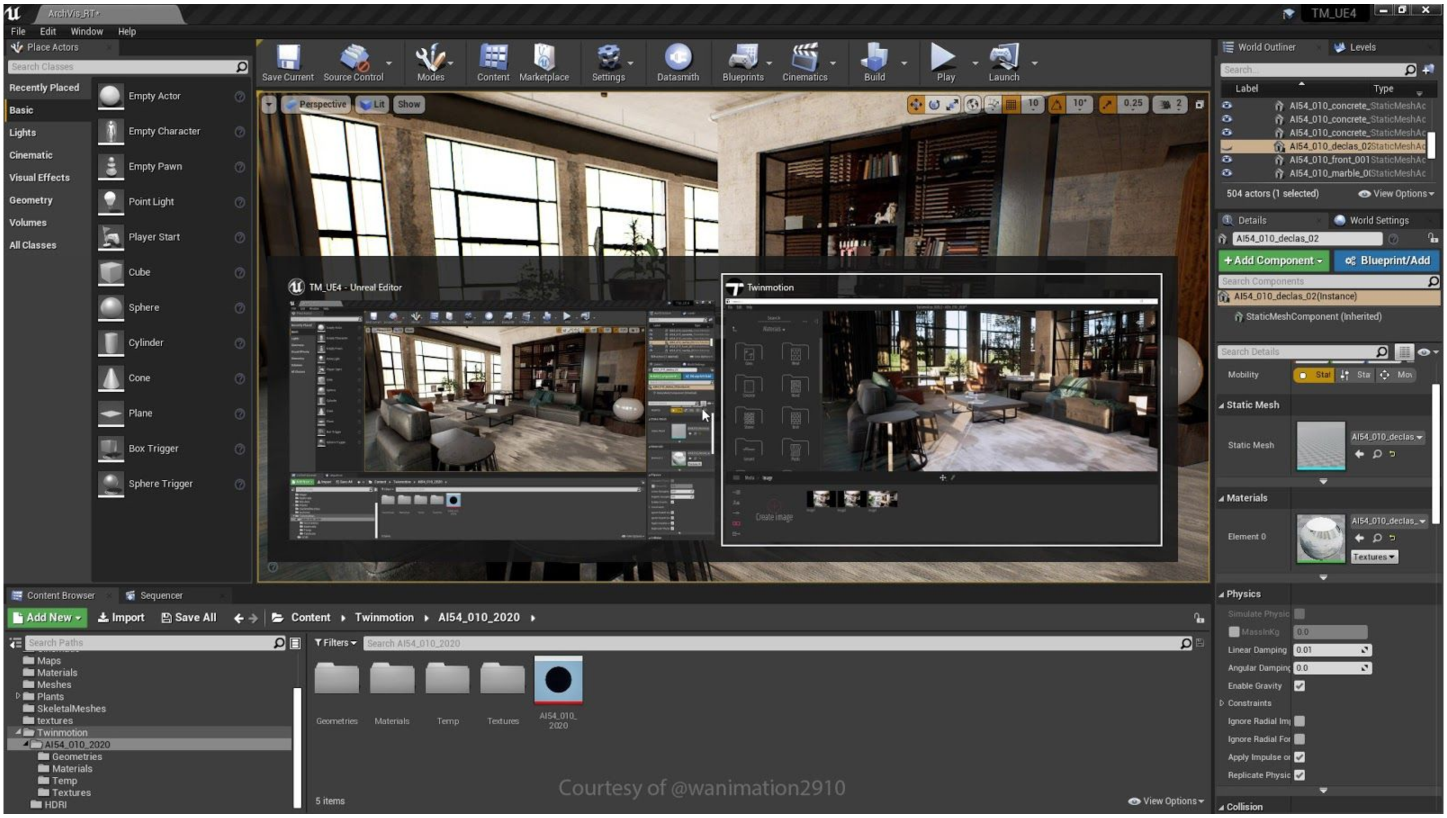
UNREAL ENGINE

Twinmotion Importer for Unreal Engine (Public Beta 1)



Courtesy of @wanimation2910

Document Version	
2020/26/11	Unreal Engine Marketplace Public Beta 1

Overview

For who and why?

Stated simply, you should be able to **start a project in Twinmotion and finish it in Unreal Engine**.
Here are a couple quotes from users illustrating this:

> *“As a designer I found Twinmotion a very powerful and fast platform to add materials and other elements to my 3D models in comparison to traditional render engines. Now I want to create an interactive environment in Unreal Engine but it is impossible to export "fbx" or "obj" files from TM or directly import tm files into UE.”*
>
> *“We believe we can enhance the clients experience through the use of Unreal Engine. Specifically by starting with early visualization efforts in Twinmotion and transferring that work to Unreal for further development.”*

How are we doing it?

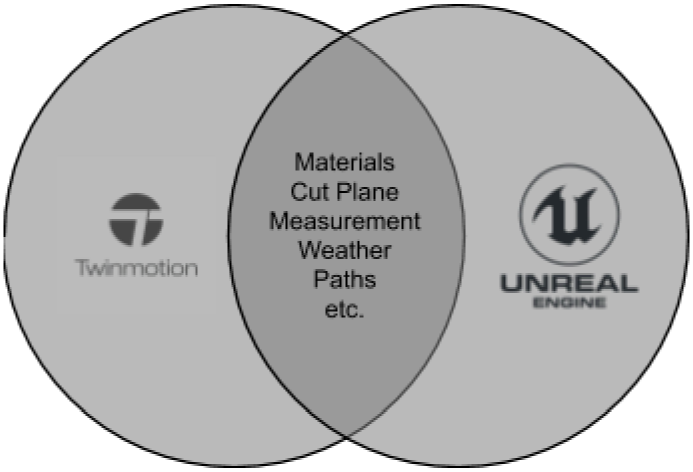
Our approach is to **leverage the existing Datasmith and Dataprep workflows** by adding the ability to import Twinmotion files to Unreal Engine exactly like we import CAD models.

We expect the same functionality found in Datasmith such as import, re-import, local overrides, etc. - (if you are unfamiliar with those workflows, check out the docs here).

Our vision

That said, **our vision is broader** than simply importing data from Twinmotion:

We also want to empower Unreal Engine users with TM-like functionality (cut planes, measurement tools, simple animation tools etc.) that can be used directly in Unreal, without necessarily importing TM files to begin with:



Executing on this entire vision is going to take time, but we want you to understand where we are headed.

In the initial implementation of this experimental plug-in, there are major limitations to what you can do with it, but these will be addressed over time.

Roughly speaking (and subject to change) we see the following high-level steps to executing our vision:

1. Import Twinmotion user-imported data and any Twinmotion-applied materials (no entourage, landscape, animated actions or effects)
2. Import Twinmotion entourage and landscape (no animated actions or effects)
3. Import Twinmotion projects and provide tools in UE4 for animating actions and effects similar to what you find in Twinmotion.

Other limitations

Can we bring Unreal Engine assets into Twinmotion?

Our vision focuses on creating a tool to carry data from Twinmotion to the Unreal Engine. Data exchange from Unreal Editor to Twinmotion will not be discussed here. It is something we're considering, but the priority is to get the Twinmotion data into Unreal Engine first.

Can we bring Twinmotion scenes into an Unreal package without having to go into the Unreal Editor?

At this time, the ability to import Twinmotion files in Unreal is possible in the editor only.

We are however looking at how we can make our Datasmith importers available at runtime. Supporting *.tm import/re-import in a custom packaged game should be possible to accomplish in the future but we don't have a date in mind for delivering the functionality.

Related efforts

Separately, we are working to unify Datasmith in both Unreal Engine and Twinmotion. This means that Twinmotion will switch over to a Datasmith-based import capability eventually.

In Twinmotion 2020.2, the first step of this is demonstrated as Twinmotion can now import uDatasmith files directly. Eventually, we will remove the existing Twinmotion importers and it will become 100% Datasmith.

This also means we will unify the various plug-ins for Twinmotion and Datasmith. There will be one plug-in per host app that will let you connect with either Twinmotion or Unreal Engine.

How to give us feedback?

Discussions:

General discussions and questions about this feature will be hosted on the Unreal Engine discussion forums here: https://forums.unrealengine.com/unreal-engine/unreal-studio

Bug reporting:

If you have a Twinmotion file that isn't importing correctly (within the limitations described below), don't hesitate to share it with us. This form will let you share your files with our R&D team in a secure way: https://forms.gle/WKw7CwJk2iq51Y5Y8

Survey:

After you've had a chance to try a few projects, help us understand how well this feature works for you and define our next priorities by filling this short anonymous survey:
https://forms.gle/hSBUNnR4zJCu9qnP6

Installation

Software prerequisites

The following software must be installed to test the workflow between Twinmotion and UE:

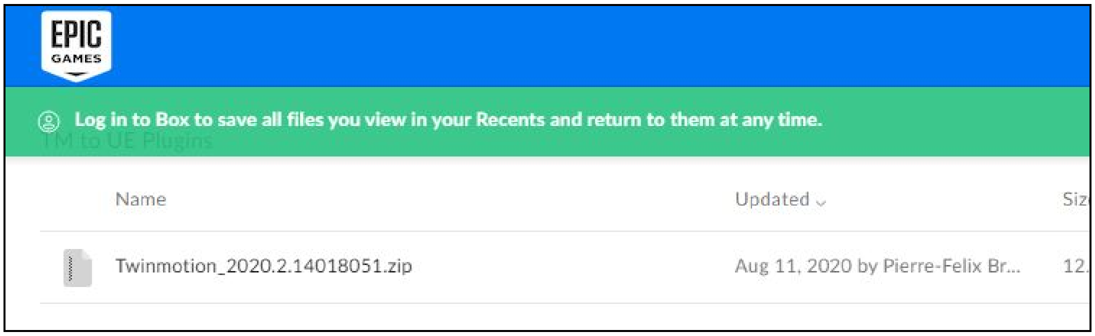
- Unreal Engine **4.25.4 or more recent**
- Twinmotion **2020.2.1 and up**
  (Files saved from previous versions of Twinmotion will not load in Unreal Properly)

Plugin installation

Twinmotion → Unreal Engine Plugins is made possible with the addition of 3 plugins to the Unreal Engine. Here is how to install them:

1. **Download** the most recent zip archive available from our box drive:

   https://epicgames.ent.box.com/s/3hx8eua9439yru612jsl3ancuhuf0nh0



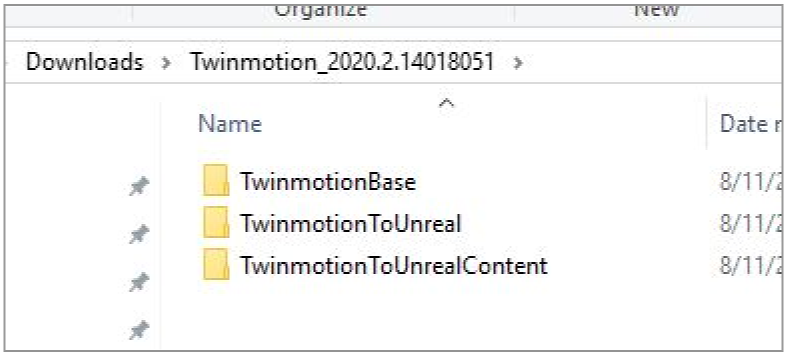
Log in to Box to save all files you view in your Recents and return to them at any time.

TM to UE Plugins

Name	Updated ⌄	Siz
Twinmotion_2020.2.14018051.zip	Aug 11, 2020 by Pierre-Felix Br...	12.

(The filename of the zip archive may vary from this illustration)

2. **Unzip** it:

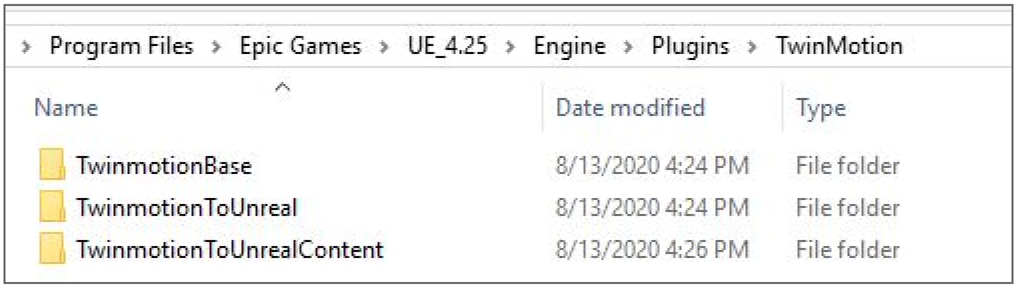


Downloads › Twinmotion_2020.2.14018051 ›

Name	Date r
TwinmotionBase	8/11/2
TwinmotionToUnreal	8/11/2
TwinmotionToUnrealContent	8/11/2

3. **Install the plugins to the Engine (or Project)**

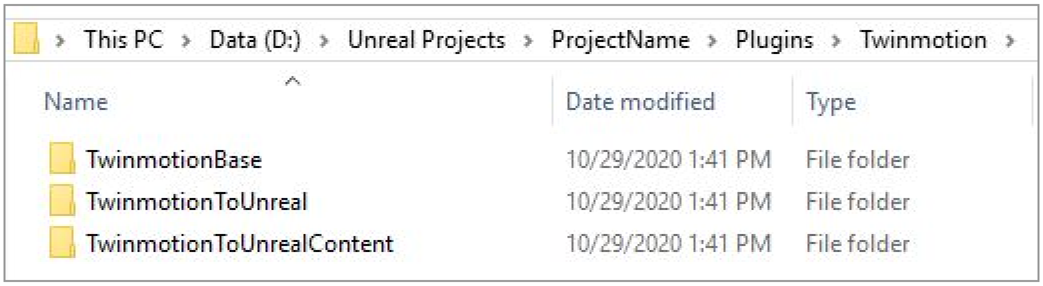
   - **Option 1: Copy the content in Unreal 4.25 (or 4.26)** under *[program files]\Epic Games\UE_4.25\Engine\Plugins\TwinMotion\*



› Program Files › Epic Games › UE_4.25 › Engine › Plugins › TwinMotion

Name	Date modified	Type
TwinmotionBase	8/13/2020 4:24 PM	File folder
TwinmotionToUnreal	8/13/2020 4:24 PM	File folder
TwinmotionToUnrealContent	8/13/2020 4:26 PM	File folder

   - **Option 2: Copy the content in the Plugins folder of your project** under *[projectname]\Plugins\Twinmotion*



› This PC › Data (D:) › Unreal Projects › ProjectName › Plugins › Twinmotion ›

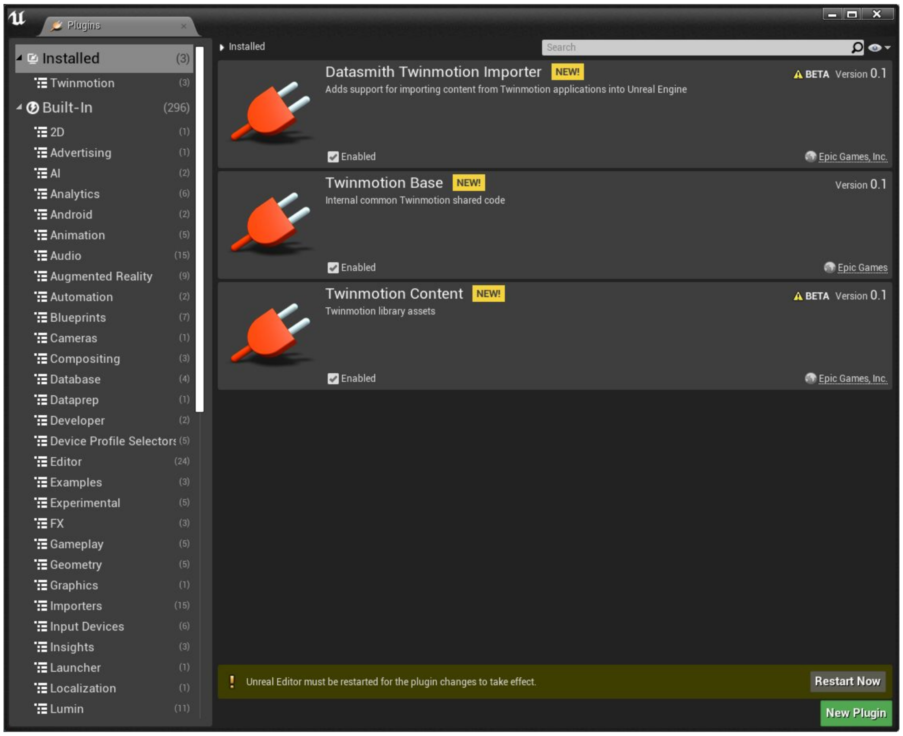
Name	Date modified	Type
TwinmotionBase	10/29/2020 1:41 PM	File folder
TwinmotionToUnreal	10/29/2020 1:41 PM	File folder
TwinmotionToUnrealContent	10/29/2020 1:41 PM	File folder

Note: Remember to completely remove what you copied before updating to a new build to avoid mixing versions of files and assets.

Set Up Your Unreal Project

Activate plugins

After launching a new Unreal project, enable the three (3) plugins from the plugins tab and restart your project:



Datasmith Twinmotion Importer NEW! — BETA Version 0.1
Adds support for importing content from Twinmotion applications into Unreal Engine
☑ Enabled — Epic Games, Inc.

Twinmotion Base NEW! — Version 0.1
Internal common Twinmotion shared code
☑ Enabled — Epic Games

Twinmotion Content NEW! — BETA Version 0.1
Twinmotion library assets
☑ Enabled — Epic Games, Inc.

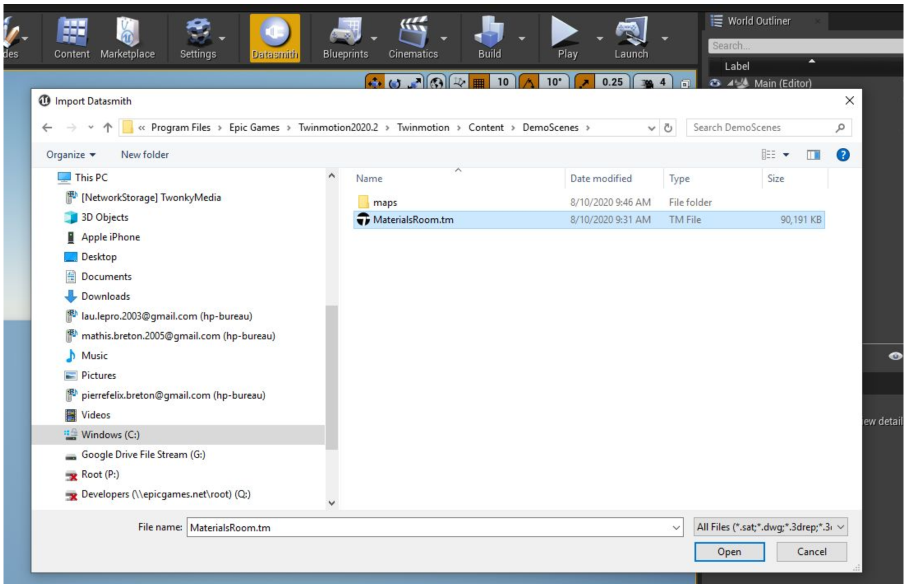
! Unreal Editor must be restarted for the plugin changes to take effect.

Datasmith Twinmotion Importer Plugin	Plugin handling file import through the existing Datasmith and Dataprep workflows.
Twinmotion Base Plugin	Code shared between Twinmotion and Unreal - you don’t need to look for functionality here - that's something we need to do to separate code from content - its purely internal mechanics.
Twinmotion Content Plugin	Content library reflecting the content shipping with Twinmotion including Master Materials, Assets etc. Feel free to use this content for your own project even if you don’t import Twinmotion files.

How to import TM files

Datasmith workflow:

The workflow, based on Datasmith, is identical to importing *.udatasmith files or CAD files. You will find an additional import format (*.TM) added to the list:



Import Datasmith

« Program Files › Epic Games › Twinmotion2020.2 › Twinmotion › Content › DemoScenes ›

Search DemoScenes

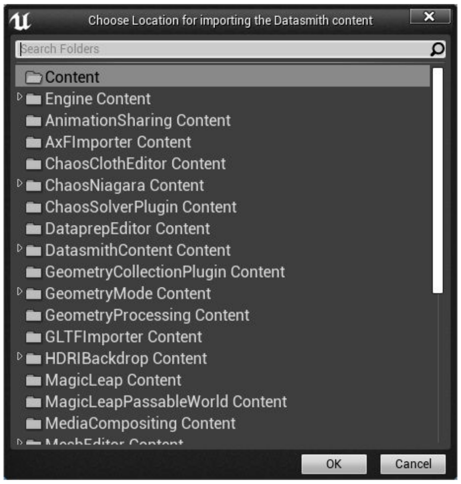
Organize
New folder
This PC
[NetworkStorage] TwonkyMedia
3D Objects
Apple iPhone
Desktop
Documents
Downloads
lau.lepro.2003@gmail.com (hp-bureau)
mathis.breton.2005@gmail.com (hp-bureau)
Music
Pictures
pierrefelix.breton@gmail.com (hp-bureau)
Videos
Windows (C:)
Google Drive File Stream (G:)
Root (P:)
Developers (\\epicgames.net\root) (Q:)

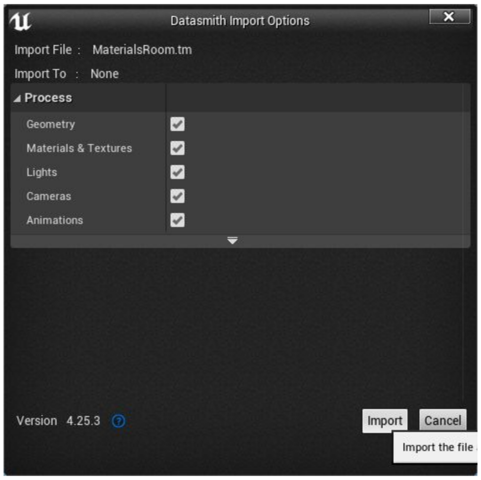
Name	Date modified	Type	Size
maps	8/10/2020 9:46 AM	File folder	
MaterialsRoom.tm	8/10/2020 9:31 AM	TM File	90,191 KB

File name: MaterialsRoom.tm

All Files (*.sat;*.dwg;*.3drep;*.3

Then, the Datasmith import workflow will prompt you for the usual input:


Choose Location for importing the Datasmith content


Datasmith Import Options

Import File : MaterialsRoom.tm

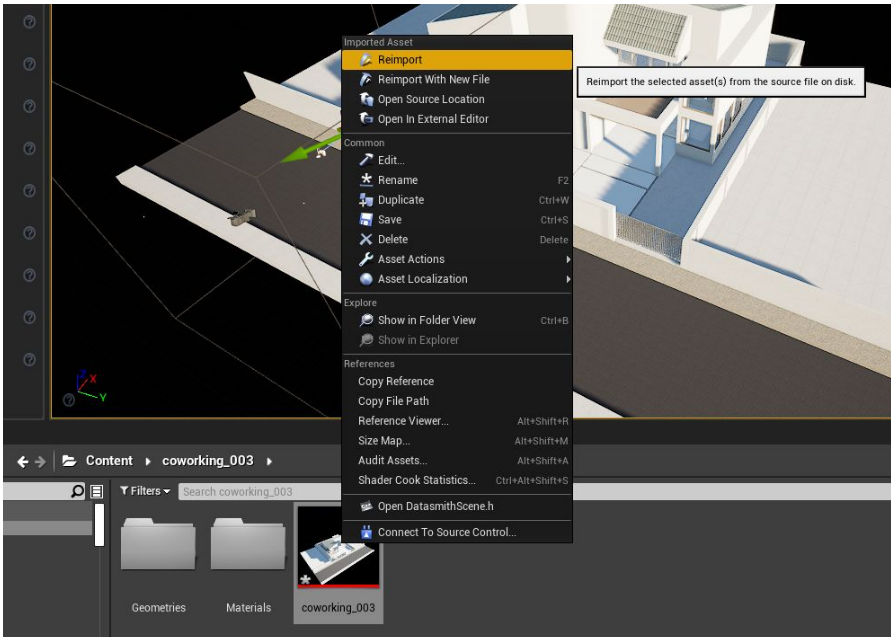
Import To : None

Process

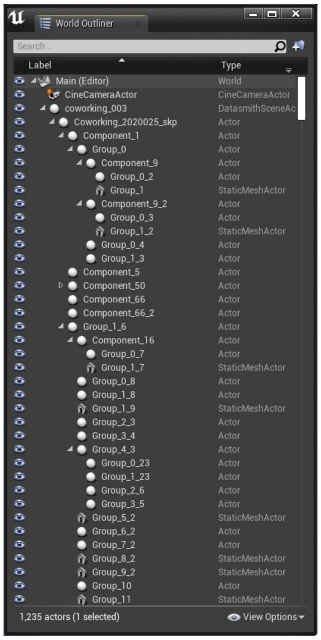
- Geometry ☑
- Materials & Textures ☑
- Lights ☑
- Cameras ☑
- Animations ☑

Version 4.25.3

Upon completion of the initial import, you will see Datasmith Scene in the assets browser with the ability to re-import the Twinmotion file, along with the ability to override/reset overrides on each actor - which is something standard with any Datasmith based imports:

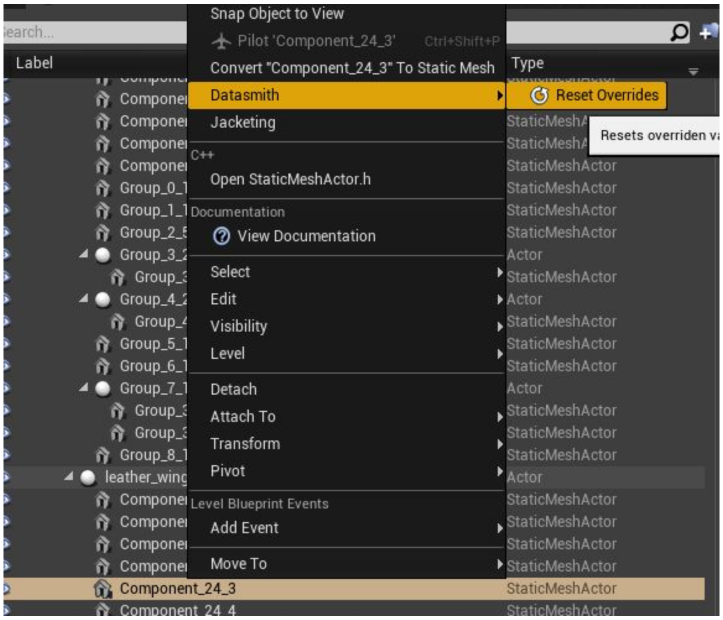


Reimport the selected asset(s) from the source file on disk.



Label	Type
Main (Editor)	World
CineCameraActor	CineCameraActor
coworking_003	DatasmithSceneAc
Coworking_2020025_skp	Actor
Component_1	Actor
Group_0	Actor
Component_9	Actor
Group_0_2	Actor
Group_1	StaticMeshActor
Component_9_2	Actor
Group_0_3	Actor
Group_1_2	StaticMeshActor
Group_0_4	Actor
Group_1_3	Actor
Component_5	Actor
Component_50	Actor
Component_66	Actor
Component_66_2	Actor
Group_1_6	Actor
Component_16	Actor
Group_0_7	Actor
Group_1_7	StaticMeshActor
Group_0_8	Actor
Group_1_8	Actor
Group_1_9	StaticMeshActor
Group_2_3	Actor
Group_3_4	Actor
Group_4_3	Actor
Group_0_23	Actor
Group_1_23	Actor
Group_2_6	Actor
Group_3_5	Actor
Group_5_2	StaticMeshActor
Group_6_2	Actor
Group_7_2	Actor
Group_8_2	StaticMeshActor
Group_9_2	StaticMeshActor
Group_10	Actor
Group_11	StaticMeshActor

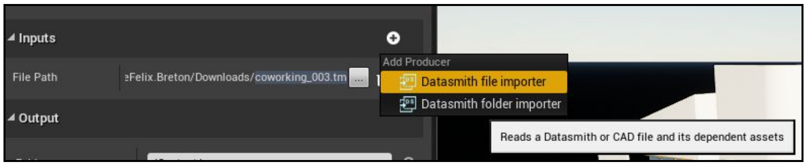
1,235 actors (1 selected)



Resets overriden va

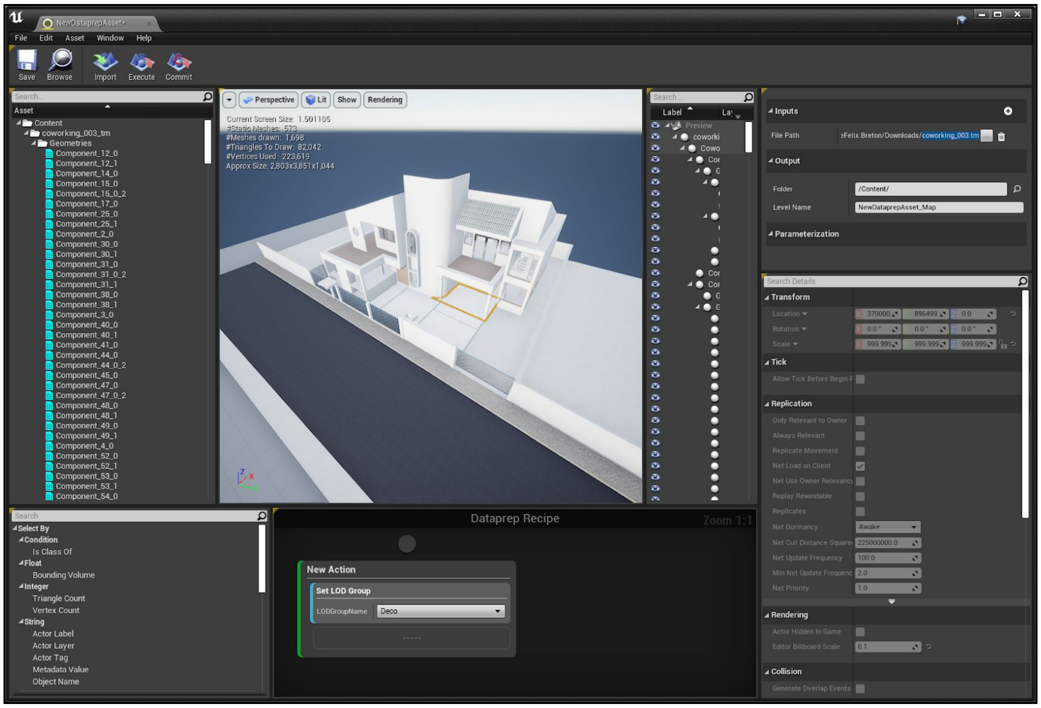
Dataprep workflow:

Just like Datasmith, the TM import plugin is compatible with the Visual Dataprep tool. Simply pick the *.tm file from the Inputs > Datasmith Importer and build your import recipes as any other type of Datasmith based sources:



File Path ɘFelix.Breton/Downloads/coworking_003.tm

Reads a Datasmith or CAD file and its dependent assets



Current Screen Size: 1.501105
#Static Meshes: 573
#Meshes drawn: 1,698
#Triangles To Draw: 82,042
#Vertices Used: 223,619
Approx Size: 2,803x3,851x1,044

LODGroupName: Deco

File Path: ɛFelix.Breton/Downloads/coworking_003.tm
Folder: /Content/
Level Name: NewDataprepAsset_Map

Location: X 370000. Y 896499. Z 0.0
Rotation: X 0.0° Y 0.0° Z 0.0°
Scale: X 999.999 Y 999.999 Z 999.999
Allow Tick Before Begin P ☐
Only Relevant to Owner ☐
Always Relevant ☐
Replicate Movement ☐
Net Load on Client ☑
Net Use Owner Relevancy ☐
Replay Rewindable ☐
Replicates ☐
Net Dormancy: Awake
Net Cull Distance Square: 225000000.0
Net Update Frequency: 100.0
Min Net Update Frequenc: 2.0
Net Priority: 1.0
Actor Hidden In Game ☐
Editor Billboard Scale: 0.1
Generate Overlap Events ☐

Scope and roadmap (Public Beta 1)

This effort will span across several Twinmotion updates.

Although Twinmotion is built with Unreal Engine, loading data in the Unreal Engine Editor from a cooked game (Twinmotion) is not as straightforward as it may sound.

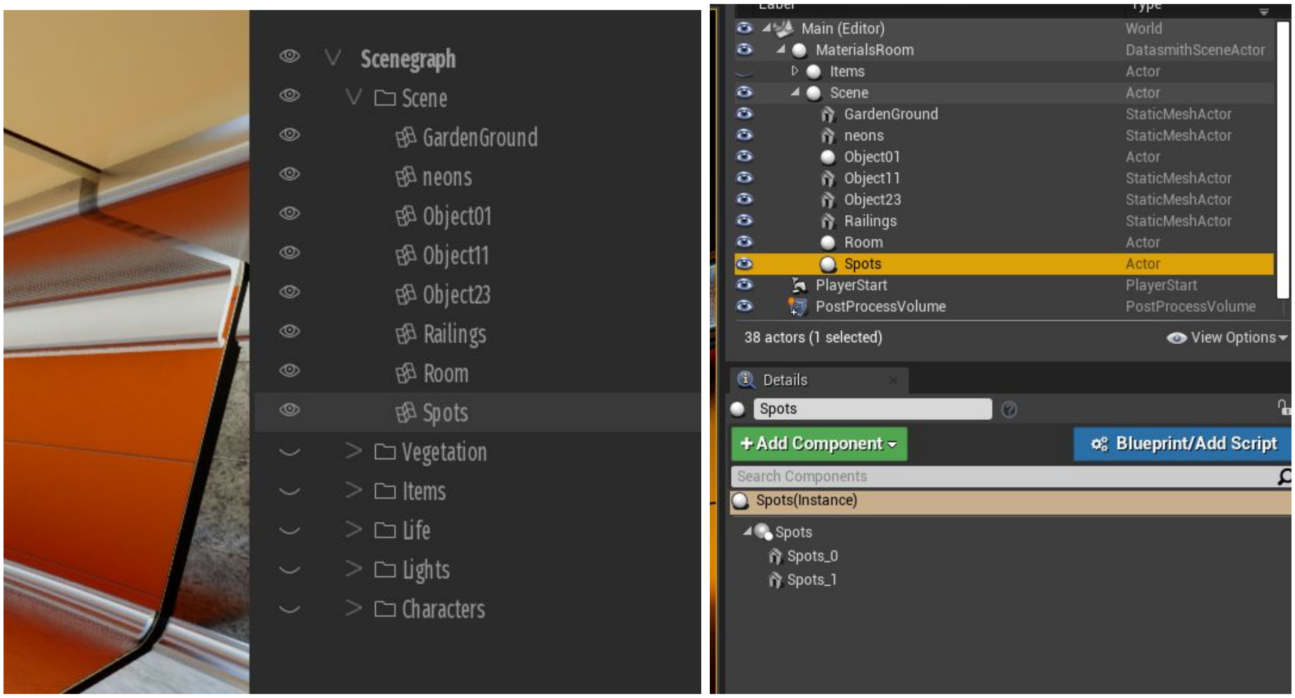
Not only do we want to load Twinmotion data in Unreal, we want to make sure that it's usable by Unreal users without carrying tons of parameters that would normally be only useful for internal mechanisms implemented for Twinmotion.

Therefore a significant refactoring of Twinmotion assets is required - we are going to accomplish this in several phases:

What works?

“Imported” geometry

Geometry imported in Twinmotion (Archicad, Rhino, Revit, Sketchup etc.) will be translated as Static Meshes.The hierarchy in the World Outliner will be identical to the one found in the Twinmotion Scenegraph, resulting in a combination of Actors and Mesh Components.



- Scenegraph
  - Scene
    - GardenGround
    - neons
    - Object01
    - Object11
    - Object23
    - Railings
    - Room
    - Spots
  - Vegetation
  - Items
  - Life
  - Lights
  - Characters

Label	Type
Main (Editor)	World
MaterialsRoom	DatasmithSceneActor
Items	Actor
Scene	Actor
GardenGround	StaticMeshActor
neons	StaticMeshActor
Object01	Actor
Object11	StaticMeshActor
Object23	StaticMeshActor
Railings	StaticMeshActor
Room	Actor
Spots	Actor
PlayerStart	PlayerStart
PostProcessVolume	PostProcessVolume

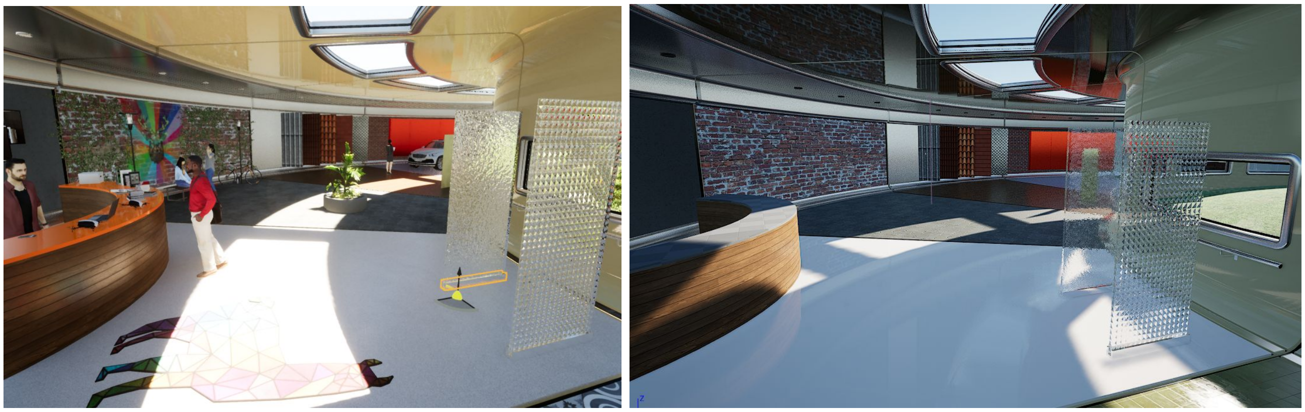
38 actors (1 selected)

Spots(Instance)

- Spots
  - Spots_0
  - Spots_1

Twinmotion Scenegraph vs Unreal Editor Hierarchy

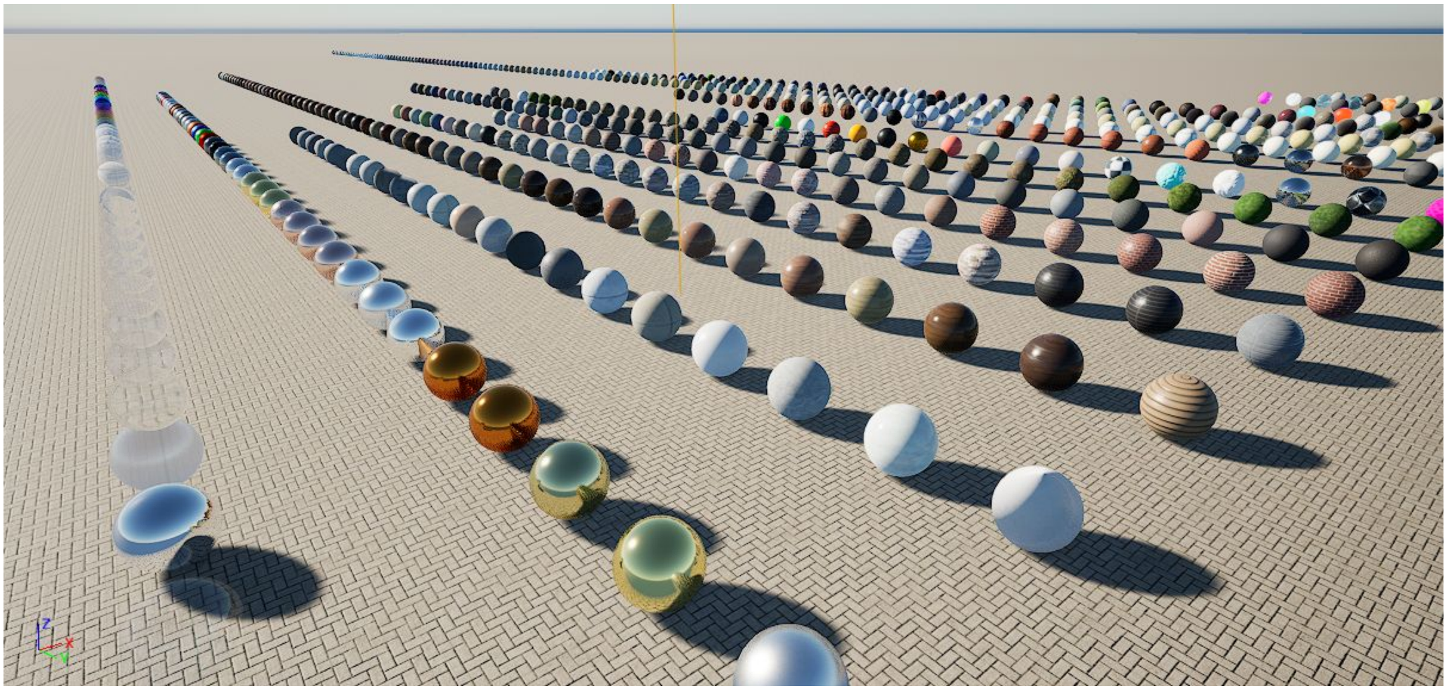
All other entities will be ignored by the importer at this time (furniture, terrain, foliage, decals, lights etc.).



Twinmotion vs Unreal - Beta 1 - Several Object Types are not supported yet

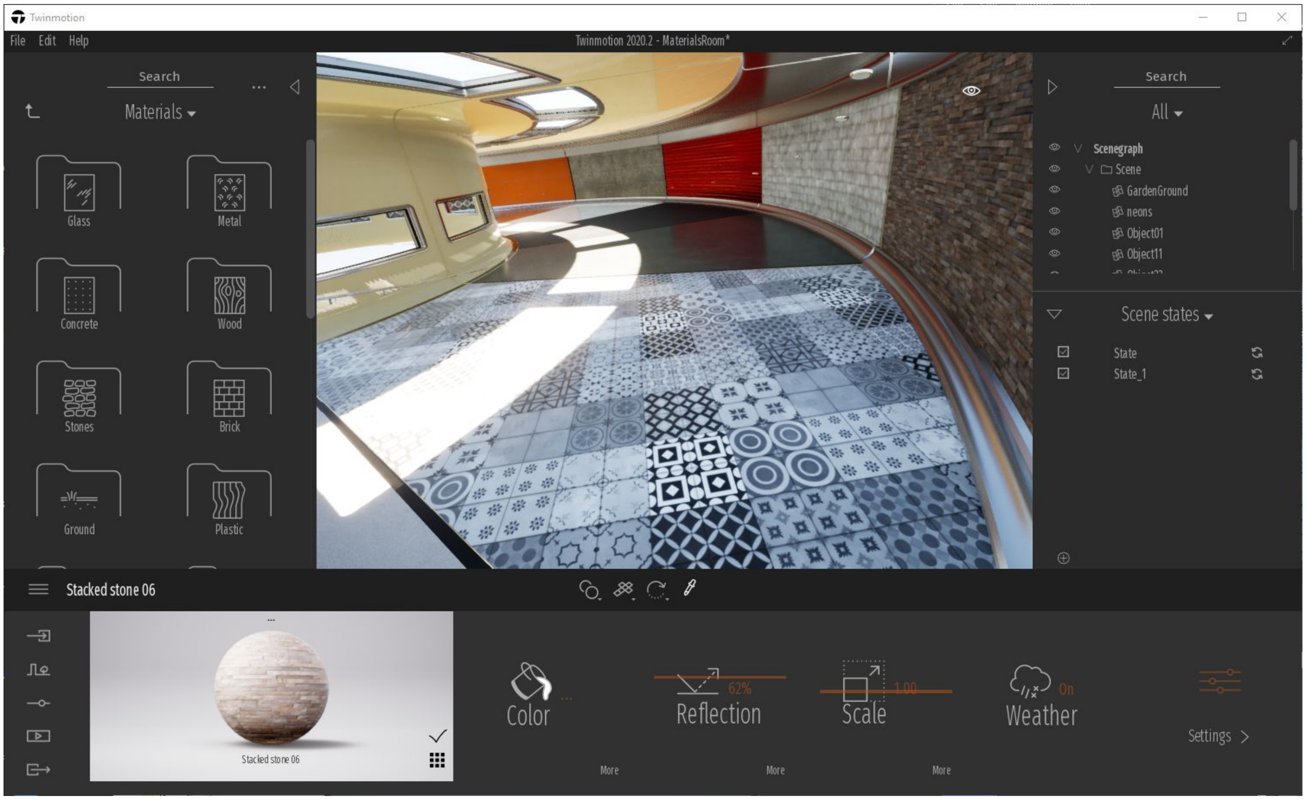
Materials

Our goal is to import 100% of the Materials available in the Twinmotion library as well as materials that you might have created with your own texture maps.

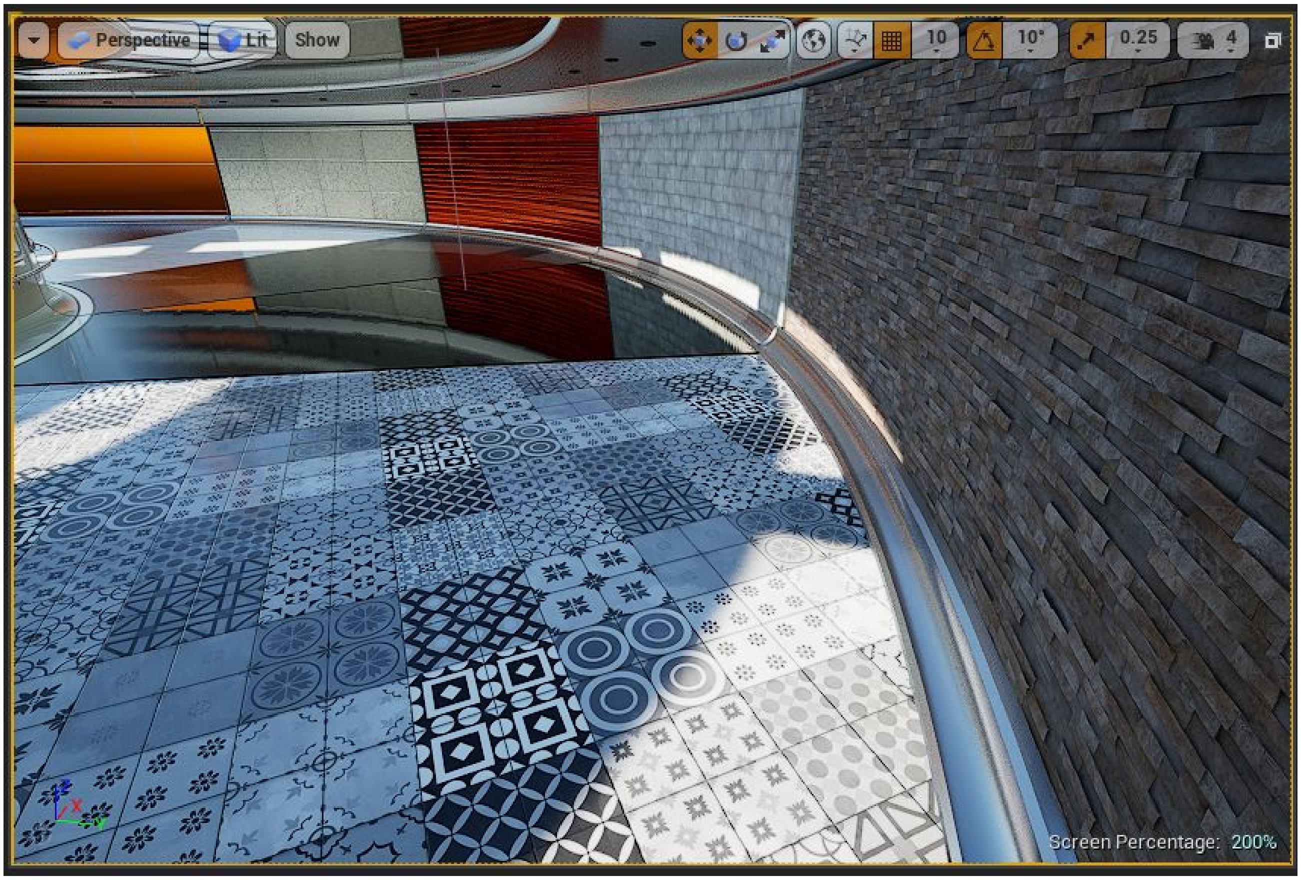


Master materials received a significant amount of our attention:

- Refactored to expose PBR parameters with a comprehensive set of parameters that reflects today's standards.
- Optimized for general performance in raster and raytracing.
- Additional parameters exposed to override UVs with Triplanar Projection

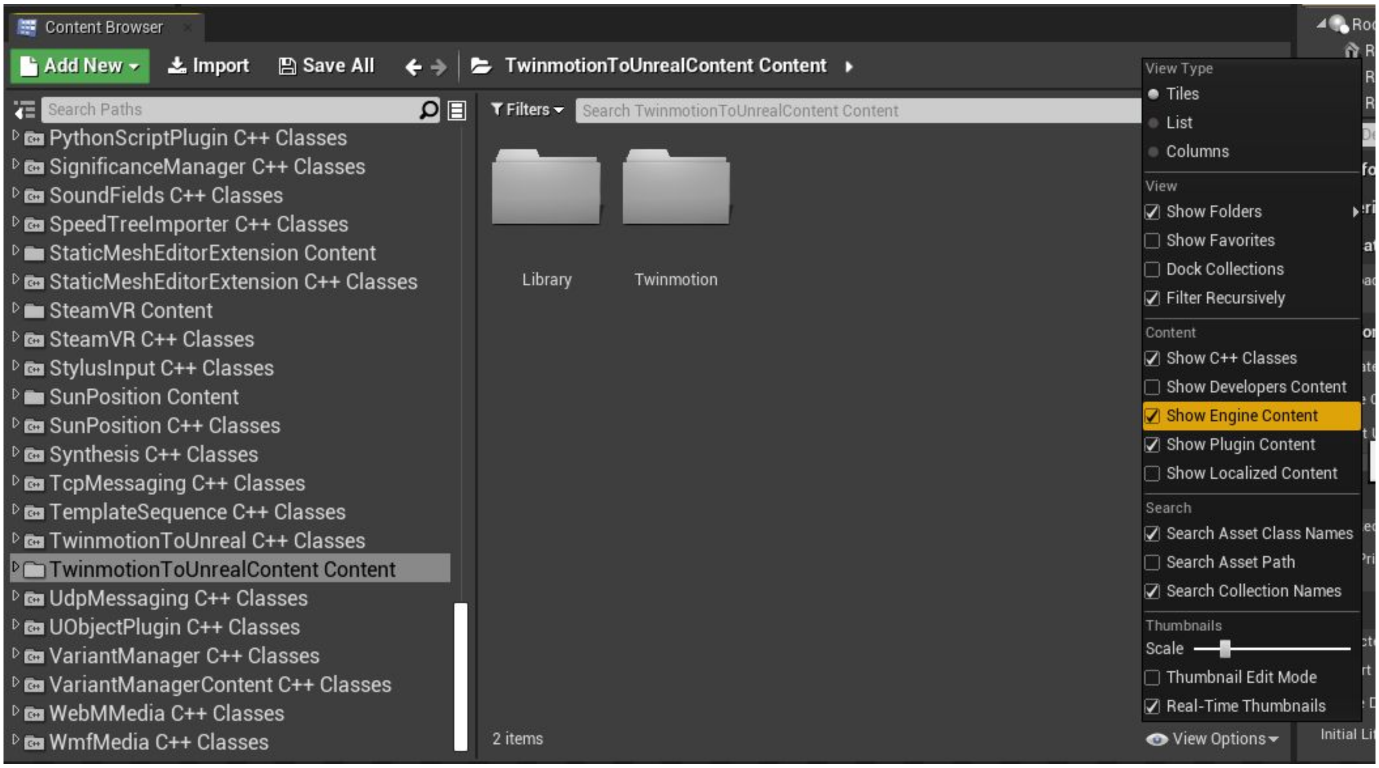


Twinmotion



Unreal Editor

Additionally, we made sure that the Material presets from Twinmotion are available without the need to import a Twinmotion project. They can be found under TwinmotionToUnrealContent Content folder - feel free to use them however you want.


Content Browser
Add New
Import
Save All
TwinmotionToUnrealContent Content
Search Paths
Filters
Search TwinmotionToUnrealContent Content
PythonScriptPlugin C++ Classes
SignificanceManager C++ Classes
SoundFields C++ Classes
SpeedTreeImporter C++ Classes
StaticMeshEditorExtension Content
StaticMeshEditorExtension C++ Classes
SteamVR Content
SteamVR C++ Classes
StylusInput C++ Classes
SunPosition Content
SunPosition C++ Classes
Synthesis C++ Classes
TcpMessaging C++ Classes
TemplateSequence C++ Classes
TwinmotionToUnreal C++ Classes
TwinmotionToUnrealContent Content
UdpMessaging C++ Classes
UObjectPlugin C++ Classes
VariantManager C++ Classes
VariantManagerContent C++ Classes
WebMMedia C++ Classes
WmfMedia C++ Classes
Library
Twinmotion
2 items
View Options

View Type
- ● Tiles
- ○ List
- ○ Columns

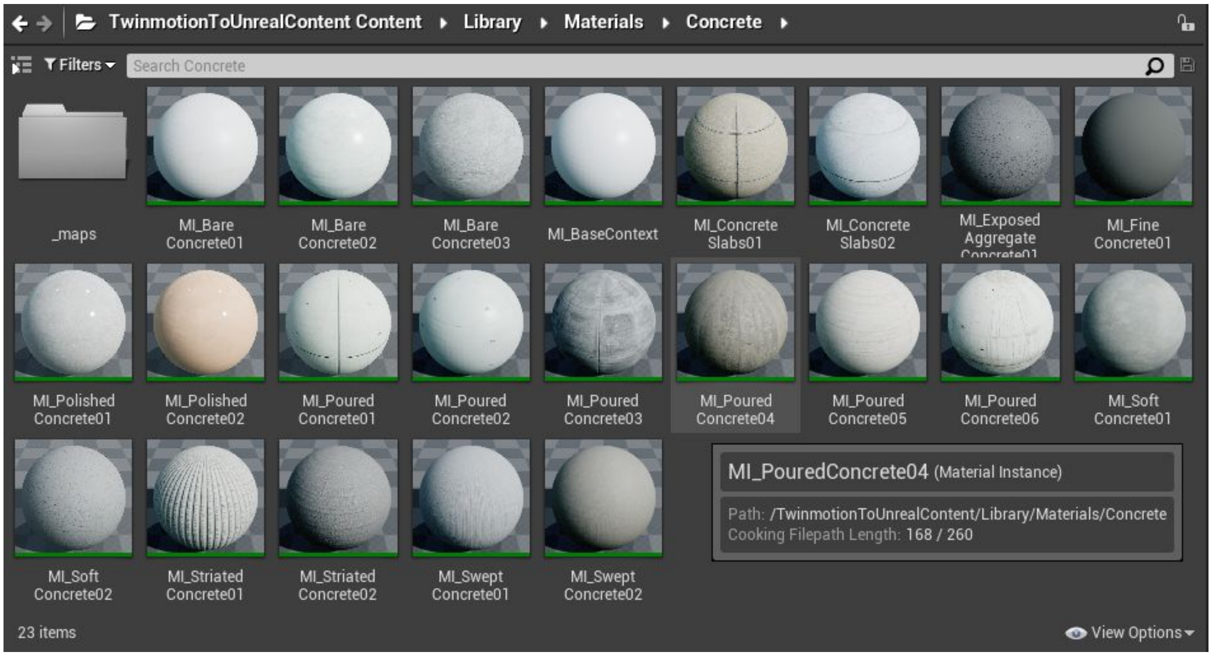
View
- ☑ Show Folders
- □ Show Favorites
- □ Dock Collections
- ☑ Filter Recursively

Content
- ☑ Show C++ Classes
- □ Show Developers Content
- ☑ Show Engine Content
- ☑ Show Plugin Content
- □ Show Localized Content

Search
- ☑ Search Asset Class Names
- □ Search Asset Path
- ☑ Search Collection Names

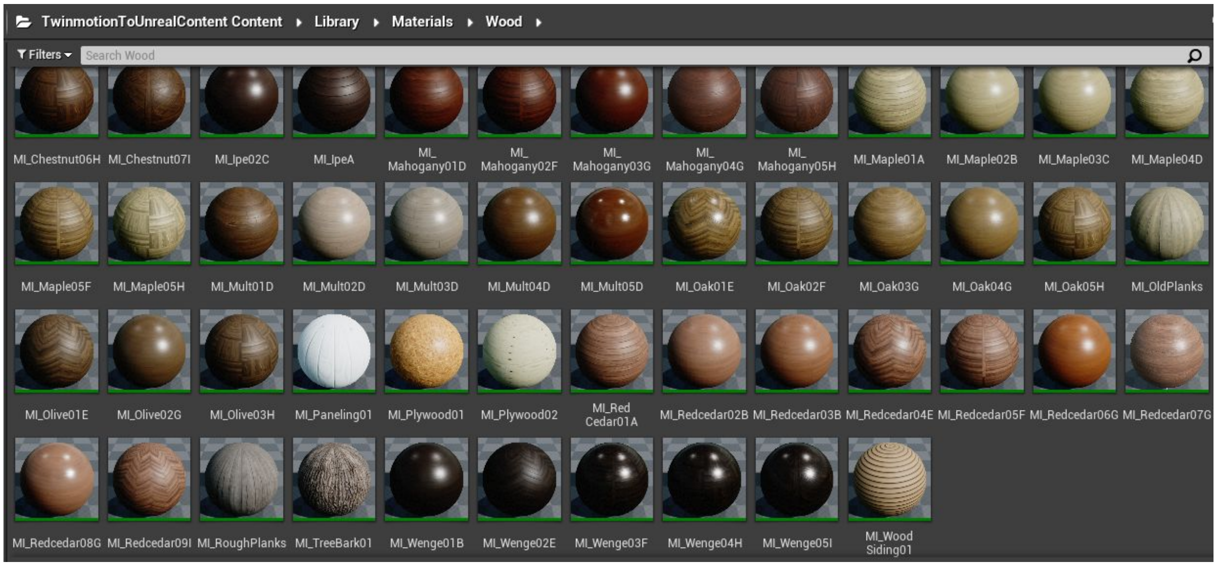
Thumbnails
- Scale
- □ Thumbnail Edit Mode
- ☑ Real-Time Thumbnails

Tip: enable "Show Engine Content" from the View Options

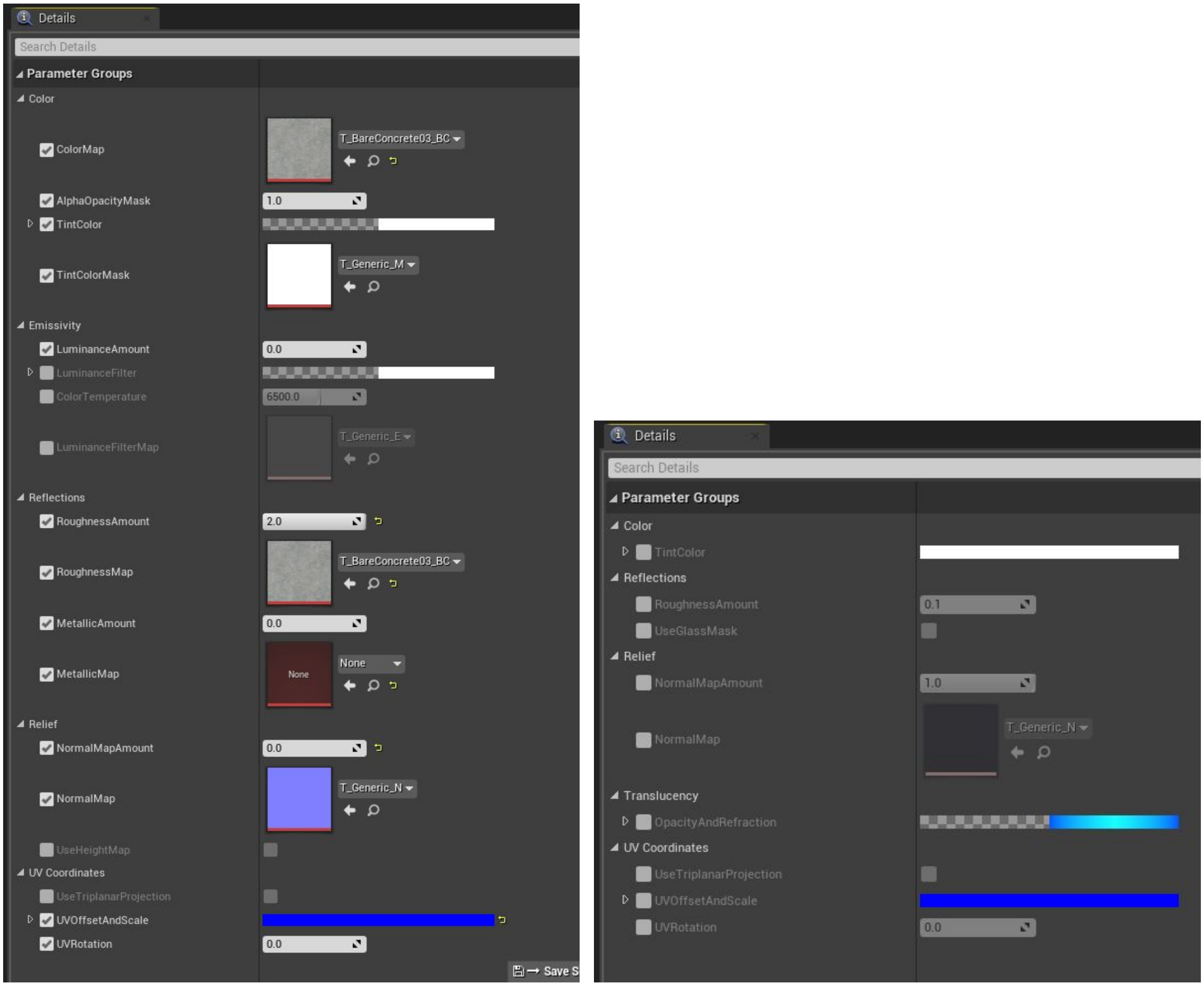


MI_PouredConcrete04 (Material Instance)

Path: /TwinmotionToUnrealContent/Library/Materials/Concrete
Cooking Filepath Length: 168 / 260



Example of material instances included with the Twinmotion Content plugin



Parameter Groups

Color
- ☑ ColorMap: T_BareConcrete03_BC
- ☑ AlphaOpacityMask: 1.0
- ☑ TintColor
- ☑ TintColorMask: T_Generic_M

Emissivity
- ☑ LuminanceAmount: 0.0
- □ LuminanceFilter
- □ ColorTemperature: 6500.0
- □ LuminanceFilterMap: T_Generic_E

Reflections
- ☑ RoughnessAmount: 2.0
- ☑ RoughnessMap: T_BareConcrete03_BC
- ☑ MetallicAmount: 0.0
- ☑ MetallicMap: None

Relief
- ☑ NormalMapAmount: 0.0
- ☑ NormalMap: T_Generic_N
- □ UseHeightMap: □

UV Coordinates
- □ UseTriplanarProjection: □
- ☑ UVOffsetAndScale
- ☑ UVRotation: 0.0

Parameter Groups

Color
- □ TintColor

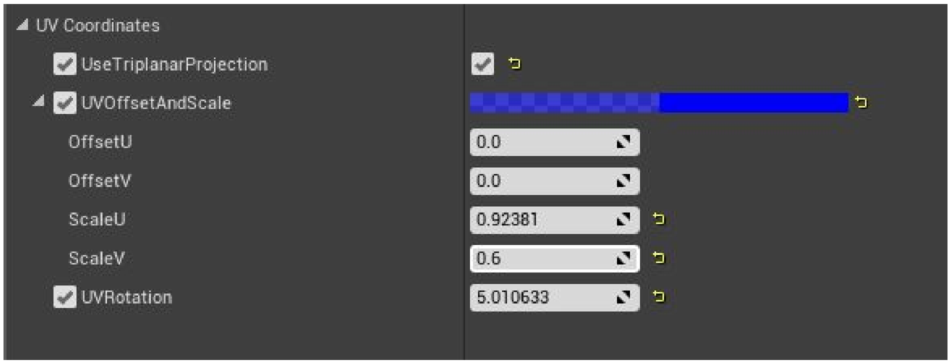
Reflections
- □ RoughnessAmount: 0.1
- □ UseGlassMask: □

Relief
- □ NormalMapAmount: 1.0
- □ NormalMap: T_Generic_N

Translucency
- □ OpacityAndRefraction

UV Coordinates
- □ UseTriplanarProjection: □
- □ UVOffsetAndScale
- □ UVRotation: 0.0

M_StdOpaque and M_StdTranslucent materials exposing familiar terminology.



UV Coordinates

☑ UseTriplanarProjection ☑

☑ UVOffsetAndScale

OffsetU 0.0

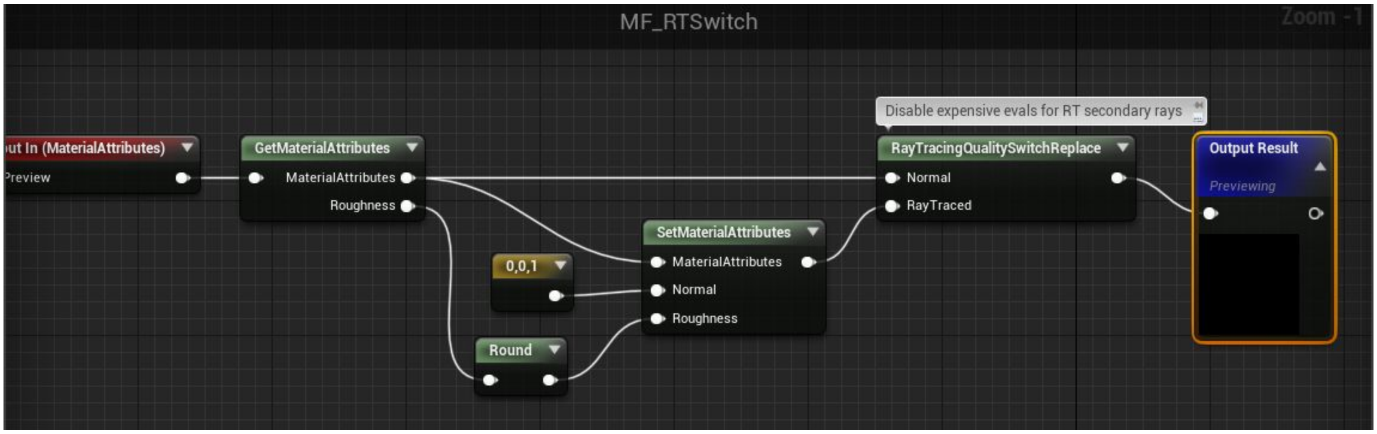
OffsetV 0.0

ScaleU 0.92381

ScaleV 0.6

☑ UVRotation 5.010633

Ability to override model`s UVs with Triplanar Projection



Example of performance optimization for raytracing

Furniture library assets

The House and City furniture assets from the library have been migrated and are now working with the importer.

Skeletal meshes and vehicles are not fully supported yet (only flags will import, use play or simulate for physics, works also with WindDirectionalSource).

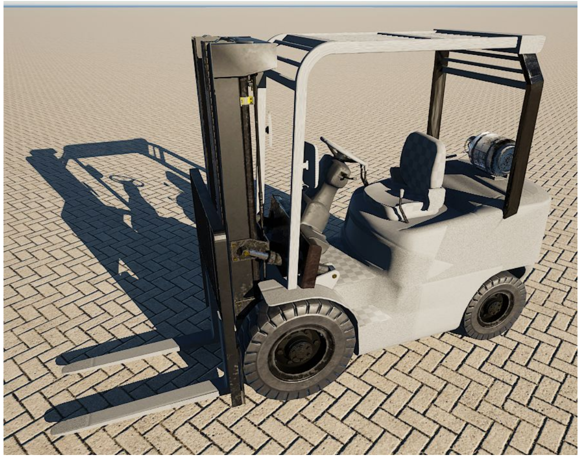
Vegetation

Grass, shrubs, trees, rocks and misc assets have been optimized and can now be imported.
We also support mesh instances placed with the paint tool.

Known issues

- Using **Dataprep** with a Twinmotion file will crash the engine
- Missing low flying plane sign will trigger an error at import

- Forklift entity is missing materials and will trigger warnings at import



Future Releases

Here are the main Twinmotion features that we intend to support in subsequent releases.

- Sculpted Landscape
- Daylight / Moonlight
- Vehicles
- Characters
- Cameras
- Artificial Lights
- Decals
- Water Objects and Materials
- Dynamic objects (ex: character paths)
- Animators and Objects referenced by Animators
- Section planes and cut volumes
- Measurement Tools
- Notes
- Weather System
- Reflection Volumes
- Sounds
- Particles