

Unreal/ CWEngine - Documentation

Other Products

Exported on 11/30/2021

Table of Contents

1	Unreal / Cineware Engine	4
1.1	Installation	4
1.1.1	Prerequisites	4
1.1.2	Copy plugins.....	4
1.1.2.1	DatasmithC4DImporter	4
1.1.2.2	CinewareForUnreal	5
1.1.3	Check if it works	5
1.1.4	Possible Errors.....	6
1.2	How to use it.....	7
1.2.1	Cinema 4D Takes.....	7
1.2.2	Datasmith Import.....	11
1.2.3	Cineware Properties Asset.....	11
1.2.4	Edit Values	11
1.2.4.1	Via UI	12
1.2.4.2	Via Blueprints	12
1.2.4.3	Via C++	14
1.2.5	Reimport.....	15
1.3	Feedback	15
1.4	Known Issues.....	15
1.5	Feature list that is not supported in this Release	15

- [Unreal / Cineware Engine](#)(see page 4)
 - [Installation](#)(see page 4)
 - [Prerequisites](#)(see page 4)
 - [Copy plugins](#)(see page 4)
 - [DatasmithC4DImporter](#)(see page 4)
 - [CinewareForUnreal](#)(see page 5)
 - [Check if it works](#)(see page 5)
 - [Possible Errors](#)(see page 6)
 - [How to use it](#)(see page 7)
 - [Cinema 4D Takes](#)(see page 7)
 - [Datasmith Import](#)(see page 11)
 - [Cineware Properties Asset](#)(see page 11)
 - [Edit Values](#)(see page 11)
 - [Via UI](#)(see page 12)
 - [Via Blueprints](#)(see page 12)
 - [Setting properties by name](#)(see page 12)
 - [Reading properties by name](#)(see page 13)
 - [Browsing for properties](#)(see page 14)
 - [Via C++](#)(see page 14)
 - [Reimport](#)(see page 15)
 - [Feedback](#)(see page 15)
 - [Known Issues](#)(see page 15)
 - [Feature list that is not supported in this Release](#)(see page 15)

1 Unreal / Cineware Engine

This project aims to improve integration of Cinema 4D and Unreal via the implementation of Cineware Engine. This generates the scene on-the-fly, eliminating the need for Cineware caches. Thanks to the flexibility this offers, there's a new option to modify Cinema 4D attributes directly within Unreal Engine.

Please make sure to read the guide carefully. Expect errors, bugs and general frustration. There is still some unpolished behavior, before reporting a bug check the *Known Issues* segment at the end of this document.

1.1 Installation

1.1.1 Prerequisites

The plugin is supported on both platforms Mac and Windows

If you do not have Unreal Engine installed, head over to <https://www.unrealengine.com/en-US/get-now>. Select your appropriate license, create an Epic Games Account, and download the Epic Games Launcher. Run the installer and open the launcher. Enter your credentials and select *Unreal Engine* in the left, navigation menu. Select the register *Library*. You should see a content area labeled *Engine Versions*. Click the plus sign to install the version needed.

The shipped plugin has been compiled for Unreal Editor Version 4.27.1 with BuildID 17155196. Downloading the latest version of 4.27.1 should fulfil these criteria. Unzipping the received package reveals two directories which will need to be copied into the Unreal environment for the setup to work.

1.1.2 Copy plugins

1.1.2.1 DatasmithC4DImporter

Datasmith is a project by Epic Games to allow the company's products to import a broad range of proprietary file formats, including .c4d. Maxon partnered with Epic Games to enhance the functionality of this particular bridge. We forked an earlier state of DatasmithC4D from Epic Games and added a significant number of changes (also means that bugs encountered in this version of the Datasmith importer may already have been fixed in the mainline). These changes will be merged with Epic's current codebase and shipped natively with later versions of Unreal Engine. For now, we have to replace the shipped DatasmithC4DImporter version.

On Windows navigate to your Unreal Engine installation, typically *C:/Program Files/Epic Games/UE_4.27* (Windows)

On Mac navigate to your Unreal Engine installation, typically *Macintosh HD/Users/Shared/Epic Games/UE_4.27* (Mac)

This path will be referred to as *\$UE4_DIR* within this document.

The DatasmithC4DImporter plugin is located at *\$UE4_DIR/Engine/Plugins/Enterprise/*. **IMPORTANT** Make a backup copy of the installed DatasmithC4DImporter folder. We may need it later. After the storing a copy of the directory, replace it with the DatasmithC4DImporter directory we provided. The path *\$UE4_DIR/Engine/Plugins/Enterprise/DatasmithC4DImporter/* should now contain the directories Binary, Content, Documentation, Intermediate, Resources, Source, and a .uplugin file.

1.1.2.2 CinewareForUnreal

This is a second plugin which stores exposed properties into an asset and allows users to edit these values via UI, Blueprints or C++. Future reimports of the imported .c4d file will apply these values to the scene. This plugin is configured to act as a (shared) project plugin. This means you can either copy this plugin to the Unreal Engine environment to have access to it across projects or copy it to projects only. For simplicity we suggest installing it as a shared plugin and simply not enabling it for specific projects.

Typically, the path for shared plugins is `$UE4_DIR/Engine/Plugins/Marketplace/`. Should this directory not yet exist, please create it. This directory is also where all plugins purchased and installed from the Unreal Engine Marketplace end up.

The path `$UE4_DIR/Engine/Plugins/Marketplace/CinewareForUnreal/` should now contain the directories Binaries, Content, Intermediate, Source and a .uplugin file.

Should you decide not to install it as a shared plugin, rethink your decision... Still?

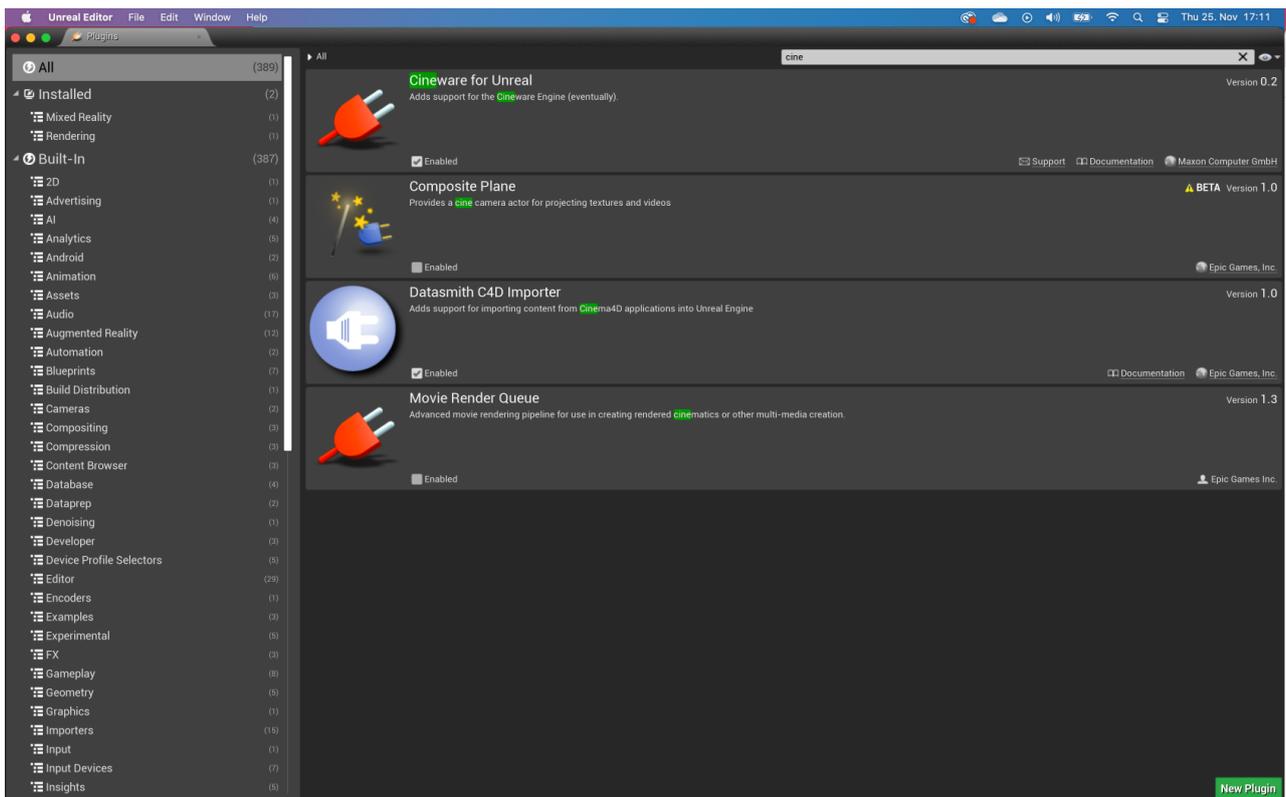
Then create a new project by launching the Unreal Engine and wait for a project hub window to appear. Create a new project with the templates of your liking. Select a location, click create and wait until the Editor has launched. Close the Unreal Editor and navigate to your project directory, from now on referred to as `$PROJ_DIR`.

`$PROJ_DIR/` should contain a bunch of directories which depend on the template you selected and a .uproject file. Create a directory, name it Plugins and copy the provided CinewareForUnreal directory into it. The path `$PROJ_DIR/Plugins/CinewareForUnreal/` should now contain the directories Binaries, Content, Intermediate, Source and a .uplugin file. Restart the project.

1.1.3 Check if it works

Close everything related, except the launcher. Navigate to your installed Unreal Engine version within the launcher and click the button labeled *Launch*. Descriptively, this will launch the Unreal Editor. The first start up of a freshly installed version usually hangs around at 45% for 10 minutes while creating caches and doing fancy preprocess work. Should the loading splash reach 100 percentage points a project hub window opens. Create a new project by selecting template options and a project location. Click *Create* and wait for the Editor to open.

On first start-up the editor will begin its shader compilation process. Give it another 10 minutes to finish this. After, navigate the menu bar to *Edit/Plugins*. This opens a window with a list of available plugins. Use the search bar to look for Cineware for Unreal and Datasmith C4D Importer. Should one of them not be visible, make sure to click on both main categories *Installed* and *Available* in the left list. The plugins will prompt if it is ok to use a beta version, accept. After both plugins have been checked as *Enabled*, restart the Editor and the project. If the Editor successfully restarts you are good to go.



1.1.4 Possible Errors

The following modules are missing or built with a different engine version. Would you like to rebuild them now?

Both plugins are shipped with the source code. To make sure you don't have to manually recompile anything we also shipped the compiled binaries for 4.27.1. As a first step please make sure you are currently working with version 4.27.1. If so, it's likely that an update has occurred between us compiling these plugins and you installing them. The binaries are still compatible, but they are flagged as outdated. To remedy that we'll need the back-up copy of DatasmithC4DImporter that we saved previously.

Open DatasmithC4DImporter/Binaries/Win64/UE4Editor.modules in any text editor. Our plugins have been compiled for "BuildId": "17155196". Should this number differ, copy paste it from the backup to \$UE4_DIR/Engine/Plugins/Enterprise/DatasmithC4DImporter/Binaries/Win64/UE4Editor.modules.

Copy the same number to \$UE4_DIR/Engine/Plugins/Marketplace/CinewareForUnreal/Binaries/Win64/UE4Editor.modules and try again.

Could not be compiled. Try rebuilding from source manually.

Presumably, you encountered the above error message before and clicked Yes. If you already applied the above suggestion and still received this error, you may have to delete certain directories for the compile process to complete. Navigate to \$UE4_DIR/Engine/Plugins/Enterprise/DatasmithC4DImporter/ and delete the directories Binaries, Intermediate and Saved. Navigate to \$UE4_DIR/Engine/Plugins/Marketplace/CinewareForUnreal and do the same. Try again.

Should you encounter any issues please visit the following website: " <https://support.maxon.net>¹ ".

¹ <https://support.maxon.net/>

Plugin failed to load because module could not be found. Please ensure the plugin is properly installed, otherwise consider disabling the plugin for this project.

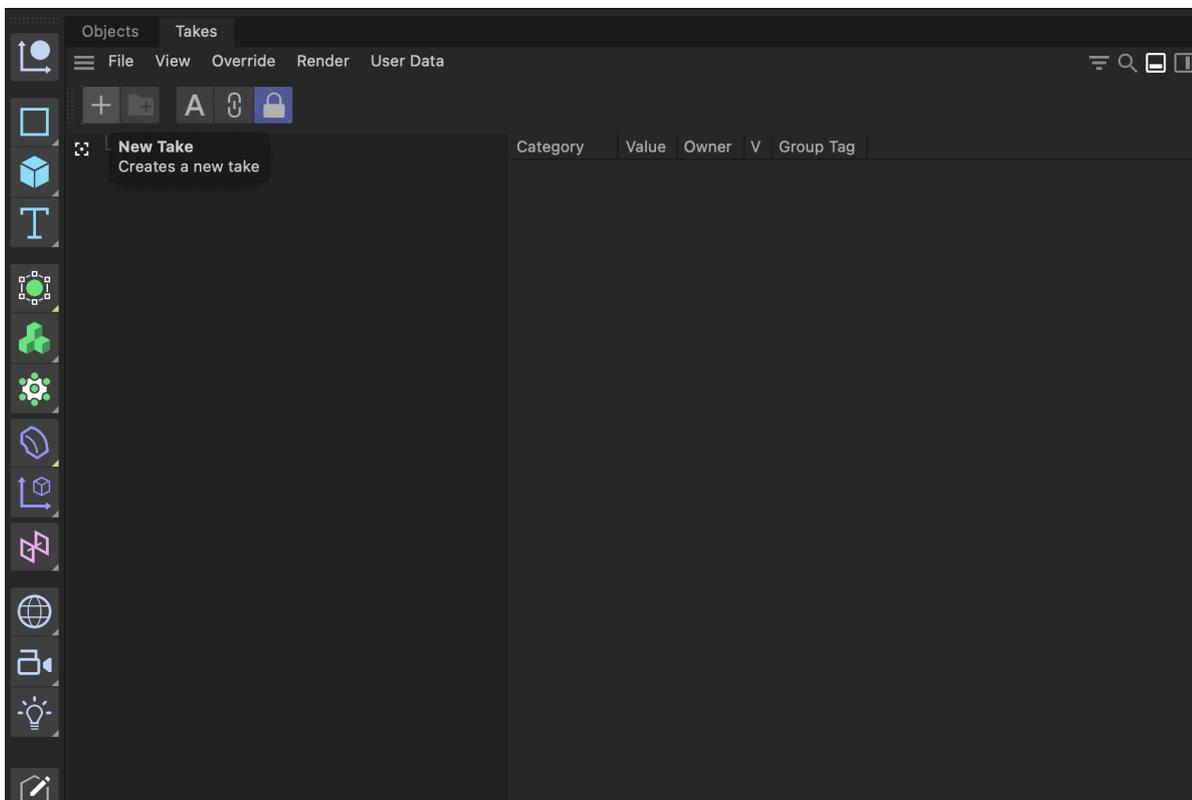
Please switch from installing CinewareAsset as a single project plugin to a shared plugin.

1.2 How to use it

1.2.1 Cinema 4D Takes

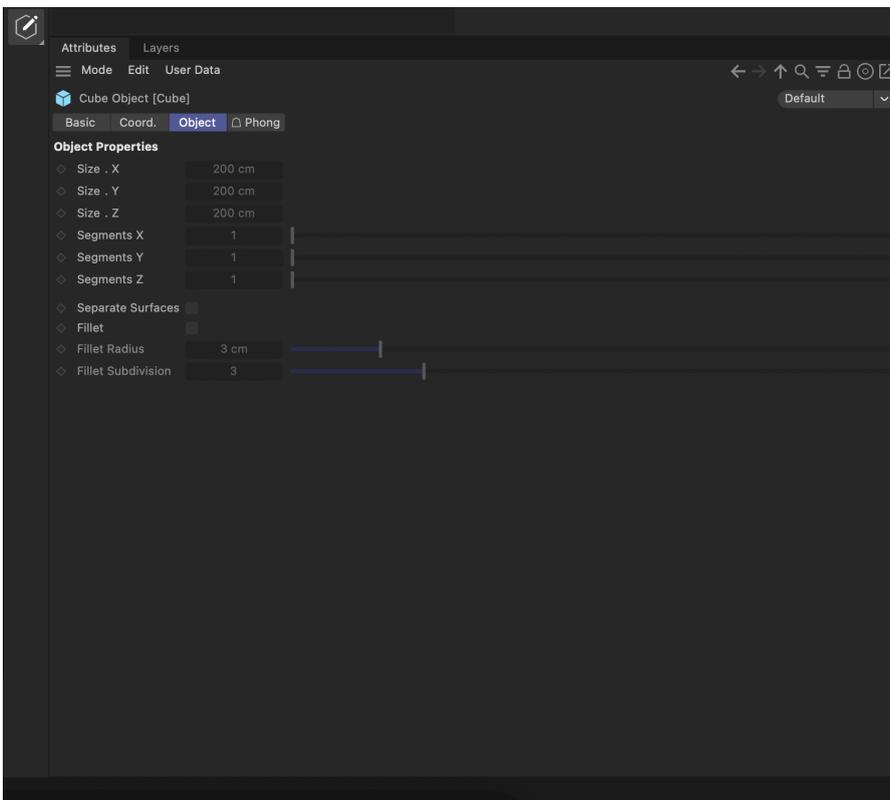
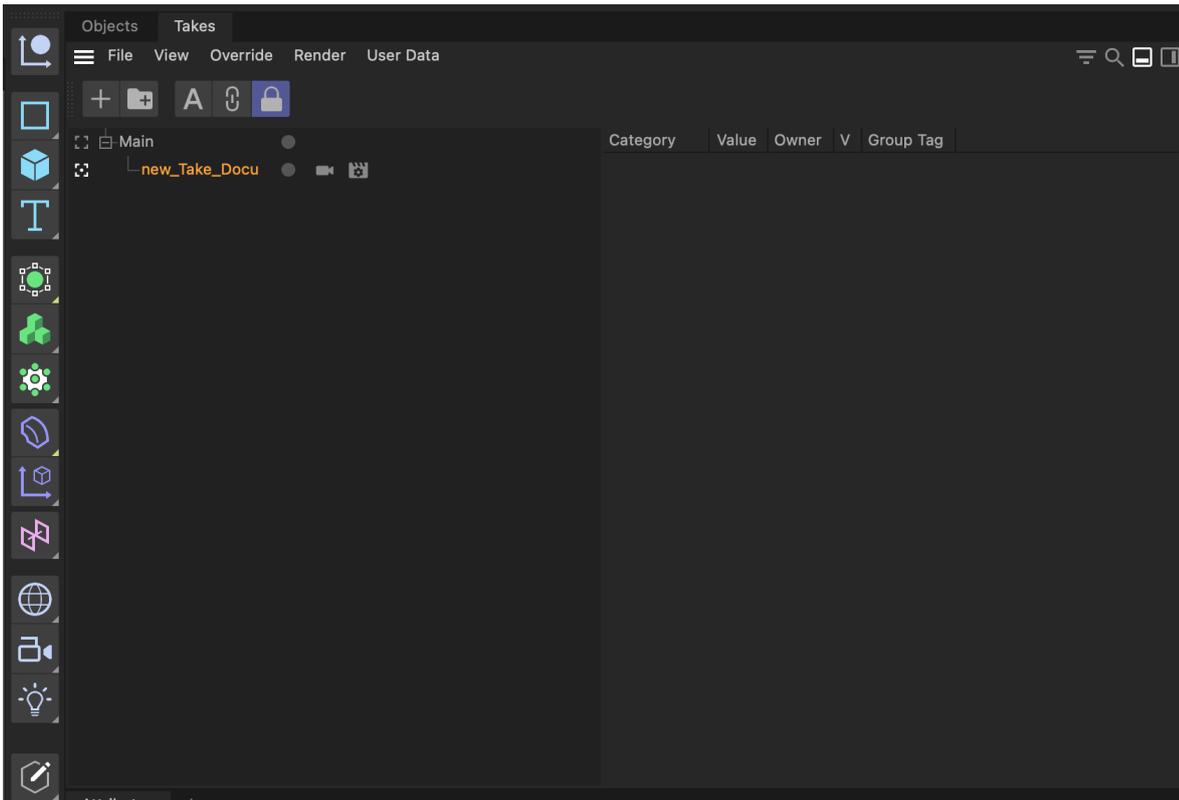
Scene properties can be exposed to Unreal Engine via Cinema 4D's take system. A video tutorial can be found here [by GreyscaleGorilla on Youtube](#)².

Open the Cinema 4D take workspace and click *Add new take*.

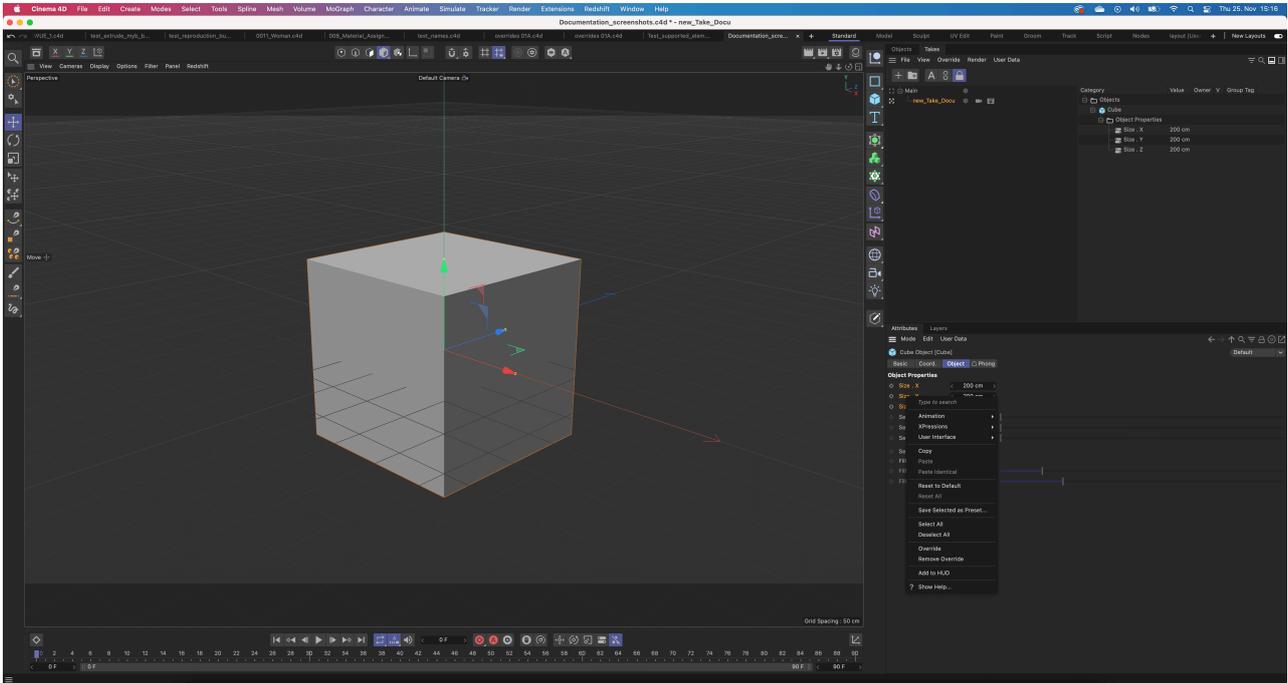


Switch to the state of the take by left-clicking on it. All properties on scene objects should now be greyed out.

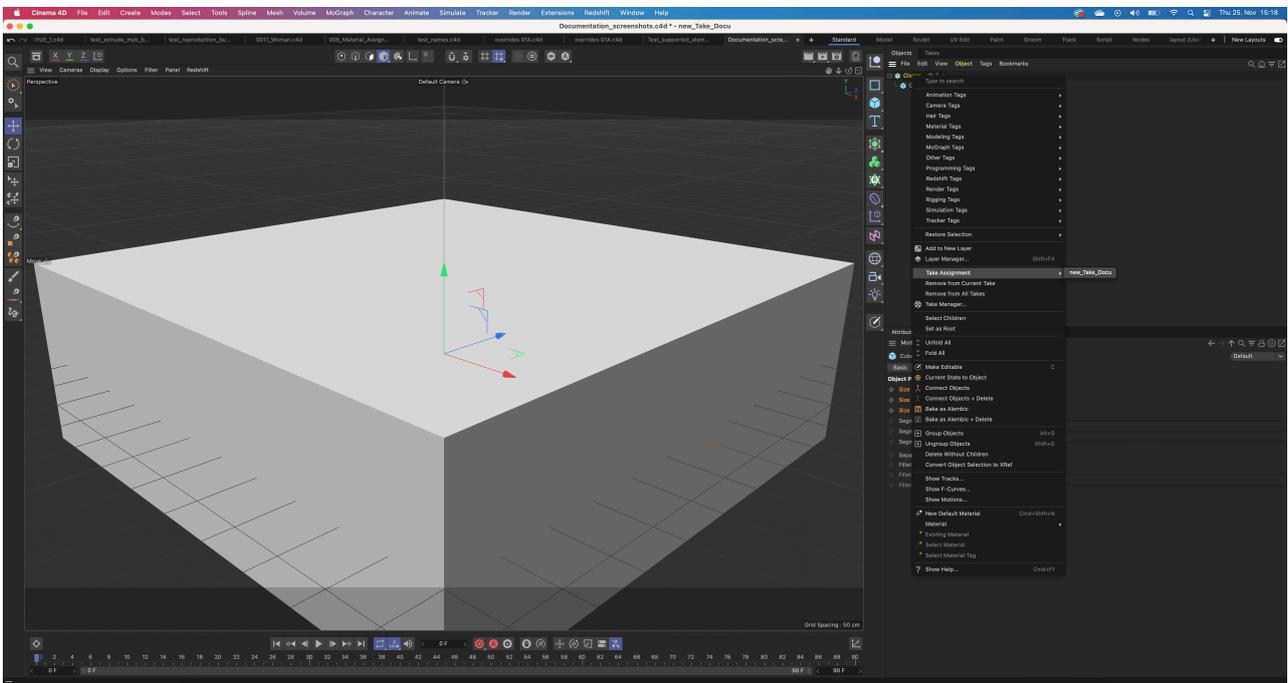
² https://www.youtube.com/watch?v=02rFn4jzDB4&ab_channel=Greyscalegorilla

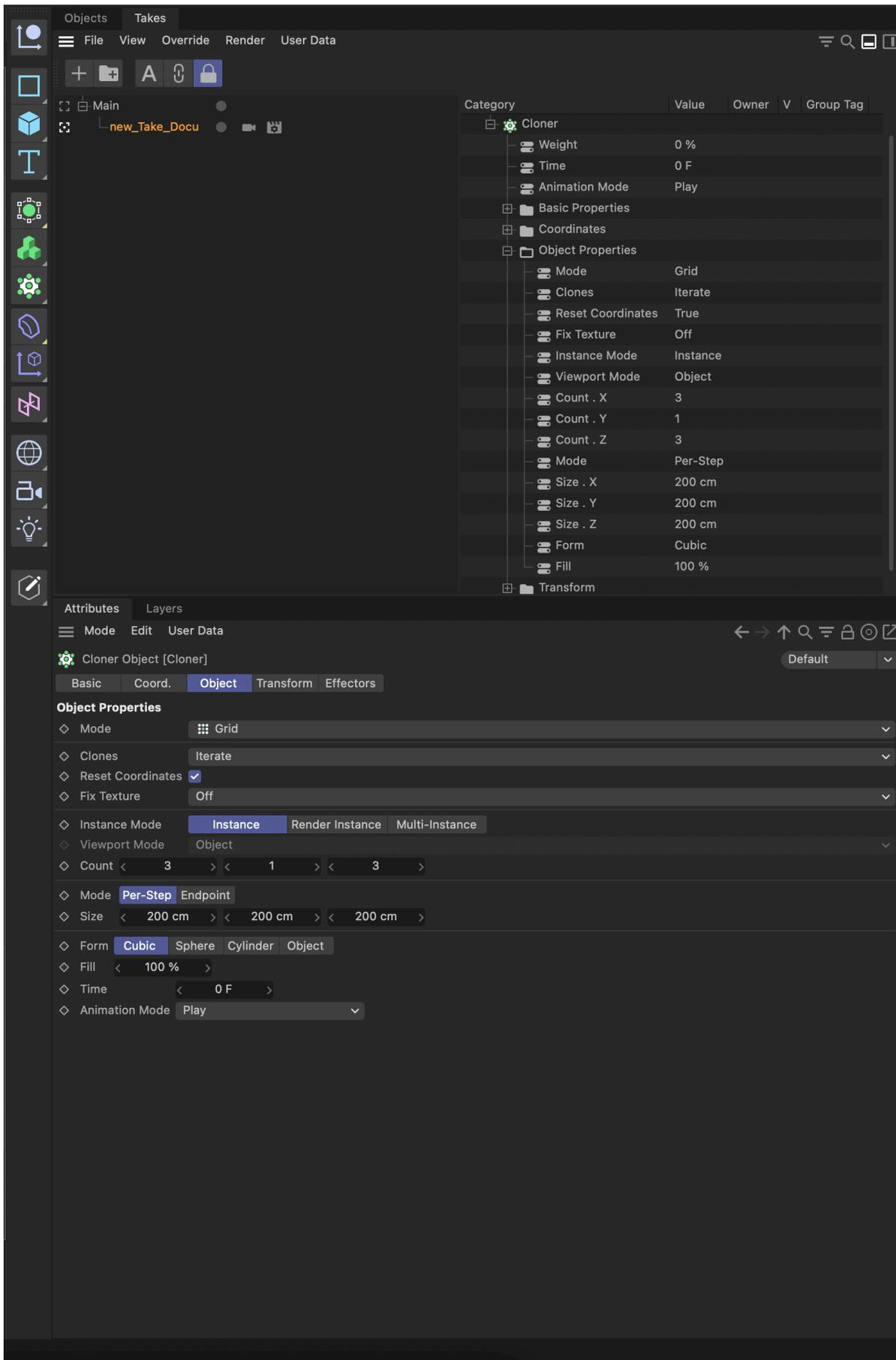


Right-click on the property you want to promote and click Override from the context menu. This will add the property to the selected take.



The user can also add whole elements to the to the CinewareAsset by a right click on the element (as an example if the user wants to add the whole cloner object and not just a single property)and then going to Take assignment > Clicking on the take that was created.





Currently supported override types are:

- Integer
- Dropdowns
- Checkboxes
- Text
- Floating point
- Vectors
- Colors

Save the file. Note that it is no longer necessary to save your project via the "Save for Cineware" command. This dramatically reduces the files sizes of shared .c4d files but slightly increases the time needed to import the scene.

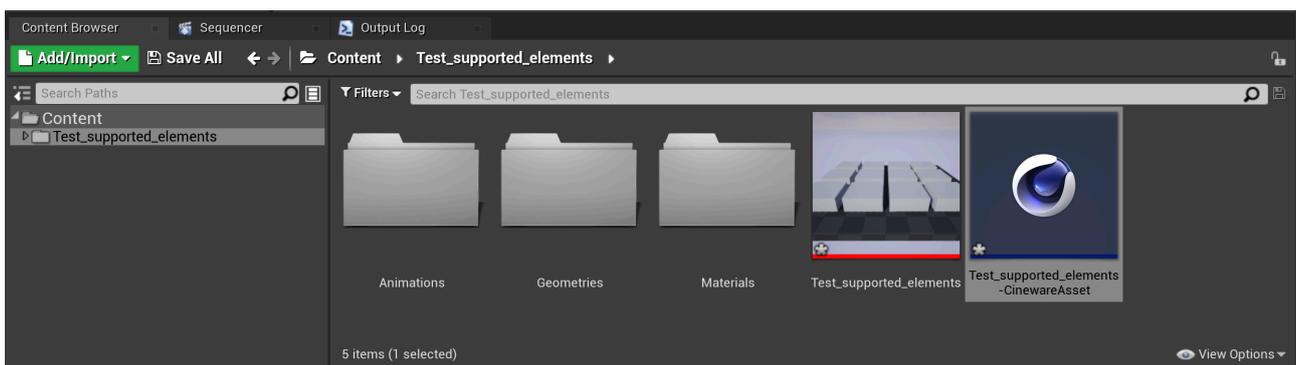
1.2.2 Datasmith Import

Open an Unreal Engine project with the plugins enabled. In the action menu bar, you should see an icon labeled *Datasmith*. Click on the button and browse for your saved .c4d file. Select the file and hit import. The import process will open to pop-up windows. Within the first one, select the import location within the game project browser. The second window enables you to set your preferred Datasmith import settings.

The initial import of the session will load up the cineware engine. This process spawns a progress bar and typically takes between 5 and 30 seconds. Note that for this version the progress bar is linked to the maximum timeout and not to the actual progress of starting up the engine. Should the progress bar reach 100 percentage points the process timed out and something went severely wrong. Send an email to the support email address should this be the case. If this is your first Cineware Engine startup on the machine you will be prompted to enter your Maxon account credentials. A Maxon account is free to create and mandatory for using the engine. After successfully signing in, the actual file import begins.

1.2.3 Cineware Properties Asset

An initial import of .c4d files will be imported with the default values of exposed properties as they are set within the Cinema 4D scene. Additionally, the import creates an asset of type *CinewareProperties* and stores it at the location of the imported scene. This asset is filled with the properties set within Cinema 4D. Future imports of the .c4d file will read the values within the asset and apply it to your scene before translating the file into Unreal Engine file formats.

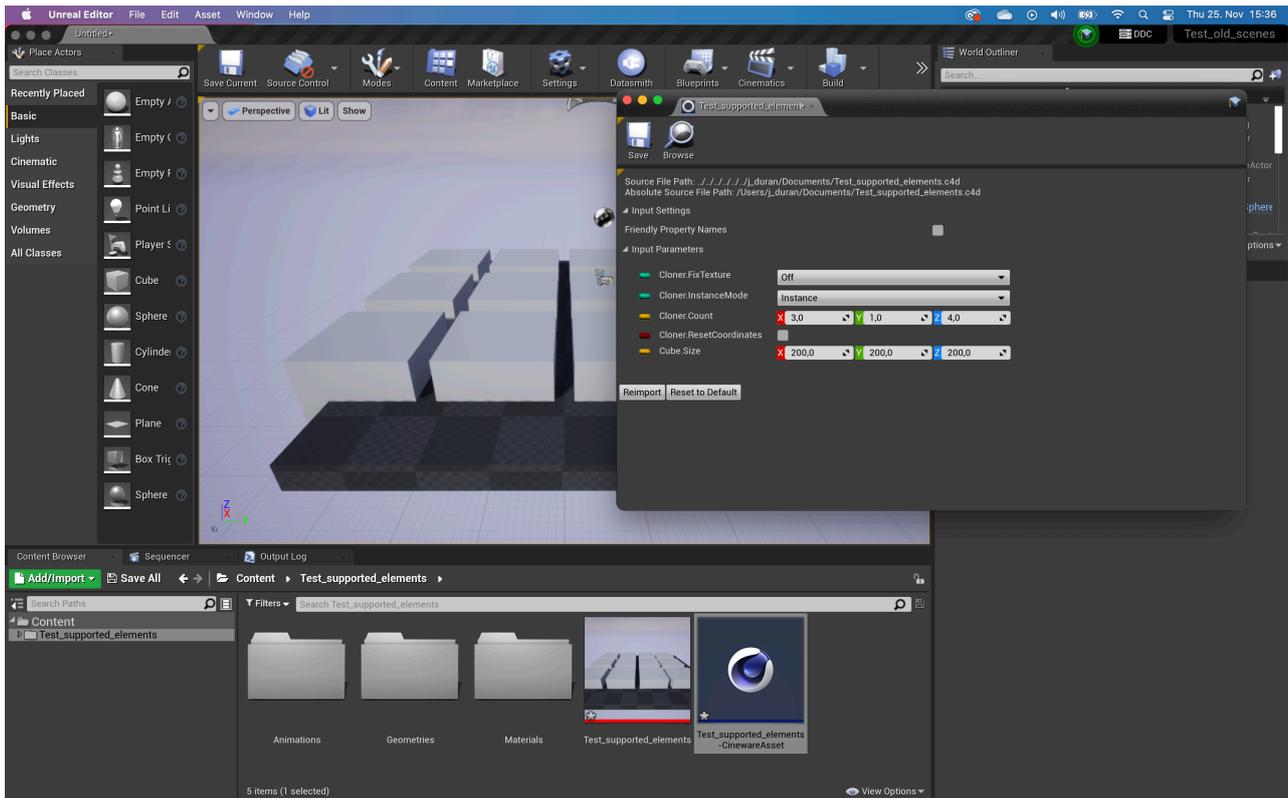


1.2.4 Edit Values

There are multiple ways to edit the stored properties stored within the asset.

1.2.4.1 Via UI

Double click the CinewareProperties asset to open up a Panel window displaying all the properties stored within the asset. The CinewareProperties asset is linked to the Datasmith asset via the original file path. Both absolute and relative paths are considered to make sure the project can be shared within a team. This means you can move the CinewareProperties asset anywhere within your project and rename it as you wish. For convenience the asset is created next to the imported Datasmith scene and named accordingly.

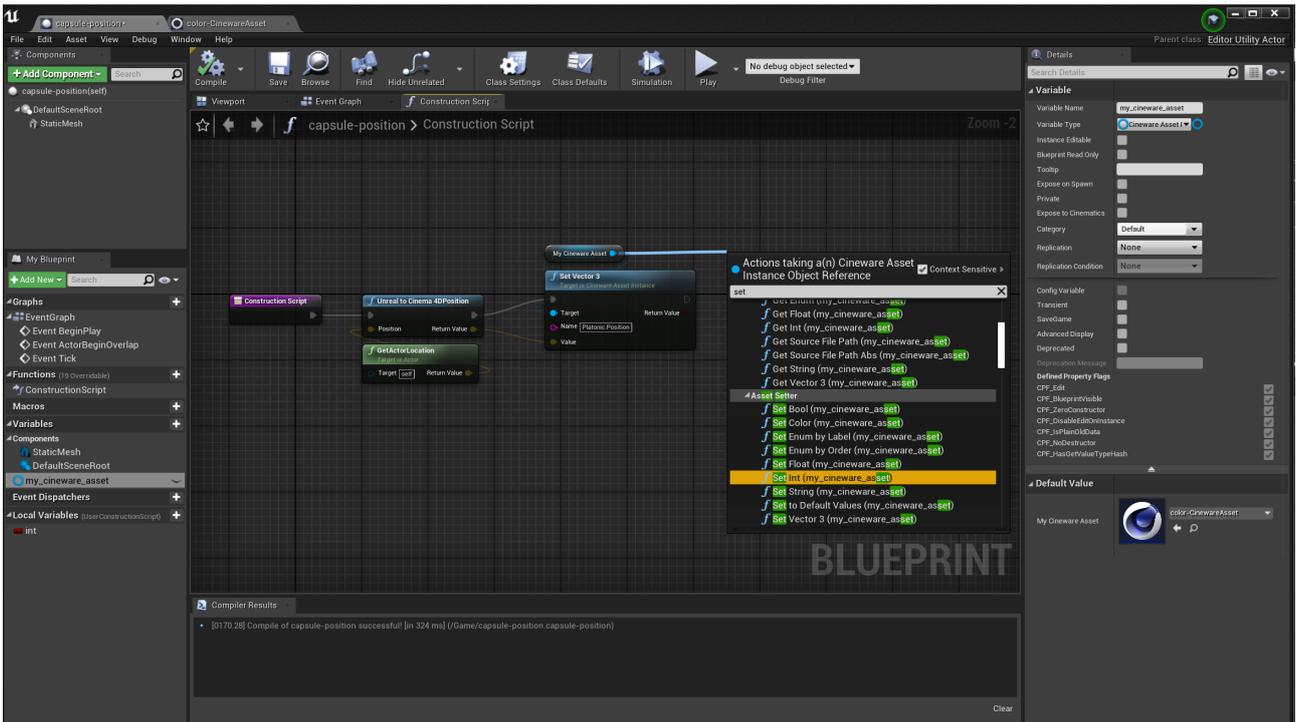


1.2.4.2 Via Blueprints

A CinewarePropertiesAsset can be references by Editor Blueprints. Since the assets are cooked when the project is packaged, no runtime blueprint is able to work with CinewareProperties assets. Cineware exposes several Blueprint Editor functions. It is possible to set data, read data or browse the given asset. This makes it possible to edit the values dynamically based on scene events.

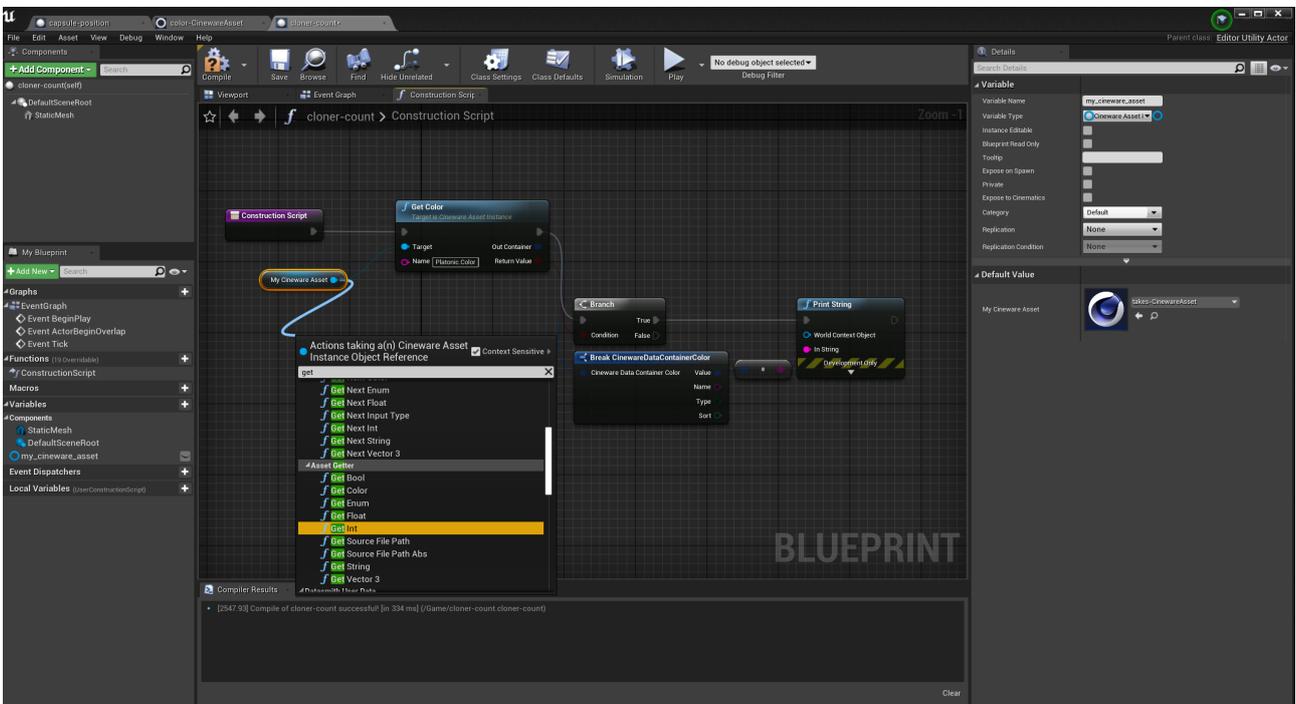
Setting properties by name

It is possible to use variables of type Cineware Asset for Editor Blueprint Actors and Editor Level Blueprints. There is a few Asset Setter functionalities for the Cineware Assets. Use the same key as is displayed in the Cineware details panel. The methods return a Boolean if setting the key was successful. In the example below you can see the user converting an actor position from Unreal space to Cinema space before applying the value.



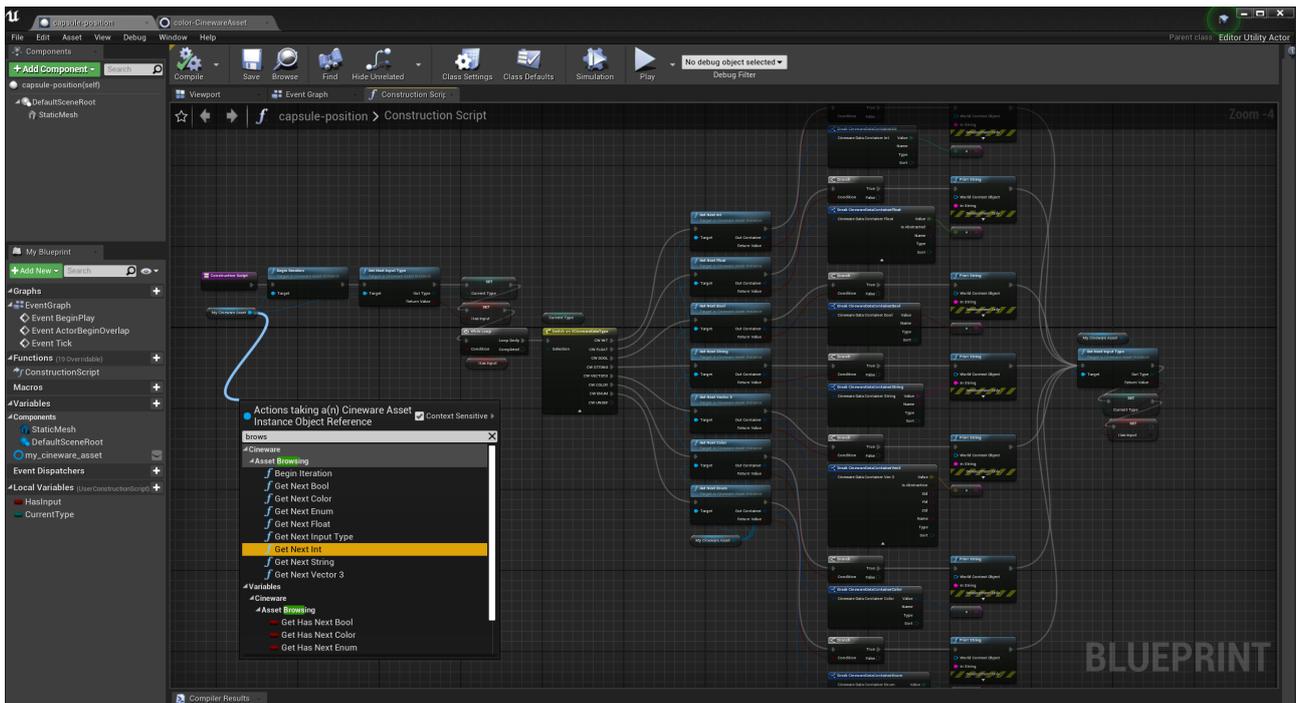
Reading properties by name

Cineware Assets have straight forward Getter methods which return a Boolean, stating if getting the value was successful and a struct containing all information depending on the type. Every struct contains the Key, Value, Type and sorted position in the Cinema 4D hierarchy. The container information varies depending on the out value.



Browsing for properties

There may be situations where you do not know the exact nature of the created asset. In these cases, you may want to browse through a given Cineware Asset and filter by name, value or type and apply some logic to it. Cineware Assets expose browsing properties for this to work. Note that in the example below the first call is the Begin Iteration, this resets all iterators of the asset. You can then browse through it by type. Via a while loop and a switch, you may then call the appropriate Get Next functionality. It is recommended to check the Boolean return value before accessing the returned container.



Things to consider:

- The axis' vary between Unreal Engine and Cinema 4D. When setting Unreal Engine positions via blueprints make sure to execute the UnrealToCinema4DPosition or Cinema4DToUnrealPosition blueprint method, before setting them in a CinewareProperties asset.
- When setting properties by their name, use the name as it is printed when having Friendly Property Names **disabled** in the CinewareProperties panel. Friendly keys are currently not supported.
- Make sure to check the returned Boolean value if it is safe to access the container
- CinewareProperty Blueprintfunctions can only be called in Editor Blueprints. If you do not have the option to call them, make sure you are running in an Editor environment. This is visualized with a yellow and black bar in the Blueprint layout.

1.2.4.3 Via C++

For example, how to browse through assets, have a look within SCinewareAssetInstanceEditor.cpp and the UI construction. Naturally direct setters and getters are also available within the C++ environment.

1.2.5 Reimport

The cinema scene needs to be reimported for the changed properties to be applied. You can reimport the asset by right clicking the Datasmith asset and selecting Reimport Asset from the context menu, or by clicking the Reimport button in the Cineware Editor. Automatic reimport is currently in development.

1.3 Feedback

Should you encounter any issues please visit the following website: "<https://support.maxon.net>³". Here you can contact our support team.

1.4 Known Issues

- CinewareProperty asset is not visible after import, editor needs redraw by reopening the folder.
- UI layout inconsistent
- Some inconsistencies when importing materials
- Animated material properties

1.5 Feature list that is not supported in this Release

- Deformers
- Rigs
- Simulation tag Animations
- Animations in general that are not PSR
- PLA animations
- Skin Deformer
- Changing the number of generated objects
- Animated deformation (A static mesh must be static)
- Node materials
- Uber materials

³ <https://support.maxon.net/>