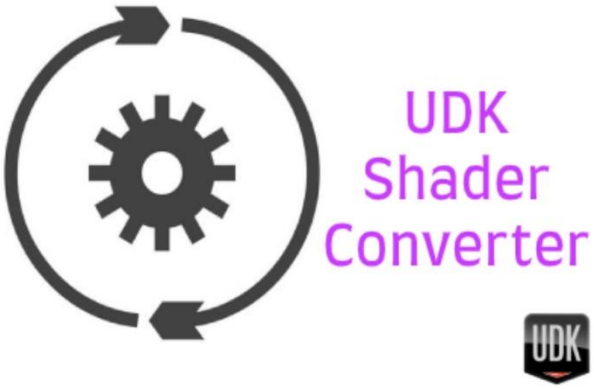


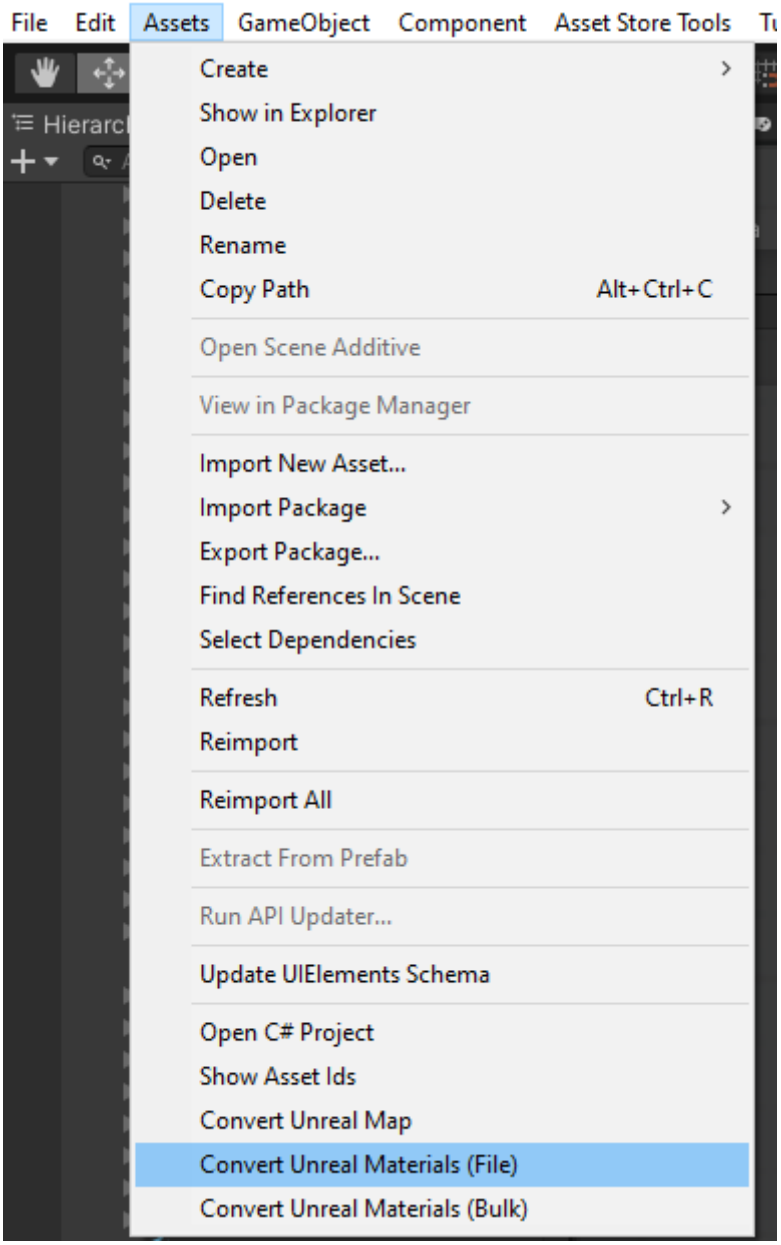
Introduction

This tool is used to convert UDK Shaders also known as materials to Amplify Shader Editor in Unity. The aim is to resemble the functionality of the UDK shader as closely as possible, this is not always possible for larger complex shader however. But using this tool in those cases will help reduce migration effort.

At present the tool also convert UE4 Shaders but not fully, support for this is actively being developed and will be released in future updates.

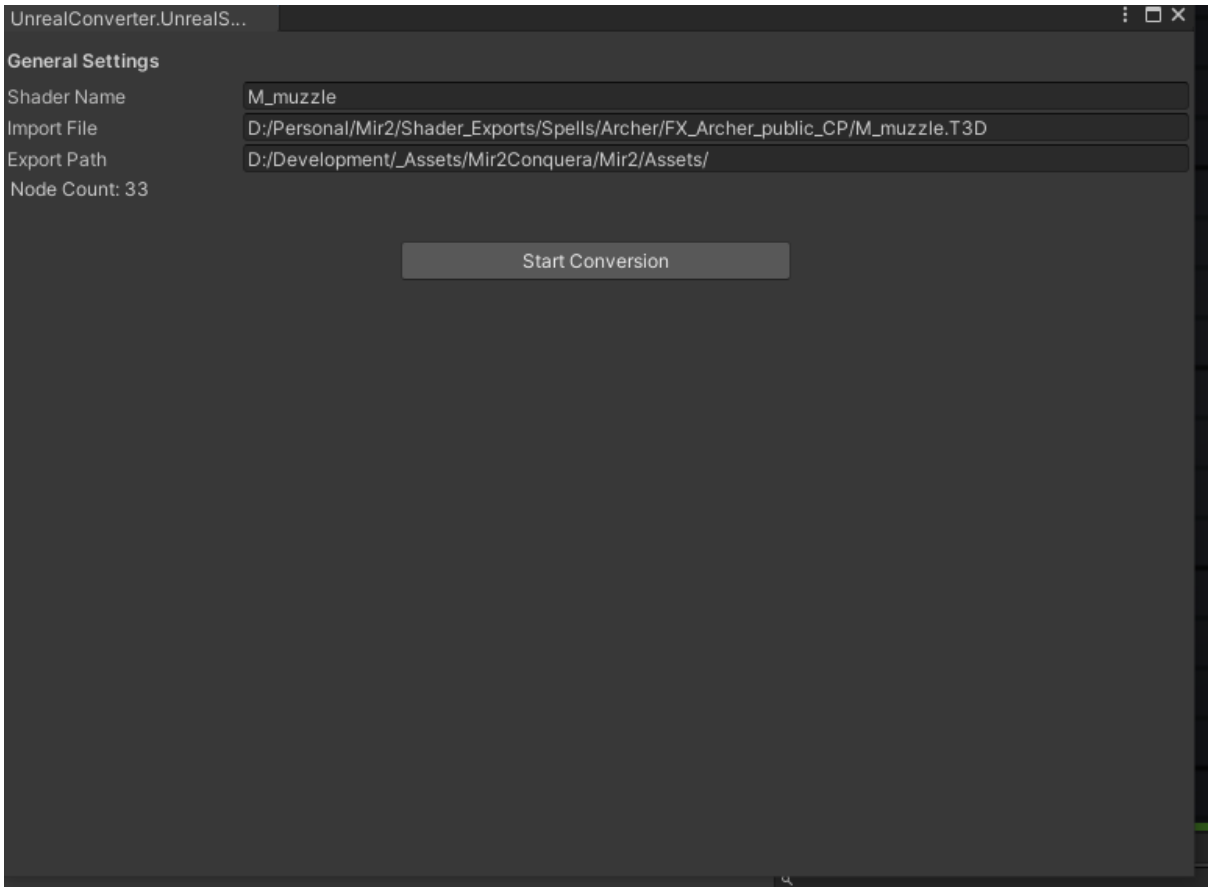
Interface

There will be two new menu items in the Assets menu.



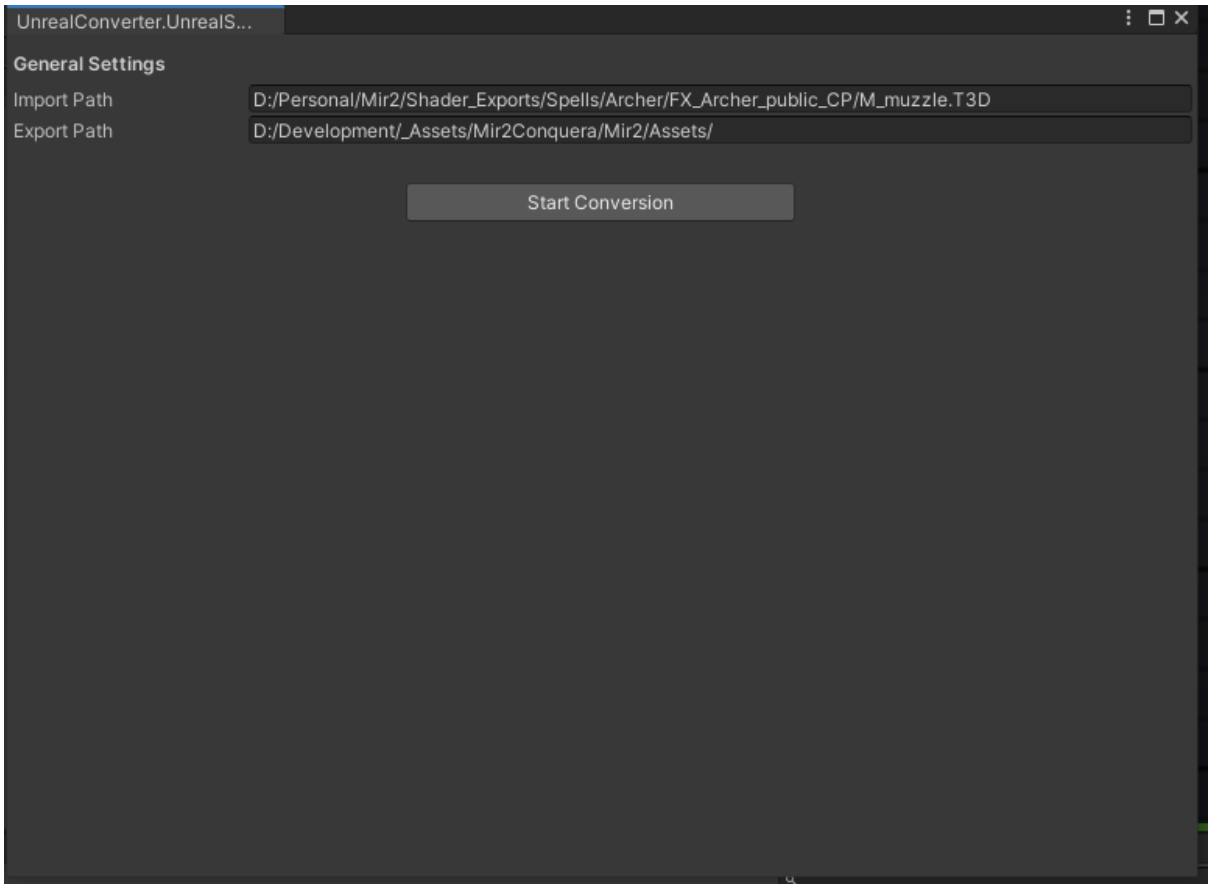
Item	Purpose
Convert Unreal Materials (File)	Open Converter for Single File Conversion
Convert Unreal Materials (Bulk)	Open Converter for Bulk File Conversion

Convert Unreal Materials (File)



Item	Purpose
Shader Name	Name of shader once imported
Import File	File Path for the T3D shader file
Export Path	Unity Path to Save the Shader
Node Count	Number of nodes being converted
Start Conversion	Executes Conversion

Convert Unreal Materials (Bulk)

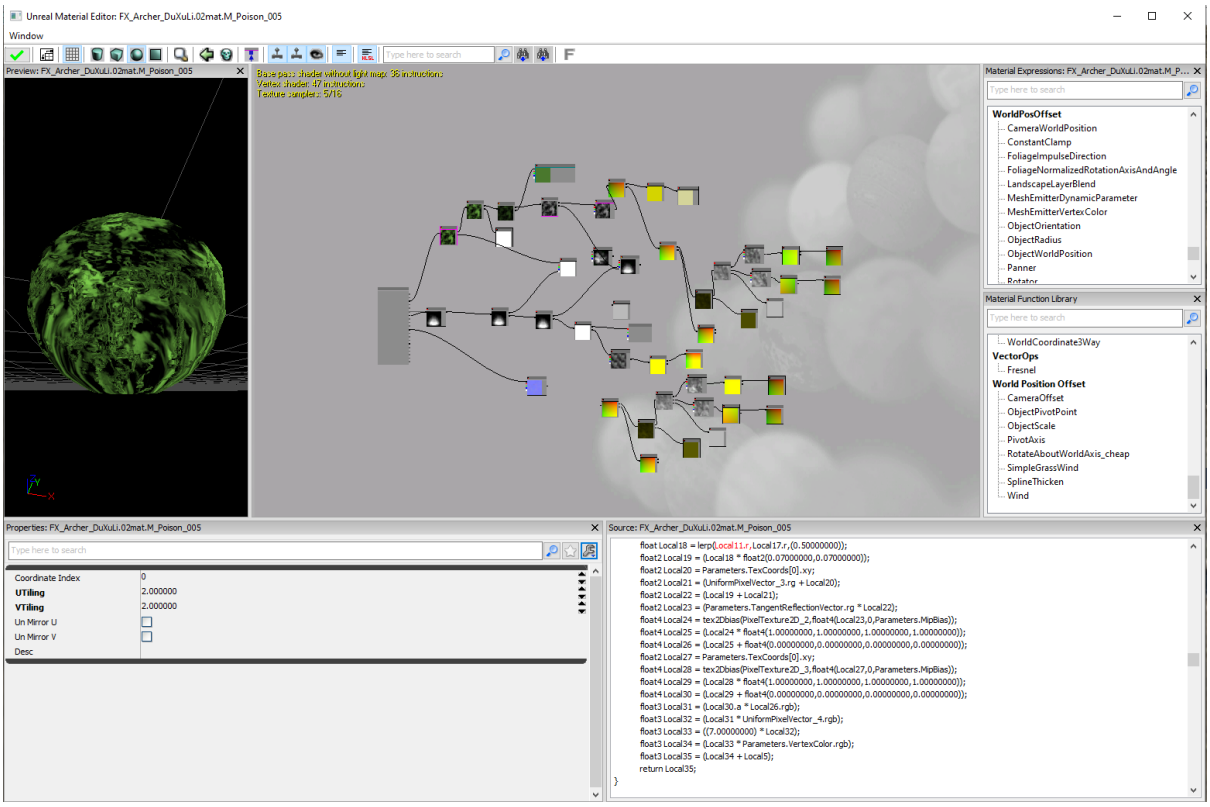


Item	Purpose
Import Path	File Path for the T3D shader files
Export Path	Unity Path to Save the Shaders

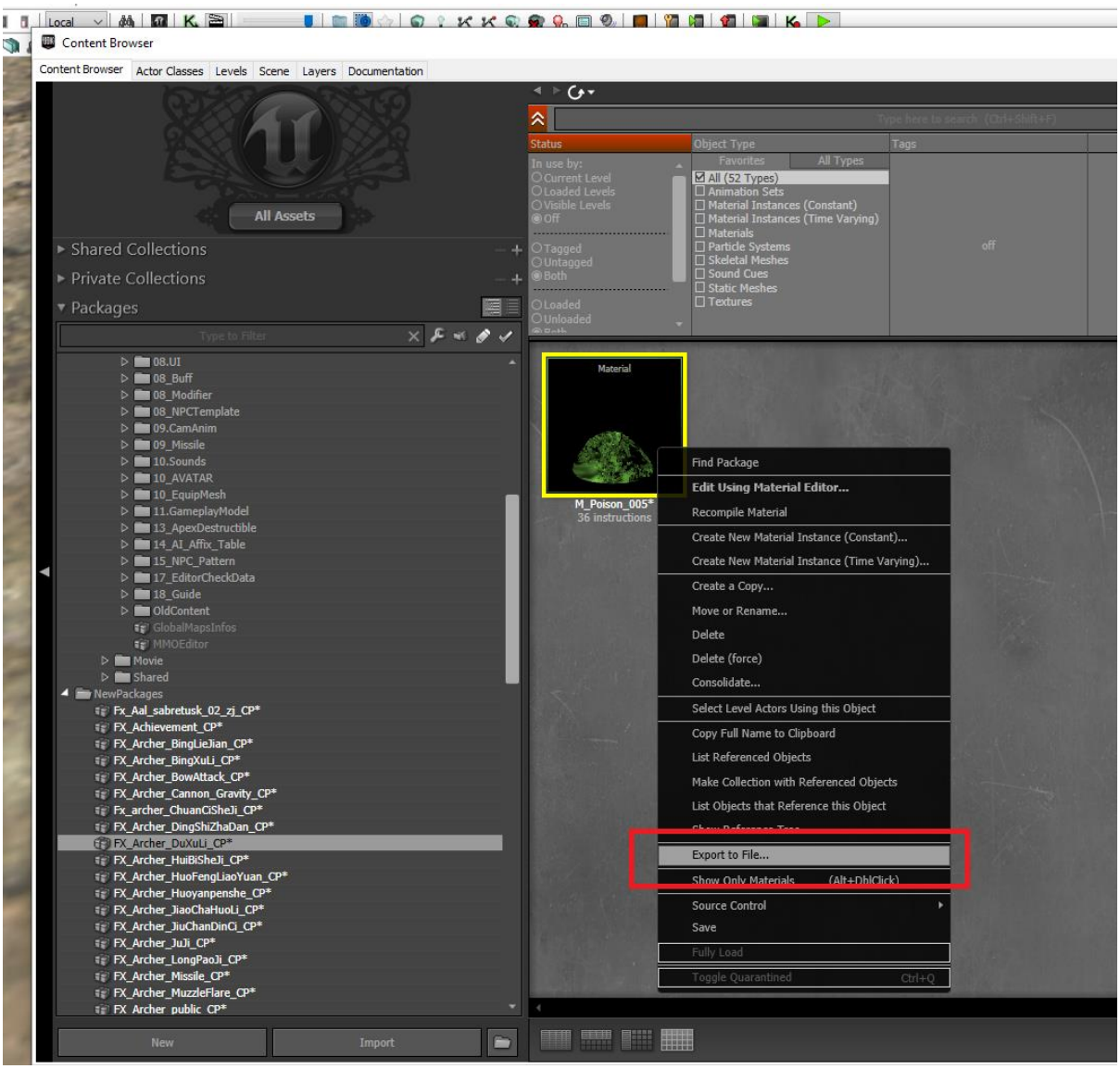
Usage Guide

File Conversion

1. The first step is to identify the Material in Unreal you want to convert, for this example we will use a medium complex shader with many nodes.



2. Now you need to export the Material, in the [**Content Browser**] for Unreal, Right Click the Material and Export to File, **Save as a T3D file**.

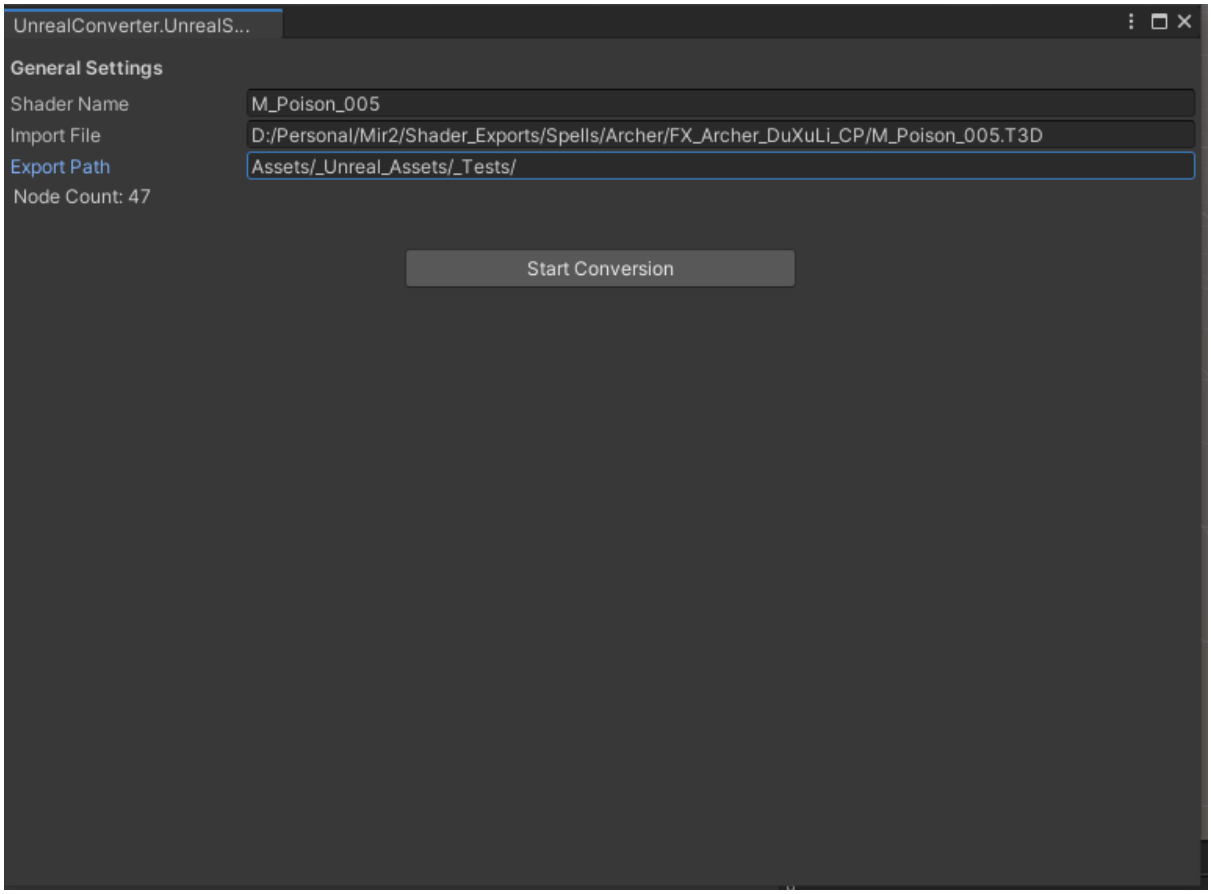


3. With the Material Exported we can now Convert the Material, Open Unity and go to in the menu "**Assets > Convert Unreal Materials (File)**".

4. You will be prompted to Select a file, Select the T3D file just exported from Unreal.

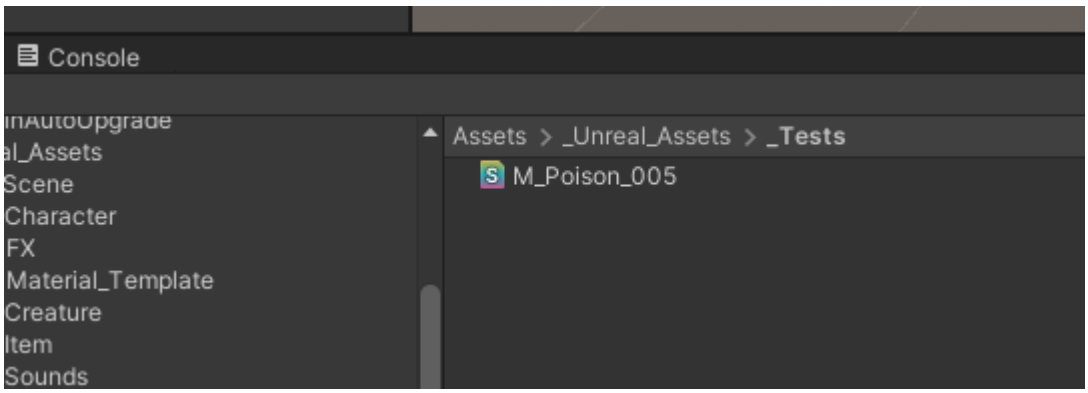
5. The Conversion Interface will be shown as below, change the Name and Export Path if needed.

Ensure the Export Path ends with a FORWARD SLASH /

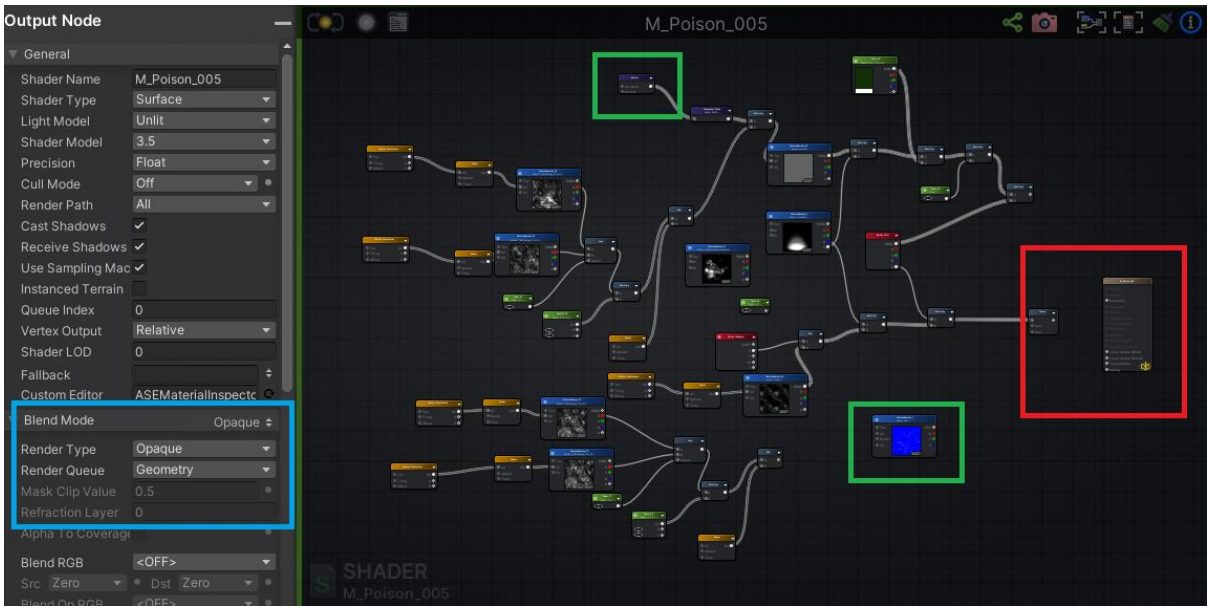


6. Press [**Start Conversion**] when ready.

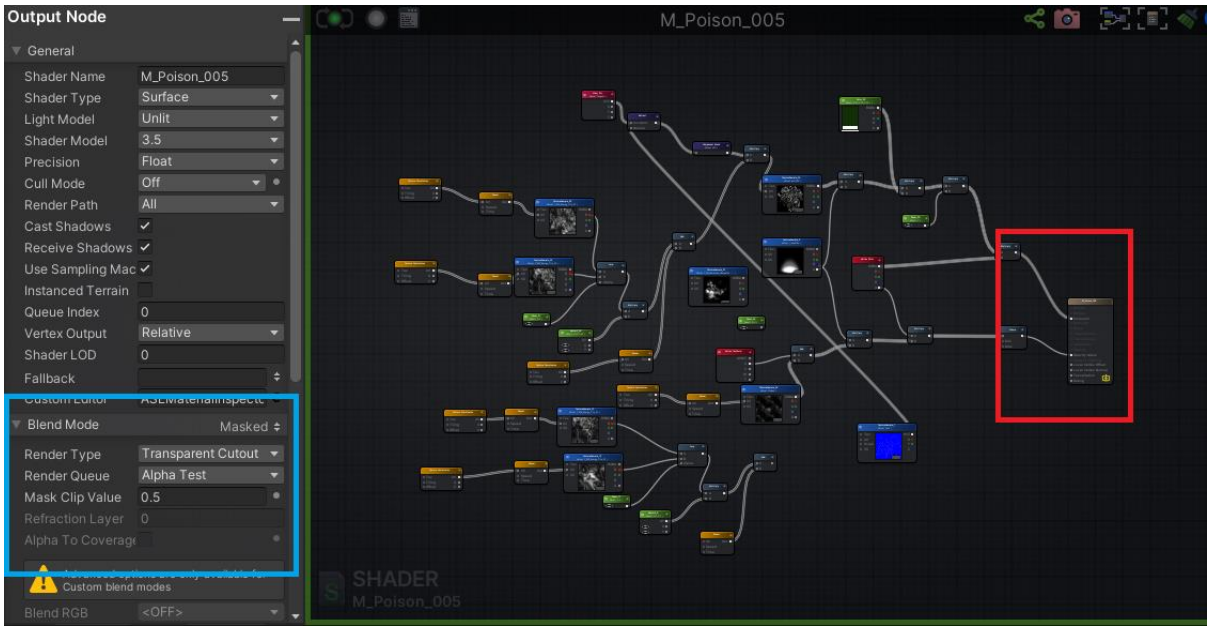
7. After Conversion you will see in the Export Path the new Shader, this will not be ready yet we must open first and fix any noticeable issues and connect the output Nodes.



8. When you open the output nodes will not be connected, this is intentional as we must adjust differently to match the Unreal Output. Also, in Green boxes you will see we have a Reflect Node and a Normal Node, the fix for this will be in the Manual Fixes Section. In blue we can also adjust the Blend Modes.



9. In Unreal this Shader is a Masked Unlit Shader, we already have automatically set the Type to Unlit but lets set the Blend Mode to Masked to match Unreal, and then connect our output nodes and Save.



10. Once Saved you can create a material and assign to see the results, adjust settings and nodes if you want a different result. In this case the result is below compared to original Unreal result.

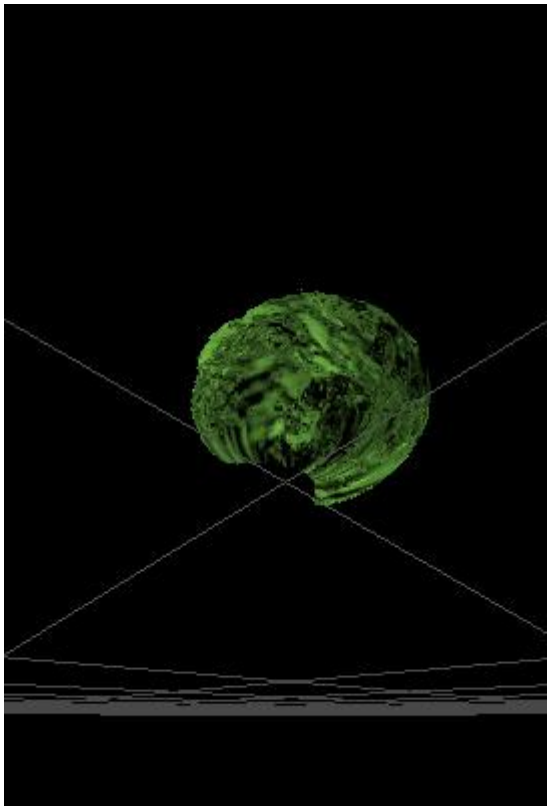


Figure 1 Unreal

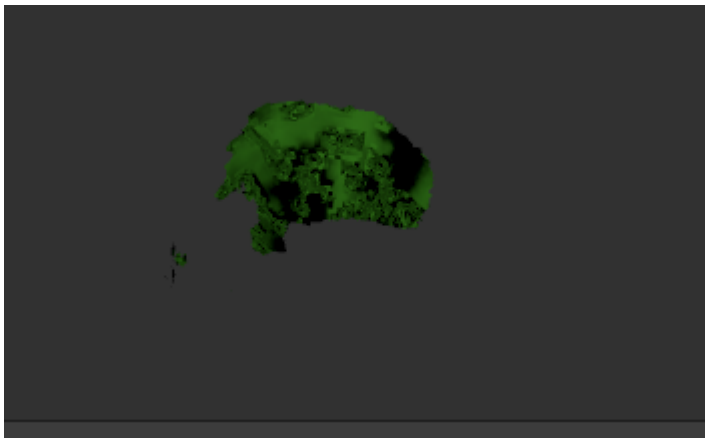


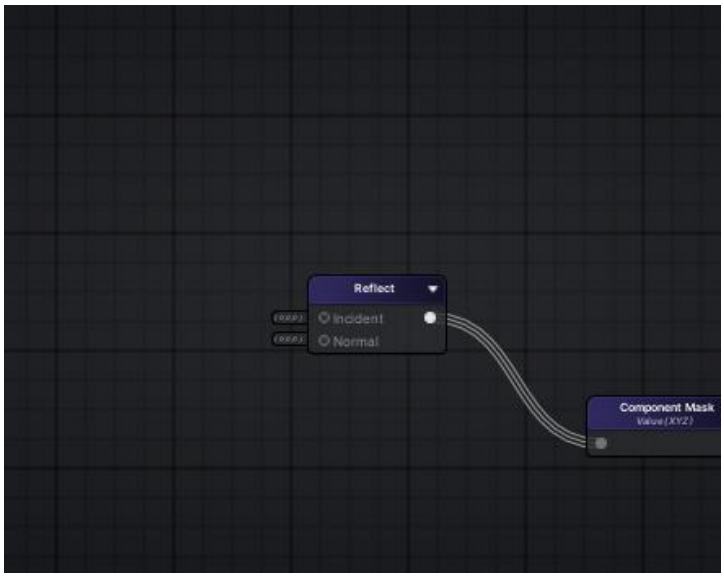
Figure 2 Unity

Manual Fixes

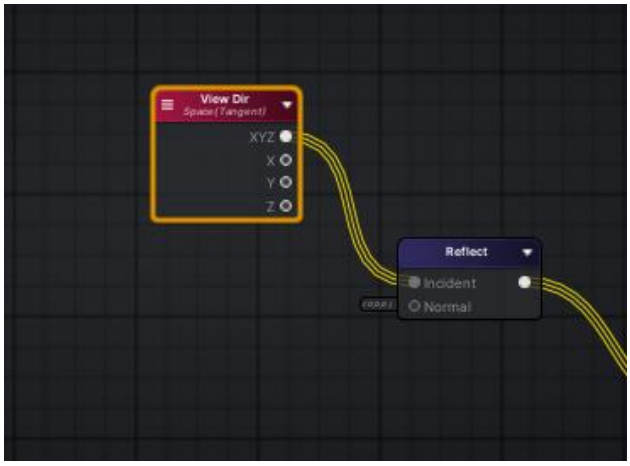
There are some manual fixes you must correct after conversions if they are applicable, this is due to how shaders in Unreal UDK and Unity work. Here will document all known fixes and when to apply them.

Reflection Nodes

Reflection Nodes after Conversion look like this as there is no Incident port in Unreal or Normal port.



To resolve in most cases connect a View Dir node and Normal if required.



Panner and Rotator Nodes

Sometimes these nodes don’t have UV attached in Unreal so conversion doesn’t put them, but they are required.