Datasmith

As a part of the Unreal Studio offering, **Datasmith for Unreal Engine 4** is a collection of tools and plugins designed to import scenes and assets created in offline rendering applications, like Autodesk 3DS Max and CAD software, into a real-time engine. Unreal Engine 4 (UE4) enables you to render your scenes and assets in real-time and make changes dynamically.

Currently, three main components make up Datasmith for Unreal Engine 4:

1. Application specific export plugins capable of creating *.UDATASMITH files
2. The Unreal Datasmith Importer for Unreal Datasmith (*.UDATASMITH) files
3. The Unreal CAD Importer for native file types

Supported Software and File Types

Datasmith supports a variety of software and file types, which can either be imported natively or through an external plugin that enables the custom Datasmith (*.UDatasmith) file type to be used. For additional information about the software and file types that are supported, see the Datasmith Supported File Types and Software page.

Datasmith Workflow

There are a couple of ways that scenes and assets are imported for use with Datasmith; by using a custom exporter plugin written for specific software (like Autodesk 3DS Max) to create the Unreal Datasmith file type, or by supporting the native file type of some CAD software.

Unlike traditional game design workflows where assets are constructed and imported one by one as individual assets, your assets are imported as fully assembled scenes with geometry, lights, cameras, materials, and textures already applied and in their proper locations. Once this data is loaded into the Engine, the data created by the importer is ultimately the same and all assets become standard Unreal Engine assets, like Actors in a level, Static Meshes, Materials, Textures, and Lights.

When your scene is saved to an Unreal Datasmith file or a supported CAD file type, the following asset information is converted to one that Unreal Engine 4 can use:

- Object Instances
- Pivot Locations
- Scene Hierarchy and Layers
- Material and Physical Based Rendering (PBR) Characteristics and Textures
- Light Positions, Colors, Sizes, and Intensities
- Camera Properties
- Meta Information and Custom Attributes
- Unit Conversion

The importer also automates the generation and translation of time-consuming tasks like:

- Generating and packing a Lightmap UV
- Converting image formats to ones supported by Unreal Engine

- Converting Bump Maps to Normal Maps

Unreal Datasmith Files

The Unreal Datasmith Importer in UE4 supports Unreal Datasmith (*.UDATASMITH) files that are created using an exporter plugin. The Unreal Datasmith file type is one created specifically for use with some applications to output that custom file type.

Unreal Datasmith Exporter Plugins

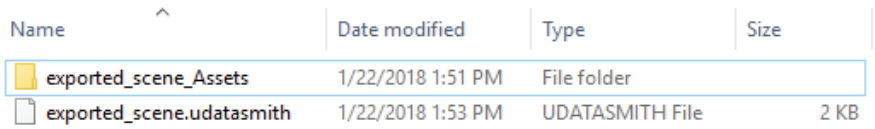
Some software applications will not natively support the Datasmith Importer for UE4. In these instances, you'll need to use one of the Unreal Datasmith exporter plugins to generate an *.UDATASMITH file type that can be imported into UE4. The exporter plugin is responsible for collecting and packaging the scene assets, geometry, lights, and cameras in a way that conforms to the Unreal Engine 4 data model. During the process, the exporter translates important elements to ones that Unreal can use.

When using an Unreal Datasmith exporter plugin, you can save and export your assets from your software by choosing **Unreal Datasmith (*.UDATASMITH)** as the save as file type.

> You can download the exporter plugins from the Unreal Studio Downloads page. For a comprehensive listing of software applications that require a custom exporter plugin to be installed, see the Datasmith Supported Software and File Types page.

Unreal Datasmith File Structure

When exporting a *.UDATASMITH file type the following folder structure and asset are created:



- A file with the *.UDATASMITH extension
- A folder which contains meshes, maps, IES profiles, and all the compatible Unreal Engine 4 scene assets

The *.UDATASMITH file contains XML with references to the folder containing all of the exported assets from your scene. For the Datasmith Importer in UE4 to work properly, the *.UDATASMITH file and its folder must be available during import.

```
<DatasmithUnrealScene>
	<Version>0.17</Version>
	<Host>Autodesk 3dsmax 18.0.873 Default Scanline Renderer</Host>
	<Application Vendor="Autodesk" ProductName="3dsmax" ProductVersion="18.0.873"/>
	<User ID="7d8a1a304be9658328c158a3d14cc774" OS="Windows 10"/>
	<StaticMesh name="2">
		<file path="exported_scene_Assets/2.udsmesh"/>
		<Size a="40000.0" x="100.0" y="100.0" z="100.0"/>
		<LightmapUV value="1"/>
	</StaticMesh>
	<ActorMesh name="2" label="Box001" layer="0">
		<mesh name="2"/>
		<material id="-1" name="01_-_Default1"/>
		<Transform tx="0.0" ty="0.0" tz="0.0" sx="1.0" sy="1.0" sz="1.0" qx="0.0" qy="0.0" qz="0.0" qw="1.0" qhex="00000000000000000000000000000003F"/>
	</ActorMesh>
	<Camera name="14" label="Camera001" layer="0">
		<Transform tx="452.243408" ty="0.0" tz="296.045898" sx="1.0" sy="1.0" sz="1.0" qx="0.187278" qy="-0.0" qz="-0.982308" qw="-0.0" qhex="DFC53F3E000000B382787B8F000000B3"/>
		<SensorWidth value="36.0"/>
		<SensorAspectRatio value="1.333333"/>
		<FocusDistance value="1000.0"/>
		<FStop value="5.6"/>
		<FocalLength value="43.455841"/>
	</Camera>
	<Light name="15" type="PointLight" enabled="1" label="Omni001" layer="0">
		<Transform tx="344.342529" ty="-206.554565" tz="337.549316" sx="1.0" sy="1.0" sz="1.0" qx="-0.5" qy="0.5" qz="0.5" qw="0.5" qhex="FFFFFFBEFEFFFF3EFEFFFF3EFDFFFF3E"/>
```

```
            <Color usetemp="0" temperature="6500.0" R="1.0" G="0.792157" B="0.572549"/>
            <Intensity value="1125.0"/>
            <IntensityUnits value="Candelas"/>
        </Light>
        <Texture name="checkermap_192f_2_200027_Tex" texturemode="5" rgbcurve="1.000012" file="exported_scene_Assets/checkermap_192f_DS.tga"/>
        <Material name="01_-_Default1">
            <Shader name="01_-_Default1">
                <IORRefraction value="1.5"/>
                <Bumpval value="1.0"/>
                <Diffuse tex="checkermap_192f_2_200027_Tex" coordinate="0" sx="1.0" sy="1.0" ox="0.0" oy="0.0" rot="0.0" mul="1.0" channel="0" inv="0" cropped="0"/>
                <Roughnessval value="0.9" desc="Roughness"/>
            </Shader>
        </Material>
        <Export Duration="0"/>
</DatasmithUnrealScene>
```

Click image for full size.

In this example XML, you can find references to four types of assets being exported and managed:

- Static Mesh Geometry
- Camera
- Light
- Material

Each of these assets store there assigned materials, textures, asset-specific properties, and their location and orientations. When they are imported into UE4, all these references are converted to properties, settings, and assignments that it can understand.

Unreal Datasmith API

The Unreal Datasmith file type API is being created to enable third-party developers to write their own file exporters. It will enable the ability to make compatible assets that work with the Unreal Datasmith Importer for Unreal Engine 4 by converting textures, packing lightmap UVs, and much more.

> The Datasmith API is evolving quickly during this time in development and has not been made public at this time due to this reason.

> You should not attempt to build any pipelines by reverse engineering this data as its format will change as needed. Consider this format an intermediate Unreal scene representation and not an exchange format between 3D applications, like FBX. The Datasmith format will change between releases as it is development and compatibility is not expected between these different versions, although it may be the case from time to time.

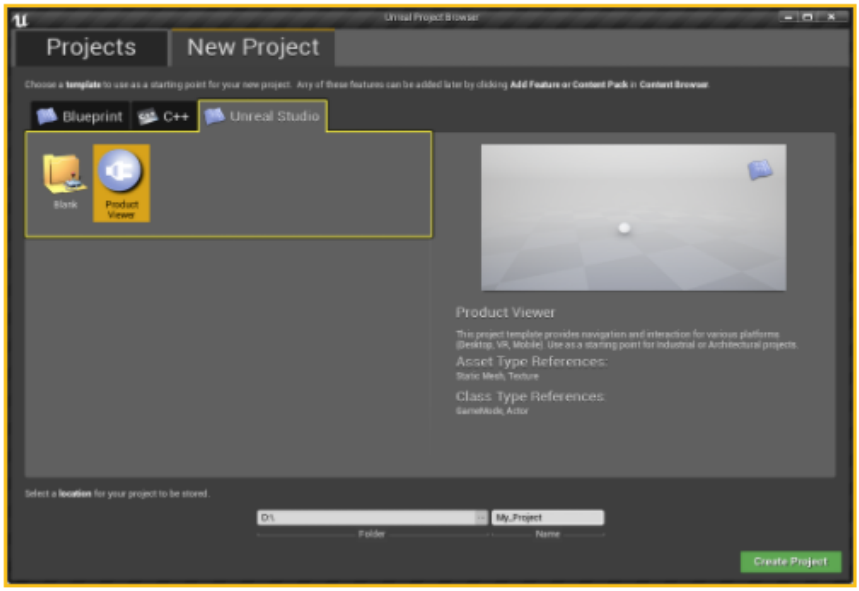
CAD Files

The Datasmith Importer in UE4 currently supports a variety of CAD software with native file support. These file types can be imported into UE4, and the importer translates important elements into ones that Unreal can use.

> With the native file support of some CAD software, there are limitations of known issues that may be resolved in a future release. For more detailed information about your CAD software package, see the Datasmith Supported Software and File Types page.

Creating an Unreal Studio Project

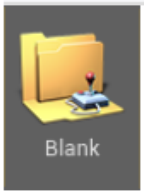
Unreal Studio projects are ones that automatically enable the Unreal Studio Plugins offering when created. Unreal Studio projects can be created by using the Unreal Project Browser or by converting an existing project to an Unreal Studio one.



Click image for full size.

Unreal Studio Templates

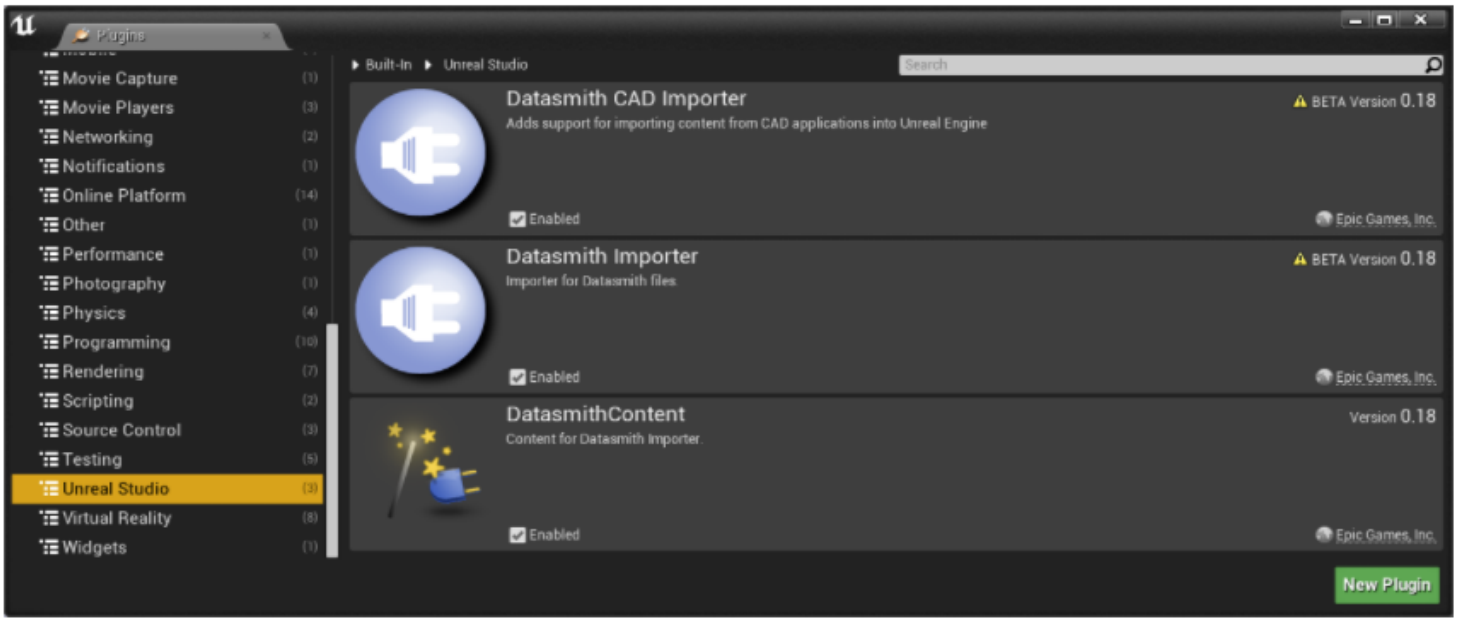
Under the New Project tab of the Unreal Project Browser, when you select the Unreal Studio tab, you'll be able to select from the available templates that are specific to Unreal Studio, similarly to how Blueprint and C++ have their own templates.

Icon	Description
	This is a clean empty project with no additional content added.
	This project template provides navigation and interaction for various platforms (Desktop, VR, and Mobile) to load individual assets for display and interaction. This project can be used as a starting point for Industrial and Architectural projects. Additional content can be added as needed and is not limited to just the Product Viewer. Use the instructions in the viewport to set up your asset interaction for the Product Viewer. 

Click image for full size.

Importing Unreal Datasmith and CAD files

For Datasmith assets to be imported there are a few Datasmith plugins that are automatically enabled when you create an Unreal Studio project. These plugins can be found in the Plugins window under the **Unreal Studio** section:



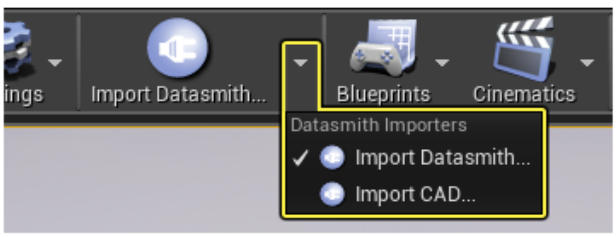
These plugins are:

- **Datasmith CAD Importer:** Supports a variety of CAD software file types.
- **Datasmith Importer:** Supports the Unreal Datasmith file type created by the Datasmith Exporter plugins.
- **Datasmith Content:** Enables support for Datasmith Content in the Content Browser.

> For Datasmith to function, the project must be associated as an Unreal Studio project. If the project was not created using one of the Unreal Studio templates, you can read how to convert a project to an Unreal Studio one in the guide for Downloading Unreal Engine 4 and Installing Datasmith.

When importing Datasmith assets, use the Main Toolbar button to select **Import Datasmith File** or **Import CAD File**.

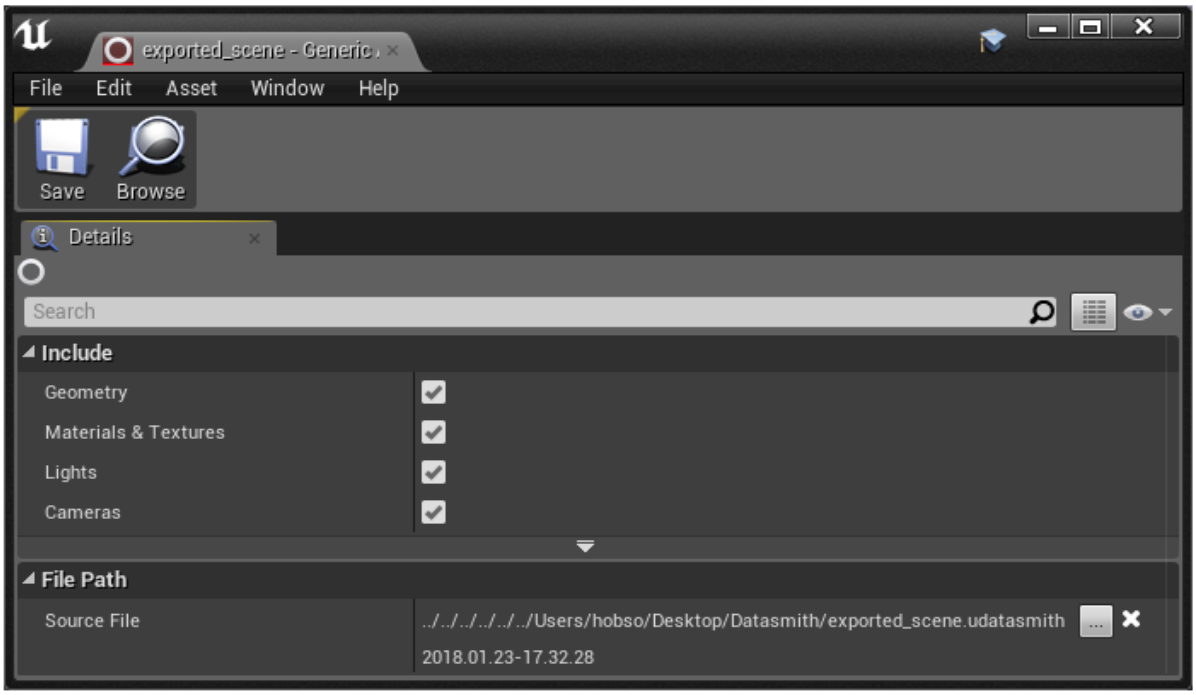
Use the drop-down selection to change which type of file you want to import.



Datasmith Scene Asset

When you import your Datasmith content, a **Datasmith Scene** asset and folder are created that store your static meshes, materials, and textures. The Datasmith Scene asset references the collection of assets imported as Datasmith Content.

The Datasmith Scene asset contains the Datasmith Import Options dialog that is available when importing your content. It also stores a reference to the file path of the source file used for import.



It is important to note that the process of importing a Datasmith file is different than importing content using the import button through the Content Browser or using the FBX Scene Import. Datasmith importing should be treated as in independent process and the Datasmith Scene as an individual asset that contains references to the following information:

- Actors that are placed in the Active (currently loaded) Level
    - The assets are placed under a root Actor in the World Outliner named according to the name of the imported Datasmith or CAD file with an appended *_Root tag.
- Scene assets are placed in the Content Browser
    - The assets are placed in a folder named according to the imported Datasmith or CAD file. This folder contains the Datasmith Scene asset and additional folders for Geometries (Static Meshes), Materials, and Textures.

Datasmith assets are imported, stored, and referenced in a specific way that prevents them from being imported directly through the Content Browser or using the FBX Scene Import

Reimporting Datasmith Assets and Scenes

The reimport process is made possible in Unreal Engine 4 with the Datasmith Scene asset, which contains references to the originally imported assets. Because of how the assets are referenced, there are two ways reimport your assets:

- As a full Datasmith Scene
- Individual Assets

As you develop your project, you will often reimport a scene at different intervals in your development cycle that includes minor or major adjustments to its content.

> Exporting a new *.UDATASMITH or CAD file for import creates a new Datasmith Scene asset if it has a different name but the same assets as one your previously imported. It's best to overwrite existing file when exporting and then use the reimport functionality in UE4.

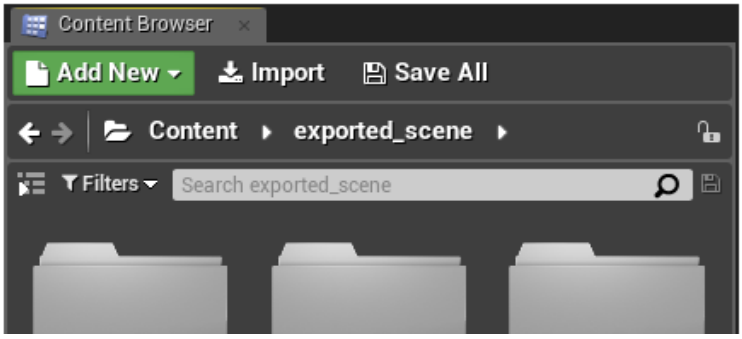
Full Scene Reimport

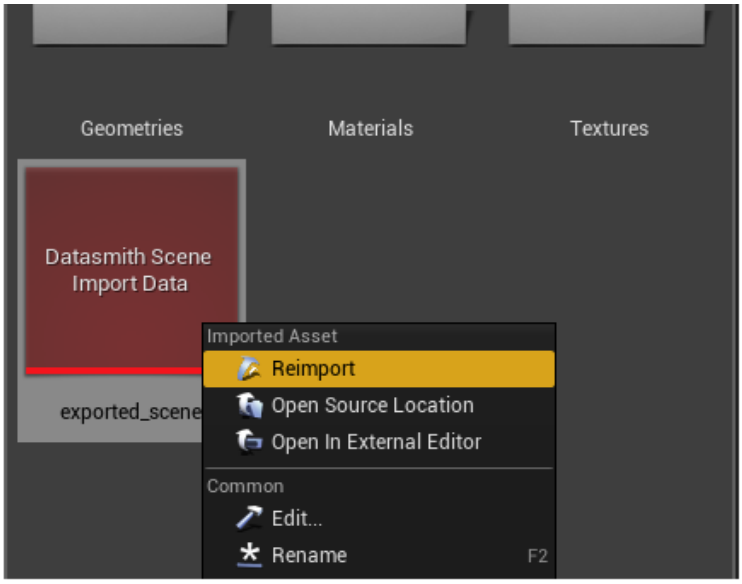
When you reimport your full Datasmith scene, Unreal performs the following actions:

1. Actor hierarchy is reset.
2. Actor positions and orientations are updated.
3. All Static Meshes referenced by Actors in the Datasmith Scene asset are reimported.
4. Materials are reassigned to Actors referenced in the Datasmith Scene asset.
5. Newly added Actors are added to the Level.
6. Actors deleted in Unreal are added back to the scene (if they exist in the file being reimported)

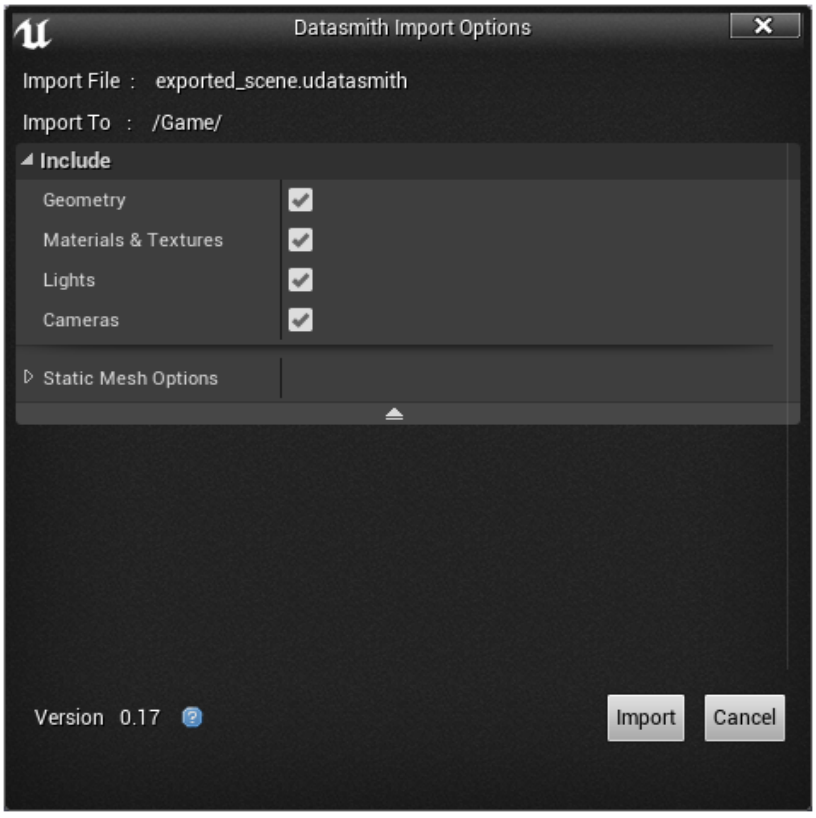
> An Actor here refers to any object that is placed into a level. That includes, but is not limited to, geometry, lights, a skydome, cameras, or any other object types not explicitly mentioned.

To reimport your Datasmith scene, right-click on the **Datasmith Scene** asset and choose **Reimport** from the context menu.





The Datasmith Import Options dialog will open, you can choose which asset types to reimport as part of the scene.



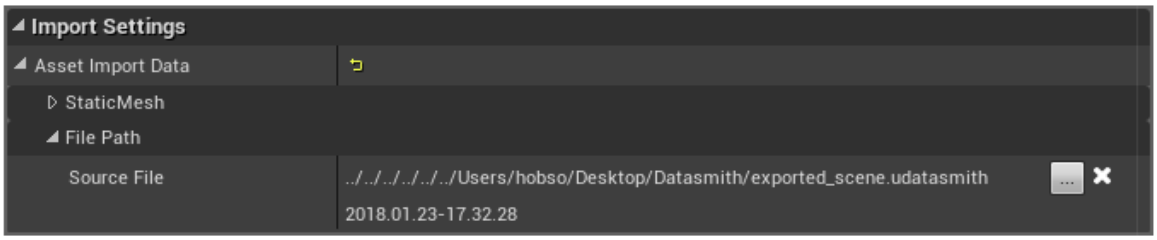
The reimport process will import any newly created or updated geometry, materials, textures, lights, or cameras. However, it will not remove any of those items if they already exist in UE4 (in the Content Browser or the Level) but have been removed from your updated Datasmith file. This is a current limitation that may be addressed in a future release.

Individual Asset Reimport

The individual assets that make up your Datasmith Scene are tracked with reference to the source file's path, just like other assets that are imported into Unreal Engine 4. It enables you to reimport a single Static Mesh or even just a Material that has been updated from your Unreal Datasmith (*.UDATASMITH) file, leaving the other Actors already in your level and Content Browser unaffected.

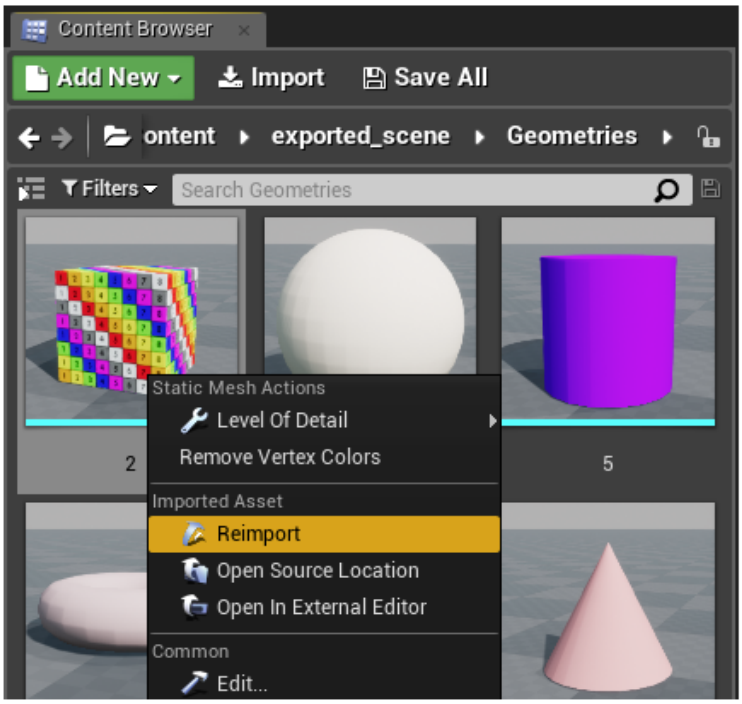
> Imported CAD data does not store the file path for individual assets. You can only do a full scene reimport. Additionally, you cannot set a source file path to these individual assets since CAD and Unreal Datasmith file is only recognized through the Datasmith Importer for UE4, which will cause the reimport process to be broken otherwise.

In the Static Mesh Editor under the Details panel, the source file path is stored under **Asset Import Data**, like all other Actors:

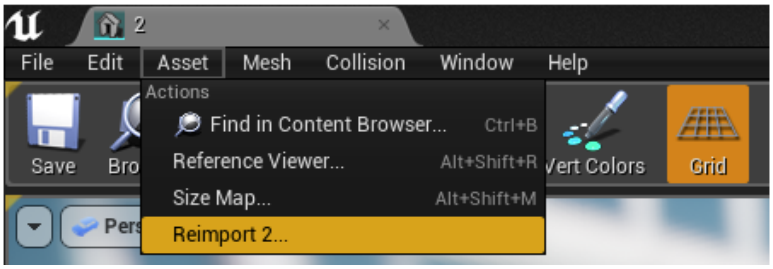


To reimport individual assets, use one of the following ways to do so:

- Use the right-click Context Menu to select **Reimport** for Static Meshes and Textures or **Reimport Datasmith Material** when selecting an asset in the Content Browser:



- In the Static Mesh or Texture Editor, use the menu to select **Asset Reimport [Name of asset]**.



Limitations and Known Issues

Below are some limitations of using the reimport process with Datasmith along with some known issues that may be resolved in a future release of the Datasmith plugin:

Limitations

- Datasmith is considered an "Import into Level" workflow. The ability to drag and drop or import CAD and *.UDATASMITH files directly into the Content Browser has been disabled. See the Datasmith Scenes section of this page for additional details.
- For CAD files, reimporting individual assets is not possible since they don't have a source file path to reference. Instead , you have to reimport the full Datasmith scene.

Known Issues:

- Changing the source file path for a Datasmith Scene asset or individual assets will break its ability to reimport the content. It is recommended not to change this path.
- Reimporting Rhino 3D (*.3dm) files create duplicate Actors in your UE4 level, the reason being that unique IDs for objects is not persistent between files. Options are being investigated to solve this issue.
- Selecting any Actor(s) that exists in the level but not the Content Browser (cameras, lights, or a selection of multiple Actors) is not currently possible. It is suggested to do a full scene reimport to capture any updates for positions or additions with these types of assets. See the Full Scene Reimport section of this page for additional details.
- Using Unreal Engine's Auto Reimport functionality to monitor files that are updated and automatically reimport their content is currently unstable for *.UDATASMITH file exports.
- The Datasmith Scene is an "additive" workflow where additional and changed assets will be handled by the Datasmith reimport workflow, but deleted objects won't be. Below are two examples:
    - If you delete an object in your Datasmith or CAD file and attempt to reimport, the assets in the Content Browser and Actors in the level will still exist in UE4.
    - If you delete an object in UE4 (from the Content Browser or in the level) and reimport your Datasmith or CAD file, the deleted assets and Actors will reappear.

Datasmith Entitlements

The Unreal Datasmith and CAD Importers for UE4 are attached to your Epic Games account and require proper entitlements to use. Before importing any files, the plugins will validate that your account's entitlements are in place and check if you are allowed to import

these file types. To verify this status, you must have signed into the Epic Games Launcher at least once to be able to import them. If you have not yet signed in, you will see one of the following sign in windows when trying to import your Datasmith files:

Once the entitlement check has succeeded, the Datasmith Import Options dialog will be enabled to allow content import.

Troubleshooting your Entitlement Check

In some instances, some issues have caused the entitlement check to fail. Below are a few things you can do to troubleshoot your entitlements check:

- Check that your firewall or anti-virus software enables appropriate permissions for Unreal. You can even disable it for a moment to see if this is the case before setting up these permissions.
- Some have reported that disabling Akamai NetSessions Client has resolved their entitlements check.

These issues are being continually investigated. If you encounter an issue that is not solved by performing one of the actions above, submit a report to Unreal Studio Help page.