UDK to UE4 T3D Tool Documentation

Table of Contents

Introduction

The UDK to UE4 T3D tool takes input generated by Unreal engine 3 (UDK) ’s T3D scripting format and reformats and converts the data so that it can be pasted into Unreal Engine 4.
This allows you to port the actors inside of a UE3 level into an UE4 level.
T3D from Unreal 3 if pasted into Unreal 4 simply causes Unreal 4 to crash.

Free vs Paid

The Free version only supports static meshes, the paid version has the following advantages:

1. By providing the path to your UE4 "Content" folder, the tool will dynamically search through the assets and match by name and use the found path instead of converting the UDK path. This allows you to have the assets in any location, the only requirement is that they have the same name

2. Support for the following actors added:

- Lights (All types except for Skylights and pickup lights)
- Kactors
- InterpActors
- Skeletal Mesh Actors
- Interactive Foliage Actors
- Fractured static meshes
- Apex Destructible actors
- Cameras
- PlayerStarts
- Exponential Height fog
- Particles
- Decals
- AmbientSound/AmbientSoundSimple

Supported Actors

The free version (1.x) supports only static meshes. In addition to the position, rotation, and scale of the actor, it will also port over the Path to static mesh, any and all Actor Material Overrides, the Light map UV Override settings, and painted vertex colors

Tutorial

1. If you do not have access to the source files themselves, Export all Meshes and Textures out of Unreal Engine 3.
2. Import assets into Unreal Engine 4. It is required that every asset have the same name as it did in UE3 and located in the same folder structure. You'll need to create a folder for each package and copy its directory structure.
3. You will have to recreate & assign materials to your meshes. (May be possible in the future to write a converter for materials)
4. In UE3, select the actors you want to port over, and Ctrl + C to copy them.
5. Paste the output into the tool and convert it.
6. Copy the converted script to clipboard and paste it into Unreal Engine 4.

Notes

- You can use the "Asset Path" text box to specify a subfolder of where the assets are located.
- Unreal Engine 4 changed the Unit scaling. One Unreal Unit (UU) is now 1cm, where it was 2cm in UE3. This means that things ported over to UE4 are going to be half the size they should be. The tool provides a way to fix this by checking the "Multiply Locations/Scale by 2" options. Alternatively you can take every actor ported into UE4 and parent it to something, then changing the scale of the parent to 2.
- **A note on Vertex Colors**: While the tool does support the porting over of Static Mesh vertex colors. There is an issue that may prevent them from working.
  The import/export process of meshes is sometimes not 1 to 1. A mesh that is re-exported out of UDK may not have the same structure as it actually does inside UDK. This means that although the Vertex color data is there the T3D script and is converted, it won't appear up on the UE4 mesh because technically, it's a different mesh now.
  In order to fix this issue, You can try re-importing the mesh back into UDK. This means both engines have the exact same data to work with, then copying the actors.
- Since the "Drawscale" value was removed in UE4 and now there's only Drawscale3D. The Drawscale value is applied to the Drawscale3D values by multiplying them by the Drawscale. So that they match their correct size.

Contact

Email: matt.swanton3d@gmail.com

Website: www.mattswanton.com

Unreal Forum topic:
https://forums.unrealengine.com/showthread.php?3172-Tool-UDK-T3D-to-Unreal-4-T3D-Tool

Change list:

Version 1.0 – Initial release that supported only static meshes

Version 1.1 – bug fixes, built-in Scaling assets by 2, Static mesh materials & Override

Version 1.2 – Static Mesh vertex color support

Version 2.0 - Lightmap UVsupport, Lights, Cameras, Decals, Kactors, Skeletal Mesh actors, Interp Actors, Particles, Audio and Exponential height fog actors all supported. Dynamic linking of assets.