

DELTACAST MEDIA PLUGIN DOCUMENTATION

VERSION 1.0
UNREAL ENGINE PLUGIN



TABLE OF CONTENTS

1. DELTACAST MEDIA	3
1.1 PREREQUISITES	3
1.1.1 VERSION AND COMPATIBILITY.....	3
1.1.2 DRIVER & SOFTWARE INSTALL	3
1.2 PLUGIN INSTALL.....	3
1.2.1 WINDOWS.....	3
1.2.2 LINUX.....	3
1.2.2.1 ADVANCED	4
1.3 PROJECT SETUP.....	4
1.4 RENDERING VIDEO INPUT	6
1.4.1 DELTACAST MEDIA SOURCE SETTINGS	8
1.5 OUTPUT CAPTURES	9
1.5.1 IN UNREAL EDITOR.....	9
1.5.2 AT RUNTIME	14
1.5.3 DELTACAST MEDIA OUTPUT SETTINGS.....	21
1.6 TIMECODE.....	22
1.6.1 VISUALIZING TIMECODE	22
1.6.2 OVERRIDING THE ENGINE TIMECODE	23
1.7 GENLOCK (CUSTOM TIME STEP)	25
1.7.1 VISUALIZING FRAME RATE.....	25
1.7.2 GENLOCKING THE ENGINE	26
1.8 PLUGIN SETTINGS	28

1. DELTACAST MEDIA

After reading this manual, you'll be able to:

- have video input from your DELTACAST board playing inside your Unreal Engine project
- have video output from your Unreal Engine project through your DELTACAST board
- override Unreal Engine's timecode with your timecode source through your DELTACAST board
- genlock Unreal Engine's frame rate through your DELTACAST board

1.1 PREREQUISITES

1.1.1 VERSION AND COMPATIBILITY

The DELTACAST Media plugin is compatible with **Unreal Engine 5 and later** and with VideoMaster SDK **6.16 and later**.

Supported cards are:

- DELTA-12G: 1c, 2c, 20, 40, 04, 4c, 2i1c, 2c- asi8c
- DELTA-12G1c-hmi10-elp
- DELTA-12G-elp-key 11
- DELTA-h4k2-elp 20
- DELTA-3G-elp-d: 4c, 8c, 8b, 84, 80
- DELTA-3G-elp-key 11
- FLEX-12g 10, 01
- FLEX-HMI 10, 01
- FLEX-DP 10, 01

1.1.2 DRIVER & SOFTWARE INSTALL

See our different board models available on www.deltacast.tv, and download the VideoMaster software package from our download center. Linux software resources requires credentials which you can request by completing our contact form or sending an email to contact@deltacast.tv.

Once the VideoMaster software package is downloaded, follow the Install Guide for both board drivers and software libraries.

1.2 PLUGIN INSTALL

1.2.1 WINDOWS

The very first step to use the DELTACAST Media plugin is to install it from the marketplace following the next steps:

1. Go to the plugin's marketplace page
2. Add it to cart and check out
3. Go to your Library
4. Click on "Install to engine" and select the engine version

1.2.2 LINUX

To install a plugin under Linux is a bit more manual than for Windows, following these next steps:

1. Download the plugin from DELTACAST download center
2. Open the directory containing your Unreal Engine project
3. Create a "Plugins" directory next to the .uproject file
4. Copy the downloaded plugin in the new directory
5. Open your project in the **Unreal Editor**
6. Accept the prompt asking to build the modules in the plugin

1.2.2.1 ADVANCED

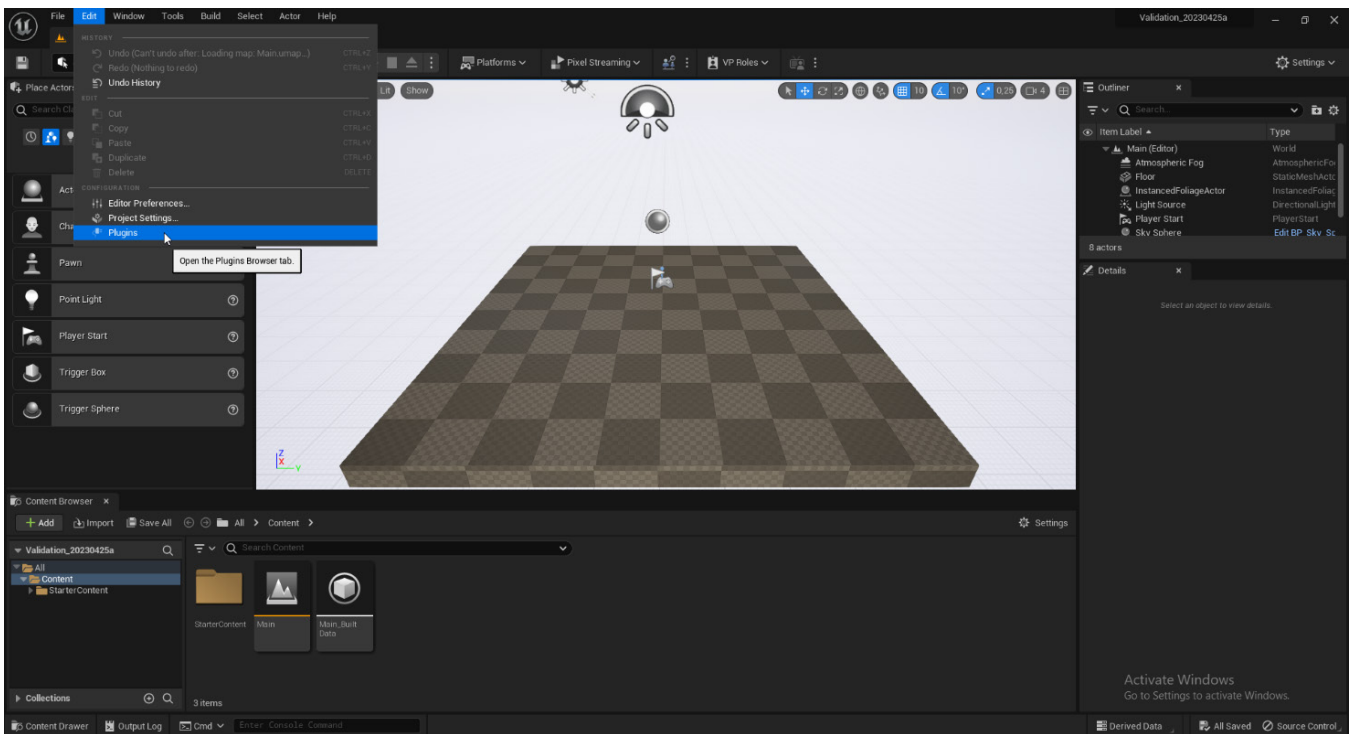
To allow quick and easy include of the plugin for all future projects without having to rebuild it manually, follow the next steps:

1. Follow the section above to build the plugin inside a project
2. Copy the pre- build plugin in your Unreal Engine install under: {install location}/Engine/Plugins/Media
3. The plugin can now be enabled/disabled from the “Plugins” panel on a per- project basis without further manual actions

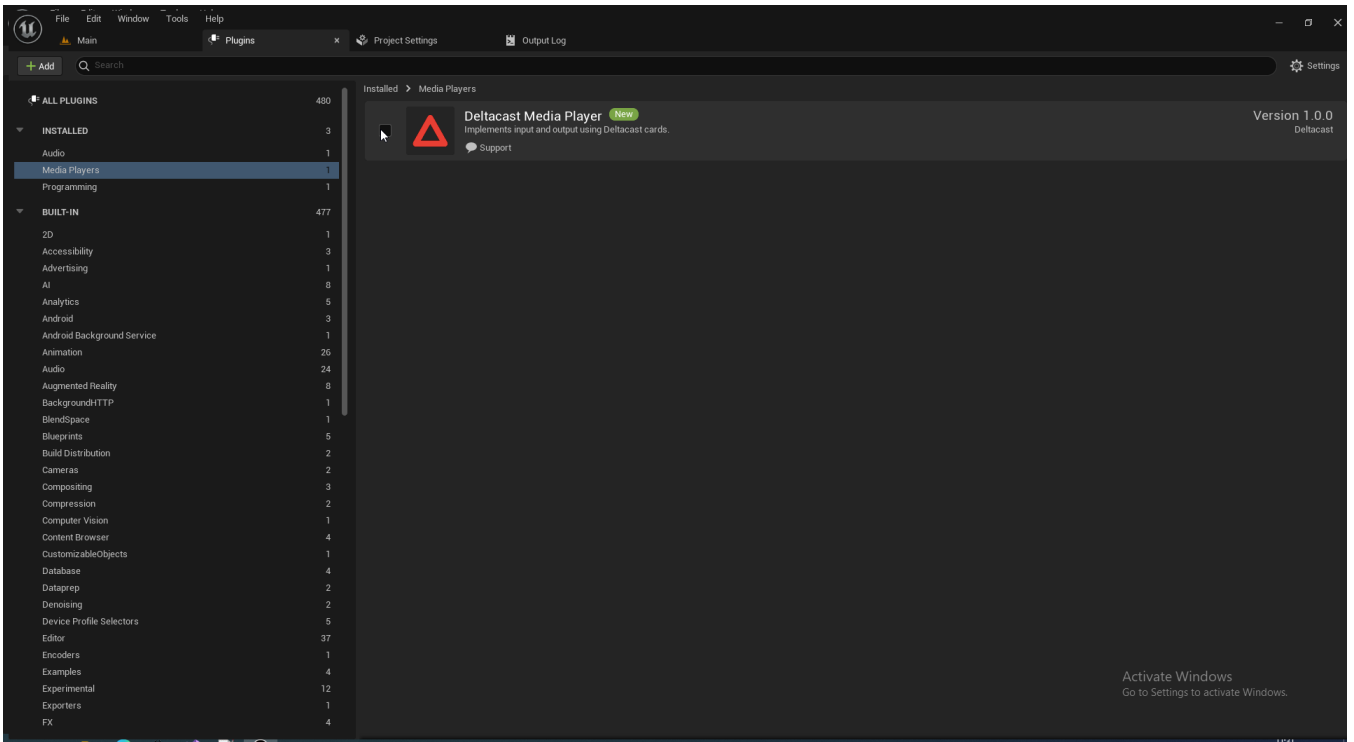
1.3 PROJECT SETUP

Once the plugin is available to your selected engine version, it can be enabled on a per project basis following the next steps:

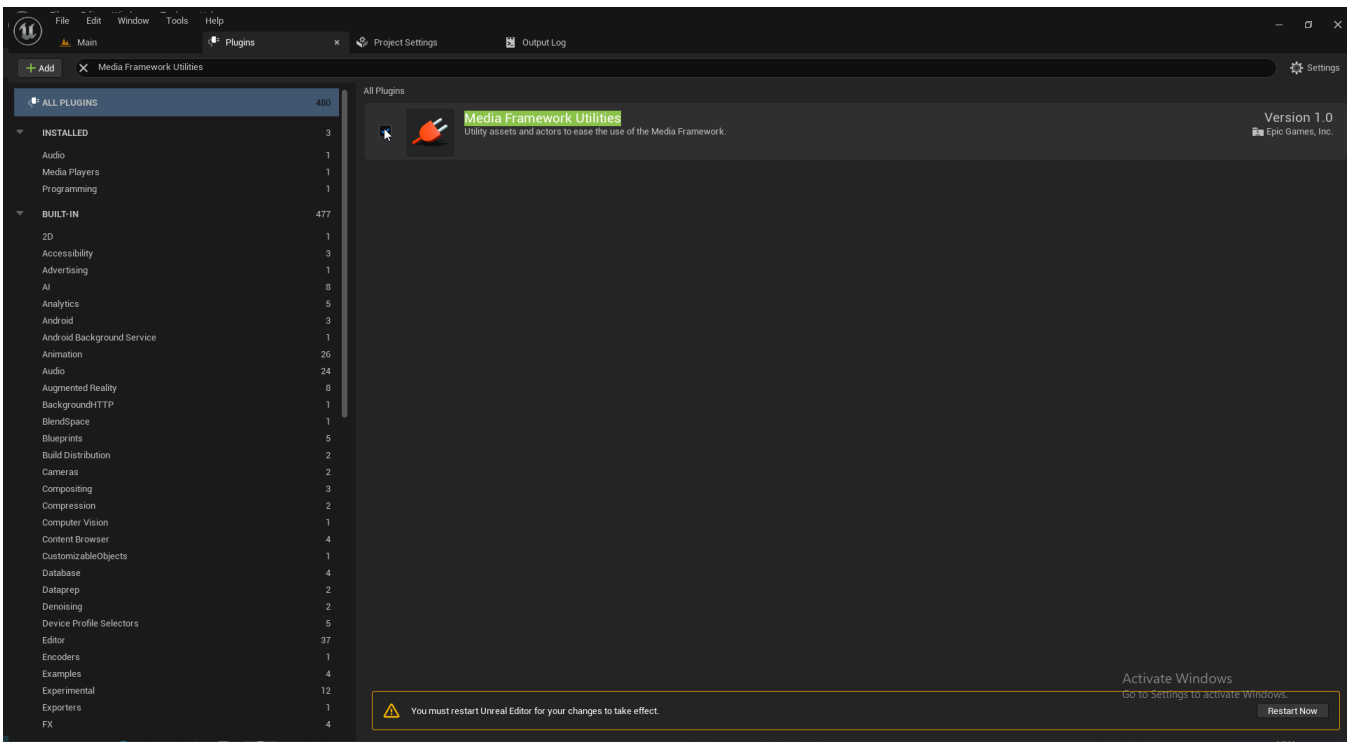
1. Open your project
2. From the main menu, choose **Edit > Plugins**



3. Search for the **DELTA CAST Media plugin** (it is under the **Media Players** category) and check its **Enabled** checkbox (a restart of Unreal Engine will be required but don't restart yet)



4. Search for the Media Framework Utilities plugin and check its Enabled checkbox if it's not already checked

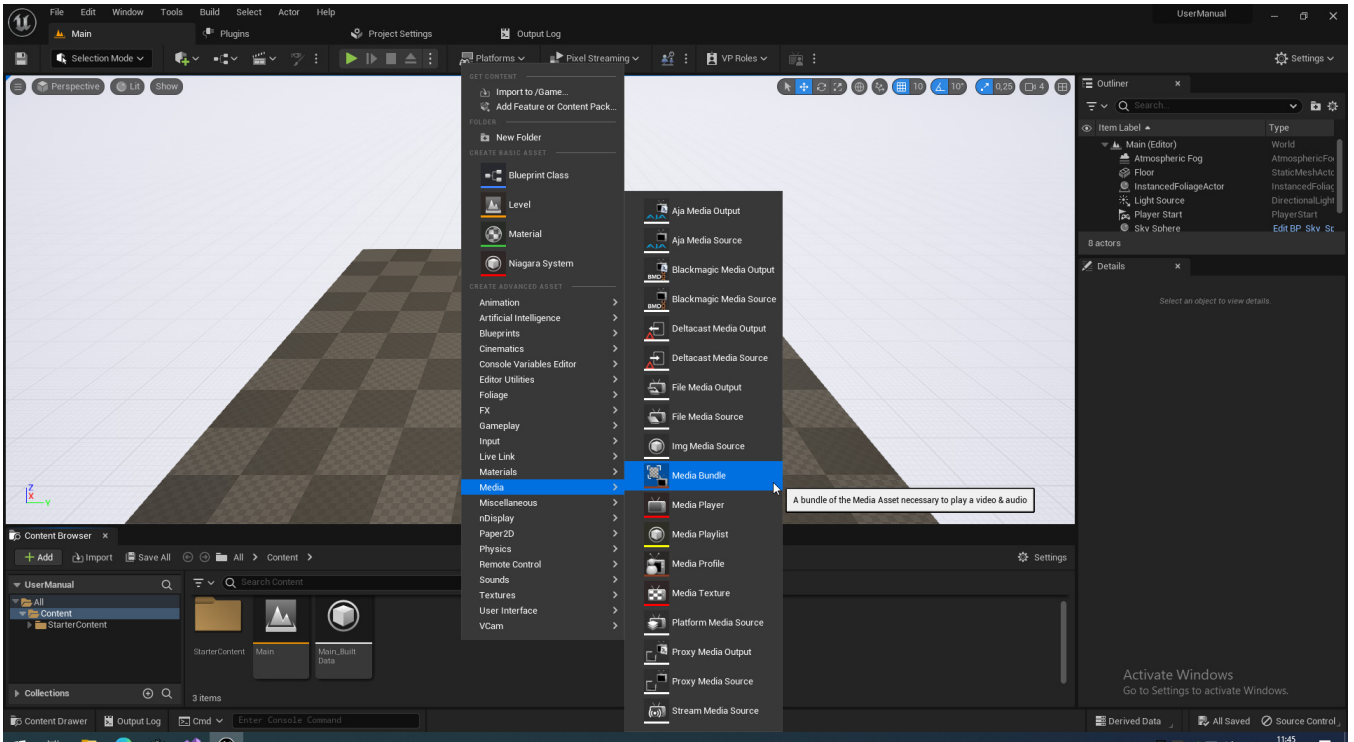


5. Reopen your project to apply the plugin changes

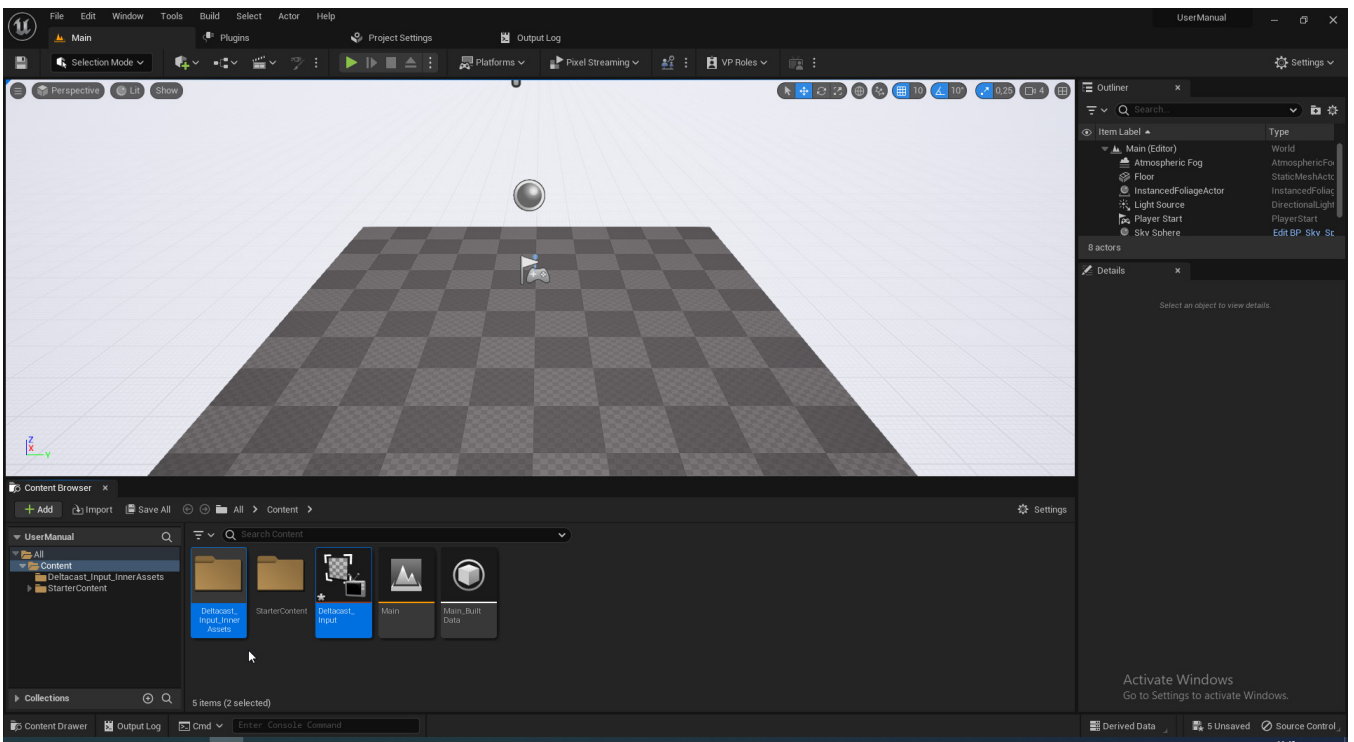
1.4 RENDERING VIDEO INPUT

In this section, you'll learn how to add a video input to your level in **Unreal Editor**. This process uses a Media Bundle: a kind of asset that packages together several different types of assets involved in the Media Framework, and that offers you control over some advanced features like lens deformation, chroma-keying, color correction, and more.

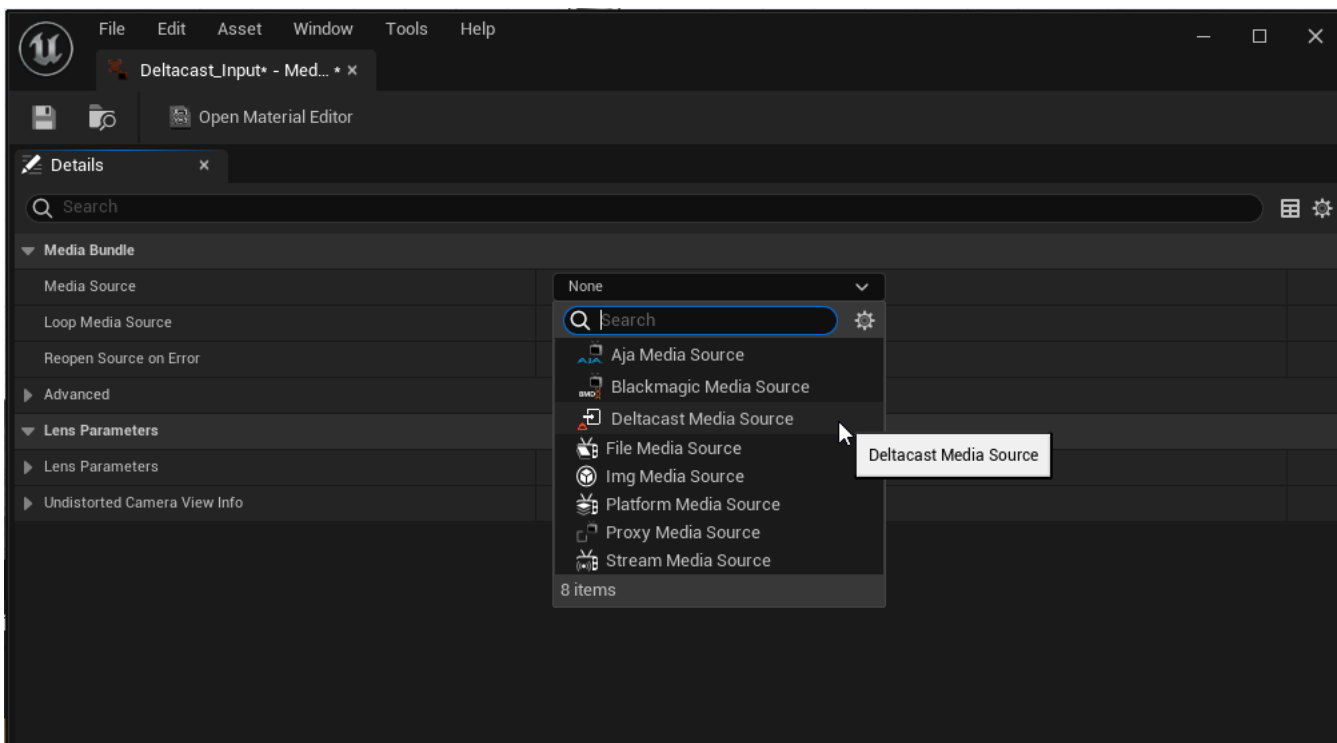
1. In the **Content Browser**, right-click and choose **Media > Media Bundle**



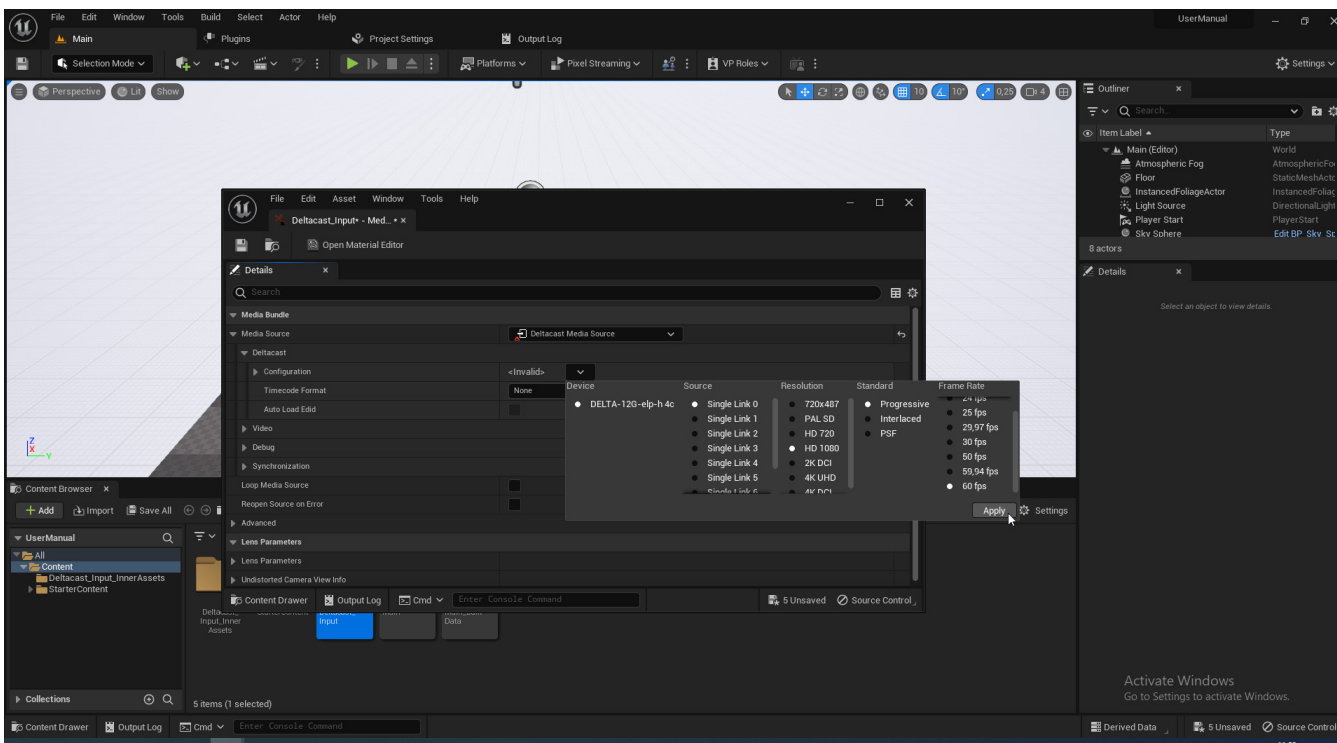
2. Give a meaningful name to the newly created asset, validate with **Enter**. A new folder is created next to the Media Bundle; it contains all the inner assets used by the Media Bundle



3. Open the new Media Bundle by double-clicking it
4. Edit the **Media Source** property to **Deltacast Media Source**

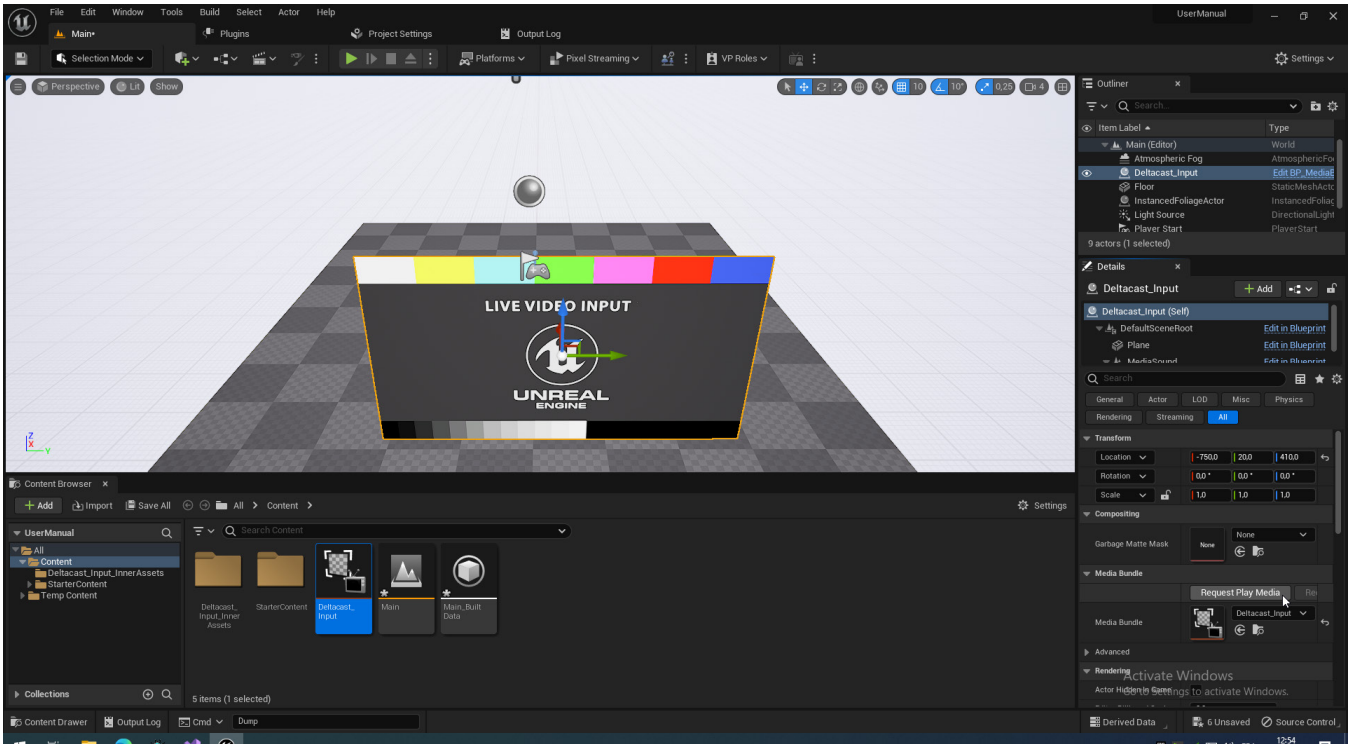


5. Configure the Media Source (see 1.4.1)



6. Drag your Media Bundle asset from the Content Browser into the level viewport

7. A new plane appeared showing the video input configured in your Media Bundle. If the Media Bundle doesn't start playing, select it then click **Media Bundle > Request Play Media**.



1.4.1 DELTACAST MEDIA SOURCE SETTINGS

Each DELTACAST Media Source exposes the following settings:

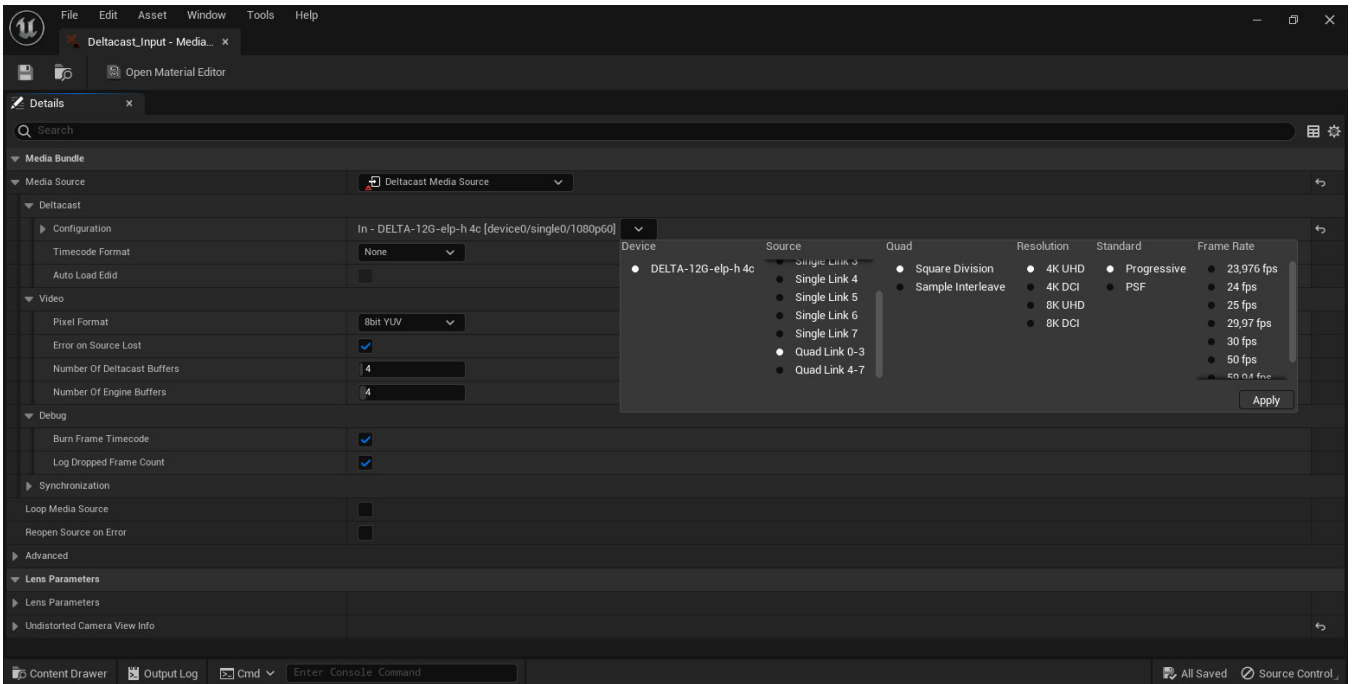


Figure 1: DELTACAST Media Source settings

Table 1: Board configurations

Property	Description
Device	List of all DELTACAST boards detected in the system.
Source	List of all available connections for the selected device. HDMI source is used for both HDMI and DisplayPort connections.
Quad	List of all available quad link configurations for the selected source. Only available when using a quad link source.
Resolution	List of all available resolution for the selected source.
Standard	List of all available standard for the selected resolution.
Frame rate	List of all available frame rate for the selected standard.

Table 2: Media Source configurations

Property	Description
Timecode format	Type of timecode to use from the input stream. VITC is not supported.
Auto load EDID	Whether to automatically load the EDID. Only available for HDMI and DisplayPort streams; ignored in other cases.
Pixel format	Format used after input signal conversion. 8Bits ARGB is not supported with SDI streams.
Error on source lost	Whether to stop or pause the input stream on error.
Number of Deltacast buffers	Number of frame used by the Deltacast SDK.
	A smaller number is most likely to cause missed frame.
	A bigger number is most likely to increase latency.
Number of Engine buffers	Number of frame used by Unreal Engine.
	A smaller number is most likely to cause missed frame.
	A bigger number is most likely to increase latency.
Burn frame timecode	Whether to incrust the value of the timecode in each frame
Log dropped frame count	Whether to log when the input stream drops frames

The options exposed depends on the type of board used.

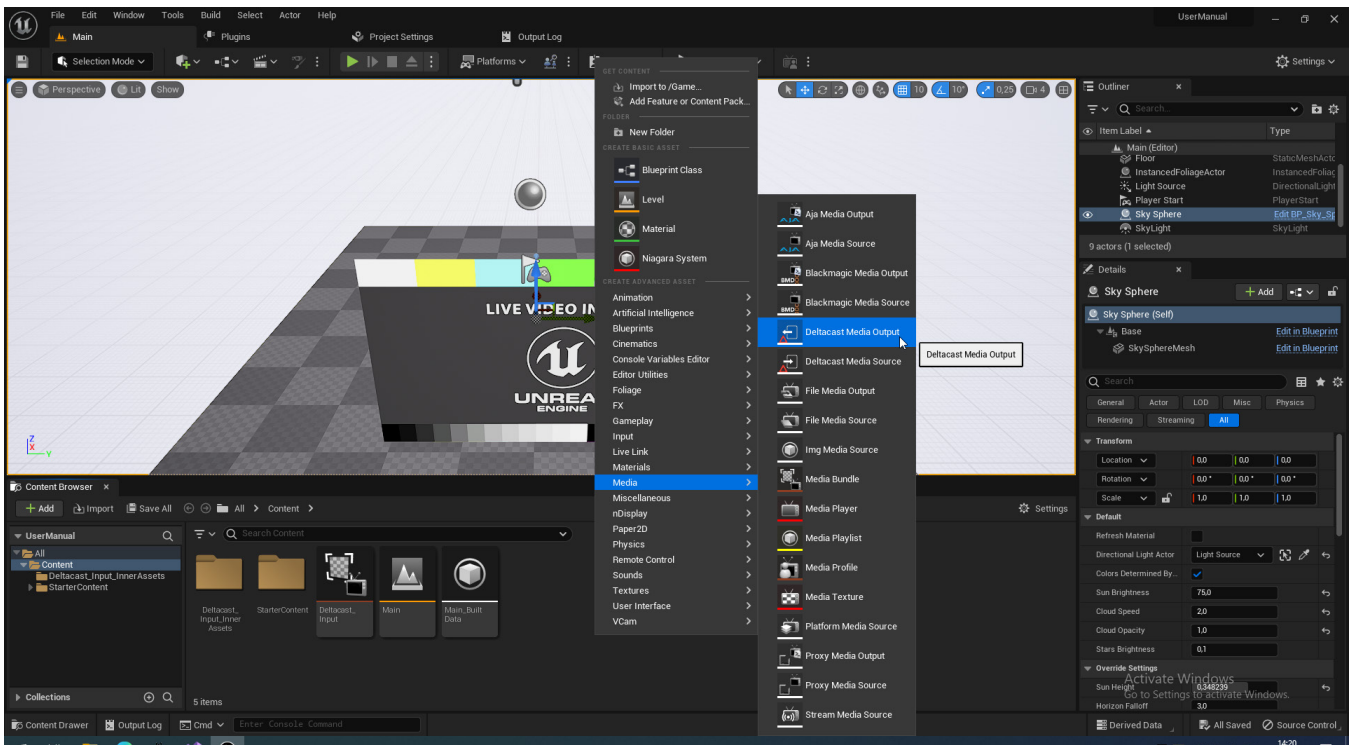
1.5 OUTPUT CAPTURES

In this section, you'll learn how to add a video output from both **Unreal Editor** and your **standalone application**.

1.5.1 IN UNREAL EDITOR

The following section will allow you to configure video outputs from **Unreal Editor** only. This won't work in a **standalone application**.

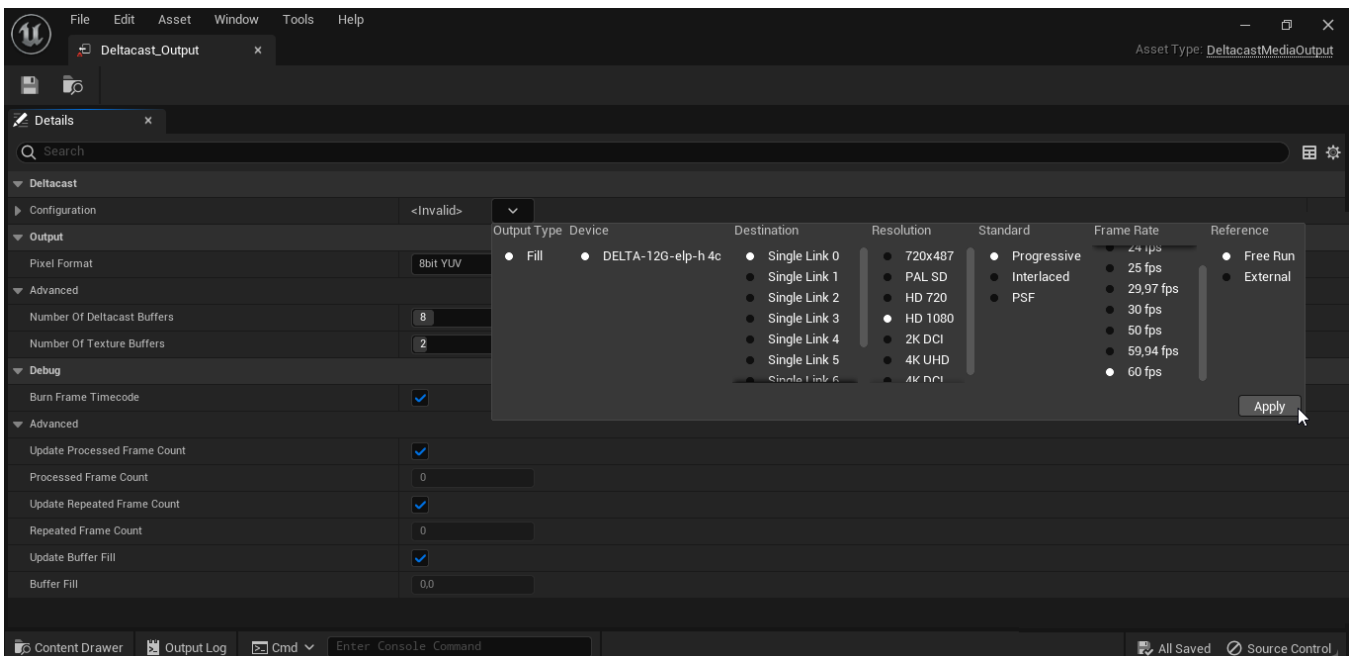
1. In the **Content Browser**, right-click and choose **Media > Deltacast Media Output**



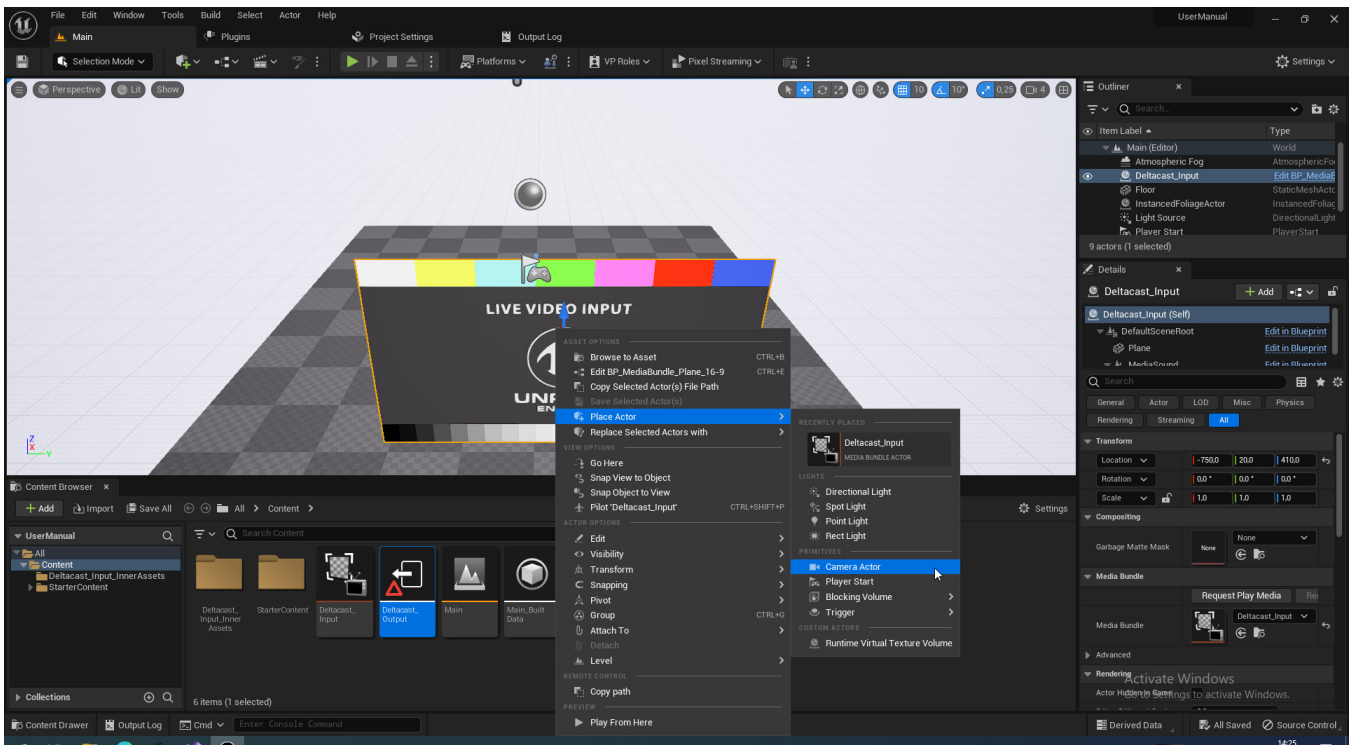
2. Give a meaningful name to the newly created asset, validate with **Enter**.

3. Open the new Media Bundle by double-clicking it

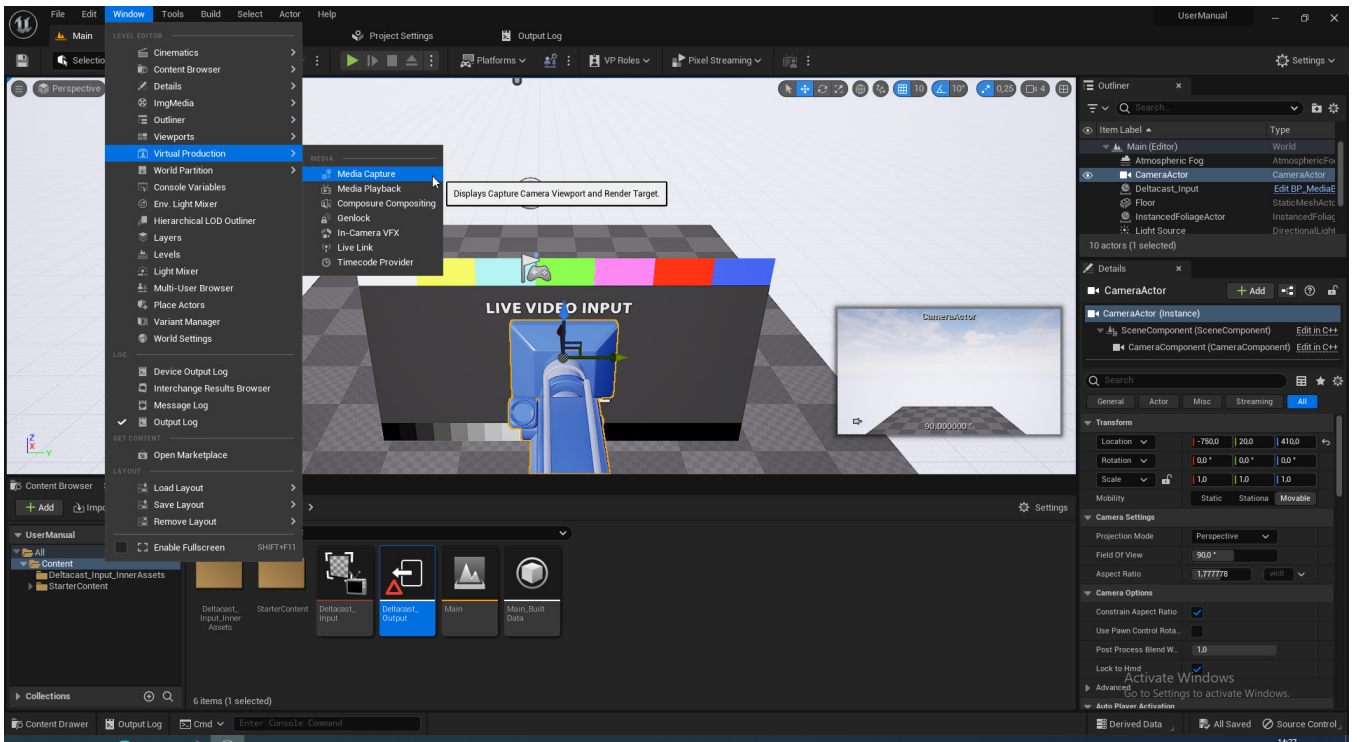
4. Configure the Media Output (see 1.5.3)



5. Right-click in the scene viewport to place a new camera actor



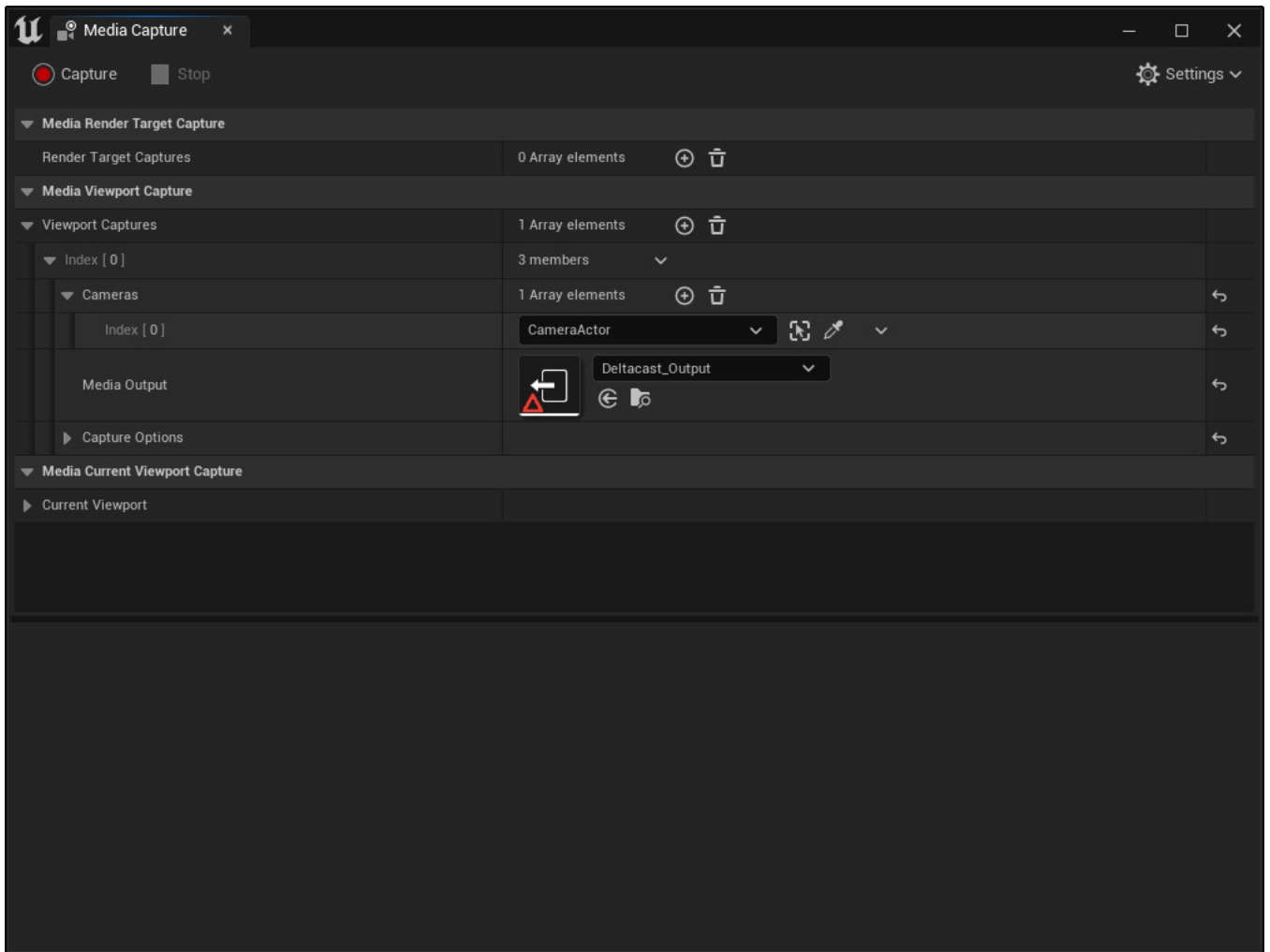
6. From the main menu, choose **Window > Virtual Production > Media Capture**



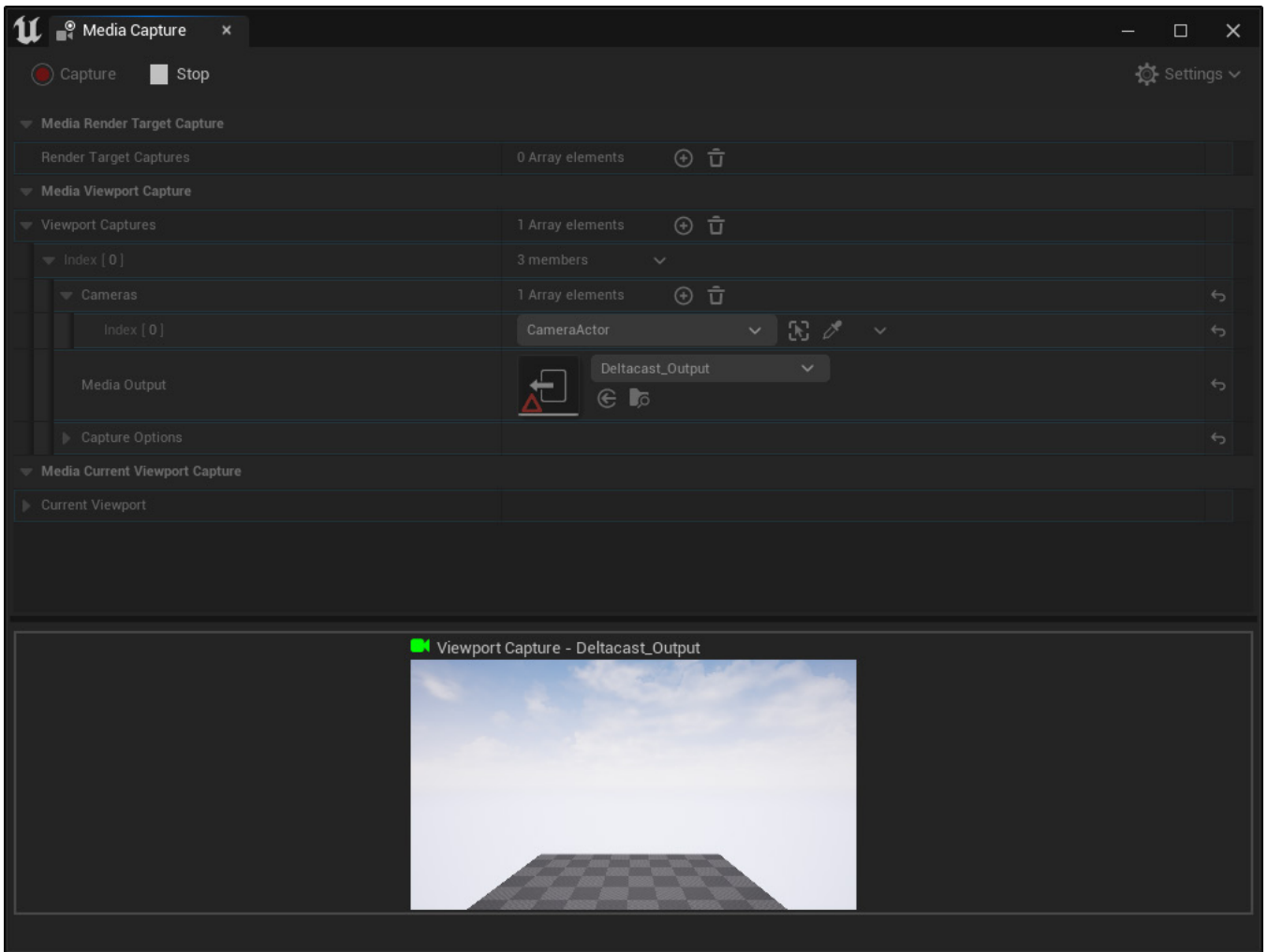
7. Under **Media Viewport Capture**, add a new element in **Viewport Captures**.

8. Inside the newly added Viewport Capture, add a new element in **Cameras**, then select the camera

9. Inside the newly added Viewport Capture, select the newly DELTACATS Media Output



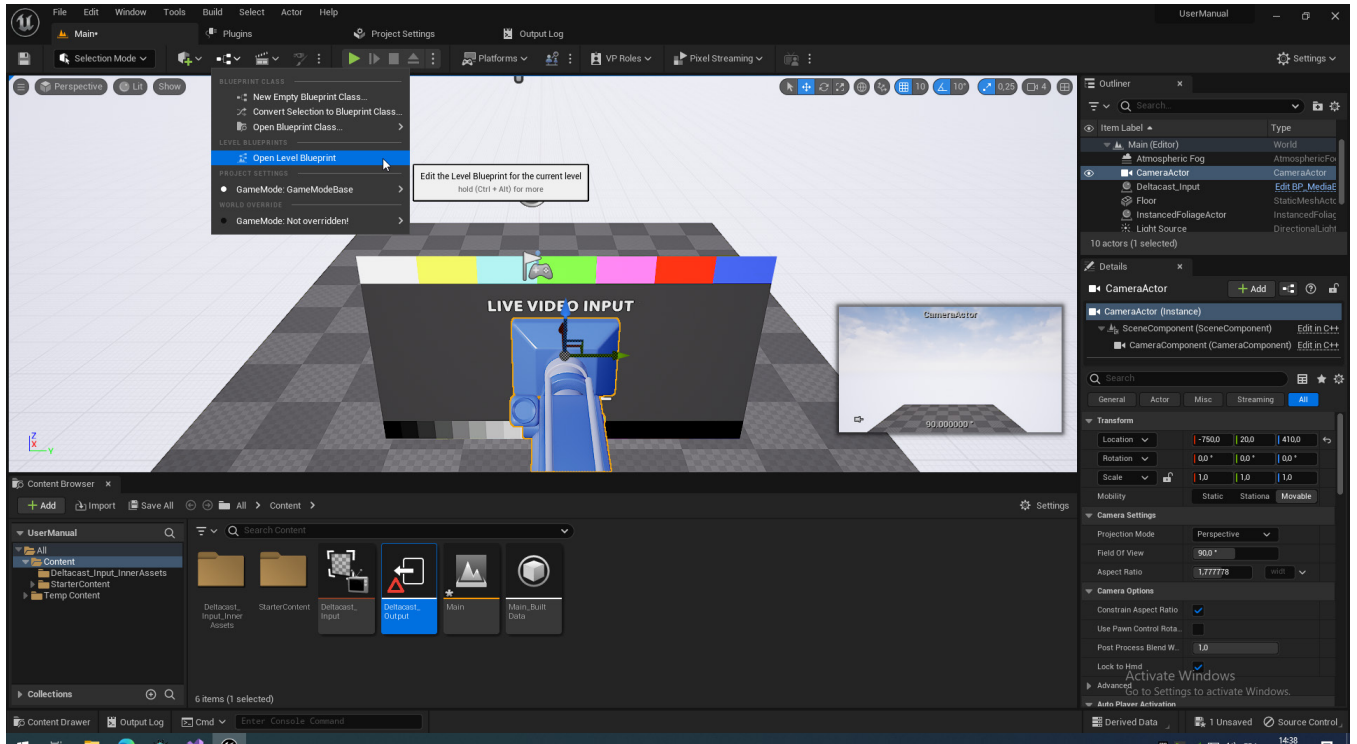
10. Click on **Capture**; the Deltacast Media Output is streaming the camera view and a preview is shown



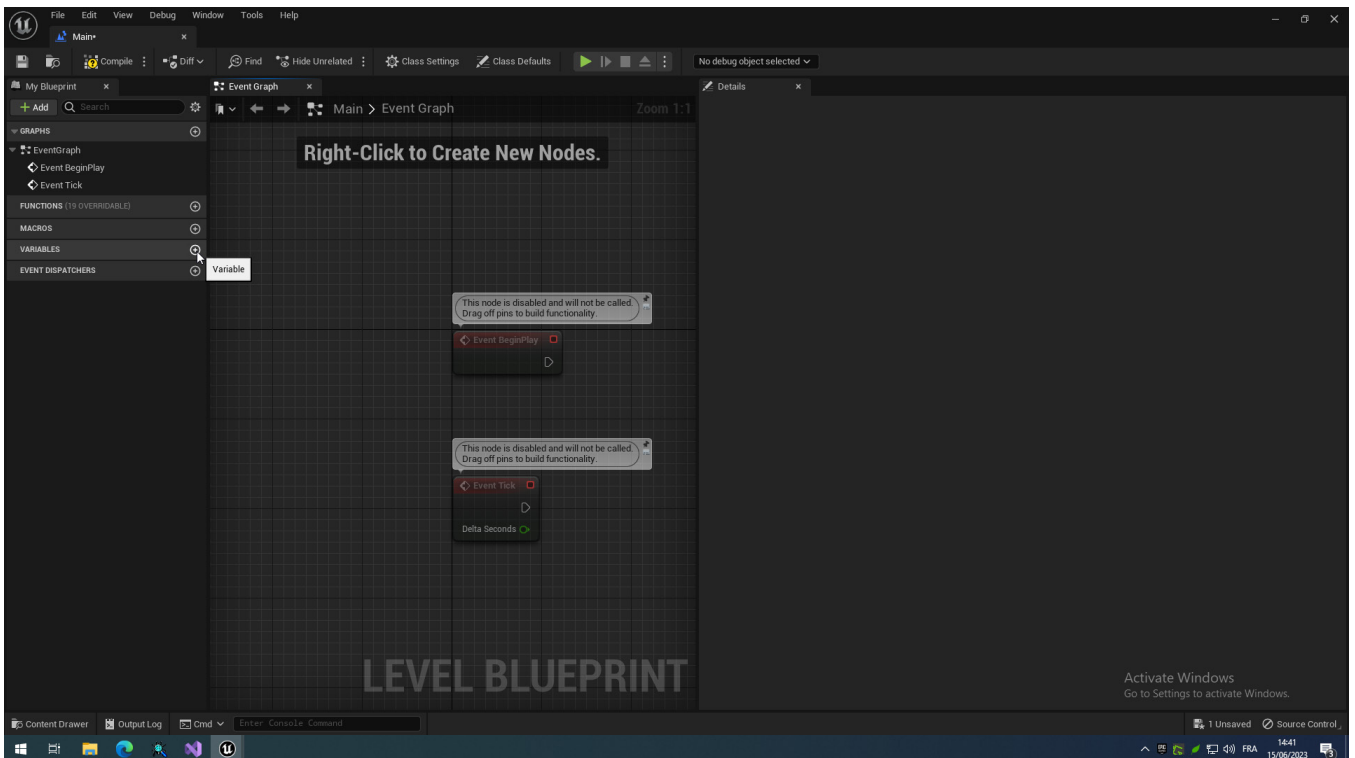
1.5.2 AT RUNTIME

The following section will show how to have a video output outside the **Unreal Editor**. The example shown will make use of the blueprint system to setup a simple toggle switch that starts/stops capturing when the user presses on a key.

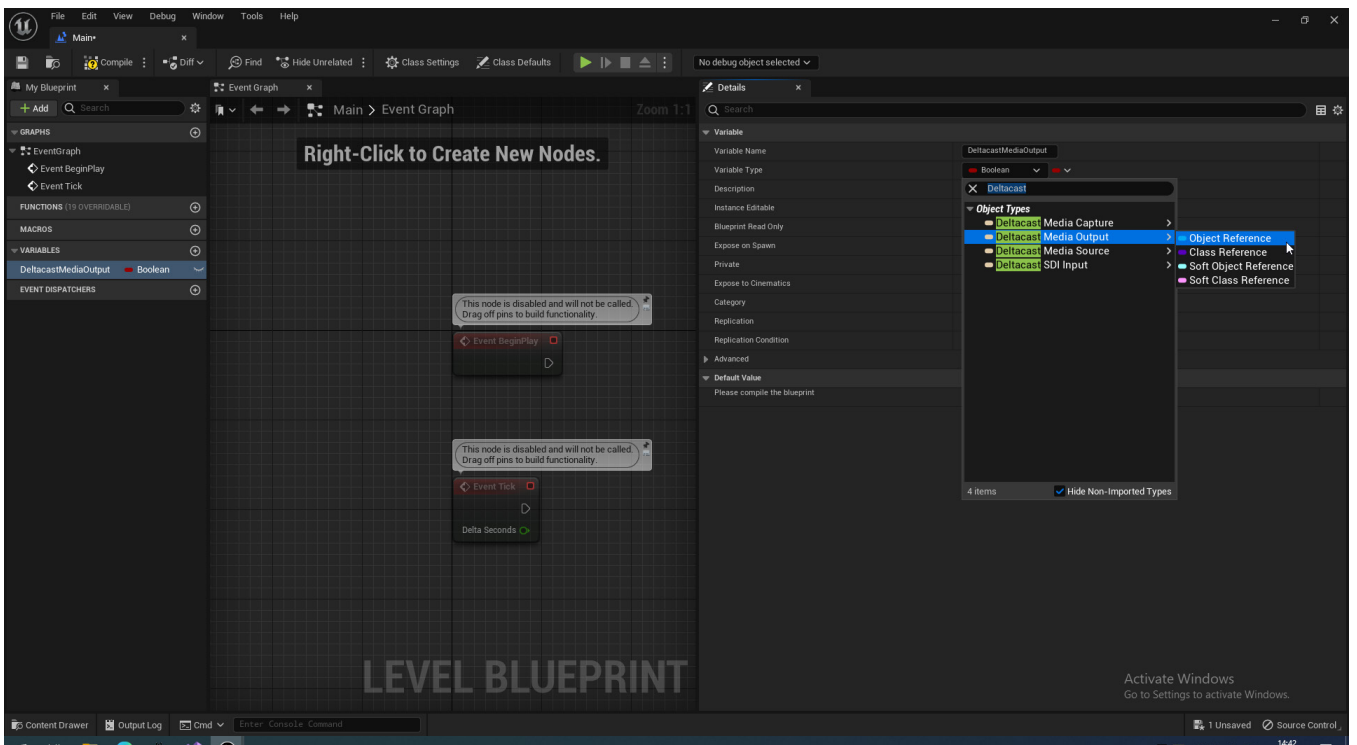
1. From the main toolbar, choose **Blueprints > Open Level Blueprint**



2. Add a new variable

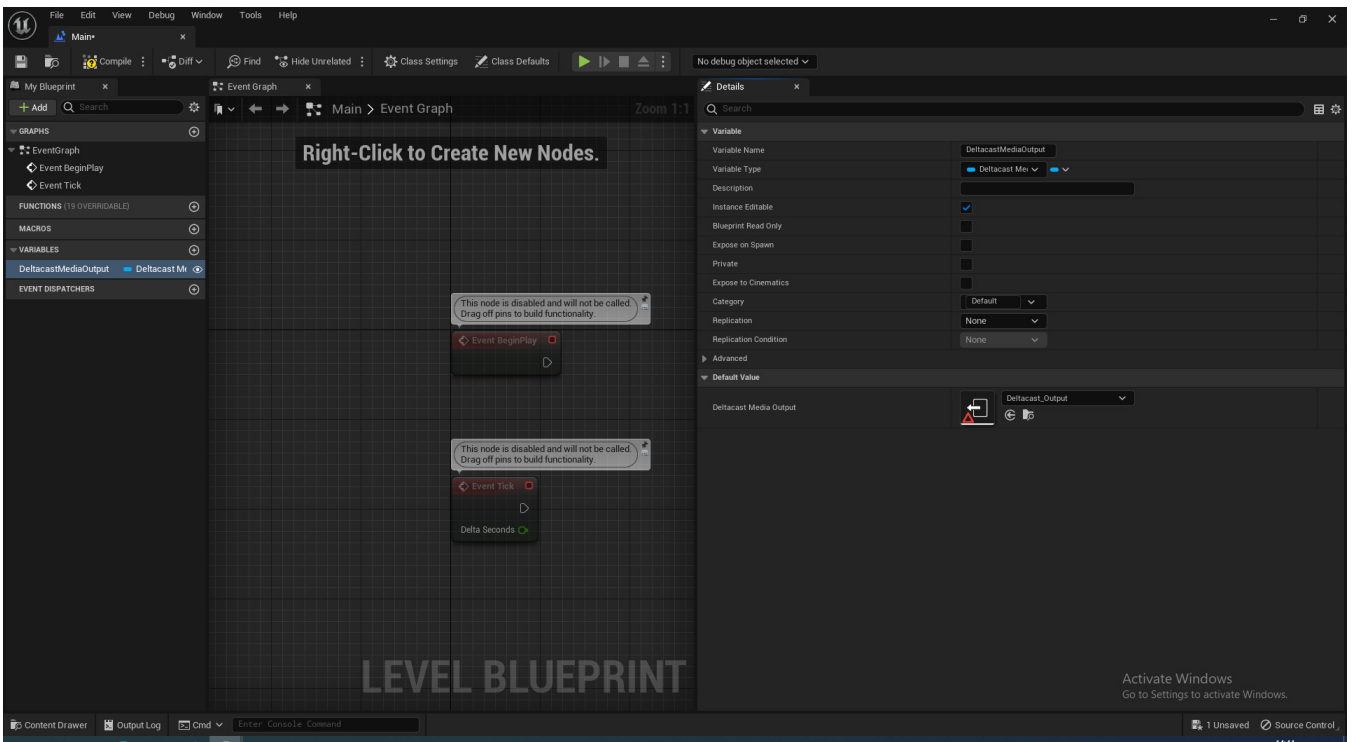


3. Set the **Variable Type** to a Deltacast Media Output **Object Reference**

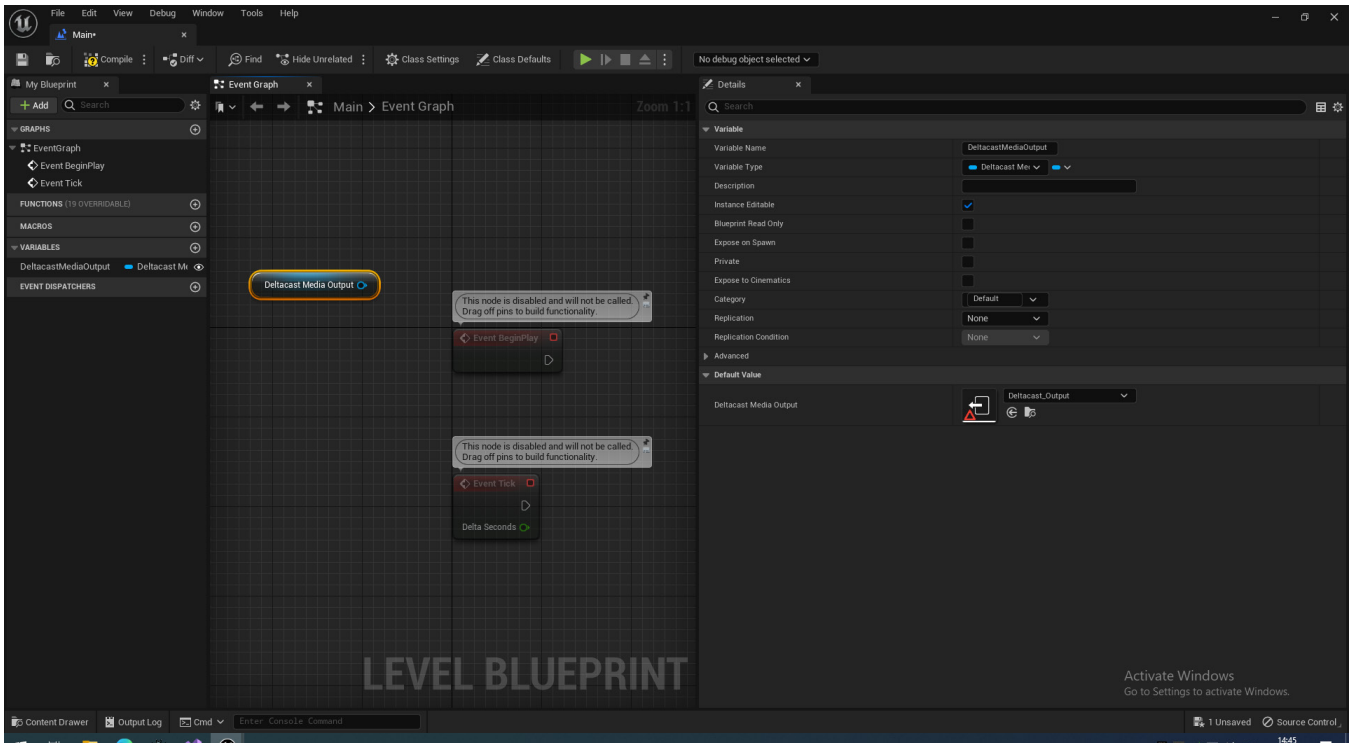


4. Set **Instance Editable** to true by checking the checkbox

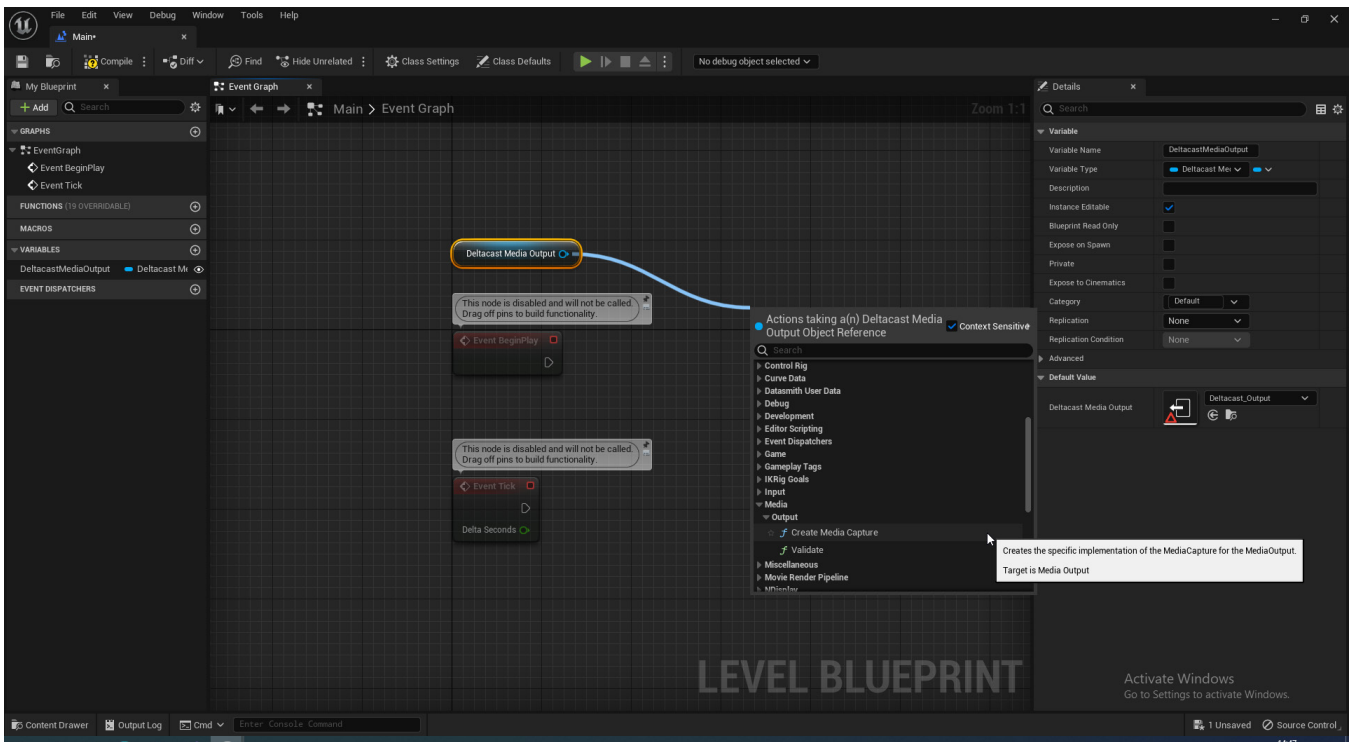
5. Set the **Default Value** to the Deltacast Media Output created in the previous section



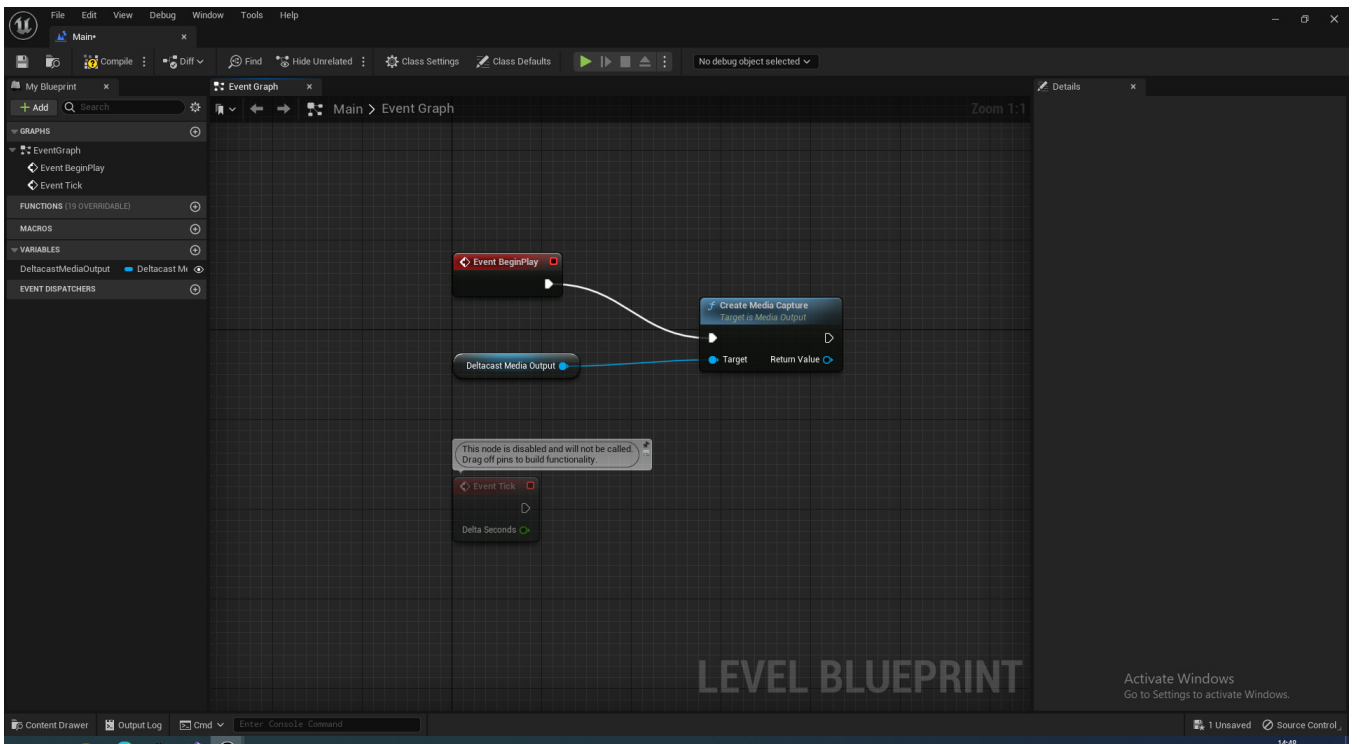
6. Press **Ctrl** and drag the variable into the **Event Graph**



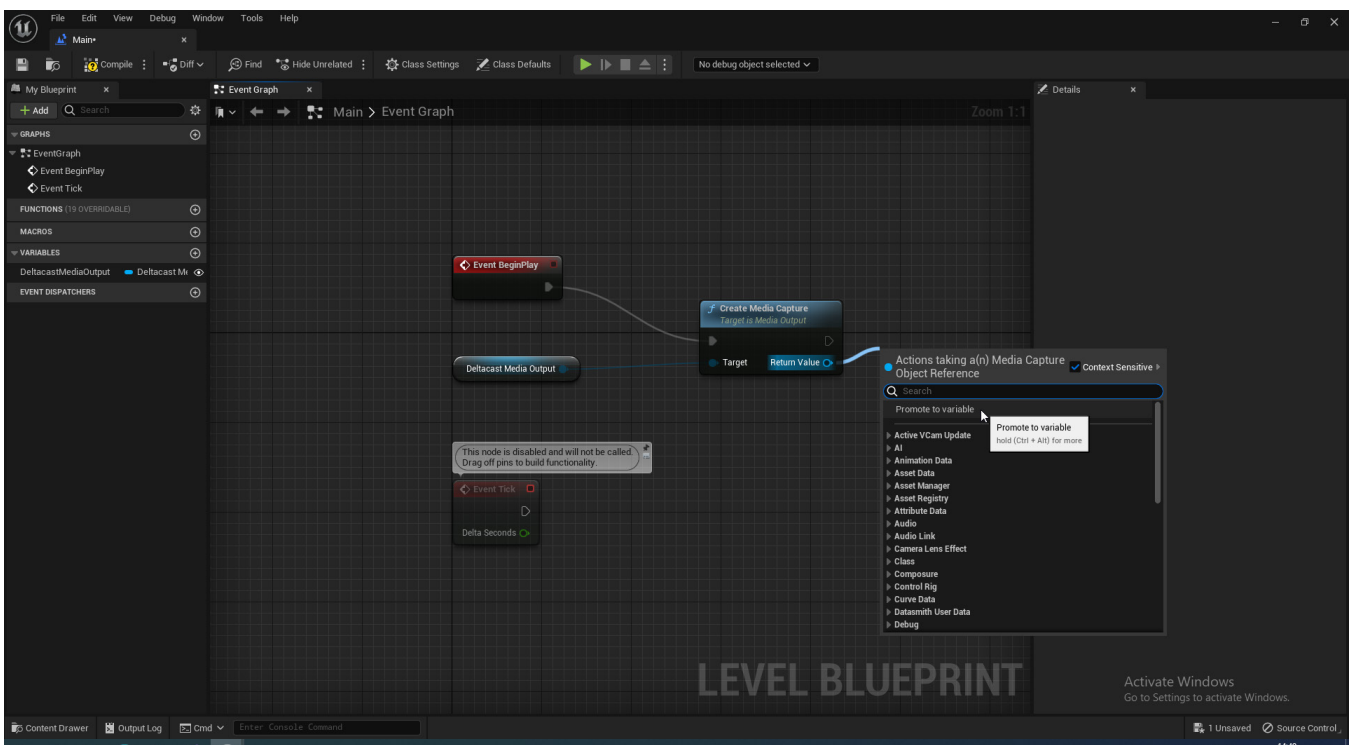
7. Drag from the output port of the Deltacast Media Output and choose **Media > Output > Create Media Capture**



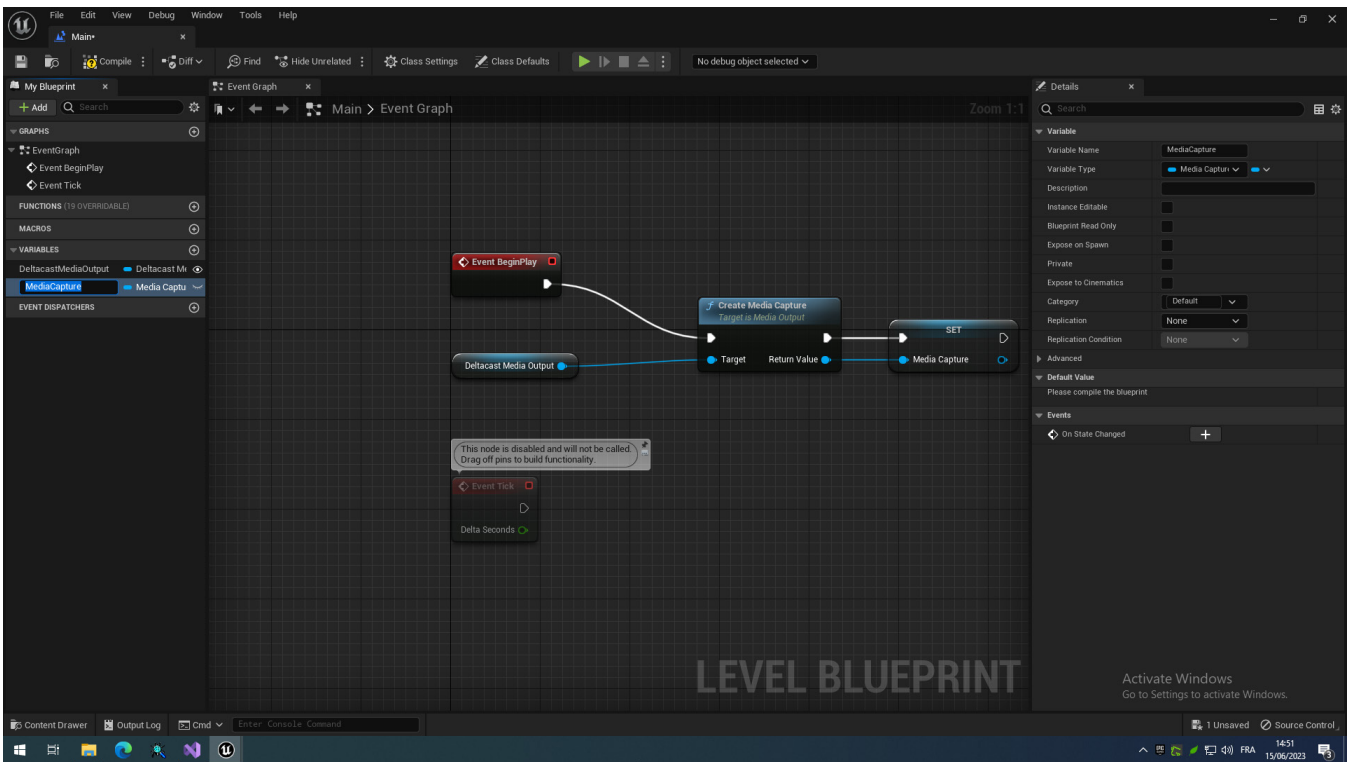
8. Drag from the **Event BeginPlay** output port to the remaining input port of the Create Media Capture node



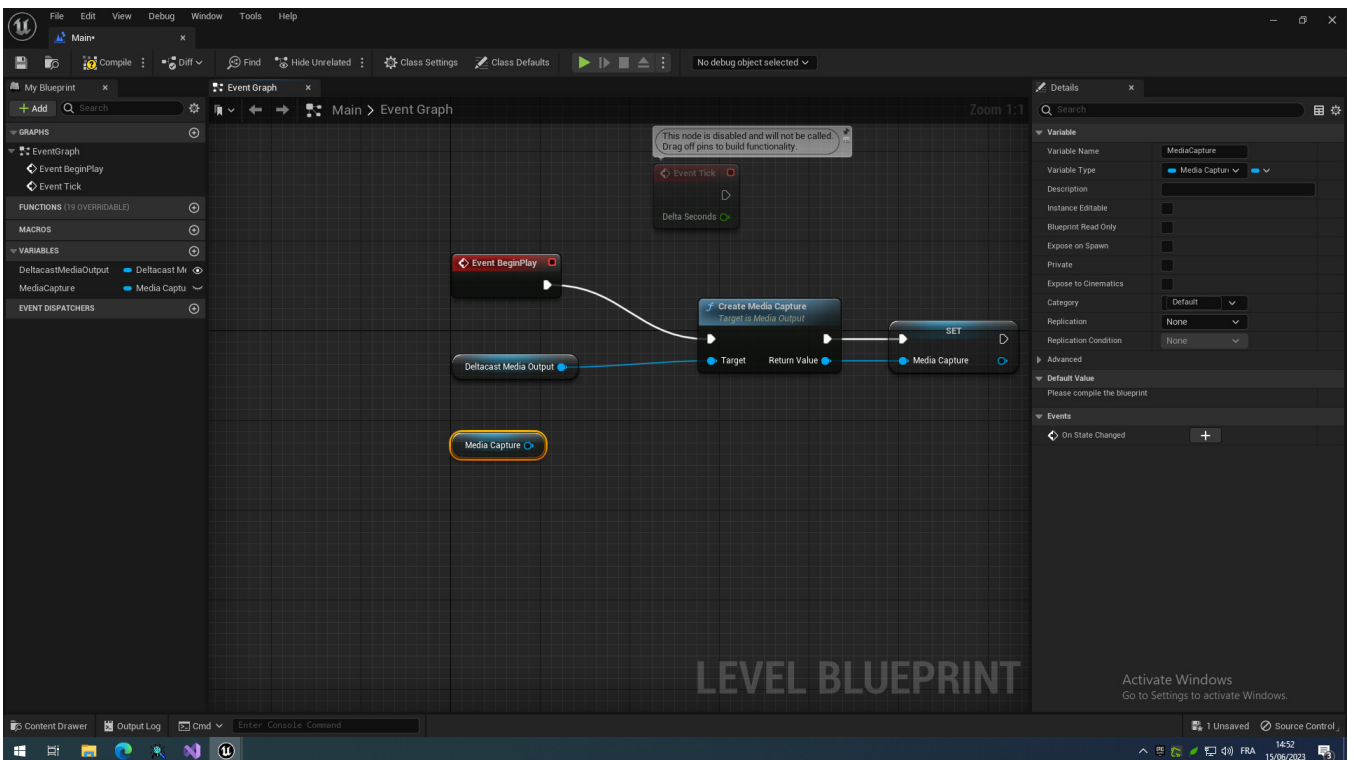
9. Drag from the **Return Value** port of Create Media Capture and select **Promote to variable**



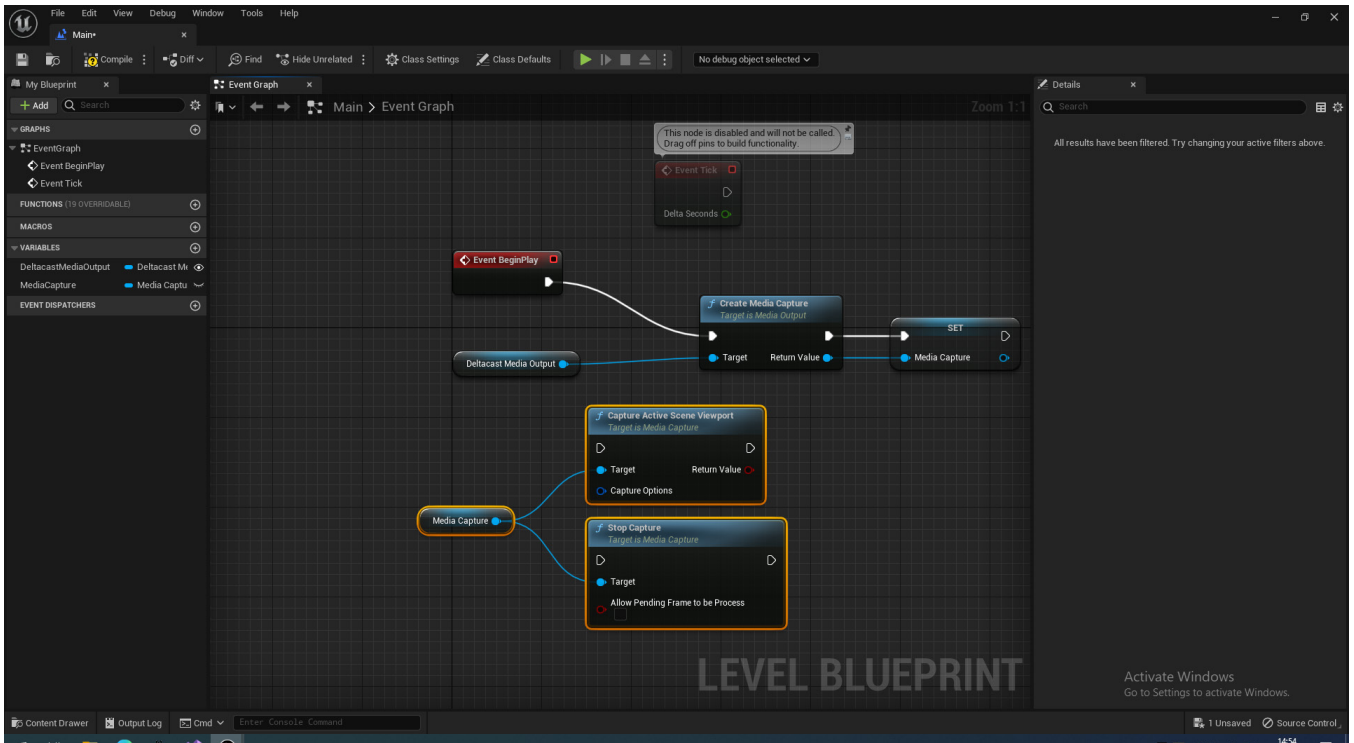
10. Rename the variable to **MediaCapture**



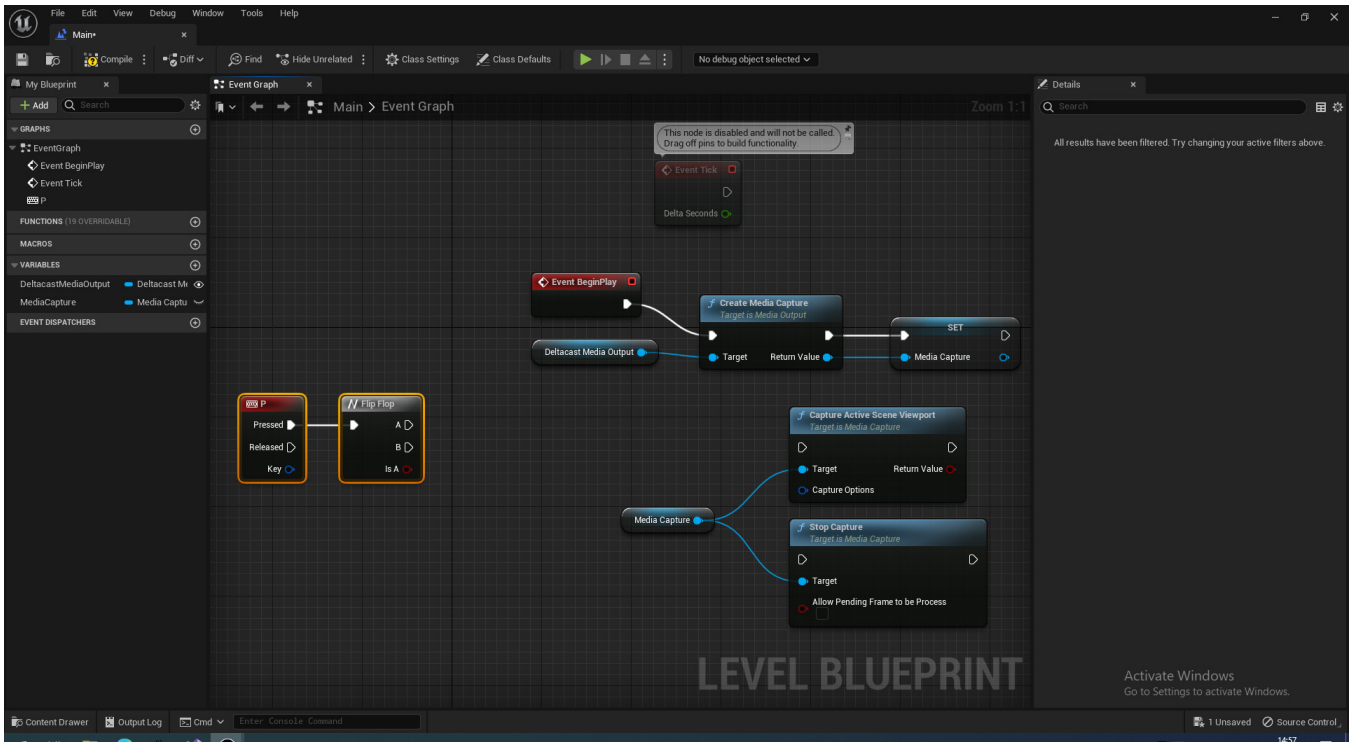
11. Press Ctrl and drag the MediaCapture variable into the **Event Graph**



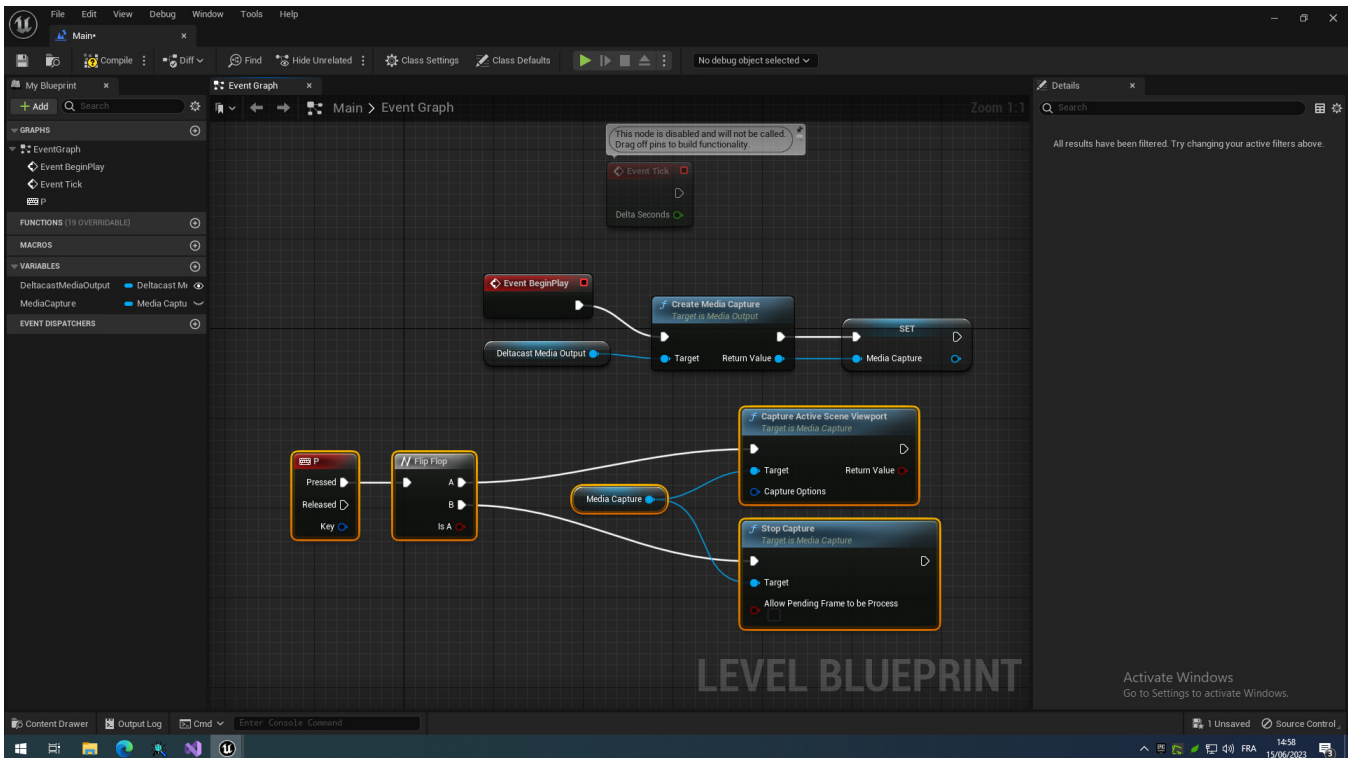
- Drag from the MediaCapture's output port and choose **Media > Output > Capture Active Scene Viewport**. Repeat once more while choosing **Media > Output > Stop Capture**



- Right-click in the **Event Graph** and choose **Input > Keyboard Events > P**
- Drag from the **Pressed** output port of the **P** node and choose **Flow Control > FlipFlop**

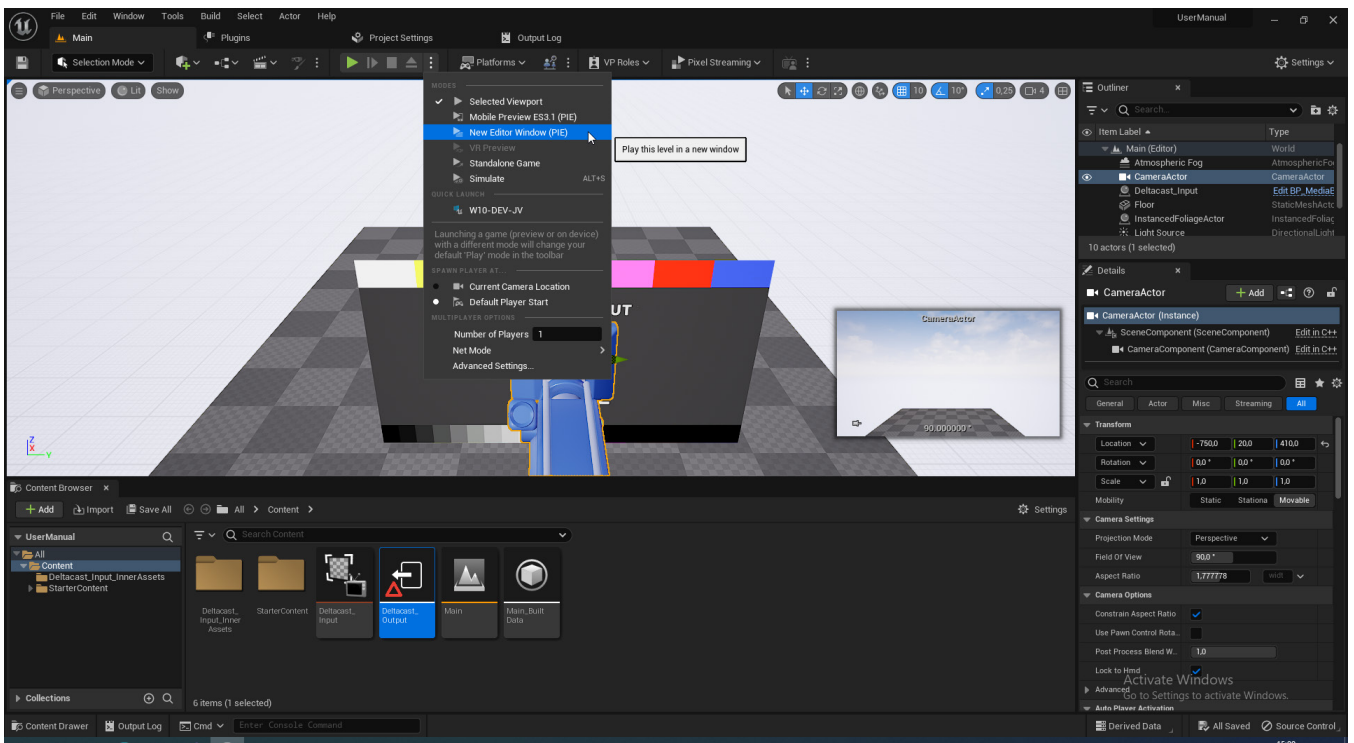


15. Connect respectively the FlipFlop **A** and **B** output ports to the input ports of **Capture Active Scene Viewport** and **Stop Capture**



16. Compile and save the blueprint

17. From the main toolbar, choose either ... > **New Editor Window (PIE)** or ... > **Standalone Game**



18. Once the game has started, pressing **P** will toggle between capturing and stopping the current capture. Note: the viewport size should match output stream resolution.

1.5.3 DELTACAST MEDIA OUTPUT SETTINGS

Each DELTACAST Media Output exposes the following settings:

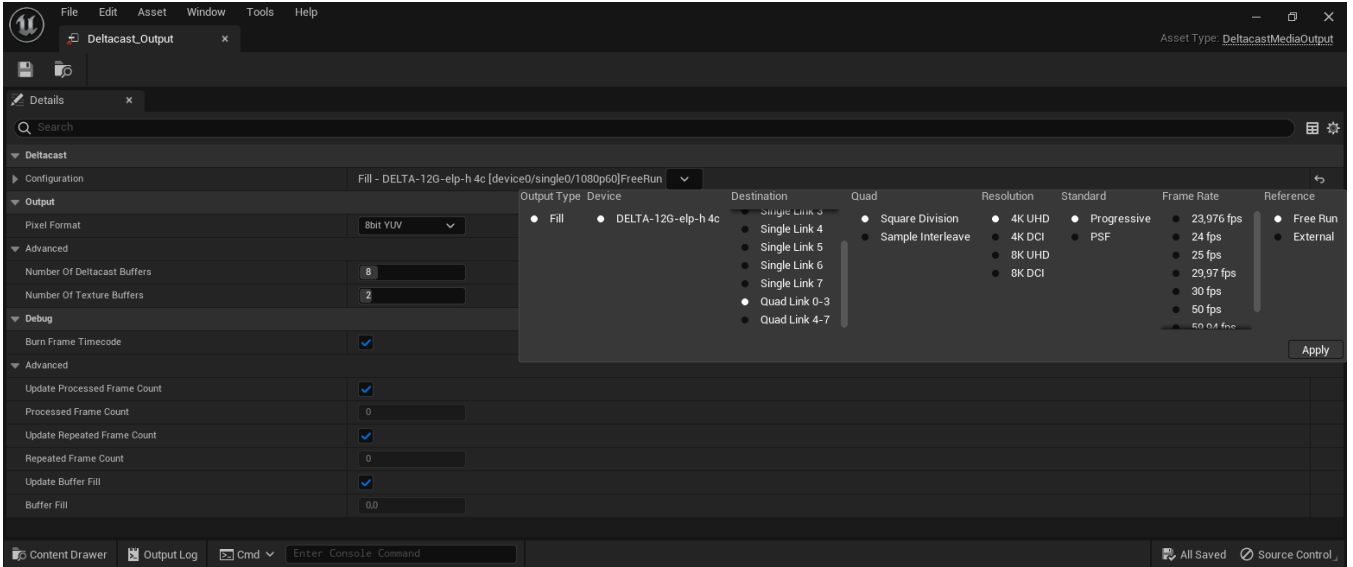


Figure 2: Deltacast Media Output settings

Table 3: Board configurations

Property	Description
Output type	Only Fill is supported.
Device	List of all DELTACAST boards detected in the system.
Destination	List of all available connections for the selected device. HDMI destination is used for both HDMI and DisplayPort connections.
Quad	List of all available quad link configurations for the selected destination. Only available when using a quad link destination.
Resolution	List of all available resolution for the selected destination.
Standard	List of all available standard for the selected resolution.
Frame rate	List of all available frame rate for the selected standard.
Reference	List of reference to be used to synchronize the video output. External is used when a synchronization should be made. The configuration of the reference source is done through the plugin settings (see 1.8).

Table 4: Media Output configurations

Property	Description
Pixel format	Format of the output signal. 8Bits ARGB is not supported with SDI streams.
Number of Deltacast buffers	Number of frame used by the Deltacast SDK. A smaller number is most likely to cause missed frame. A bigger number is most likely to increase latency.
Number of Texture buffers	Number of frame used by Unreal Engine. A smaller number is most likely to cause missed frame. A bigger number is most likely to increase latency.

Burn frame timecode	Whether to incrust the value of the timecode in each frame
Update Processed Frame Count	Whether to update the counter of processed frames.
Processed Frame Count	The number of processed frames since the start of the capture. Zero if update is disabled.
Update Repeated Frame Count	Whether to update the counter of repeated frames.
Repeated Frame Count	The number of repeated frames since the start of the capture. Zero if update is disabled.
Update Dropped Frame Count	Whether to update the counter of dropped frames.
Dropped Frame Count	The number of dropped frames since the start of the capture. Zero if update is disabled.

1.6 TIMECODE

1.6.1 VISUALIZING TIMECODE

The timecode can be visualized either on top of the viewport or in a dedicated panel.

To enable or disable the viewport visualization, use the following console command: `stat timecode`

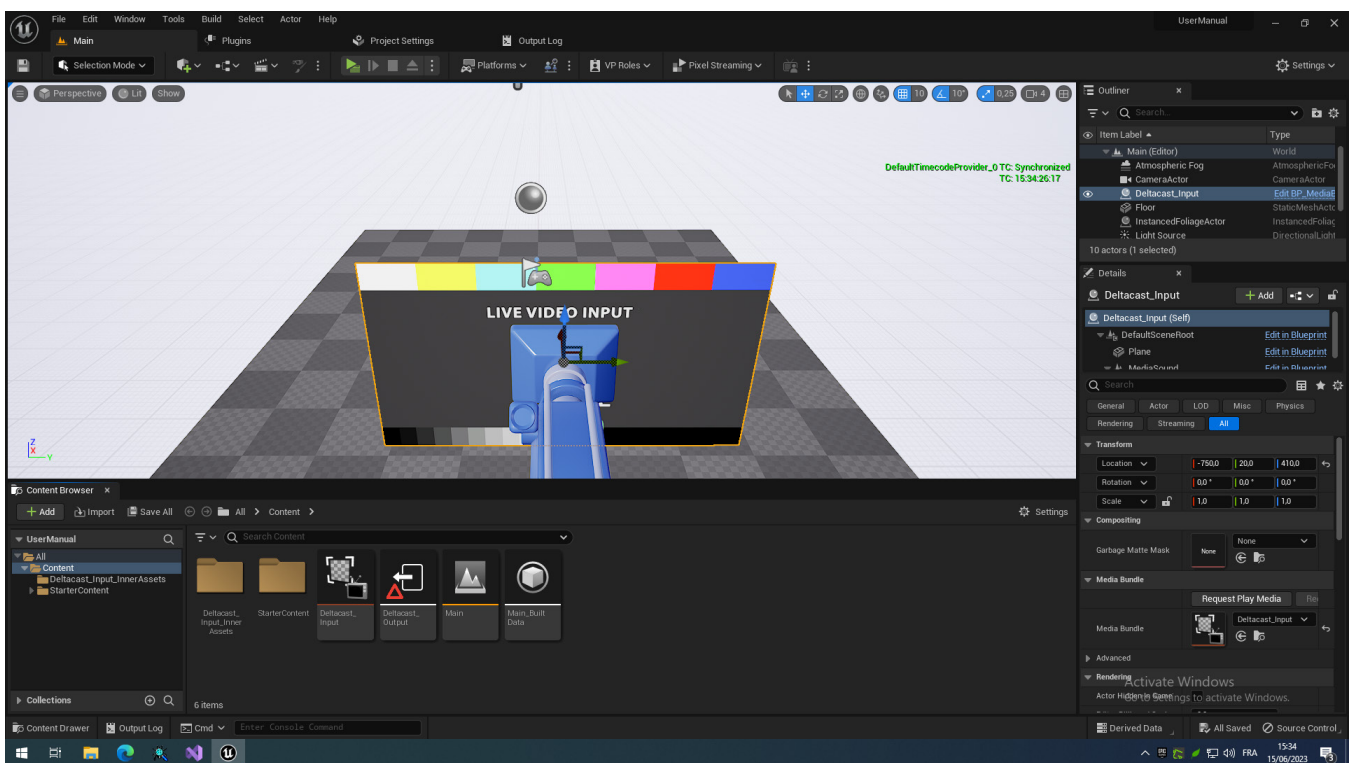


Figure 3: Timecode on top of the viewport

To show the dedicated panel, choose from the main menu **Window > Virtual Production > Timecode Provider**.

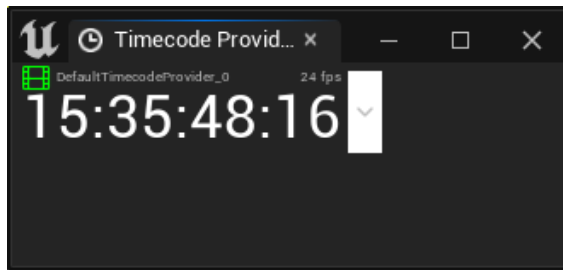
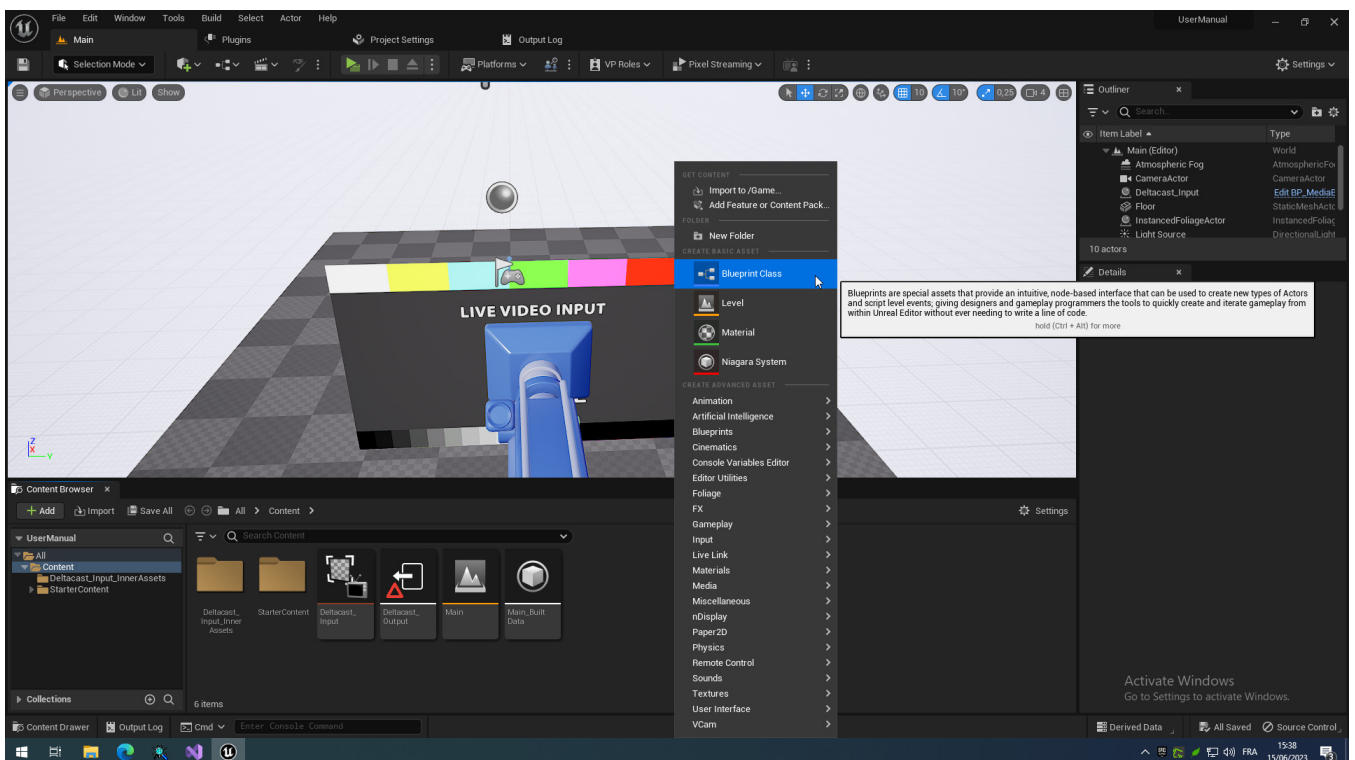


Figure 4: Timecode dedicated panel

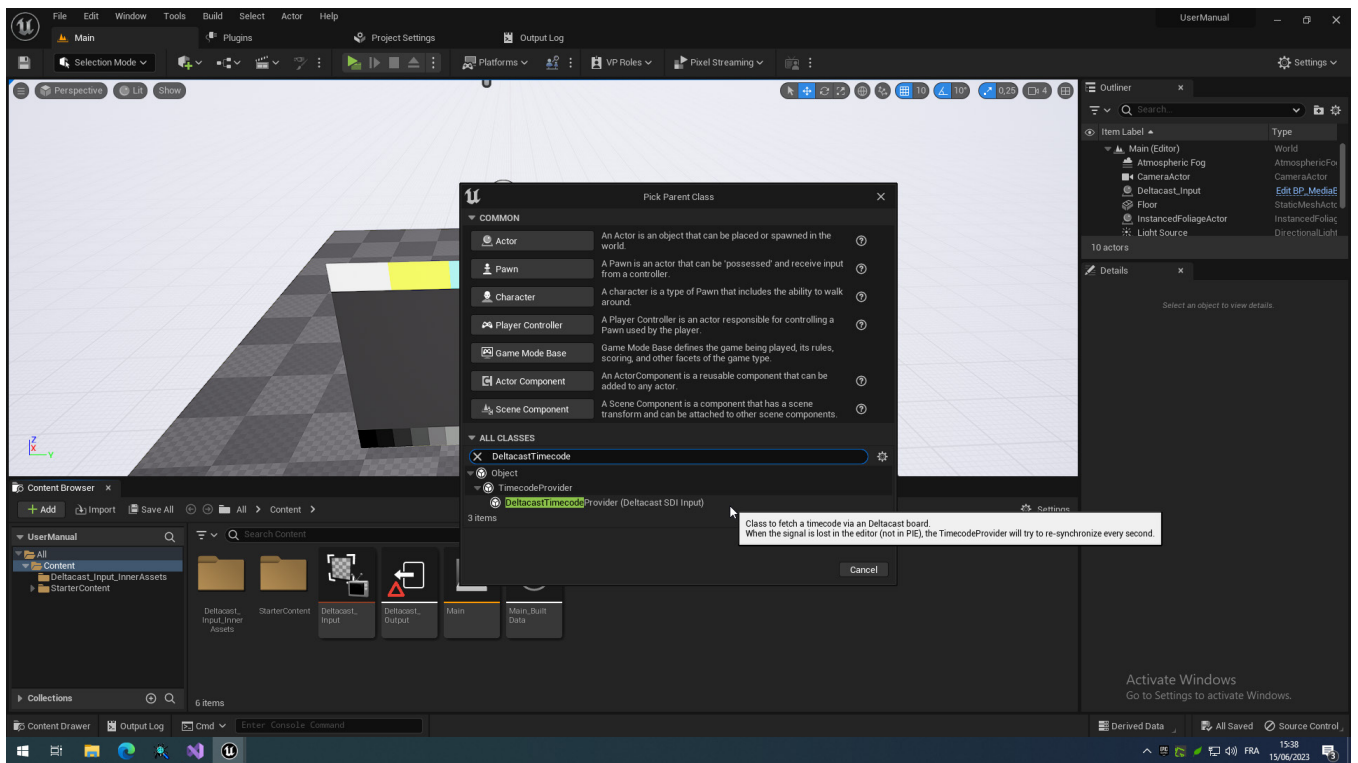
1.6.2 OVERRIDING THE ENGINE TIMECODE

To override the engine timecode, follow these steps:

1. In the **Content Browser**, right-click and choose **Blueprint Class**



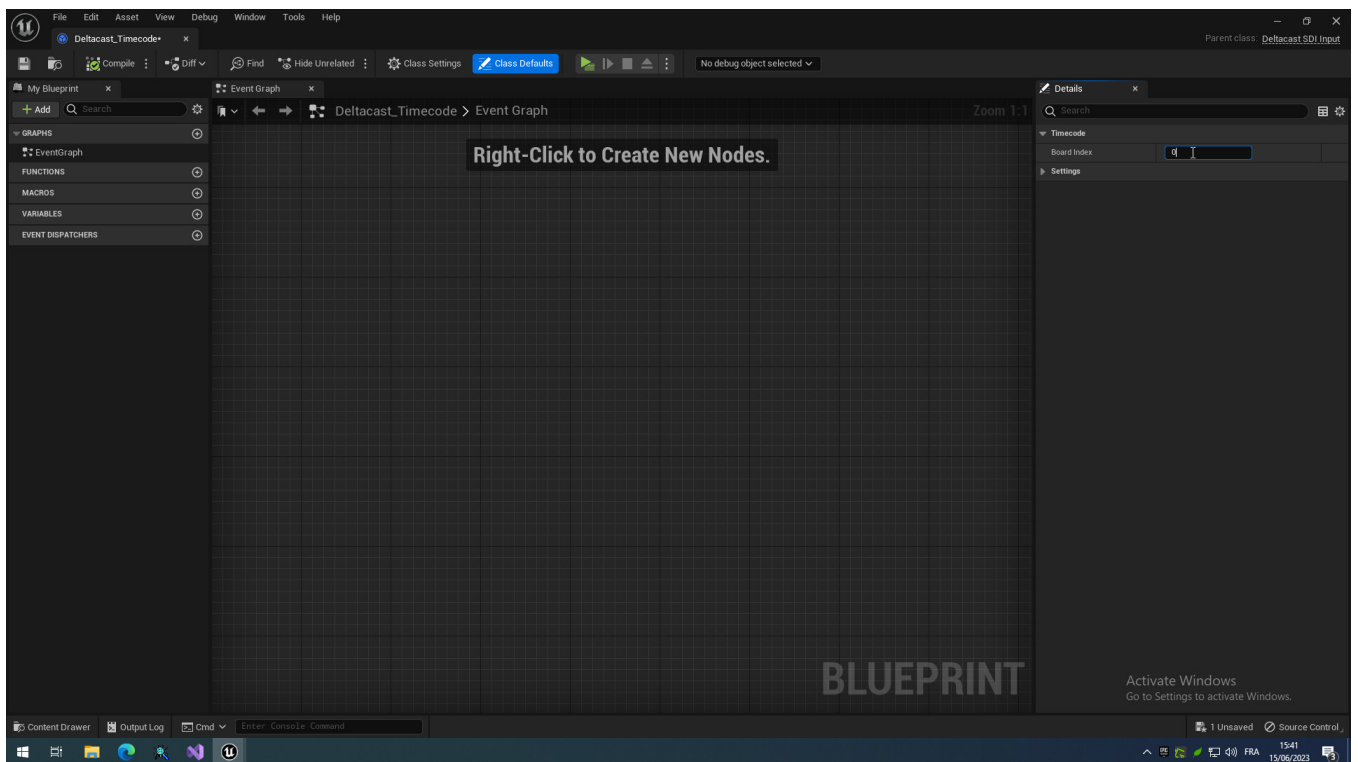
2. Under **All Classes**, search for and choose **DeltacastTimecodeProvider**



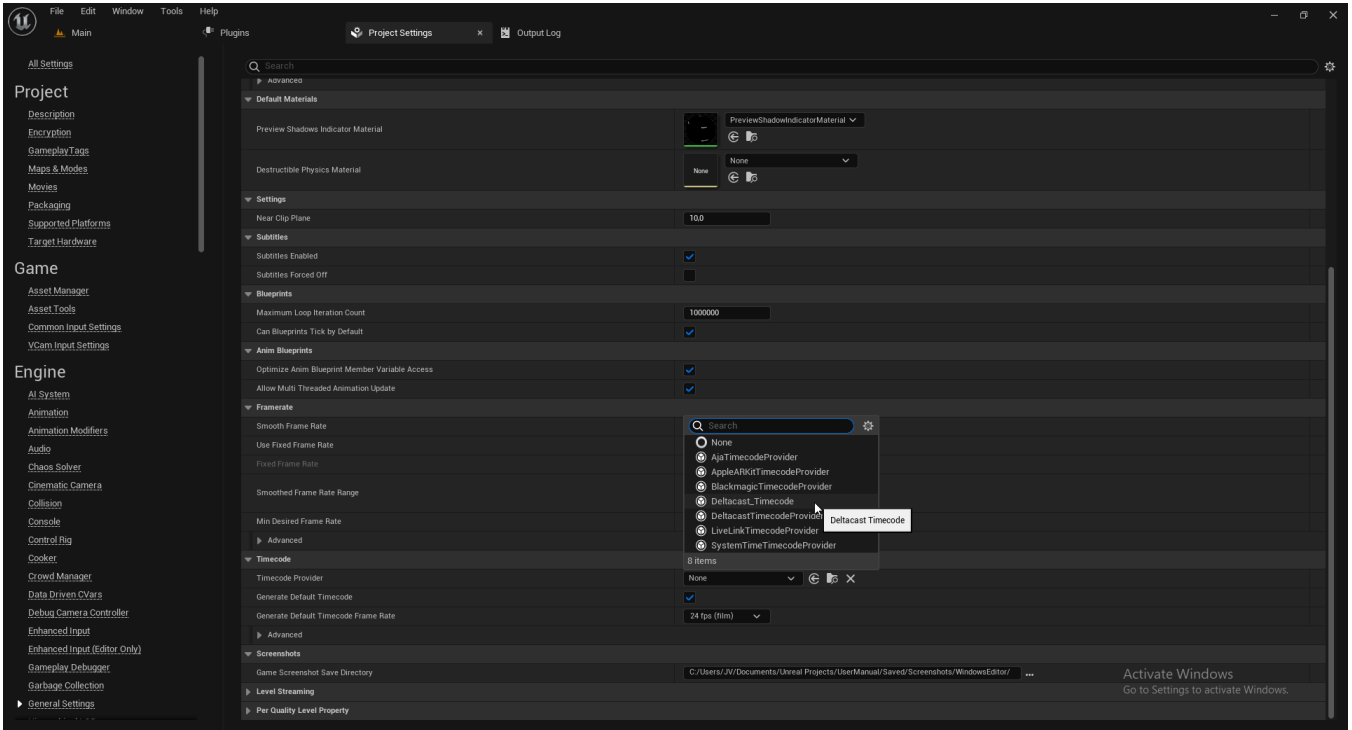
3. Rename the newly created blueprint asset

4. Double-click on the newly created blueprint asset to open the **Blueprint Editor**

5. Under **Timecode**, choose the board index receiving the timecode



6. Compile and save the blueprint
7. From the main menu, choose **Edit > Project Settings** then under **Plugins > Deltacast Media** and configure the timecode source for the board receiving it (see 1.8 for more details)
8. Under **Engine > General Settings** then in the timecode category set the **Timecode Provider** to the blueprint asset newly created



1.7 GENLOCK (CUSTOM TIME STEP)

1.7.1 VISUALIZING FRAME RATE

The Engine frame rate is visualized on top of the viewport, to toggle its display use the following console command: **stat fps**

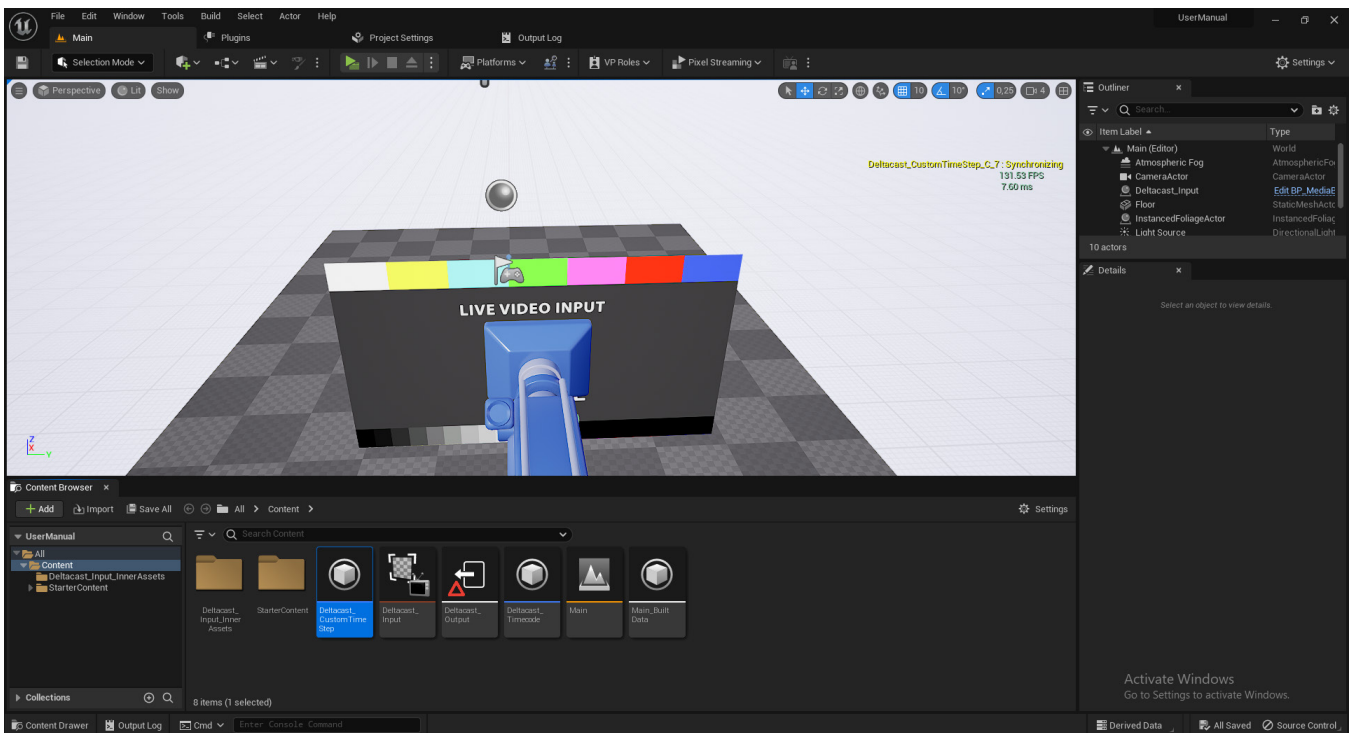
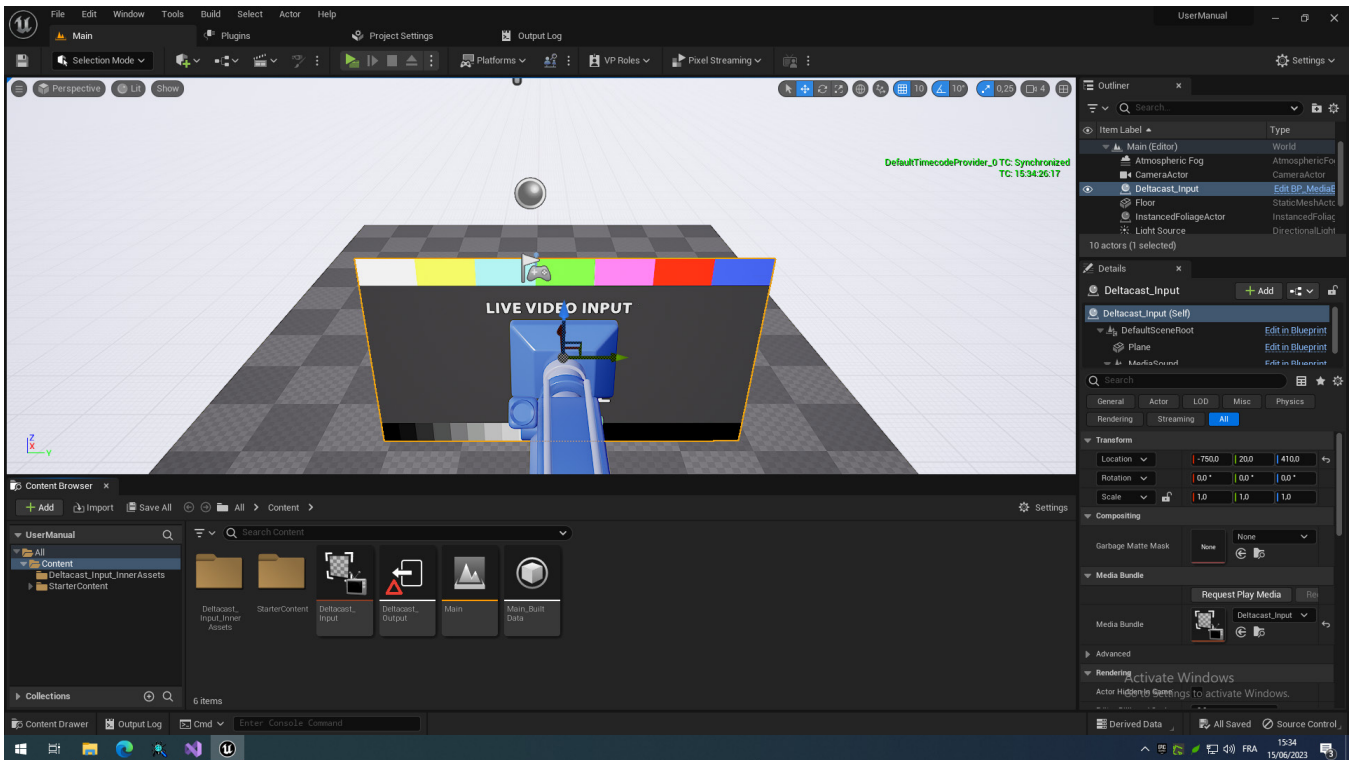


Figure 5: Engine frame rate on top of the viewport (with custom time step status)

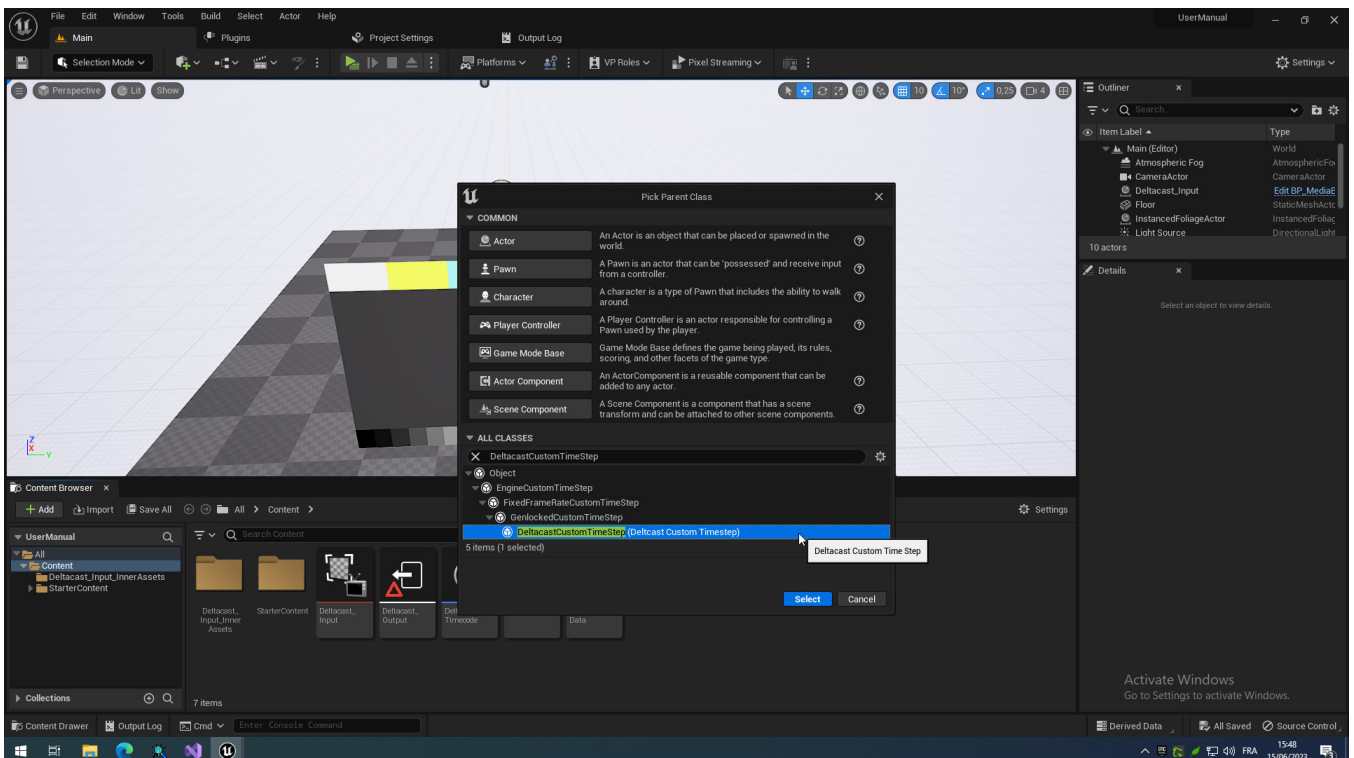
1.7.2 GENLOCKING THE ENGINE

To override the Engine time step, follow these next steps:

1. In the **Content Browser**, right-click and choose **Blueprint Class**

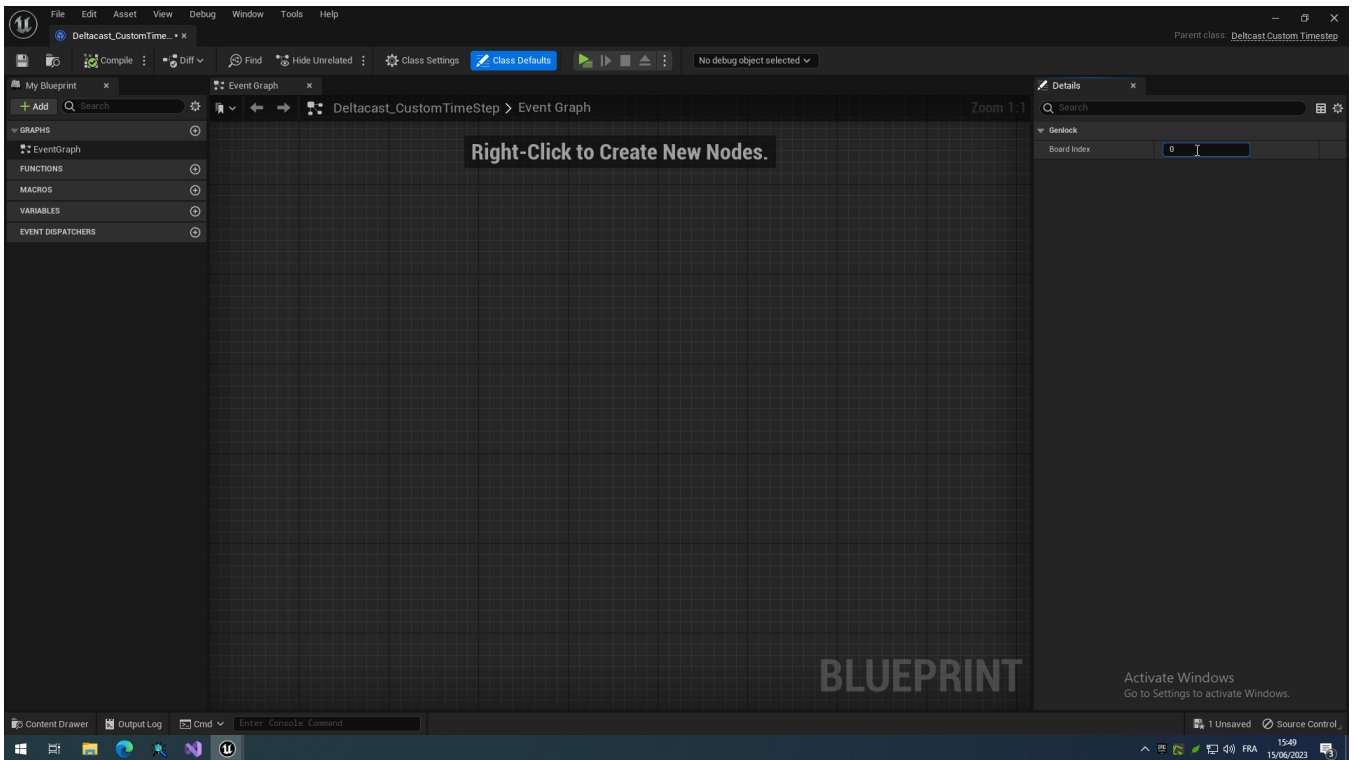


2. Under **All Classes**, search for and choose **DeltacastCustomTimeStep**



3. Rename the newly created blueprint asset
4. Double-click on the newly created blueprint asset to open the **Blueprint Editor**

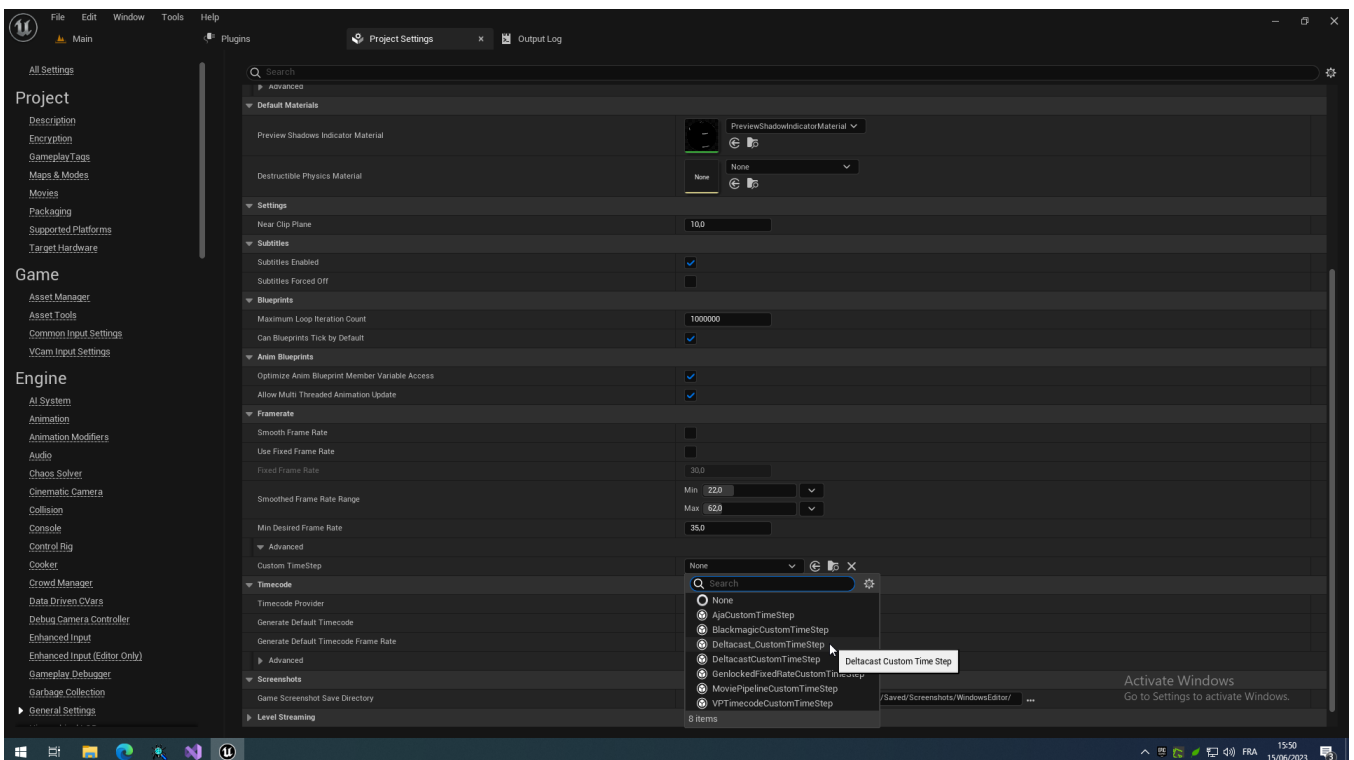
5. Under **Genlock**, choose the board index using as source for the genlock



6. Compile and save the blueprint

7. From the main menu, choose Edit > Project Settings then under Plugins > Deltacast Media and configure the timecode source for the board receiving it (see 1.8 for more details)

8. Under **Engine > General Settings** then in the framerate category under **Advanced** set the **Custom TimeStep** to the blueprint asset newly created



1.8 PLUGIN SETTINGS

The plugin settings contains board specific settings that are common to all inputs and outputs.

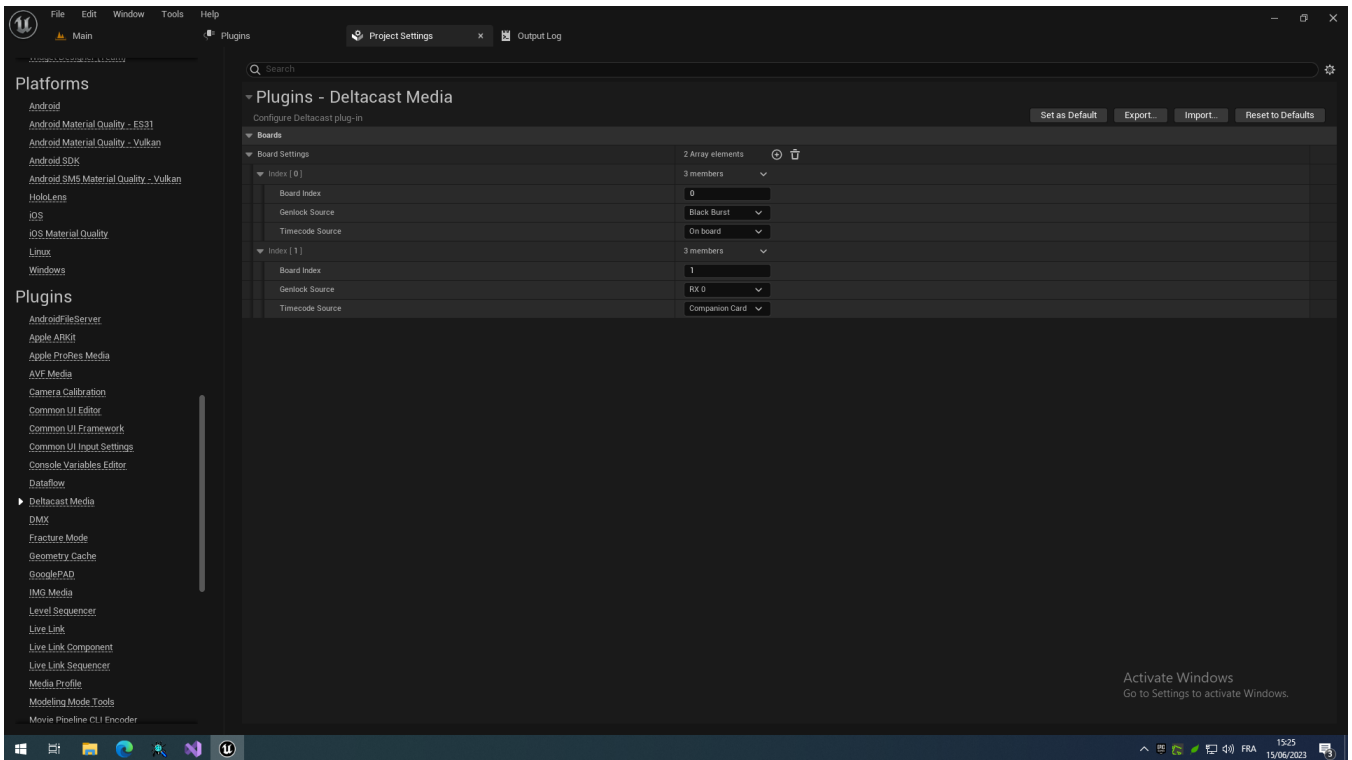


Figure 6: Plugin settings

Table 5: Board common settings

Property	Description
Board Index	The board index for which the following settings should apply.
Genlock source	One of the following possibilities: Local, BlackBurst or RXi (i being a port index).
Timecode Source	Either on board or on a companion card

DELTA Δ CAST

VIDEO SOLUTIONS FOR DEVELOPERS

Rue Gilles Magnée 92/6 | B-4430 Ans | Belgium
T : +32 (0) 4 239 78 81 | F : +32 (0) 4 239 78 89
contact@deltacast.tv | www.deltacast.tv

