/Pan) towards the camera
- Matte plane look at camera - should the detached plane keep rotating (Yaw/Pan) towards the camera
- Disable alpha capture - when this option is enabled, the VSAR Output Key channel stops updating, freezing on the last captured frame. This is primarily to keep the Garbage Matte frozen while the camera is virtually moving. For the Tracked Transition, this option is disabled by default, as it is understood that on tracked cameras, the Garbage Matte will update in real time along with the camera.

Transition to trackless

This function transitions to custom transform with custom lens data

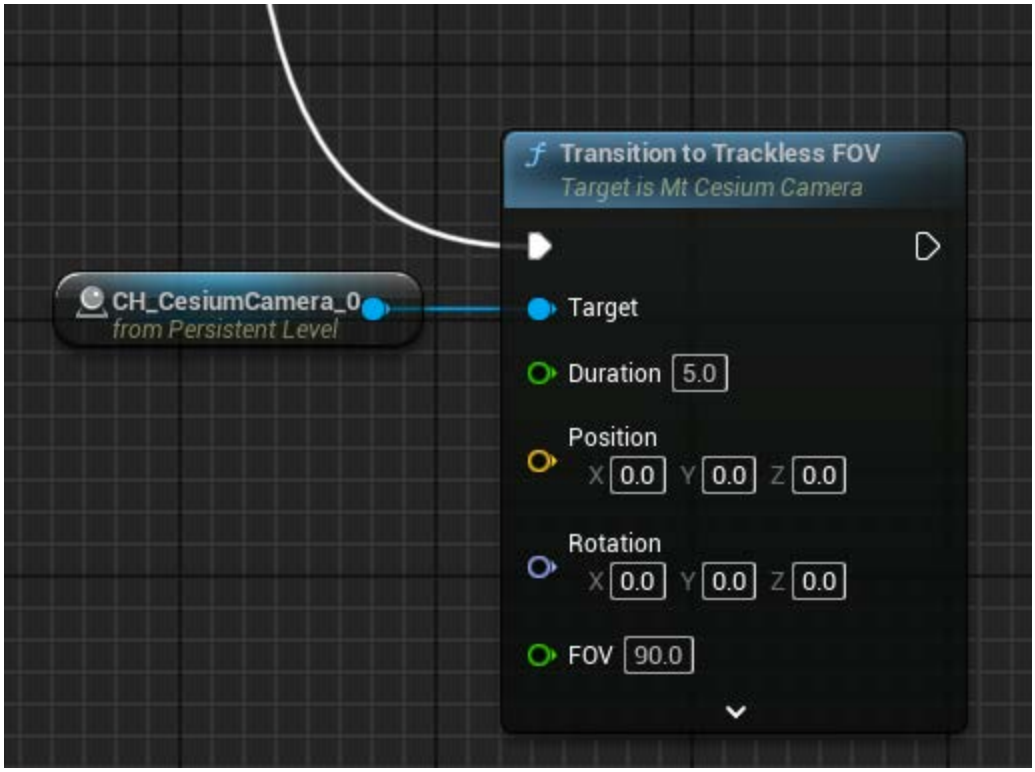


- Duration - duration that the transition takes
- Ease In - if we want to slow down on the end of the transition
- Ease Out - if we want to slow down when starting the transition
- New Transform - custom transform in the world
- In Receptor Props - custom Lens data
- Custom Float Curve - replaces the default Curve with the provided one, note that Duration, Ease In, Ease Out is ignored
- Detach Matte Plane - detaches matte plane if it is enabled
- Reattach Matte Plane - tries to reattach detached matte plane if it exist
- Track Matte Plane - should the detached plane keep tracking as if it would be attached to Cesium Camera, if detaching from Tracked Camera
- Mette plane look at camera - should the detached plane keep rotating (Yaw/Pan) towards the camera

- Disable alpha capture - when this option is enabled, the VSAR Output Key channel stops updating, freezing on the last captured frame. This is primarily to keep the Garbage Matte frozen while the camera is virtually moving. For the Trackless Transition, this option is enabled by default, as it is understood that when transitioning from Tracked to Trackless, users will want the alpha channel to remain still as the virtual camera moves.

Transition to trackless FOV

simplified version of Transton to trackless

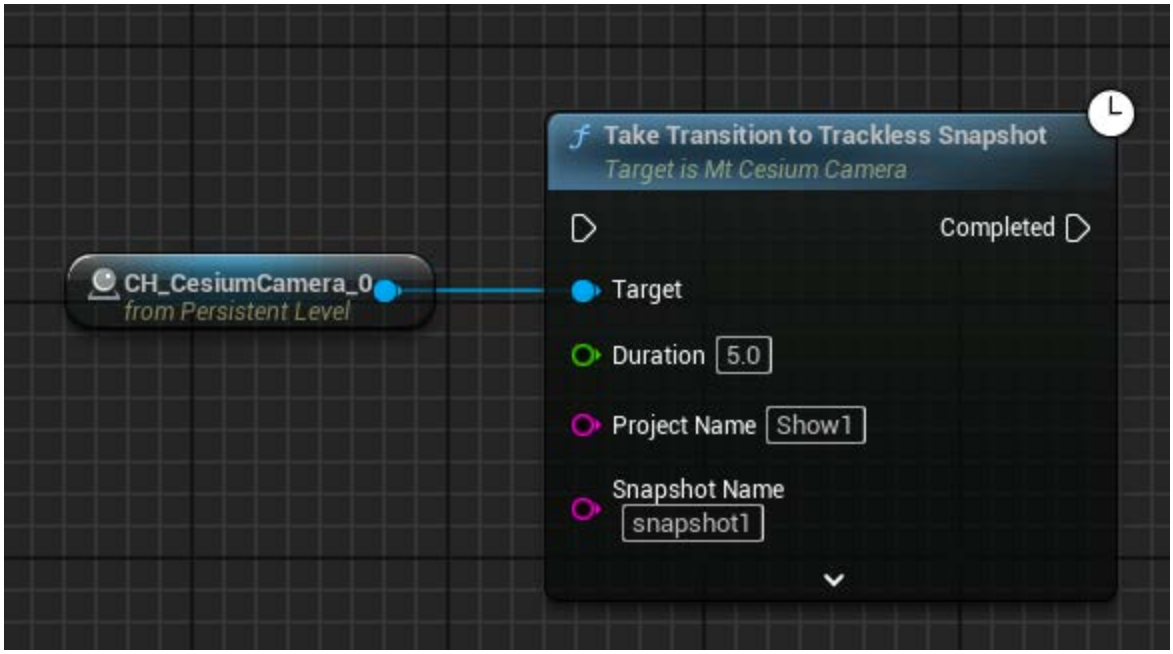


- Duration - duration that the transition takes
- Ease In - if we want to slow down on the end of the transition
- Ease Out - if we want to slow down when starting the transition
- Position - position in the world
- Rotation - rotation in the world
- FOV - Field of view
- Detach Matte Plane - detaches matte plane if it is enabled
- Reattach Matte Plane - tries to reattach detached matte plane if it exist

- Track Matte Plane - should the detached plane keep tracking as if it would be attached to Cesium Camera, if detaching from Tracked Camera
- Mette plane look at camera - should the detached plane keep rotating (Yaw/Pan) towards the camera
- Disable alpha capture - when this option is enabled, the VSAR Output Key channel stops updating, freezing on the last captured frame. This is primarily to keep the Garbage Matte frozen while the camera is virtually moving. For the Trackless Transition, this option is enabled by default, as it is understood that when transitioning from Tracked to Trackless, users will want the alpha channel to remain still as the virtual camera moves.

Transition to trackless snapshot

tries to transition to snapshot defined by the Mercury (Trackless) app, currently Ease In, Ease Out is always enabled

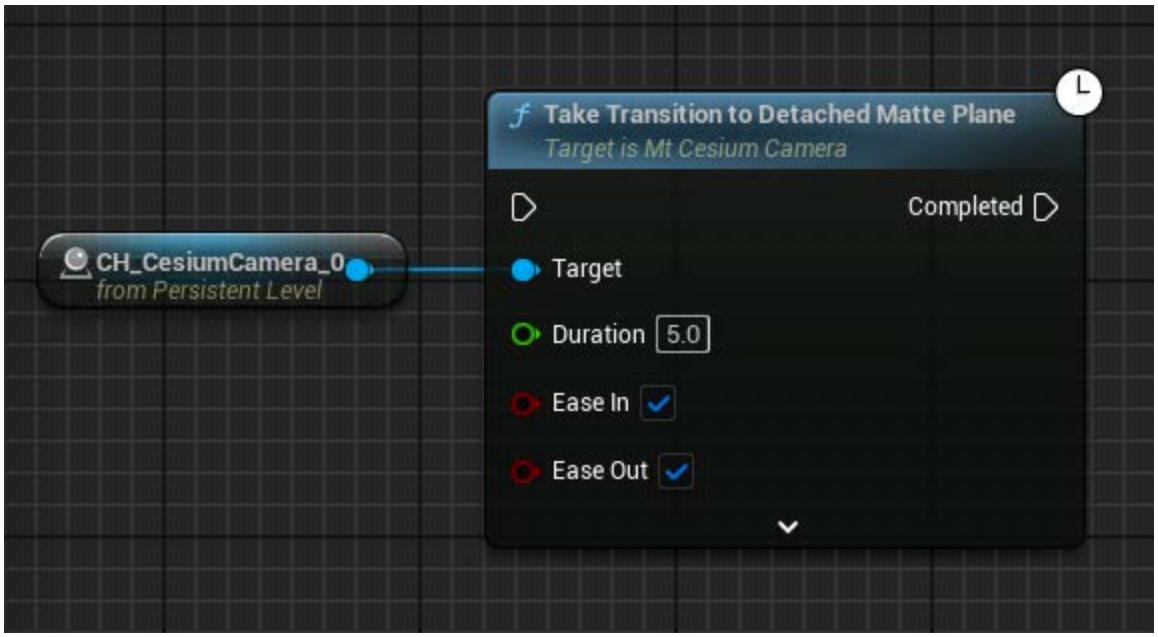


- Duration - duration that the transition takes
- Project Name - project or Show name in Mercury (Trackless) app
- Snapshot Name - name of the Snapshot in Mercury (Trackless) app
- Detach Matte Plane - detaches matte plane if it is enabled
- Reattach Matte Plane - tries to reattach detached matte plane if it exist
- Track Matte Plane - should the detached plane keep tracking as if it would be attached to Cesium Camera, if detaching from Tracked Camera

- Mette plane look at camera - should the detached plane keep rotating (Yaw/Pan) towards the camera
- Disable alpha capture - when this option is enabled, the VSAR Output Key channel stops updating, freezing on the last captured frame. This is primarily to keep the Garbage Matte frozen while the camera is virtually moving. For the Trackless Transition, this option is enabled by default, as it is understood that when transitioning from Tracked to Trackless, users will want the alpha channel to remain still as the virtual camera moves.

Transition to detached Matte Plane

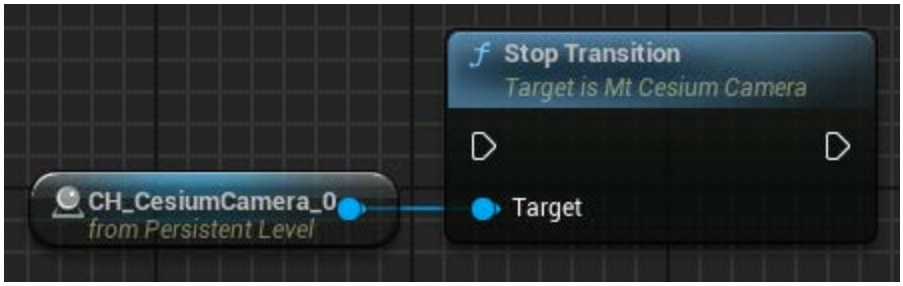
tries to transition to already detached matte plane from this camera if it exist



- Duration - duration that the transition takes
- Ease In - if we want to slow down on the end of the transition
- Ease Out - if we want to slow down when starting the transition
- Reattach Matte Plane - tries to reattach detached matte plane if it exist

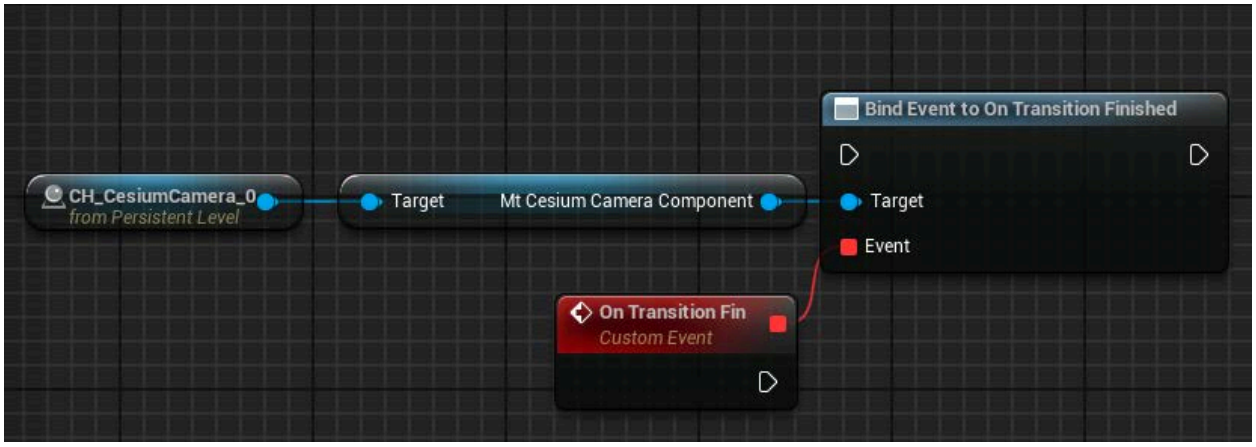
Stop transition

stops currently running transition



On transition finished

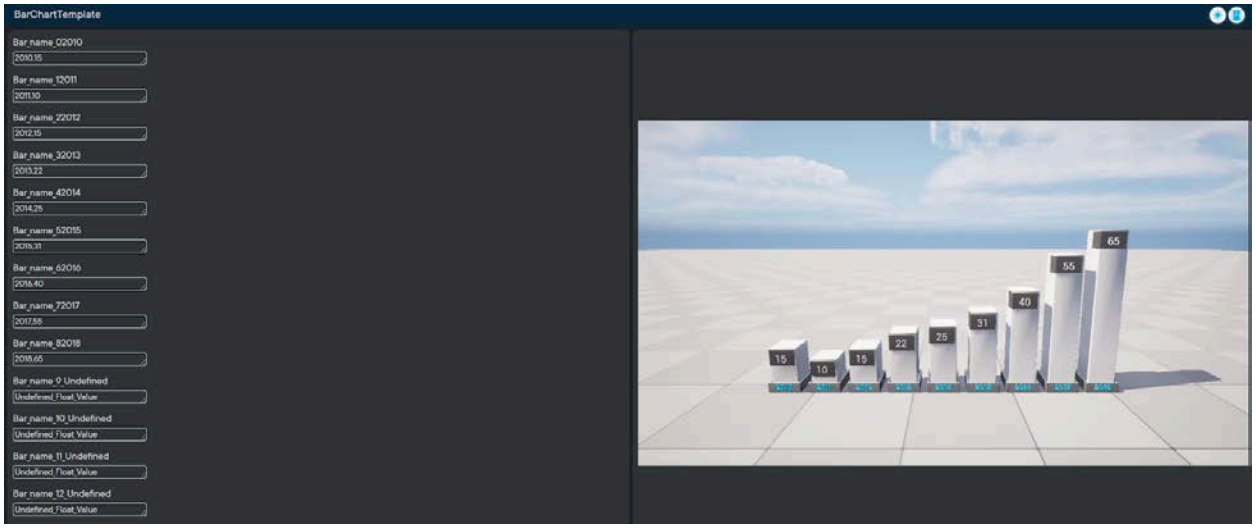
if needed you can also bind event to Cesium Camera Component to get notified when any transition of this Cesium Camera is done



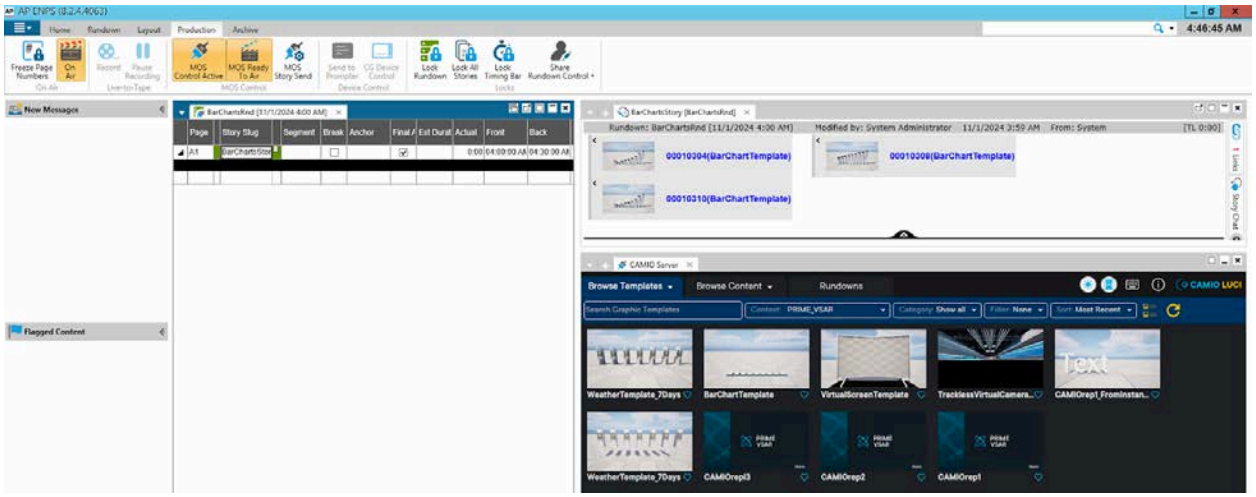
VSAR Scenes for CAMIO

Scenes within PRIME VSAR have to be created using CAMIO related specific objects, in order to make them available within LUCI as Templates, and thus useable within NRCS systems such as iNews or ENPS for creating new Graphic objects, which can be then played out with ISQ.

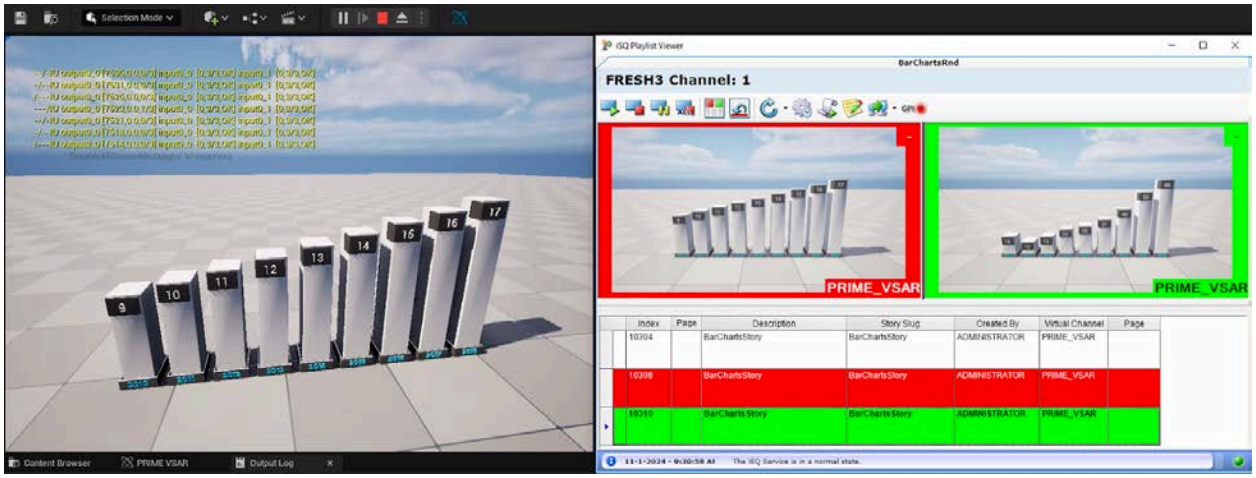
Check out the following **document** to set up the PRIME VSAR ↔ CAMIO Integration.



LUCI5 with PRIME VSAR Template



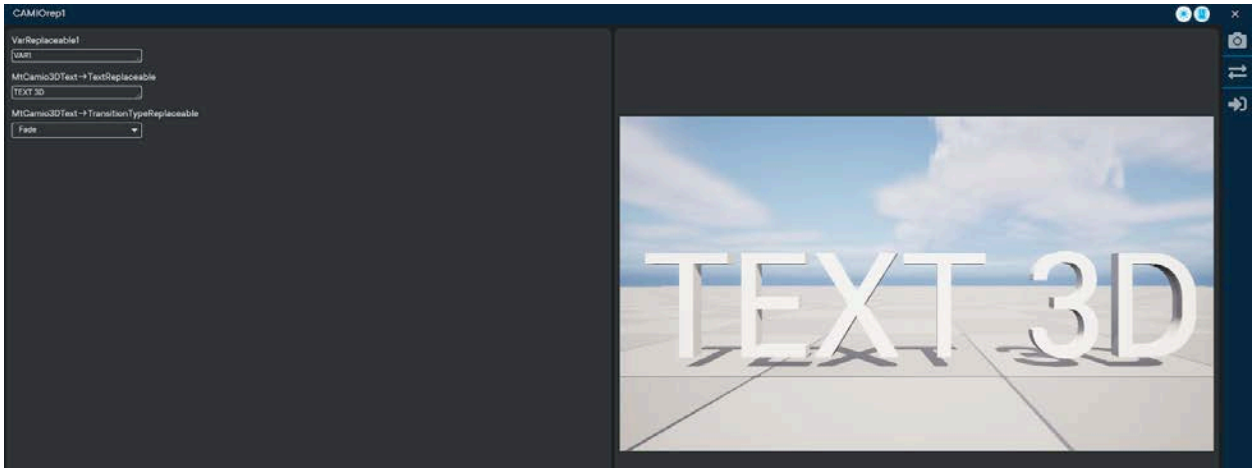
ENPS with LUCI5 and PRIME VSAR Template



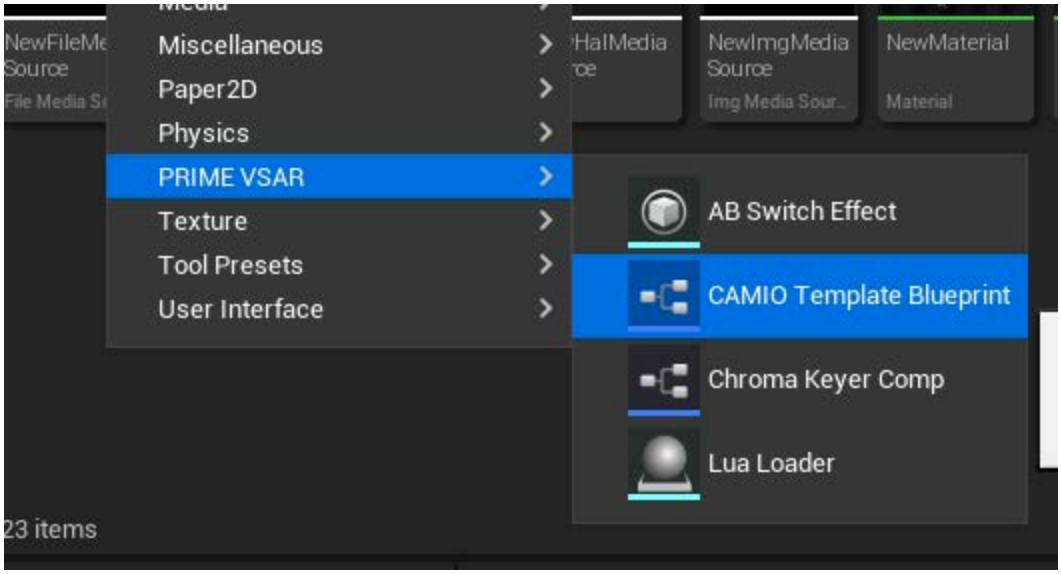
Playout with ISQ

Creation example with a simple Text Scene

The goal of this example is to create a simple Text Scene in VSAR to be accessible in CAMIO for creation, rendering and playout with LUCI, a NRCS and ISQ.



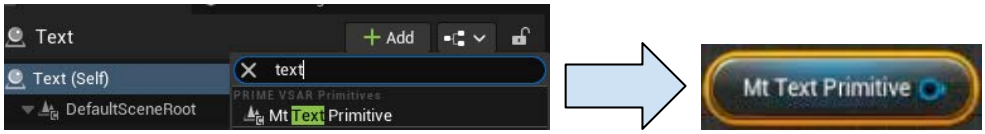
VSAR Scenes should be of a specific Blueprint Class: “CamioTemplate” in order to be able interact with them in CAMIO.



Within the newly created Scene's (or 'BluePrint Class' in Unreal environment) Event Graph window, the following Items should be created:

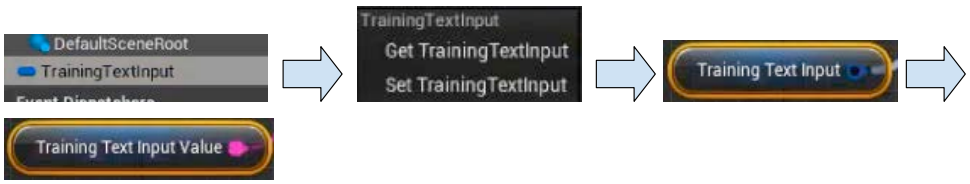
Primitives

- Click 'Add Component' to create a Text Replaceable by selecting 'Mt Text Primitive'. Drag and drop the created element onto the Canvas. This should also add the text object into the main design window.



Variables

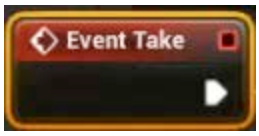
- Create a TextInput variable of type 'Camio Replaceable'.
- Compile to get access to the 'Value Fields'.
- Drag and drop the created component into the Canvas and select the 'Get TextInput' option that is displayed. Then right-click the created element and select 'Split Struct Pin' to display the Value element of this Text Variable.



Actions

- Right-Click and create an Event Render.

- Right-Click and create an Event Take.



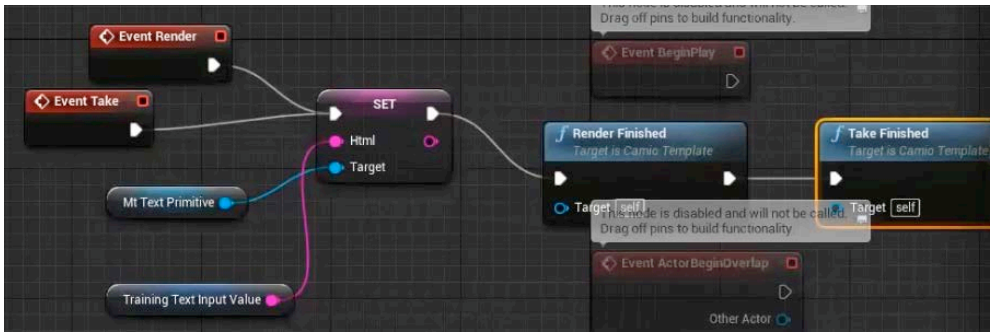
- Right-Click and create a ‘Set HTML’ Action.

- Right-Click and create a ‘Render Finished’ Action.

- Right-Click and create a ‘Take Finished’ Action.

Nodes Mapping

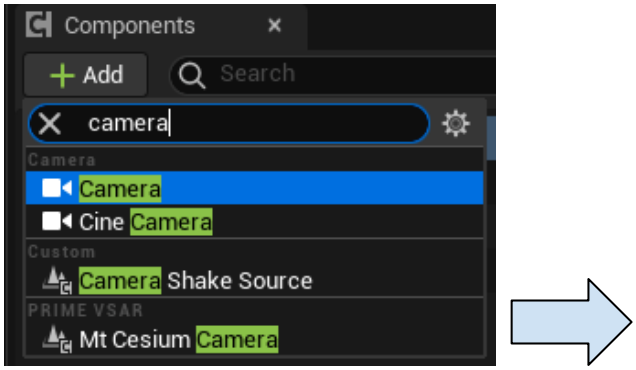
- Link all the blocks like in the picture below:

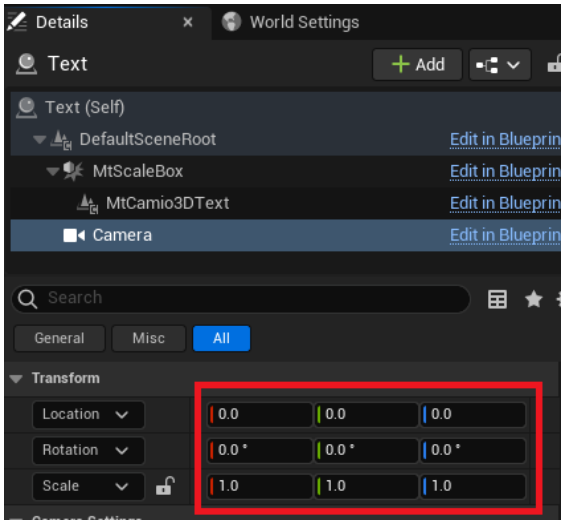


- Press ‘Compile’ and Save the scene.

Camera

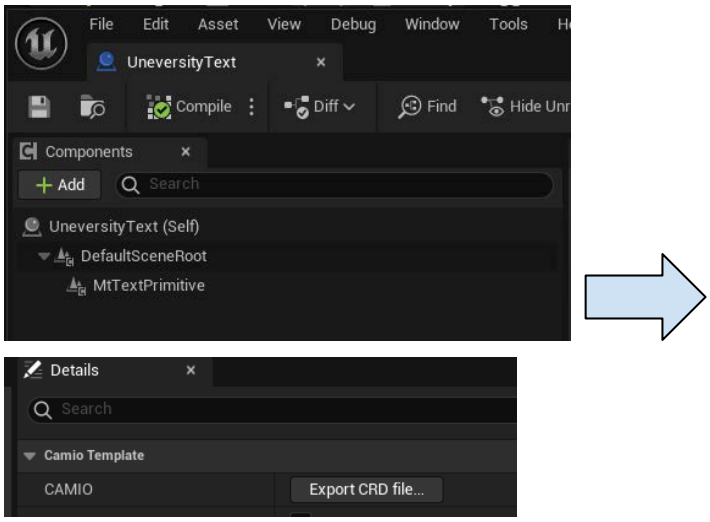
- Create a Camera Composition and positionate it using the right menu to capture the text object frontally.



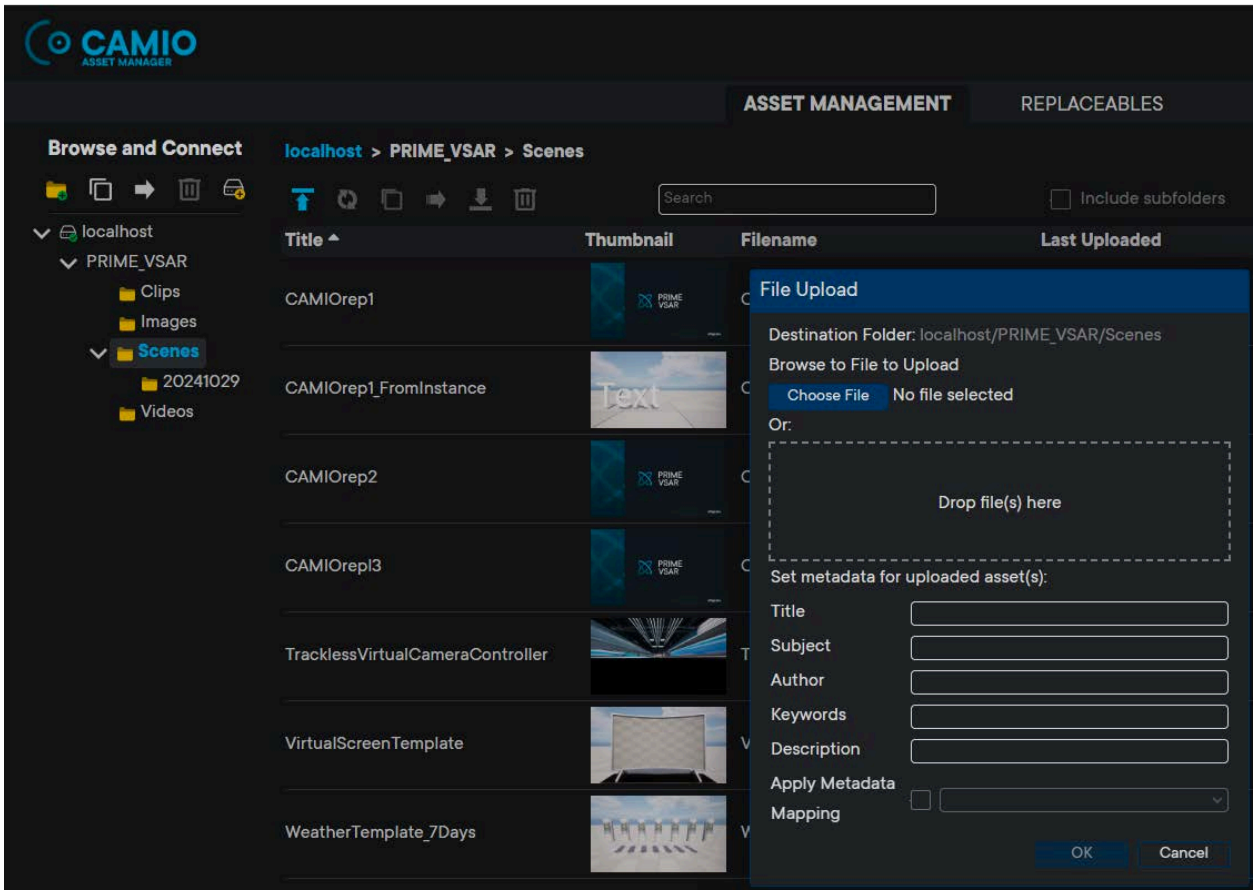


Scene import in CAMIO

- In order to export that Scene to CAMIO, select the corresponding component from the left-side menu and click ‘Export CRD file…’ from the right-side menu.

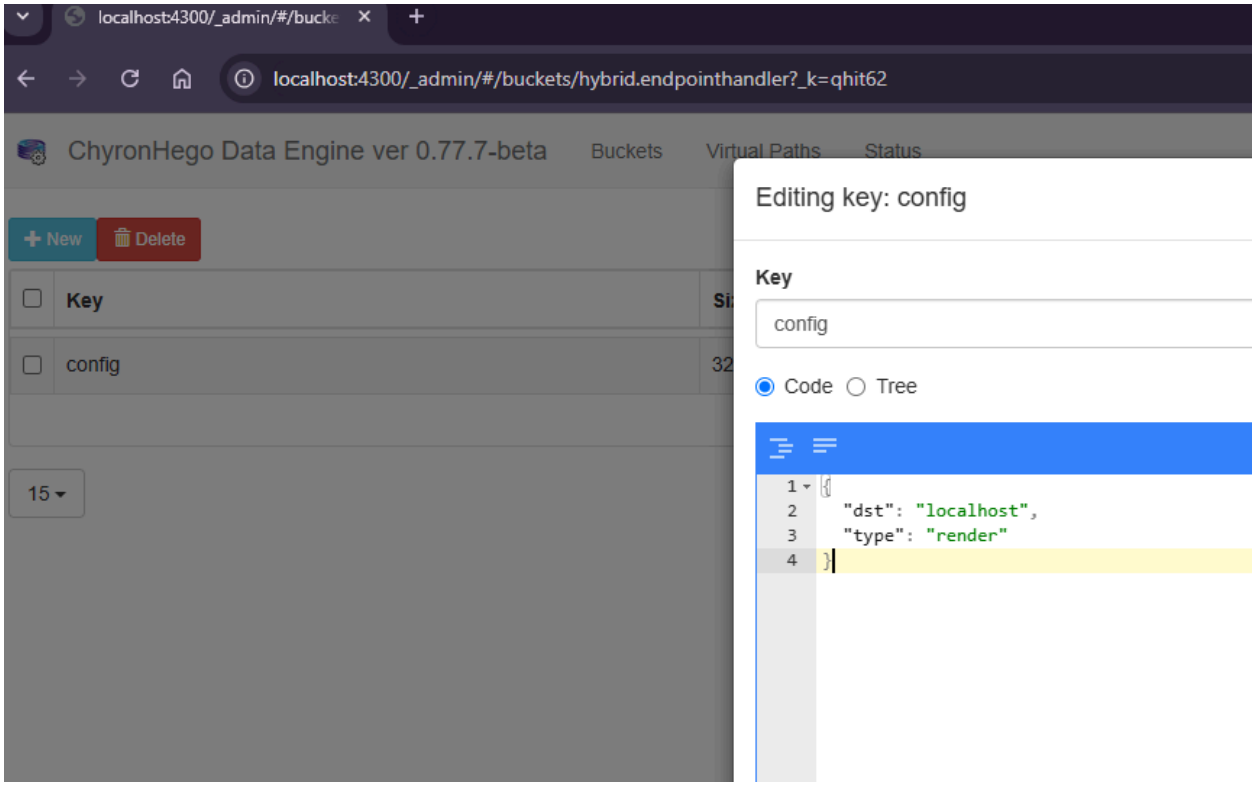


- Save the CRD file in any folder. Open CAMIO Asset Manager, select a Context and a sub-category, click the ‘Upload’ button and upload the CRD file.

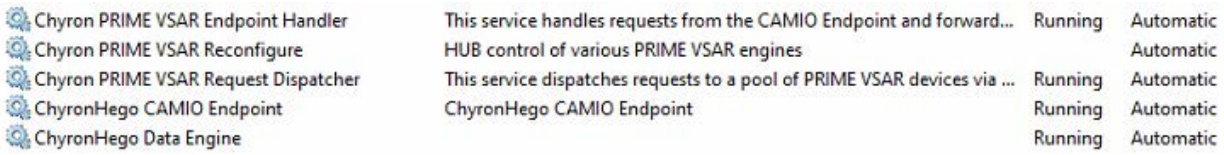


Test of the VSAR Scene Rendering with LUCI

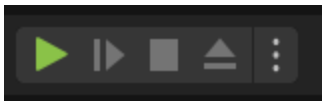
- Check that CAMIO Data Engine is configured correctly by opening the Data Engine web page and the settings of the 'hybrid.endpointhandler' bucket. it should be set as follow:



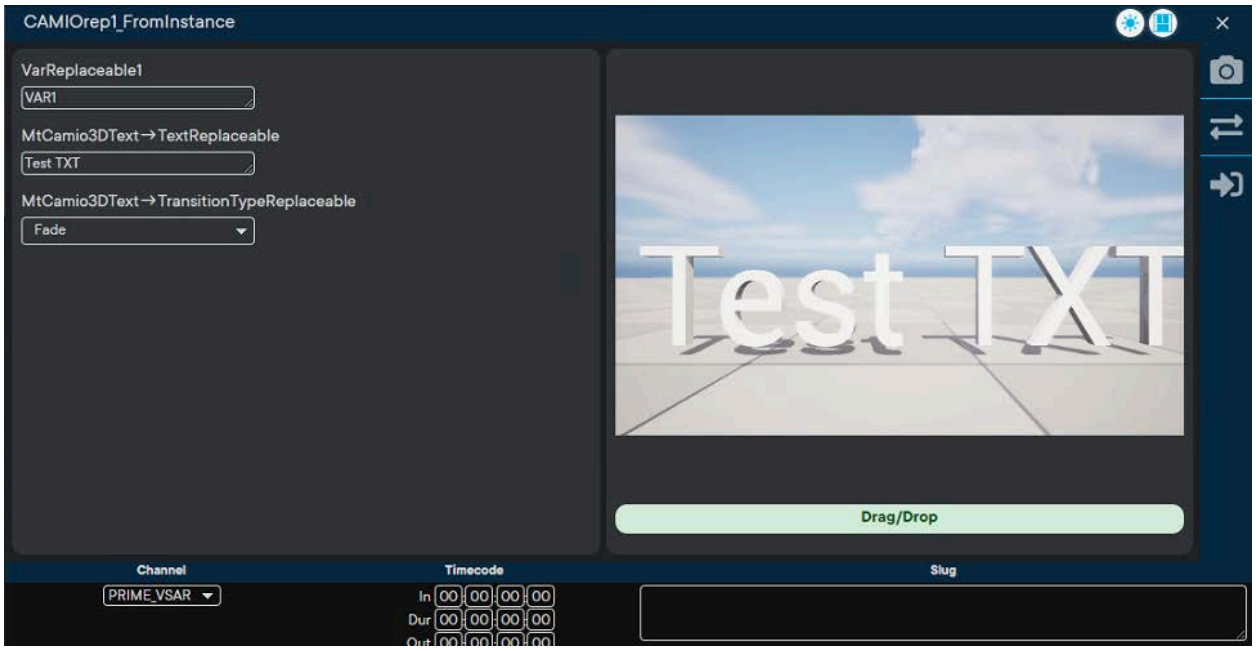
- Restart the 'ChyronHego Endpoint Handler' Windows service on the VSAR system.



- Press the 'Play' (▶) button on the Scene's window in VSAR.



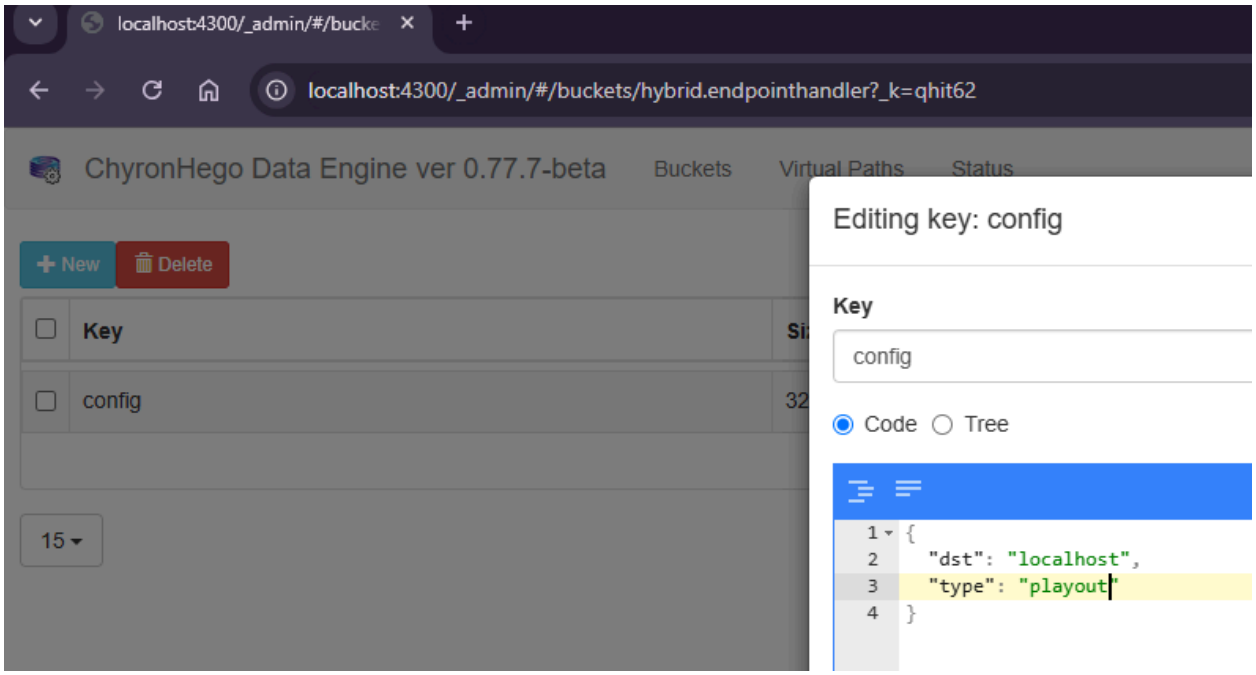
- Open the created Template in LUCI5, add some text in the available field and press the 'Generate Preview' button.



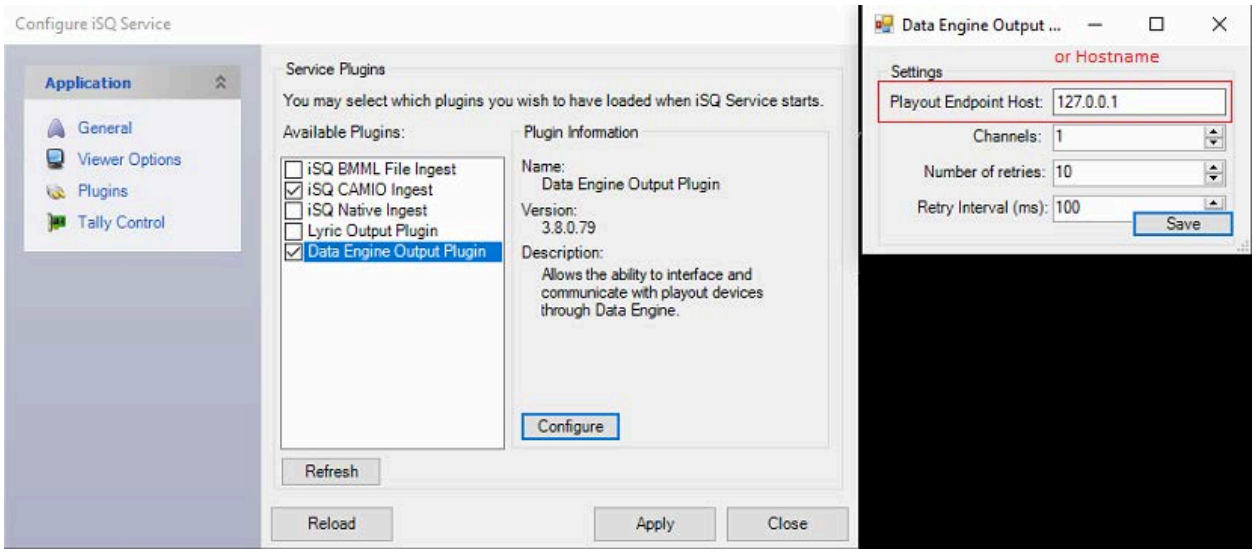
Test of the VSAR Scene Playout with ISQ

For such test you would need to have a full system including ENPS or iNEWS NRCS, CAMIO and ISQ available.

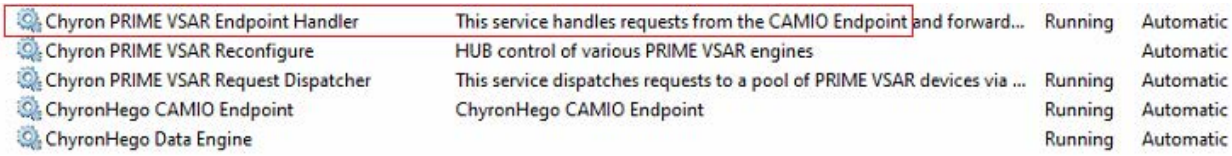
- Change the Data Engine's 'hybrid.endpointhandler' bucket settings back to 'playout':



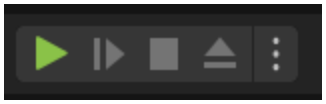
- In ISQ Service configuration page, under Plugins, select ‘Data Engine Output Plugin’, press ‘Configure’ and set the VSAR server hostname as Playout Endpoint Host.



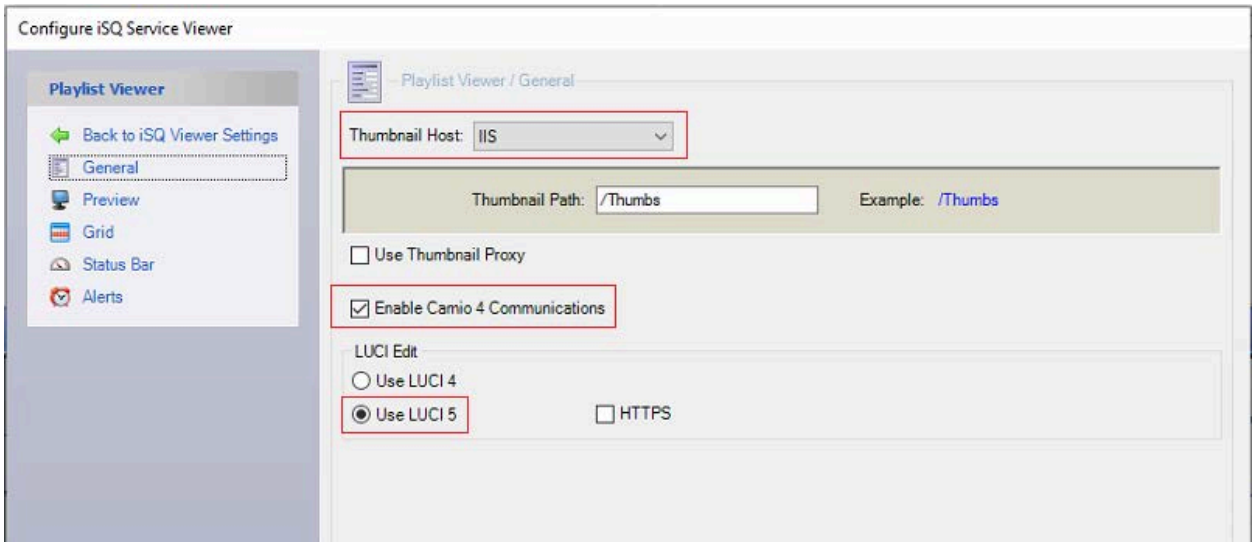
- Restart the ‘ChyronHego Endpoint Handler’ Windows service on the VSAR system.

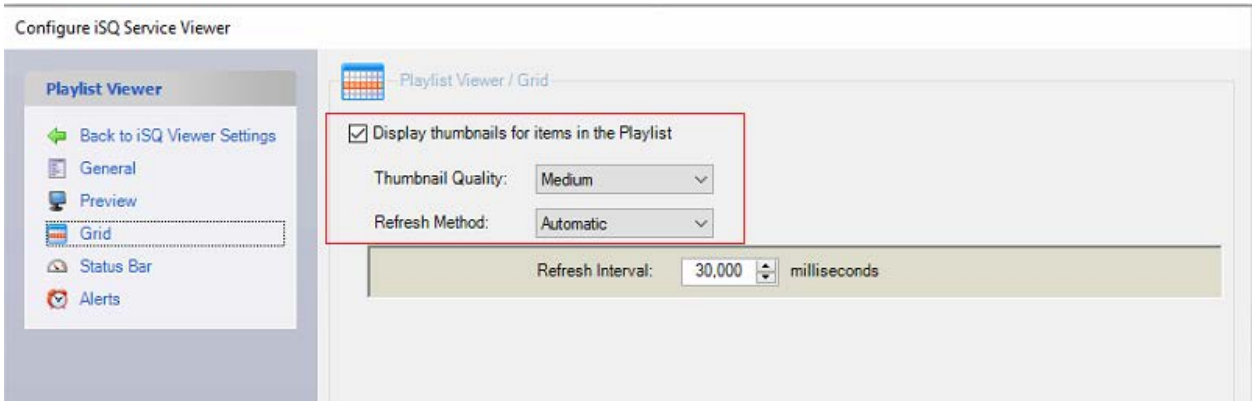


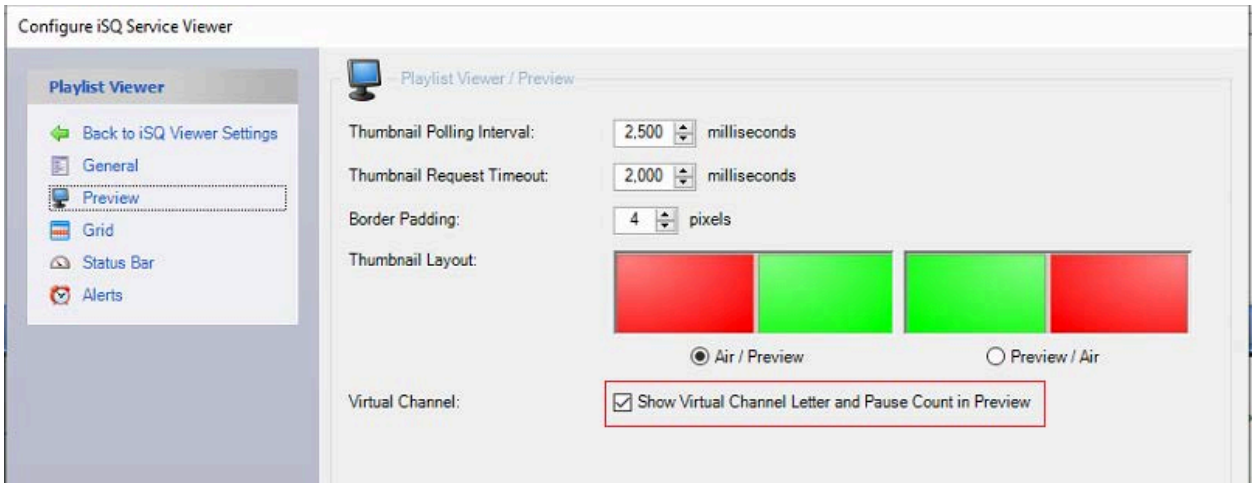
- Press the ‘Play’ button on the Scene’s window in VSAR.

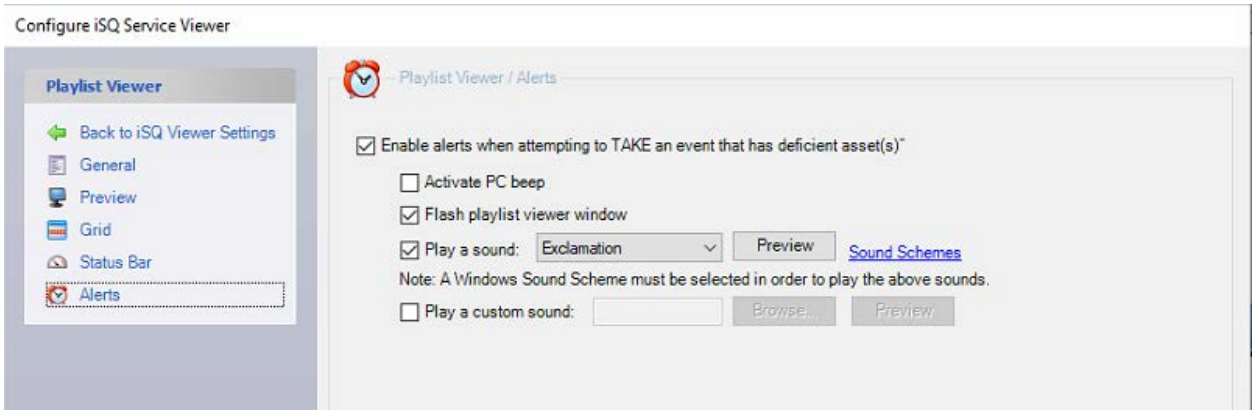


- Configure iSQViewer

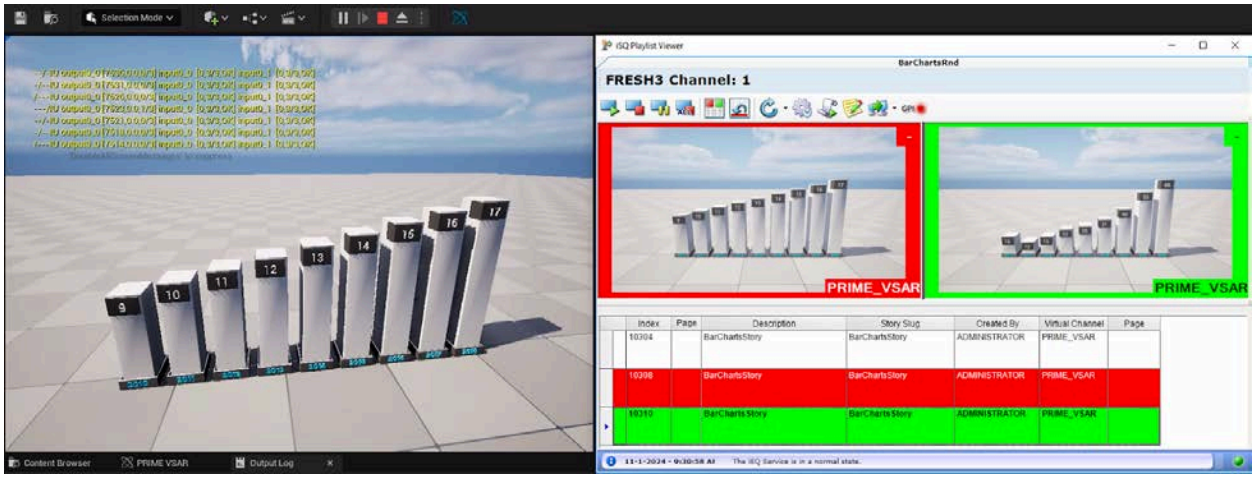








- From the NRCS connected to CAMIO and ISQ, create a Rundown, a Story and new Graphics from the VSAR Scene, load the Rundown to ISQ and Play.



LUCI Render

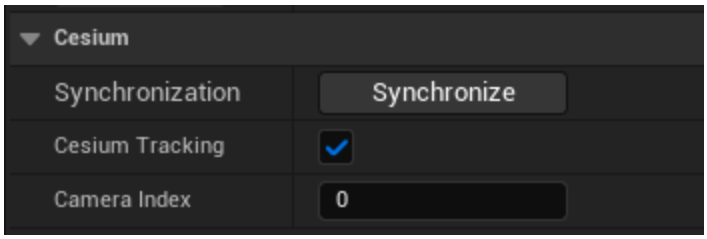
The render will use the Camera component (the base Unreal Camera, UCameraComponent) inside the CAMIO Blueprint if available.

Depth of field with Cesium

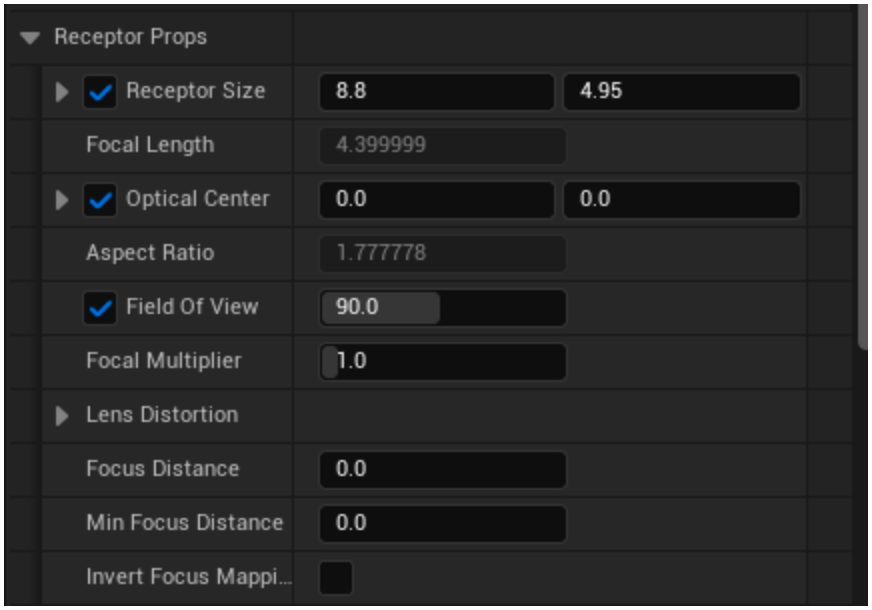
This section explains how to set up Depth of field in VSAR Cesium camera, which represents the Focus on real lens.

To enable depth of field with focus distance driven by Cesium:

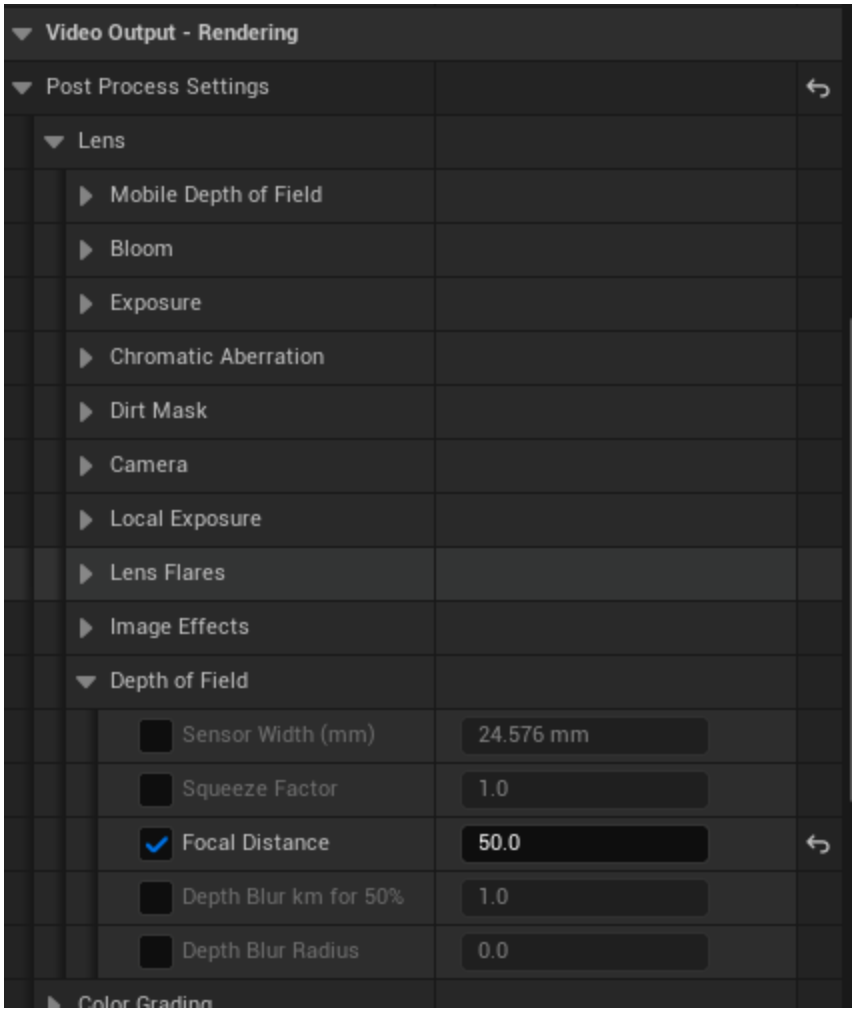
- Select the Cesium camera
- Enable Cesium Tracking



- If Cesium does not provide the actual focus distance directly (most of the time it's not the case), set the lens Minimum Focus Distance based on the real lens specs (in meters, eg. 0.5m)



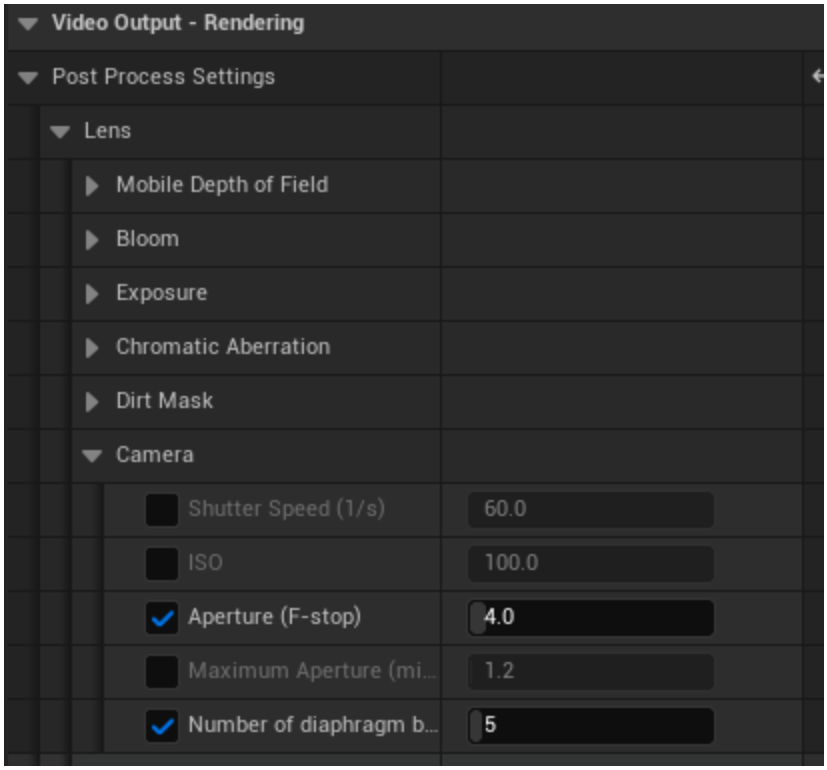
- Enable Focal Distance post processing (its value is displayed in cm and is automatically filled from cesium data)



- Optional: depending on how the focus distance is mapped, it may be necessary to invert the mapping.

By default, min focus distance is mapped to an encoder value of 0 and infinity (~10km) to 1.

- Other options are available, such as Aperture (F-stop) which can increase or decrease the blur effect, change the look of the bokeh. Smaller F-stop values increase the blur.



Video tutorial

DOF with Cesium.mp4

Lens distortion with Cesium

When connected with Cesium, the rendering from VSAR can be altered to simulate the real lens distortion.

Lens distortion modes

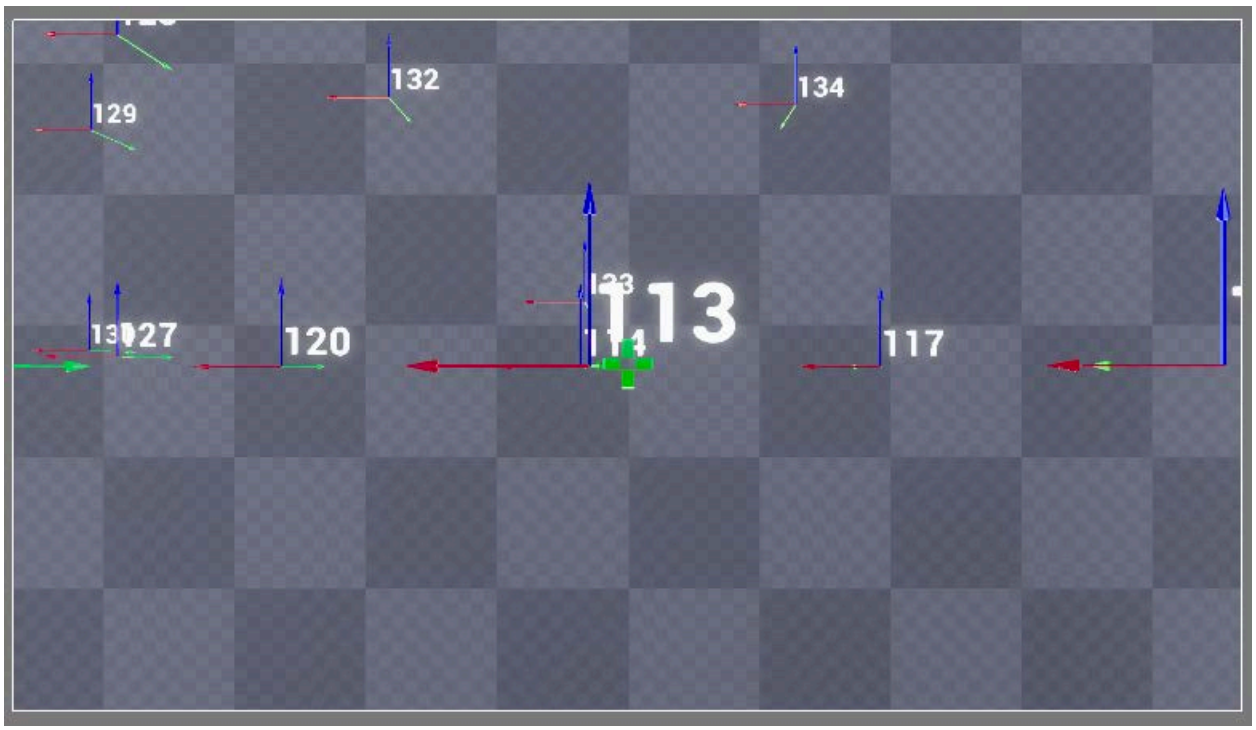
There's multiple ways to model the lens distortion, VSAR simulates radial distortion with the following modes:

- "Radii Array", an array of radii (pair of source radius and destination radius, up to 10 pairs).
- "K3 Coefficients" (default), a polynomial representation of radial distortions where:
  rsrc = rd*(1+k1.rd^2+k2.rd^4 + k3.rd^6)
    - `rsrc: source radius ie. undistorted radius`
    - `rd: distorted radius`
    - `k1,k2,k3: radial distortion coefficients`

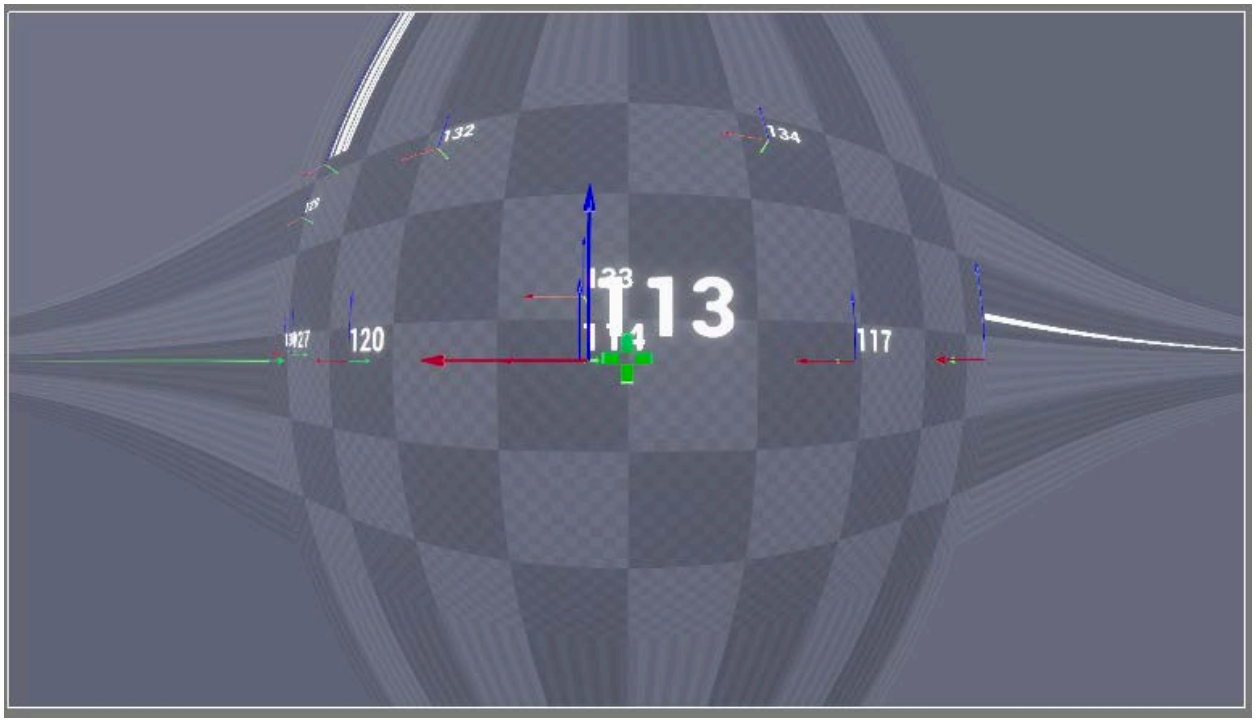
This model is used by Stype for example and is more precise than “Radii Array”.

To enable lens distortion, select a CesiumCamera and check the box “Enabled” in the “Lens Distortion” menu.

The mode is automatically selected depending on the data received from Cesium. To force a specific mode, check the box “Mode” then select one mode in the drop down list:



No distortion



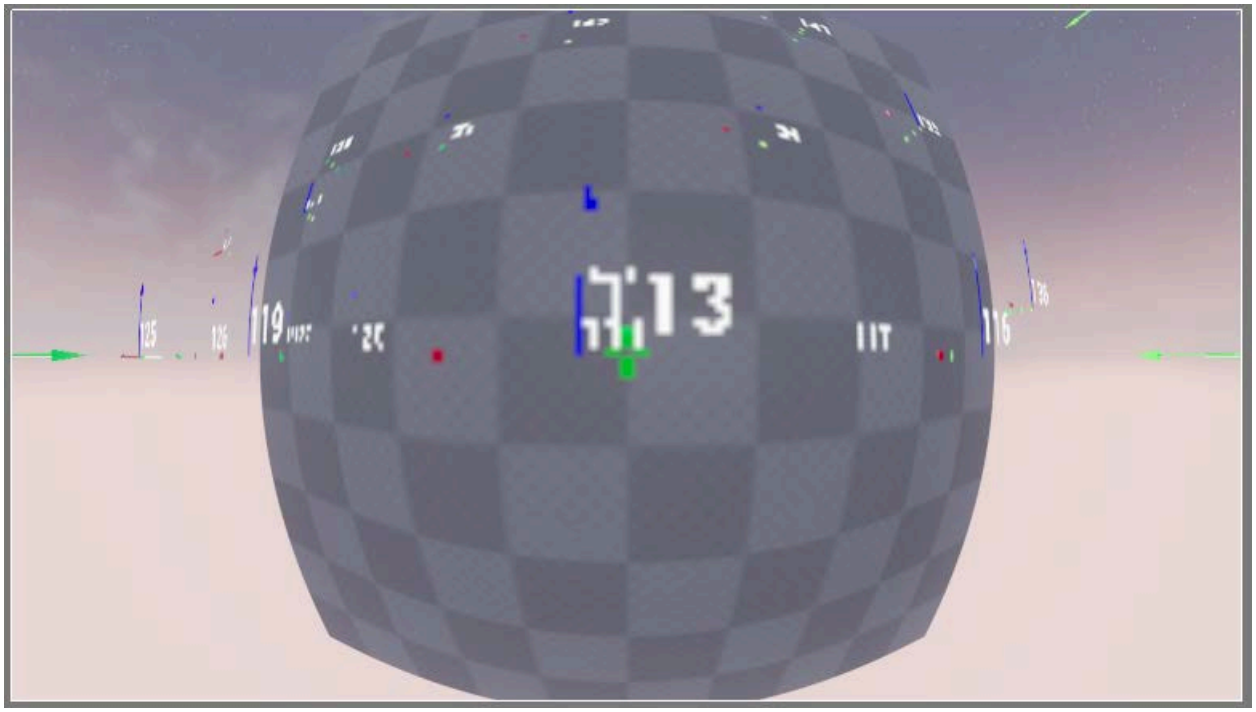
Barrel Distortion simulated (Exaggerated for visualization) with no Extra Focal

Auto extra focal

VSAR automatically computes and applies an extra focal multiplier (a.k.a. overscan factor) needed to calculate the area “outside” the distorted render. This removes “bleeding edges” on the final output.

Note: On severe lens distortion, this setting can introduce blurriness as the rendering is upscaled.

To force a specific extra focal multiplier value, check the box “Extra Focal Multiplier”, then input a value (eg. a value of 1.0 will cancel the extra focal effect):



“Bleeding edges” removed, the blurriness is only noticeable with heavy distortion (Exaggerated for visualization)

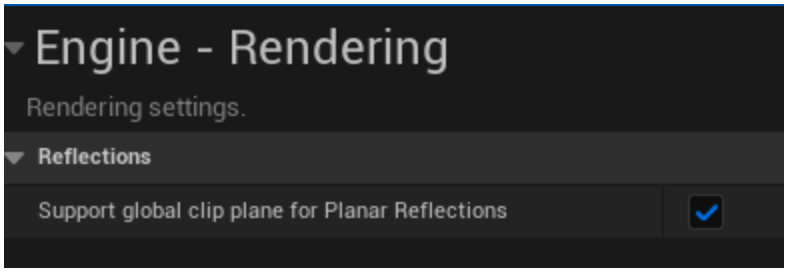
Planar Reflection in Trackless

This section describes how to create planar reflection in the Trackless scenario. The Planar reflection provided by Unreal does not work with “SceneCapture” which is a component used by PRIME VSAR cameras, so the “VR Reflection Plane” object has been developed to work around that limitation.

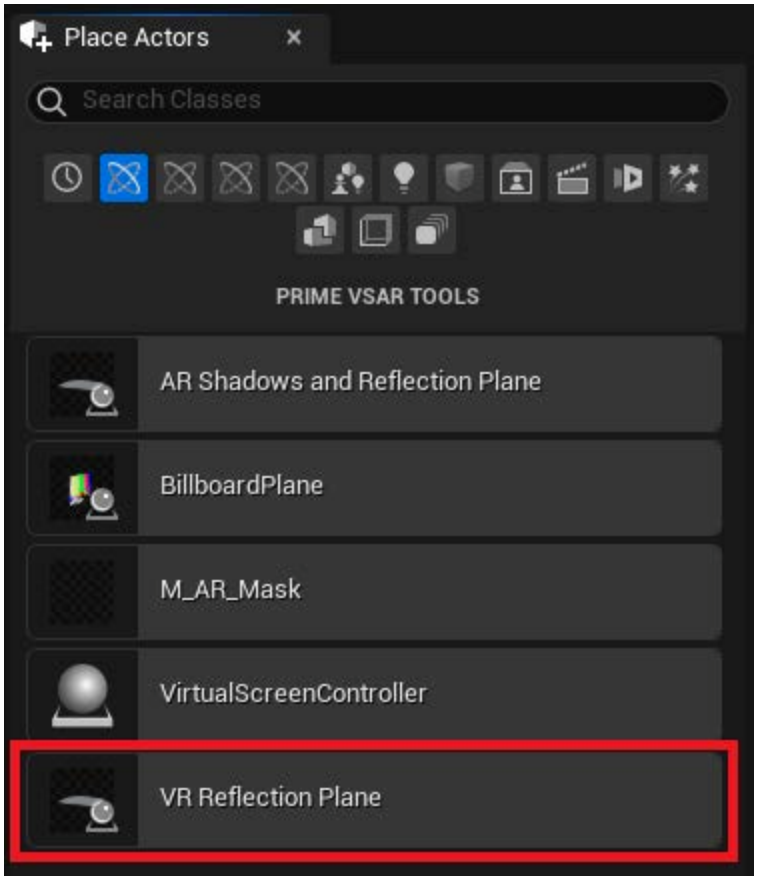
Setup

1. In the project’s settings, enable “Support global clip plane for Planar Reflections”.

📝 Note that this will restart the editor and recompile all the project’s shaders. According to Epic, it will also increase the BasePass triangles cost by ~15% regardless of whether planar reflections are active.



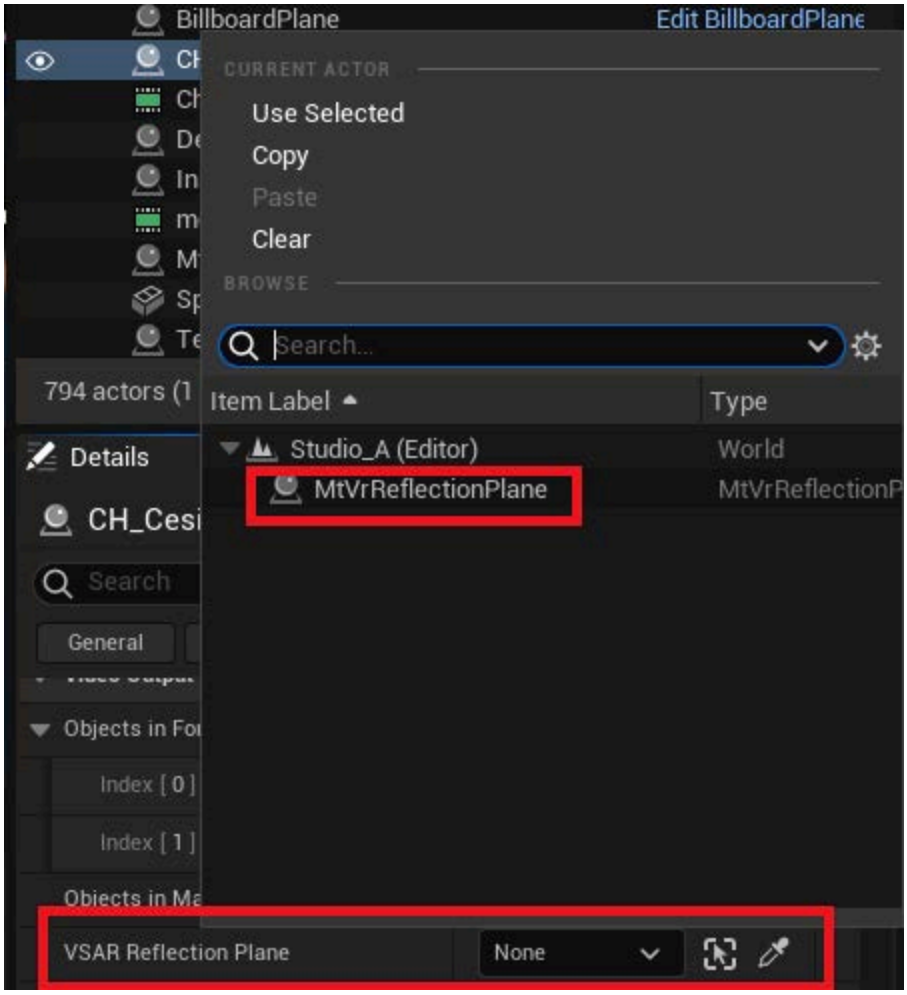
2. Create a VR Reflection Plane object, its purpose is to render the planar reflection from Cesium Camera point of view. Search for “VR Reflection Plane” and drag and drop the object in the level.



3. Adjust the height (Z axis) of the VR Reflection Plane in order to match the floor’s height and place it slightly above it in order to avoid collision (aka “Z-fighting”).

The following step should be done for each Cesium Camera where the effect is visible:

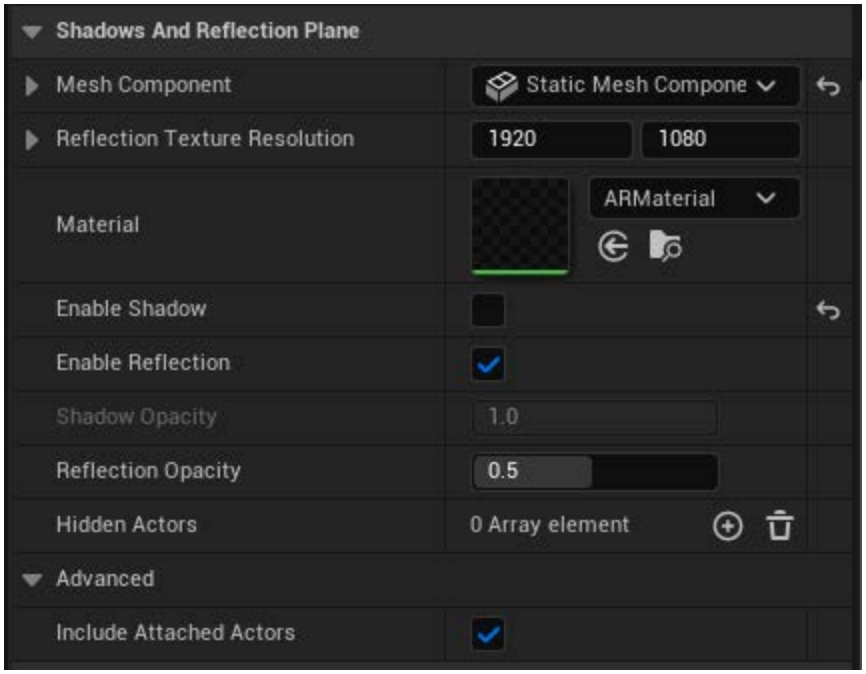
4. Select the Cesium Camera (create one if it is not already done) and bind the VR Reflection Plane object via the Details Panel at the section Video Output Key:



📝 The reflection is not displayed according to the Editor viewport's camera. In order to visualize the final result, check the Cesium camera output in the VSAR Config panel or the video output.

Parameters

The VR Reflection Plane object has the following parameters to alter the rendering.



- Mesh Component: The default Static Mesh is a plane. This property is accessible in case of custom shape (e.g. reflection on a rounded surface).
- Reflection Texture Resolution: The resolution of the texture used for the reflection pass.

Default: 1920x1080

- Material: For advanced usage with custom Materials (e.g. blur, alpha mask...),
- Enable Shadow: Enable/Disable shadow, default value is Disabled. (It should not be used in Trackless mode)
- Enable Reflection: Enable/Disable reflection, default value is Enabled.
- Shadow Opacity: Affects the shadow opacity, 1 is fully opaque and 0 is fully transparent. (It should not be used in Trackless mode)

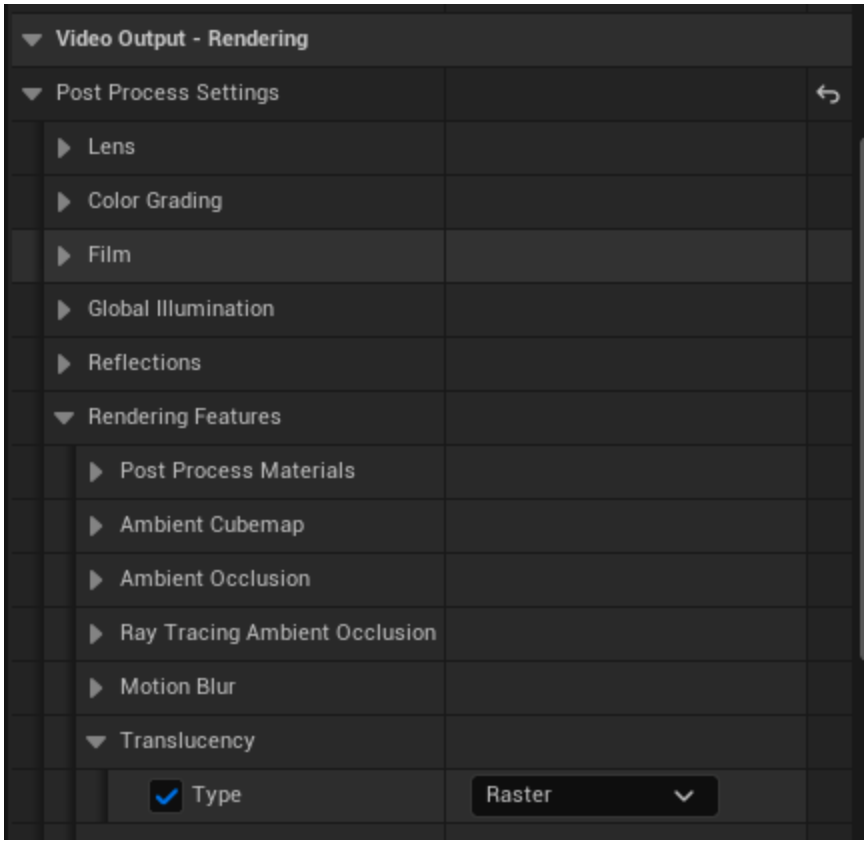
Default: 1

- Reflection Opacity: Affects the reflection opacity, 1 is fully opaque and 0 is fully transparent.

Default: 0.5

- Hidden Actors: List of Actors excluded from the reflection rendering pass, intended for performance optimization.
    - Include Attached Actors: Also hide the Actors attached to the hidden Actors, default value is Enabled.

⚠ VR Reflection Plane does not work when using Ray tracing: in Cesium Camera → Output Component → Video Output - Rendering → Post Process settings → Rendering features → translucency → type should be set to “Raster” instead of “Ray Tracing”



VSAR Web

The focus of VSAR Web is to provide a web based user interface for controlling VSAR in a more intuitive streamlined way. It's designed as an alternative to Live Assist Panels (Chyron Panels) for controlling VSAR.

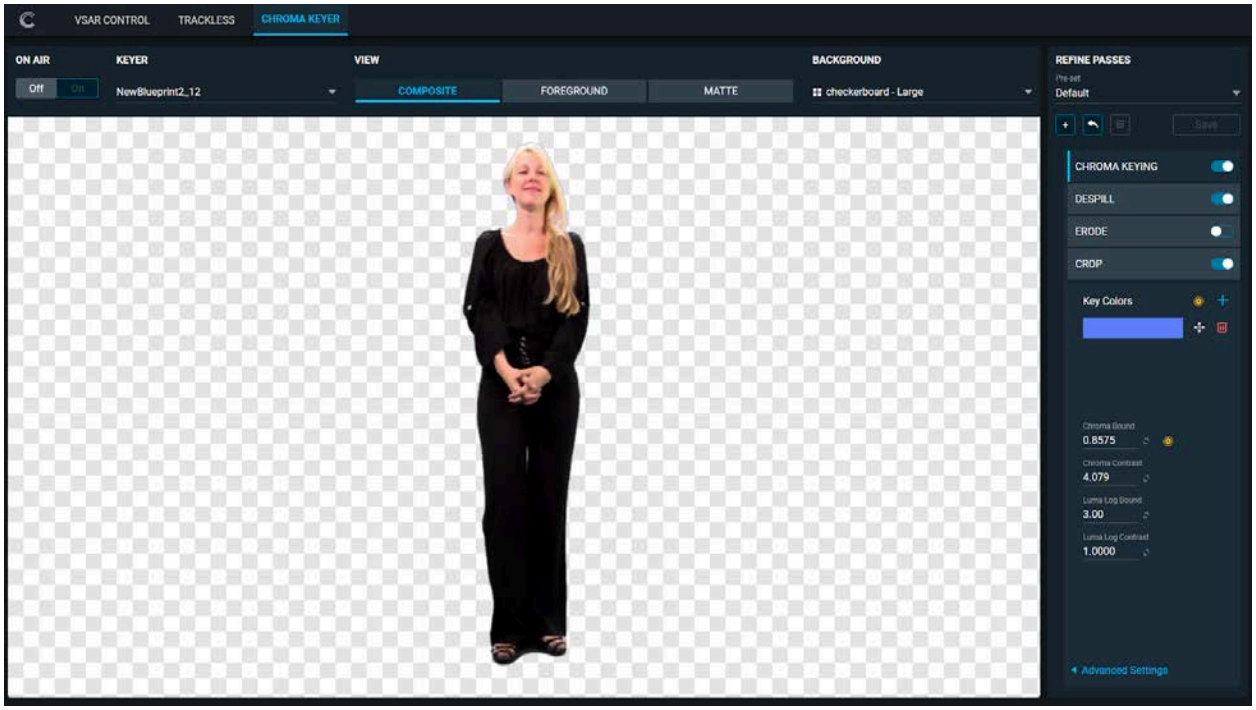
By default VSAR Web can be accessed on **http://localhost:8080/** or on local network with the ip address of the machine followed by the port (:8080)

📝 Supported browser is Chrome

⚠ Please note that VSAR Web is currently not yet optimized for Mobile/Tablet devices.

Chroma Keyer

The CHROMA KEYER tab can be found in the top left section of the web page. The Chroma keyer panel provides the ability to remotely control **Internal Chroma Keyer**.



On air

While the On Air mode is on, the preview window does not update this to alleviate performance requirements for VSAR. On Air mode gets automatically activated when VSAR is in Play mode. The On Air mode can be manually overridden by clicking on the switch manually.

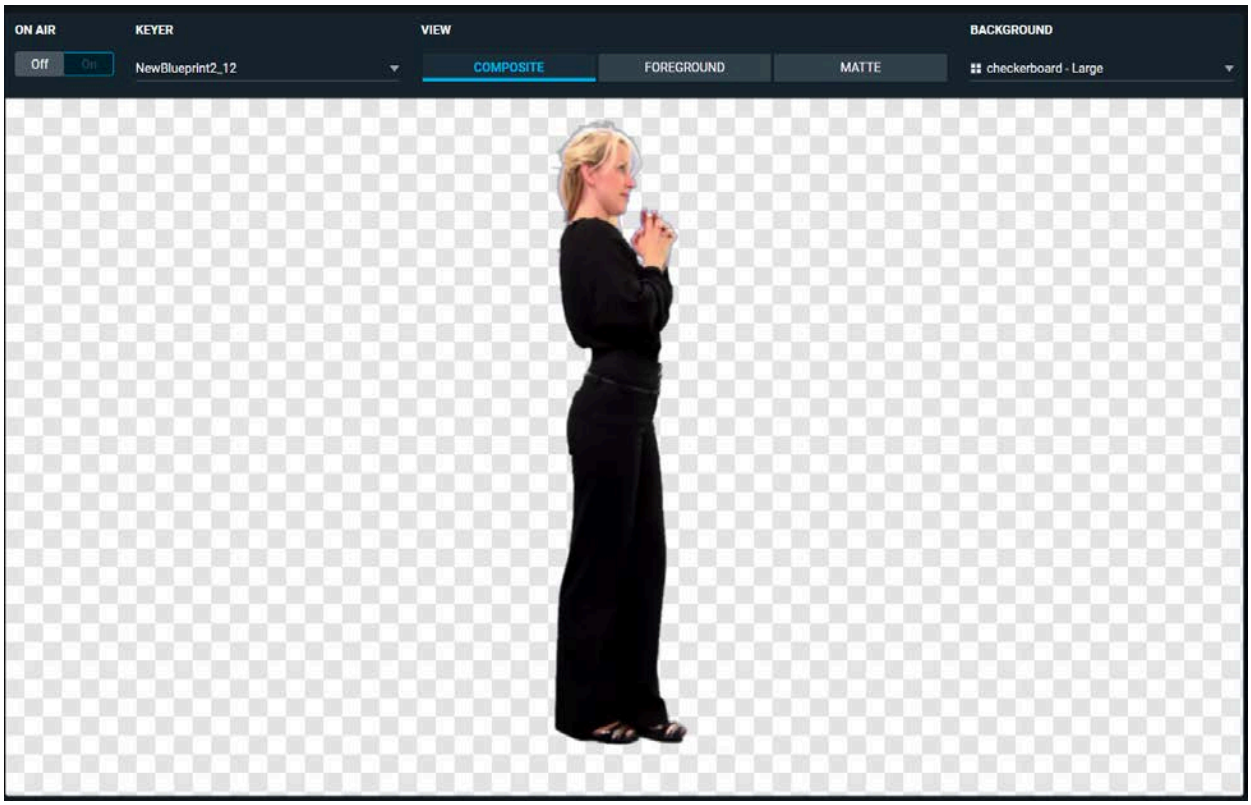
Keyer

In this dropdown a Chroma keyer that you wish to control is selected. if multiple VSARs (connected to VSAR Controller) have a Chroma keyer with the same Actor ID they are considered the same.

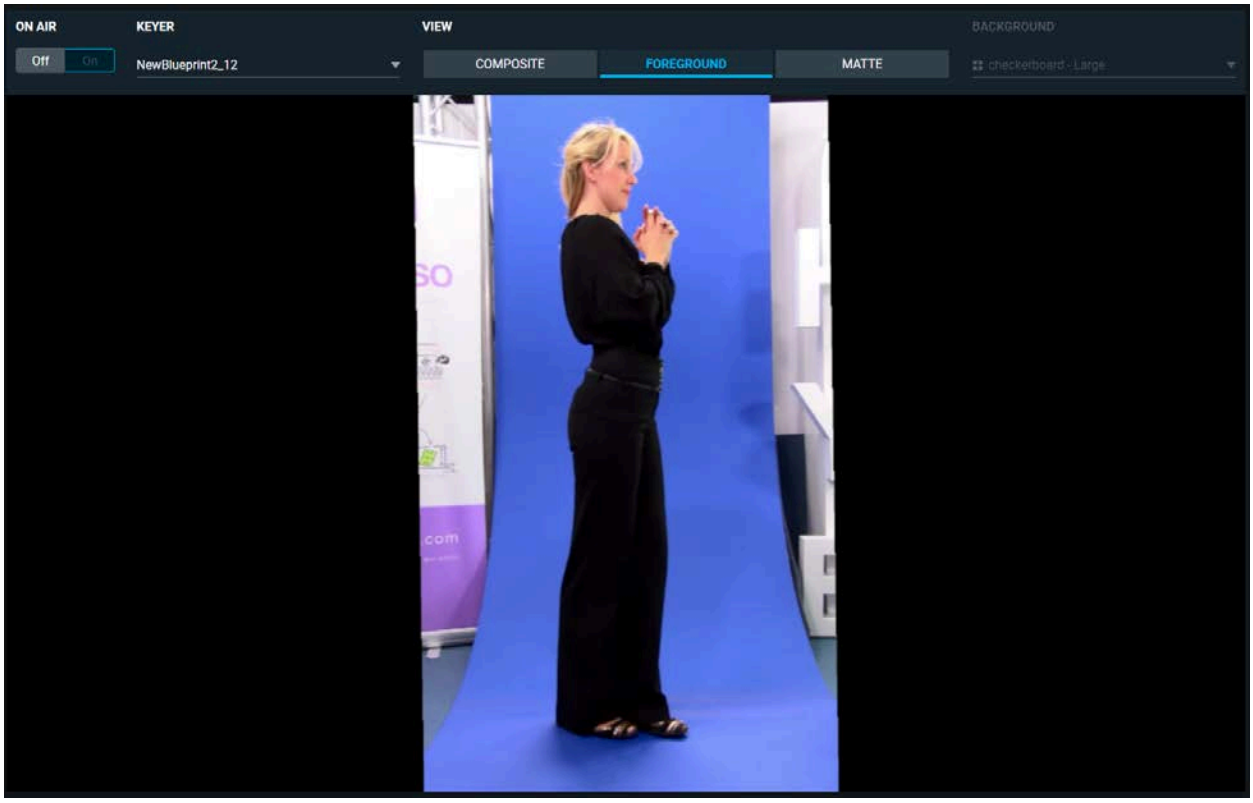
View

View mode gives the ability to change preview viewing mode.

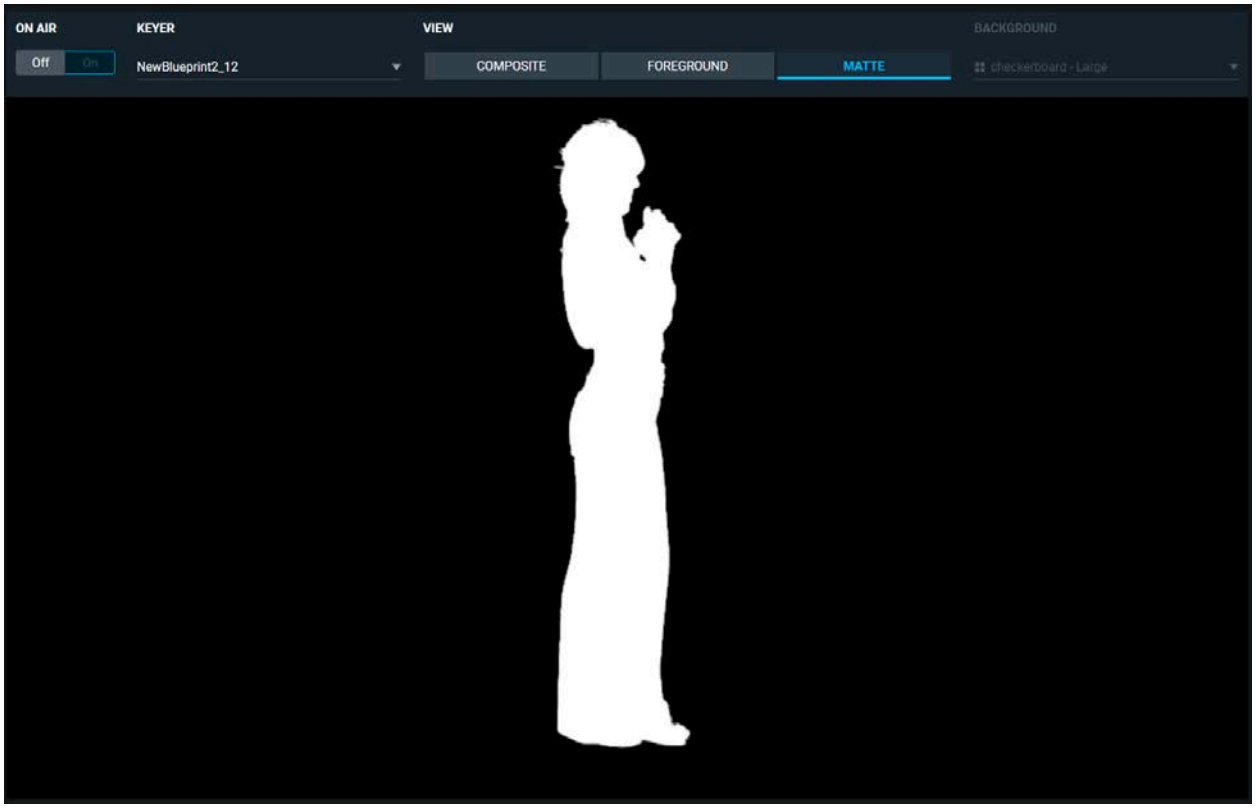
Composite - it's the Output of the Chroma Keyer



Foreground - it’s the unmodified Input of the Chroma Keyer

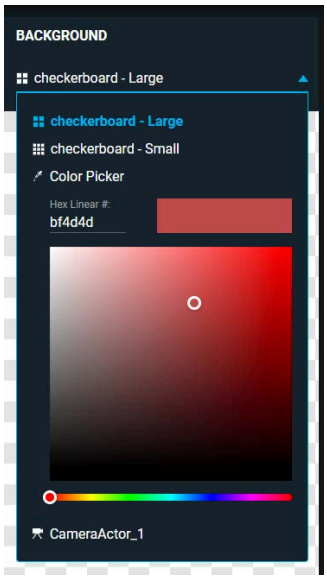


Matte - is the Alpha channel visualized in grayscale

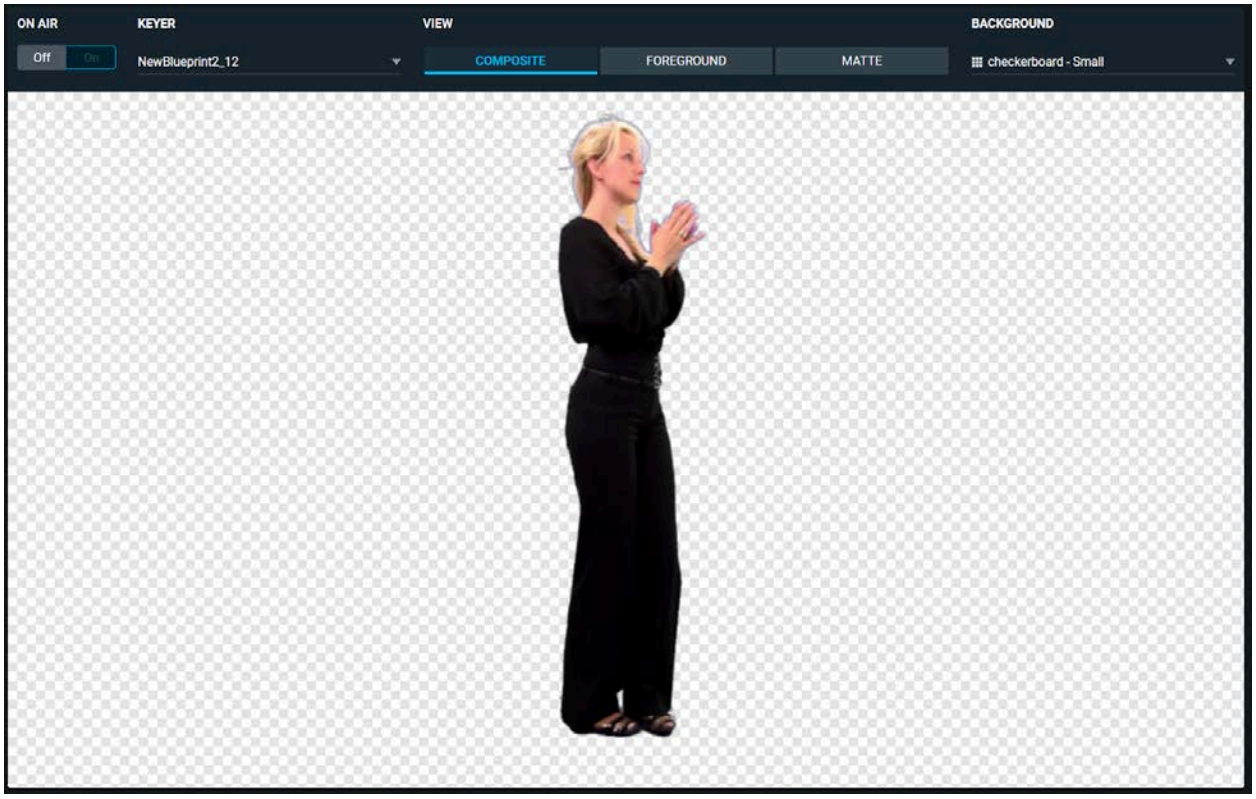


Background

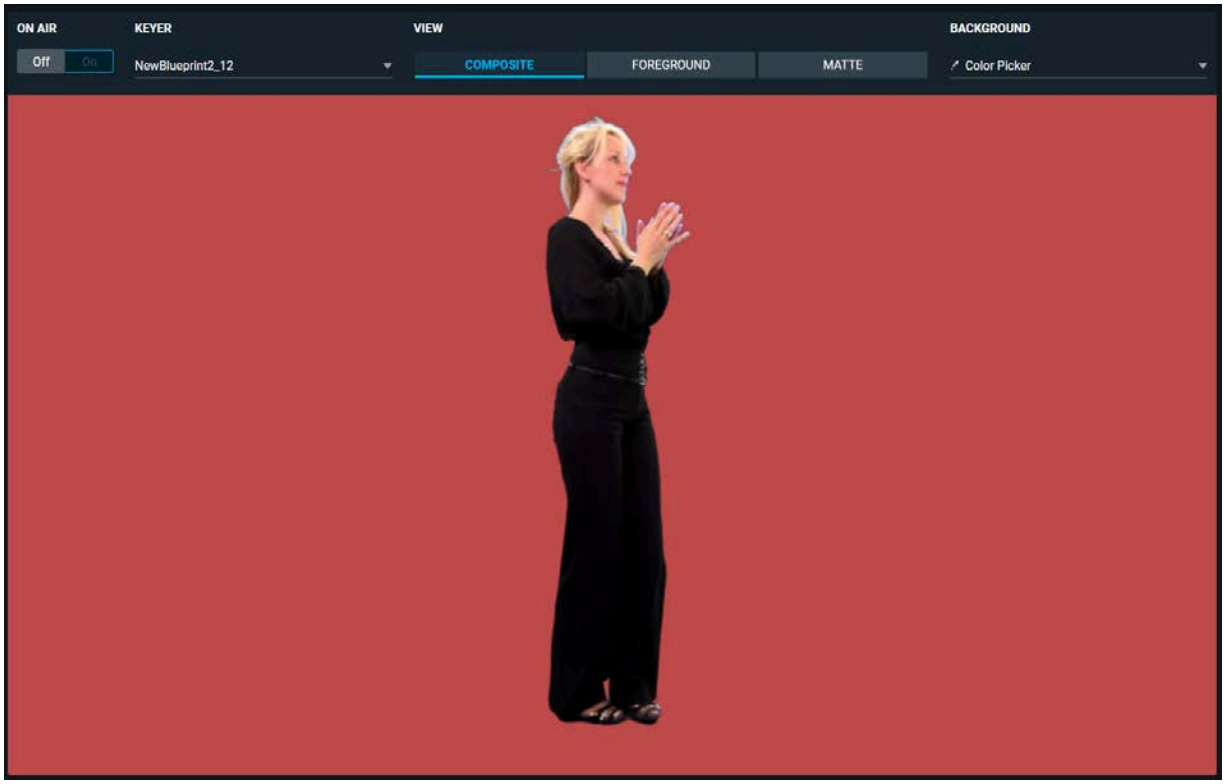
Dropdown for selecting background for Composite View. Background works only for Composite View mode and is only visible in VSAR Web.



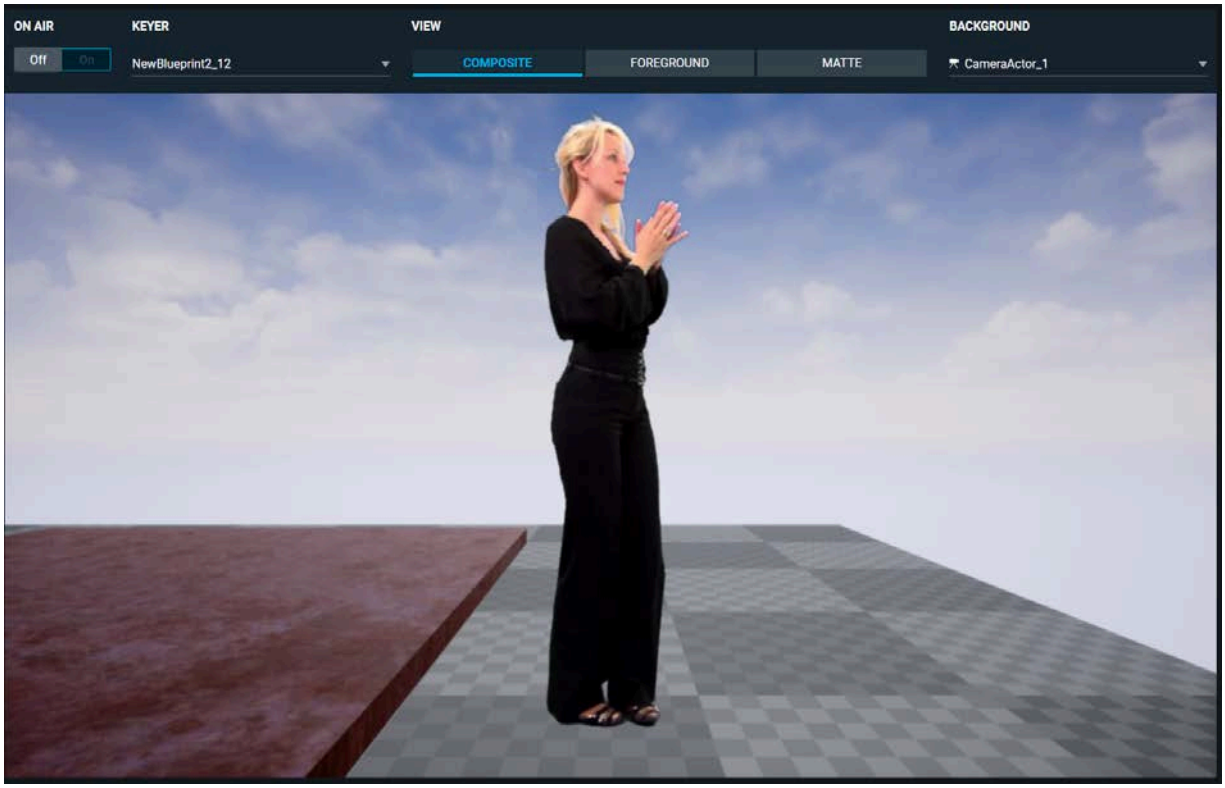
Checkerboard - Small/Large



Color- simple color picked by the color picker



Camera - image from camera in a VSAR (automatically detects new cameras in levels)

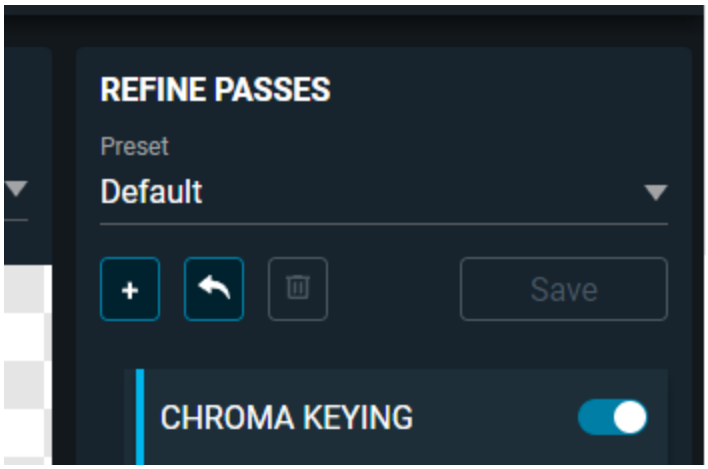


Refine Passes

This section is for editing key and values of a chroma keyer.

Preset

Preset section is for saving values of chroma keyer.

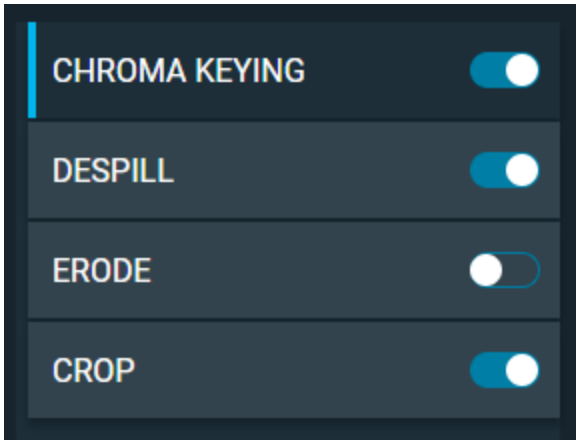


- Create present - (the [+] icon) allows the creation of a new present from current chroma keyer values.
- Revert present - (the [⟵] icon) allows to revert current values to preset values. If the selected preset is Default, values are reverted to default values of the chroma keyer.
- Delete preset - (the [🗑] icon) allows to delete current preset, default preset can't be deleted.
- Save - saves changes to current preset.

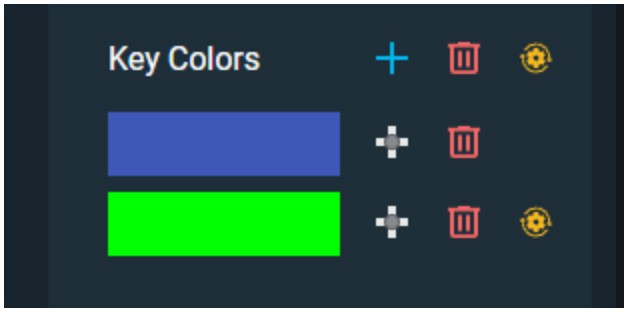
Presets are stored in VSAR Project content next to a Chroma keyer asset as Data Table, with the suffix "_Presets". This asset is considered unsaved after changes/creation by Unreal Engine, if you wish to keep these presets save this asset before or on project close.

Passes

This section allows you to select for what chroma keyer pass the values are edited. more on what each pass is doing can be found **here**.



Individual passes can be disabled/enabled with the small switches on the right side.

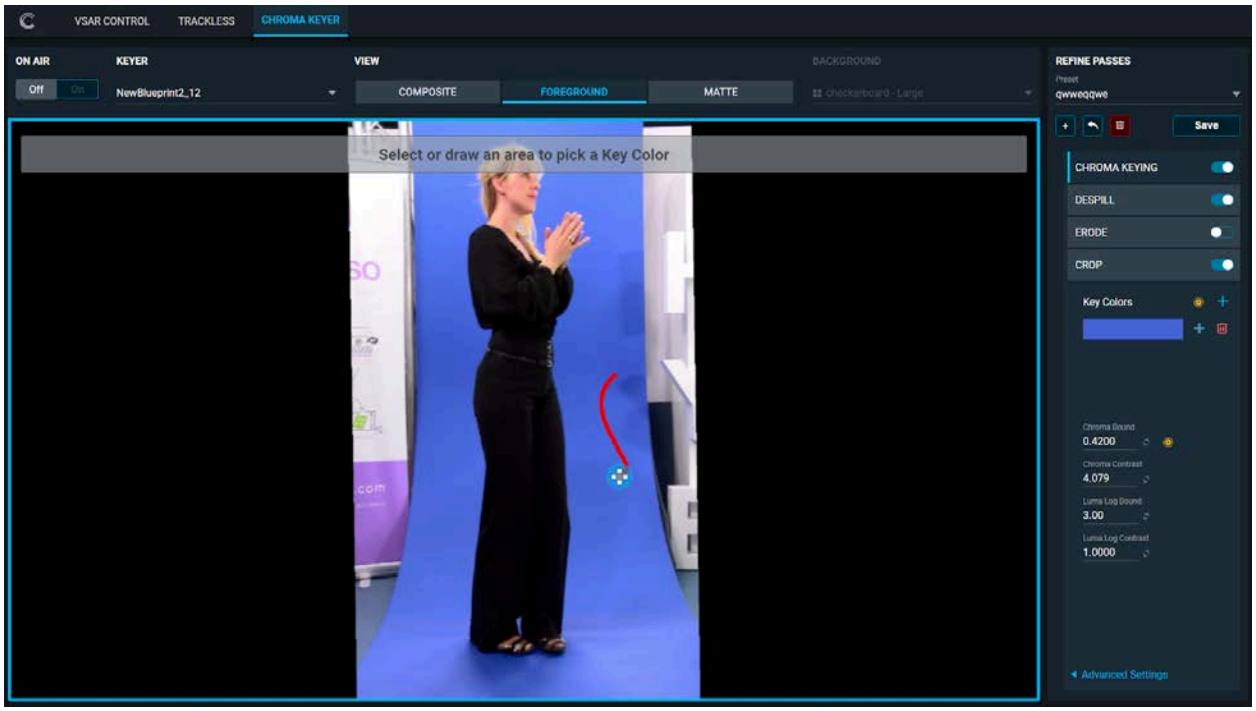


For Chroma keying and Despill passes Key Colors can be selected, both allow for multiple keying colors.

- New key color can be added with the blue [+] icon.
- All key colors can be reverted to default with the yellow gear icon.
- Clicking on the color bar opens color adjustments pop up.
- Clicking on the Gray [+] icon, starts a draw pick.
- Clicking on the the [🗑] icon removes the key color.

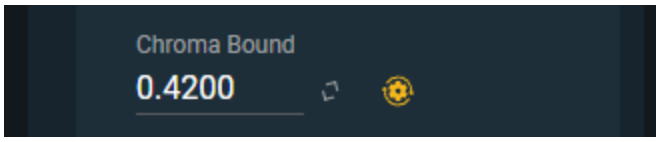
Draw pick

When selecting color through the draw pick you can click on the preview to select a single point or click and hold the left mouse button to draw, this selects an array of points that gets averaged resulting in a single color.

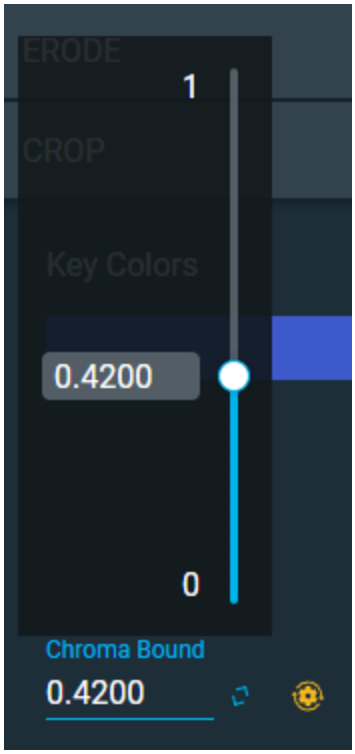


Values

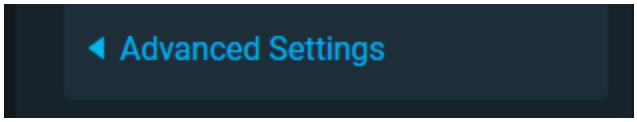
This section is for editing individual pass parameters.



Each input field has a name, value section where a new value can be entered, slider button and a reset button.



The slider button opens a slider popup that allows changing the value with the range slider.



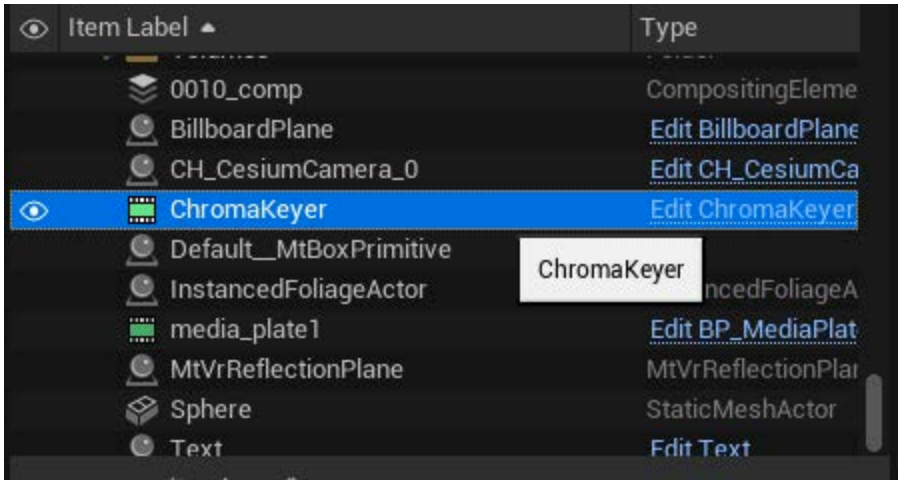
Chroma keying pass has advanced settings that can be revealed by clicking on the advanced settings button.

Settings for ranges of values and what values are considered advanced are in VSAR Web settings config.js, more on where to find config is in **Setup guide**.

Synced Chroma keyers

When multiple machines are connected to VSAR Controller **(VSAR Controller service configuration)**

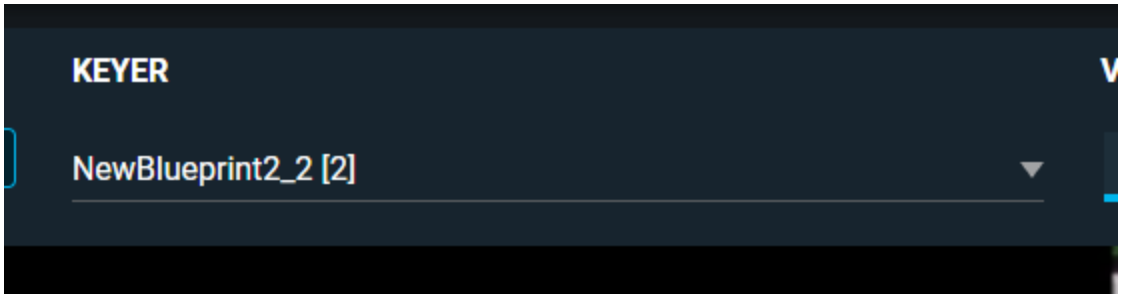
Chroma keyer supports synchronized workflow where a single chroma keyer represents chroma keyers on multiple machines this happens while there is a chroma keyer with the same Actor ID.



("ChromaKeyer_2" in this example)

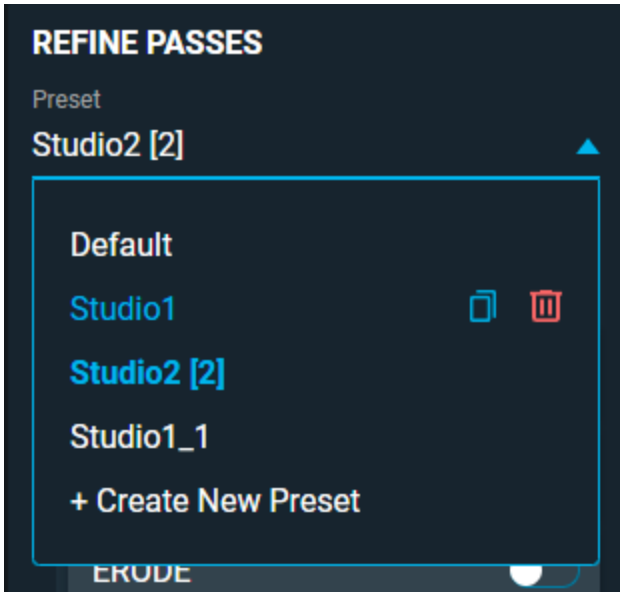
The idea is to have a copy of the project on multiple machines to support multiple camera inputs and control a single chroma key across them.

When keyers are in sync there is numeric indicator next to them how many machines it's representing



(the [2] indicating two in this example)

While in this mode presets are stored for multiple projects simultaneously and work the same as usual. There are cases where presets can be desynchronised, then the same numeric indicator "[number of machines]" would show for the synchronized presets and the desynchronised get additional "_index"

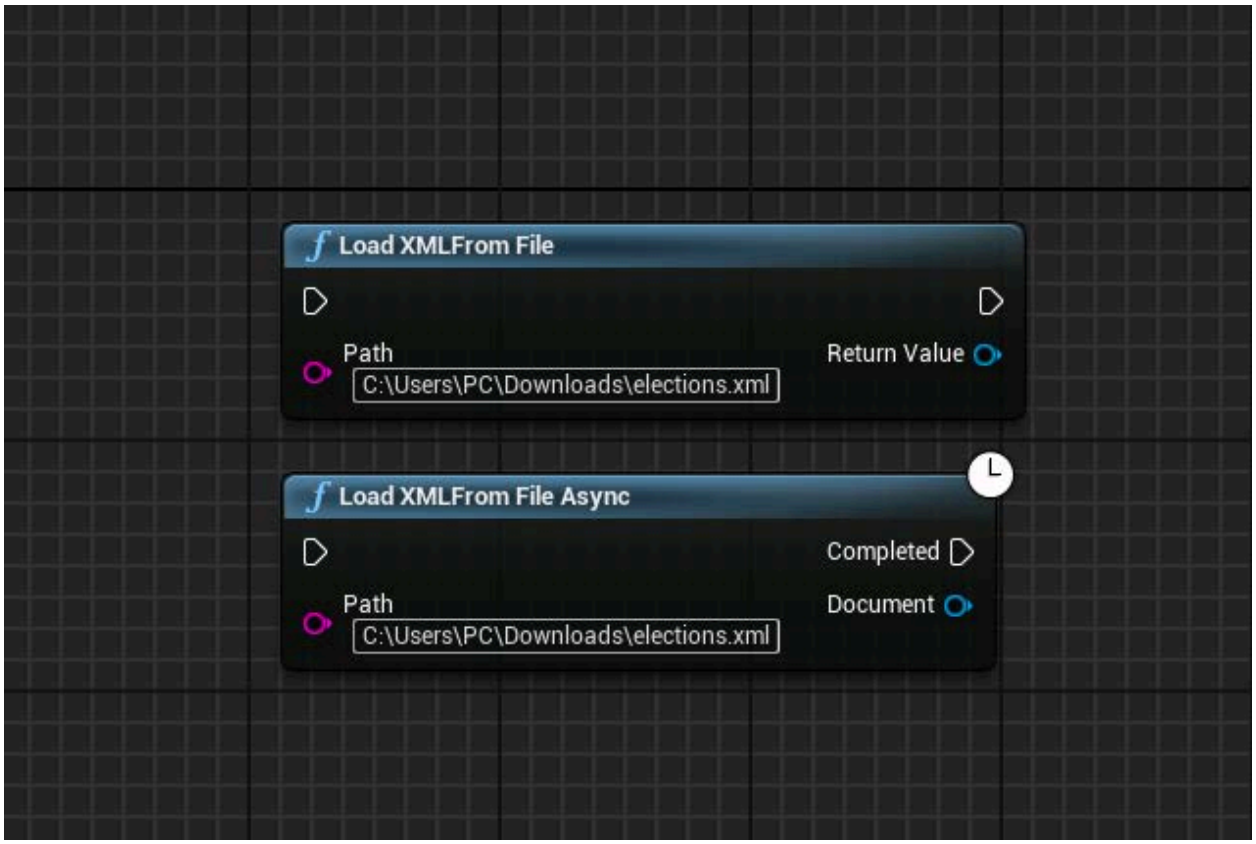


In this example, Studio2 preset is synchronized and the Studio1 preset is desynchronised creating a second preset instance Studio1_1.

Data Binding

XML

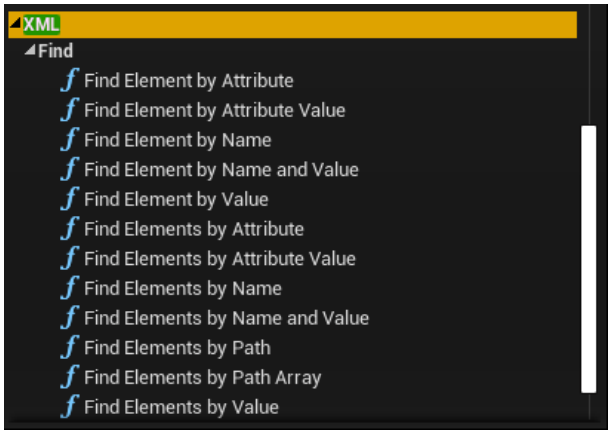
You can load XML from file using the “Load XML From File” or “Load XML From File Async”



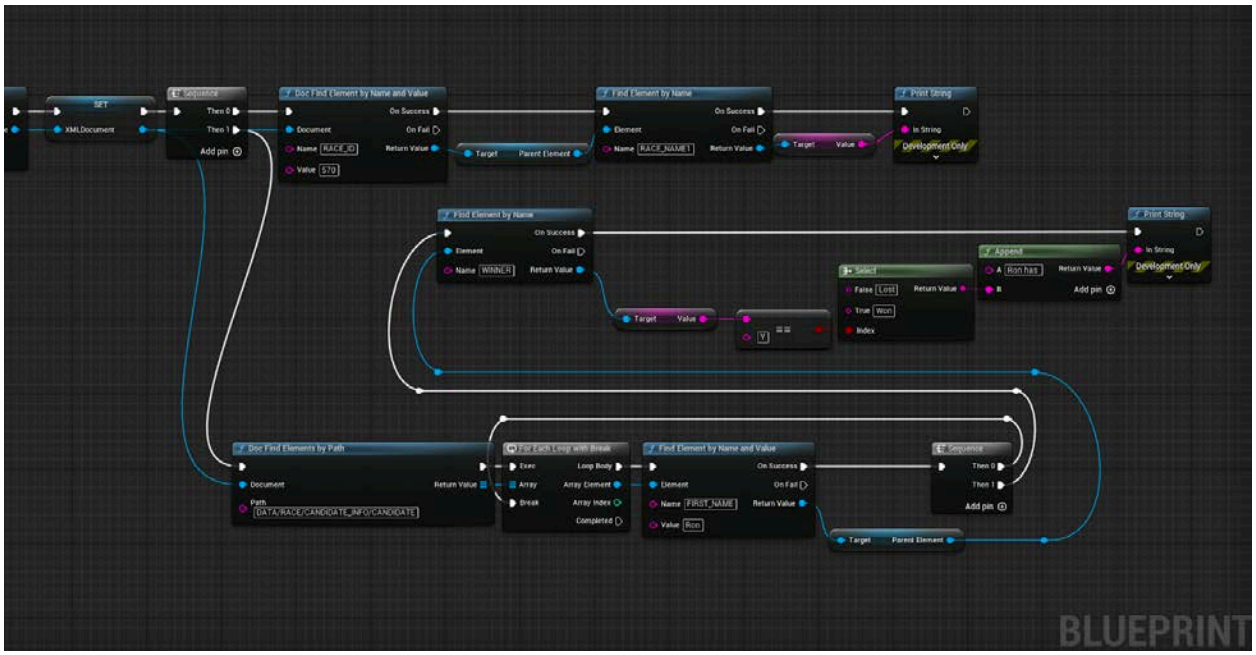
For larger files it’s recommended to use “Load XML From File Async”

Path is the path to the XML file in standard Windows format

We can then read from the XML document using Find functions:



Example of a use case:



if we have a XML file example:

<note>

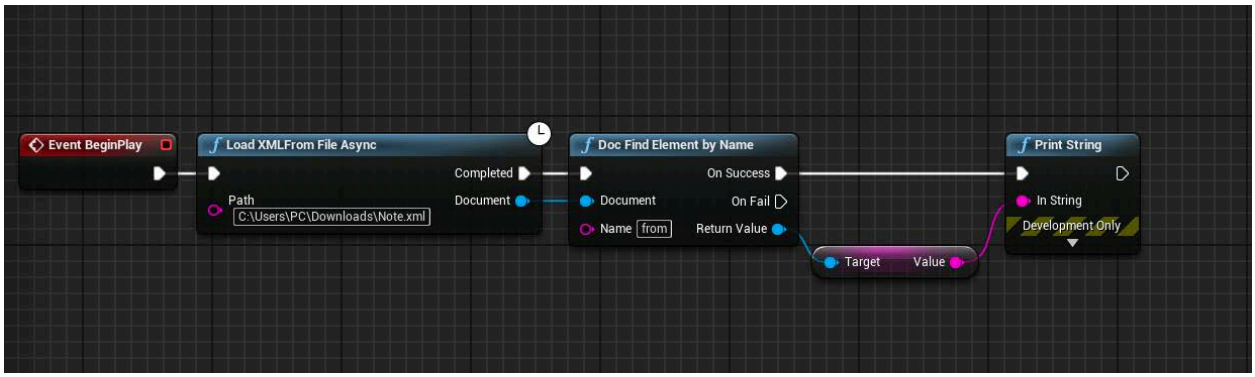
<to>Tove</to>

<from>Jani</from>

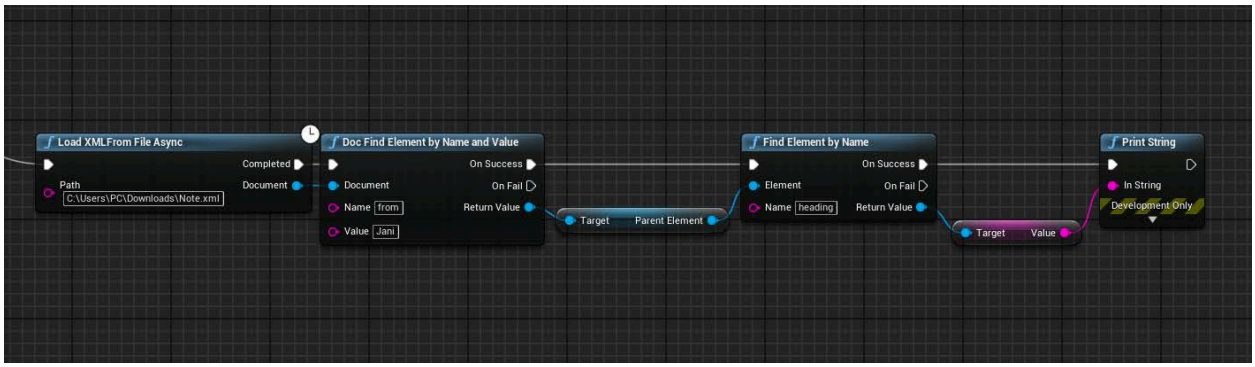
<heading>Reminder</heading>

</note>

we might want to find who we have the note from



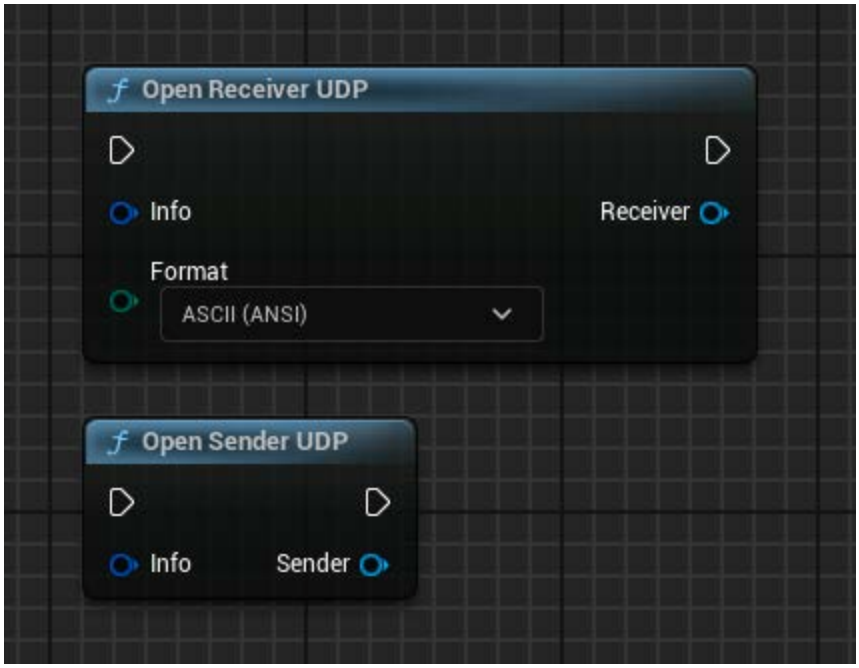
we might also like to find any note from “Jani” and get it’s heading



notice that we are using the Parent element to get to the <note> element, this can be utilized to go up in the XML hierarchy.

UDP

Allows the connection to the UDP Sockets using blueprints.

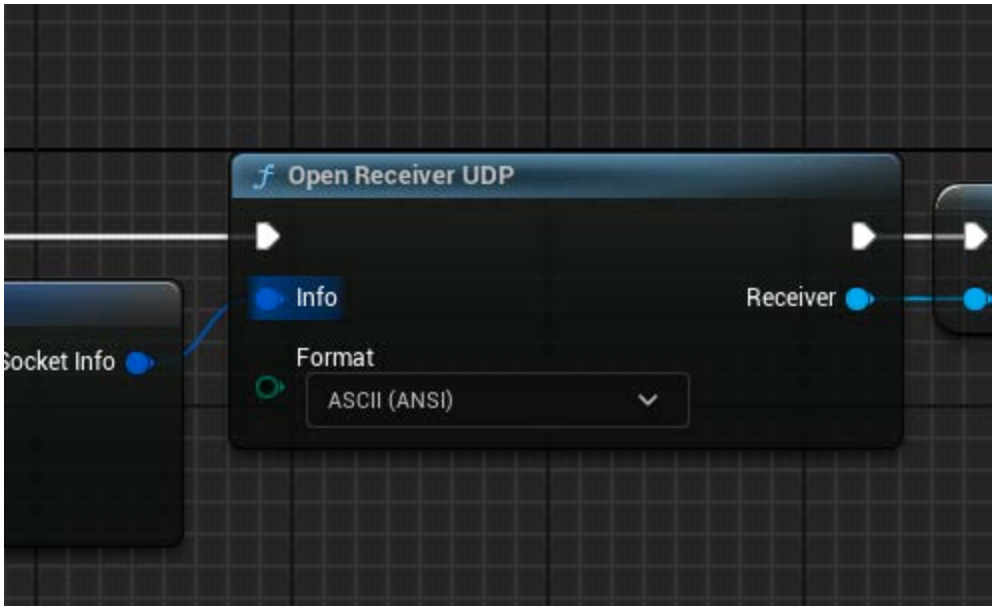


There are two main functions: “Open Receiver UDP” to receive UDP messages and “Open Sender UDP” to send UDP messages.

Both accept info

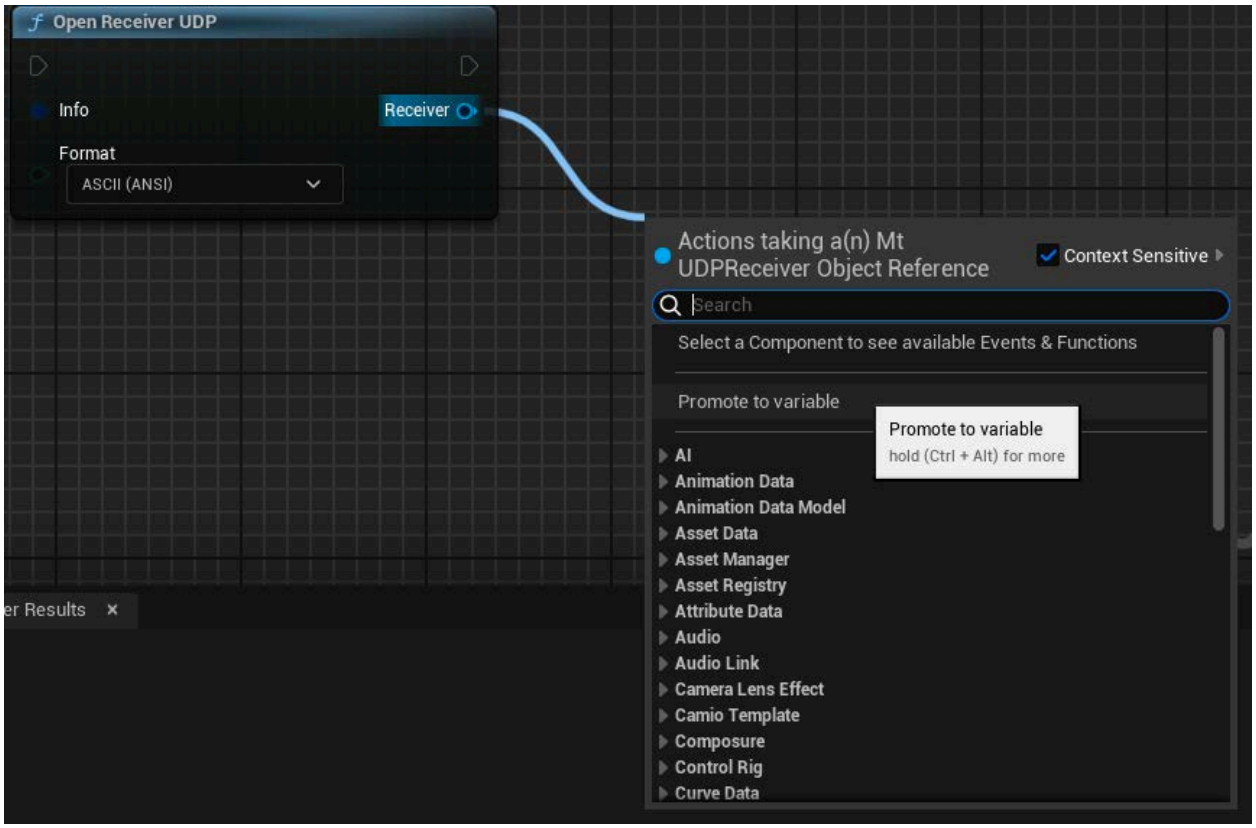
where you can specify IP, Prot for the socket connection and Socket Name

Receiver UDP

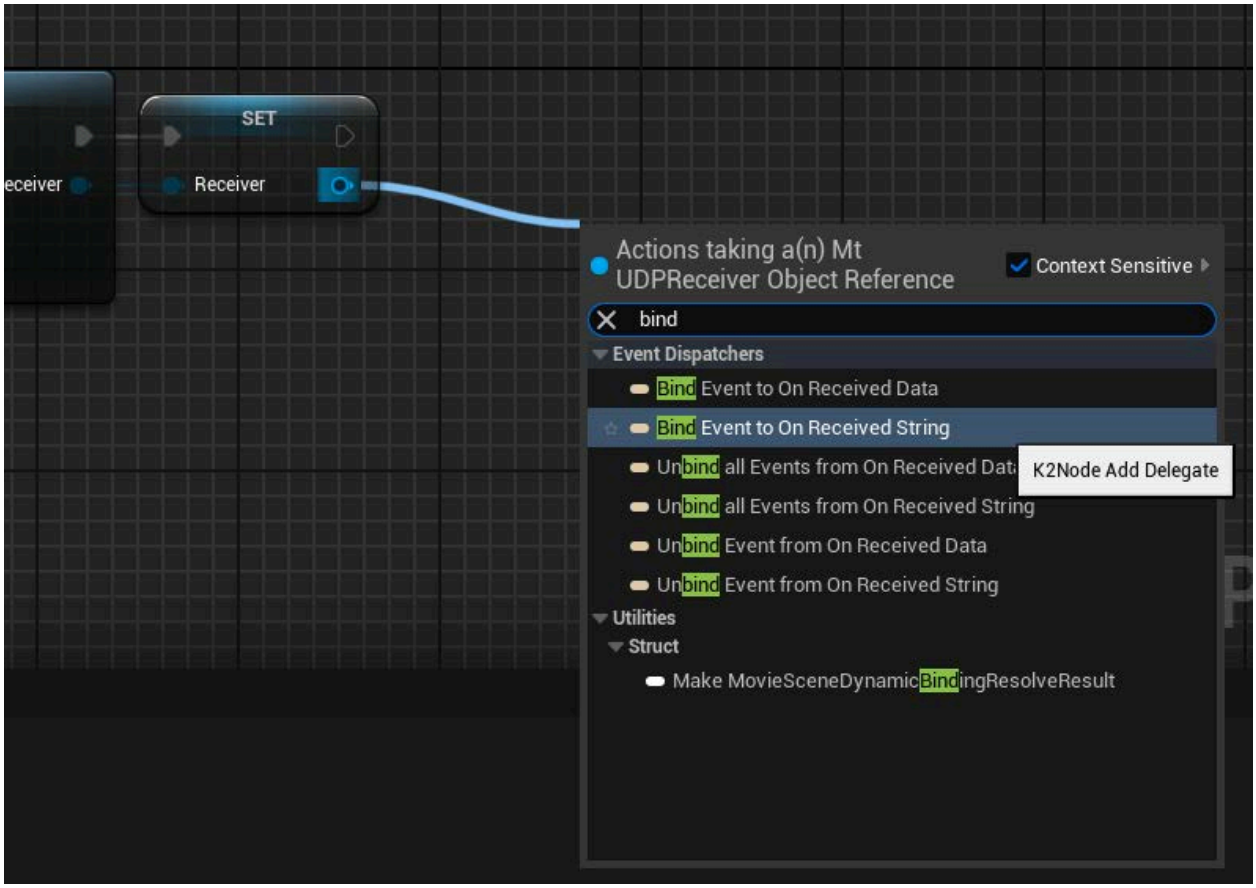


Receiver has a Format dropdown that is used when processing data to string and is used for On Received String

We suggest promoting the receiver to a variable using the “Promote to variable” this will prevent the destruction of the receiver and it will allow later use.



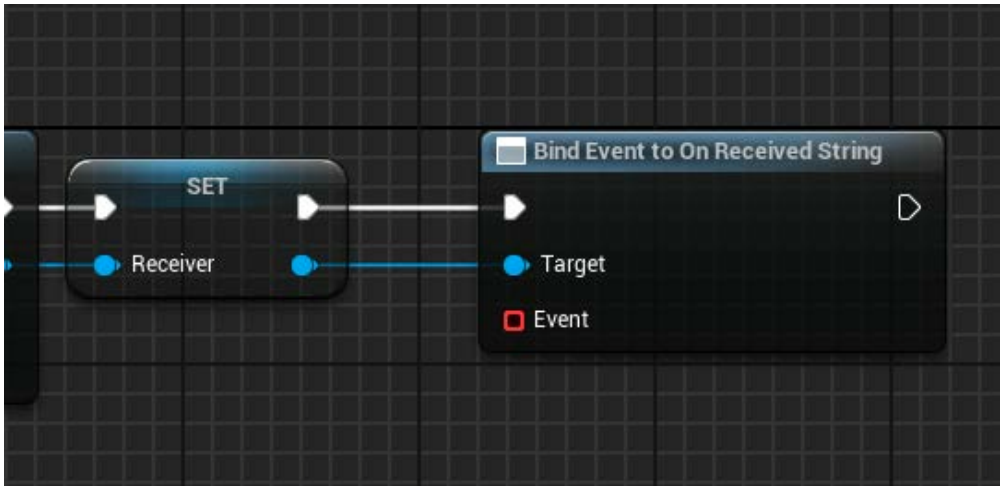
Then we can bind event to the receiver



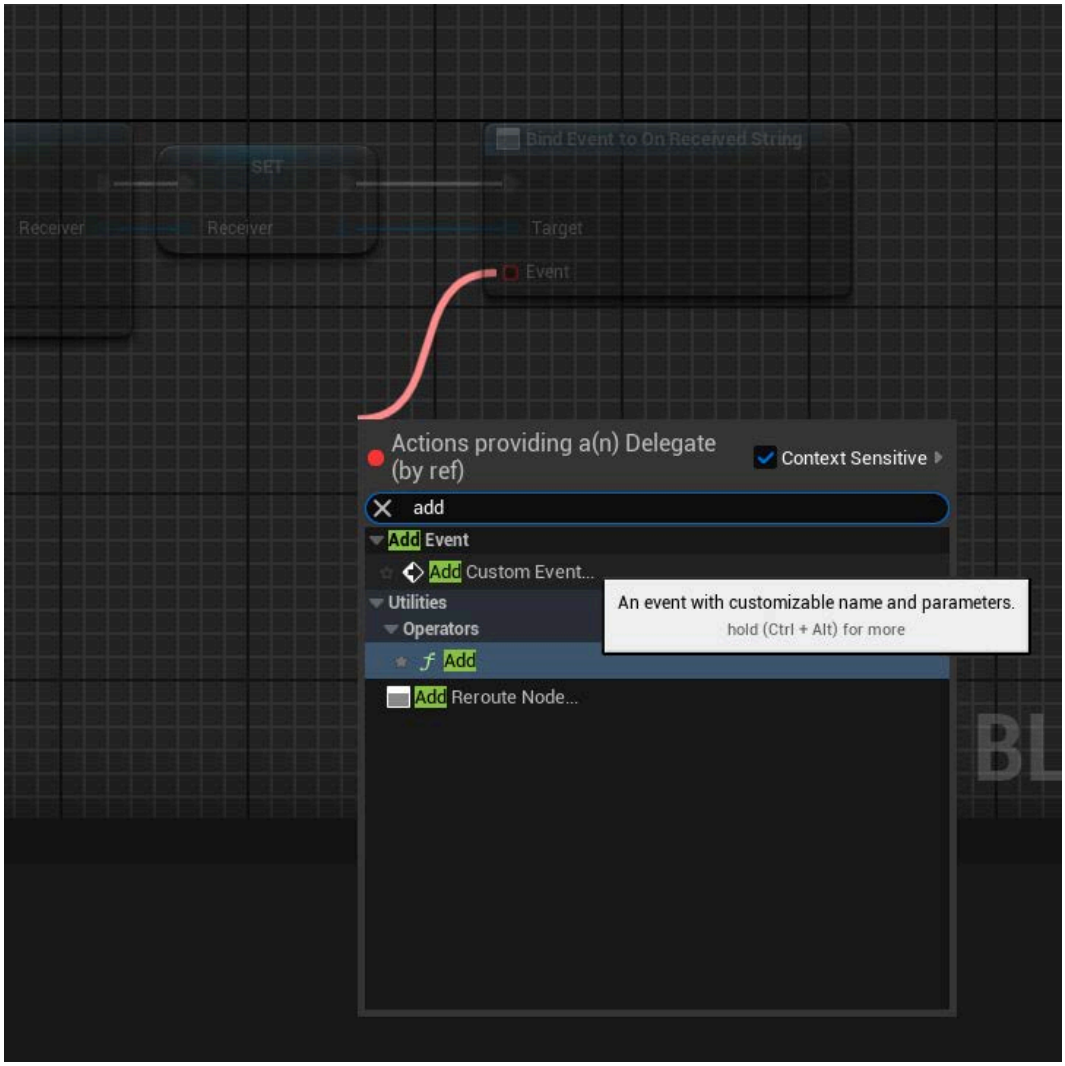
On Received String interprets the data as a string using the encoding while calling the “Open Receiver UDP”

On Received Data returns raw data without any interpretation and it is for the user to interpret them in some way.

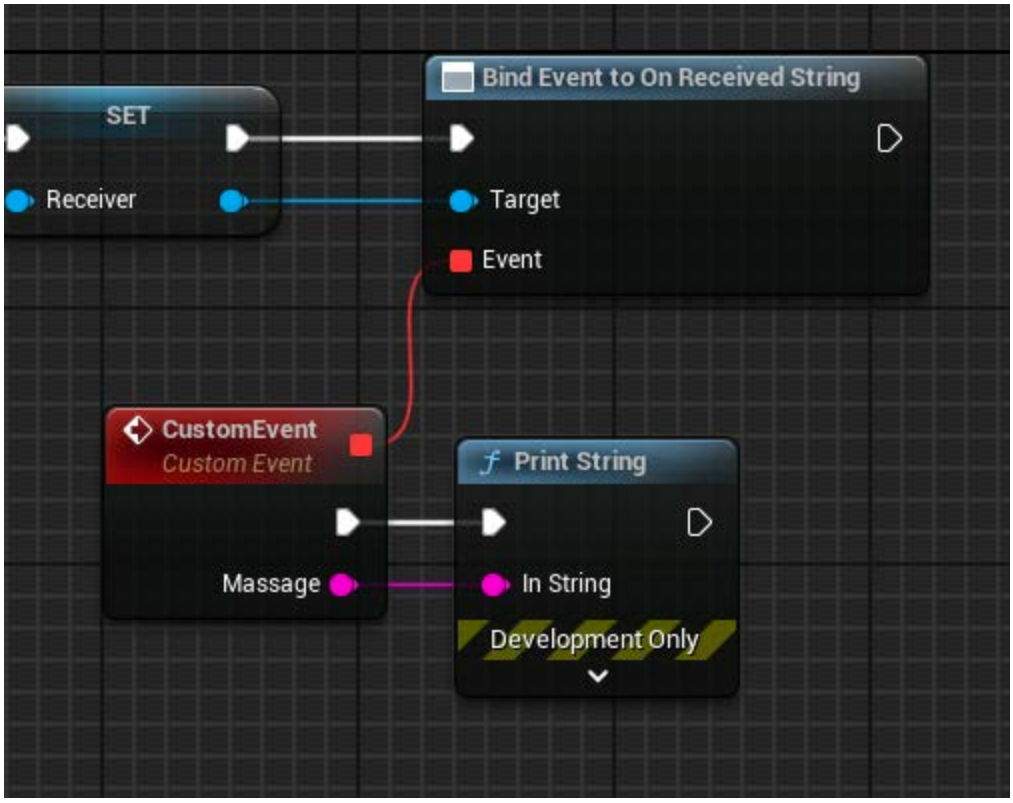
we will use On Received String in this example



we need to bind our event function, we can Add Custom Event



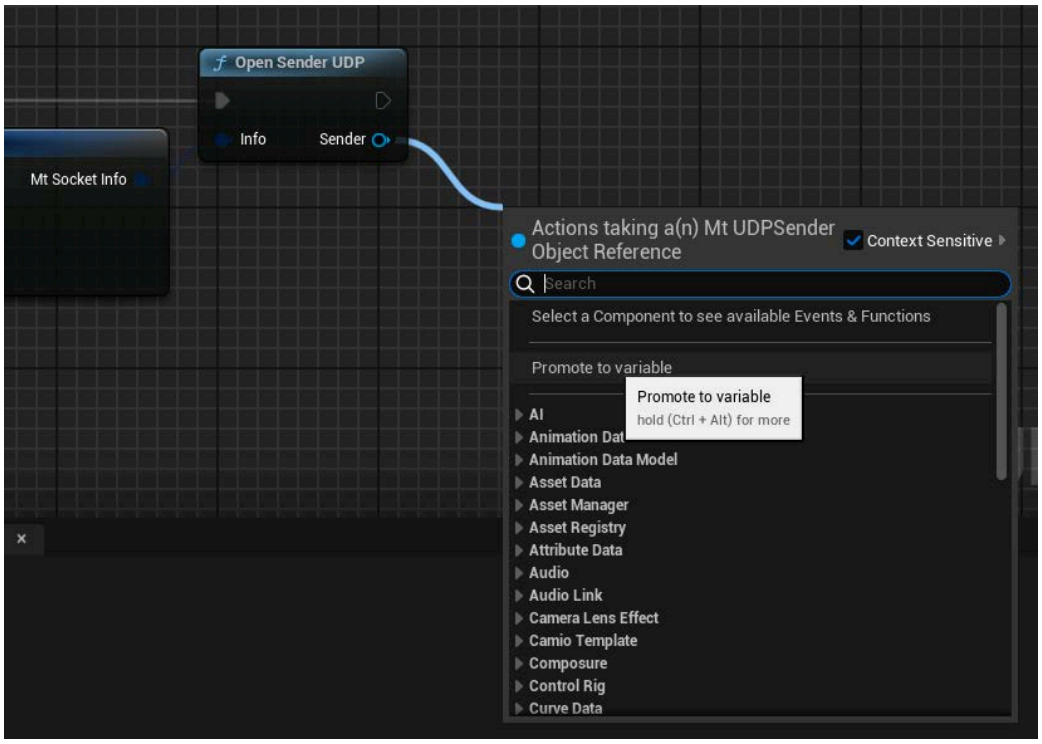
Then we can use the function to interpret the string and do something depending on the string.



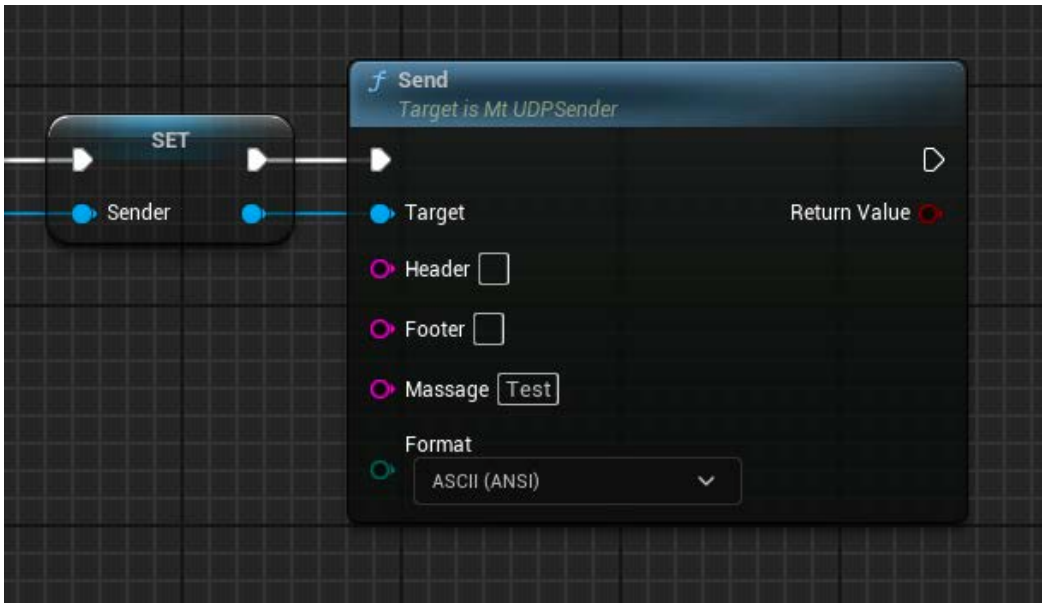
In this example we just use Print String to print the message that we have received

Sender UDP

We suggest promoting the sender to a variable using the “Promote to variable” this will prevent the destruction of the sender and it will allow later use.



We can send Messages using the Send function



Header

it is text that gets appended before the Massage

Footer

it is text that gets appended after the Massage

Massage

it is the text that we want to send

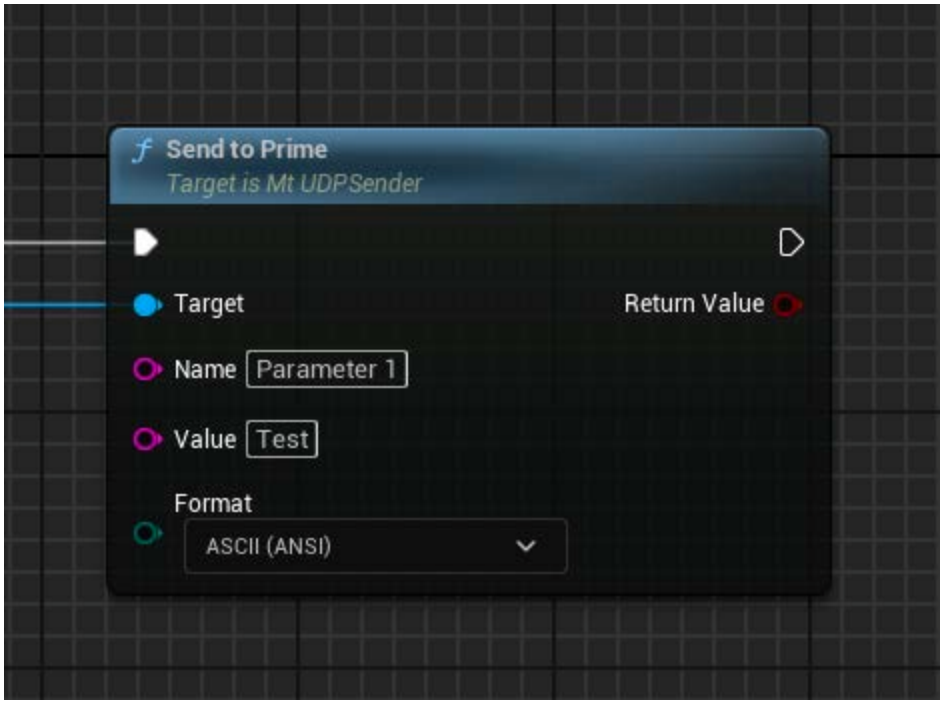
Format

it determines the formatting that Massage gets converted to before the send

Return Value

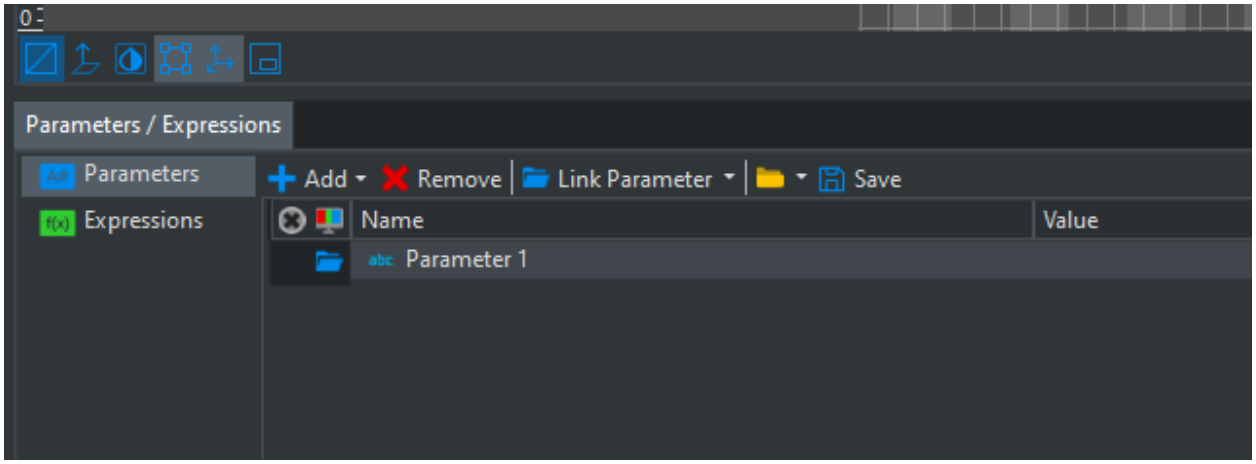
send returns true if the send has been successful.

Alternatively we can use Send To Prime which is similar to the Send except tries to set Parameter in PRIME



Name

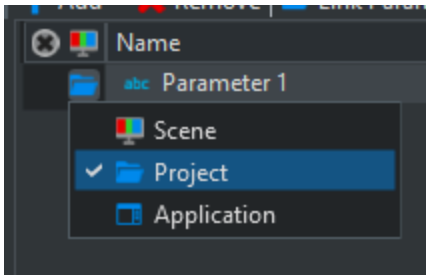
refers to the name of the Parameter in PRIME



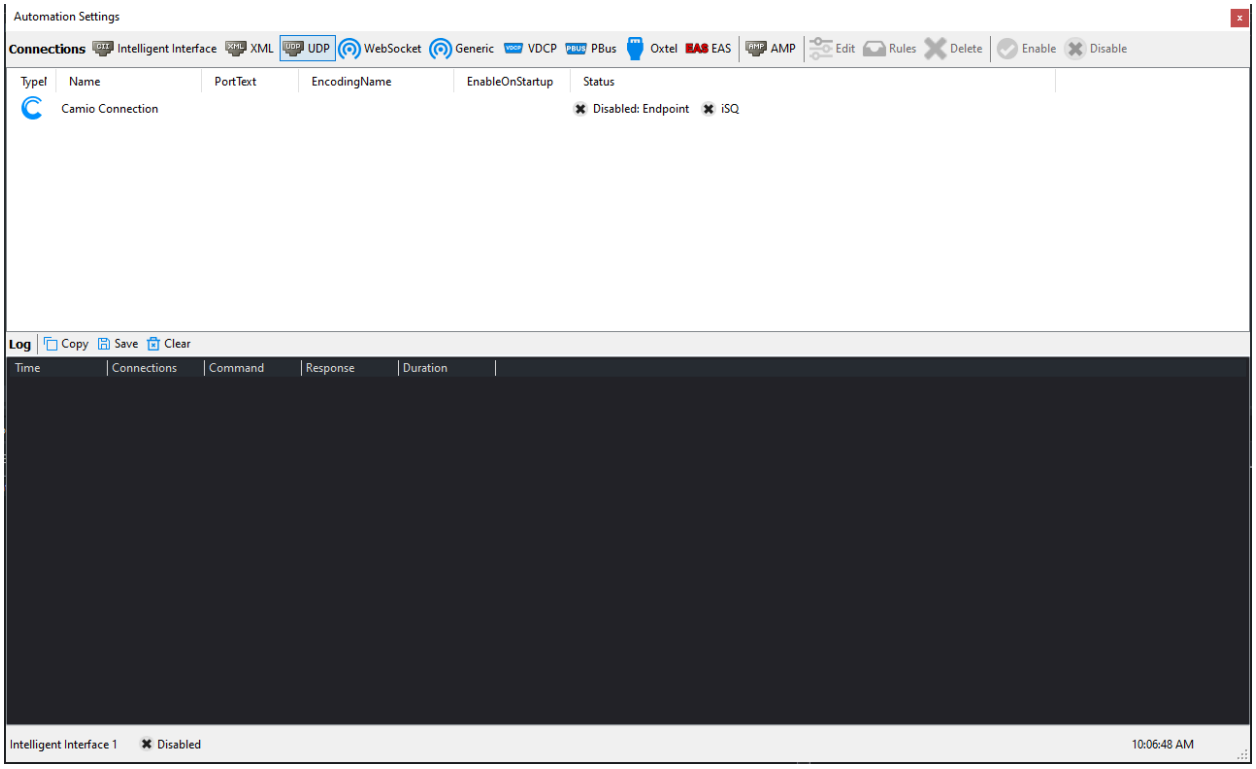
Value

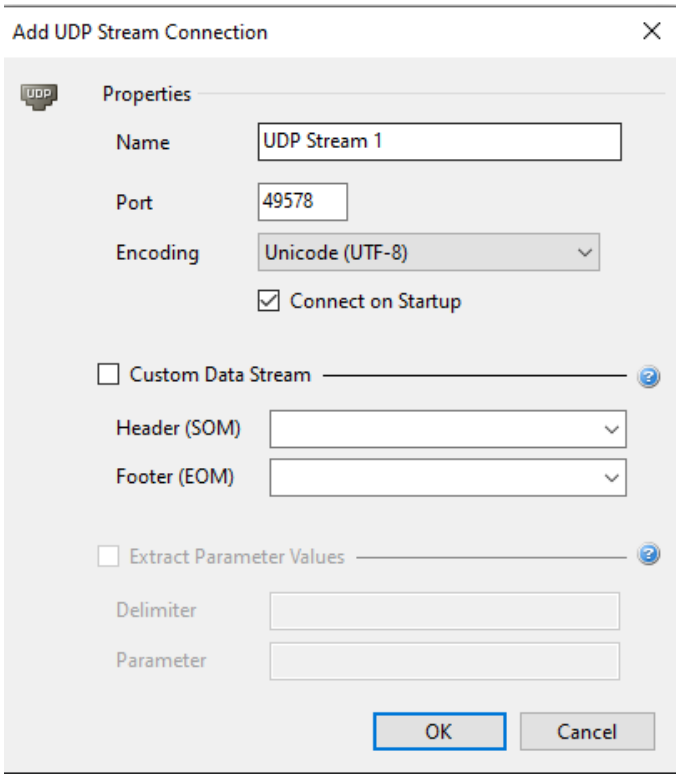
value that will be set

Note: it is required to have the Parameter in PRIME scoped to the Project or Application.



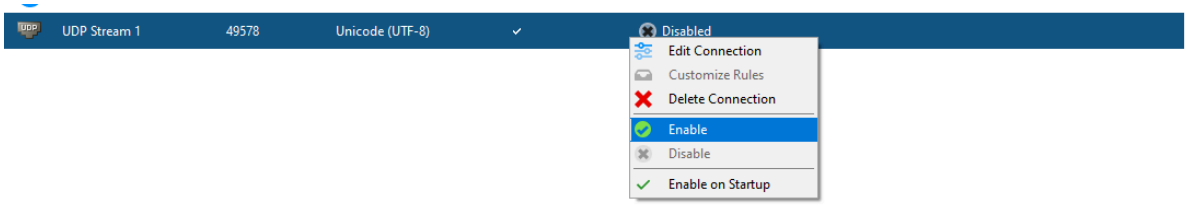
In PRIME we can add the listener using the Automation Settings


Automation Settings
Connections
Intelligent Interface
XML
UDP
WebSocket
Generic
VDCP
PBus
Oxtel
EAS
AMP
Edit
Rules
Delete
Enable
Disable
TypeI
Name
PortText
EncodingName
EnableOnStartup
Status
Camio Connection
Disabled: Endpoint
iSQ
Log
Copy
Save
Clear
Time
Connections
Command
Response
Duration
Intelligent Interface 1
Disabled
10:06:48 AM

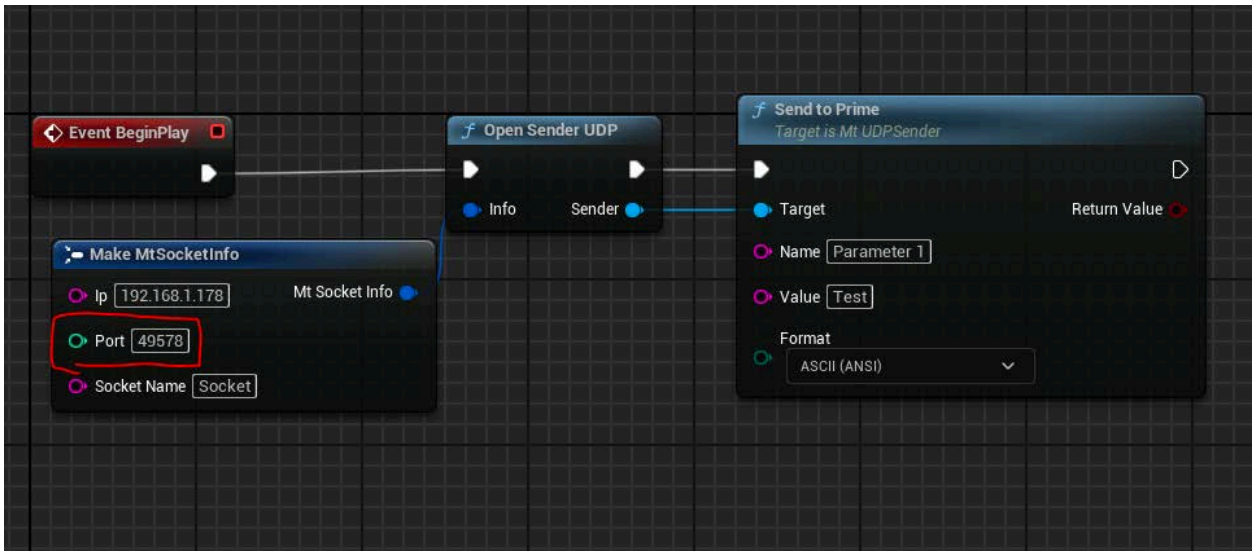

Add UDP Stream Connection
Properties
Name
UDP Stream 1
Port
49578
Encoding
Unicode (UTF-8)
Connect on Startup
Custom Data Stream
Header (SOM)
Footer (EOM)
Extract Parameter Values
Delimiter
Parameter
OK
Cancel

we use the default settings

make sure to enable the listener



in VSAR we specify the same port



TCP

TCP Listener

This class implements a frame for TCP server. It listens on a specified port, accepts incoming connections and initializes TCP Connection class for each of them. After default Timeout on connection (one minute), connection is automatically closed and resources freed. If a user wants this behaviour to change, writing custom logic or changing connection attributes is possible by implementing OnTcpConnectionAccepted delegate.

Attributes:

- Info - Socket info, contains socket name, ip address and port number. After changing, calling the Open method is needed.
- ReceiveFormat - This setting will affect encoding of all currently opened connections and all future connections. Supports ANSI, UTF8, UTF16, TCHAR.

Delegates:

- OnTcpReceivedString - invoked on all accepted connections when received message, decodes incoming data to string with specified encoding from ReceiveFormat
- OnTcpReceivedData - invoked on all accepted connections when received message
- OnTcpReceivedStringWithConnection - invoked on all accepted connections when received message, decodes incoming data to string with specified encoding from ReceiveFormat, has second parameter of type TCP Connection, to allow user specifying custom two way communication
- OnTcpReceivedDataWithConnection - invoked on all accepted connections when received message, has second parameter of type TCP Connection, to allow user specifying custom two way communication
- OnTcpConnectionAccepted - invoked immediately after accepting connection, has parameter of type TCP Connection, to allow communication initialization on server side. This should be invoked before first packet processing by above mentioned delegates.

Methods:

- Open - Begins listening on socket specified by Info
- Close - Closes socket
- IsSocketValid - Returns bool, if socket is valid and listening.

TCP Connection

This class implements a frame for TCP client. It is able to receive or send data (or strings, with specified encoding). This class also implements an auto reconnect mechanism, which can be enabled by the user.

Attributes:

- Info - Socket info, contains socket name, ip address and port number. After changing, calling the Open method is needed.
- ReceiveFormat - This setting will affect encoding of all received data. Supports ANSI, UTF8, UTF16, TCHAR.
- Timeout - Time from last socket activity to invoking OnTcpConnectionTimeout delegate. This is supposed to handle closing the socket, if no incoming or outgoing communication goes through. This setting can be changed, if the user wants to have a custom implementation.
- ReconnectTimeout - This attribute can specify a period for the reconnect timer in seconds. The default value is 5 and it is not recommended to set it too low (minimum 2.0). If a reconnect attempt is invoked, with the last one not finished yet, it is skipped.
- AutoReconnect - bool value, enabling auto reconnect mechanism. This should not be set true, on connection created by the listener (but is not checked - responsibility of user) because connections created by listeners are just "mirrors" - hence, they should not reconnect, but wait for the reconnection attempt from the other side.

Delegates:

- OnTcpReceivedString - invoked on all accepted connections when received message, decodes incoming data to string with specified encoding from ReceiveFormat
- OnTcpReceivedData - invoked on all accepted connections when received message
- OnTcpReceivedStringWithConnection - invoked on all accepted connections when received message, decodes incoming data to string with specified encoding from ReceiveFormat, has second parameter of type TCP Connection, to allow user specifying custom two way communication
- OnTcpReceivedDataWithConnection - invoked on all accepted connections when received message, has second parameter of type TCP Connection, to allow user specifying custom two way communication
- OnTcpConnectionTimeout - This delegate is invoked after Timeout amount of time of inactivity - no data sent or received. User is supposed to free resources, close socket and stop using it - since it was not used, there is not a big probability that it will be used in future. Setting the Timeout attribute can change the delay of invoking this delegate. Also, not

implementing it is not a mistake and can be done - but, there will be a socket and some additional resources allocated and reserved.

- OnTcpConnectionFailed - This delegate is invoked, after an Open method was called, but connection was not successful. If AutoReconnect is set, reconnection will be attempted, if not, the user can implement its own logic, using this delegate.
- OnTcpConnected - This delegate is invoked, after a successful connection attempt. If a user wants to send any data through connection, implementation should be done after this event. (because before this, socket is not valid and all sending attempts will result in fail)

Methods:

- Open - Attempts to connect to socket, specified in Info.
- Close - Closes socket and frees any resources.
- IsSocketValid - Checks if socket is allocated. Does not care about state of connection.
- IsConnected - Checks if socket is allocated, connected and if connection is not timeouted.
- SendString - Sends string with specified encoding (set as parameter, defaults to UTF8)
- SendBytes - Sends bytes
- GetSocketInfo - Returns Info

Opening functions:

- OpenListenerTCP

Open TCP listener. Listener can be used immediately after this function is executed. Recommended to check if socket is valid after initialization. Afterwards users can bind custom events to delegates.

- OpenConnectionTCP -

ABOUT US

Chyron is ushering in the next generation of storytelling in the digital age. Founded in 1966, the company pioneered broadcast titling and graphics systems. With a strong foundation built on over 50 years of innovation and efficiency, the name Chyron is synonymous with broadcast graphics. Chyron continues that legacy as a global leader focused on customer-centric broadcast solutions. Today, the company offers production professionals the industry's most comprehensive software portfolio for designing, sharing, and playing live graphics to air with ease. Chyron products are increasingly deployed to empower OTA & OTT workflows and deliver richer, more immersive experiences for audiences and sports fans in the arena, at home, or on the go.

CONTACT SALES

EMEA • North America • Latin America • Asia/Pacific
+1.631.845.2000 • sales@chyron.com

Chyron