

# Unreal/ CWEngine - Package Documentation

- [Unreal / Cineware Engine](#)
  - [Installation](#)
    - [Prerequisites](#)
    - [Plugin Installation](#)
      - [CinewareForUnreal \(Cineware by Maxon\)](#)
    - [Verify if the Plugin is installed correctly](#)
    - [Possible Errors](#)
  - [How to use it](#)
    - [Cinema 4D Takes](#)
    - [Datasmith Import](#)
    - [Cineware Properties Asset](#)
    - [Edit Values](#)
      - [Via UI](#)
      - [Via Blueprints](#)
        - [Setting properties by name](#)
        - [Reading properties by name](#)
        - [Browsing for properties](#)
      - [Via C++](#)
    - [Reimport](#)
  - [Feedback](#)

## 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.

### Installation

### 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.

This version of the plugin is compatible with any version of 5.0.XX so it will work with your installation out of the box.

On OSX its also required to download the latest version of XCODE so that Unreal Engine can compile. ([https:// apps.apple.com/us/app/xcode/id497799835?mt=12](https://apps.apple.com/us/app/xcode/id497799835?mt=12))

For this version of the plugins to work it is also required to have Cinema 4D R25.120 or newer R25 installed on your system in its default path.

### Plugin Installation

CinewareForUnreal (Cineware by Maxon)

This package contains the new and improved Datasmith importer and the CinewareAsset (Property editor)

This plugin will allow you to import a Cinema 4D file (.c4d) as well as, store exposed properties into an asset and will also allow you 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.

On Windows navigate to your Unreal Engine installation, typically C:/Program Files/Epic Games/UE\_5.0. (Windows)

On Mac navigate to your Unreal Engine installation, typically Macintosh HD/Users/Shared/Epic Games/UE\_5.0. (Mac)

This path will be referred to as \$UE5.0\_DIR within this document.

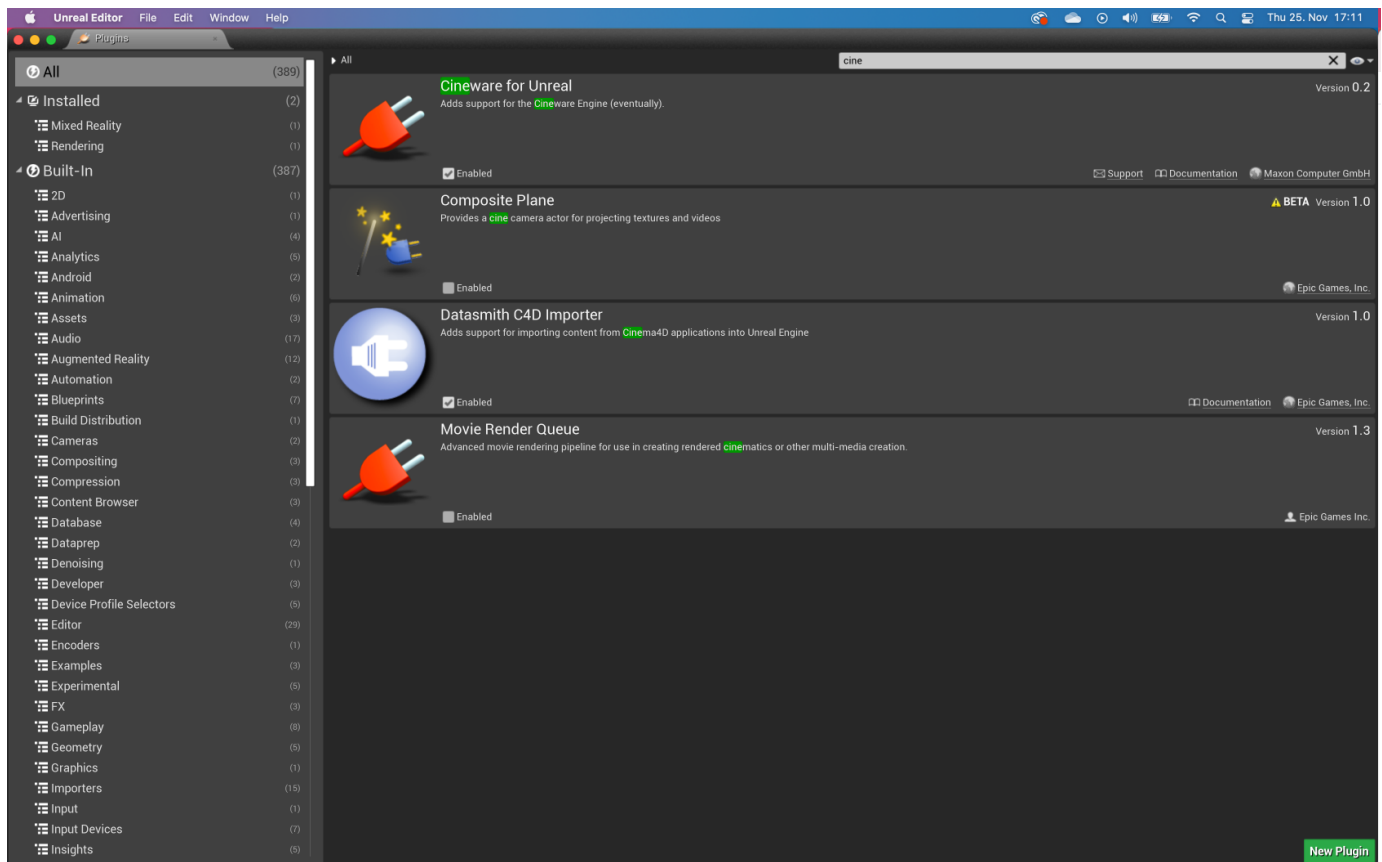
Typically, the path for shared plugins is \$UE5.0\_DIR/Engine/Plugins/MarketPlace/. Should this directory not yet exist, please create it. This directory is also where all plugins purchased and installed from the Unreal Engine Marketplace end up.

The path \$UE5.0\_DIR/Engine/Plugins/Marketplace/CinewareAsset/ should now contain the directories Binaries, Content, Intermediate, Source and a .uplugin file.

## Verify if the Plugin is installed correctly

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 Unreal 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. Afterwards, navigate the menu bar to Edit/Plugin. 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.



## Possible Errors

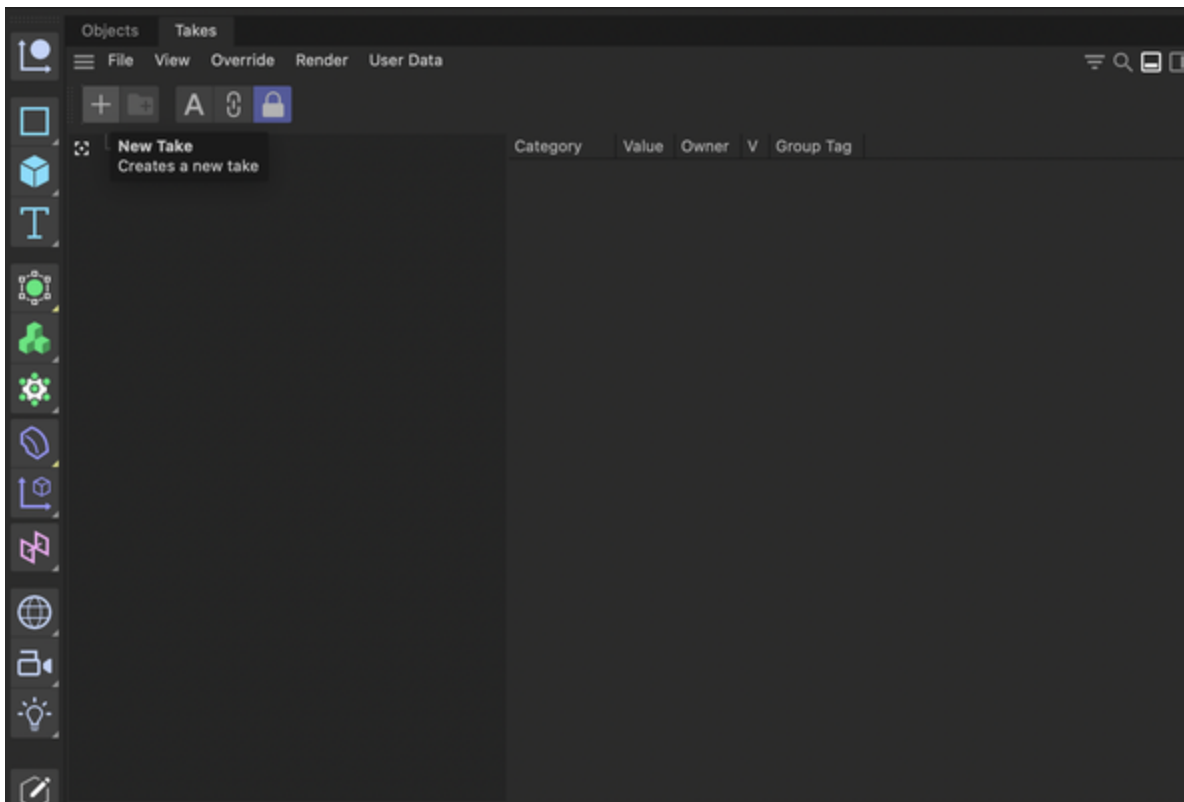
Should you encounter any issues please visit the following website: " <https://support.maxon.net1> ".

## How to use it

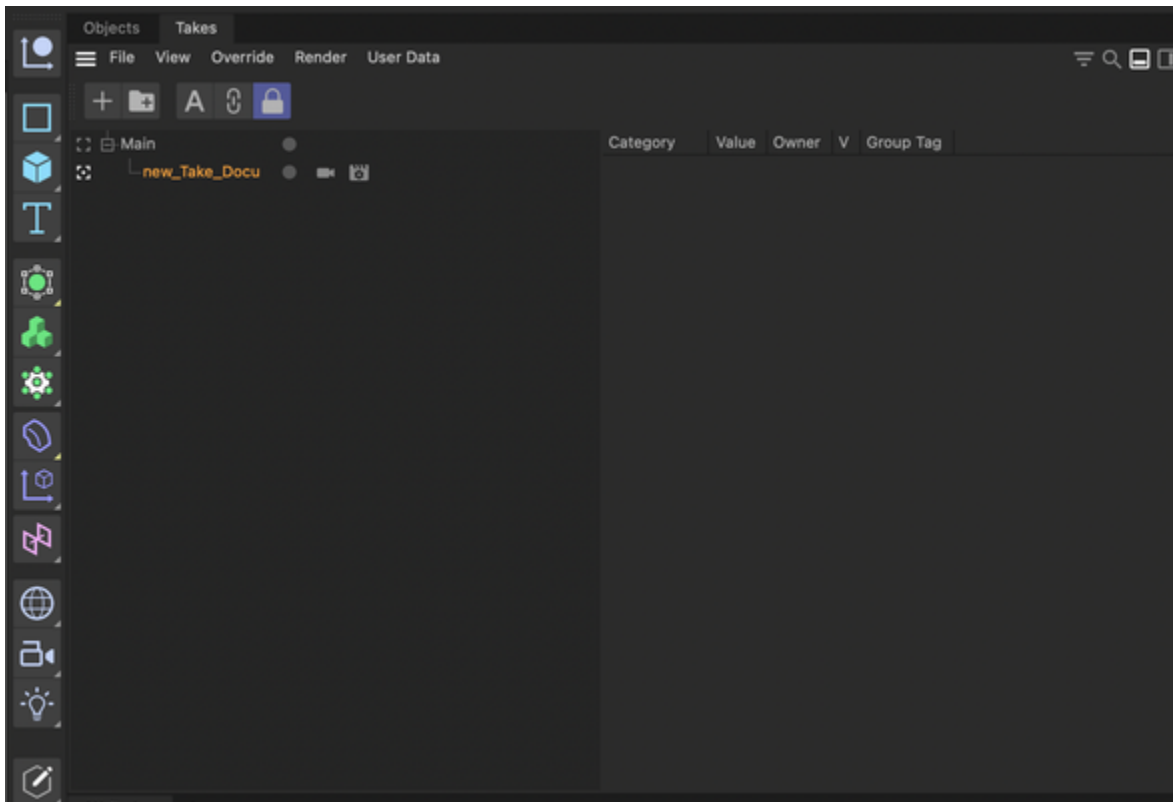
### Cinema 4D Takes

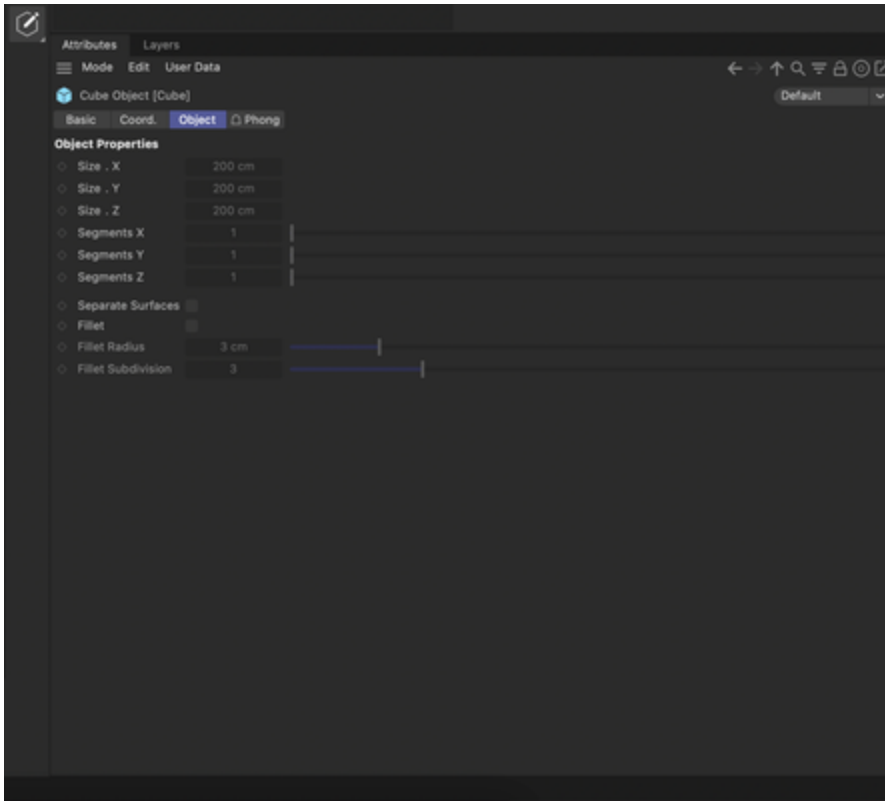
Scene properties can be exposed to Unreal Engine via Cinema 4D's take system. A video tutorial can be found here [by GreyscaleGorilla on Youtube](#).

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

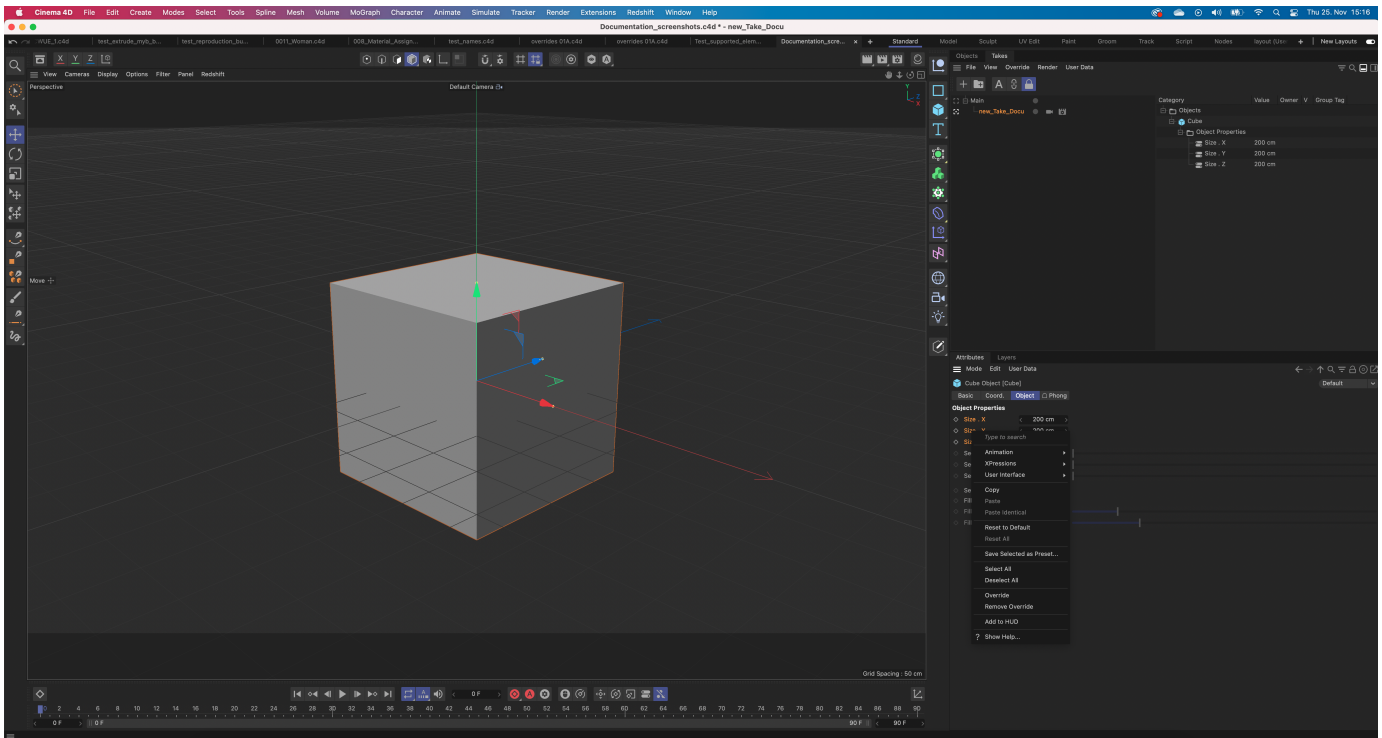


Switch to the state of the take by left-clicking on it. All properties on scene objects should now be greyed out.

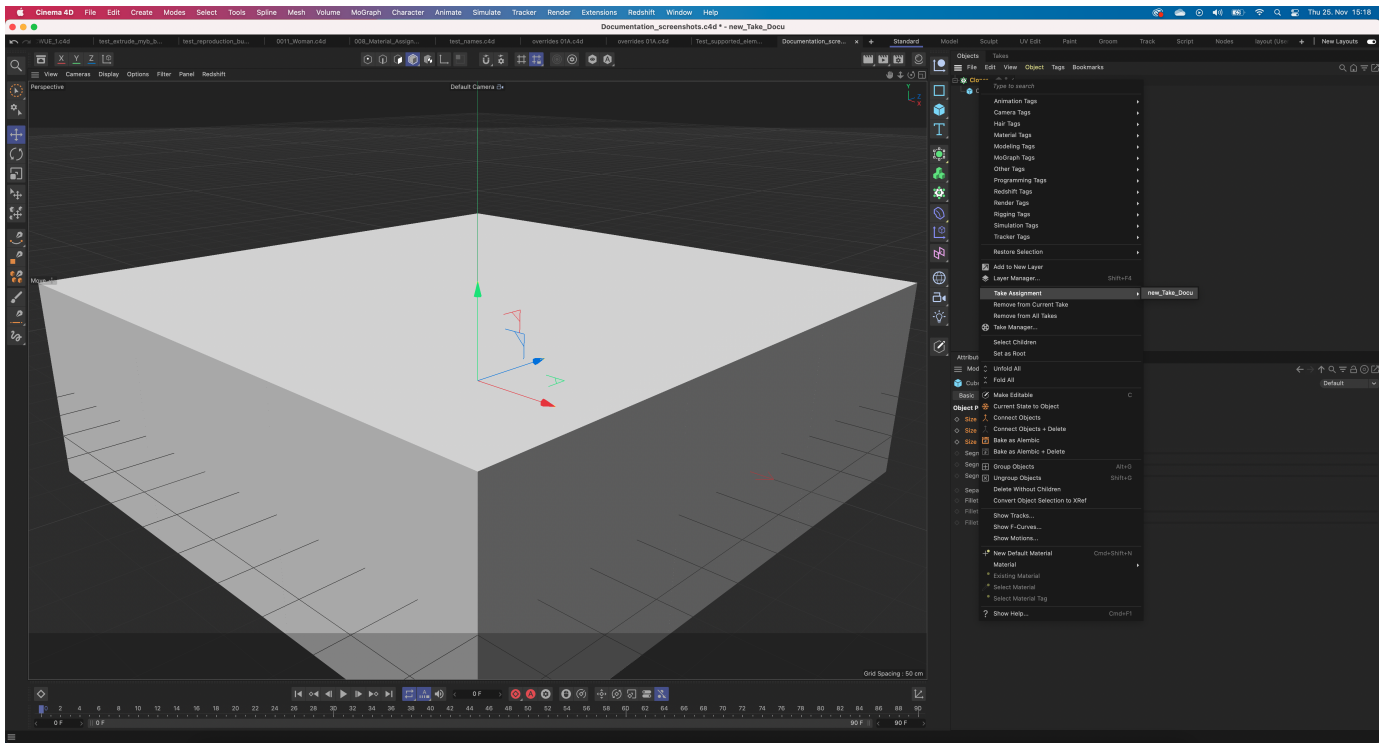


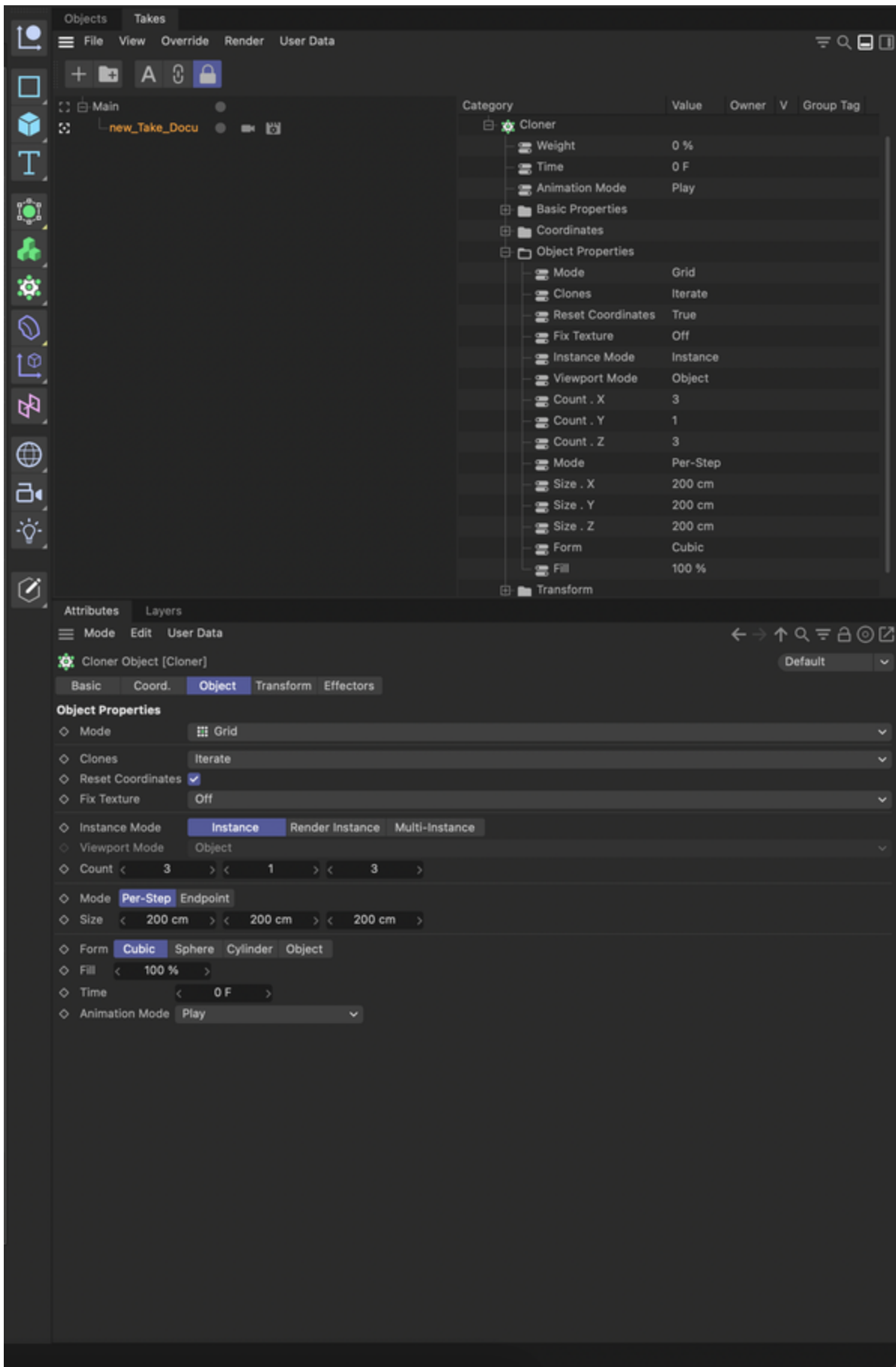


Right-click on the property you want to promote and click Override from the context menu. This will add the property to the selected take.



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.





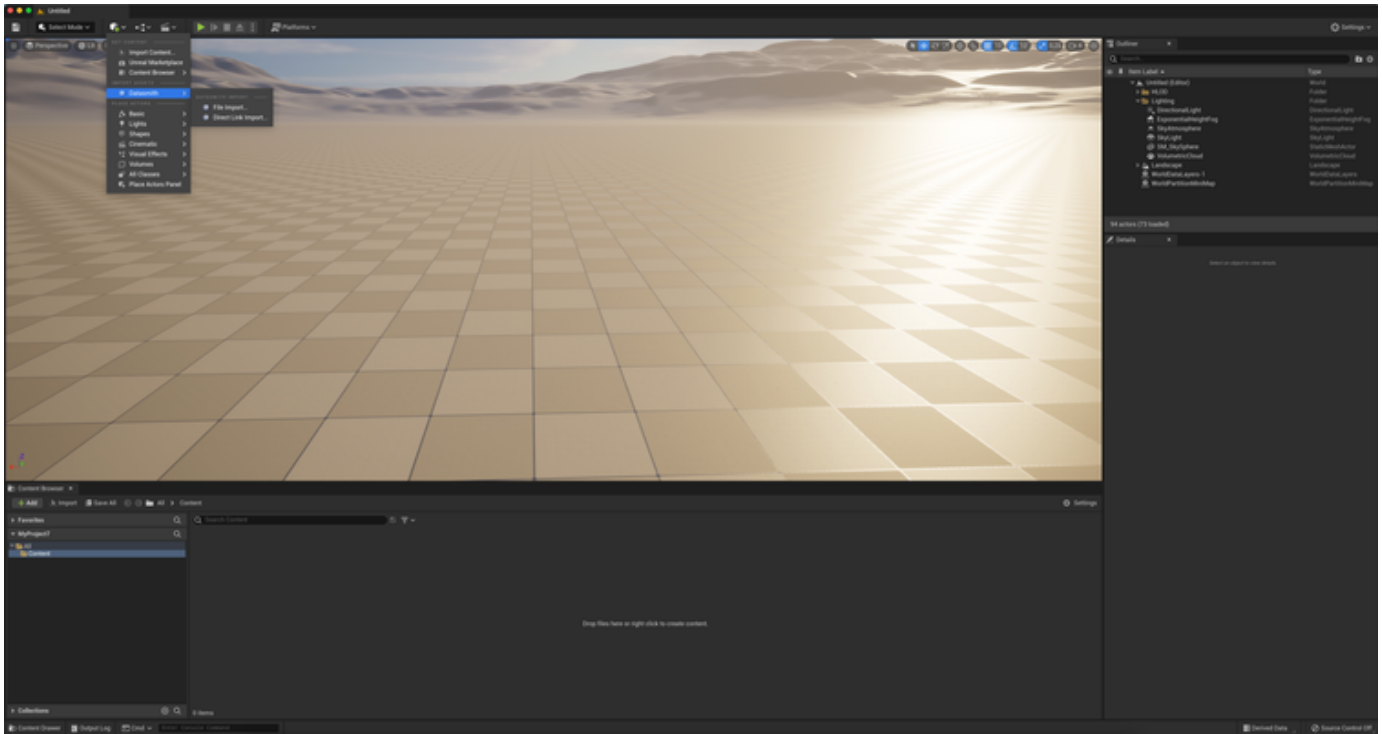
Currently supported override types are:

- 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 file sizes of shared .c4d files but slightly increases the time needed to import the scene.

## Datasmith Import

Open an Unreal Engine project with the plugins enabled. In the action menu bar, you should see a drop-down menu with a box as the icon, click on it and navigate to the section called import assets, there you will find the Datasmith importer.

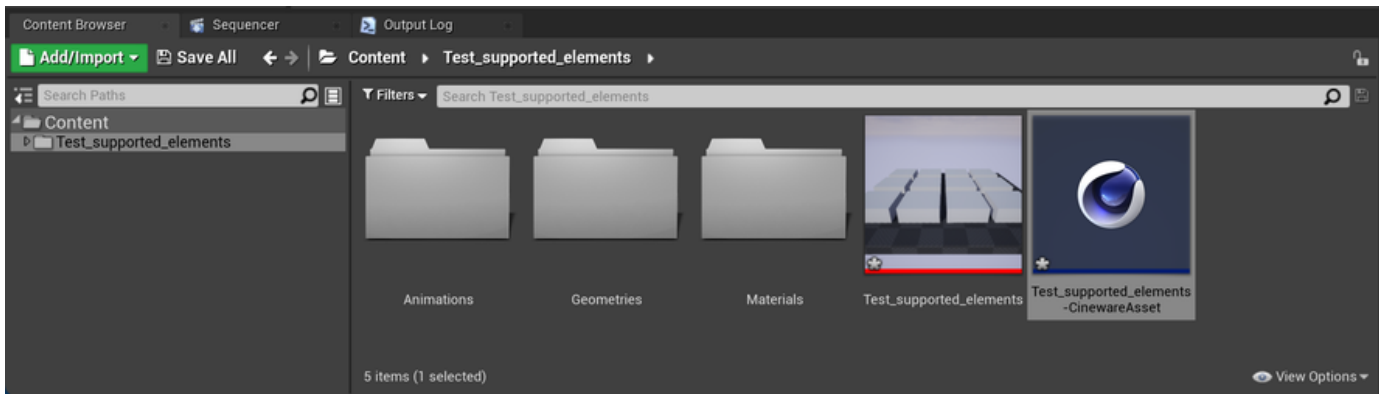


Click on the button and browse for your saved .c4d file. Select the file and hit import. The import process will open to pop-up windows. Within the first one, select the import location within the game project browser. The second window enables you to set your preferred Datasmith import settings.

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.

## 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 CinewareAsset 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.

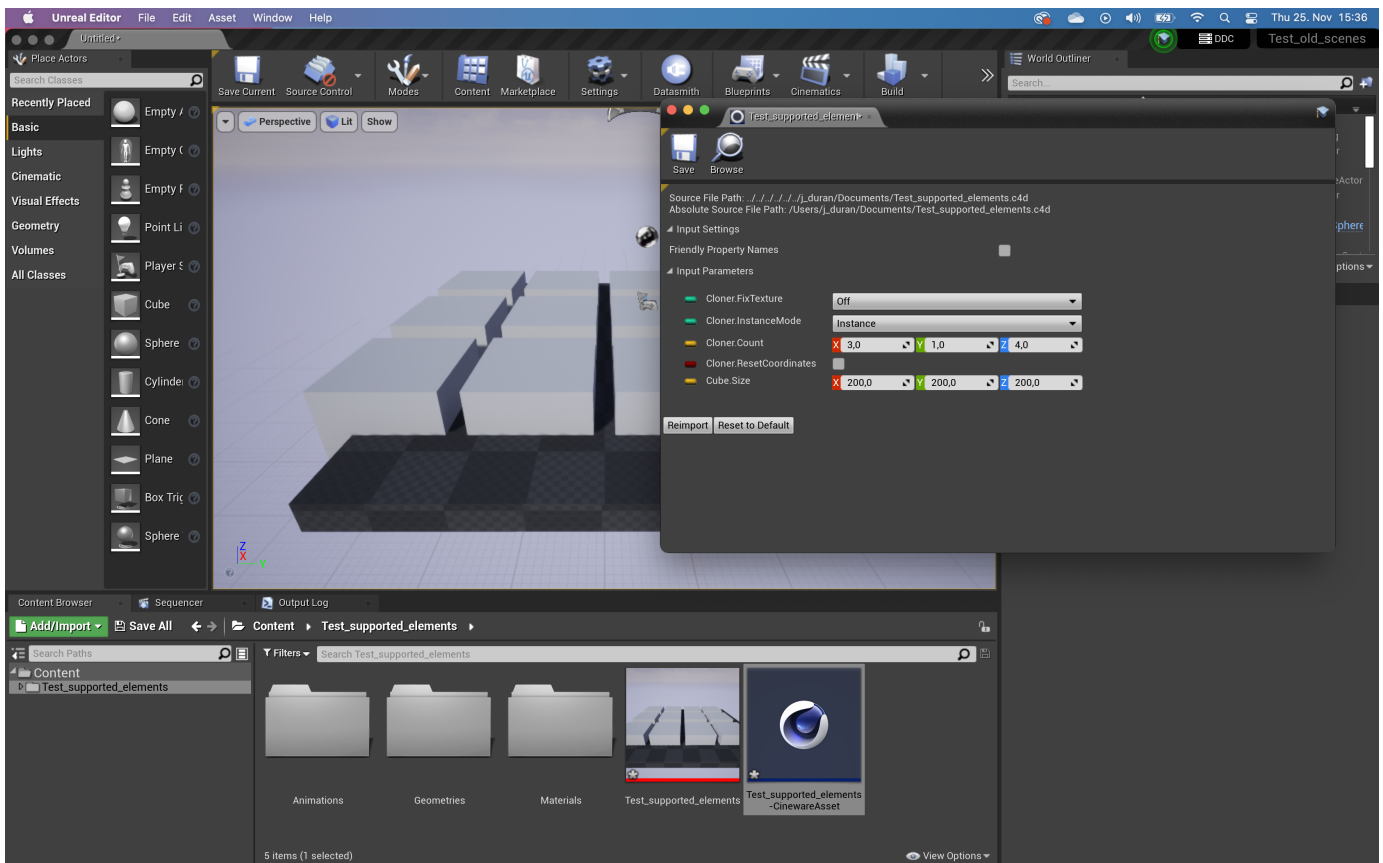


## Edit Values

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

Via UI

Double click the CinewareAsset asset to open up a Panel window displaying all the properties stored within the asset. The CinewareAsset 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 CinewareAsset 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.

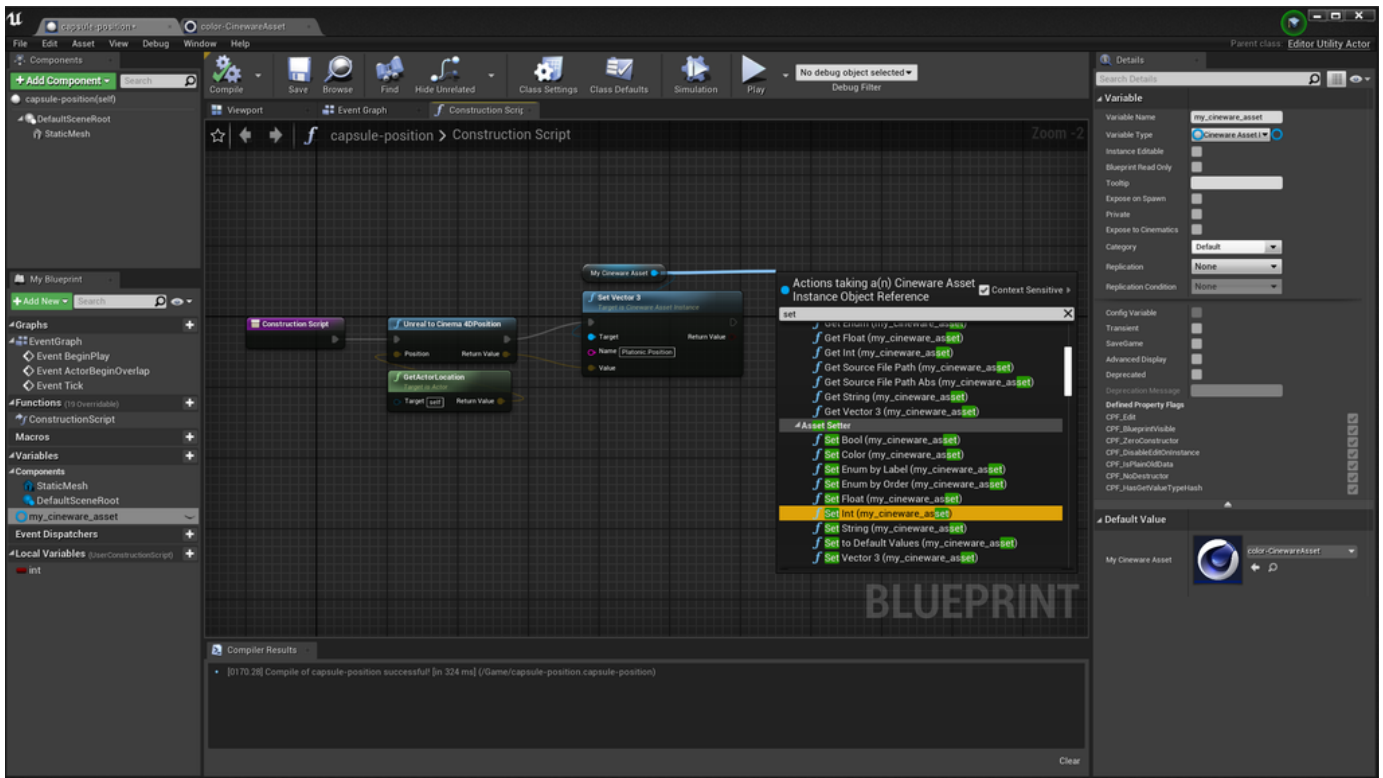


Via Blueprints

A CinewarePropertiesAsset can be references by Editor Blueprints. Since the assets are cooked when the project is packaged, no runtime Blueprints is able to work with the CinewareAsset. 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.

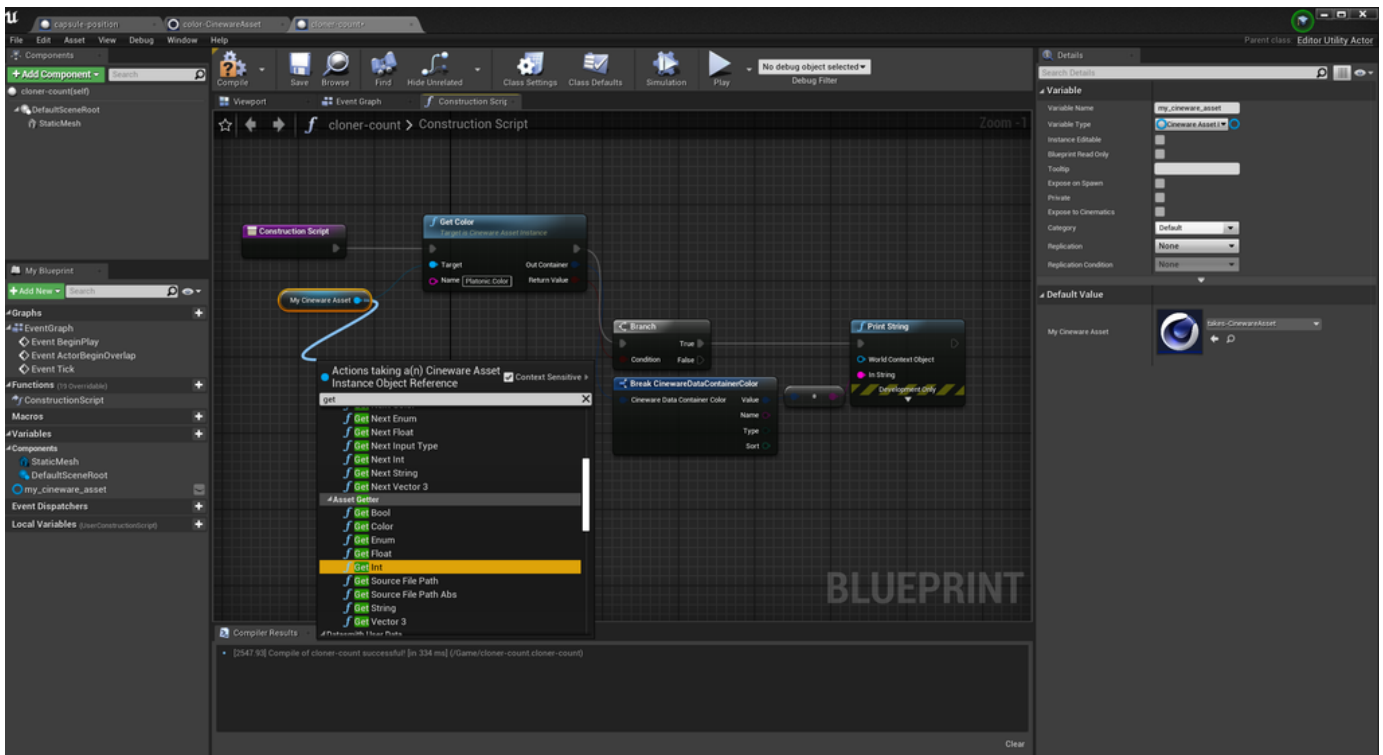
## Setting properties by name

It is possible to use variables of type Cineware Asset for Editor Blueprint Actors and Editor Level Blueprints. There are a few Setter methods for the CinewareAssets. Use the same key as is displayed in the Cineware details panel. The methods return True if setting a value 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.



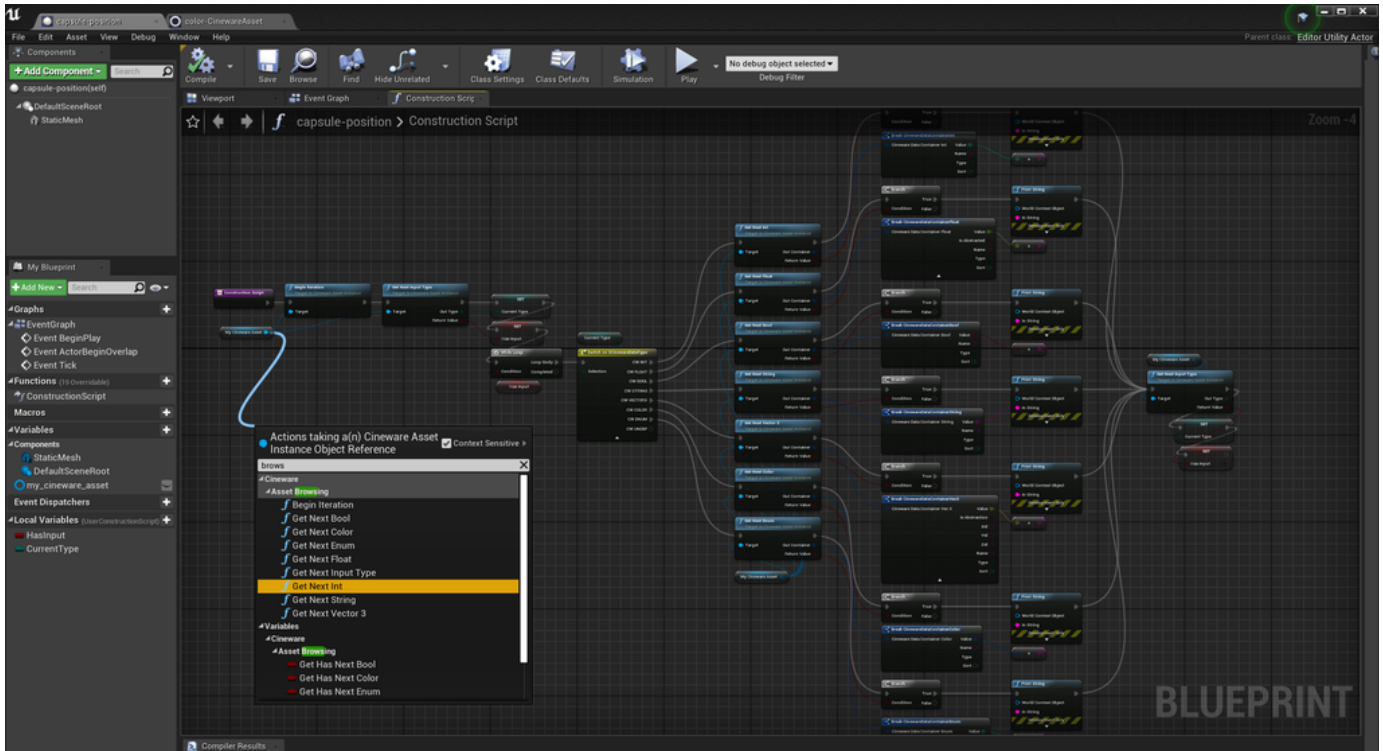
### Reading properties by name

Cineware Assets have straightforward 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 coordinate systems differ 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` Blueprint functions 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.

Via C++

To see an example of 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.

## Reimport

For the changes of the `CinewareAsset` to be applied, press the 'Reimport' button in the `CinewareAsset` Editor. You can reimport the scene as it was saved in Cinema by right-clicking the `Datasmith` asset and clicking 'Reimport'. Objects in invisible layers or objects with an annotation tag "exclude" will not be imported.

## Feedback

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