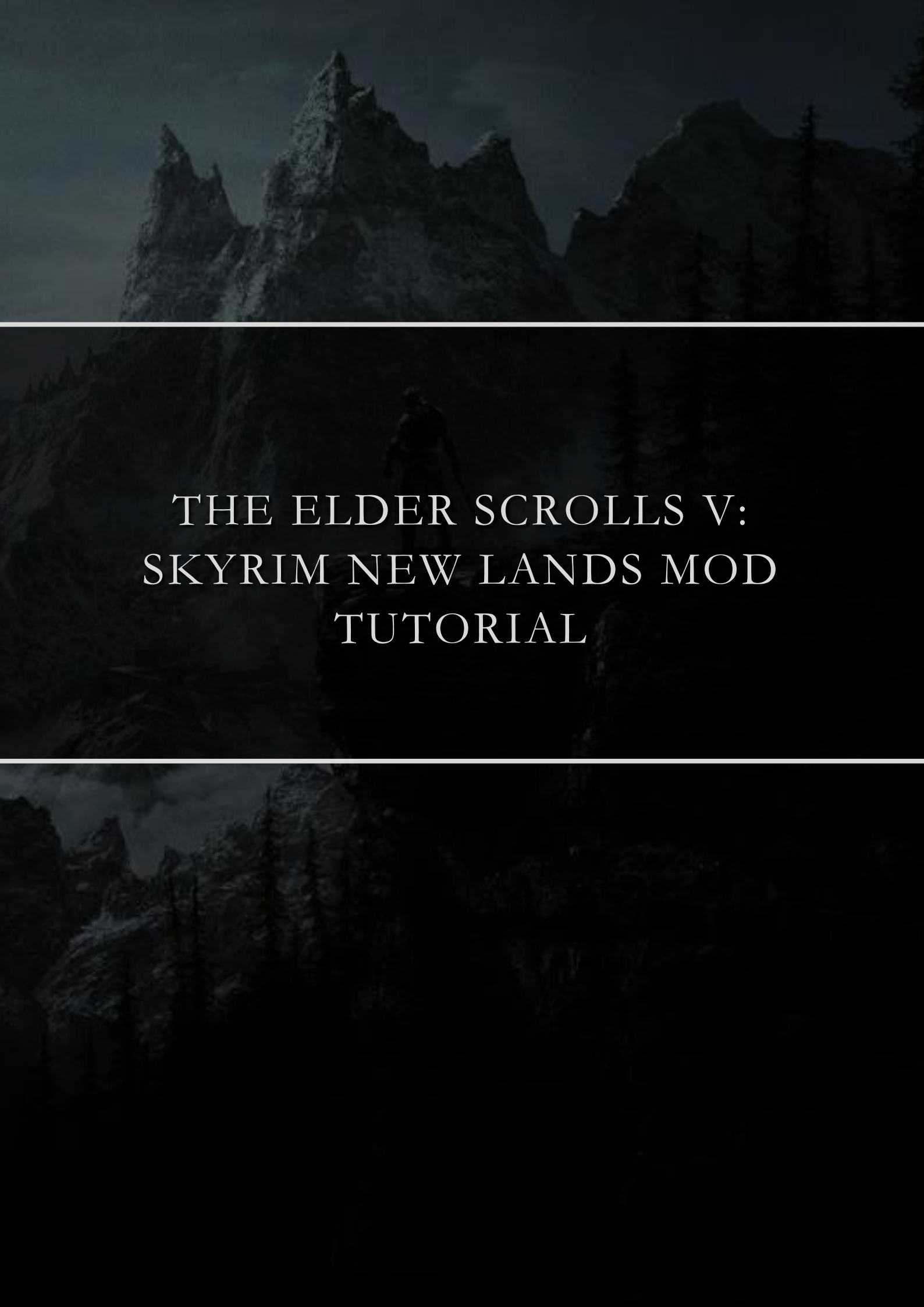


THE ELDER SCROLLS V: SKYRIM NEW LANDS MOD TUTORIAL

REVISION HISTORY

Version:	Date:	Description:
1.0	19/01/2022	Initial release.

SUPPORT

If you found this document useful, you can follow me on Reddit and YouTube for updates.





PREFACE

Initially I wrote up this guide for personal use because I'm an idiot and kept forgetting how to do specific things. Lately I've been adding to it to cover as many topics as possible relevant to making a typical new lands mod for The Elder Scrolls V: Skyrim.

I'm Jonx0r and I made the Wyrmstooth mod which is available for download on Nexusmods, Steam, and Bethesda.net. As of this writing I'd estimate that I've spent roughly 2000 hours working on the mod since May 2012.

This tutorial outlines the way I approached making a new lands mod, which, as a disclaimer, may not necessarily be the best way to do it anymore. When I started modding in 2012 there wasn't much out there to follow in terms of guides and tutorials, so a lot of this was learned through trial and error. Mostly error. I'm far from being some sort of mod expert and this document is a work-in-progress so if you spot any mistakes, know a better way to do something, or would like me to cover additional topics please let me know. I've tried to make the instructions as easy and as straightforward to follow as I could, but let me know if I need to reword or clarify anything. Maybe at some later stage I'll consider recording this tutorial as a series of YouTube videos but right now I'd just like to make sure I have as many relevant topics written down first.

Anyway, with that said, I hope someone finds this at least remotely useful.

- Jonx0r



CONTENTS

Preface 2
Introduction 8
Before we begin 9

Chapter 1: World space 13
Creating a new world space 14
Creating a height-map using World Machine 19
Importing a height-map with TESAnnwyn 45
Sculpting a landscape with the landscape edit tool 51
Adding landscape textures 63
Creating tileable landscape textures 74
Adding lakes and rivers 101
Setting up water flow and currents 112
Adding regions 128
Region generated textures and objects 142
Setting up locations 183
Setting up encounter zones 192
Adding navmesh 196
Adding occlusion geometry 218
Cluttering a world space 225

Chapter 2: Markers 244
Adding idle markers 245
Creating a custom idle marker 262
Adding map markers 266
Adding common radiant markers 272

Chapter 3: Sounds and music 289
Adding sounds 290
Adding acoustic spaces 315
Adding music 322

Chapter 4: NPCs 347
Creating an NPC 348
Adding and creating AI packages 376
Making a merchant 392
Making an innkeeper 406
Making a bard 414
Making a follower 429
Setting up a home owner 441
Setting up a boss and making a location clearable 465
Adding ambient creatures 473

Chapter 5: Random encounters 480
Adding random encounter triggers 481
Creating our own list of random encounters in the story manager 489
Creating new random encounters 511

Chapter 6: Quests, dialogue and voice acting 535
The basics of setting up quests 536
Adding dialogue to a quest 563
Chaining dialogue topics together 588
Blocking branches 606
Generating an .SEQ file 615
Adding voice acting and lip synching 619
Adding hello, combat, crime and other incidental dialogue 635
Adding scenes 677
Using talking activators 724
Creating an MCM 736

Chapter 7: Building LOD and generating files 764
Building land LOD using Oscape 765
Building object LOD using xEdit 777
Building tree LOD using xEdit 783
Final adjustments to world space settings 790
Generating max height data 817
Generating TVDT occlusion 820
Generating grass .gid files 825

Chapter 8: Asset pipeline 827
Importing assets from Skyrim in 3ds Max 828
Exporting assets from SpeedTree to 3ds Max 832
Fixing transparency in NifSkope 844
Exporting assets from Zbrush to Substance Painter to 3ds Max 852
Exporting assets from 3ds Max to Skyrim 911

Chapter 9: Motion-capture animation 928
Setting up the character skeleton in 3ds Max 929
Constraining the skeleton to a biped and importing mocap data 941
Exporting animation from 3ds Max with Havok Content Tools 967

Chapter 10: Speech synthesis 992
Introduction to speech synthesis 993
Preparing a dataset using voice acting from The Elder Scrolls V: Skyrim 996
Mounting your Google Drive storage in Google Colab 1021
Training a Tacotron model in Google Colab 1028
Training a WaveGlow model in Google Colab 1033
Checking progress in Tensorboard 1040
Synthesizing audio from the models we've trained 1044
Improving audio quality in Audacity 1050
A couple tips and tricks 1056

Chapter 11: Other tasks 1058
Unpacking the BSA archives 1059
Adding additional master files 1062
Removing accidental edits 1065
Adding texture paths to 3ds Max 1069
Adding texture paths to NifSkope 1074
Optimizing interior cells with roombounds 1082
Version control and merging changes 1095
Cleaning your mod in xEdit 1129
Checking for errors in xEdit 1131
ESM-ifying your mod in xEdit 1134
Generating papyrus debugging trace 1137
Creating a .bik video file 1140

Chapter 12: Packaging the mod for release 1146
Porting your mod to Skyrim Special Edition 1147
Packaging your mod for Legendary Edition 1153
Packaging your mod for Special Edition 1160
Packaging your mod for Nintendo Switch 1170

Index 1174

INTRODUCTION

In this tutorial I'll be covering the general process of creating a new lands mod using the Creation Kit and various other tools.

I'll be using the Creation Kit for the Legendary Edition of Skyrim which is available from Steam, but you can follow along with the Creation Kit for Special Edition as well. You might also find some of the information or workflows useful for other Creation Engine games such as Fallout 4.

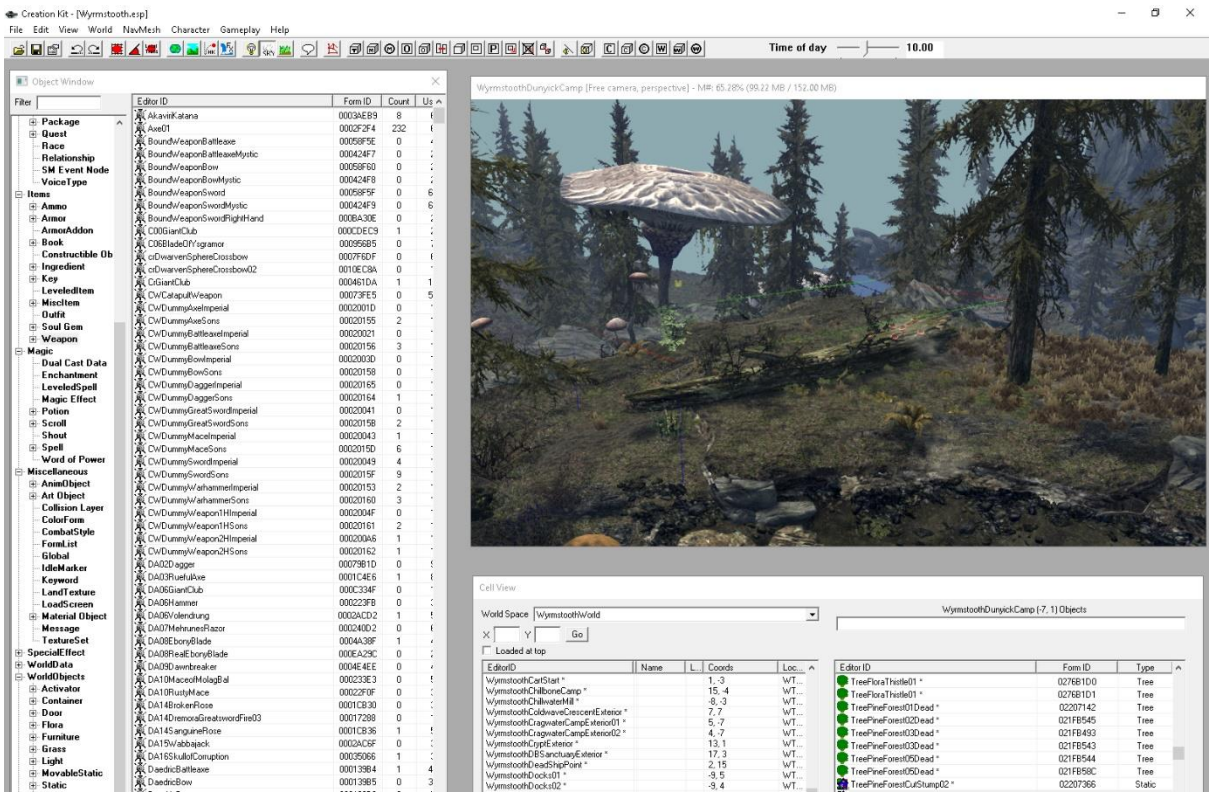


Figure 1 - A custom world space loaded in the Creation Kit.

While there are a lot of topics to cover in this guide, it's far from being an exhaustive look at all facets of modding. For the sake of brevity I'm only going to be covering the topics relevant to the development of a new lands mod and environment art. I'll be referring to Wyrmstooth in numerous examples to show practical implementations of what I'm talking about.

This tutorial assumes you have the Creation Kit already installed and are somewhat familiar with using it. There are also certain actions that may cause the Creation Kit to crash and I've tried my best to highlight the cases where that might occur.

Alright, let's get started.

BEFORE WE BEGIN

Firstly, if you plan on working with a group of developers, see the section Version Control and Merging Changes for steps on setting up version control.

This might seem a bit obvious but I figure I should mention it anyway. Before making a start on your mod, I'd recommend setting up a project folder.

This should include a copy of your mod's assets from your Skyrim\Data or Skyrim Special Edition\Data folder and **only** the assets related to your mod.

As you work on your mod and generate new files, be sure to copy them to your project folder. This will come in handy later on when you're ready to package your mod for release.

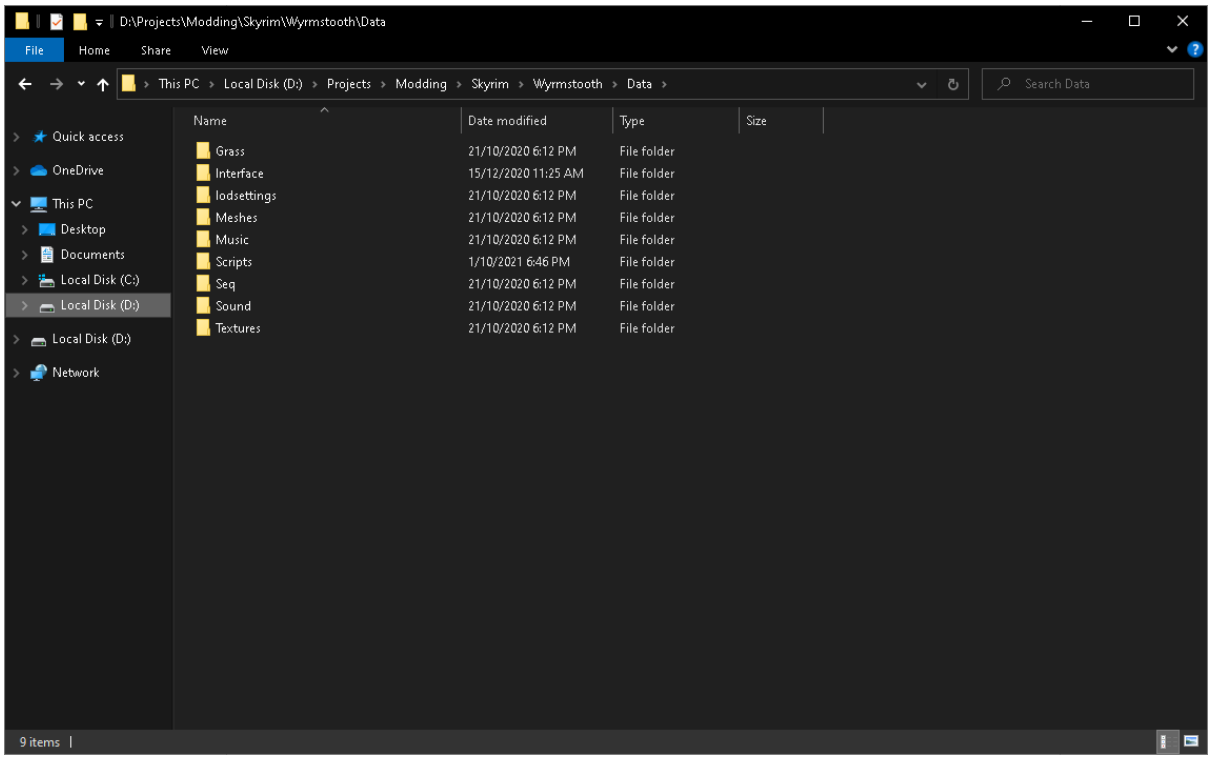


Figure 2 - Project files.

You should also keep incremental backups of your .esp or .esm file in a folder as you work on your mod in case you need to roll back to a previous version.

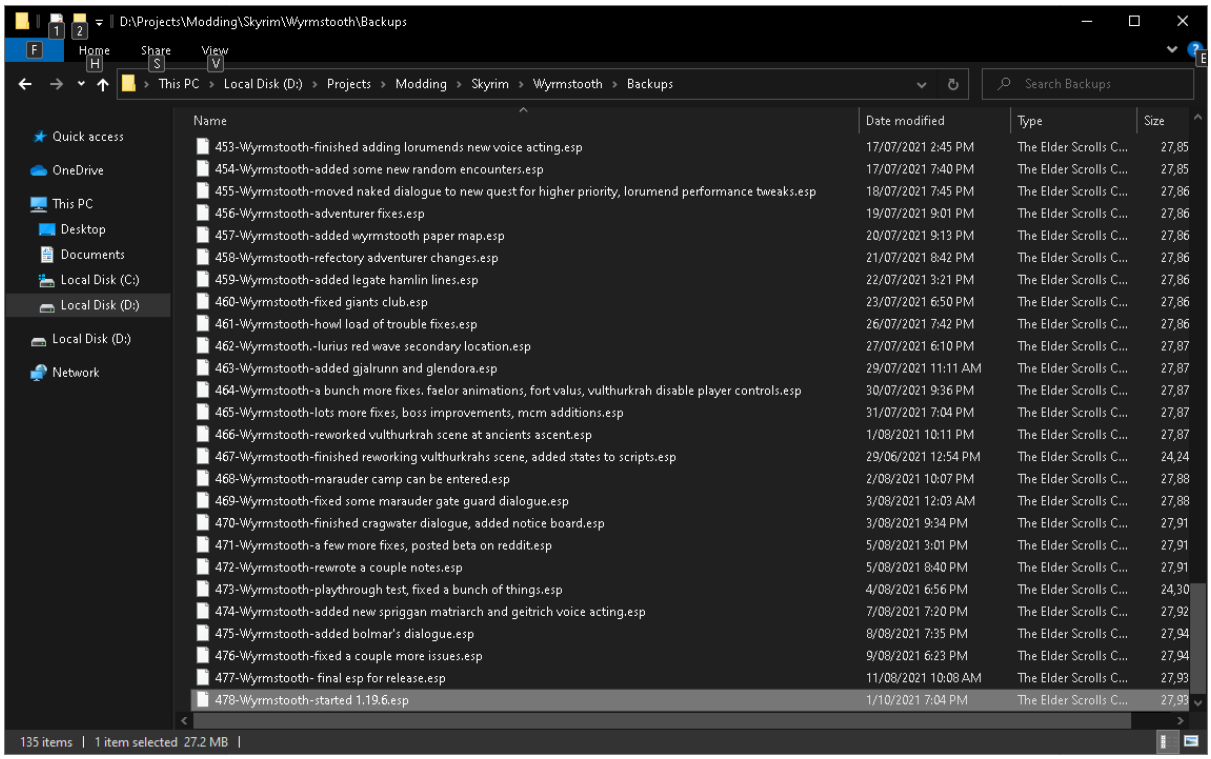


Figure 3 - Keeping incremental saves.

I'd also recommend routinely backing up your project folder to a cloud service like Google Drive in case you experience hardware failure (this did in fact happen to me once).

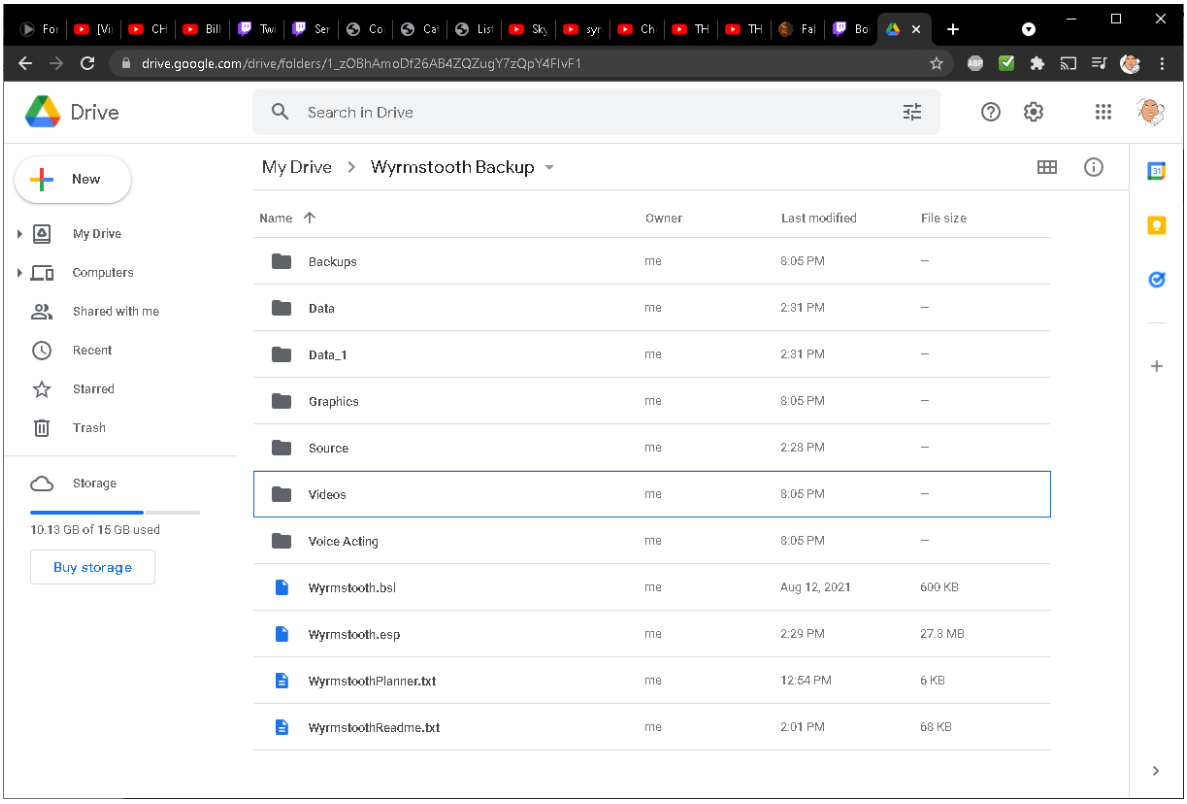


Figure 4 - Saving to the cloud.

You should also set up a unique prefix for your mod in the Creation Kit before doing anything related to scripting.

In the Creation Kit go to File > Preferences.

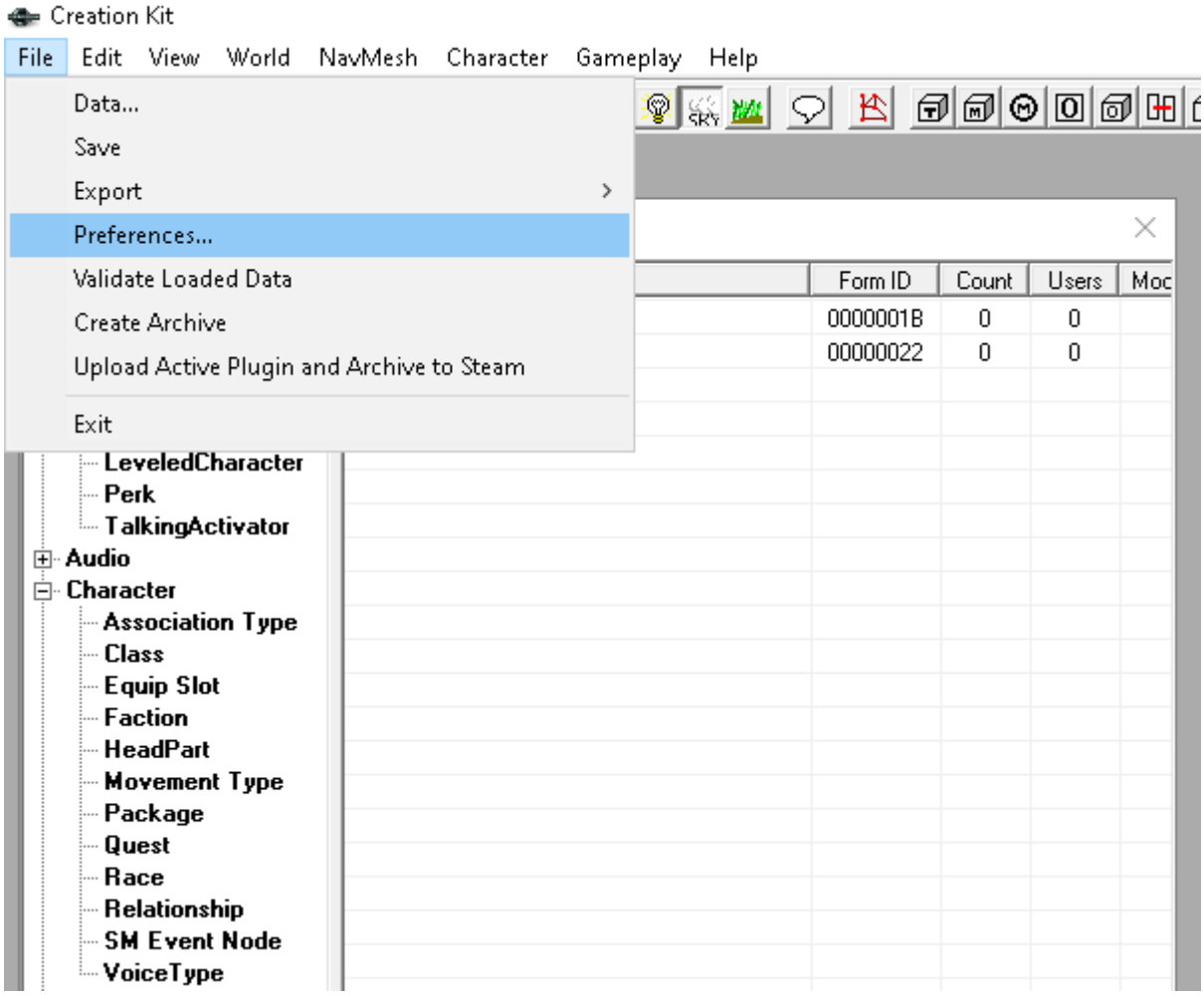


Figure 5 - File menu.

Under the Scripts tab, set an extension in the 'Fragment file prefix' field.

This prefix will be used when the Creation Kit generates papyrus fragment files.

For example, when working on Wyrmstooth I use the 'WT' prefix for just about everything.

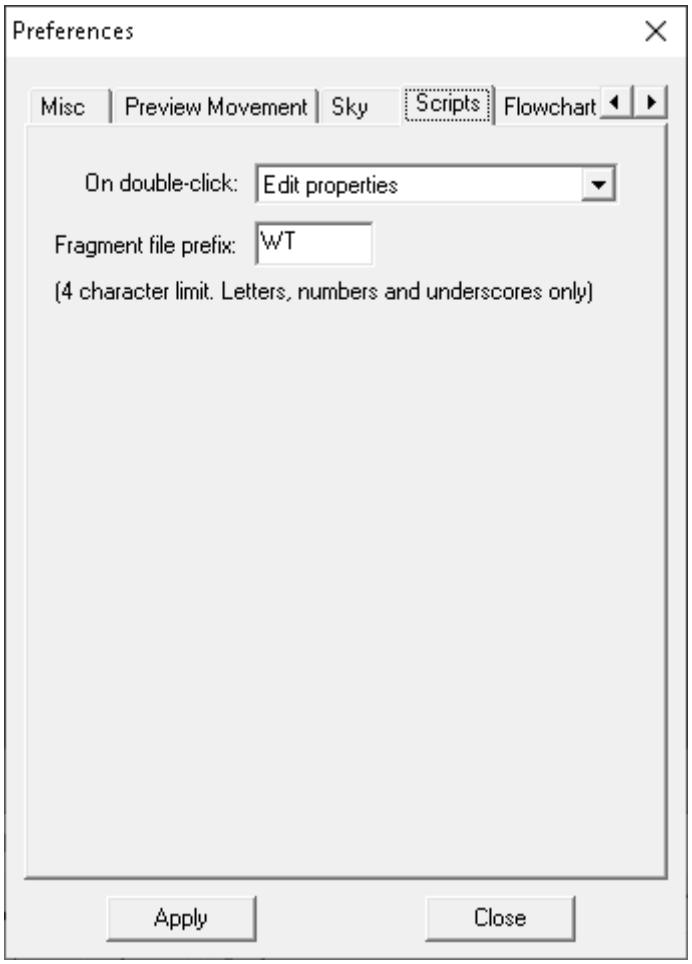


Figure 6 - Setting up an automatic prefix for papyrus fragments.

You should also use this prefix when naming assets in your mod. For example, all NPCs in Wyrmstooth are prefixed with 'WT' to differentiate them from NPCs from the base game.

Important: When working on another mod with a different prefix, you will need to change the Fragment file prefix manually.

CHAPTER 1: WORLD SPACE

CREATING A NEW WORLD SPACE

Important: Before you begin, if you plan on starting with a height-map generated by a third-party program such as World Machine, refer to the Creating a height-map with World Machine section instead.

This section will focus on creating a brand new world space **manually** in the Creation Kit. Typically this is the first thing you're going to want to do when starting on a new lands mod.

A world space is basically our overworld map. Unlike an interior cell, a world space is divided into cells to allow the game engine to load and unload assets based on the player's proximity to them.

Launch the Creation Kit. Go to File > Data, tick Skyrim.esm, click OK and let it load.

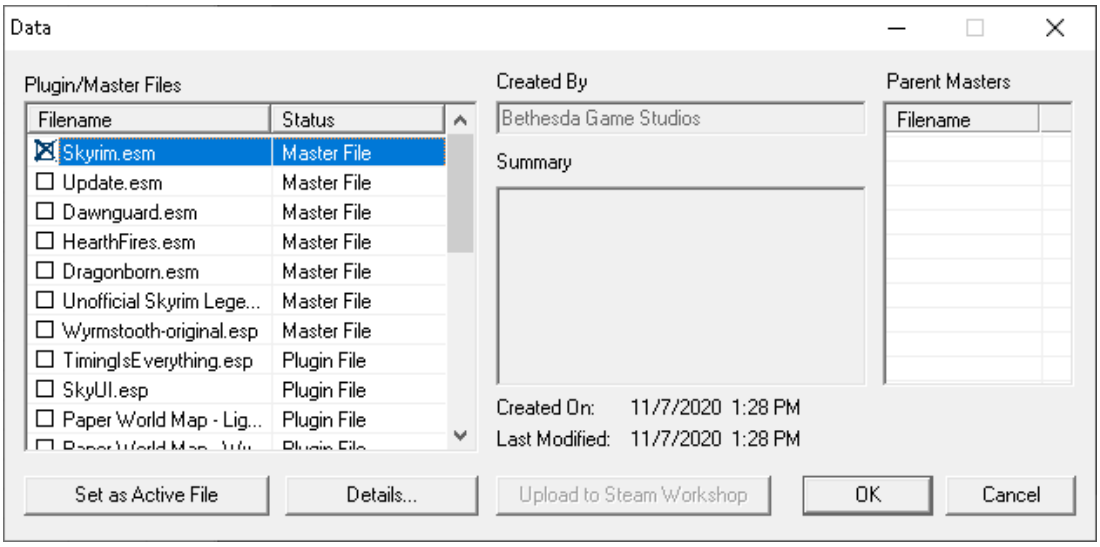


Figure 7 - Plugin/Master file loader.

Go to World > World Spaces.

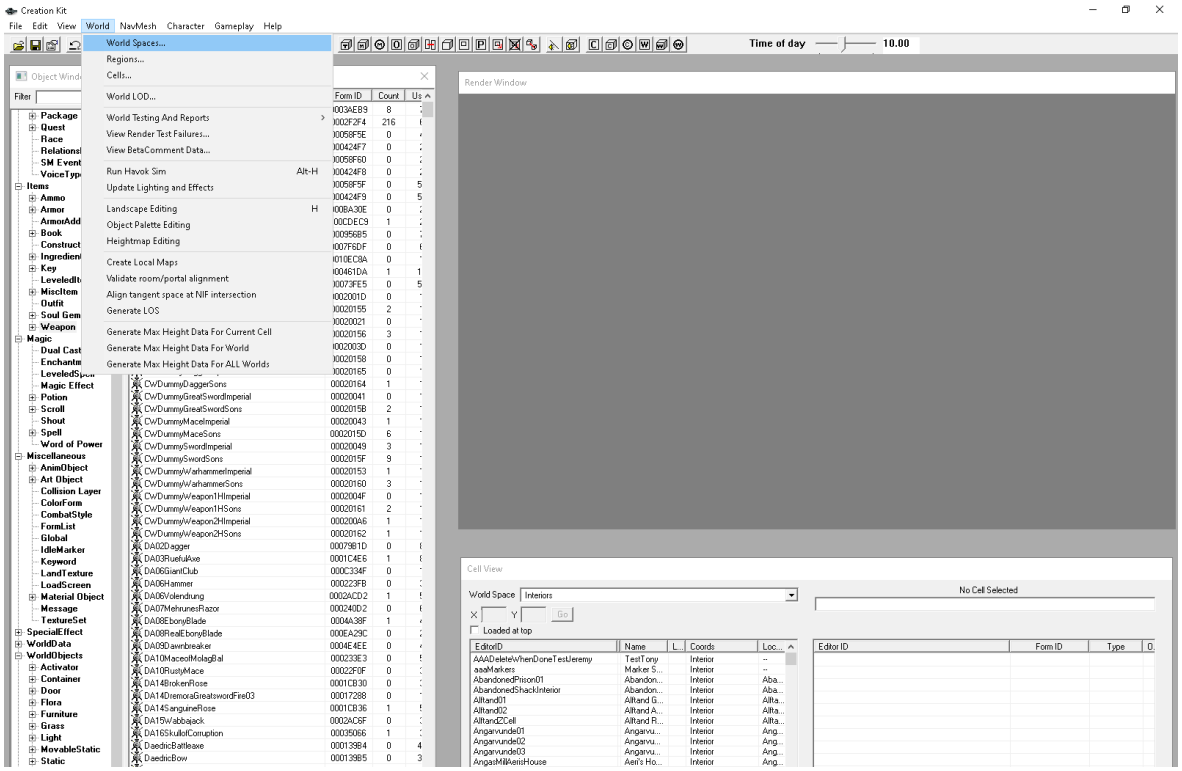


Figure 8 - World menu.

Right-click in the empty space beneath the list of world spaces and select New.

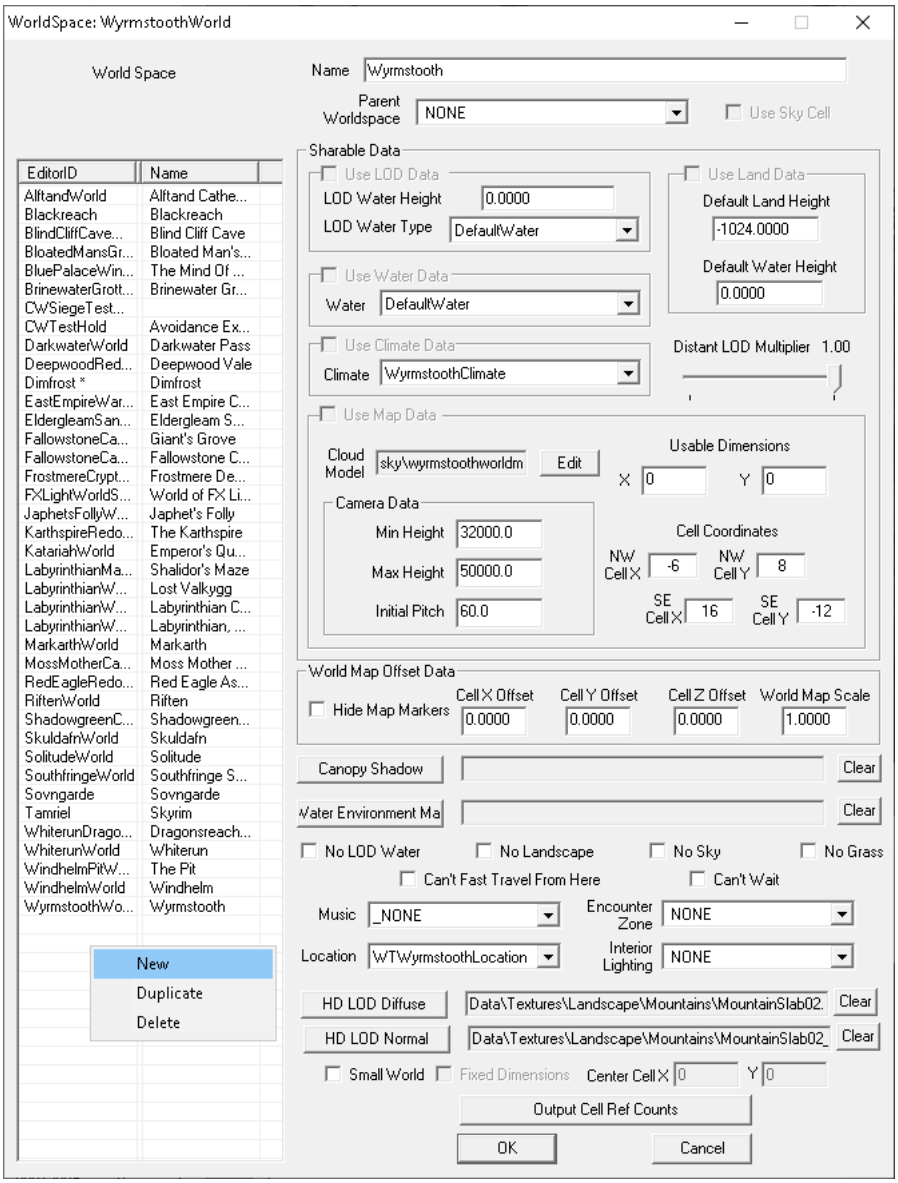


Figure 9 - World Spaces.

Enter a form ID for your new world space and click OK.

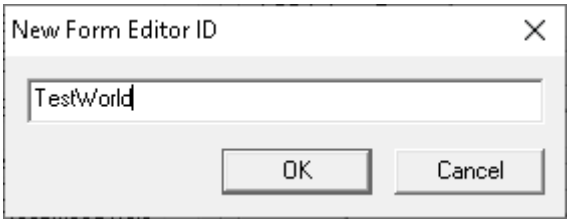


Figure 10 - World Space form ID.

Select your new world space in the list and give it a name.

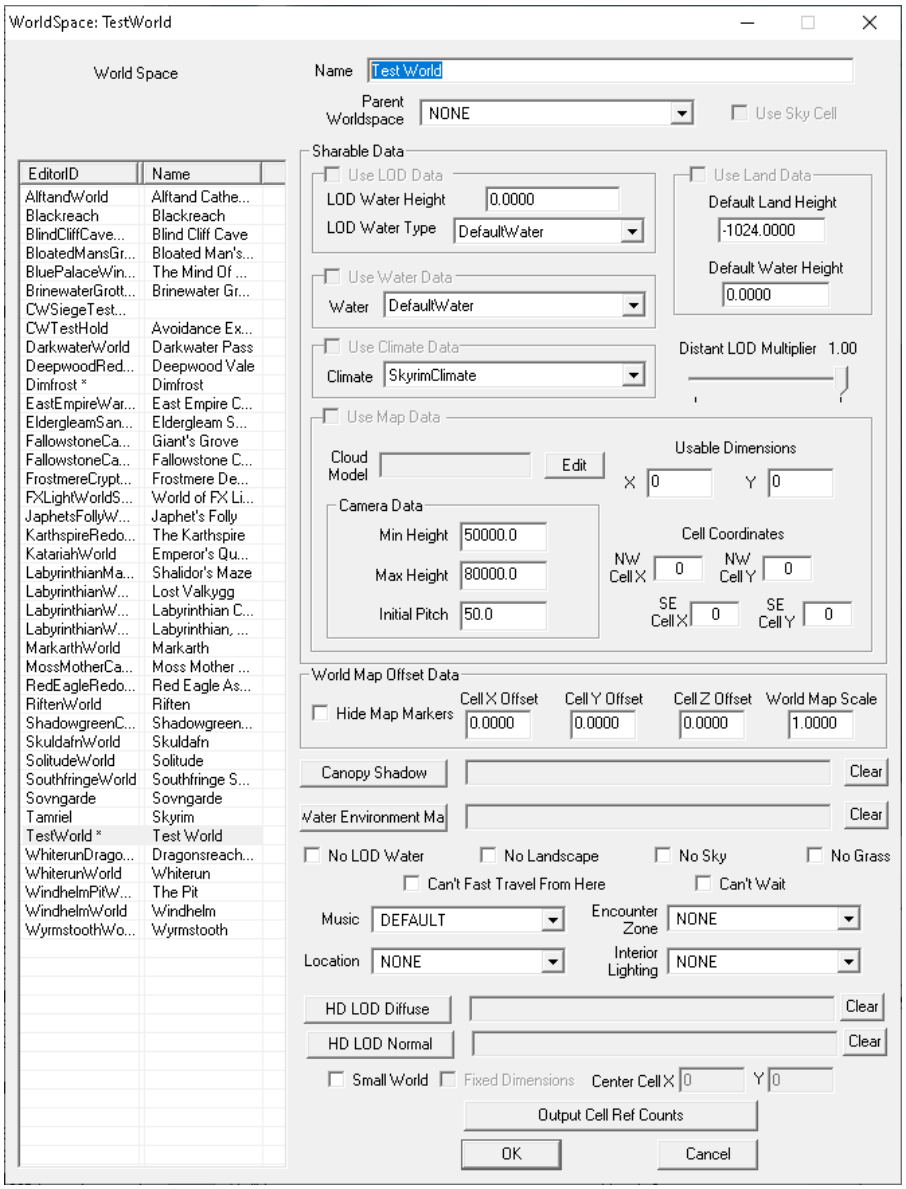


Figure 11 - World Space details.

For now let's just set the Climate to SkyrimClimate, set the Default Land Height to -1024 and untick Small World. Everything else can be left at its default values for now.

Click OK to close.

In the Cell View, change the current world space to the world space you just created. You should see a single Wilderness cell at the coordinates '0,0'. Double-clicking on this cell will load the new world space at that cell and will automatically generate cells around it.

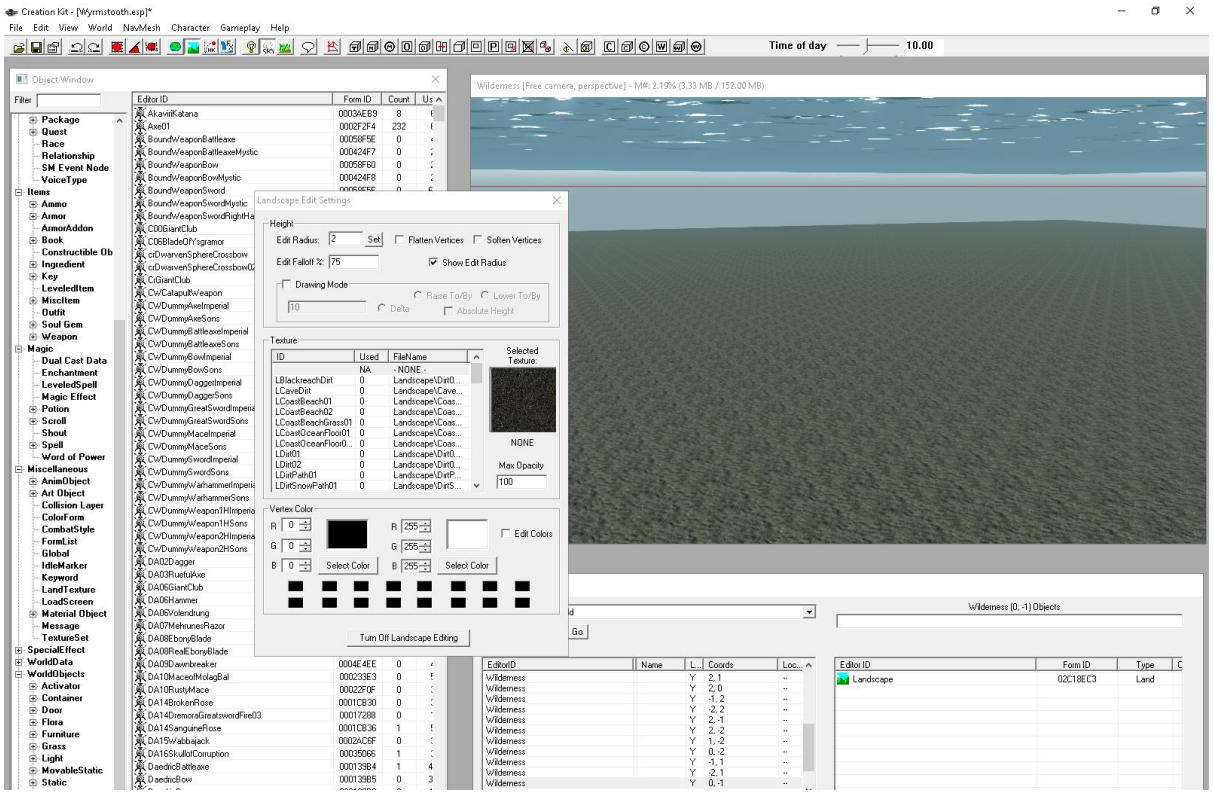


Figure 12 - The new world space.

Right now everything will be underwater. You can use the 'H' hotkey to bring up the Landscape Editing tool. With the tool active, you can rotate the camera around by holding down the Shift key while moving the mouse to get a better view. Hold down the Spacebar while moving your mouse or hold down the middle mouse button to pan around.

CREATING A HEIGHT-MAP USING WORLD MACHINE

In this section I'll be covering the process of generating a height map in World Machine that we can use to create a new world space for Skyrim.

You'll need a copy of World Machine which you can download from their website.

Install World Machine and launch it.

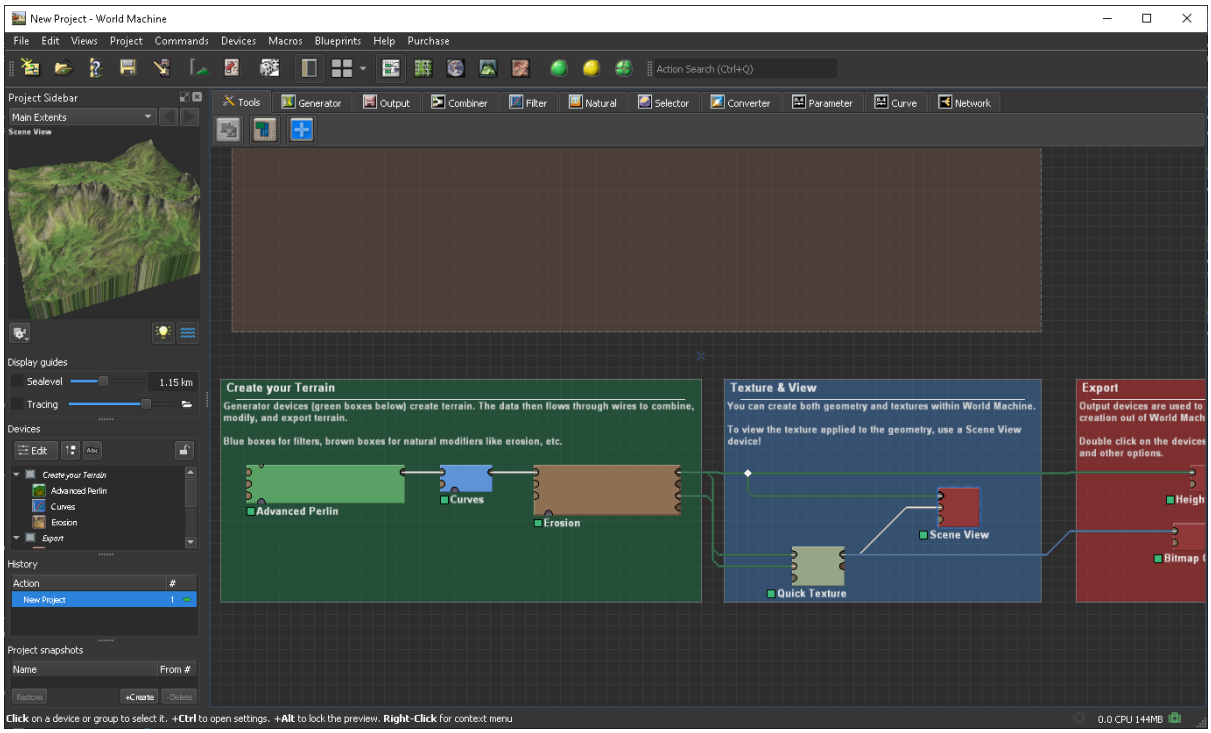


Figure 13 - The World Machine default project.

World Machine is a node-based height map generator. It can be used to create a height map from scratch and apply realistic erosion.

Note: this section won't be a deep dive into every facet of World Machine. I'm just going to cover the basic process of combining different nodes to generate a height map that we can export.

Later we'll be using TESAnnwyn to create a new .esp file using the height-map we create to shape our first world space.

I'm going to start by clearing out the existing nodes. Just click on everything and press Delete.

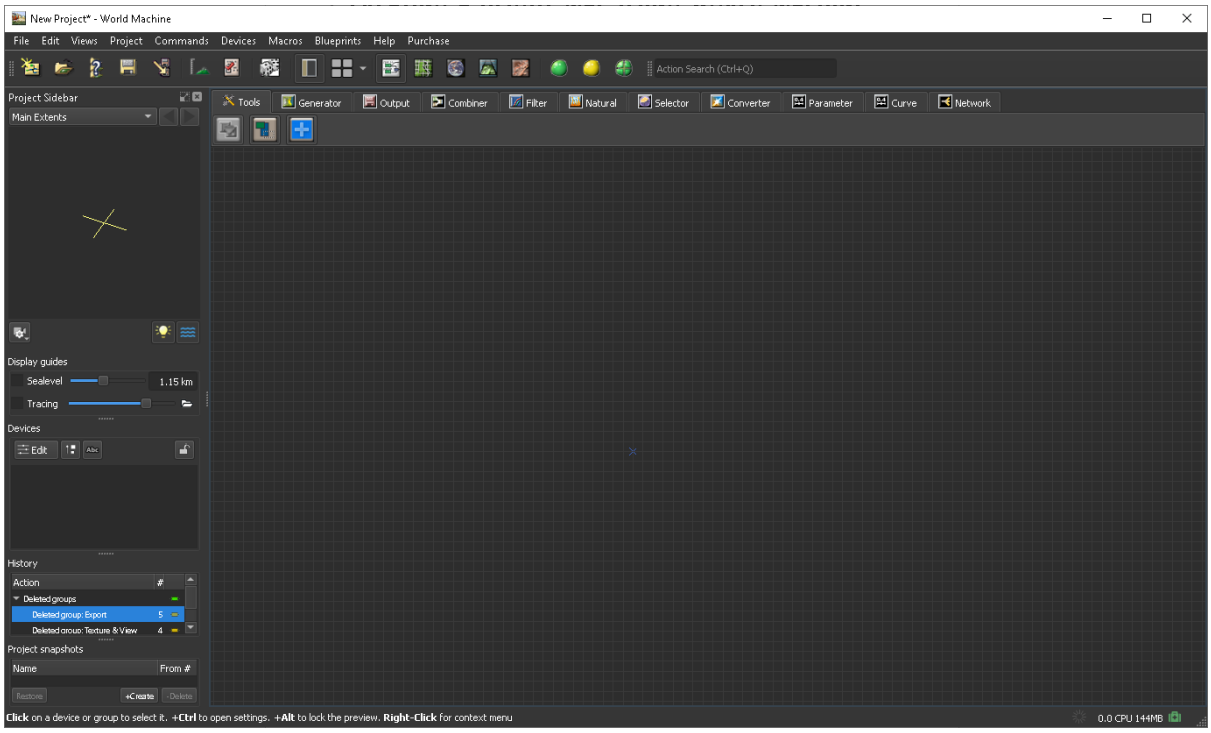


Figure 14 - Blank project.

So let's say we wanted to create an island world space. Go to the Generator tab, click on the Radial Grad button and click on the workspace to place the node.

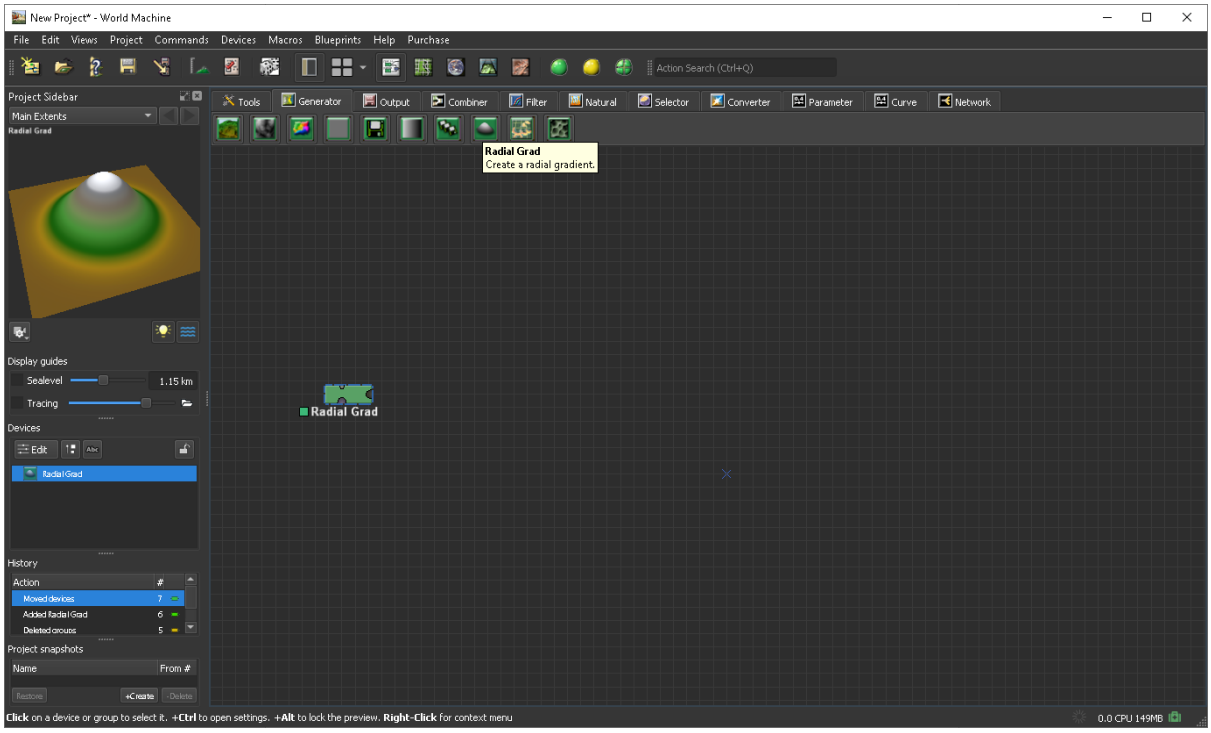


Figure 15 - Adding a Radial Grad node.

Near the top-left you can see a small preview of what our landscape looks like so far.

Double-click on the Radial Grad node to open its properties.

Reduce the radius to about 3.2 km then click Close.

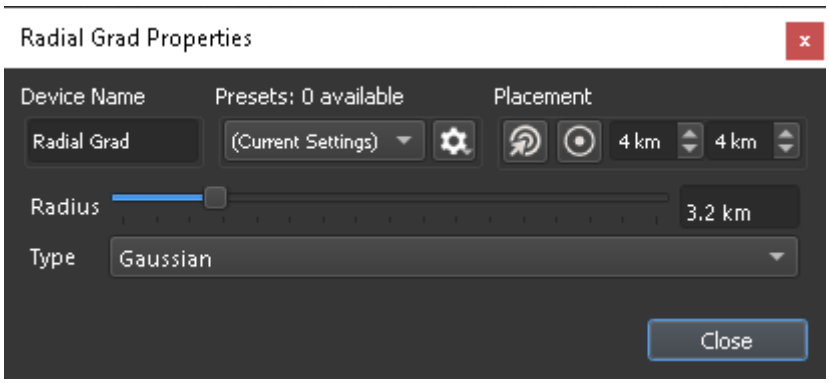


Figure 16 - Reducing Radial Grad radius.

Click on Advanced Perlin and place a node next to your Radial Grad node.

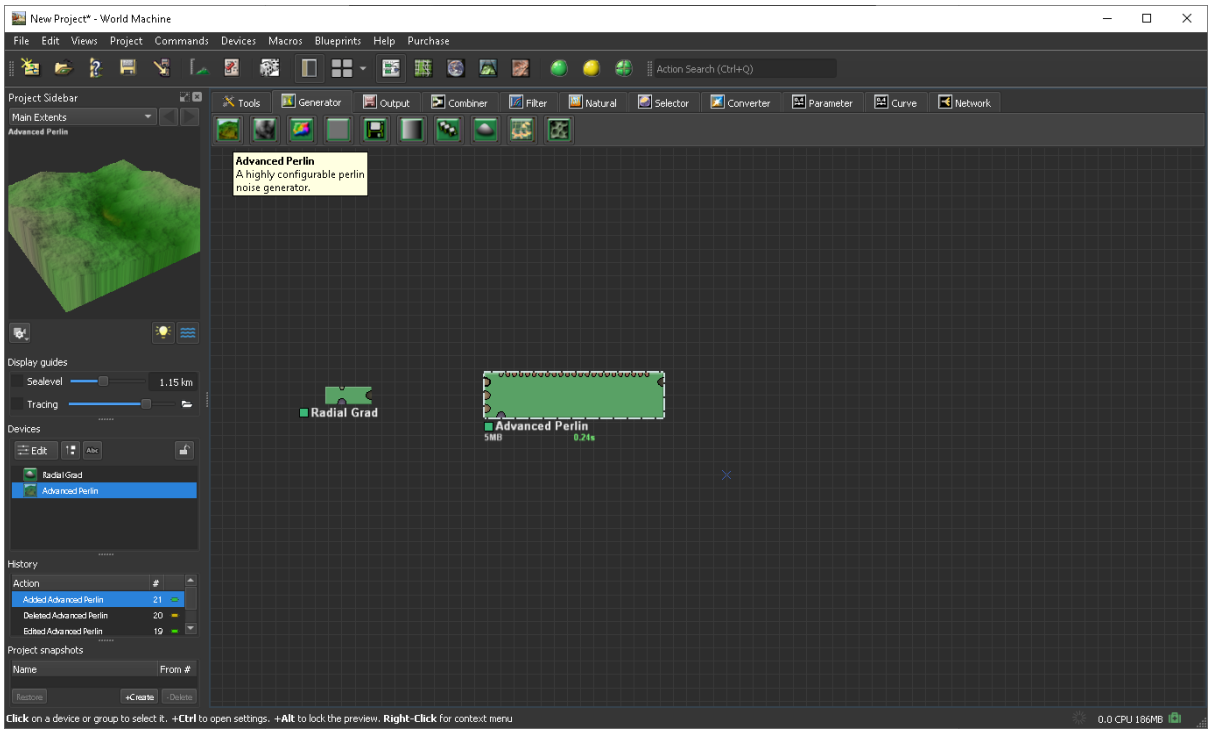


Figure 17 - Placing an Advanced Perlin node.

Connect the two nodes by clicking and dragging on the Radial Grad's primary output and link it to the Advanced Perlin's Shaping Guide input.

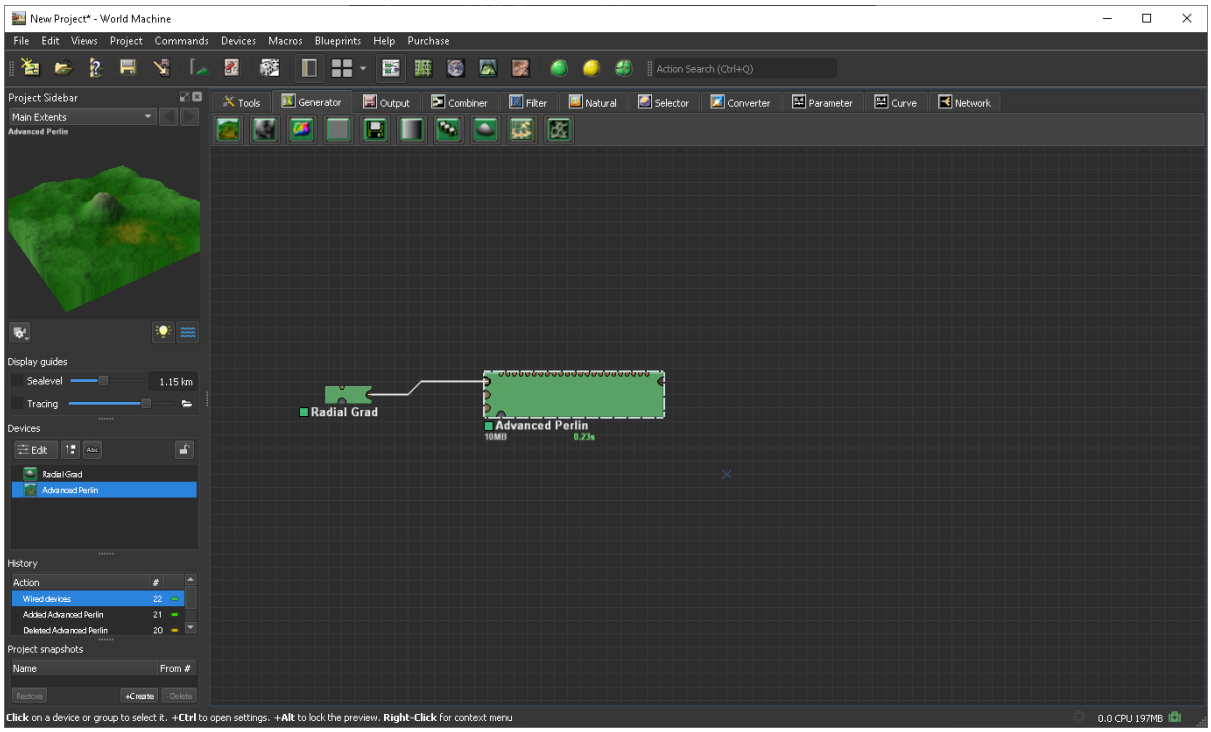


Figure 18 - Linking the Radial Grad to the Advanced Perlin.

In the preview you can see how both nodes are being combined.

Double-click on Advanced Perlin to open its properties.

For my example, I set the Feature Scale to 3 km, the Style to 'Smooth Billowy' and the Middle Elevation to 1 km.

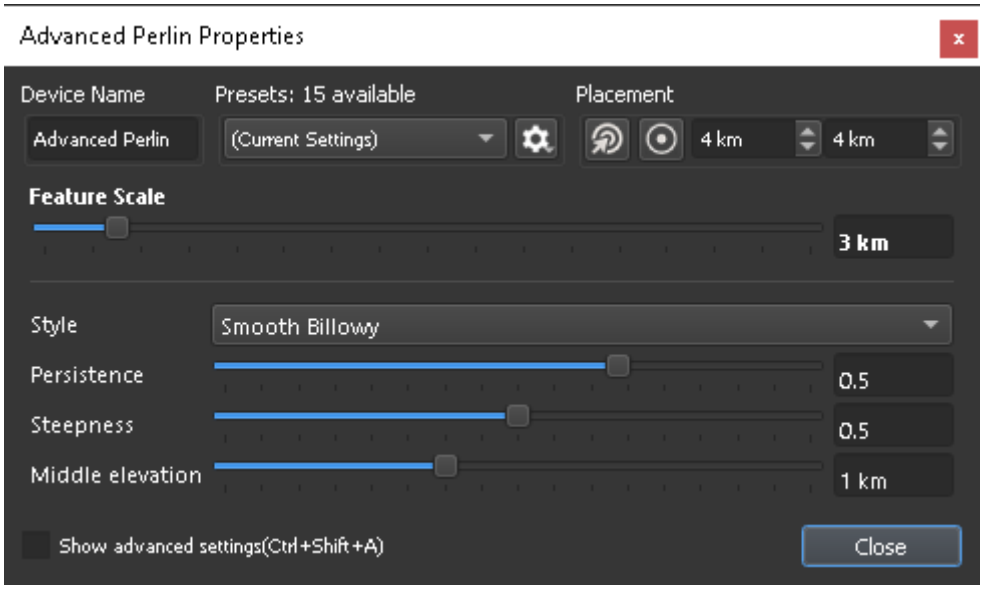


Figure 19 - Advanced Perlin properties.

Tick 'Show advanced settings'.

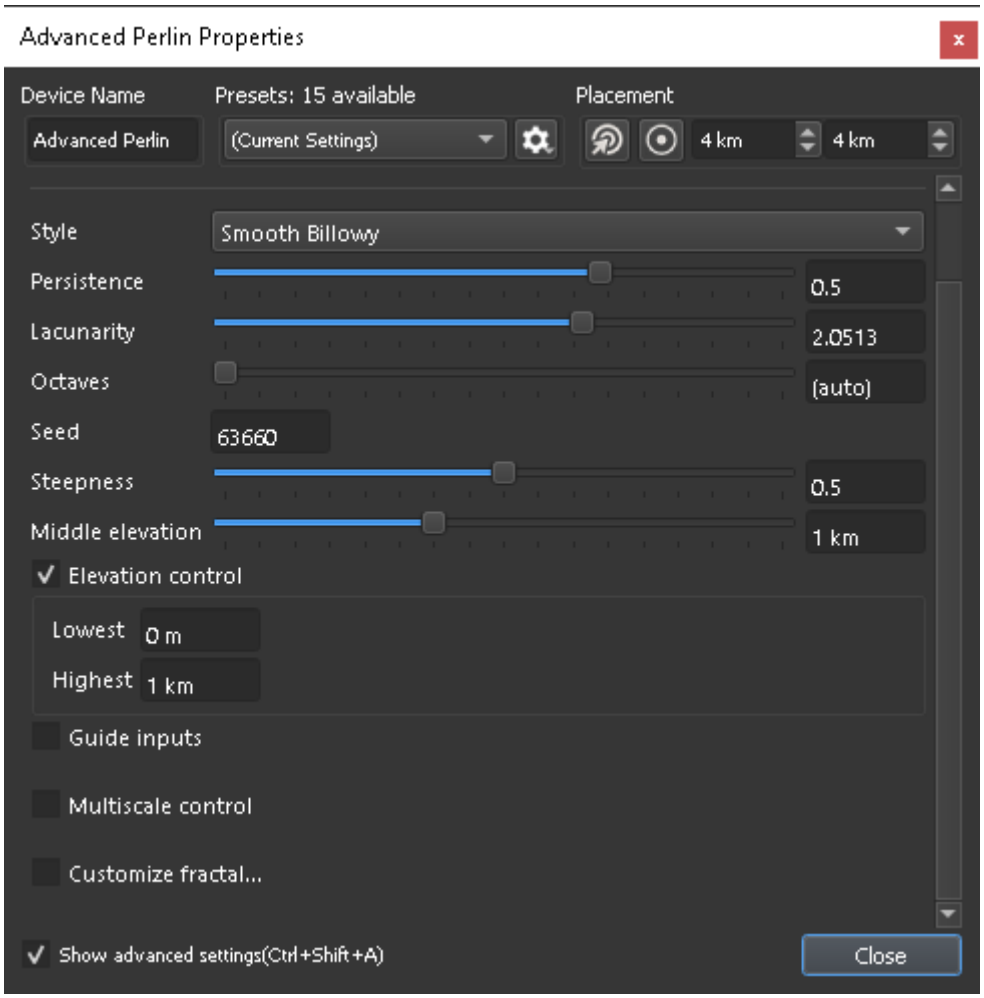


Figure 20 - Elevation control.

Tick 'Elevation control' and set the Highest field to 1 km.

Click Close.

Our central island mountain looks too conical, so let's try and break that shape up.

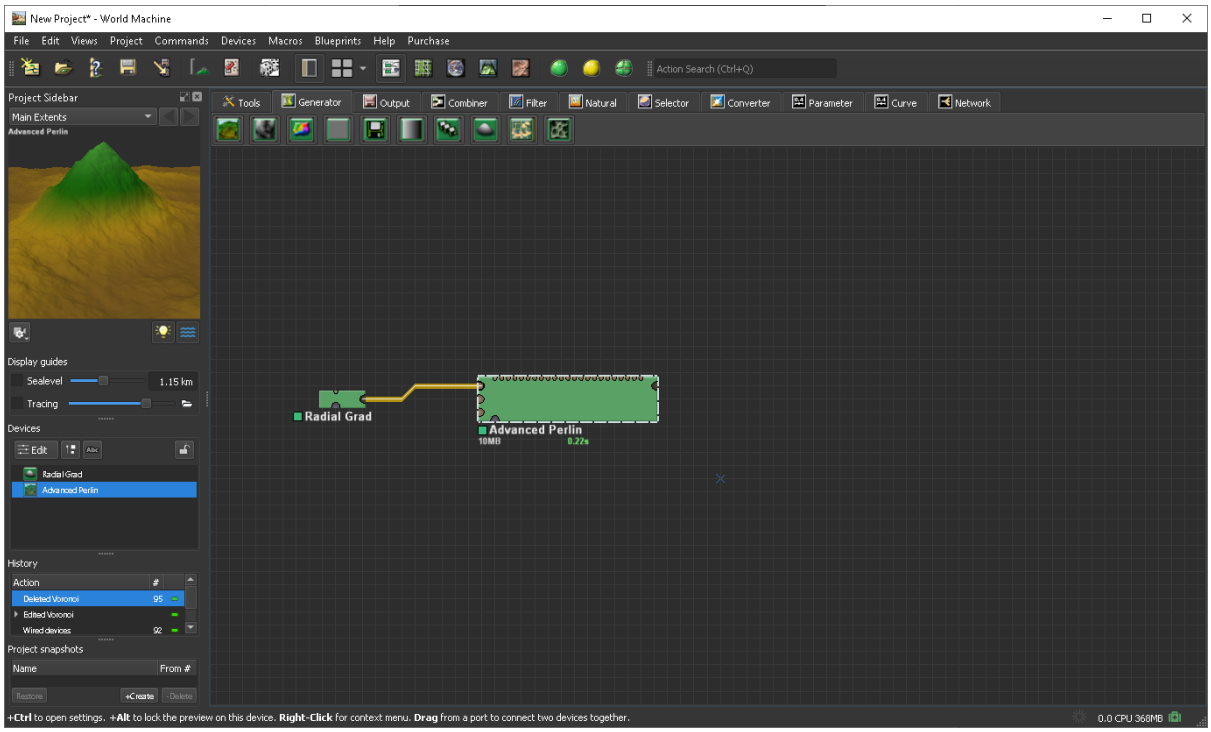


Figure 21 - Height-map preview.

Click on Voroni and place a node on your workspace.

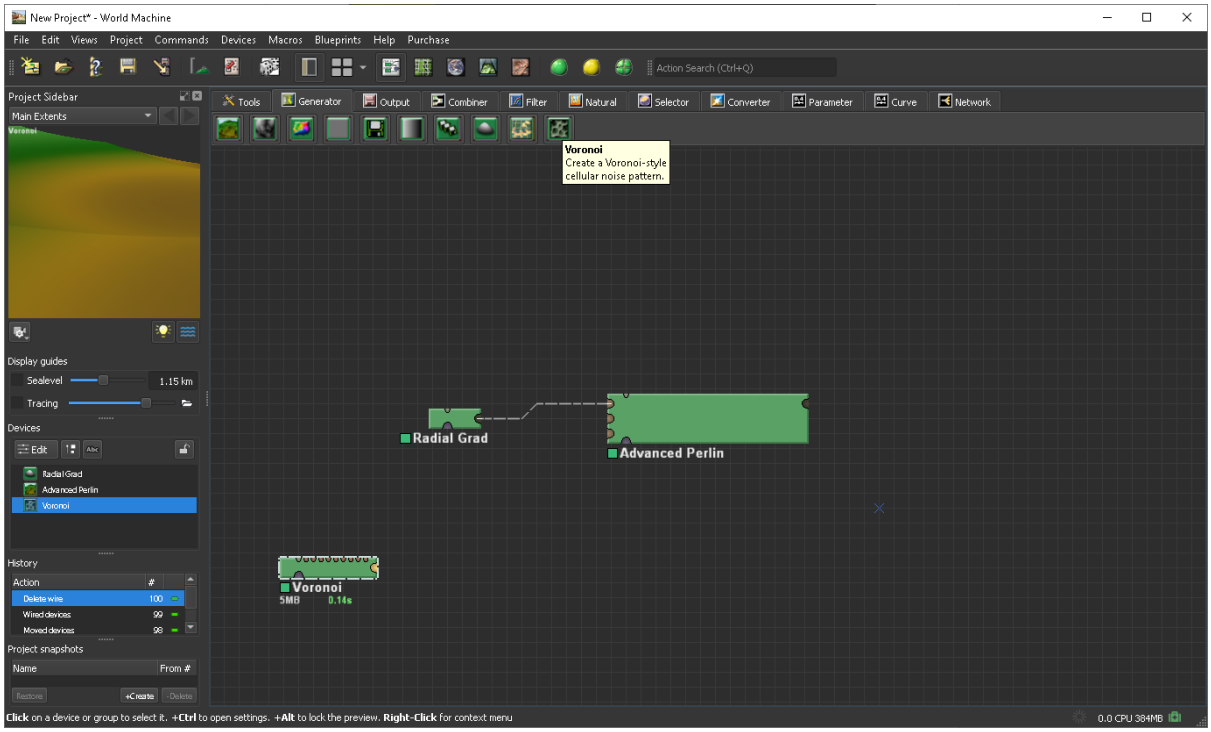


Figure 22 - Adding a voronoi node.

Double-click on the Voronoi node to open its properties.

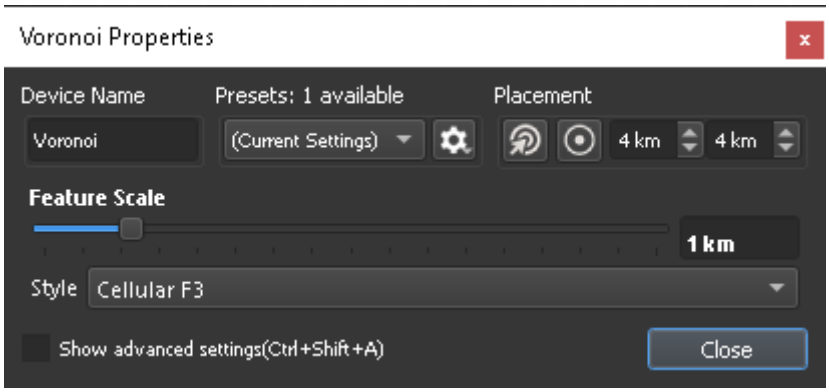


Figure 23 - Voronoi properties.

Set Feature Scale to 1 km and Style to 'Cellular F3'.

Click Close.

Connect the Voronoi node's primary output to the Radial Grad node's Mask Input.

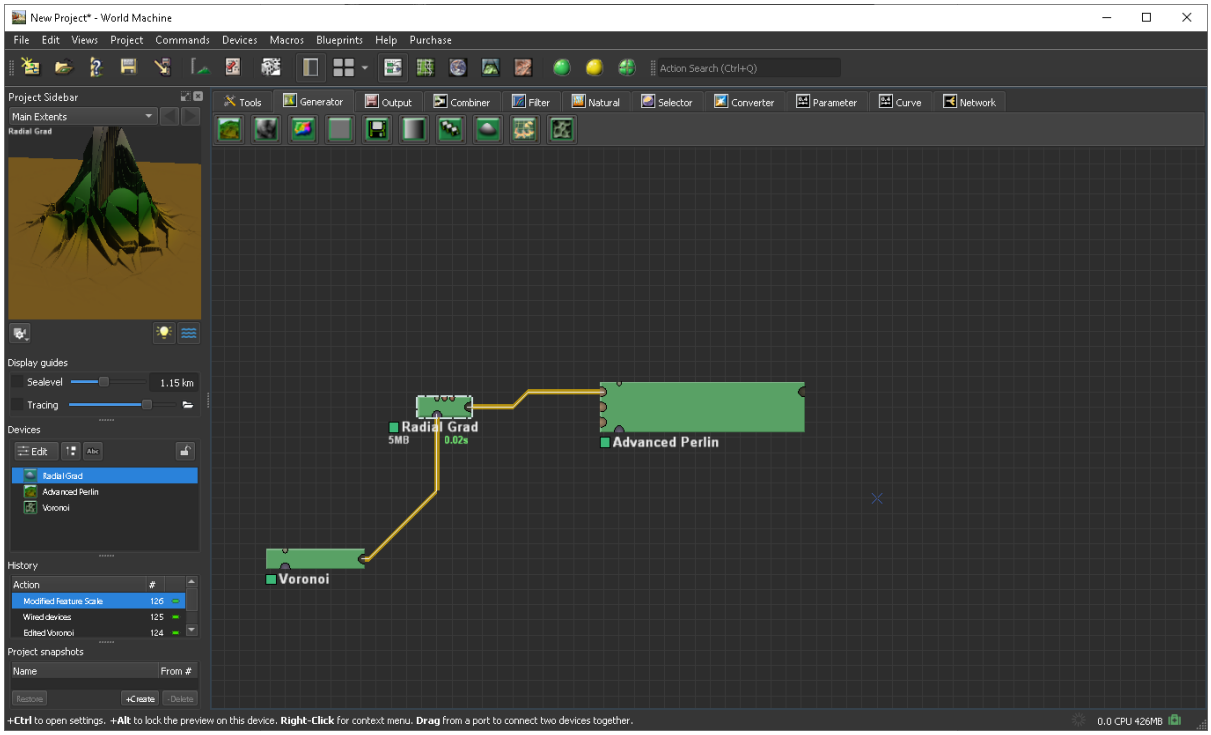


Figure 24 - Connecting the Voronoi node to the Radial Grad node.

This should help break up that cone shape.

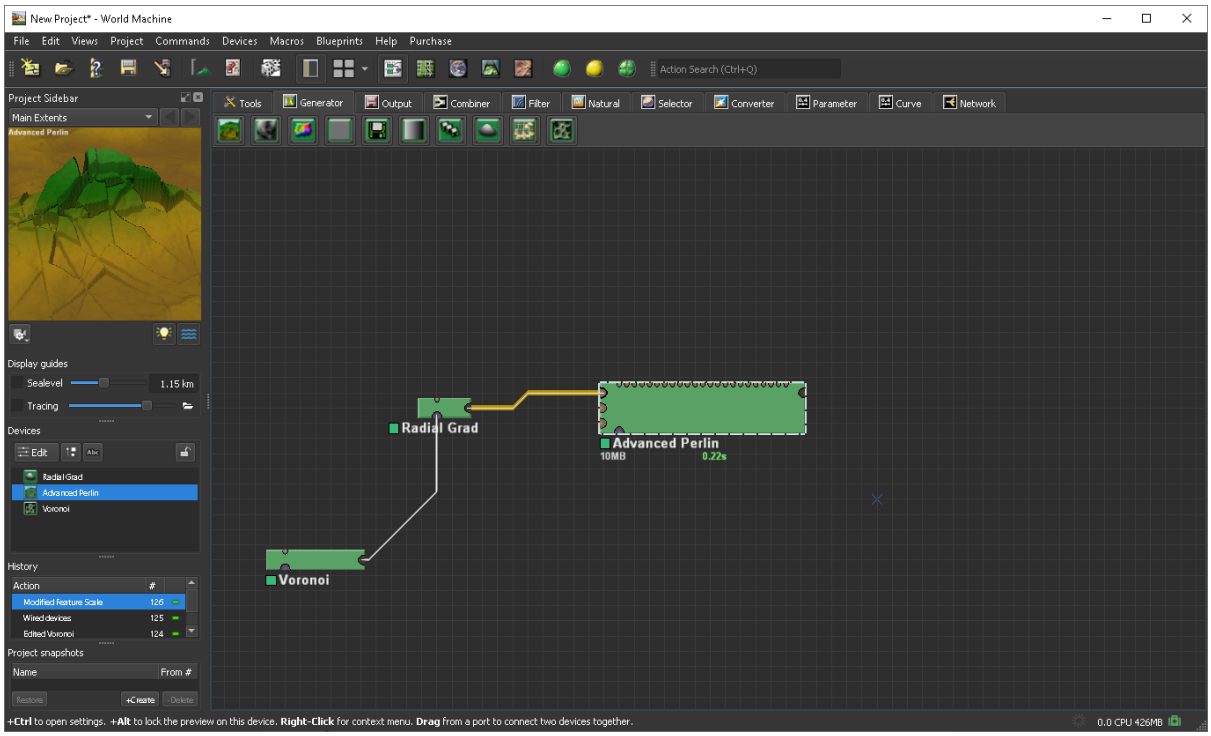


Figure 25 - Radial Grad with Voronoi.

We can further change the overall shape of the island by combining two inputs and passing the output to the Advanced Perlin node.

Right-click on the line between Radial Grad and Advanced Perlin and select 'Delete wire'.

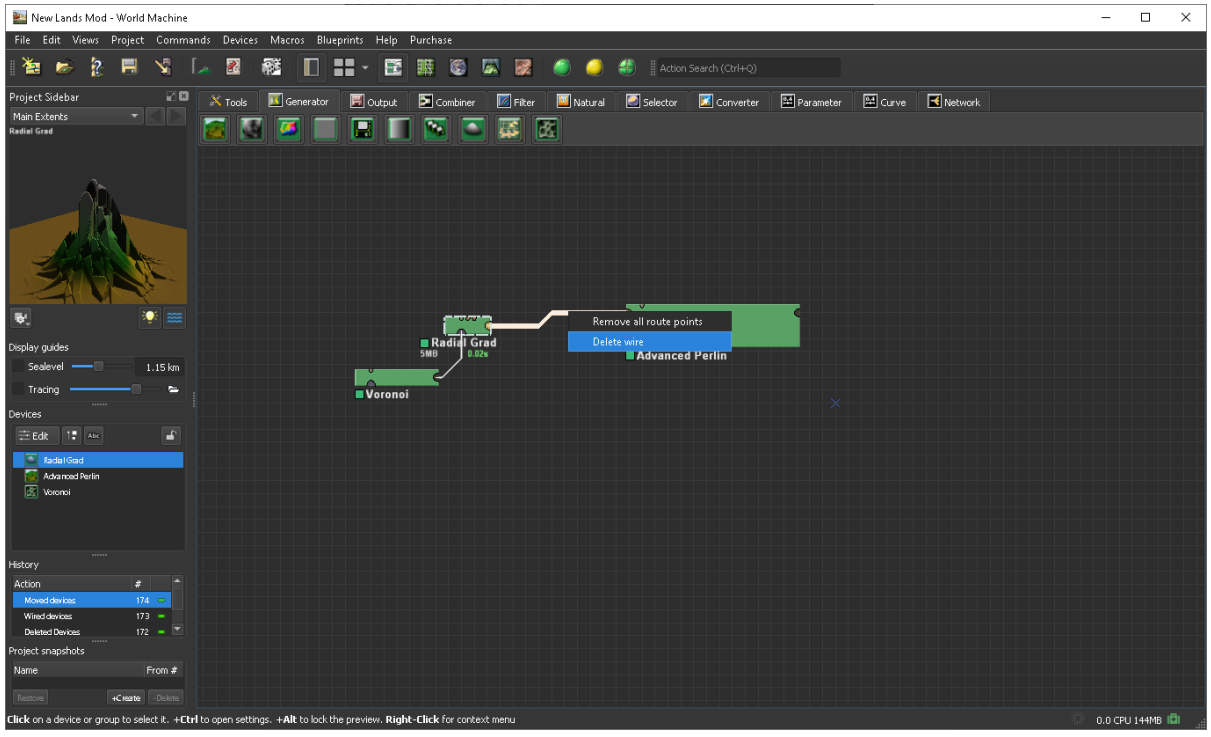


Figure 26 - Removing the link between Radial Grad and Advanced Perlin.

Go to the Combiner tab and select Combiner.

Click in your workspace to add it.

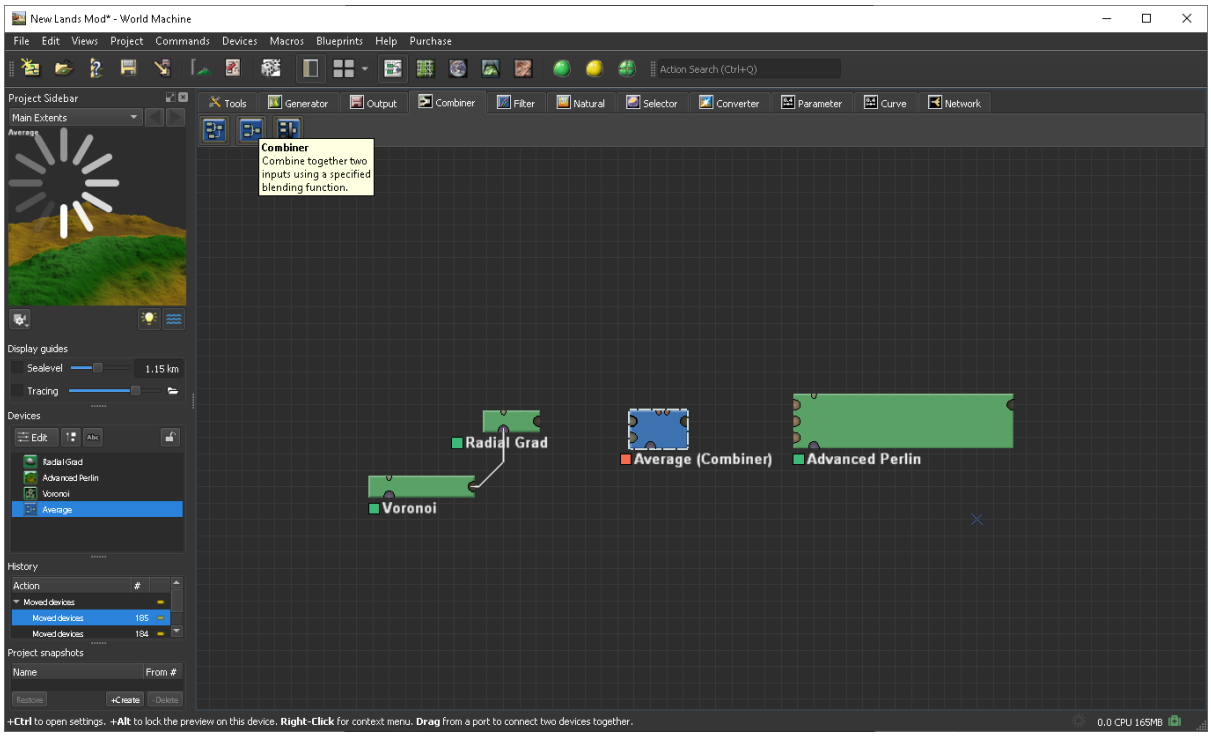


Figure 27 - Adding a combiner node.

Link the Radial Grad to the first input on the combine node, and link the combiner node's output to the Advanced Perlin node.

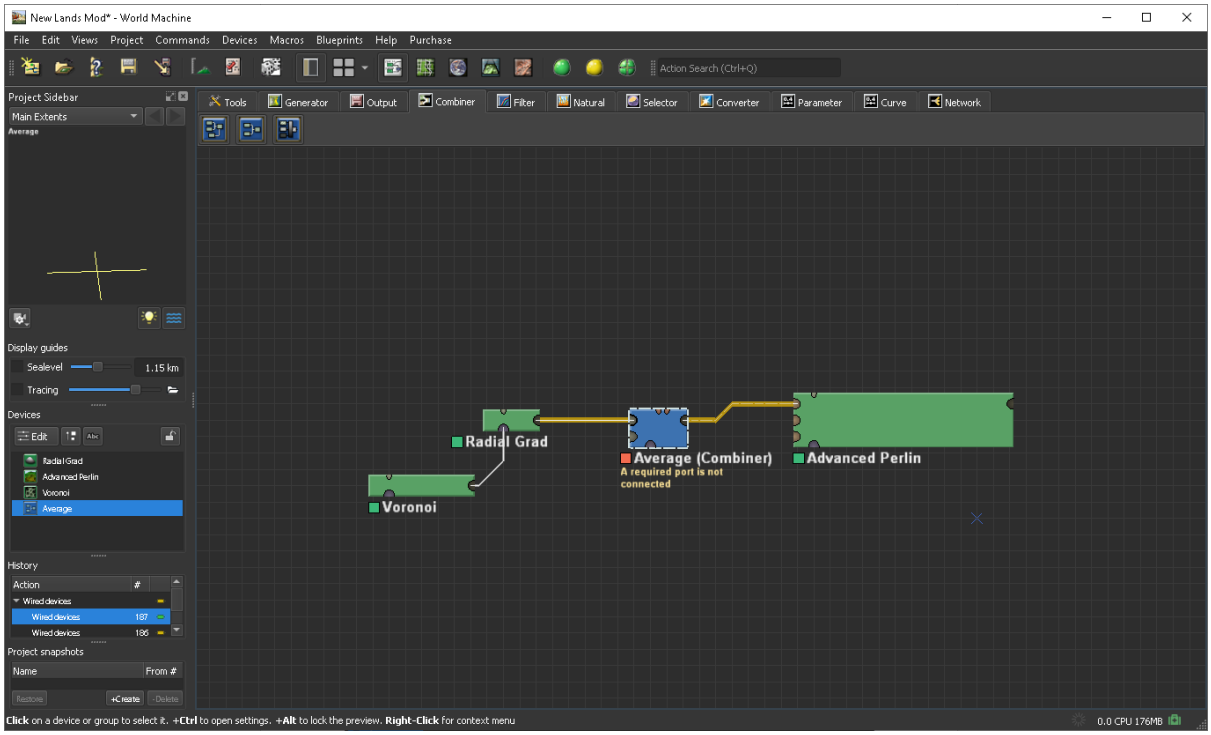


Figure 28 - Linking the first set of nodes to the combiner node.

Click on the Voronoi node to select it. Press CTRL+C to copy it and CTRL+V to place a copy of the node in your workspace.

Do the same for the Radial Grad node.

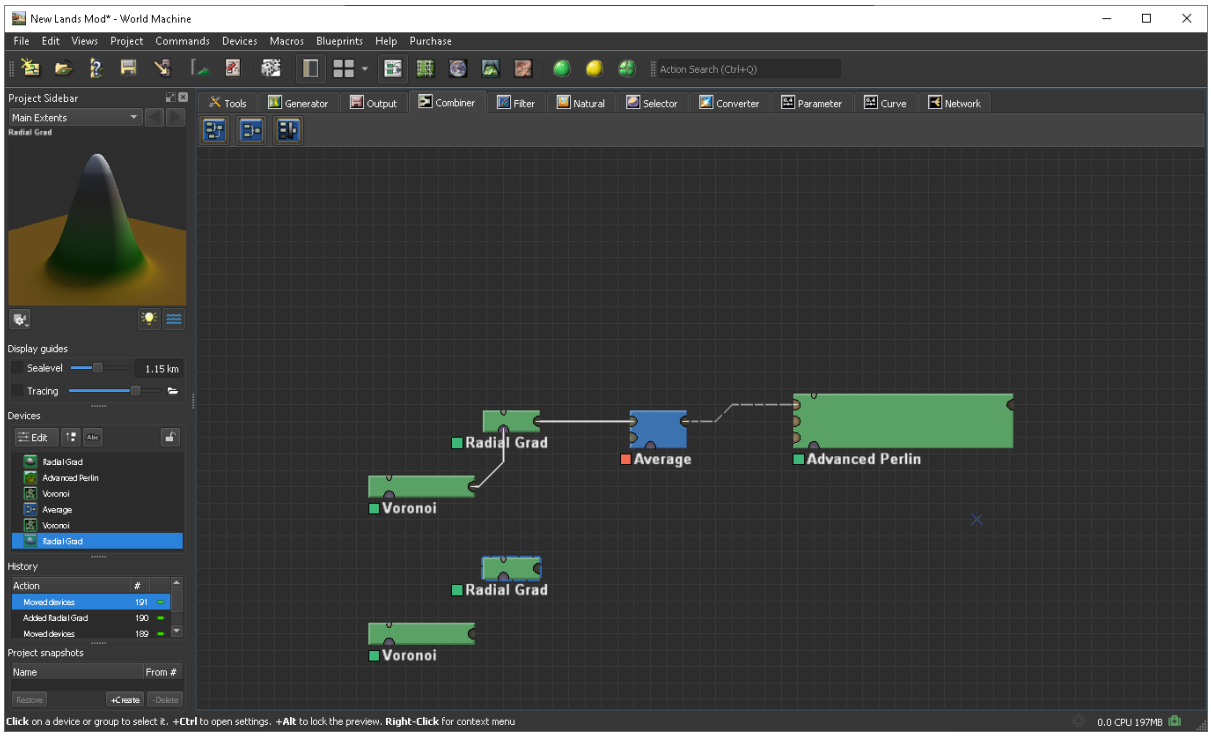


Figure 29 - Copied the Voronoi and Radial Grad nodes.

Link the Voronoi node to the Radial Grad node and link the Radial Grad node to the second input on the combiner node.

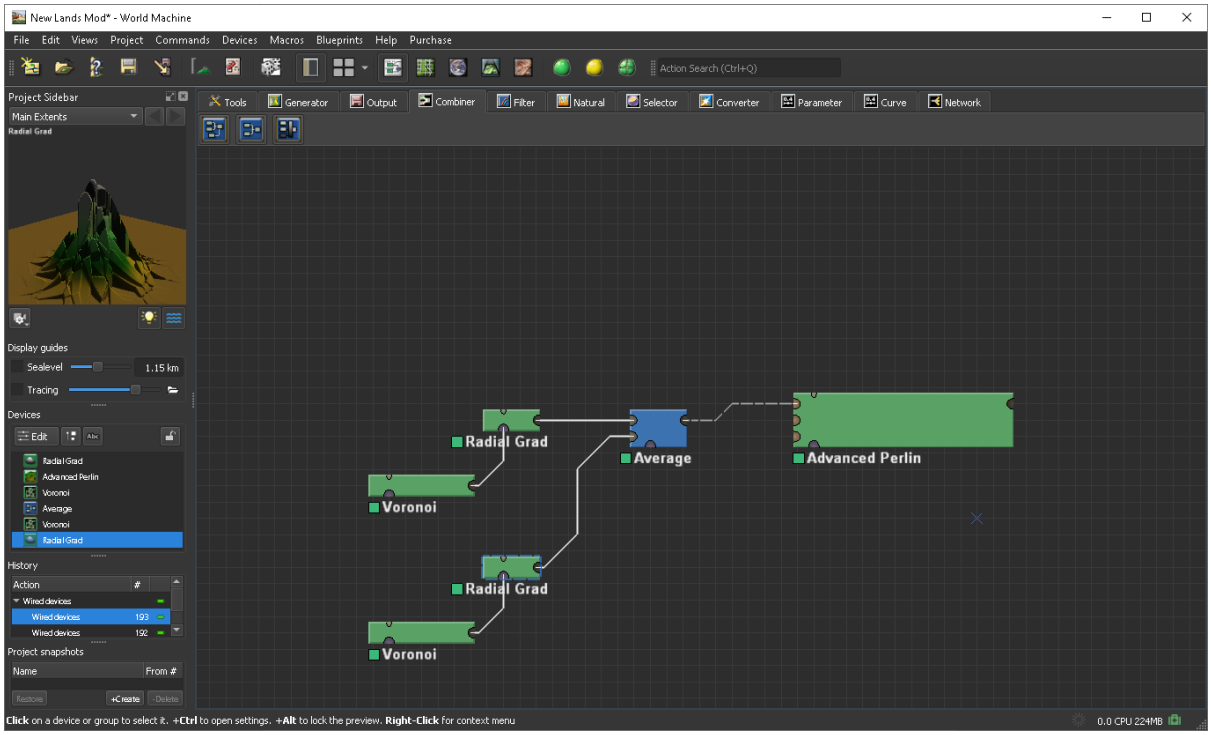


Figure 30 - Linking the second set of inputs to the combiner node.

Double-click on the second Radial Grad node to open its properties.

In my example, I offset its Placement to 3 km in both the X and Y fields.

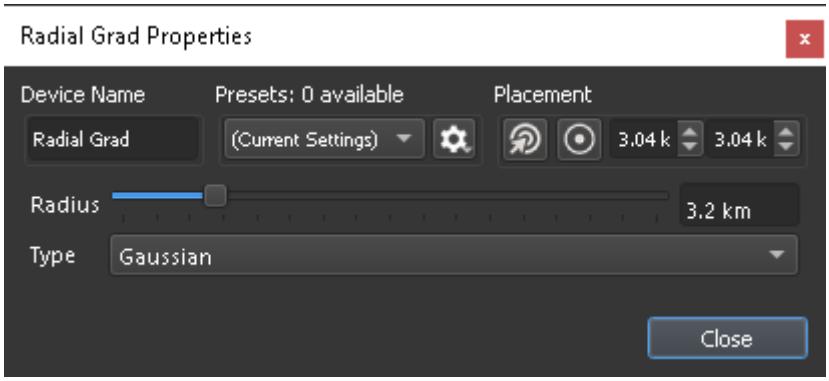


Figure 31 - Changing the Placement offset of the second Radial Grad node.

Click Close.

Now double-click on the combiner node to open it's properties.

Set the Method to Add and the Strength to 0.45.

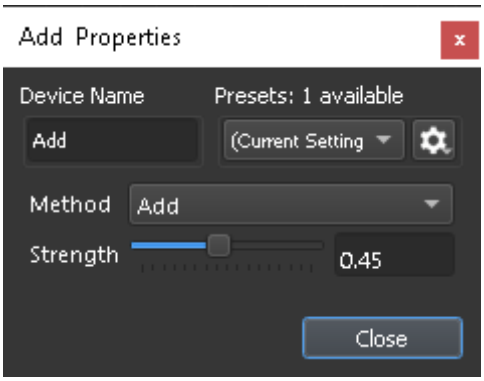


Figure 32 - Changing the combiner node's method and strength.

Click Close.

Alright, so let's change things up a bit more.

Delete the link between the combiner and the Advanced Perlin node.

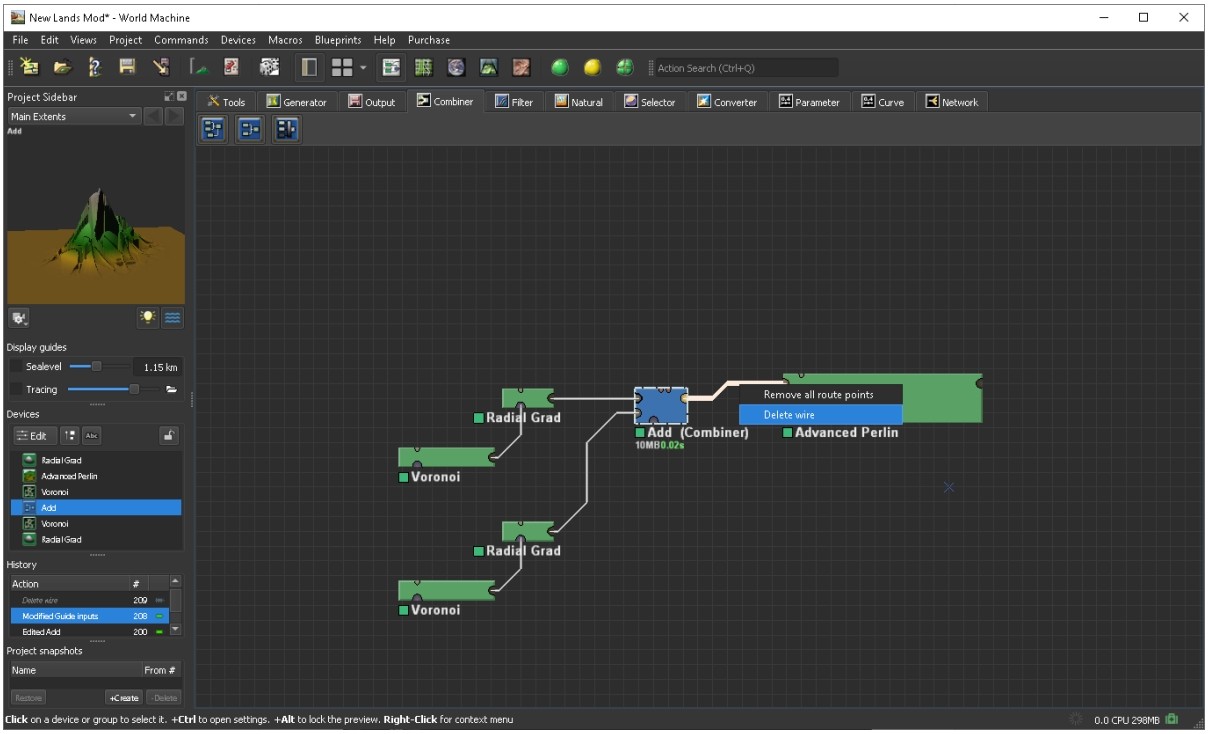


Figure 33 - Removing the link between the combiner node and the Advanced Perlin node.

Add another combiner node.

Connect the output from the Advanced Perlin node to the first input on the new combiner node, then add the output of the first combiner node to the second input on the new combiner node.

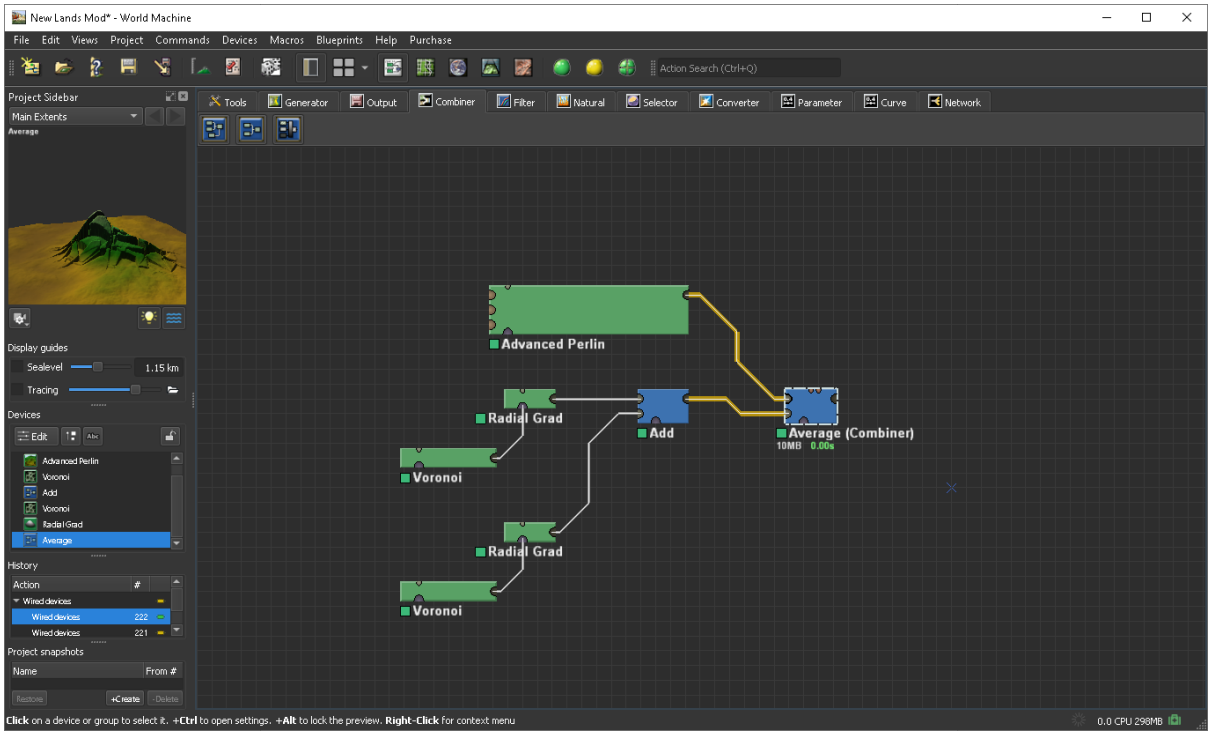


Figure 34 - Second combiner node.

Alright, now we have a less regular island shape. Click on the 3D View button to get a closer look.

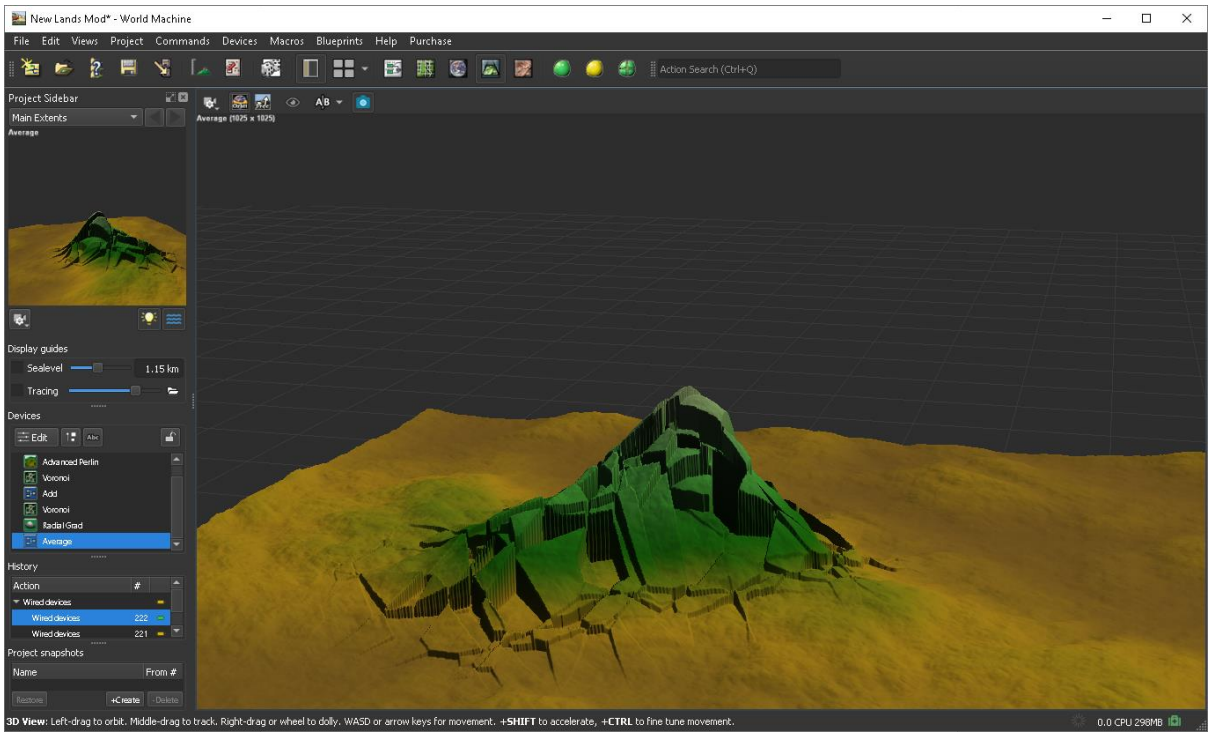


Figure 35 - Preview of our island so far.

Click on the Device Workflow button to return to the workspace.

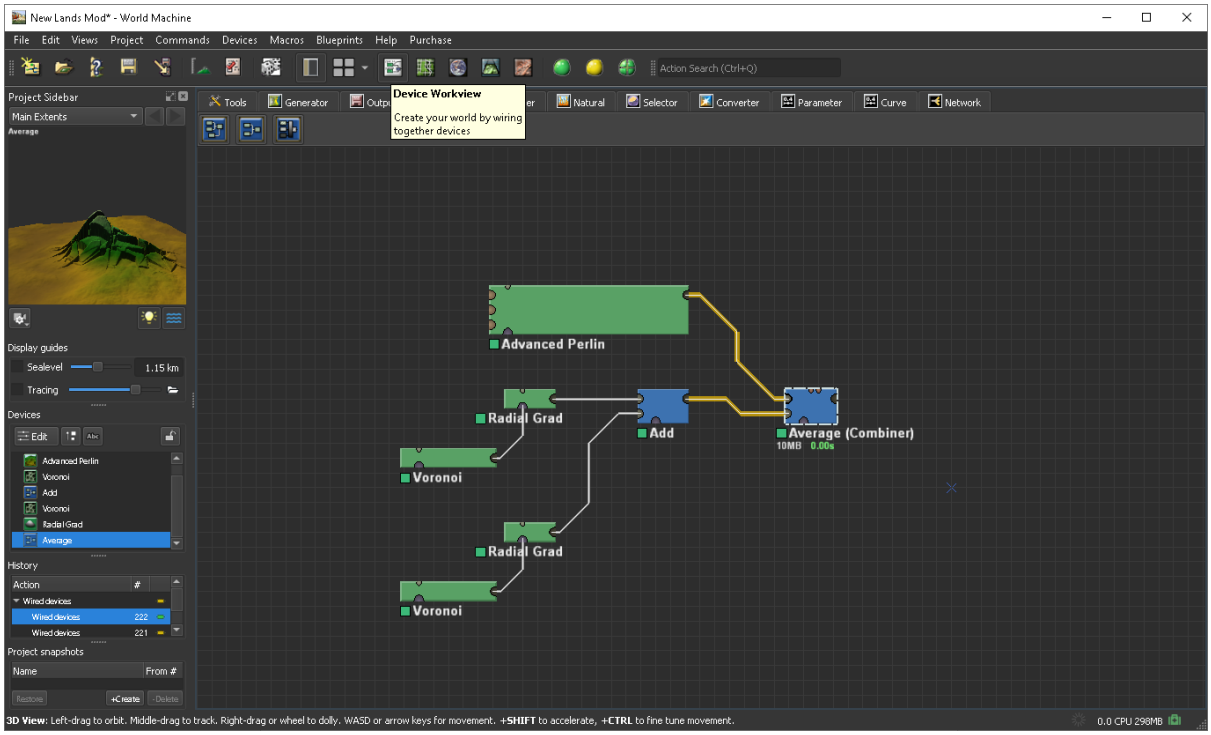


Figure 36 - Back to device workflow.

Open the properties of the second combiner node.

Set the Strength to 0.25 then click Close.

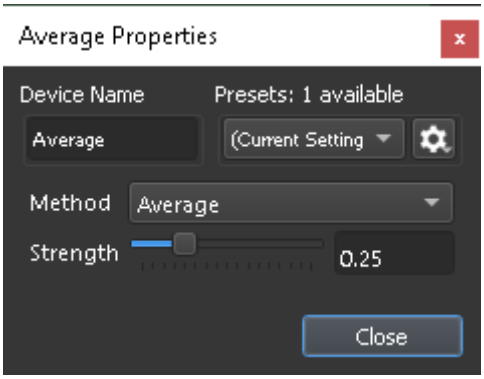


Figure 37 - Second combiner node's properties.

Under the Natural tab, click on the Erosion button to add an Erosion node.

Link the output of the second combiner node to the input of the erosion node.

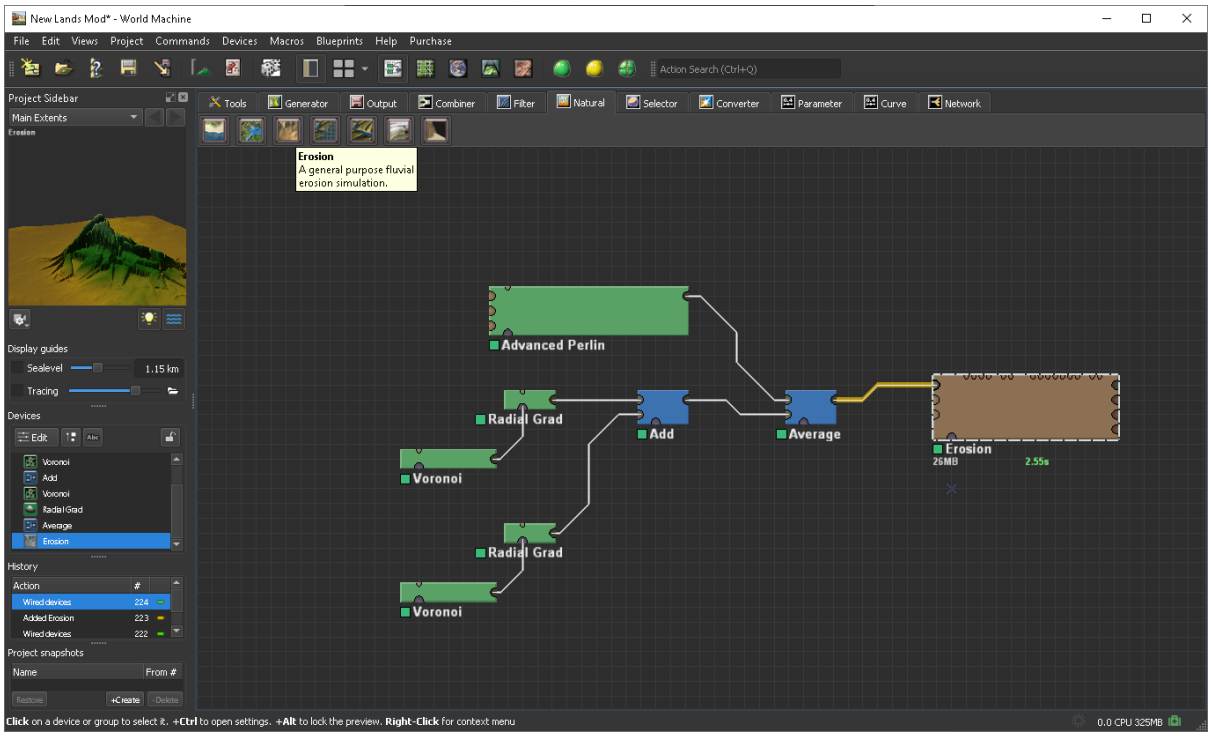


Figure 38 - Adding an erosion node.

Double-click on the erosion node to bring up its properties.

I set the Duration to 165, the Rock Hardness to 0.4 and the Sediment Carry Amount to 0.95.

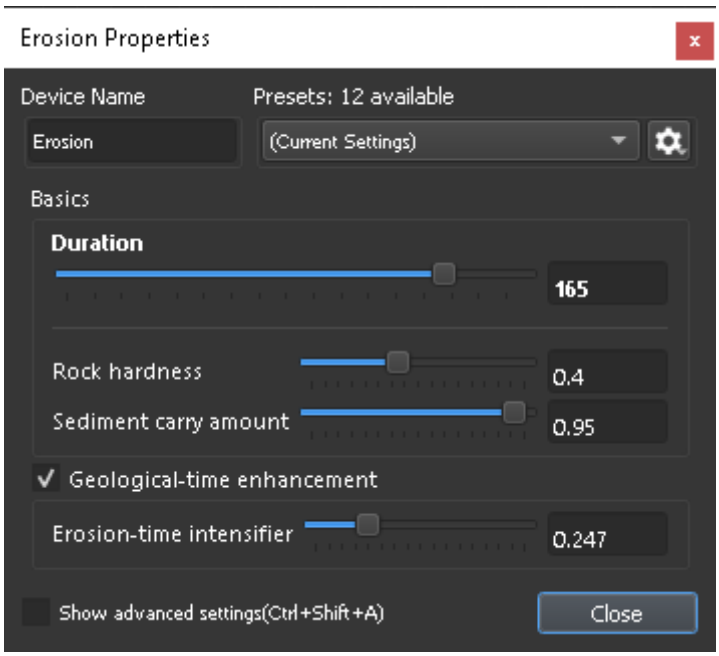


Figure 39 - Erosion properties.

Click Close.

The shape of the island should now look a little bit more realistic now that erosion is being applied.

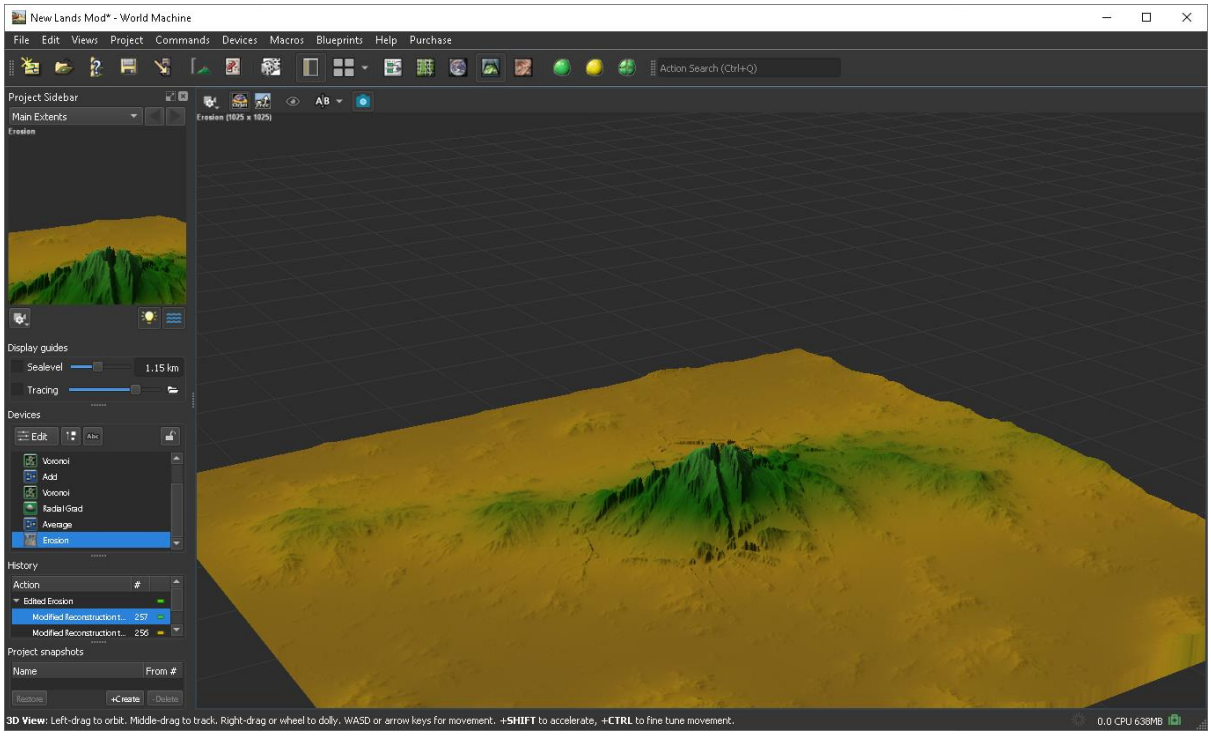


Figure 40 - The island with erosion.

Now, let's try and add some flat areas along the coast to give us some room to maybe build a settlement on.

Under the Natural tab, click on Coast Erosion and add it after the Erosion node.

Link the output of the Erosion node to the input of the Coast Erosion node.

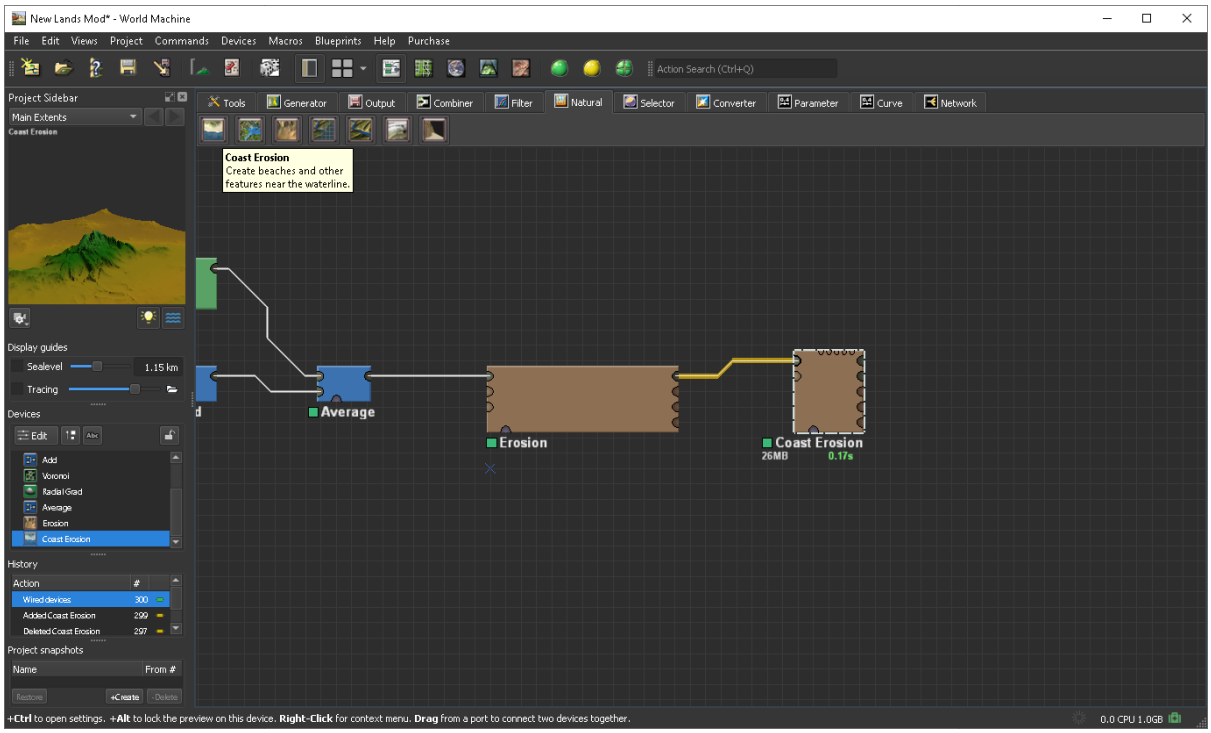


Figure 41 - Adding a Coast Erosion node.

Double-click on the Coast Erosion node to open its properties.

For this example, I set the Sea Level to 350 m, the Beach Size to 250 m, and the Inland Height Influence to 150 m.

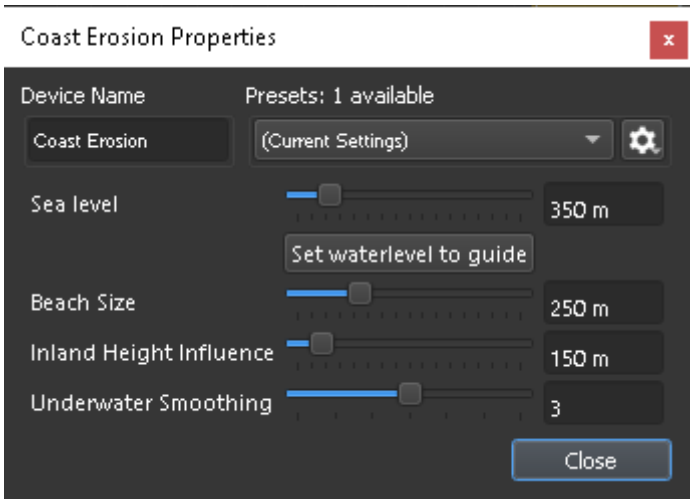


Figure 42 - Coast Erosion properties.

Click Close.

So the centre of the island still looks a bit steep, so I'm going to try and cut a bit more space out of the mountain range out with addition nodes.

Break the link between the second combiner node and the Erosion node.

Add a third Radial Grad node.

Under the Filter tab, you'll find the Inverter node. Add it in and link the output of the third Radial Grad node to the input of the Inverter node.

Lastly, add in a new combiner node. Link the output of the second combiner node to the first input of the third combiner node, and the output of the Inverter node to the second input of the third combiner node as per the screenshot below:

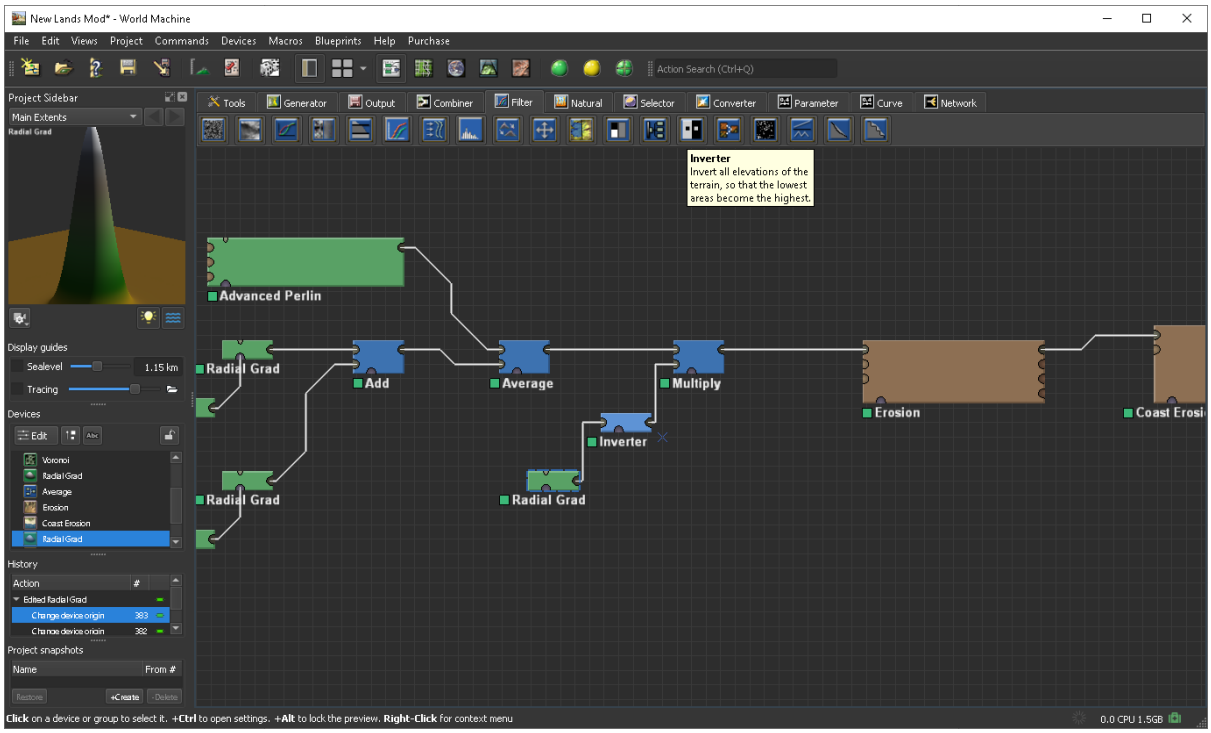


Figure 43 - Added a third combiner node.

Open the properties of the third Radial Grad node.

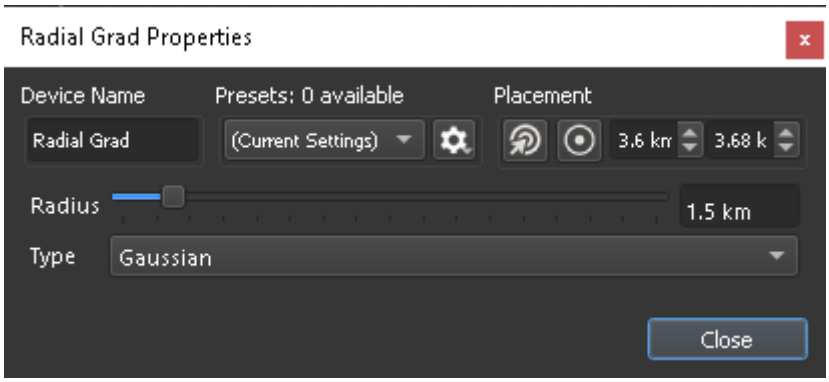


Figure 44 - Third Radial Grad node properties.

Click Close.

Open the properties of the third combiner node.

Set Method to Multiply and set the Strength to 0.7.

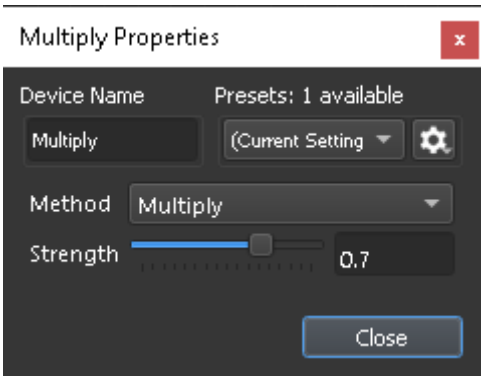


Figure 45 - Third combiner node properties.

Click Close.

We should now have a bit of a crater in the middle of the island.

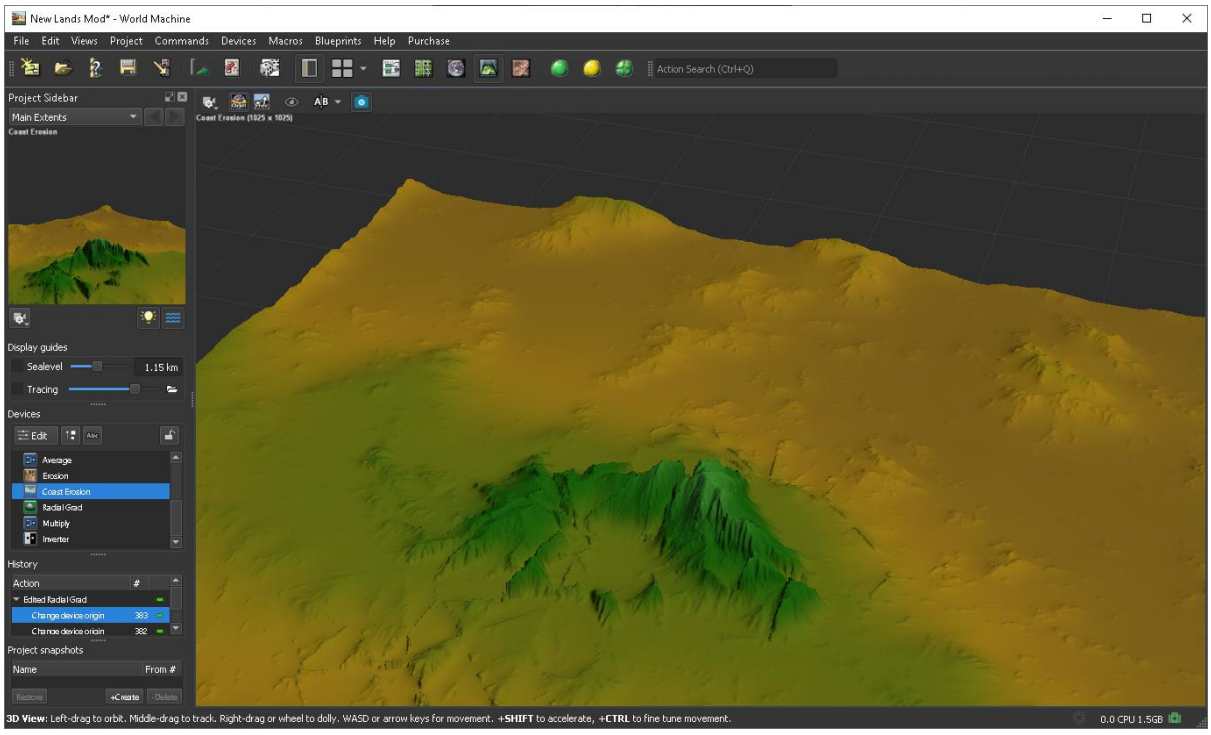


Figure 46 - Crater in the middle of the island.

Looking at the island from another angle, we can see a mountain pass as well.

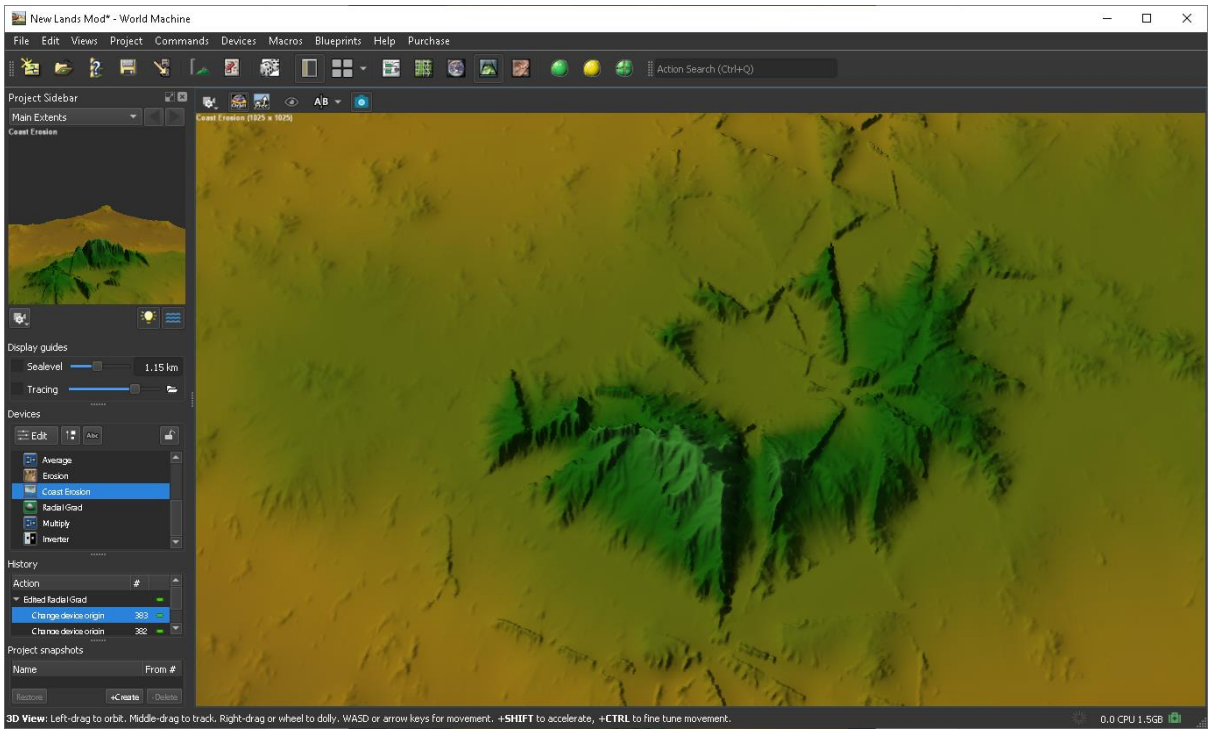


Figure 47 - The island from another angle.

The main thing to consider is the fact that Skyrim doesn't rely on the height-map to create mountain peaks. It uses static objects, so right now we're more concerned with how much playable area there is in our world space and what kind of unique geological features we can riff off-of when we begin cluttering the world space.

Looking around the height-map you'll likely see some sharp spikes sticking out. While we can easily smooth that out with the Creation Kit's landscape edit tool, let's try and save ourselves a bit of work by eroding those features in World Machine.

Under the Natural tab, click on Thermal Weathering and add it to the workspace.

Connect the output from the Coast Erosion node to the input on the Thermal Weathering node.

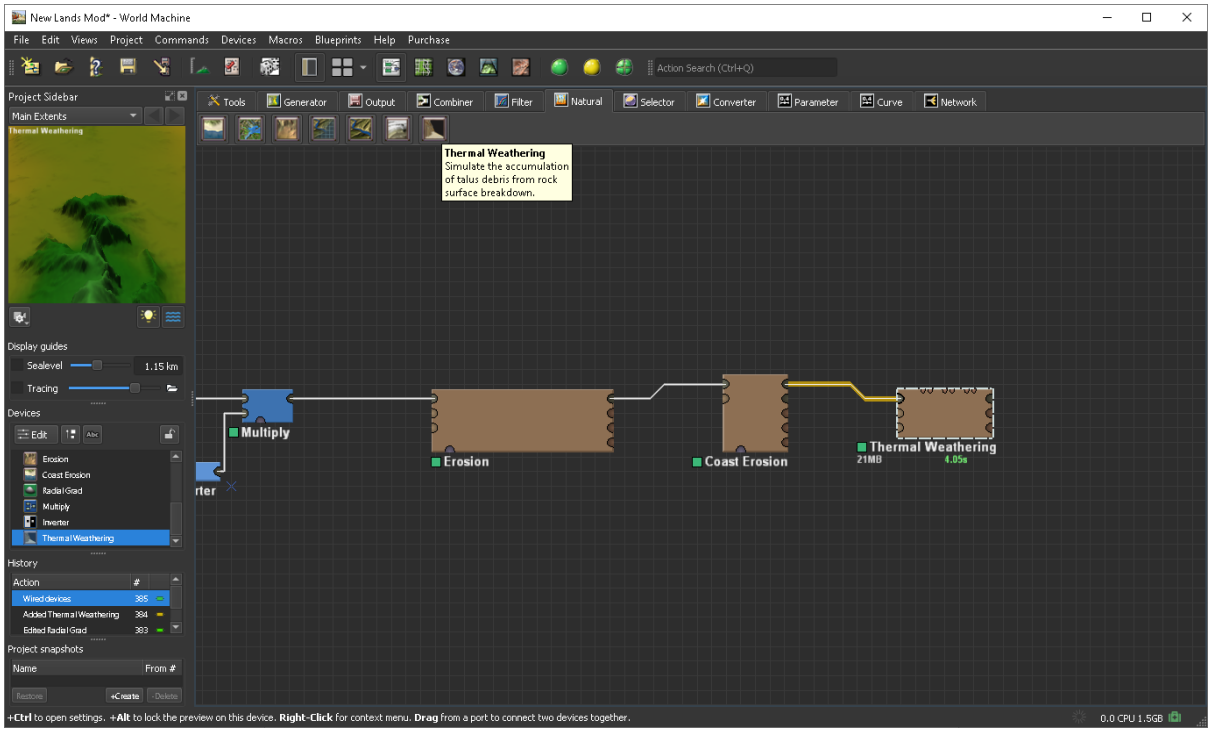


Figure 48 - Adding a Thermal Weathering node.

Double-click on the Thermal Weathering node to open its properties.

Click on 'Show advanced settings' and set the Talus Size to 20 m.

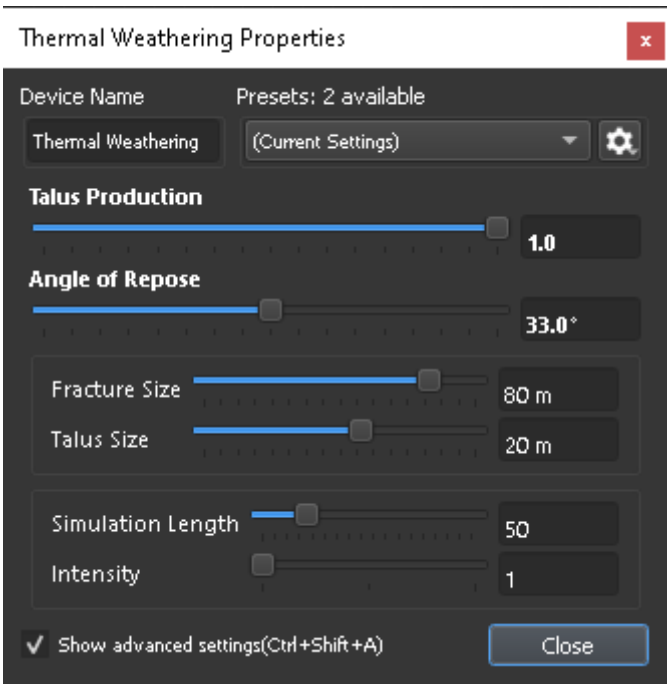


Figure 49 - Thermal Weathering properties.

Click Close.

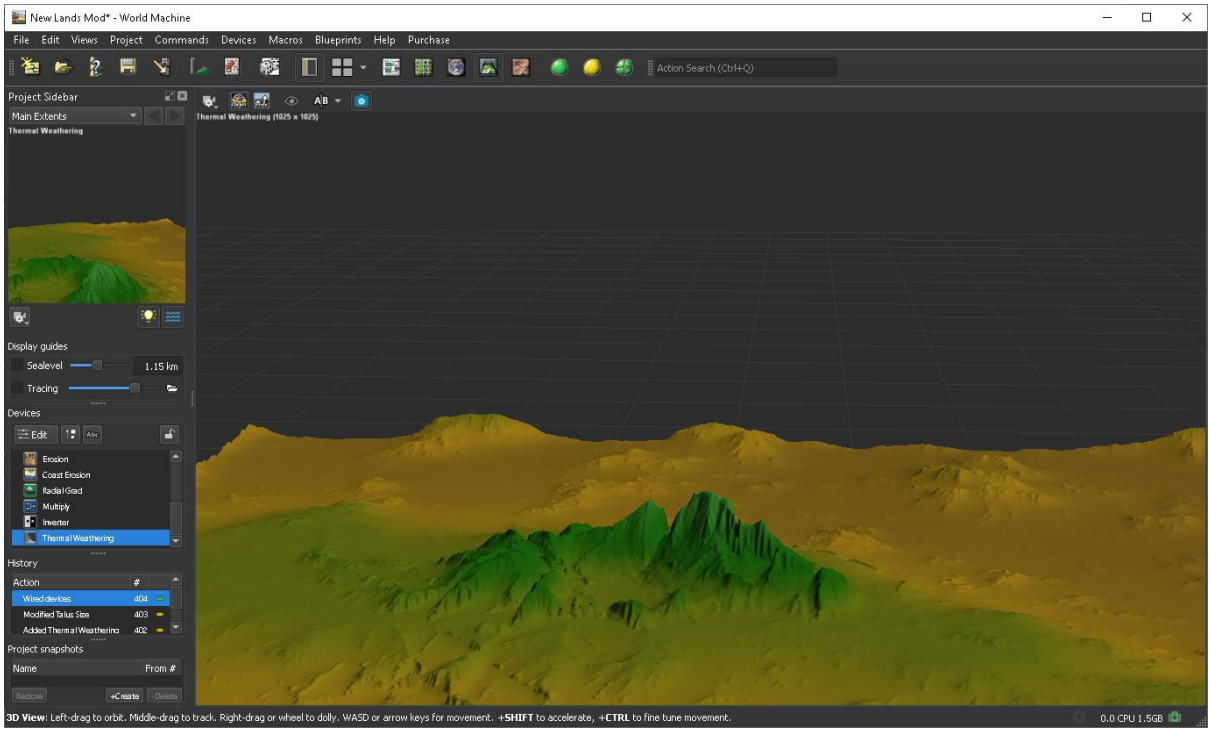


Figure 50 - The island after thermal weathering has been applied.

Go to Project > Project Settings.

Set the Resolution to 'Power of two' and the dimensions to 1024 x 1024.

Note: The free version of World Machine is limited to 1024 x 1024. If you have a license for World Machine, set this to 4096x4096 to create a -64,-64x64,64-sized world space or larger.

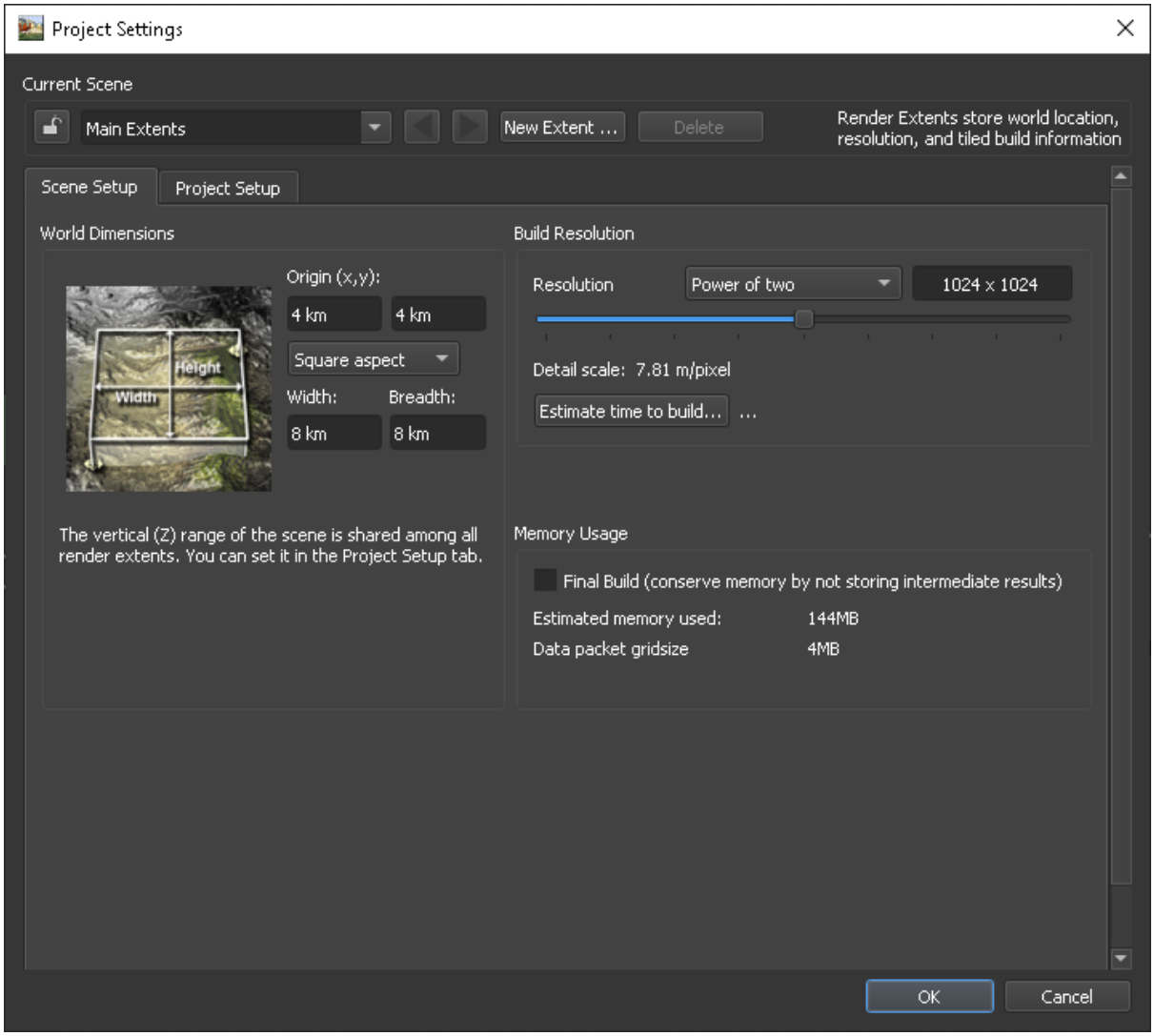


Figure 51 - Setting the build resolution.

Lastly, let's add a node to export our height-map.

Go to the Output tab, click on the Height Output button and add it to your workspace.

Connect the output of the Thermal Weathering node to the input of the Height Output node.

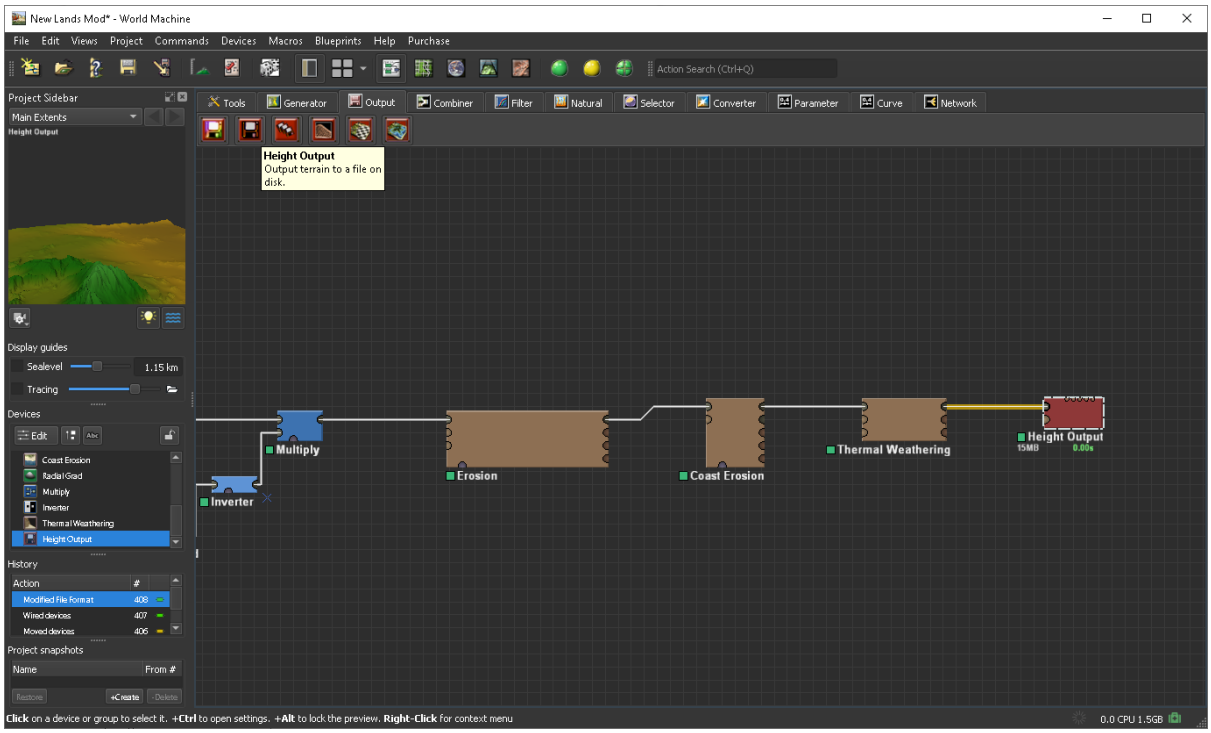


Figure 52 - Height Output node.

Double-click on the Height Output node to open its properties.

Set the File Format to 32bit Float RAW.

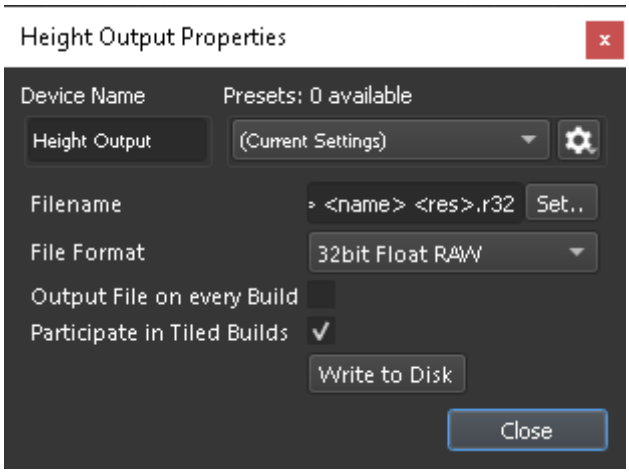


Figure 53 - Height Output properties.

Click Set.

Set the location to save the file to and the name of the .r32 file.

Ensure 'Save as type' is set to '32bit float RAW (*.r32)'.

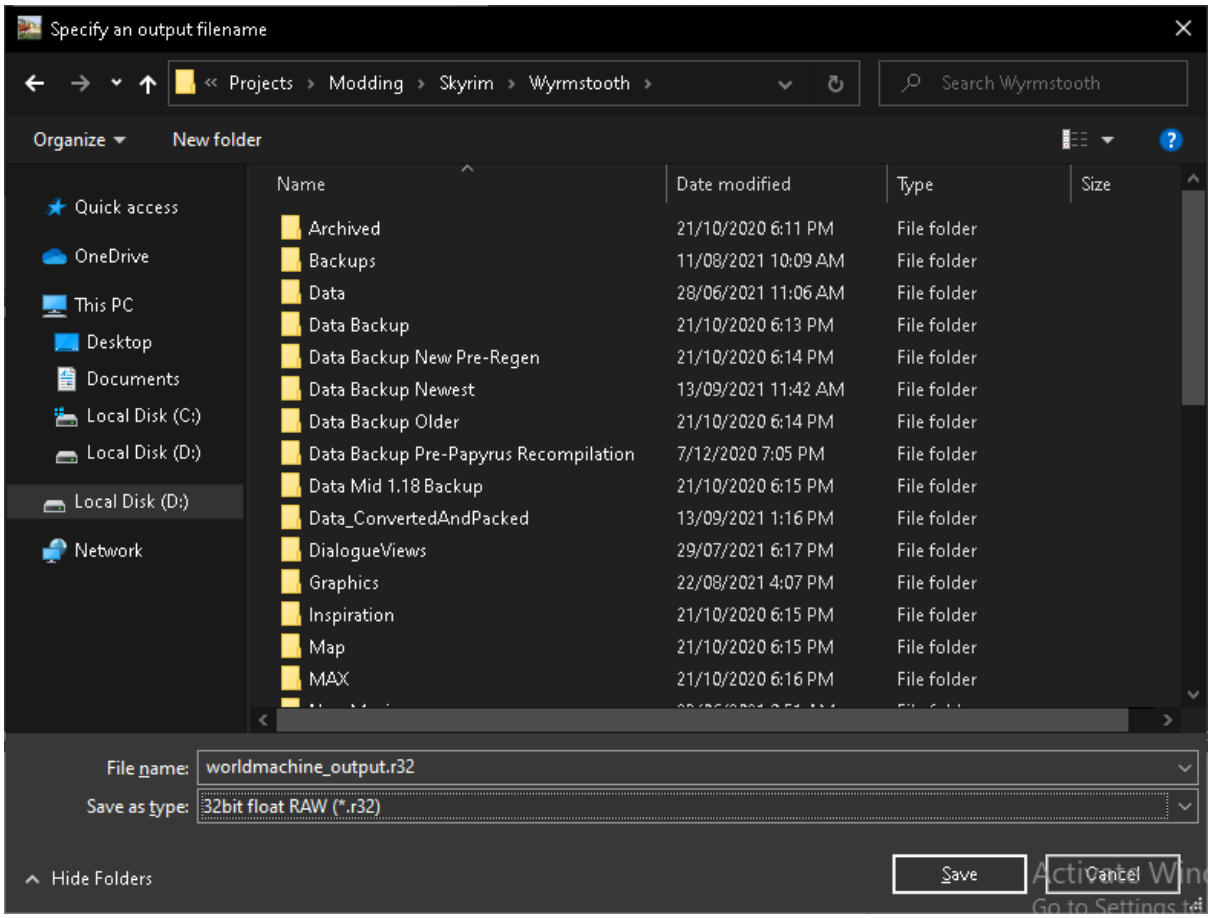


Figure 54 - Setting the location to save the .r32 file to.

Click Save.

Now click on the Write To Disk button.

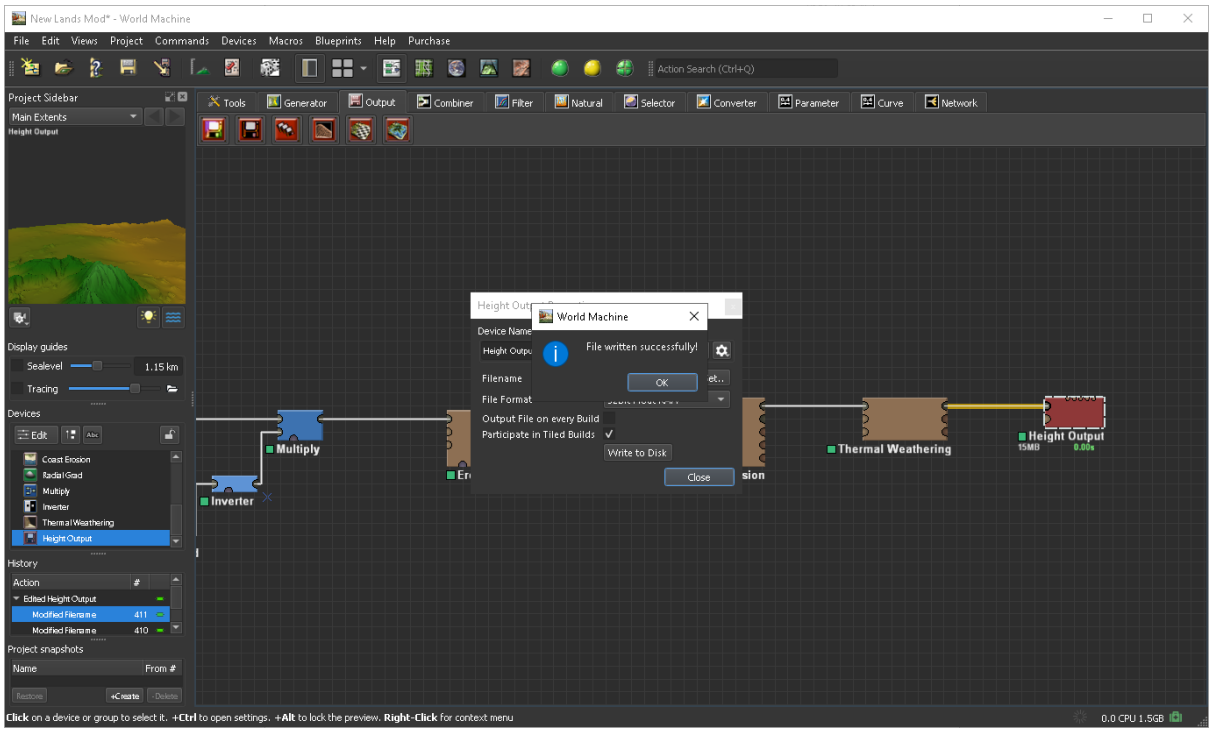


Figure 55 - Height map successfully outputted from World Machine.

Click OK to the confirmation pop-up.

Click Close to the Height Output properties.

Confirm that the .r32 file has been successfully saved to the directory you specified.

IMPORTING A HEIGHT-MAP WITH TESANNWYN

TESAnnwyn can be used to create a new .esp containing a world space based on a height map created in a terrain tool such as World Machine.

In this section I'll be importing the 1024x1024-sized height-map we created in World Machine previously.

TESAnnwyn is available for download from the Oceanlightwave website.

Extract the TESAnnwyn utility to a new folder and open a command prompt.

In the command prompt, navigate to the folder you exported TESAnnwyn to.

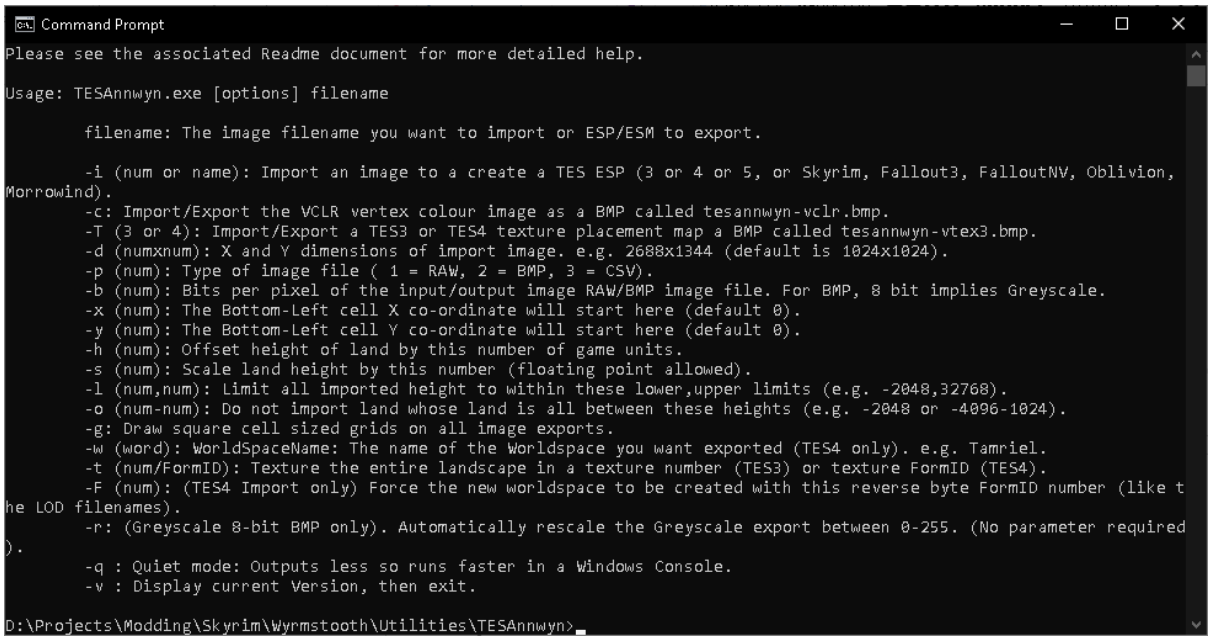


Figure 56 - TESAnnwyn options.

With TESAnnwyn, you can generate an .esp file, creating a world space based on your height map with the following command:

```
TESAnnwyn.exe -i Skyrim -p 1 -b 32 -d 1024x1024 -x -16 -y -16 -h -416000 -s
0.00005 "D:\Projects\Modding\Skyrim\Wyrmstooth\worldmachine_output.r32"
```

Replace the folder path and filename in quotes with the folder path and filename to your .r32 or .raw file.

Just a few words on the values I used in that command.

The -h switch specifies the height offset of the height map. I set this to -416000 so the land intersects the ocean plane roughly where it's supposed to. Without this switch, the height map would be far in the sky. You can adjust the height of the land and water in the Creation Kit under World > World Spaces manually, but I'd prefer to do this in TESAnnwyn at import instead.

The -s switch was set to 0.00005 to limit vertical stretching. Finding this value takes a bit of trial and error. Too low and your landscape will be completely flat. Without the -s switch, the height map is most likely going to look like jagged spikes as per the screenshot below:

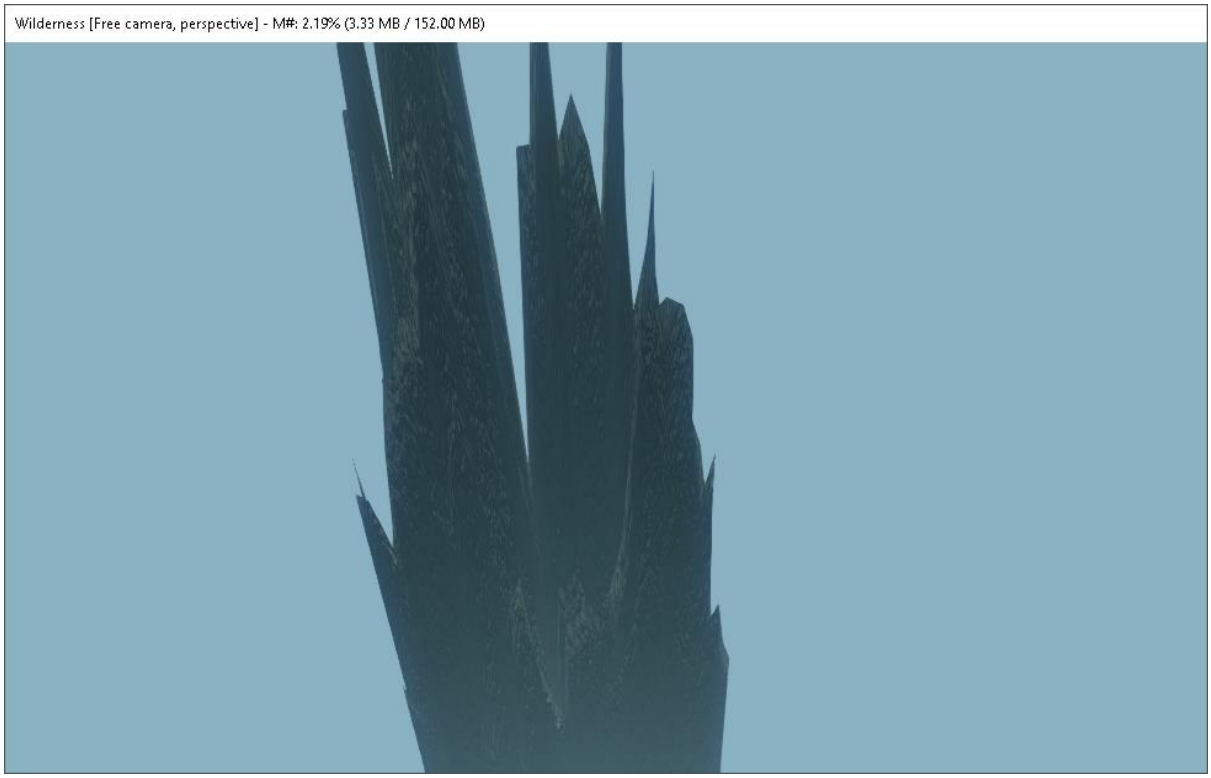


Figure 57 - No height scaling.

-x and -y set the origin cell to -16,-16. Seeing as though we're working with a small height map exported from World Machine, the size of our world space ended up being -16,-16 to 15,15. Without -x and -y, the cells will start at 0,0.

Note: You can always manually sculpt the terrain and add new cells if you want to expand the size of the island and the world space.

Lastly, the -d switch specifies the dimensions of the height map we're using. I set this to 1024x1024 as this is the size of the height map exported by World Machine. While you can double this to 2048x2048 to create a -32,-32,32,32-sized world space or 4096x4096 to create a -64,-64,64,64-sized world space using a 1024x1024-sized height map, TESAnnwyn doesn't seem to handle this particularly well and your height map will end up looking stretched on one axis.

Important: If you want a -32,-32,32,32-sized world space, use a 2048x2048-sized height map. If you want a -64,-64,64,64-sized world space, use a 4096x4096-sized height map.

For more information on the TESAnnwyn command line options, see the TESAnnwyn readme file.

When done, a new .esp file named 'tesannwyn.esp' will be created in the folder you extracted the TESAnnwyn utility to.

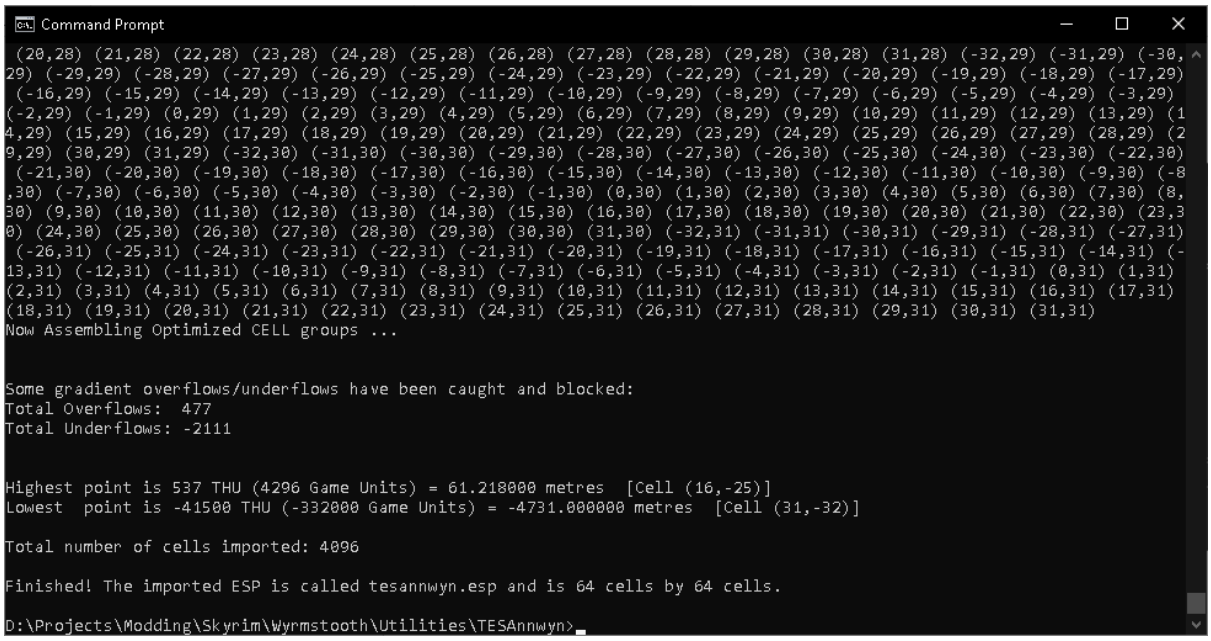


Figure 58 - TESAnnwyn finished.

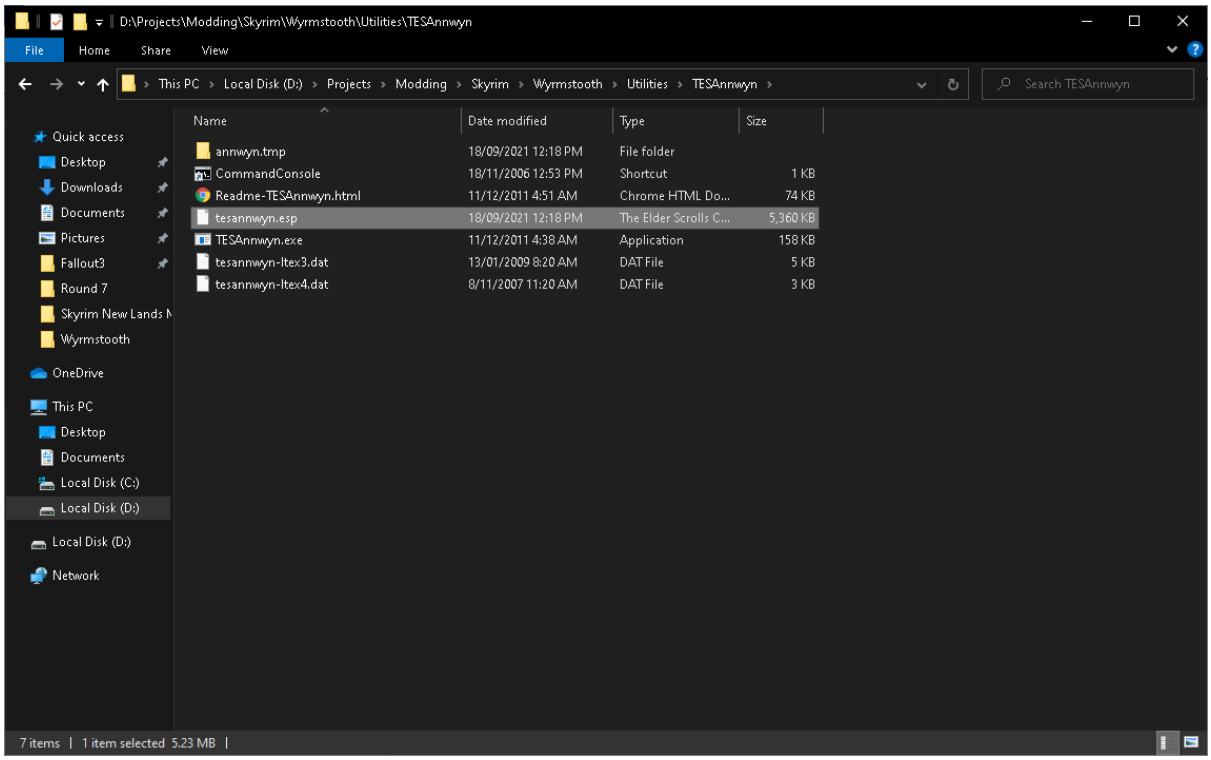


Figure 59 - tesannwyn.esp.

Copy this file to your Skyrim\Data or Skyrim Special Edition\Data folder.

You can rename this file however you like.

Launch the Creation Kit.

Go to File > Data.

Tick the Skyrim.esm file in the Plugin/Master Files list.

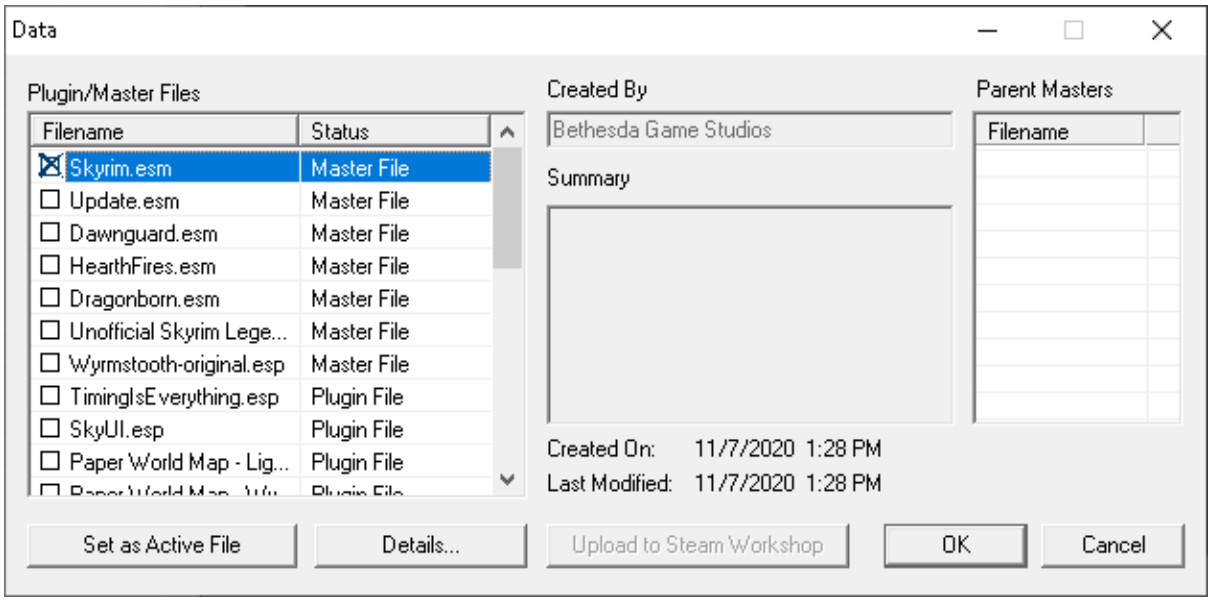


Figure 60 - Ticking the Skyrim.esm master file.

Tick the .esp file created by TESAnnwyn and click 'Set as Active File'.

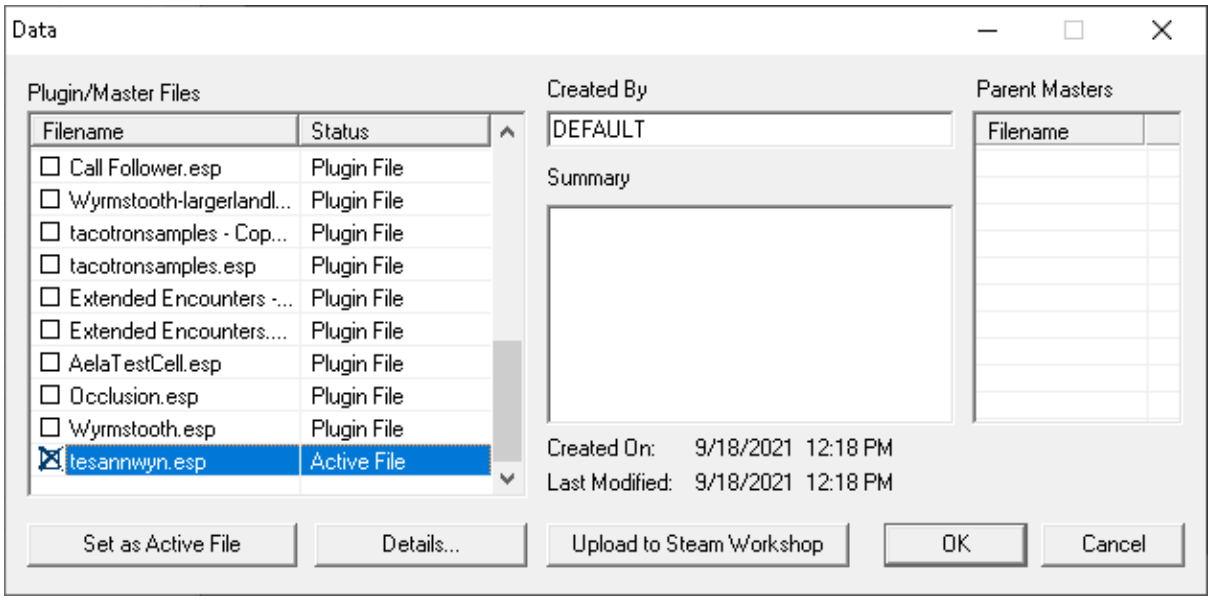


Figure 61 - Setting the tesannwyn output .esp as the active file.

Click on the OK button to begin loading it.

Go to World > World Spaces.

You should see a TESAnnwyn world space listed.

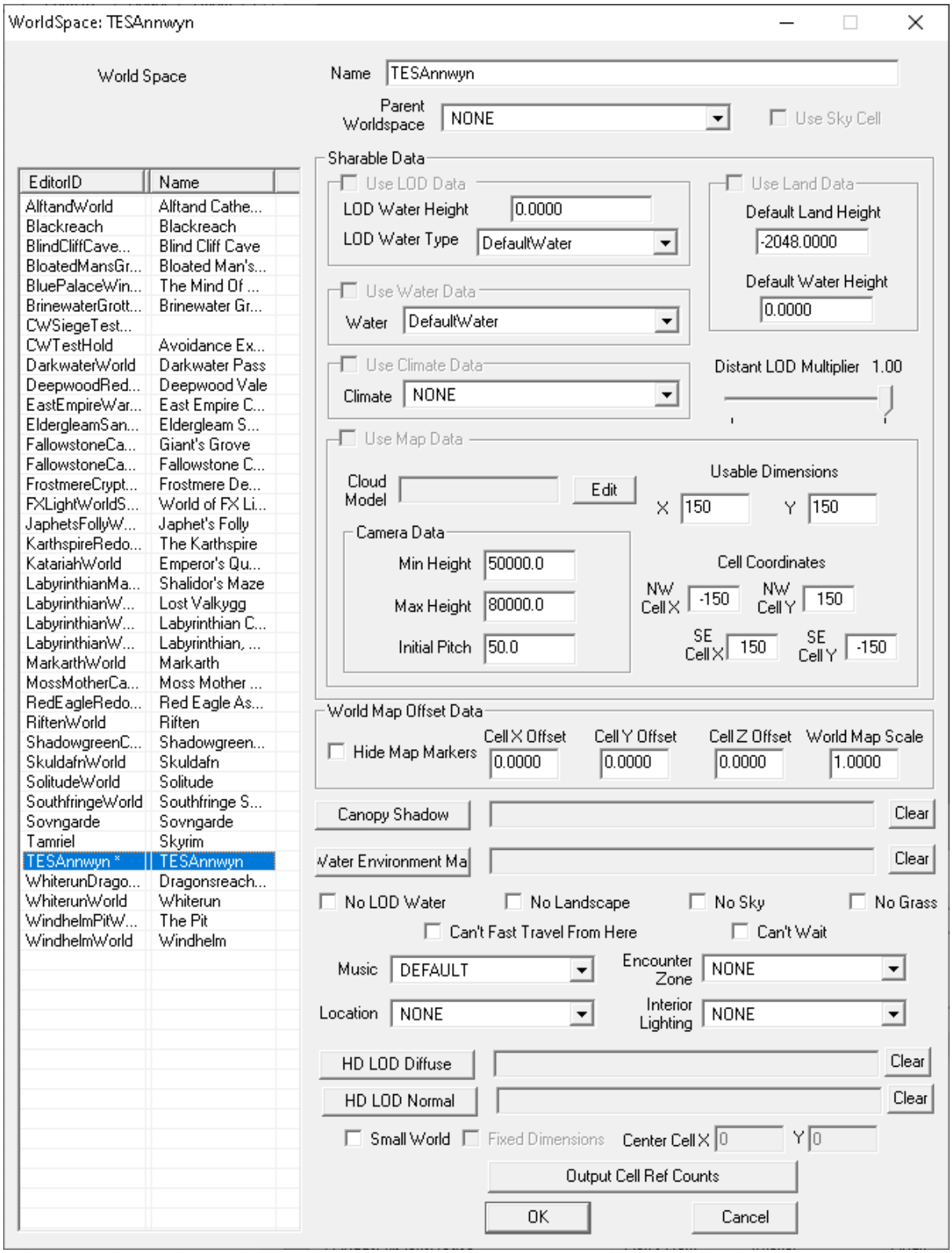


Figure 62 - TESAnnwyn world space.

Press 'F2' and rename this world space to whatever you like, then click OK.

In Cell View, select your world space then double-click on one of the cells to load it.

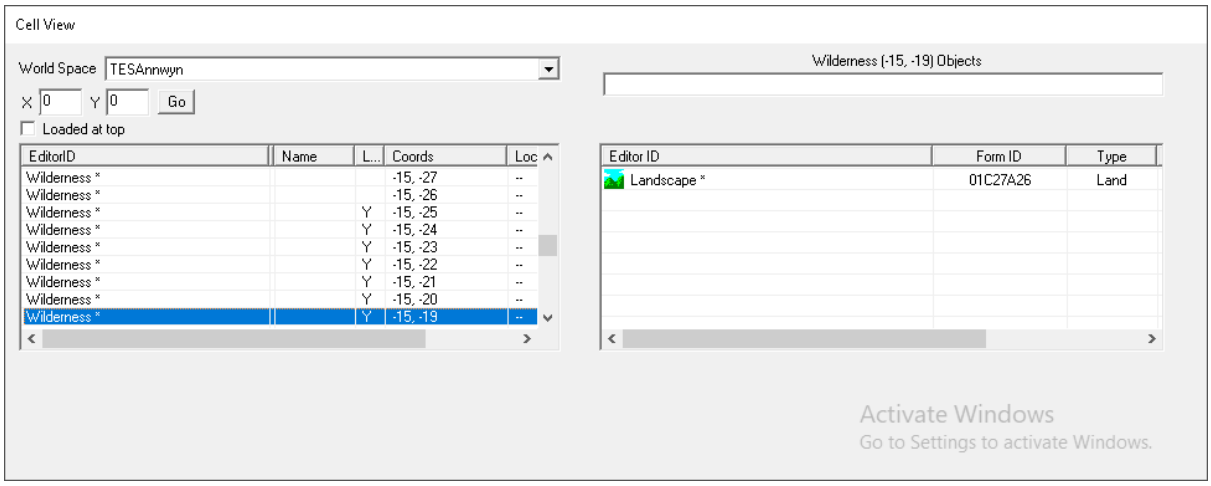


Figure 63 - Loading the world space.

And there's our height map in the render window.

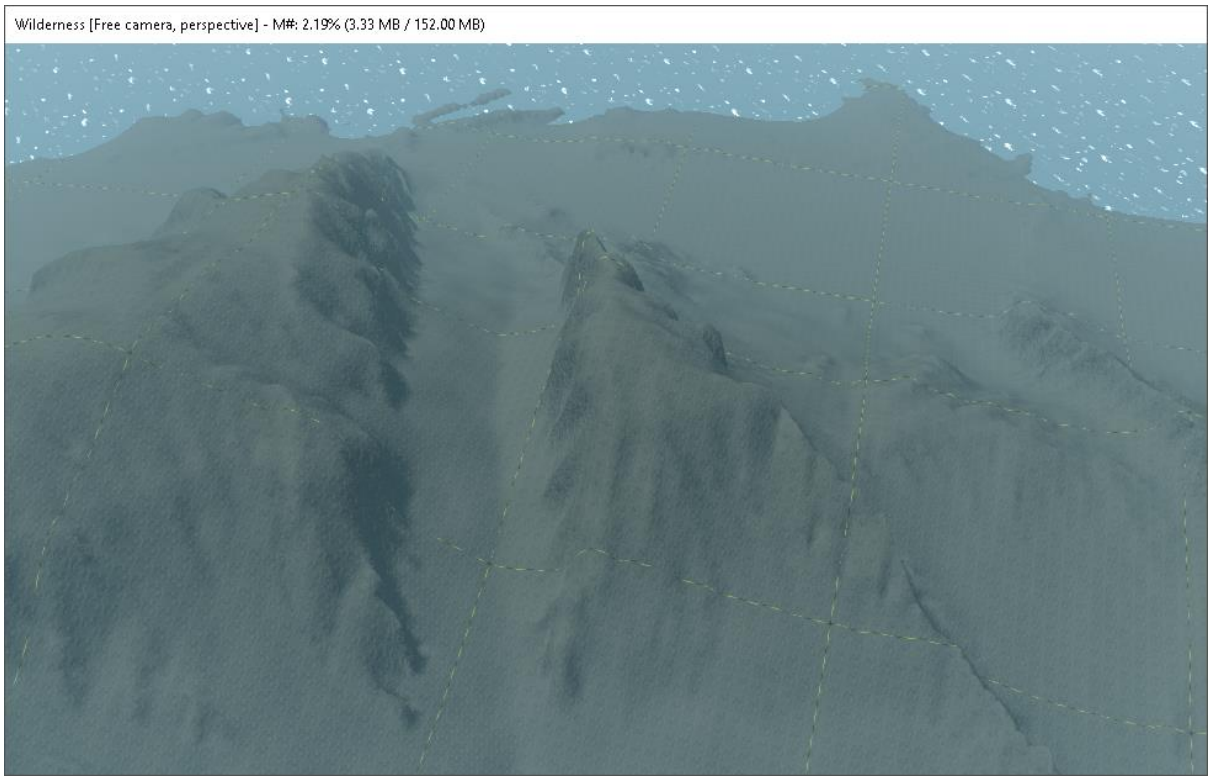


Figure 64 - Height map in the render window.

Note: If you see holes in your landscape, your height map may be too steep. Reduce the value in the -s switch in TESAnnwyn further to reduce the steepness of the mountains.

SCULPTING A LANDSCAPE WITH THE LANDSCAPE EDIT TOOL

The landscape edit tool is the tool we'll be using to manually sculpt the landscape in our world space. This tool can be used to raise or lower our terrain and flatten or smoothen slopes. Later on we'll also use it to paint landscape textures.

Press 'H' to bring up the landscape edit tool.

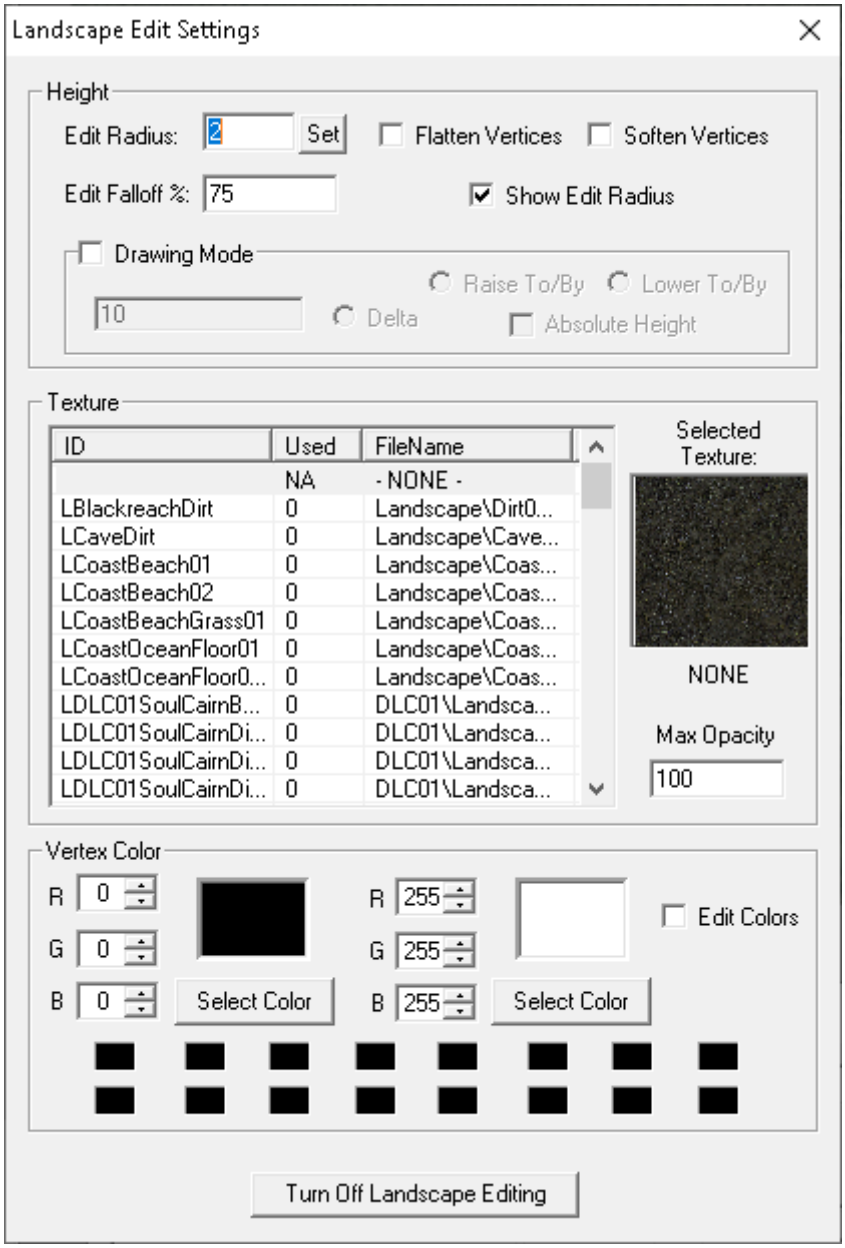


Figure 65 - The Landscape Edit tool.

The Edit Radius shows the current radius of the landscape edit tool. This the red circle that appears around the mouse cursor in the render window when the landscape edit tool is enabled.

The Edit Falloff represents the hardness of the brush. It's like brush hardness in image editing programs. The lower the number the softer the edge.

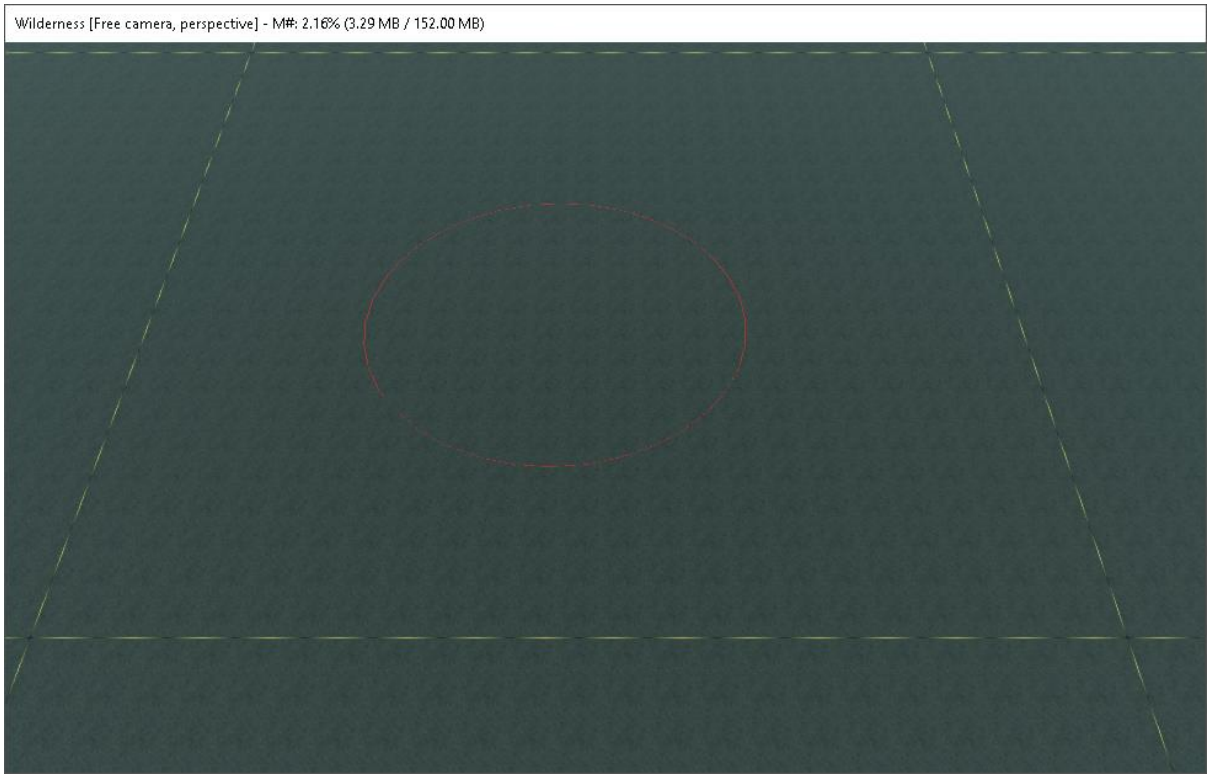


Figure 66 - Edit Radius.

Set the Edit Radius to 9.

Note: You can also use the '[' key to reduce the Edit Radius by one and ']' to increase the Edit Radius by one.

In the following examples, I'll be picking up where I left off in the first chapter.

At the moment our landscape is still entirely underwater. We'll need to see the landscape as we're sculpting, so press 'M' to bring up the Show/Hide menu and untick Water to hide it for now.

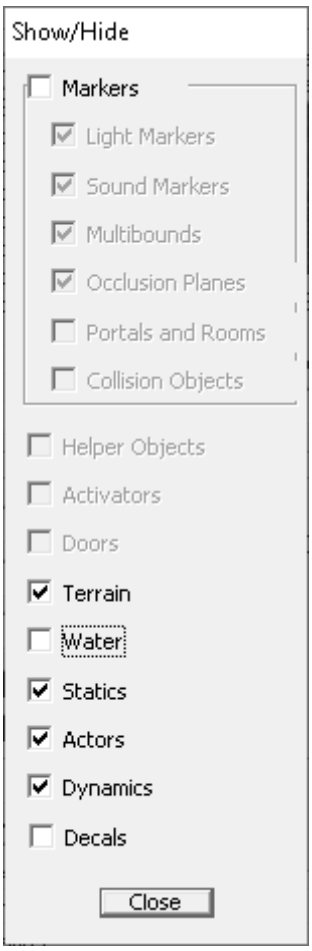


Figure 67 - Turning water rendering off.

Let's start by raising some land up from the sea floor. Left-click and drag the terrain up in the render window.

We should now have a bit of a mountain sticking up from the sea floor.

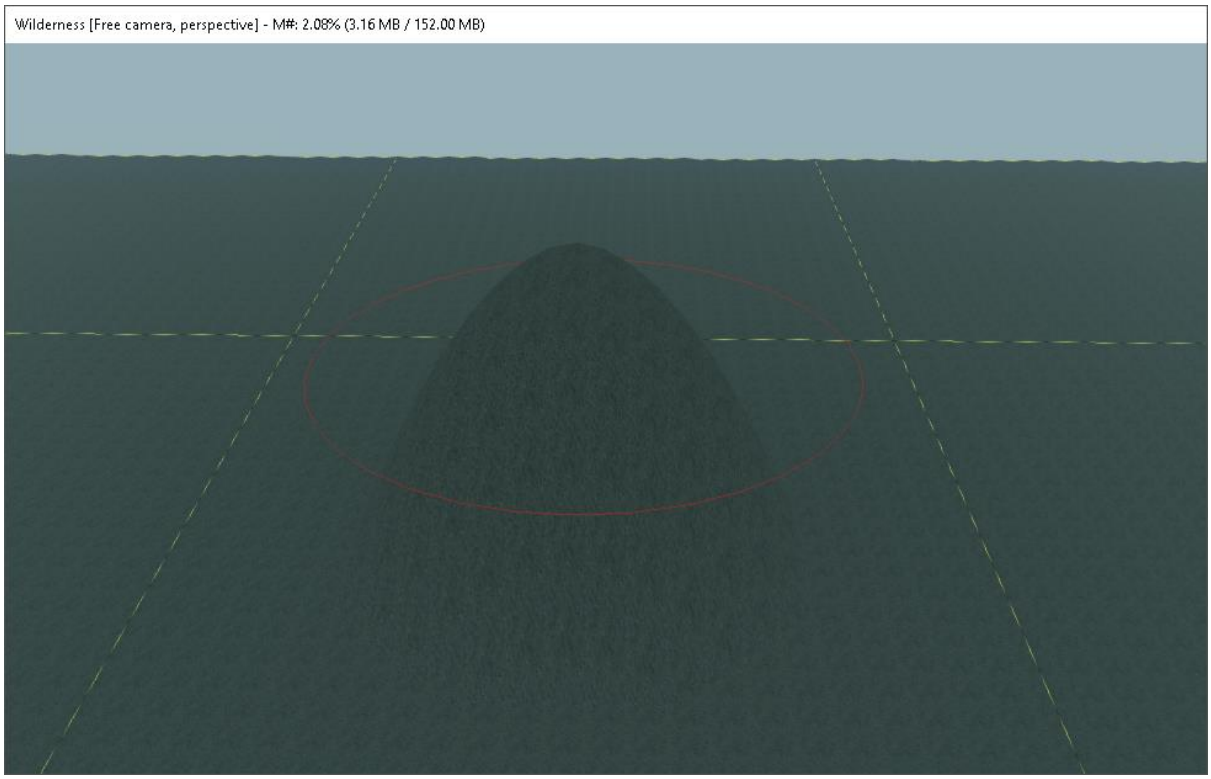


Figure 68 - Dragging terrain up.

Note: If you see floating grass or rocks up in the sky, just ignore it. This is a bug.

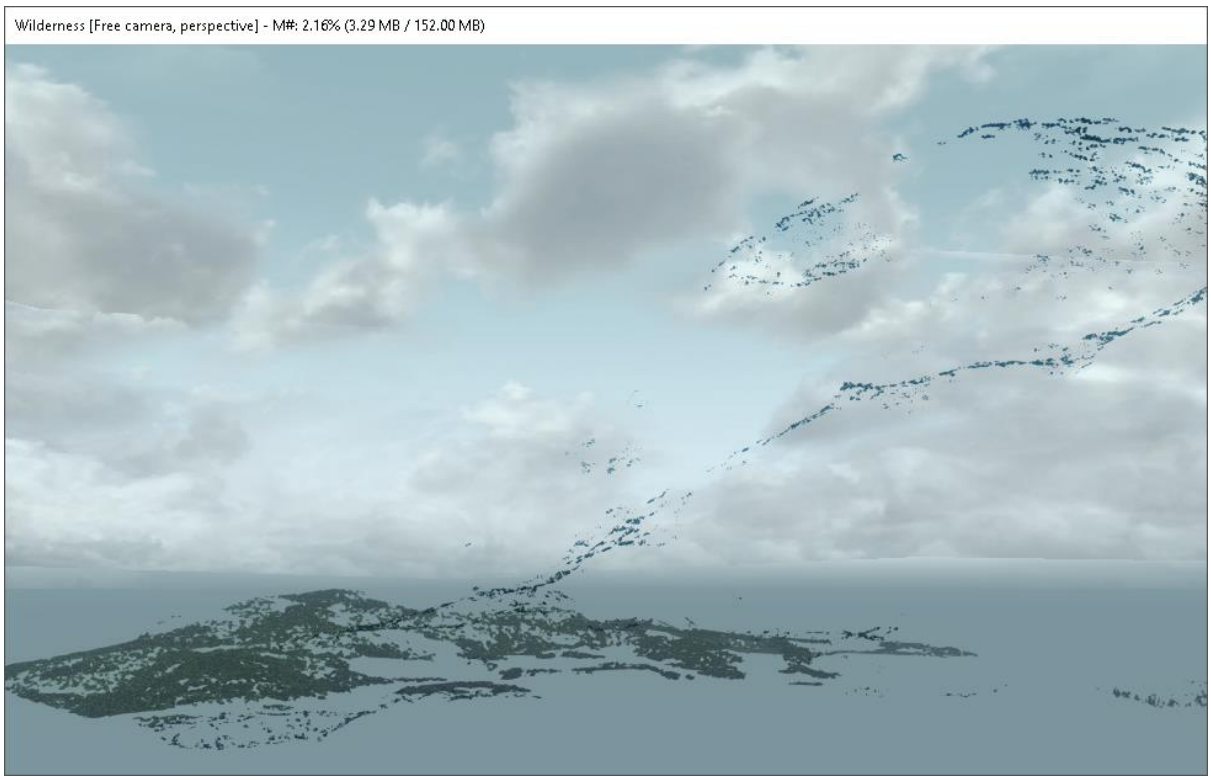


Figure 69 - Ignore the floating rocks and grass.

If we turn water back on in Show/Hide, our render window should now look like this:



Figure 70 - A bit of land poking through the ocean.

Let's increase the size of our shoreline next.

Important: If the Creation Kit crashes when editing your world space for the first time, make a small adjustment to the landscape first, then save your mod. If you press 'B' to show cell borders, but cell borders aren't being overlaid on the landscape, you might be in for a crash.

Set the Edit Radius to 4 and tick 'Flatten Vertices'.

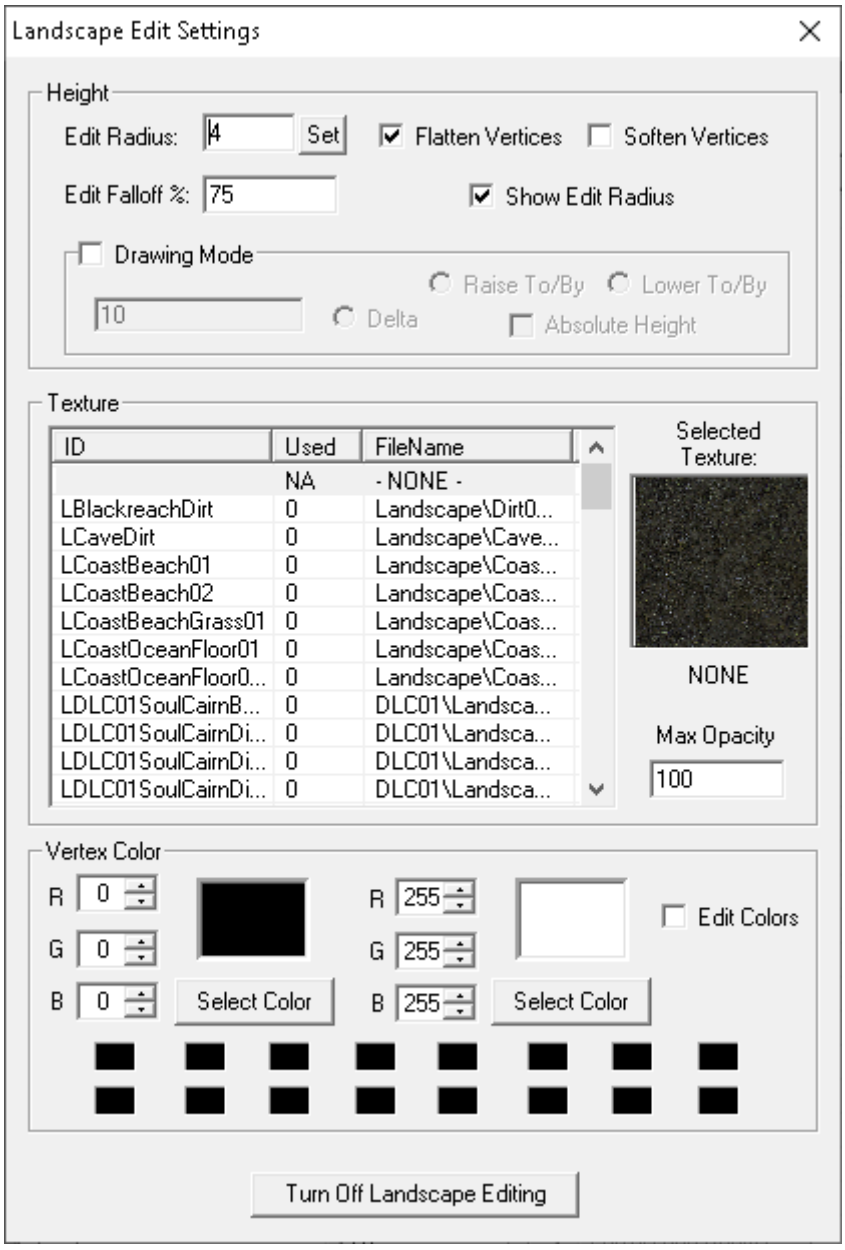


Figure 71 - Turning on Flatten Vertices mode.

Left-click and drag on the land poking above the water.

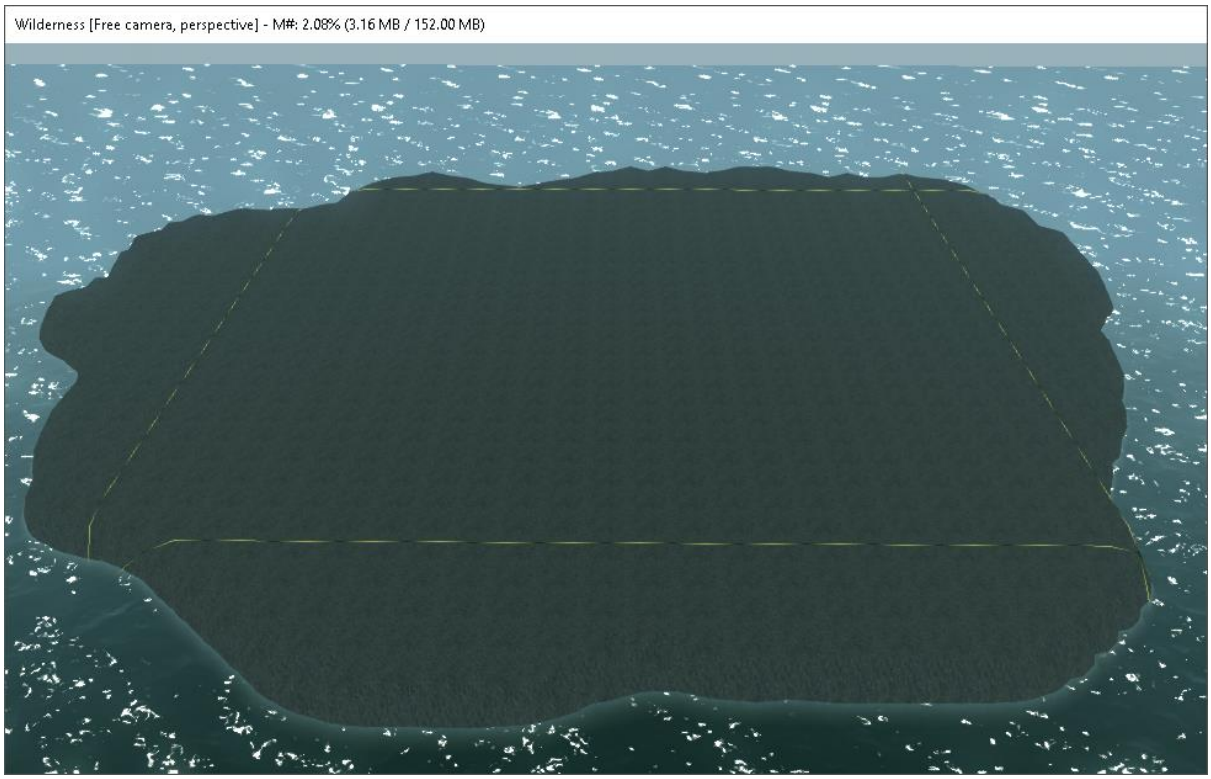


Figure 72 - We now have a bit of an island.

When you left-click and start dragging with Flatten Vertices ticked, the height of the landscape you drag over will be set to the same level as the part of the landscape you began dragging from.

Anything below the ocean will be raised up, similarly, any hills above the point where you began dragging from will be lowered. We can use Flatten Vertices to quickly create the outline of an island.

If we turn off water again, we'll see that the slope from the ocean floor to the top of our island is rather steep.

So let's smoothen that out a bit.

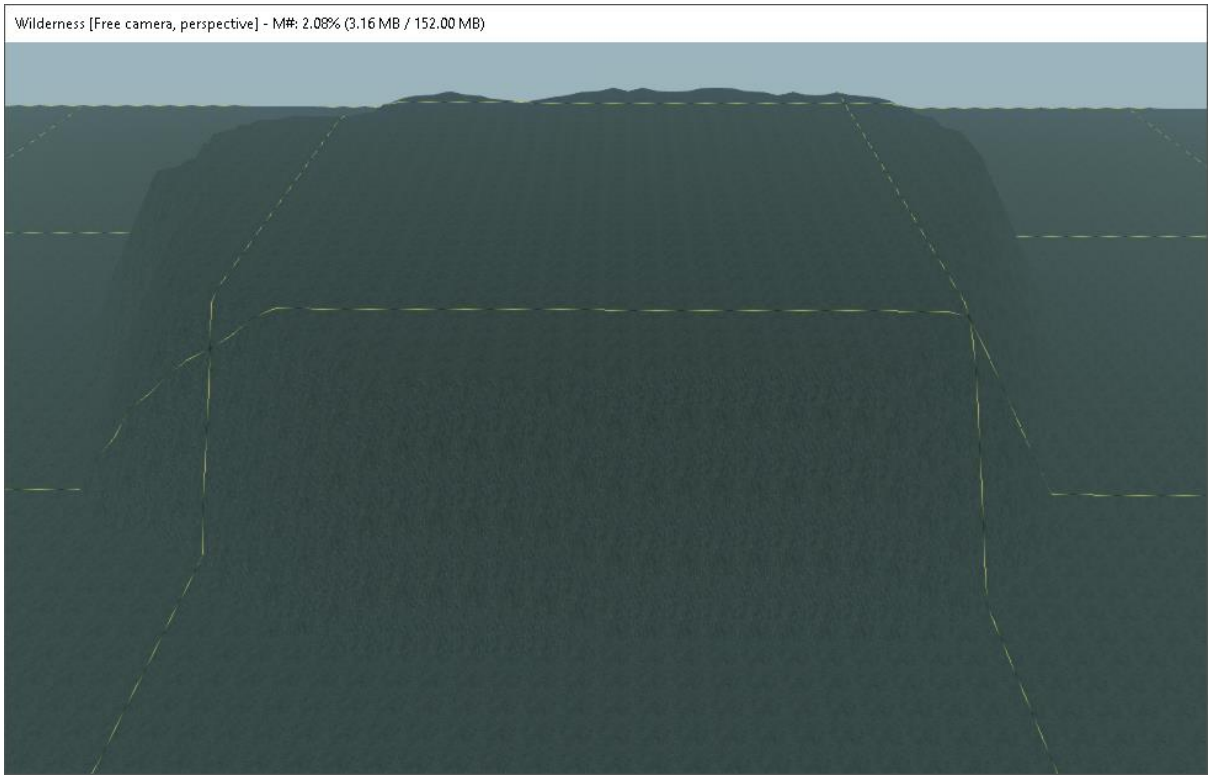


Figure 73 - Steep slopes from ocean floor to island surface.

In the landscape edit tool, tick 'Soften Vertices'. This will automatically untick 'Flatten Vertices' and vice versa.

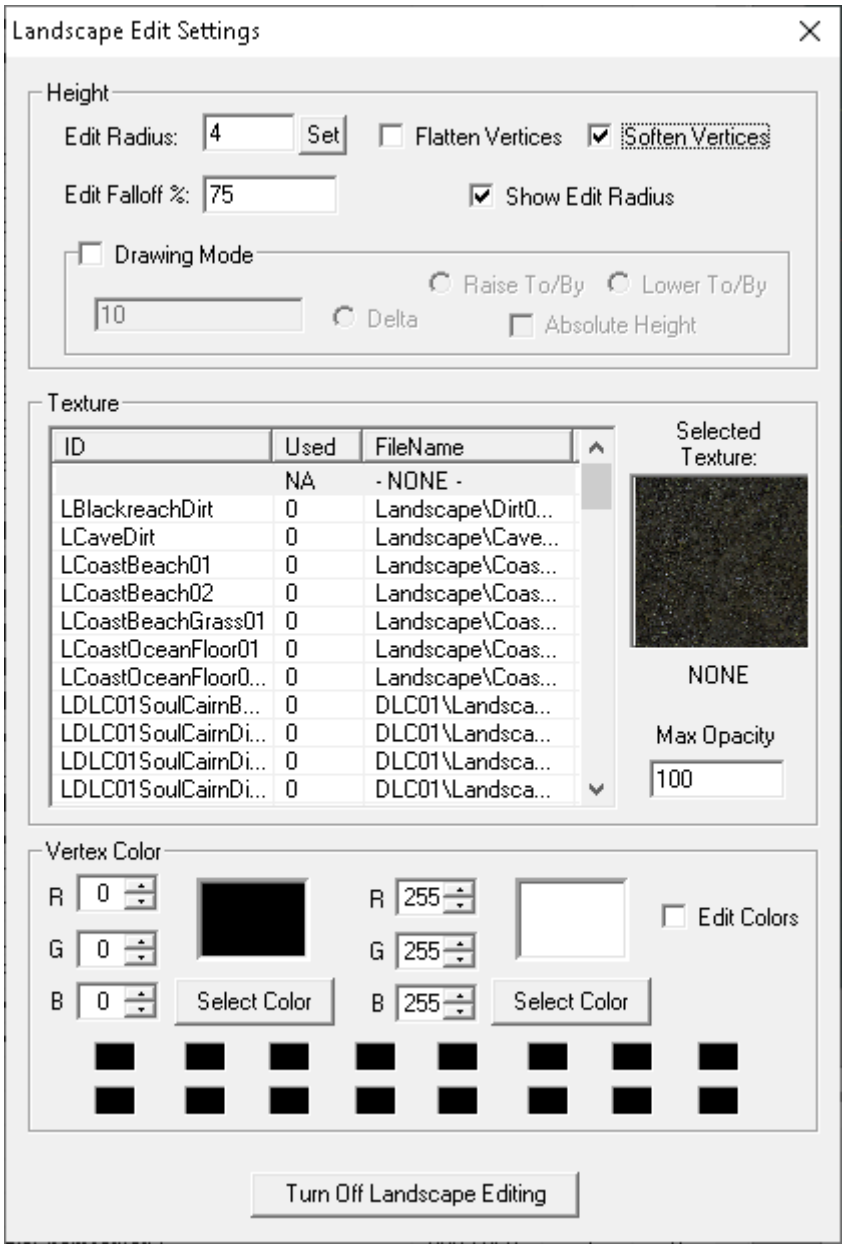


Figure 74 - Turning on Soften Vertices mode.

Left-click and drag along the slope to reduce its steepness.

When softening the slope, try not to do it too evenly so it looks a bit more natural.

Our island should now look a bit more like this.

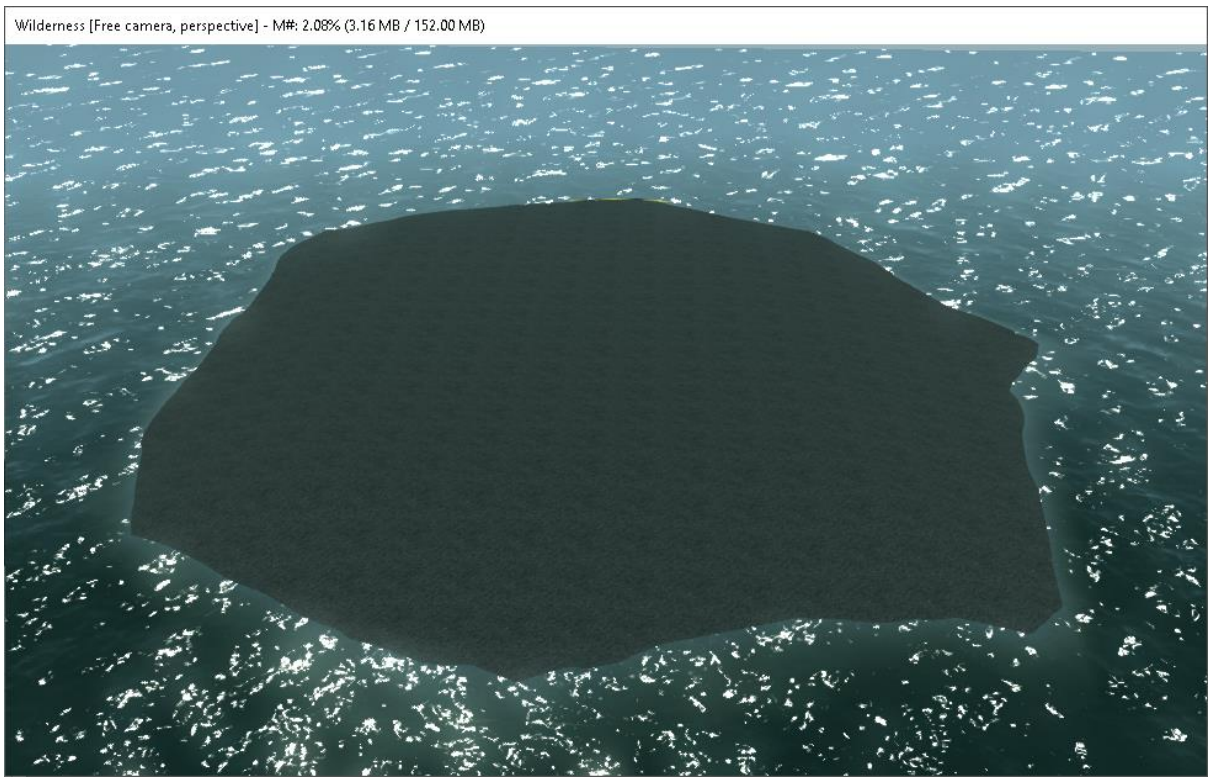


Figure 75 - Softened slopes around the shoreline of our island.

Important: When sculpting with the landscape edit tool, I would not recommend setting the Edit Radius above 14. It might lead to issues panning around your scene and the edit radius indicator might disappear.

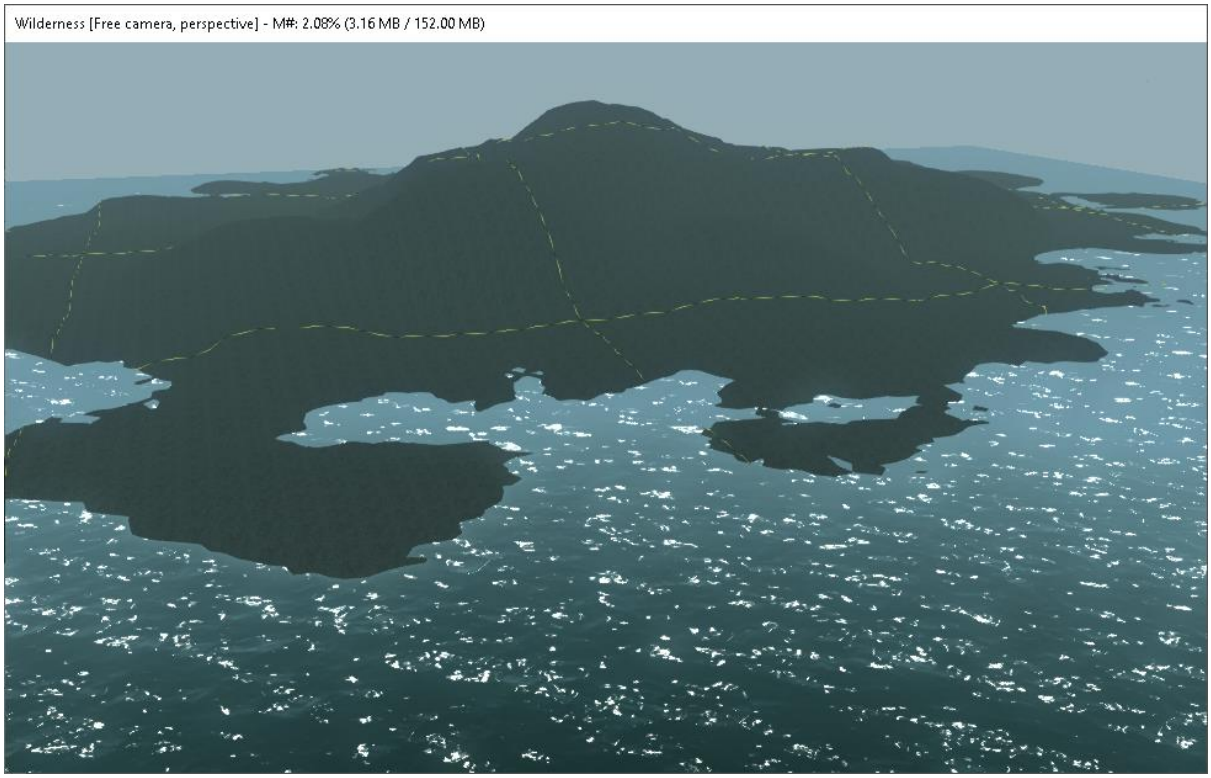


Figure 76 - The island gets a bit bigger.

In the screenshot above I added some flat areas near the shoreline around a central hill; enough room perhaps for a small town. Some tidal pools and small islands were added around the coast of our main island to help make the shoreline look a bit more natural.

I also added some shallows for the player to stand in, before the landscape drops off to the ocean floor.

To better understand the scale of your landscape as you sculpt it, you can toggle cell borders by pressing the 'B' key.

While sculpting your island, I'd also recommend increasing the number of cells loaded.

Go to File > Preferences and go to the Misc tab.

Set the 'Grids to Load' to 9 then click Apply.

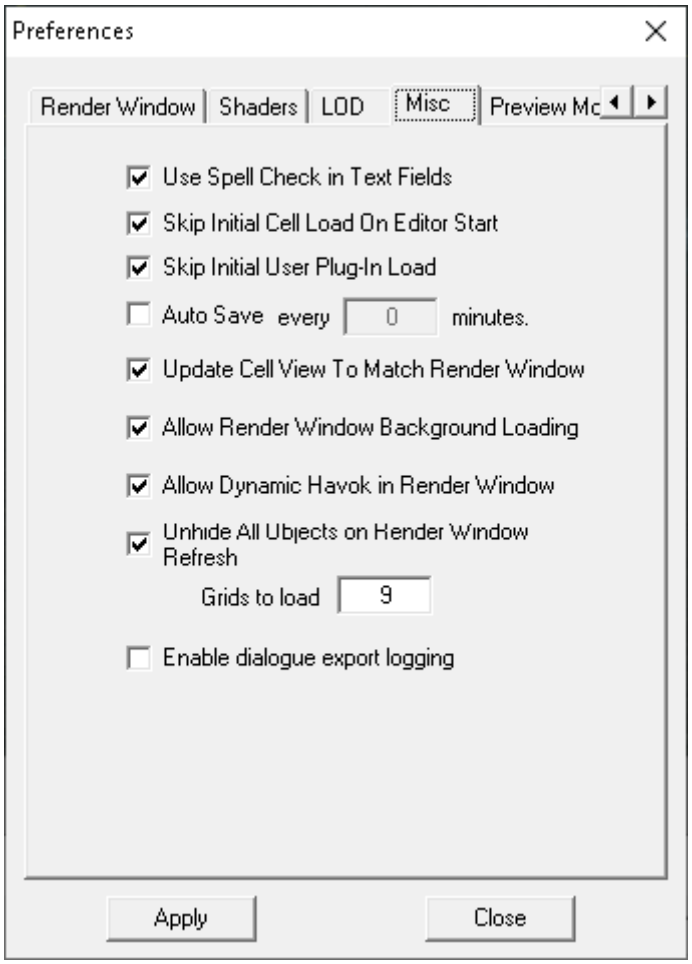


Figure 77 - Increasing Grids to Load.

This will increase the number of cells loaded into the preview window. Just be careful with how high you set this value.

Important: Setting Grids to Load too high may inadvertently crash the Creation Kit, especially with a fully cluttered landscape.

ADDING LANDSCAPE TEXTURES

Landscape textures can also be added using the landscape edit tool.

Important: If you want to learn how to automatically paint your terrain based on slope, altitude and object placement, see Region generated textures and objects.

Press 'H' to bring up the landscape edit tool if you don't already have it open and press 'B' to toggle cell borders.

Before we start adding textures to a landscape, it's important to consider how each cell is divided and the texture limitations we'll need to be aware of.

Each cell consists of 4 quads. The screenshot below should help illustrate this:

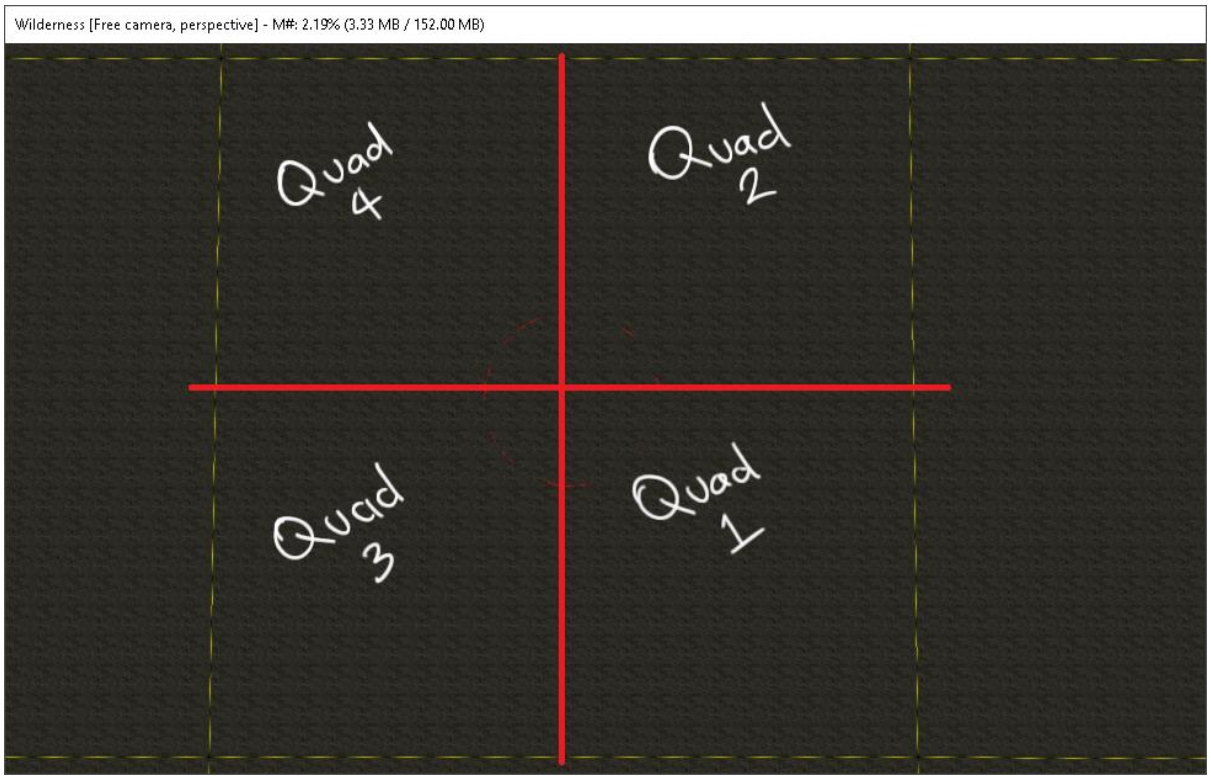


Figure 78 - Quads in a cell.

Important: Each quad can have up to **6** different textures.

Press 'I' when the landscape edit tool is active to see a list of textures that each quad of the currently loaded cell has.

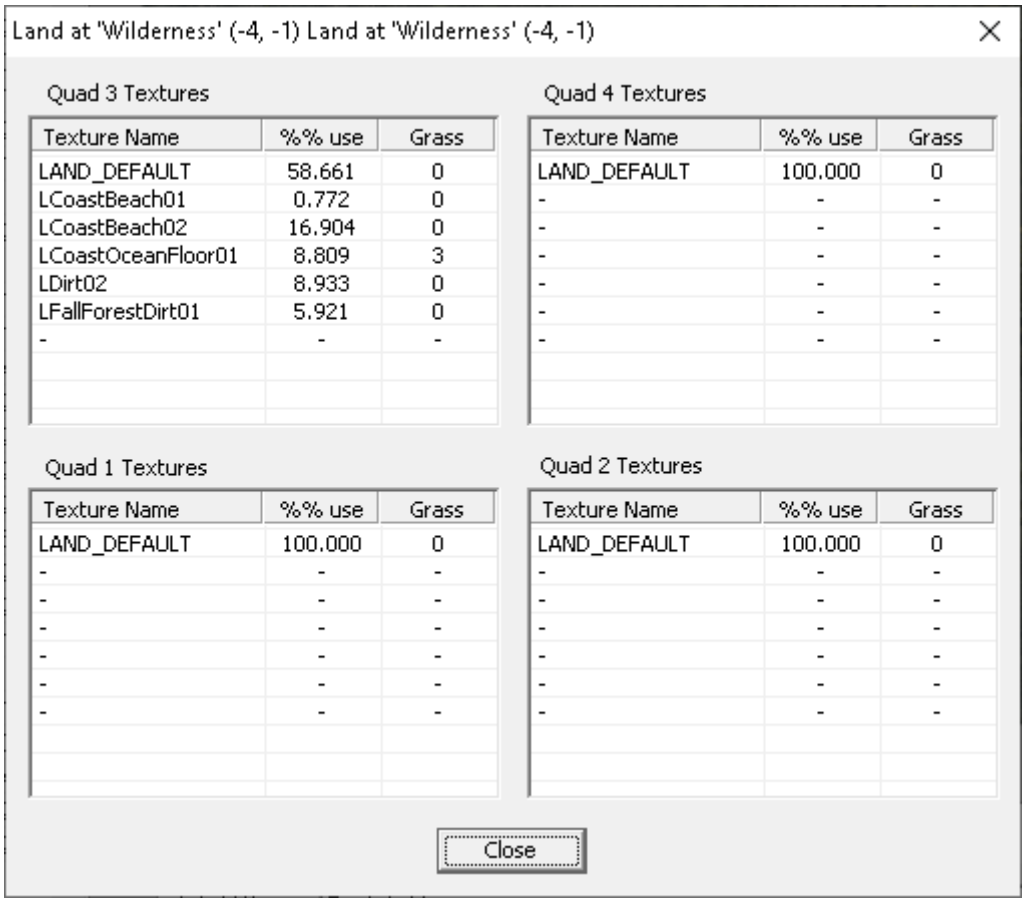


Figure 79 - Textures applied to each quad of the currently loaded cell.

In the screenshot above, you can see that I currently have the cell '-4,-1' loaded. I've been painting Quad 3 with five different textures: LCoastBeach01, LCoastBeach02, LCoastOceanFloor01, LDirt02 and LFallForestDirt01. If we include the default landscape texture, that's six separate landscape textures in quad 3.

If you try to add a seventh texture or more, you'll simply be painting with the currently selected vertex colour which by default is black. And yes I'm aware there are seven slots in the texture quad screen.



Figure 80 - Painting vertex colour.

In the example above, I tried painting with a seventh texture LFrozenMarshLichen01 in Quad 3 which resulted in a flat black texture being painted instead.

We can fix this by right-clicking on LFrozenMarshLichen01 and selecting Delete.

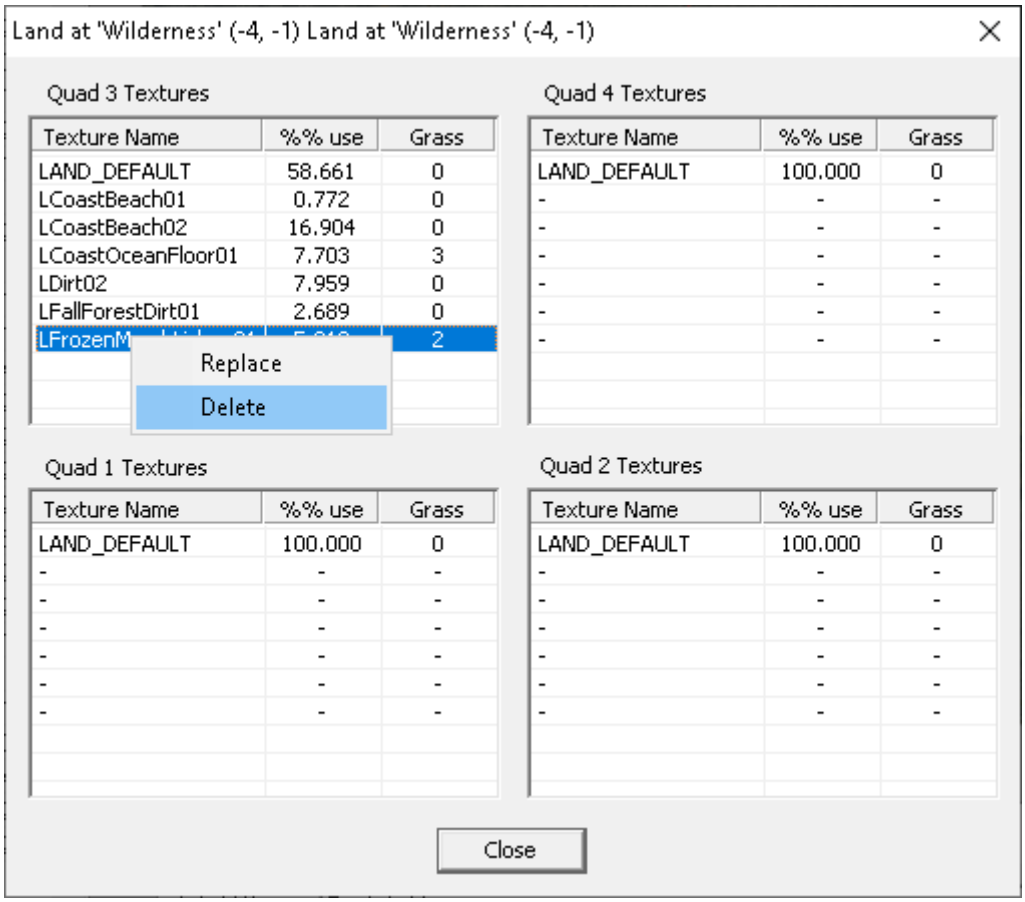


Figure 81 - Deleting the seventh texture.

Let's start painting our landscape.

In the landscape edit tool, you can select the texture you want to paint onto your landscape in the Texture list. In the following example, I'll be painting with LCoastBeach01 to paint along the coastline of our landscape.

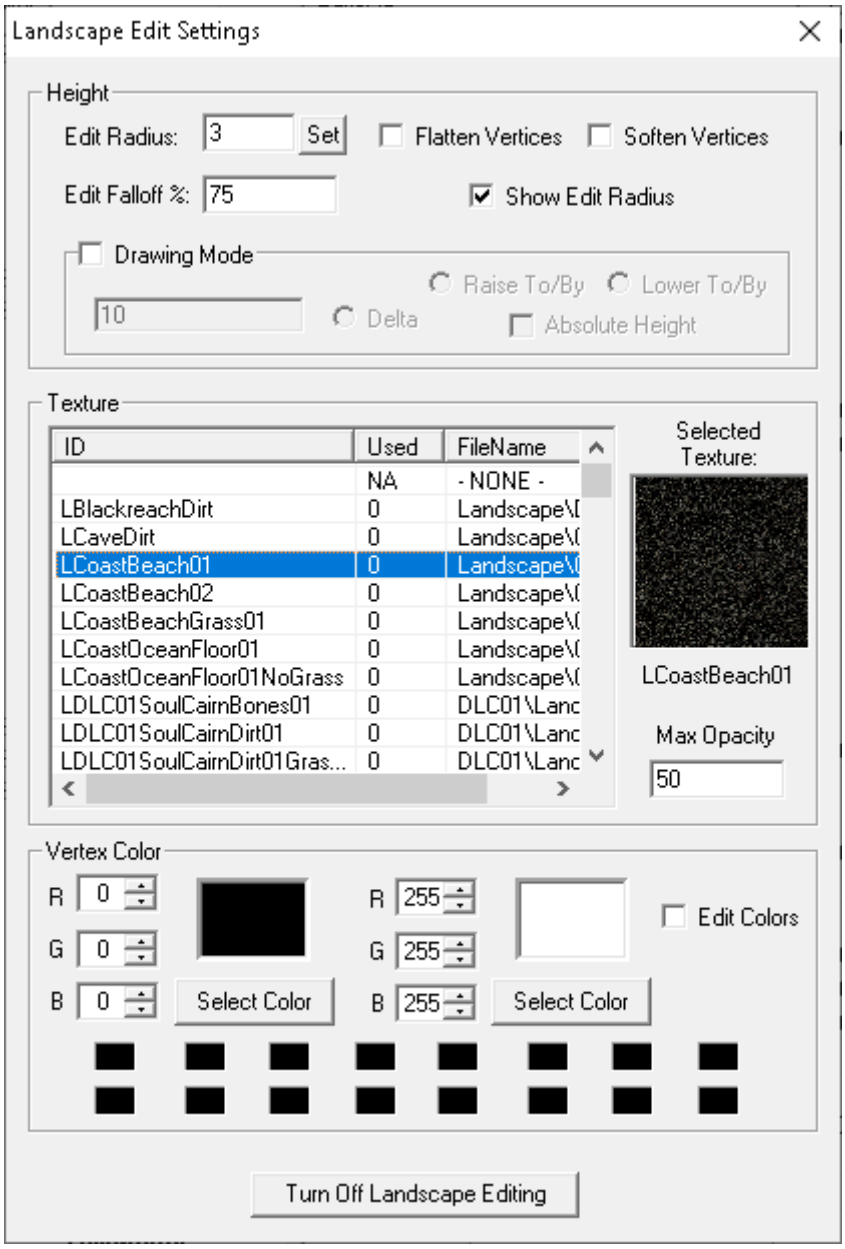


Figure 82 - Selecting our texture and reducing Max Opacity.

I also reduced the Max Opacity to 50. This will help us blend the LCoastBeach01 texture with the existing default landscape texture.

The size of your paintbrush is determined by the Edit Radius.

To erase a texture from your landscape, you can either press CTRL + Z to undo or select the blank texture shown at the top of the Texture list which acts as an eraser.

Right-click on the landscape with the landscape edit tool active to begin painting with the currently selected texture.

Remember: Left-click to sculpt, right-click to paint.

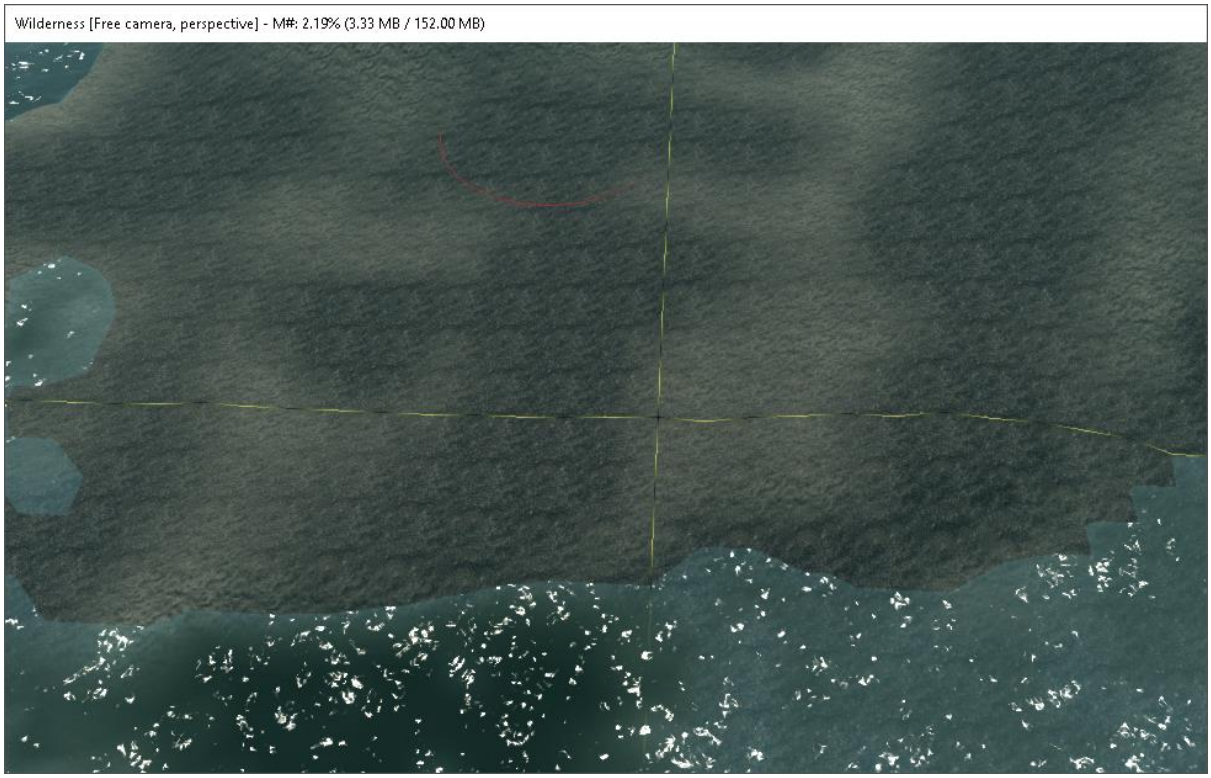


Figure 83 - Painting LCoastBeach01 around the coastline.

Rather than painting the landscape uniformly, I like to right-click here and there to try and blend the textures together in a kind of splotchy manner.

In the following screenshot, I painted a section of the coastline with a single texture, and as you can see, the tiling pattern of the texture that I used is glaringly obvious.

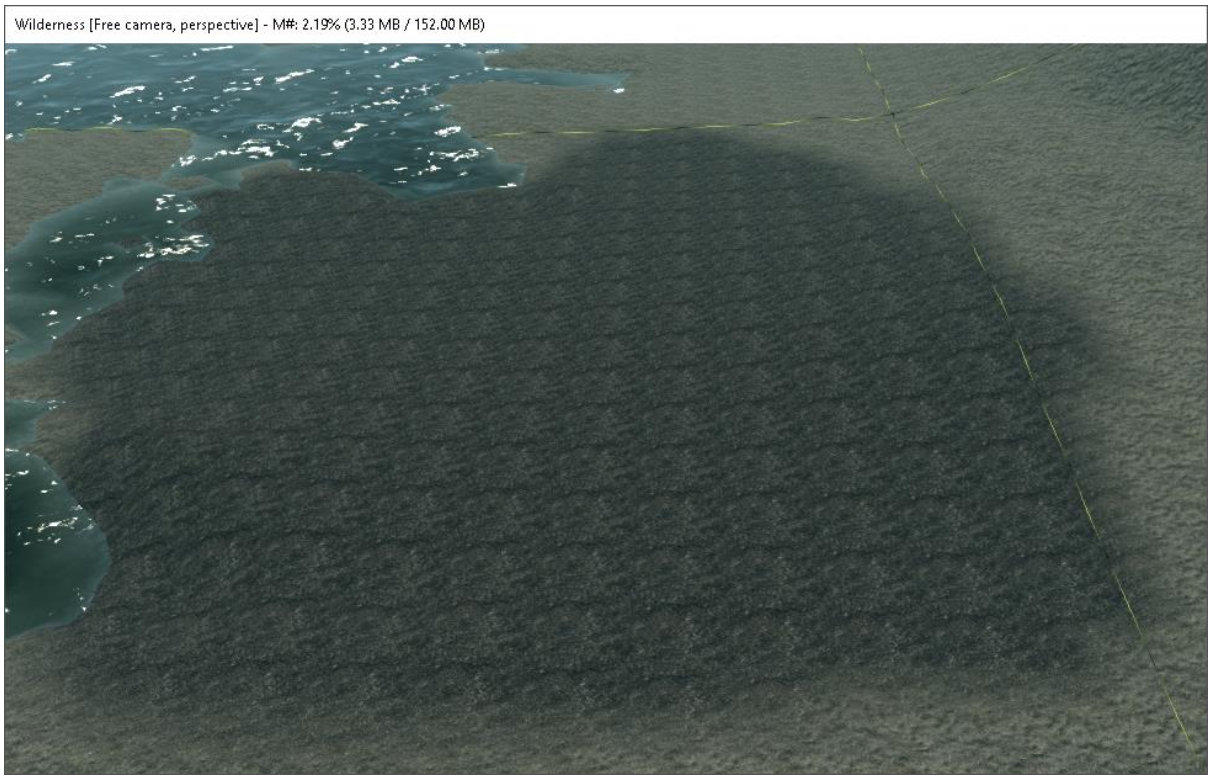


Figure 84 - Texture tiling.

But we can use other textures to help break up that repeating pattern.

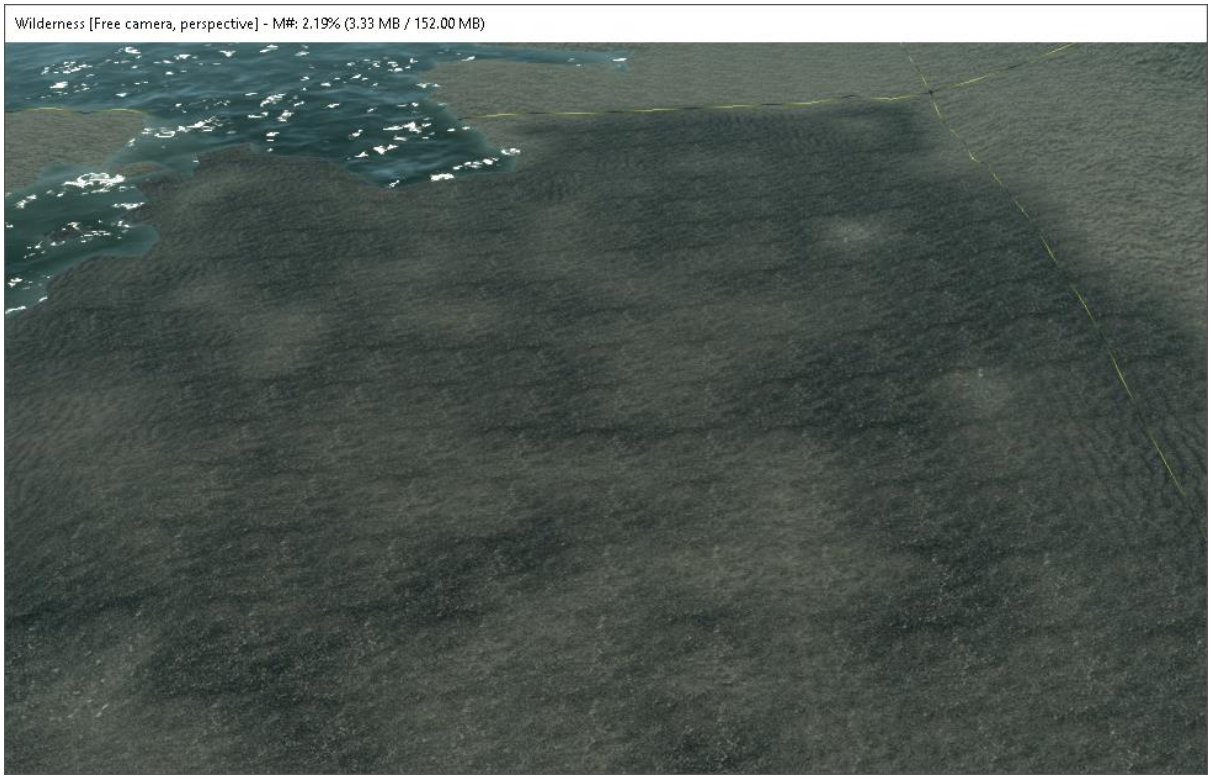


Figure 85 - Breaking up texture tiling.

Some textures such as LCoastBeachGrass01 or LRocks01 will also add grass or small rocks to your landscape and will further help hide that tiling effect.



Figure 86 - Adding textures with grass and rocks.

If you chose the wrong texture, you can replace a texture in a quad by right-clicking on it and choosing Replace.

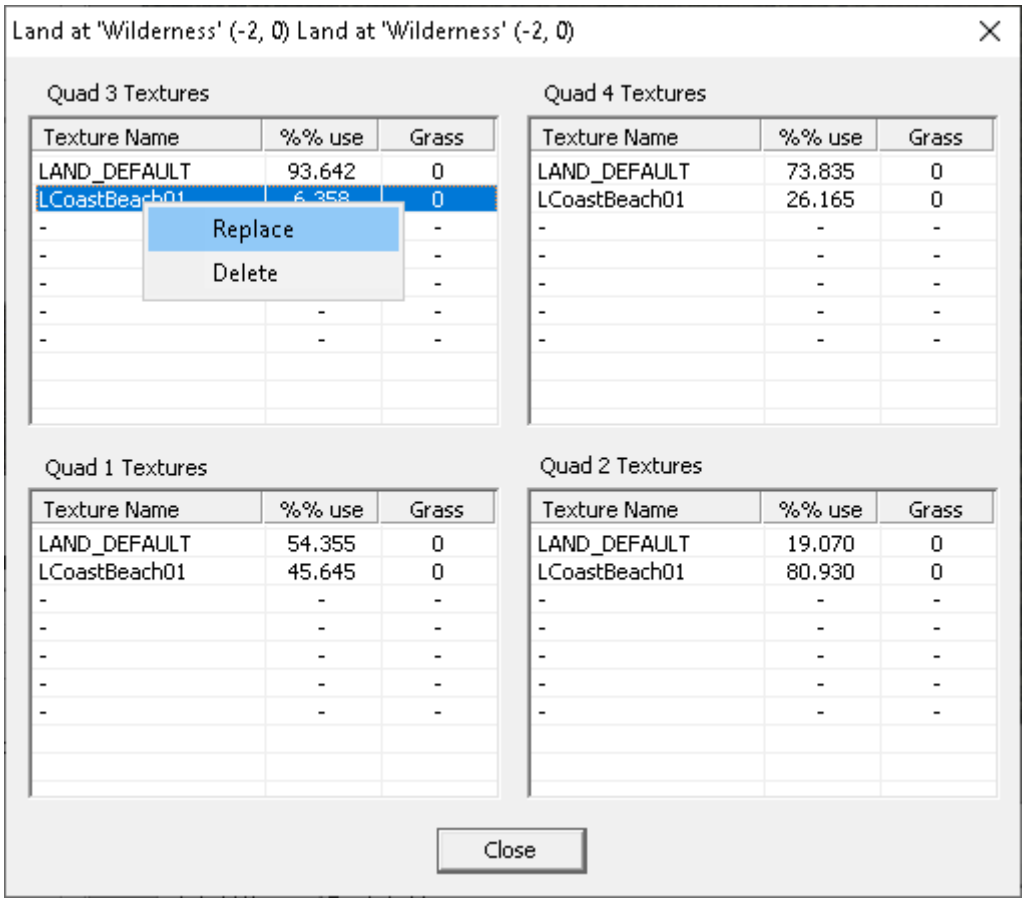


Figure 87 - Replacing a texture in a quad.

In the Select Form pop-up, choose the texture you want instead and click OK.

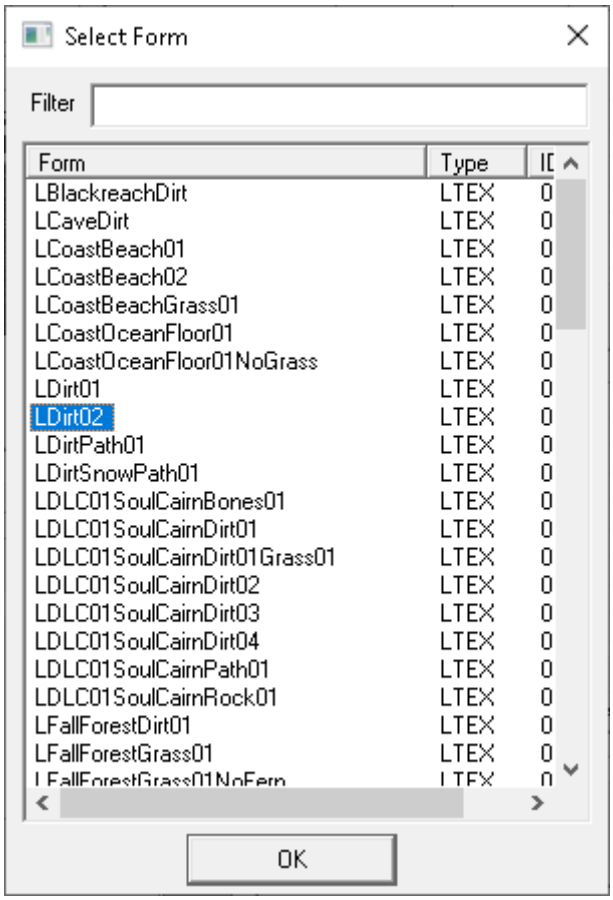


Figure 88 - Choosing the texture we want to replace LCoastBeach01 with.

When replacing textures, just be aware that you may create a texture seam, like the one in the screenshot below along the edge of the quad if you replaced the texture in one quad but not the other.

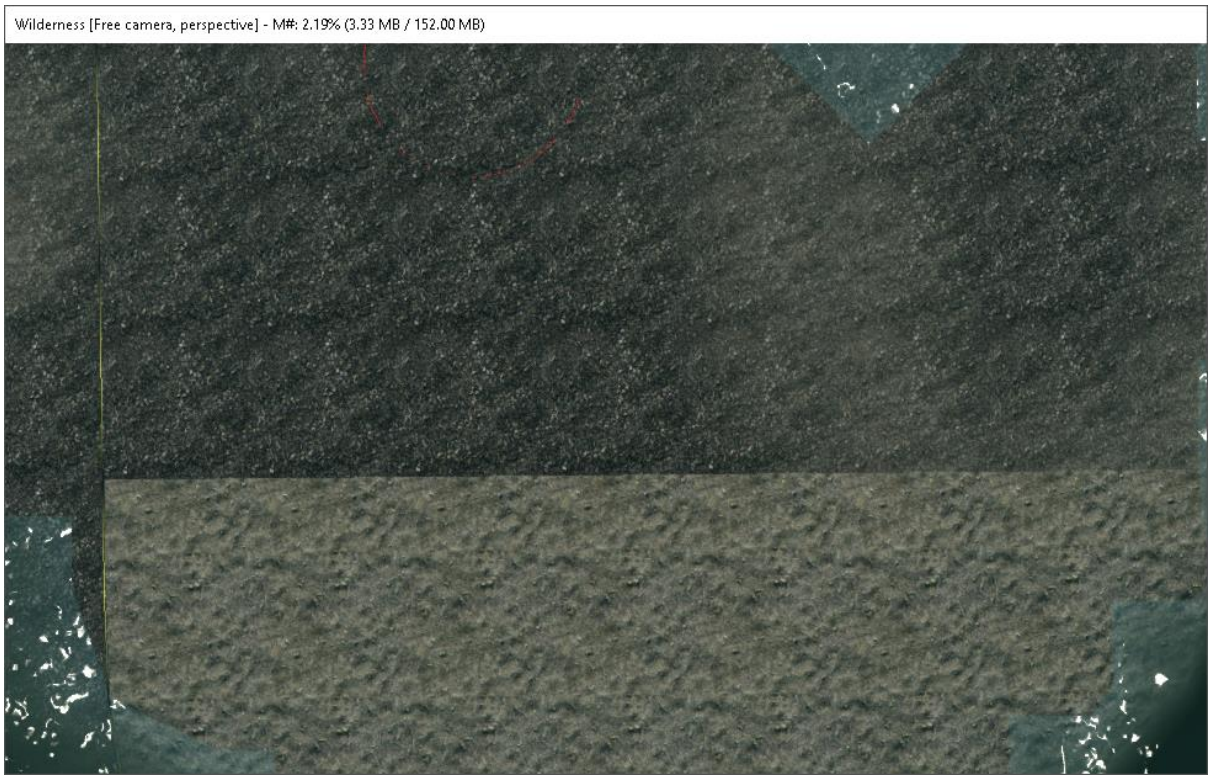


Figure 89 - A texture seam caused by replacing or deleting a texture.

CREATING TILEABLE LANDSCAPE TEXTURES

In this section I'll be showing you how to create your own tileable terrain textures that you can paint onto your landscape. I'll be using Photoshop CS5 in my example.

Note: For Photoshop you will need the NVIDIA Texture Tools Plugin which is available on the NVIDIA website.

Firstly, we'll need a source image. For this example, I just went to Google Images and searched for 'dirt', limiting my search results to images under a Creative Commons license.

The image below is free for commercial use with no attribution requirements.

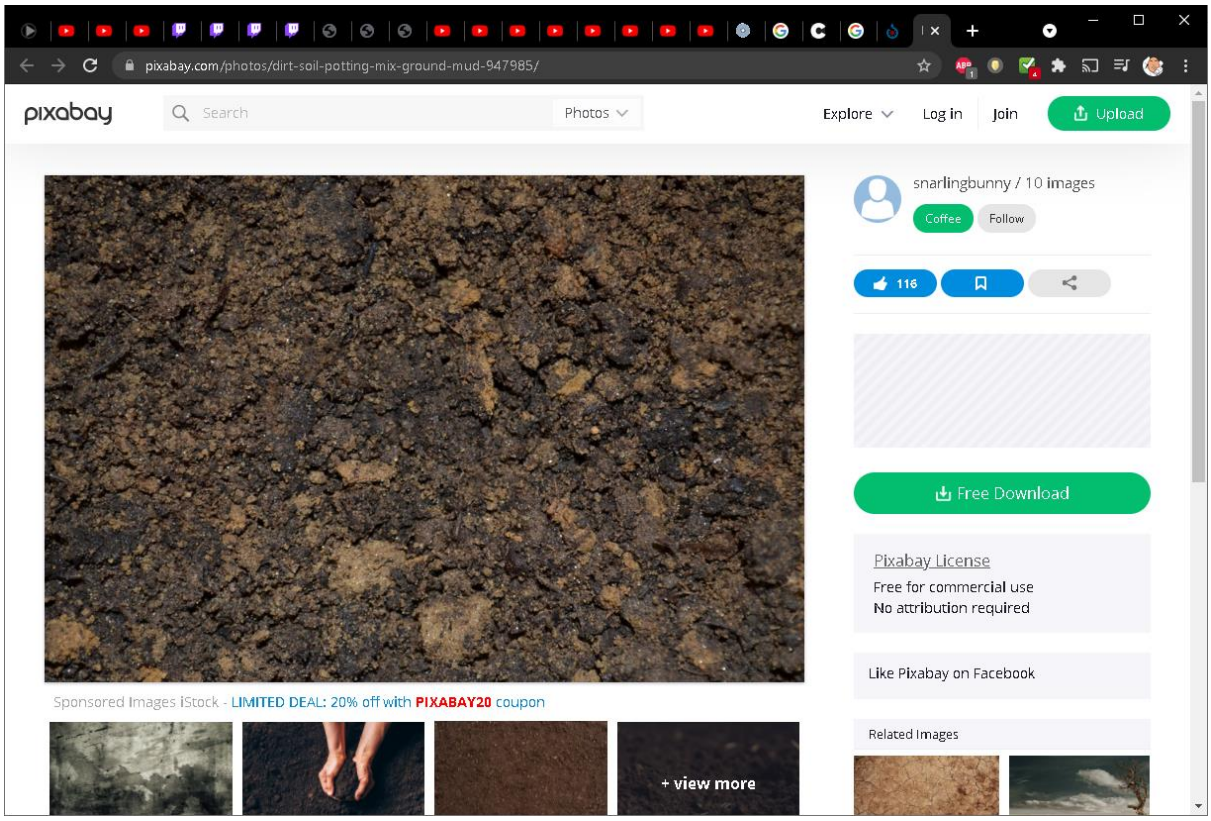


Figure 90 - Source dirt texture.

In Photoshop, go to File > New.

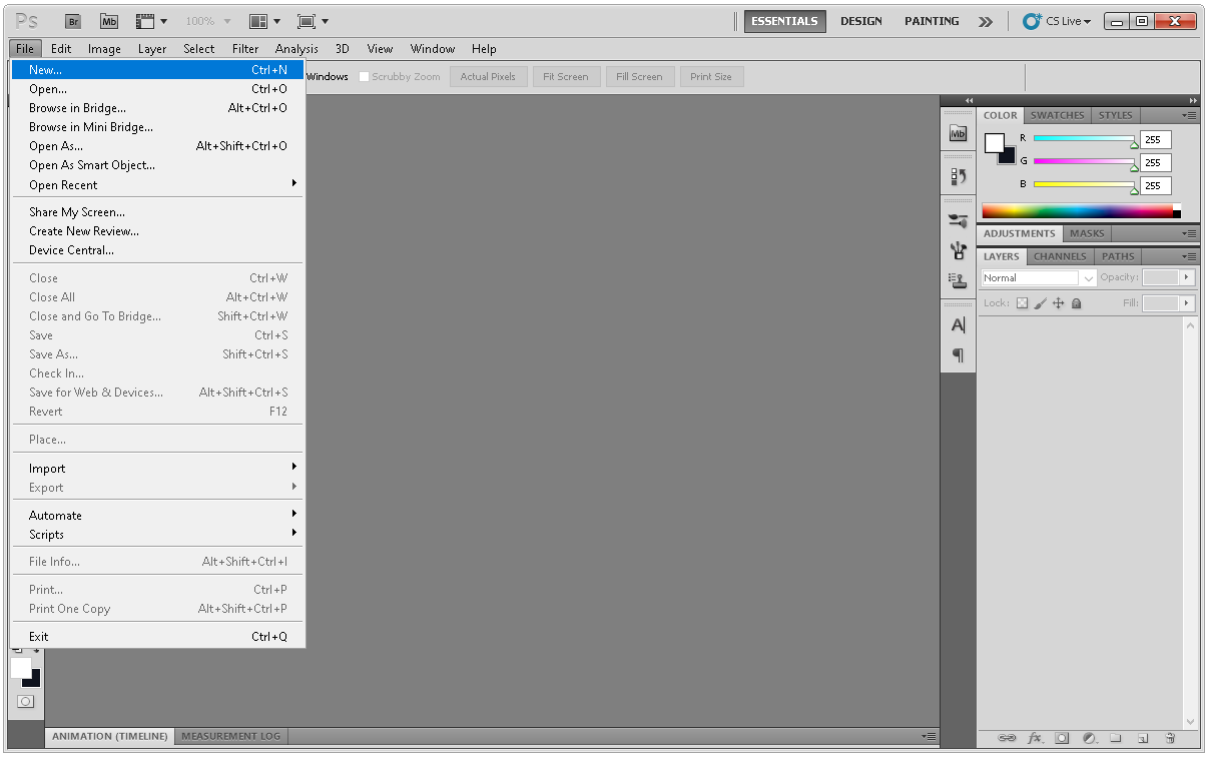


Figure 91 - Creating a new image file in Photoshop.

For the sake of this example, I'm going to create a new 512x512 texture which is the same size as the default landscape textures in Skyrim. If you want to make it bigger that's up to you, just make sure to double the size, e.g.: 1024x1024, 2048x2048, and so on.

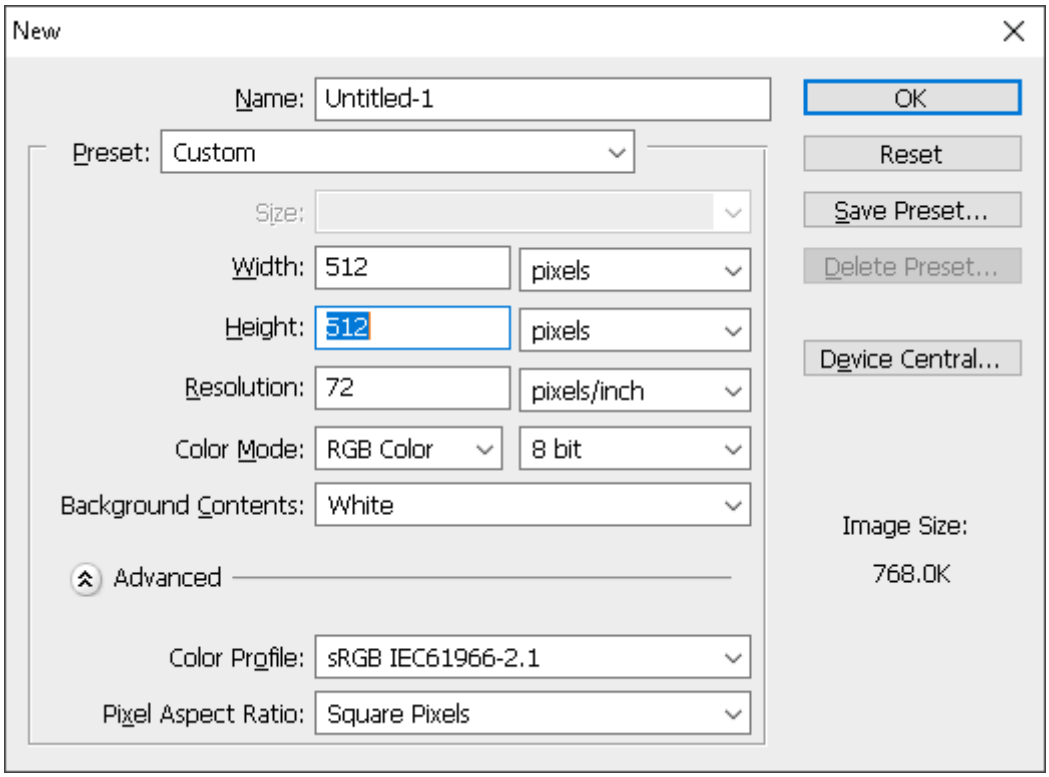


Figure 92 - Size of new image.

Click OK to create a new blank document.

Drag and drop your texture into Photoshop and expand it to fill the height and width of the canvas.

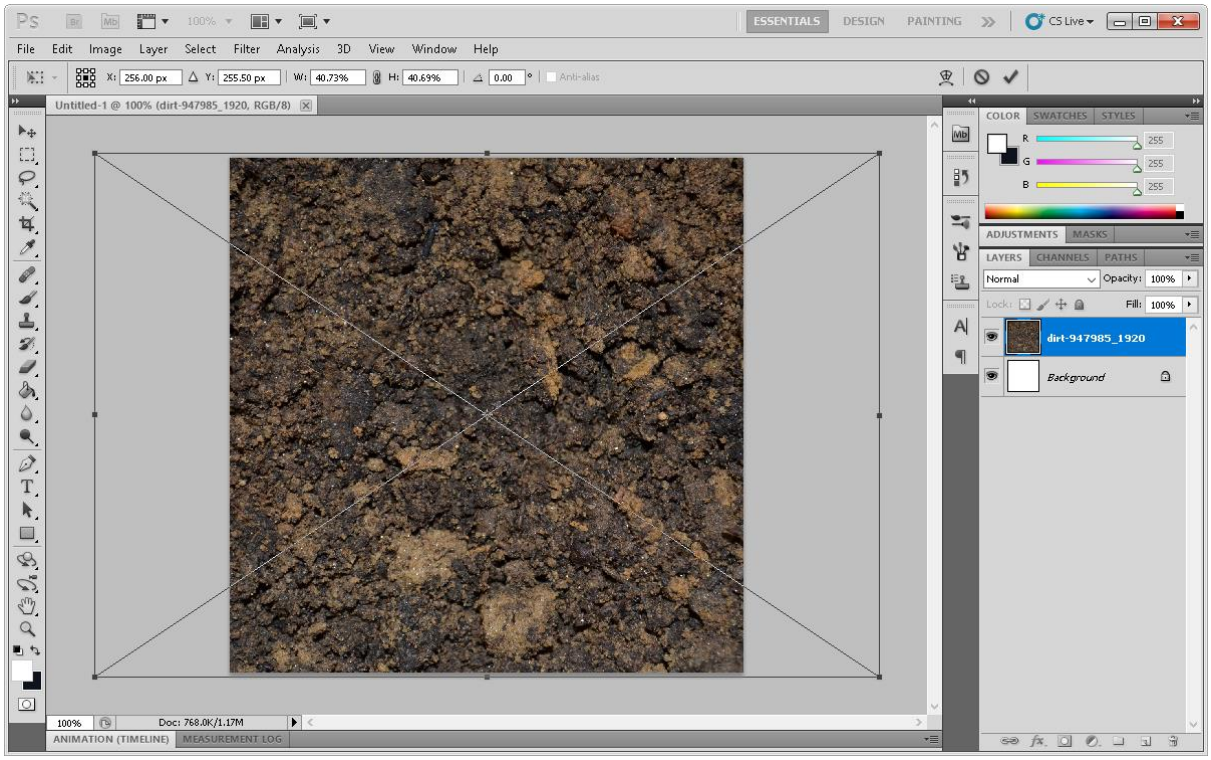


Figure 93 - Importing the source image.

I made some adjustments to the image, mostly out of personal taste. I desaturated the texture, reduced its brightness, shifted the hue slightly to make it less red, then ran a sharpen filter. The idea was basically to try and make it match the existing landscape textures in the base game which are all fairly dark and desaturated.

Don't worry, our texture will look a lot lighter in-game in day time lighting.

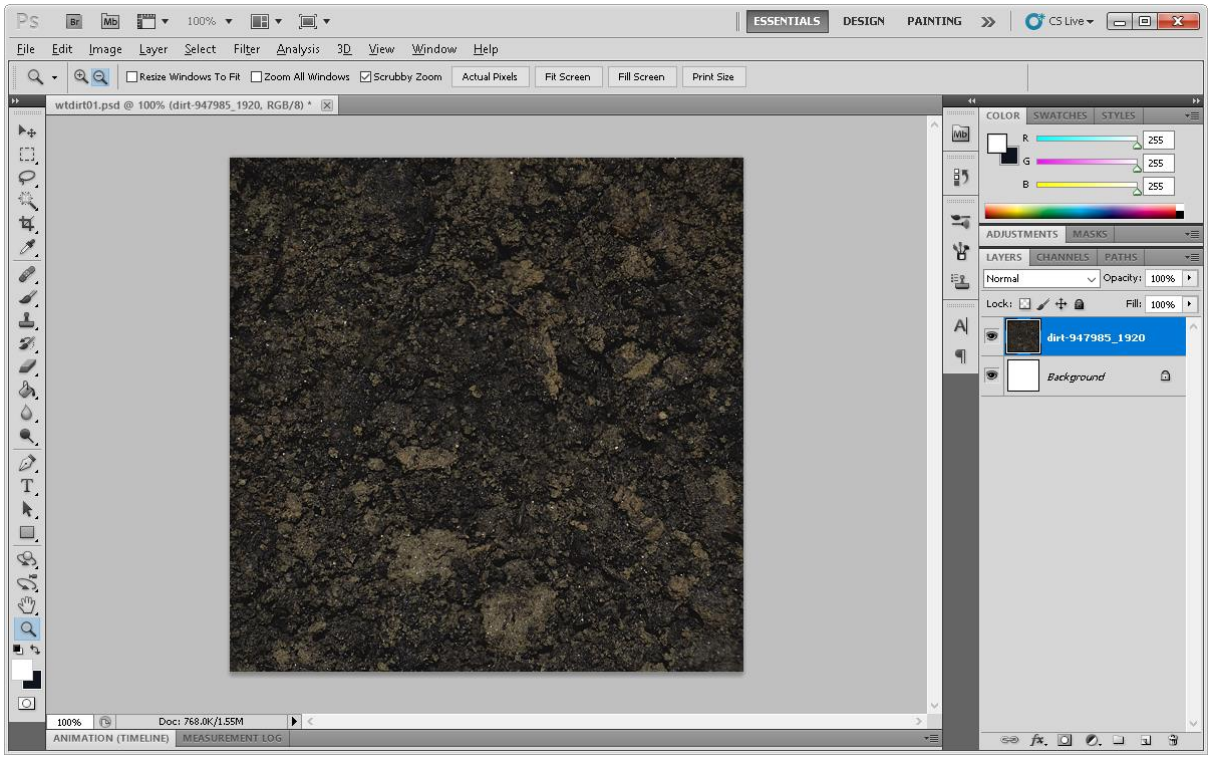


Figure 94 - Dirt image after some adjustments.

Now to make the image tileable.

Go to Filter > Other > Offset and change the Horizontal offset until the vertical texture seam is roughly in the middle of the image.

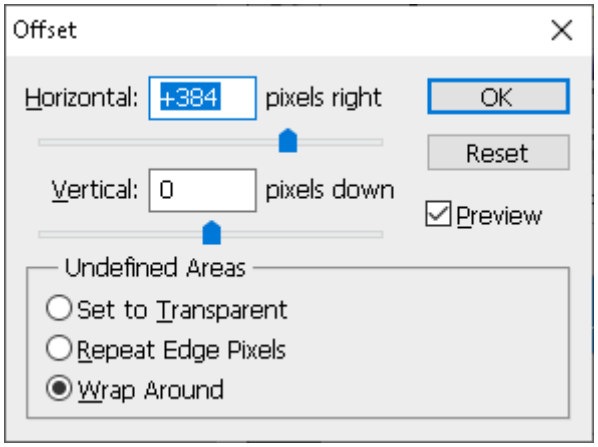


Figure 95 - Changing the horizontal texture offset.

Ensure Undefined Areas is set to Wrap Around then click OK.

In the screenshot below you can see the vertical texture seam running down the centre of the image.

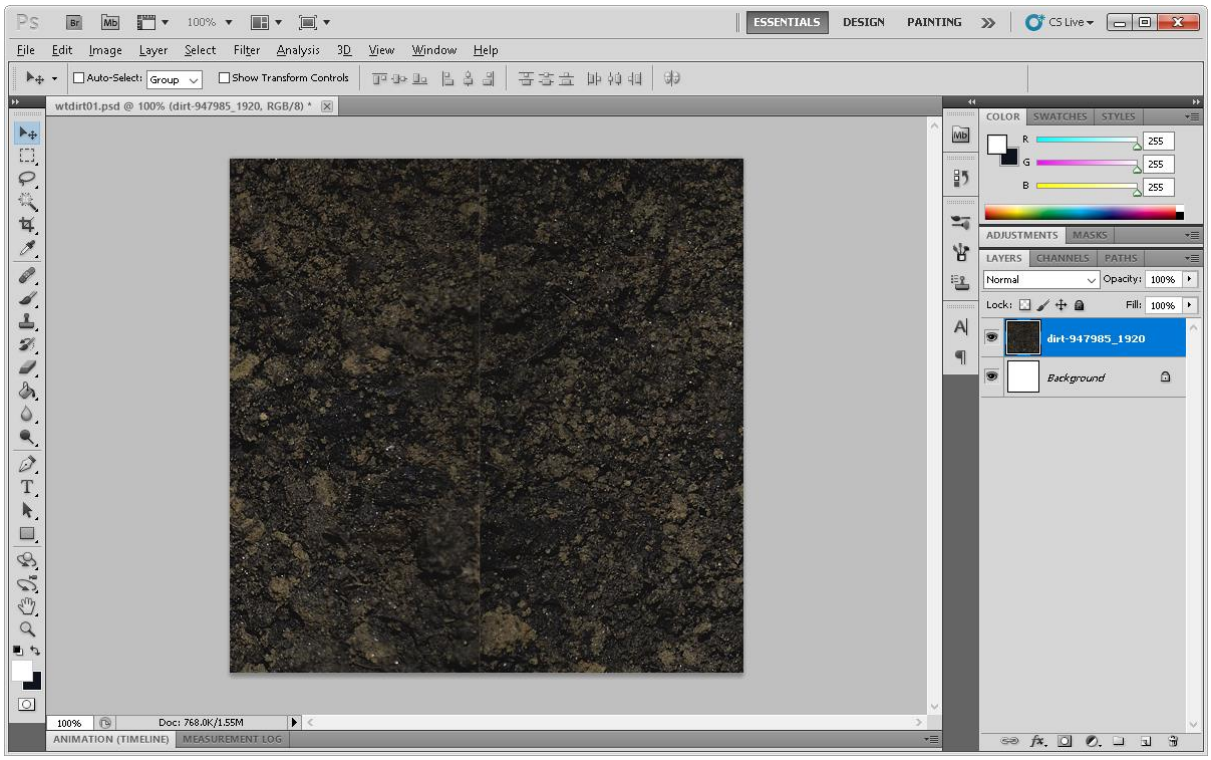


Figure 96 - Vertical texture seam.

Use the Healing Brush to remove the seam. If you hold down ALT your cursor will change into a crosshair. ALT left-click on a part of the image away from the seam to sample that part of the image, then left-click over the seam to begin replacing it with the part of the image you sampled.

In the following screenshot, the vertical seam has been removed.

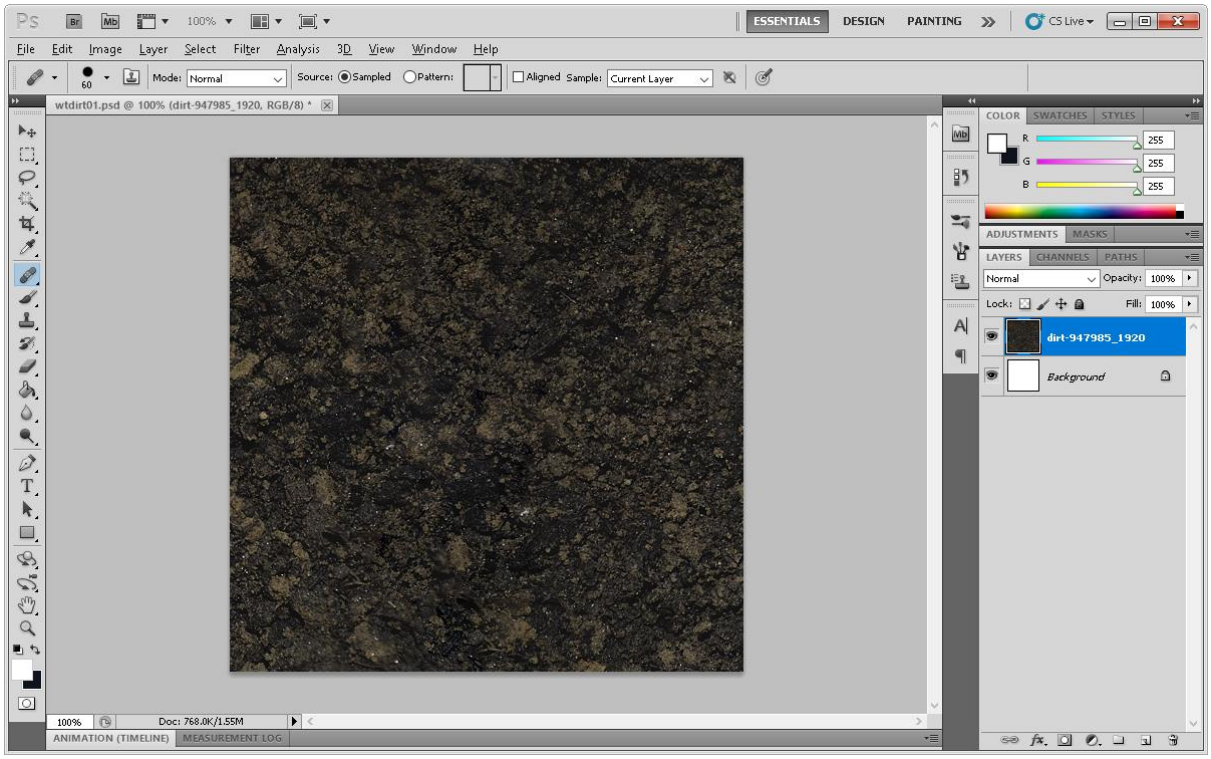


Figure 97 - Vertical seam removed.

Go back to Filter > Other > Offset.

This time, set a Vertical offset. Make sure the Horizontal offset is set back to 0 first.

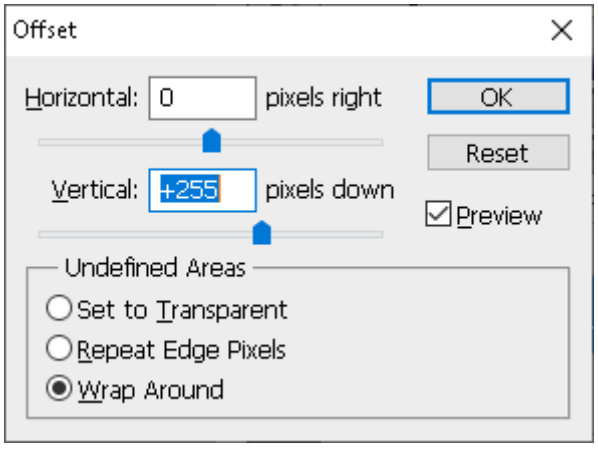


Figure 98 - Setting a vertical offset.

Again, make sure Undefined Areas is set to Wrap Around, then click OK.

You should now see a horizontal seam running across the image.

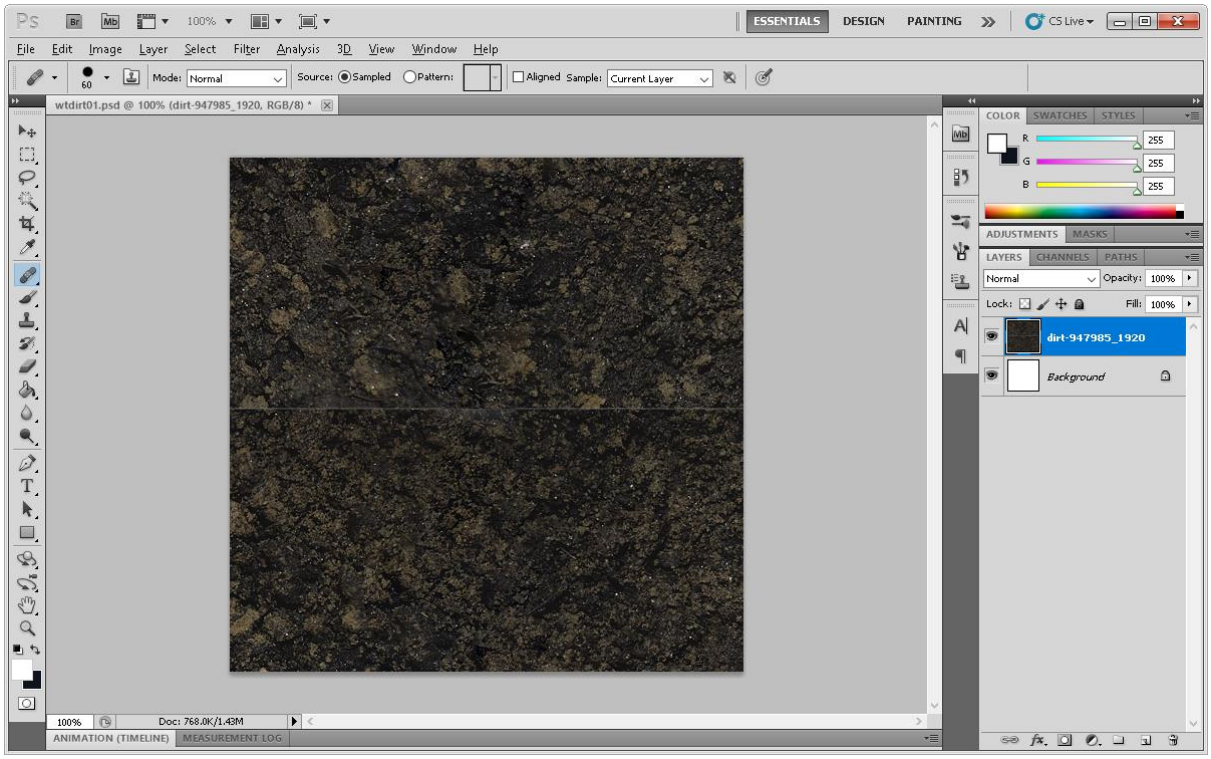


Figure 99 - Horizontal seam.

Use the Healing Brush tool to remove it in the same way you removed the vertical seam earlier.

Again, go back to Filter > Other > Offset.

This time, change both the Vertical and Horizontal offsets and just make sure no more seams are visible. If there are, use the Healing Brush tool to remove them.

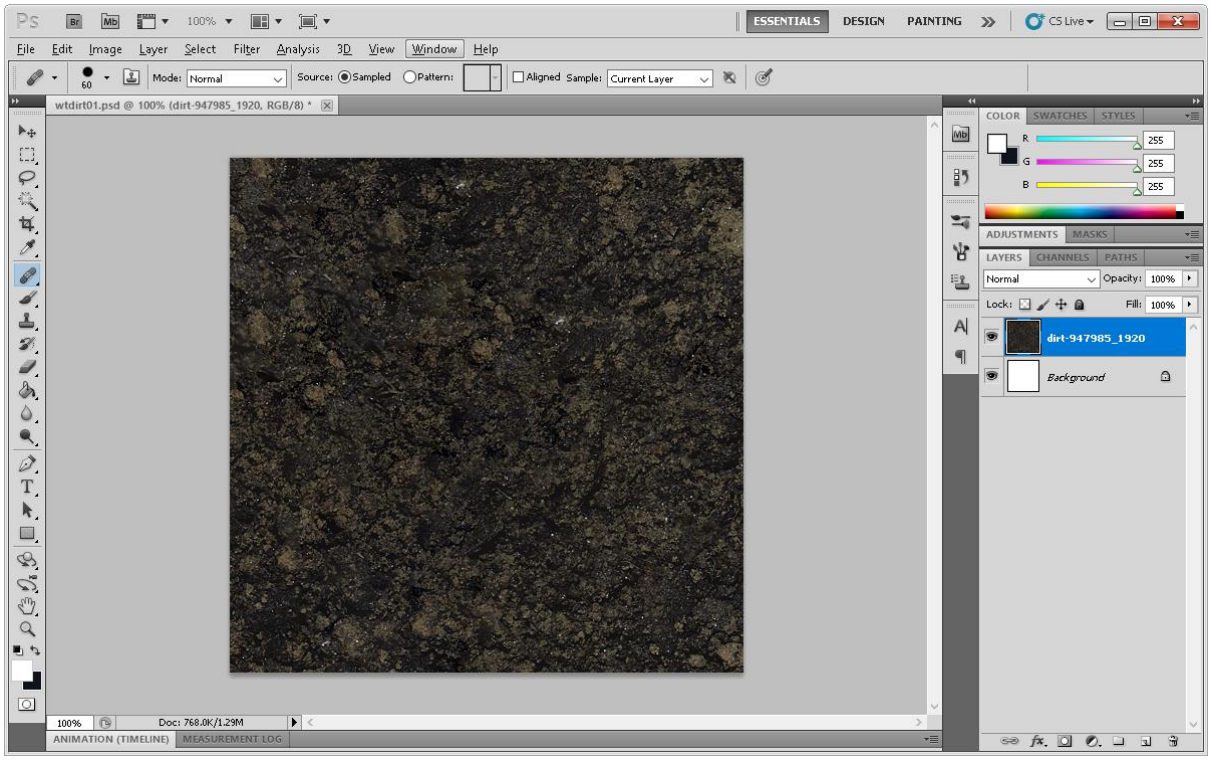


Figure 100 - Making sure there are no more seams.

Alright, our diffuse map is done. Time to export the image as a .dds file.

Go to File > Save As.

Set the Format to 'D3D/DDS' and choose a location to save the .dds file to.

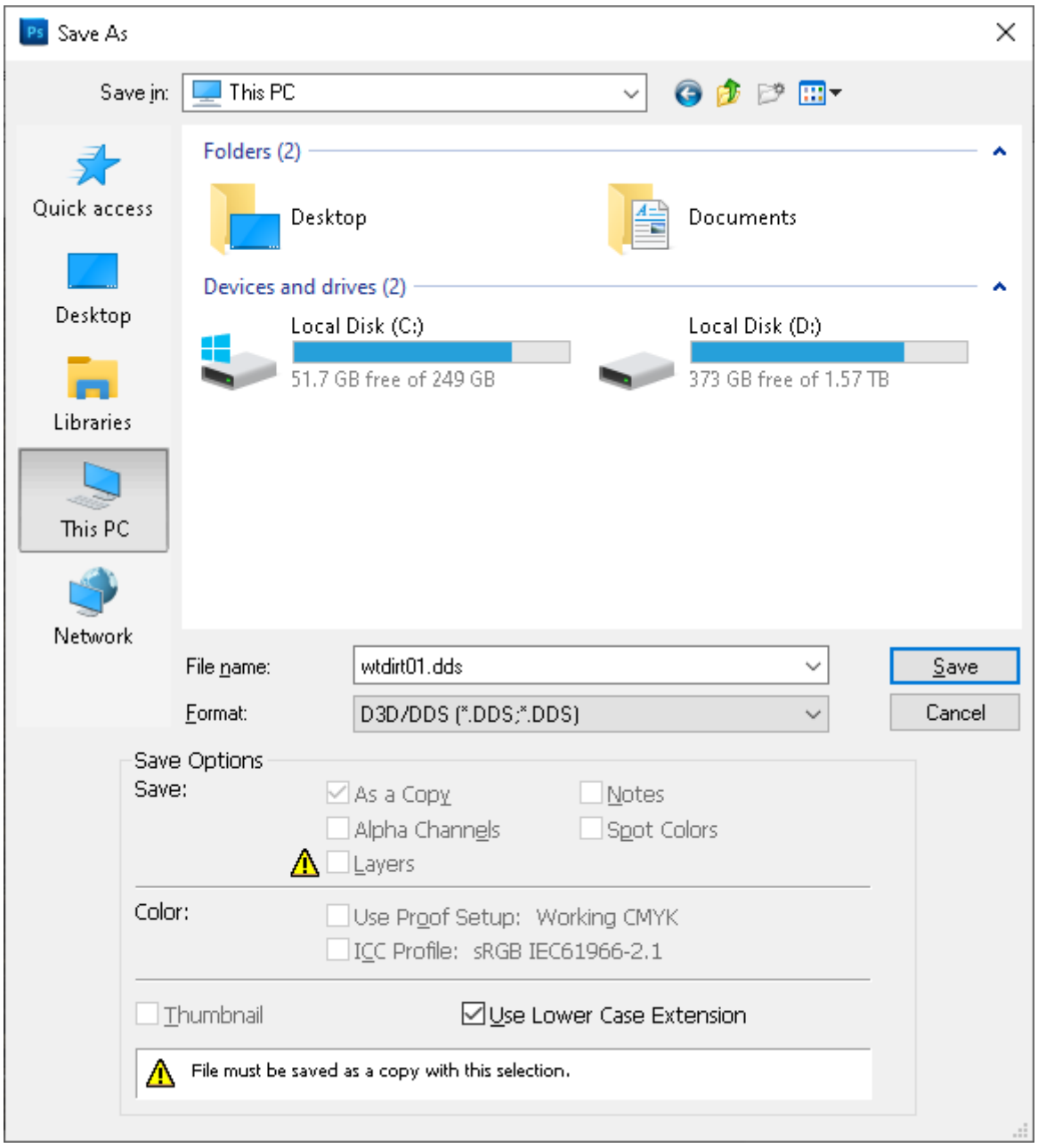


Figure 101 - Saving our file as a .dds file.

Set the .dds format to DXT5. Make sure MIP Map Generation is set to 'Generate MIP Maps' then click Save.

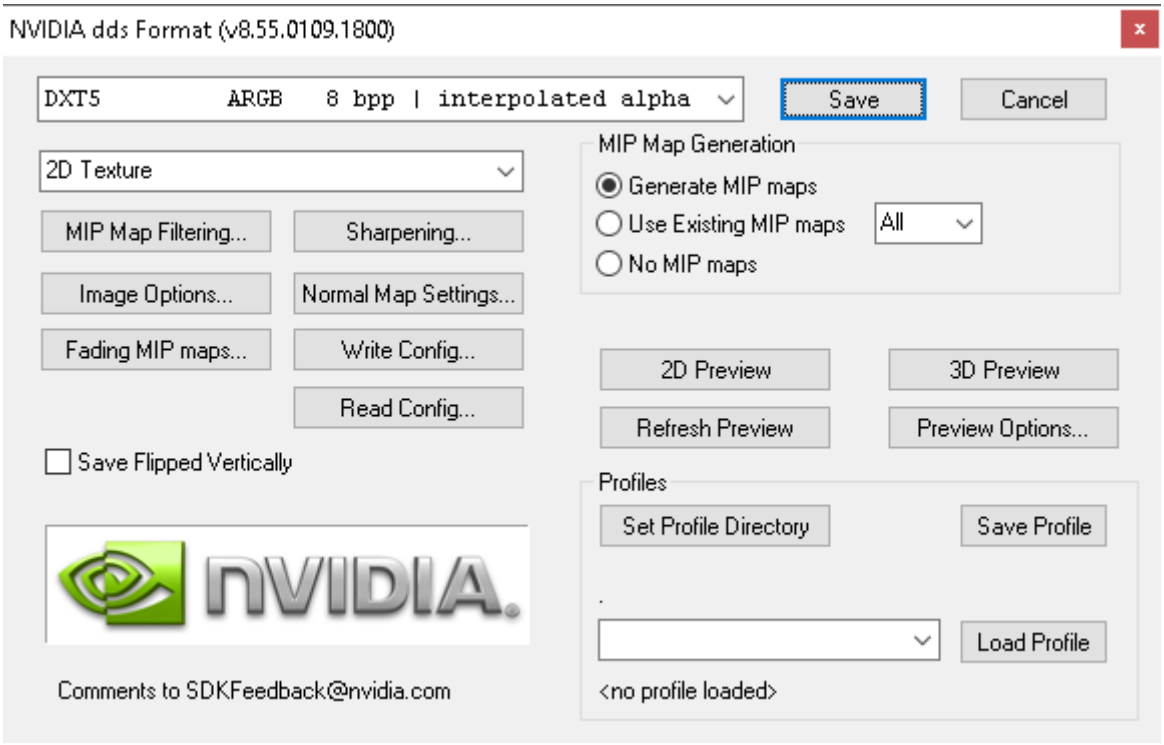


Figure 102 - NVIDIA Texture Tools Plugin.

Next, we need to create a normal map for our texture. For this, I'm going to use the njob utility.

Launch njob, go to File > Open and select your .dds texture file.



Figure 103 - Our texture loaded in njob.

The first thing we need to do is convert our image into a height map. Go to Filter and select Diffusemap > Heightmap or press '0'.

What you set the sliders to is up to your personal preference, but to try and bring out more granular detail I'm going to reduce the Scale and Coarse Detail sliders.

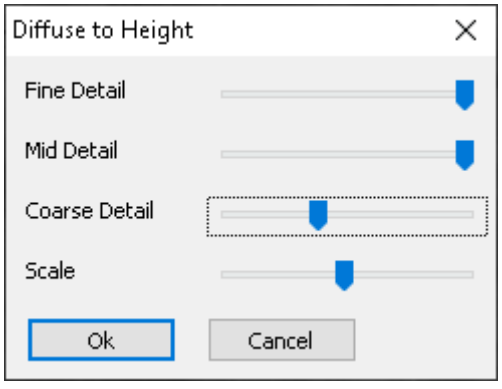


Figure 104 - Diffuse to Height.

We should now have a black and white image similar to the following screenshot.

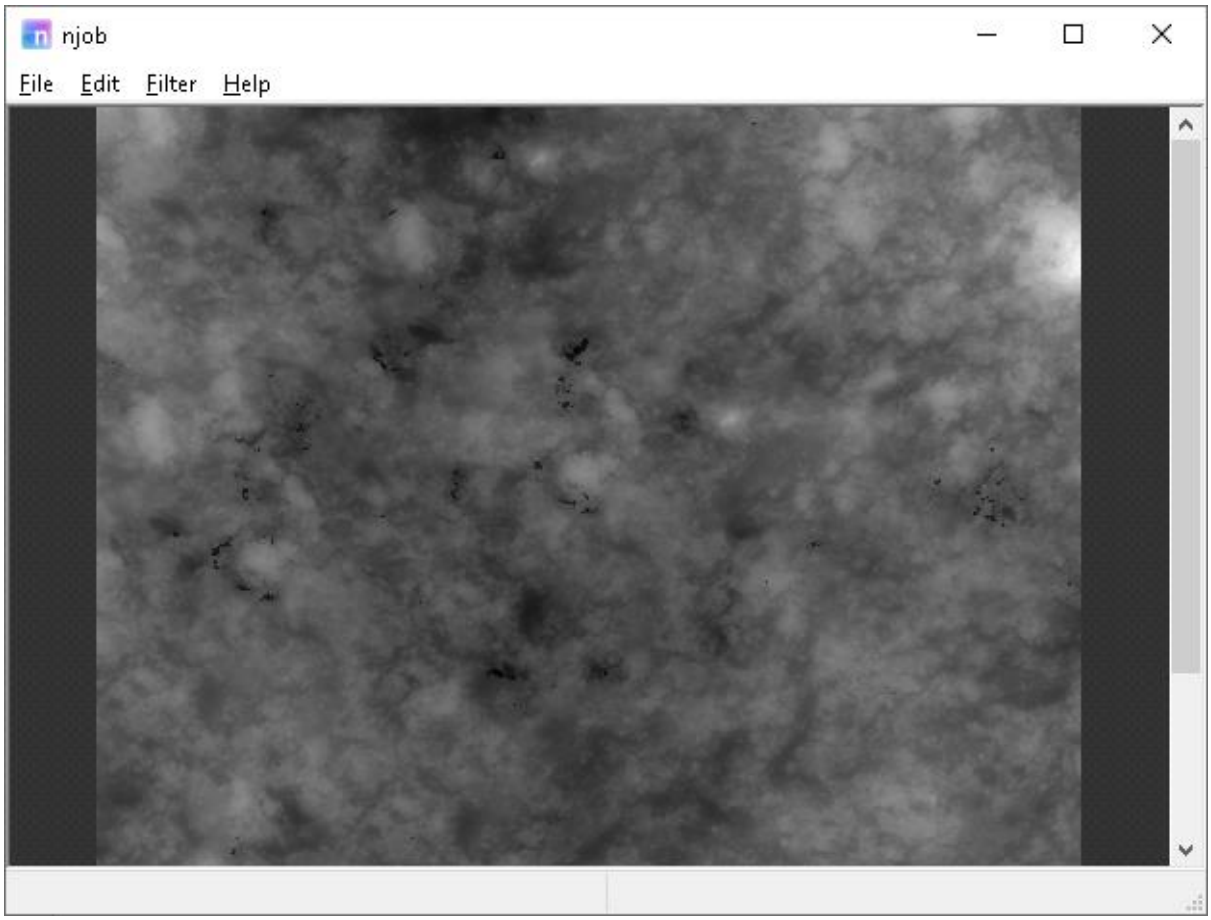


Figure 105 - Our texture converted into a height map.

Now we need to convert this to a normal map.

Go to Filter and choose Heightmap > Normalmap.

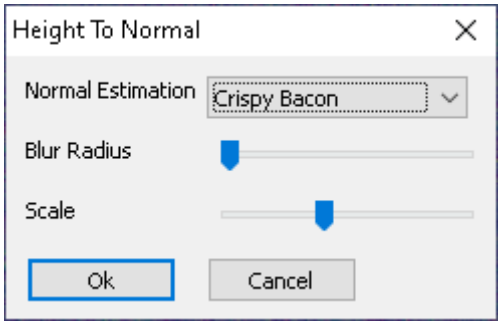


Figure 106 - Height to Normal.

I would recommend changing the Normal Estimation to 'Crispy Bacon' and increase the Scale slider a bit for a closer approximation to the normal maps used by default Skyrim landscape textures. If you take a look at the normal maps of existing textures, they all look pretty sharp.

Click OK.

Now we have our normal map.

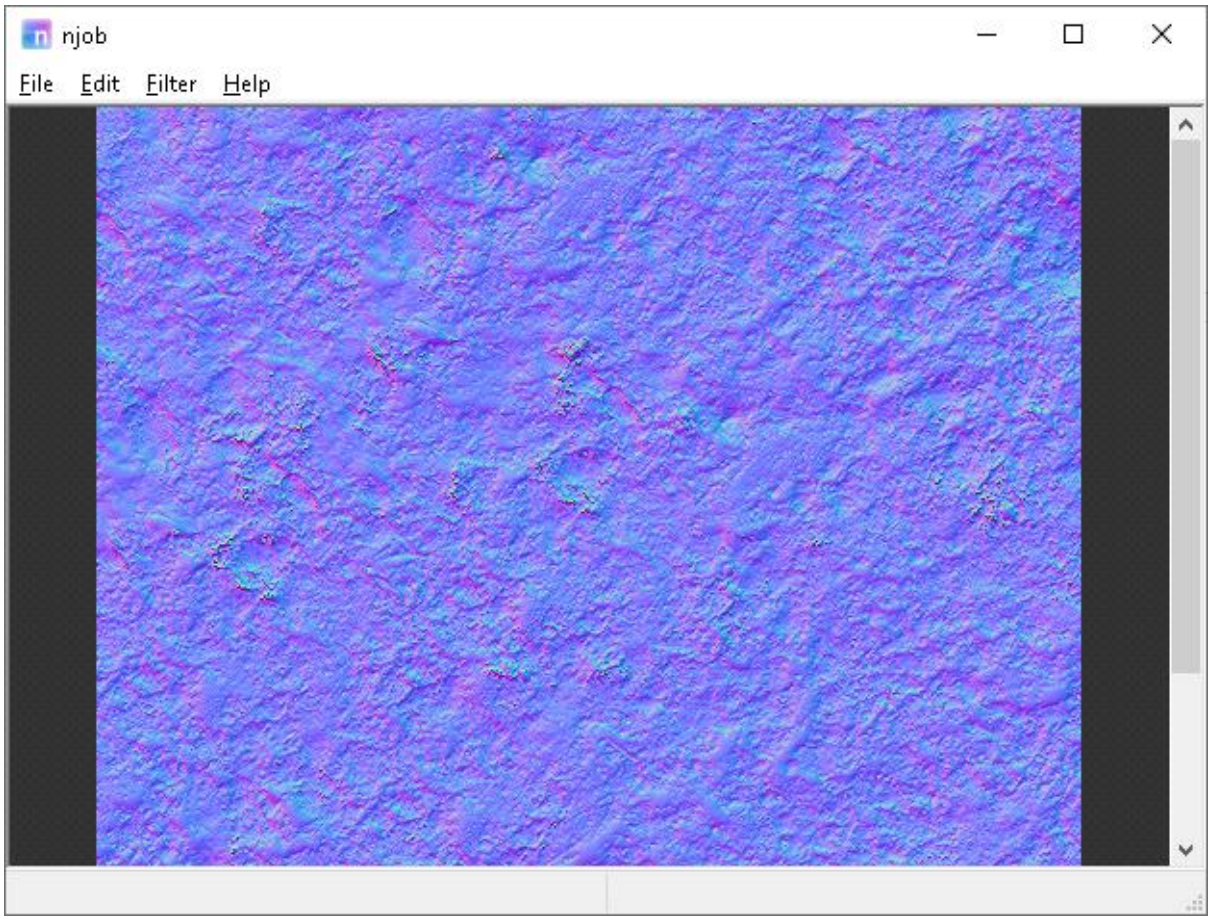


Figure 107 - Normal map for the new landscape texture.

To export it, go to File > Save As.

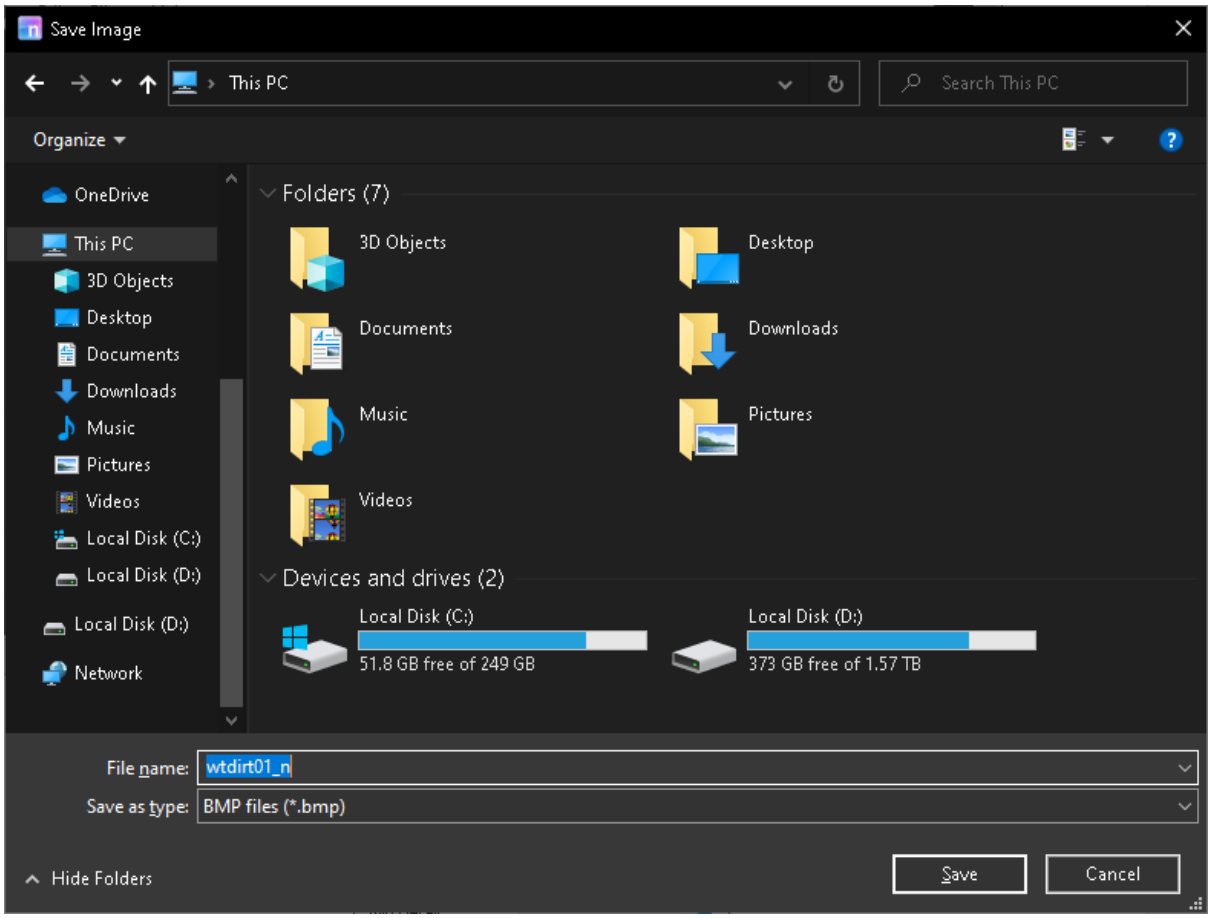


Figure 108 - Saving our normal map.

Choose a location to save to. njob can't save files in the .dds format so we're going to have to save our image as a bitmap for now. Normal map textures in Skyrim have '_n' appended to the end of the file name so don't forget to add that as well.

Click Save.

Time to convert that bitmap to .dds format.

Open the normal map in Photoshop.

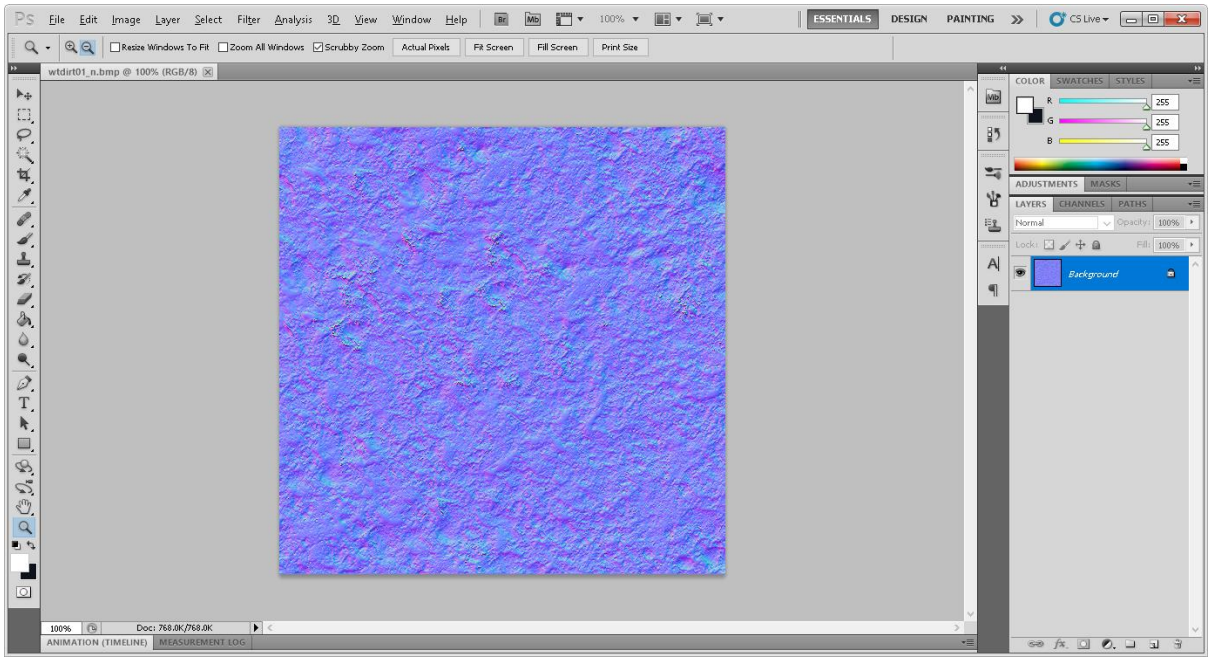


Figure 109 - Our normal map in Photoshop.

Go to File > Save As.

Choose a location to save the .dds file to.

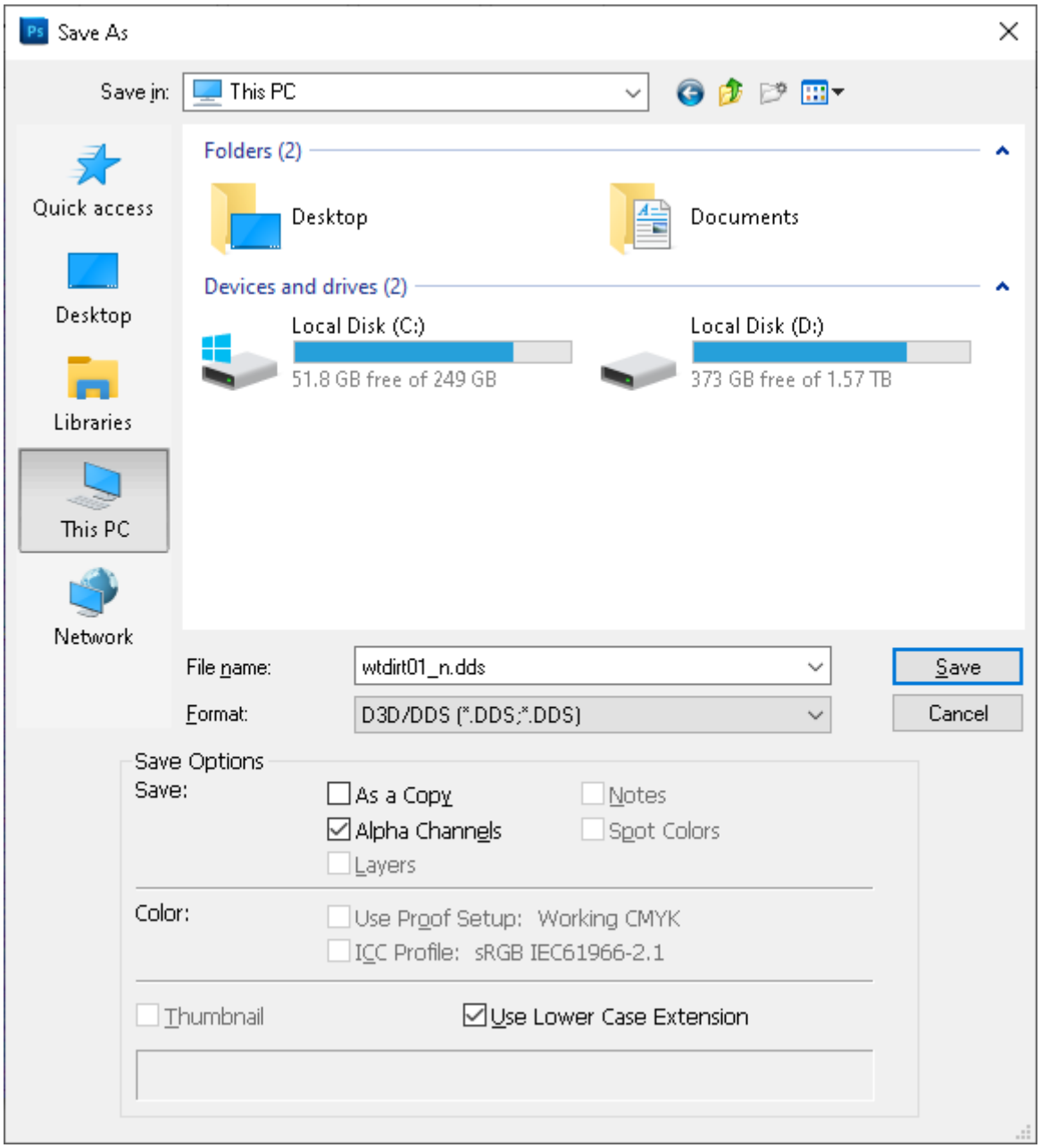


Figure 110 - Saving our normal map as a .dds file.

Set the .dds format to DXT5. Make sure MIP Map Generation is set to 'Generate MIP Maps' then click Save.

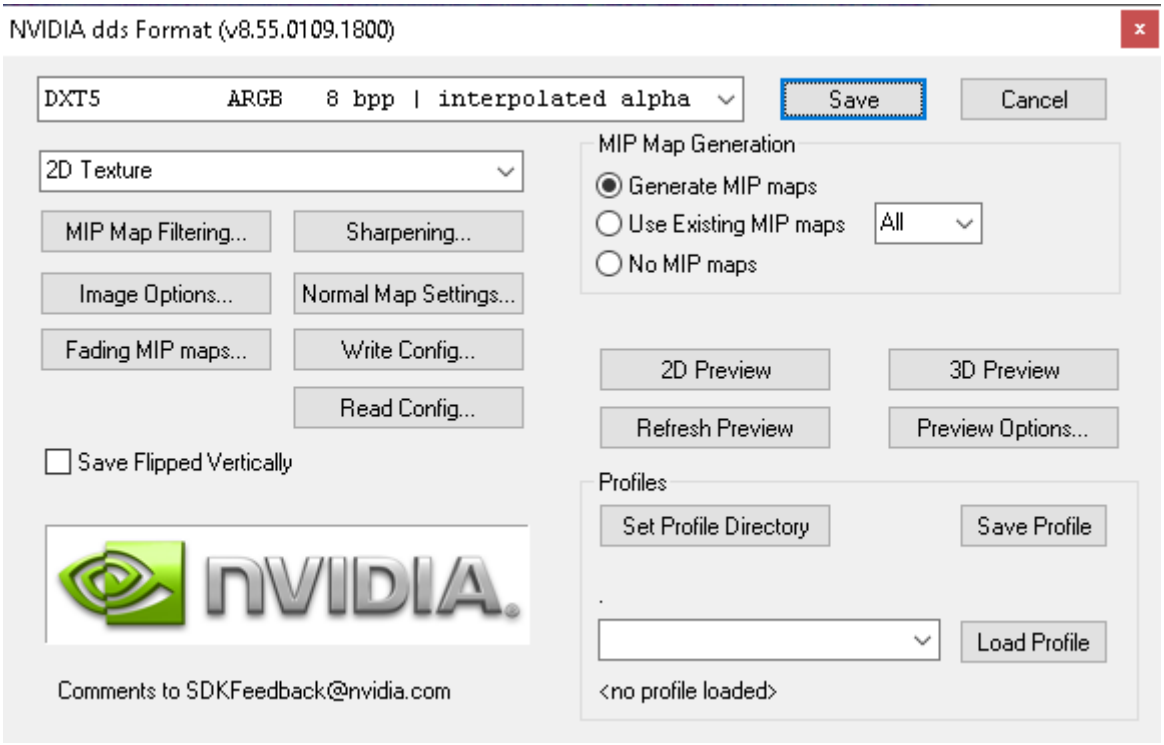


Figure 111 - NVIDIA Texture Tools Plugin.

Copy your diffuse texture and its normal map to the 'Skyrim\Data\Textures\landscape' folder. If the Textures or Landscape folders don't exist under Data, create them first.

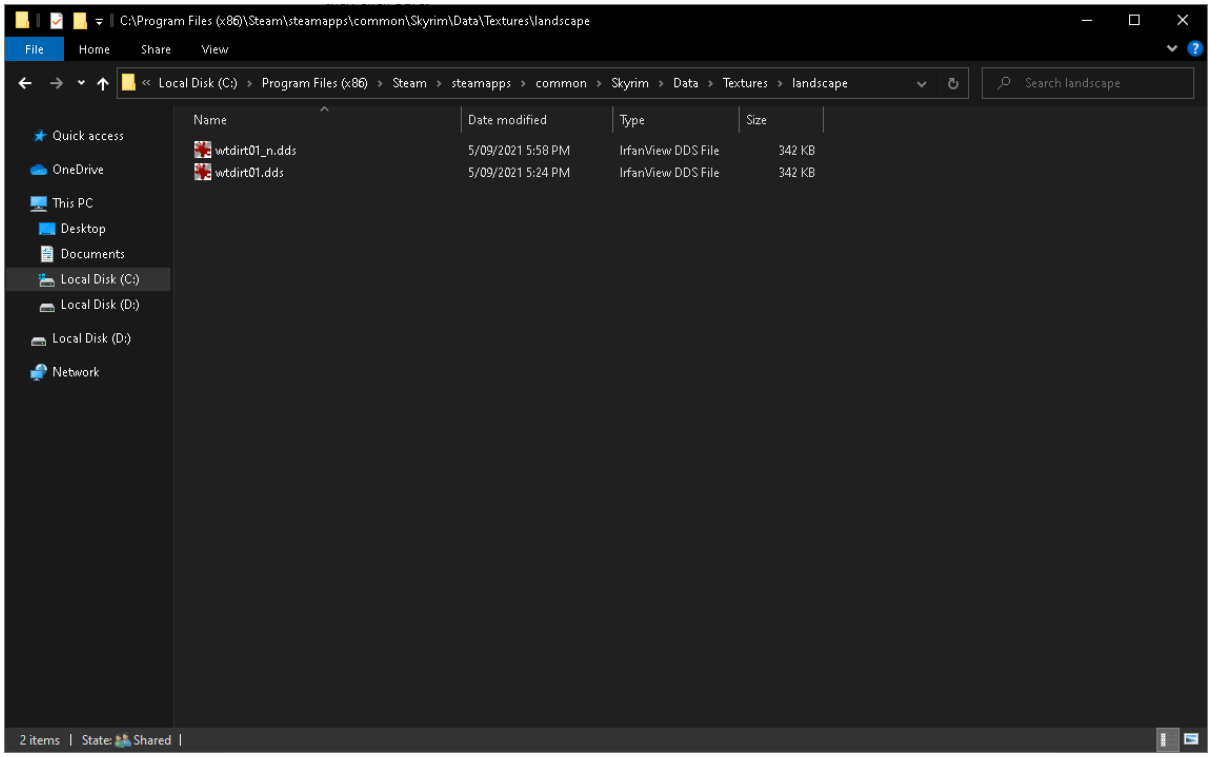


Figure 112 - Our files in copied to the correct directory.

Alright, we're almost there.

Back in the Creation Kit, in the Object Window go to Miscellaneous > TextureSet.

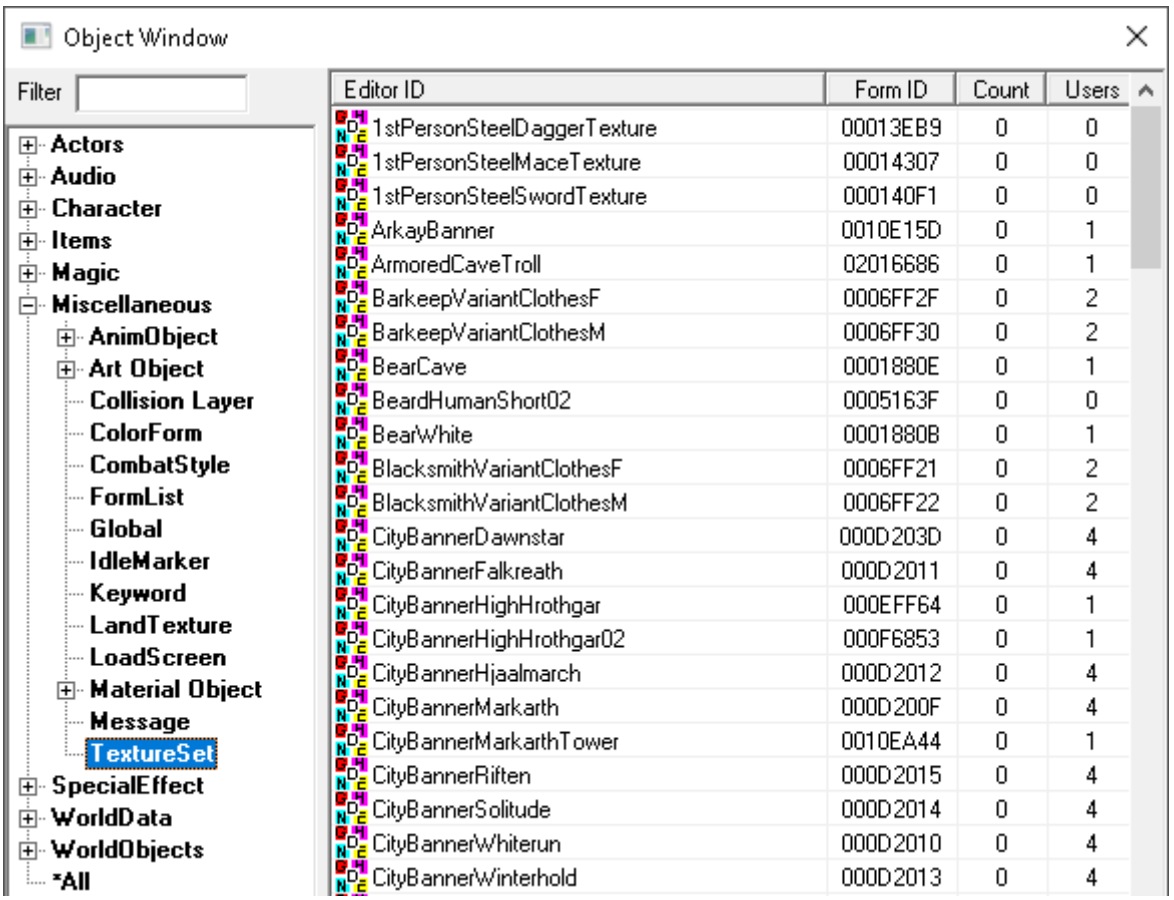


Figure 113 - Texture sets.

Right-click on one of the existing texture sets and select New.

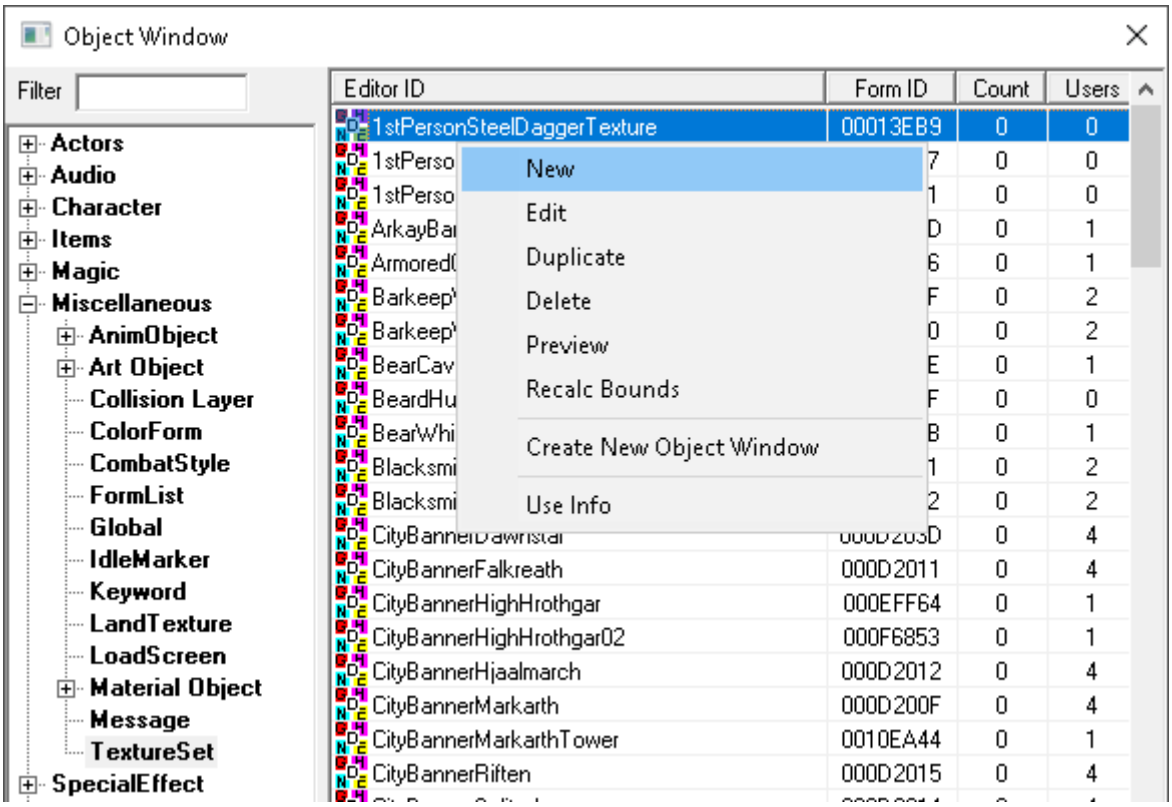


Figure 114 - Creating a new texture set.

Enter in a name for your texture set. In this example, I just called it LandscapeDirt01 and added the 'WT' prefix.

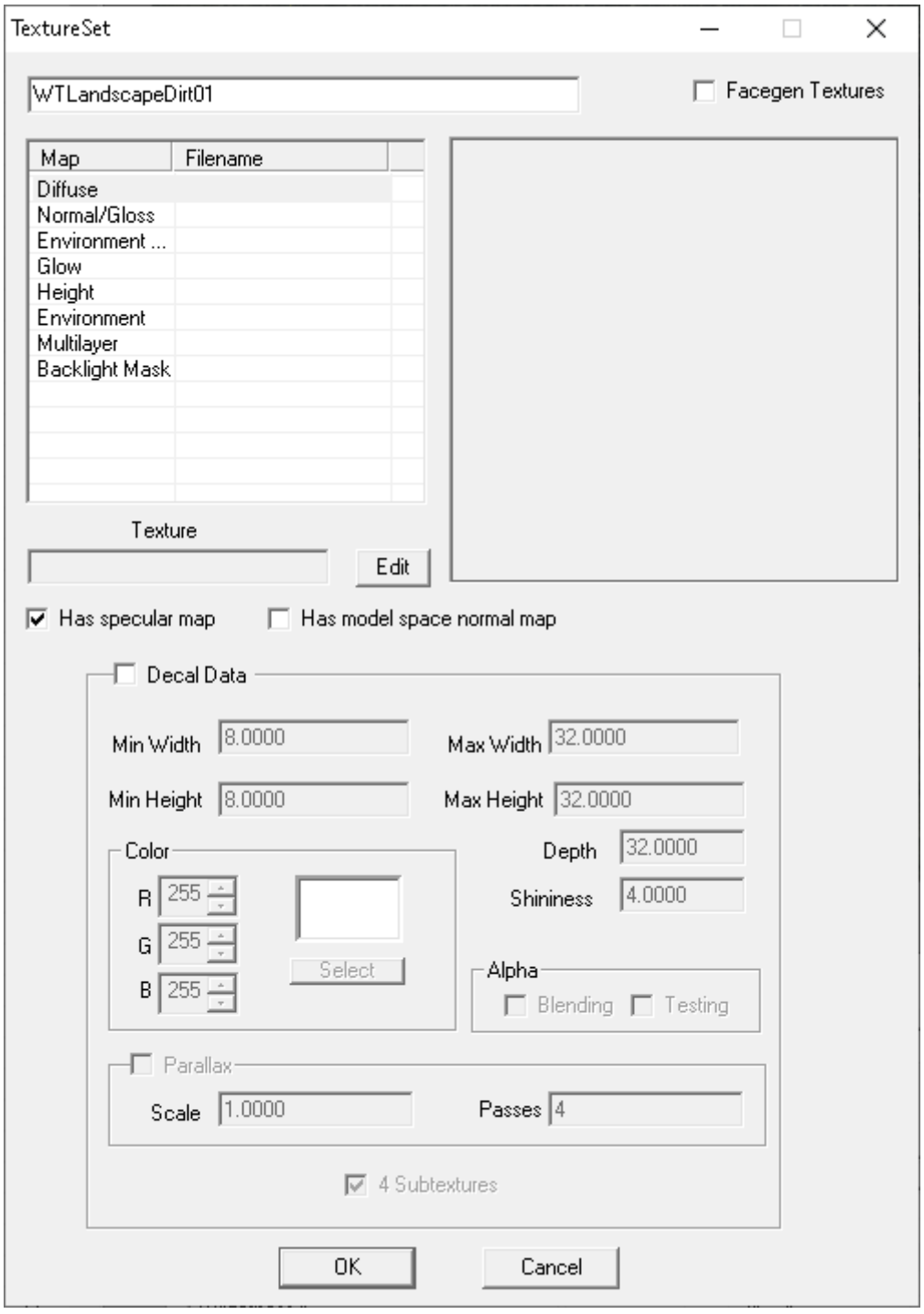


Figure 115 - Naming the new texture set.

Click on Diffuse then click on the Edit button to the right of Texture.

Select your diffuse map located under 'Skyrim\Data\Textures\landscape' and click Open.

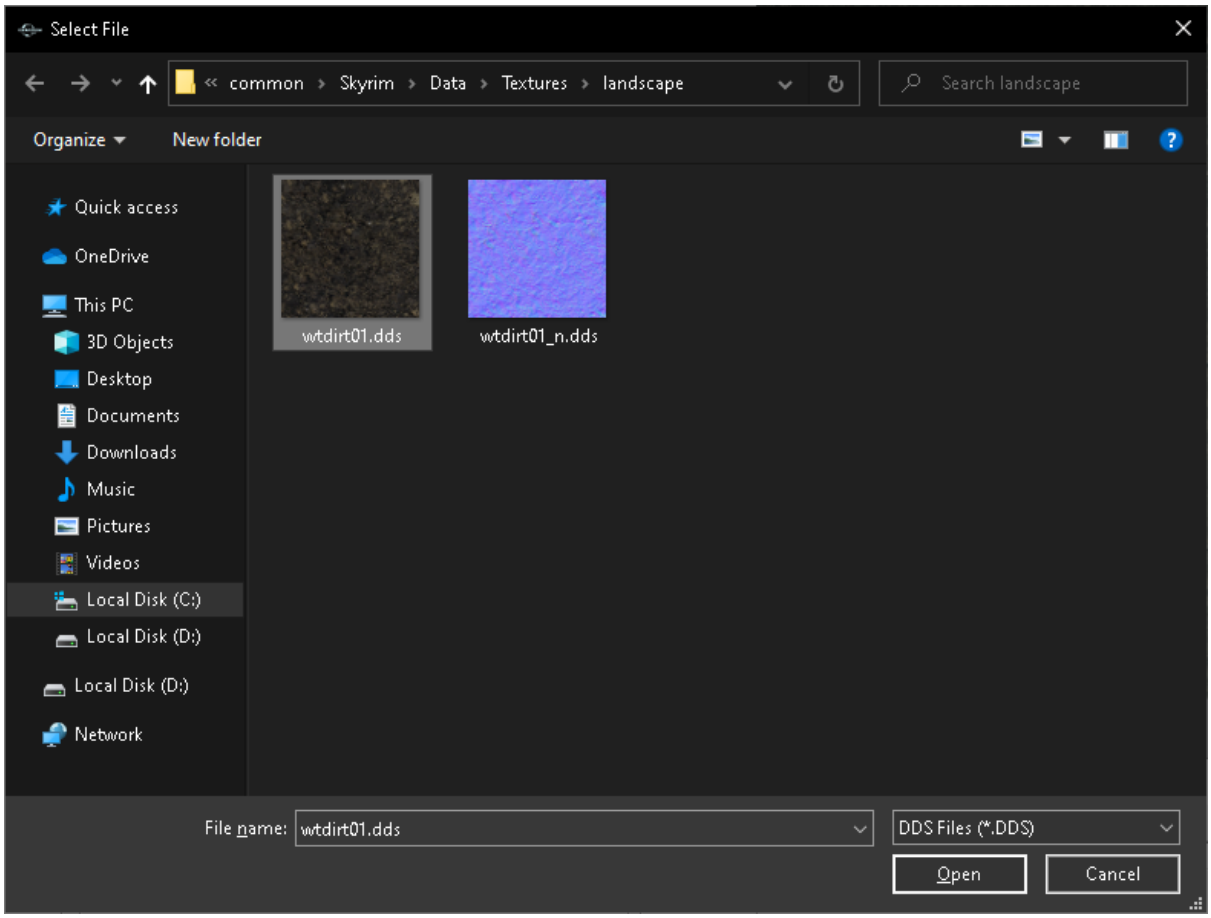


Figure 116 - Diffuse map selected.

Repeat those steps to add your normal map to Normal/Gloss.

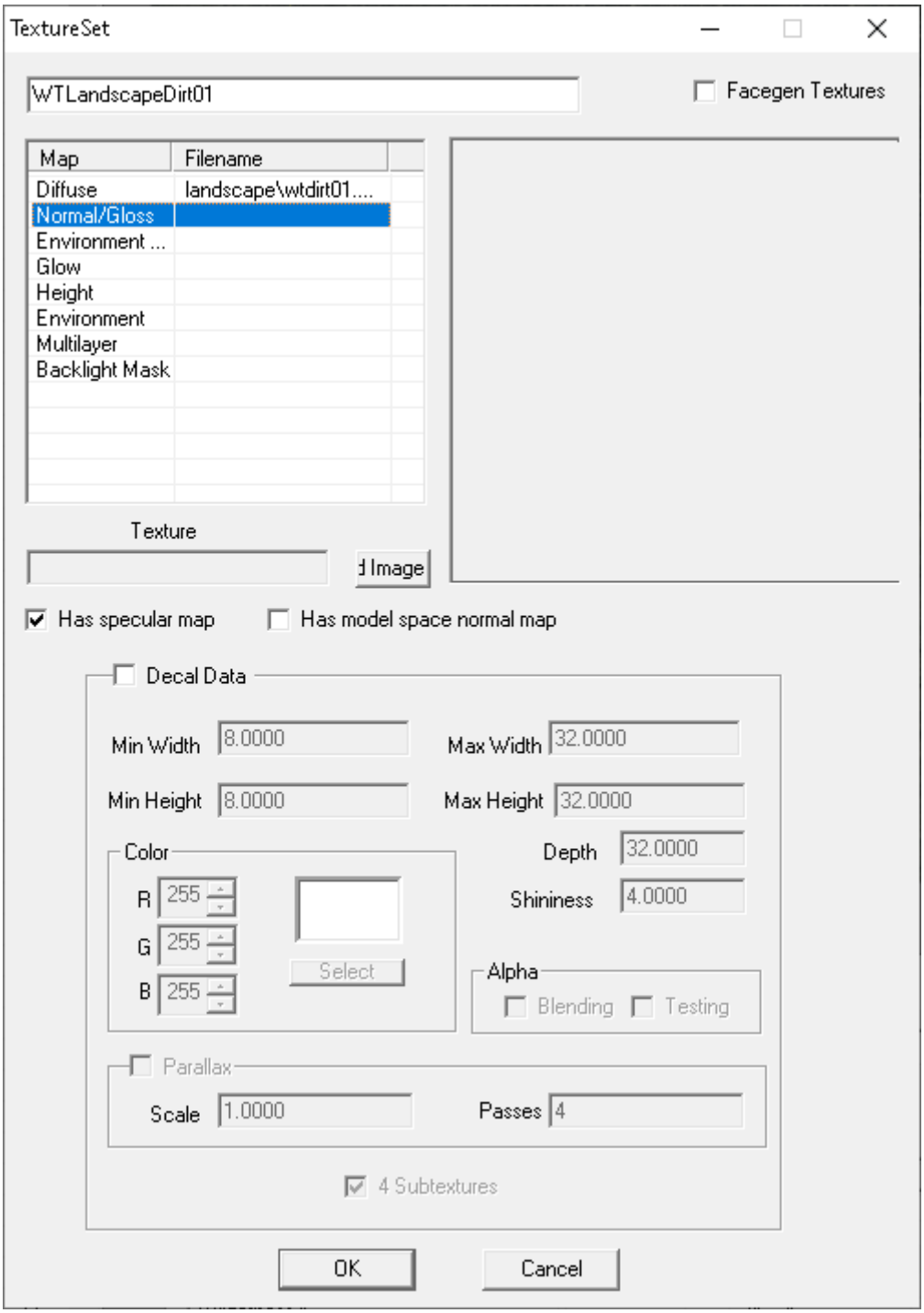


Figure 117 - Diffuse map added.

You should now have a diffuse and normal map added to the texture set as per the screenshot below.

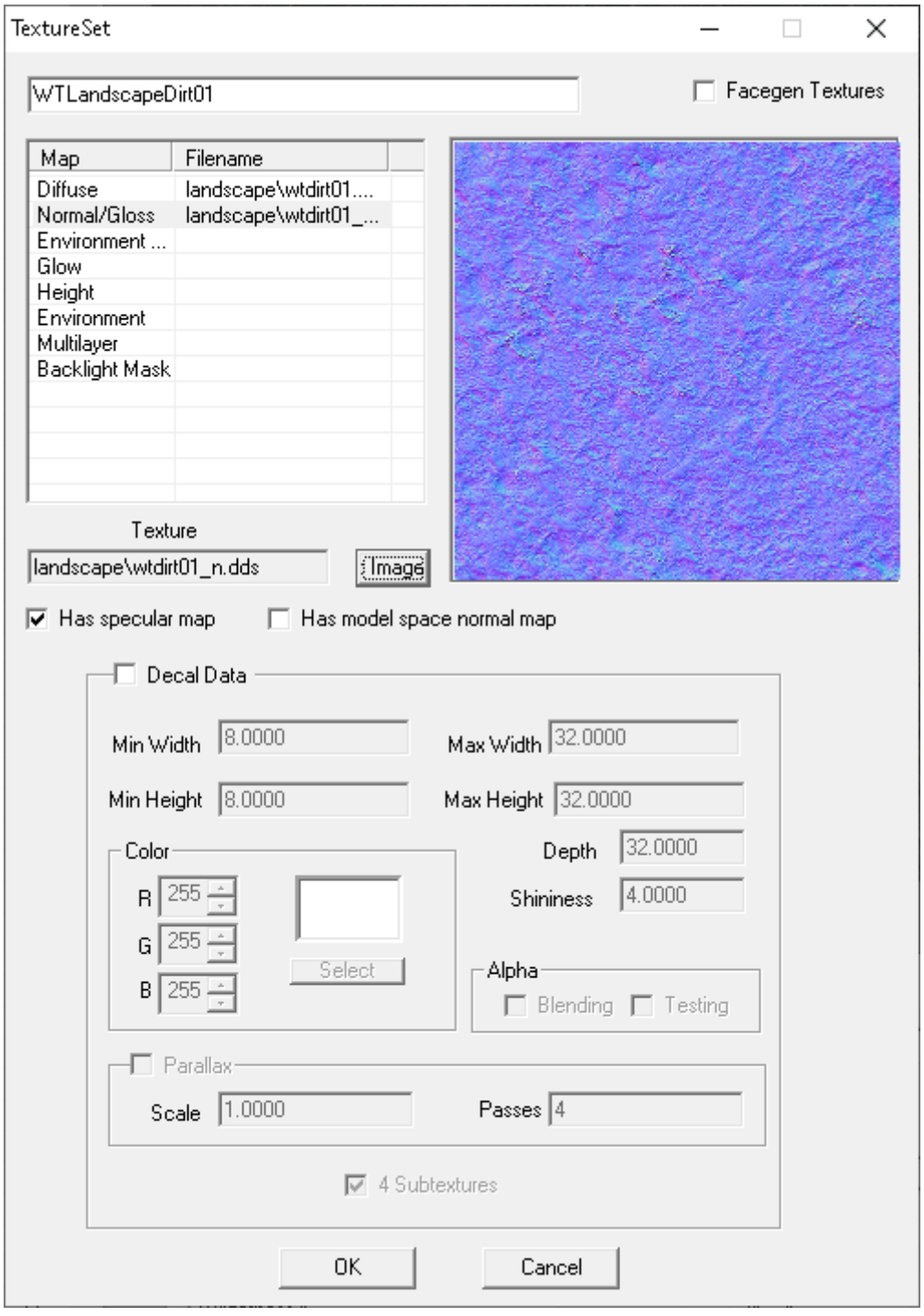


Figure 118 - Both diffuse and normal maps added.

Click OK to close out of the TextureSet properties.

Back in Object Window go to Miscellaneous > LandTexture.

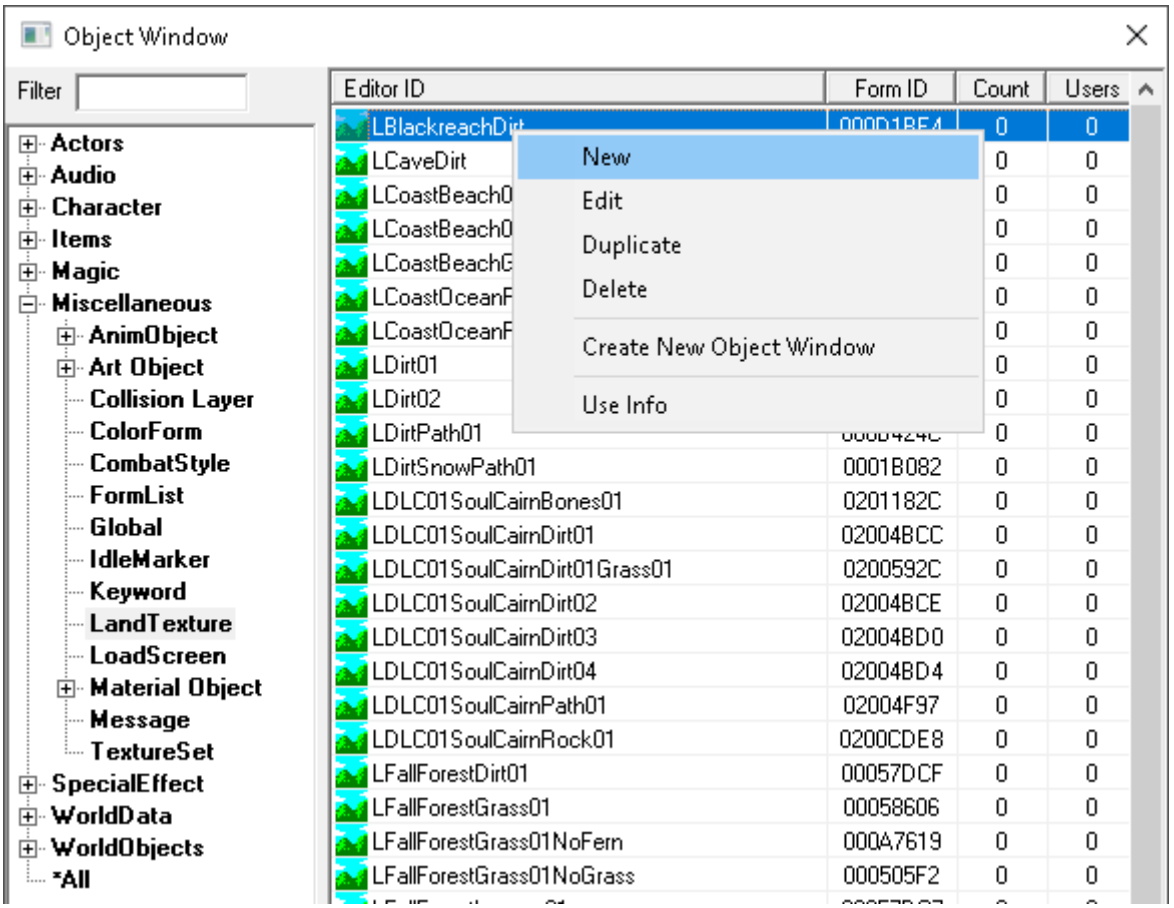


Figure 119 - Adding a new land texture.

Right-click on one of the existing land textures and select New.

Set the Texture Set drop-down to point to the texture set you created. For my example, I'm going to set the Material Type to 'MaterialSand', the Friction to '2' and the Restitution to '0'.

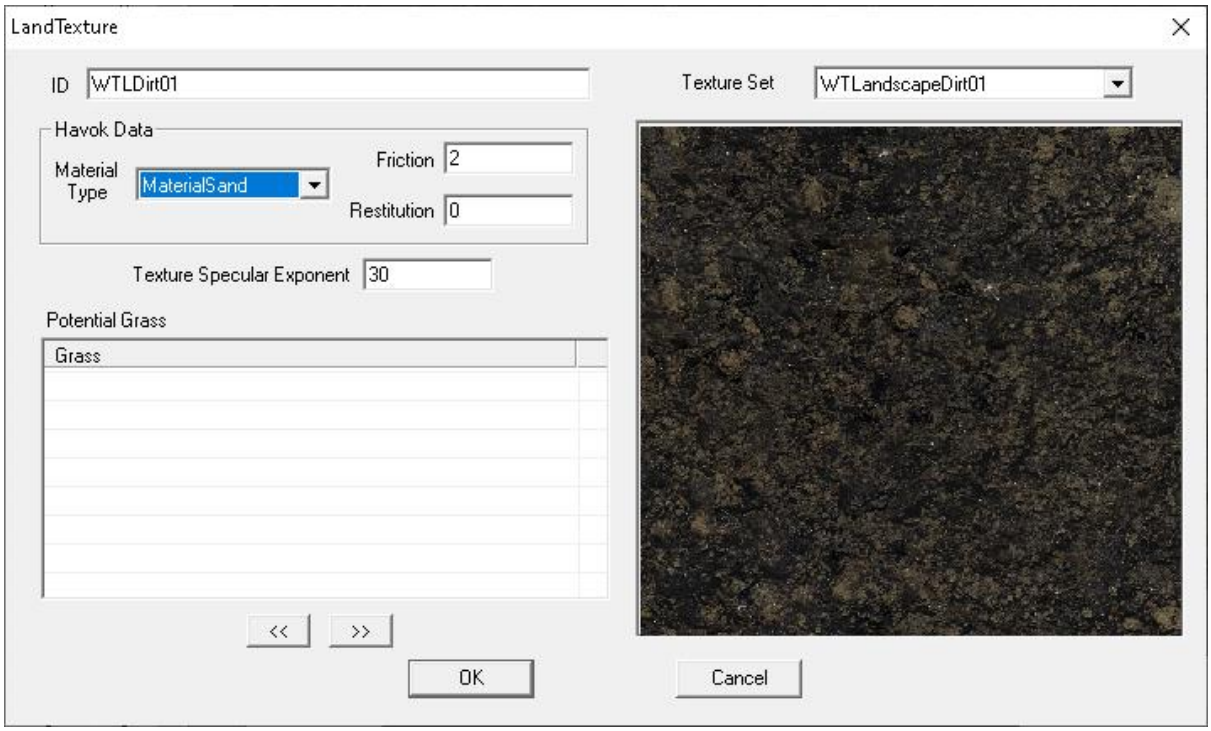


Figure 120 - Setting Land Texture properties.

To get the right values, I'd recommend taking a look at a similar existing texture.

Let's say we also wanted to add grass or rocks as we paint with our new landscape texture.

In the Object Window, browse to WorldObjects > Grass.

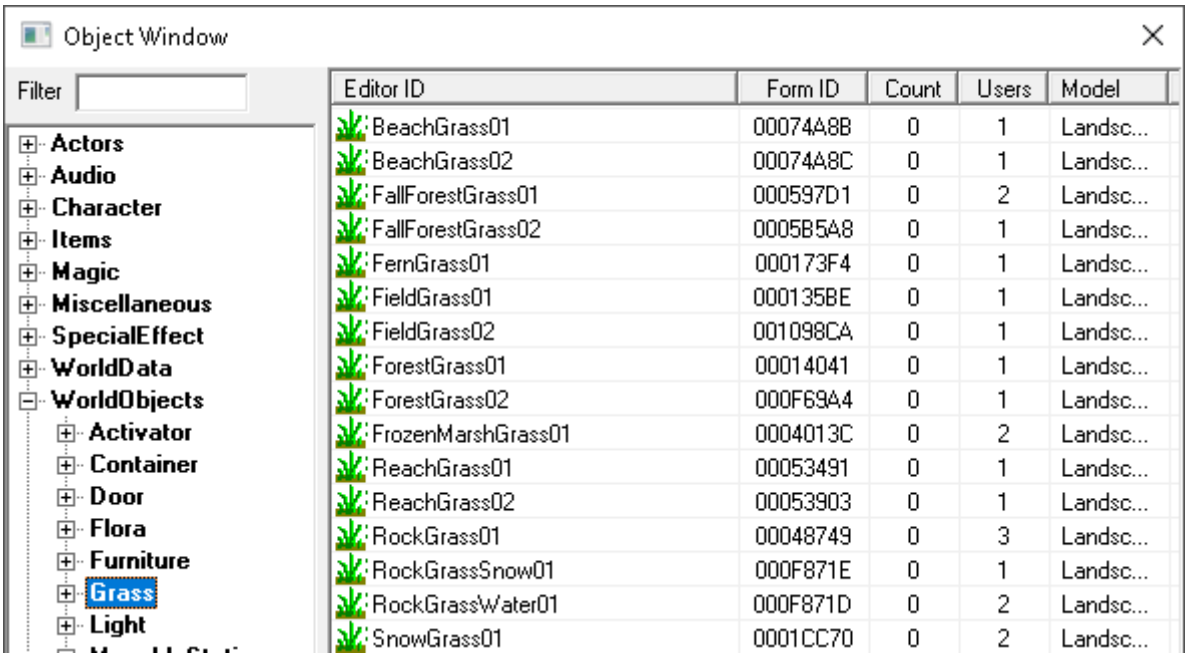


Figure 121 - Grass.

Highlight the object you want the landscape texture to potentially create, and drag and drop it into the Potential Grass field.

You can drag and drop multiple selected grass types at a time.

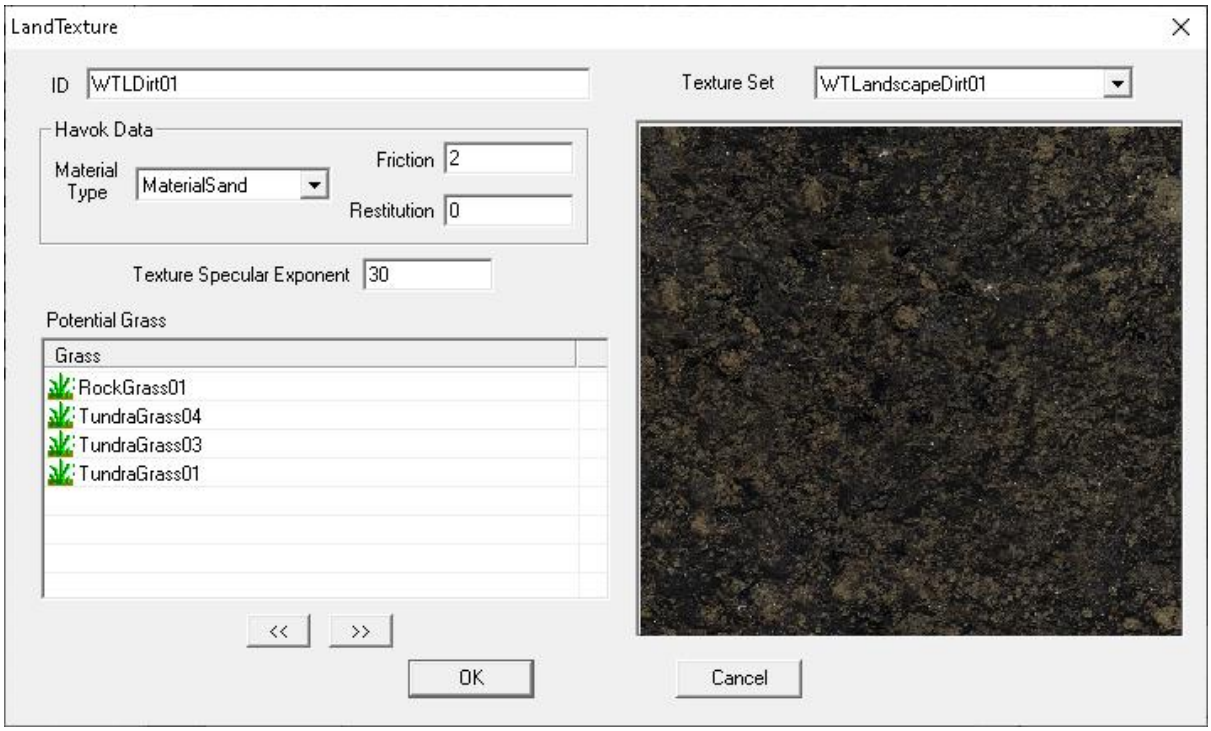


Figure 122 - Potential Grass added to our landscape texture.

Click OK.

Now we can start painting with our new landscape texture.

Important: The Creation Kit may crash after adding grass to a landscape texture, so make sure you save immediately and relaunch the Creation Kit.

And there's our new texture painted on the landscape, with grass.

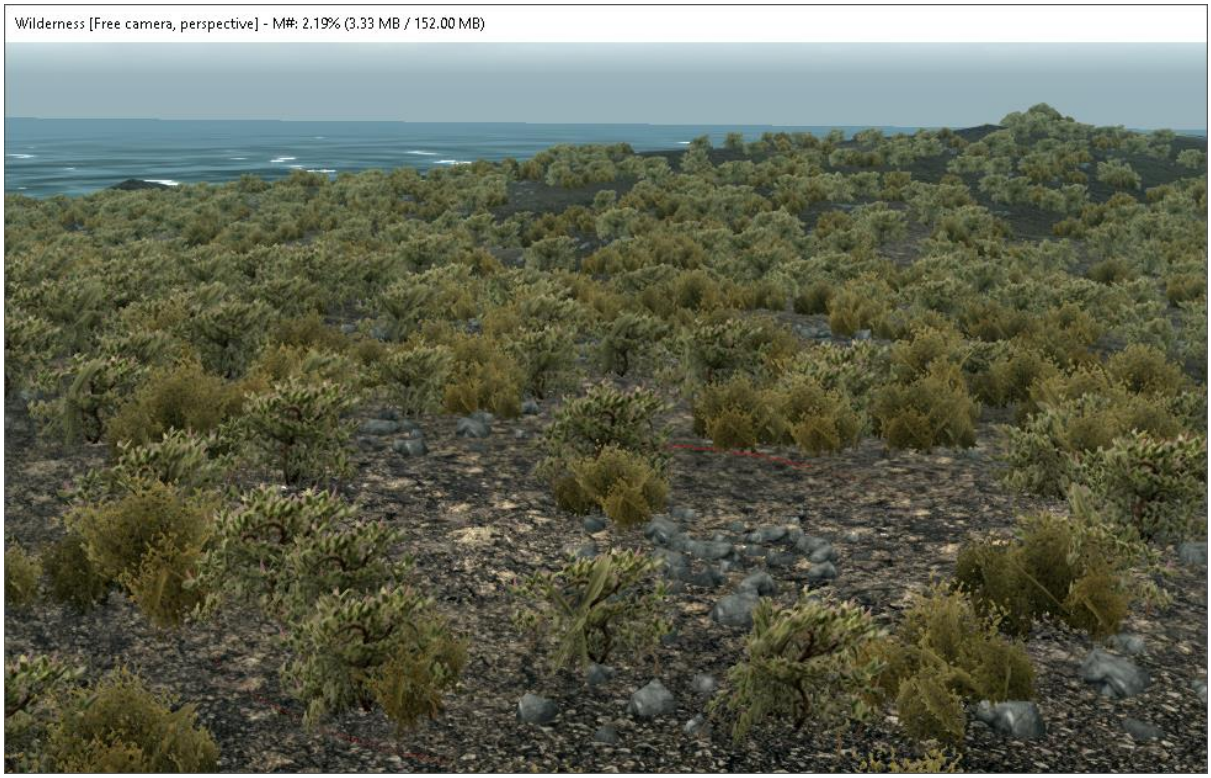


Figure 123 - Our new texture painted on the landscape.

Click on the 'Toggle Grass' button in the toolbar or press '8' to toggle grass off to see the textured landscape without it.

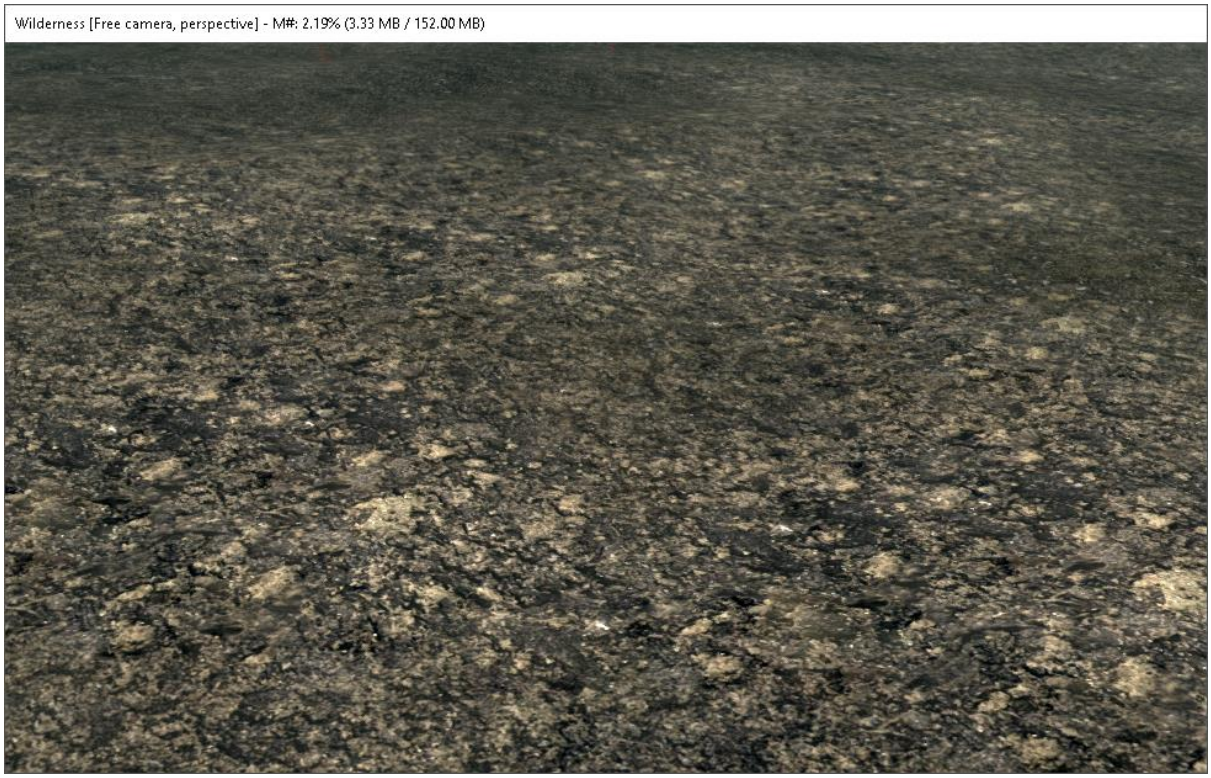


Figure 124 - Our new texture with grass toggled off.

ADDING LAKES AND RIVERS

Lakes and rivers can be added to your world space in a couple of different ways.

In this first example, I'm going to use a water plane. Water planes are useful for adding small lakes or rivers that are either above or below sea level.

In the Object Window, go to WorldObjects > Activator and filter by 'water'.

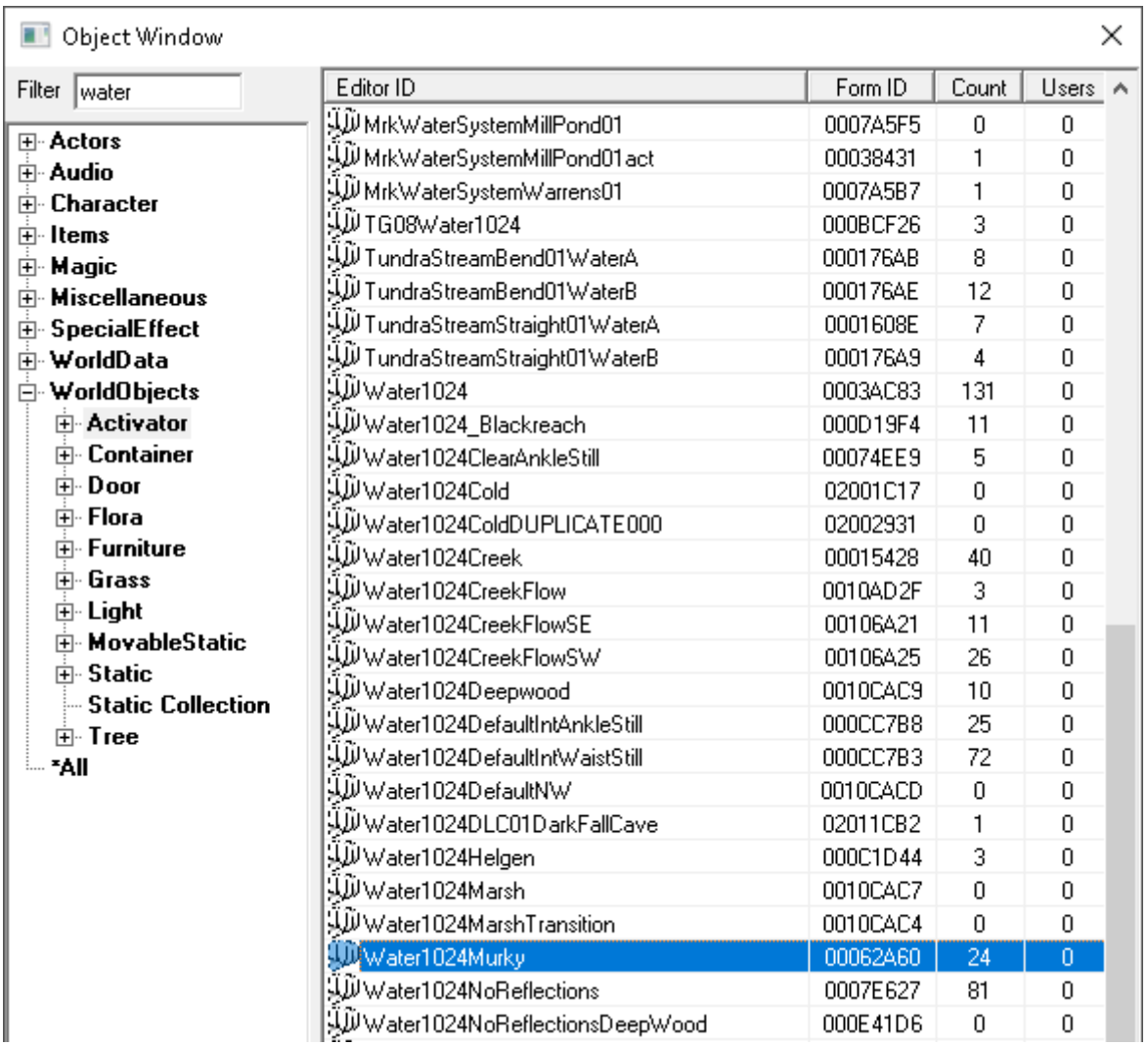


Figure 125 - Water planes.

I'm going to use Water1024Murky in this example because it doesn't flow in any particular direction, which is more realistic for filling a small pond.

Drag and drop the water plane into the render window.

Here you can see the water plane floating above the small pond I sculpted into the terrain.

To raise or lower the plane, hold down 'Z' while left-click dragging the mouse to lock movement to the Z axis.

Note: The water transparency may not update immediately after moving a water plane up or down, so press 'F5' to refresh the render window.

As you can see, the water plane is too small. So let's resize it so that it fills up the pond entirely.

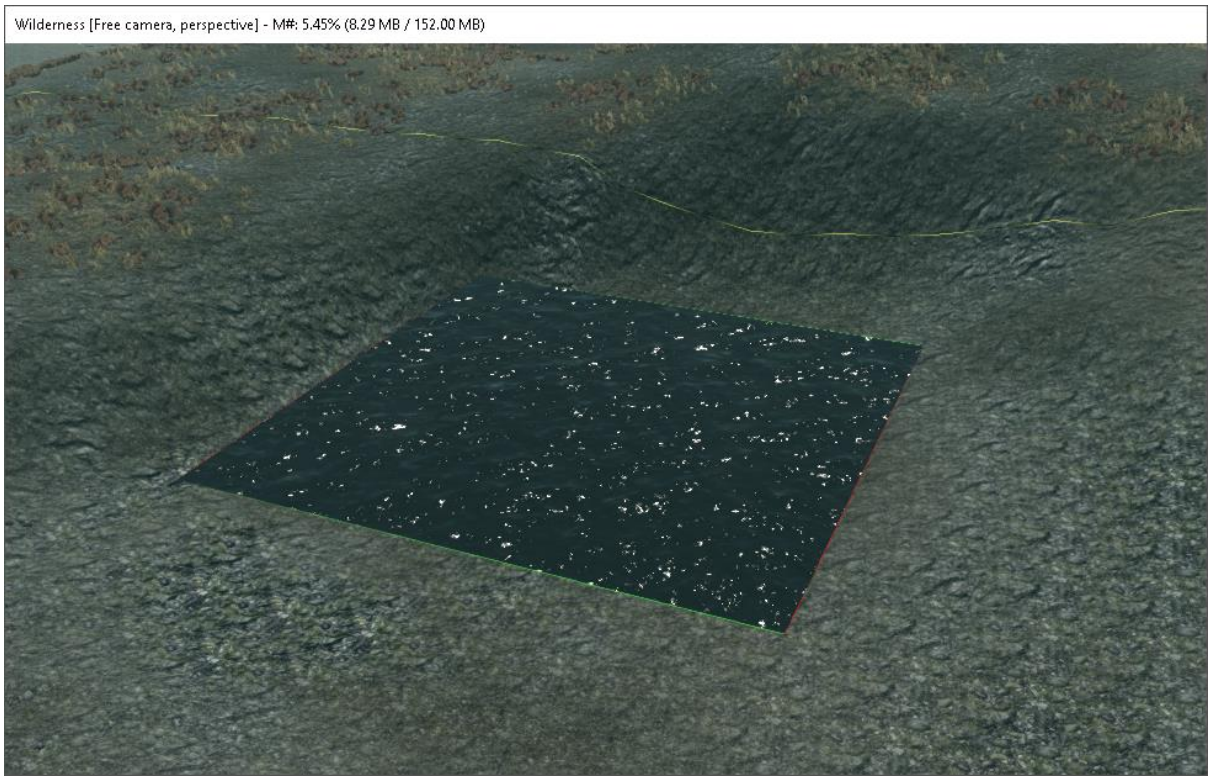


Figure 126 - Water plane added to our world space.

Double-click or right-click on the water plane and select Edit.

In the Reference properties, go to the 3D Data tab and change the Scale. In my example, I'm going to set the Scale to 3.0000.

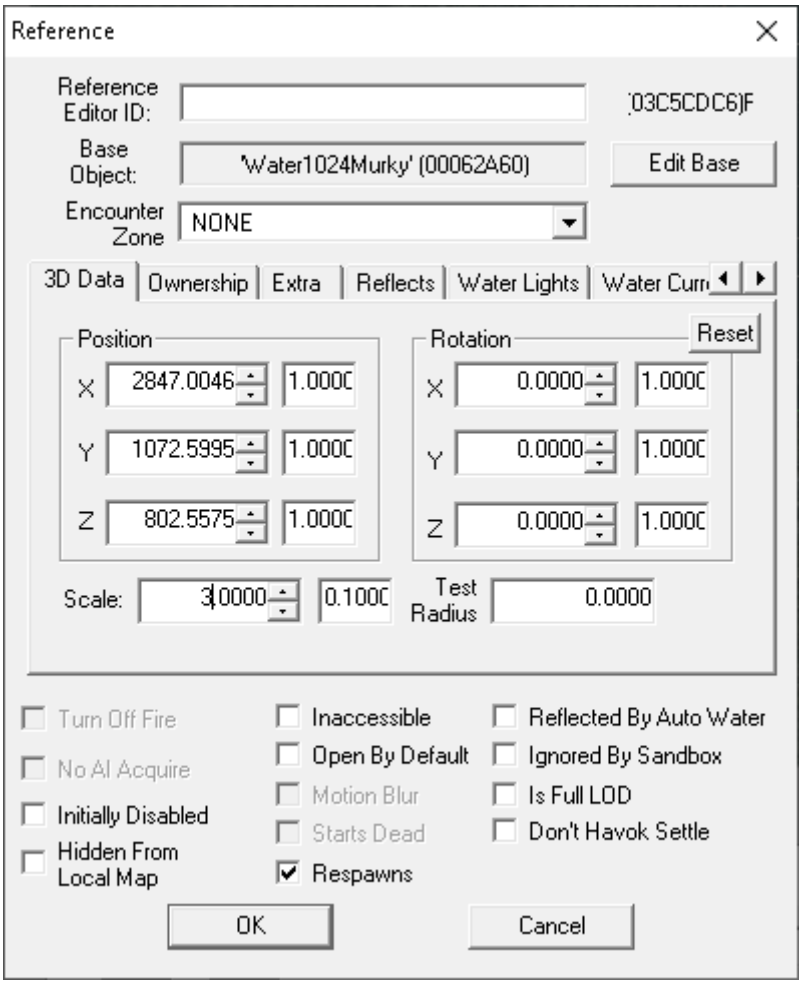


Figure 127 - Changing the scale of the water plane.

Click OK to close out of the Reference properties.

Note: Unlike increasing the scale of an object, increasing the scale of a water plane won't affect the scale of the water texture.

In the following screenshot, you can see that the water plane now fills the pond entirely.



Figure 128 - The water plane now reaches the edge of the pond.

Another way to fill a pond would be to turn on 'Snap to Grid' by pressing 'Q' and snap multiple water planes together. This may work better for ponds that are irregular in shape.

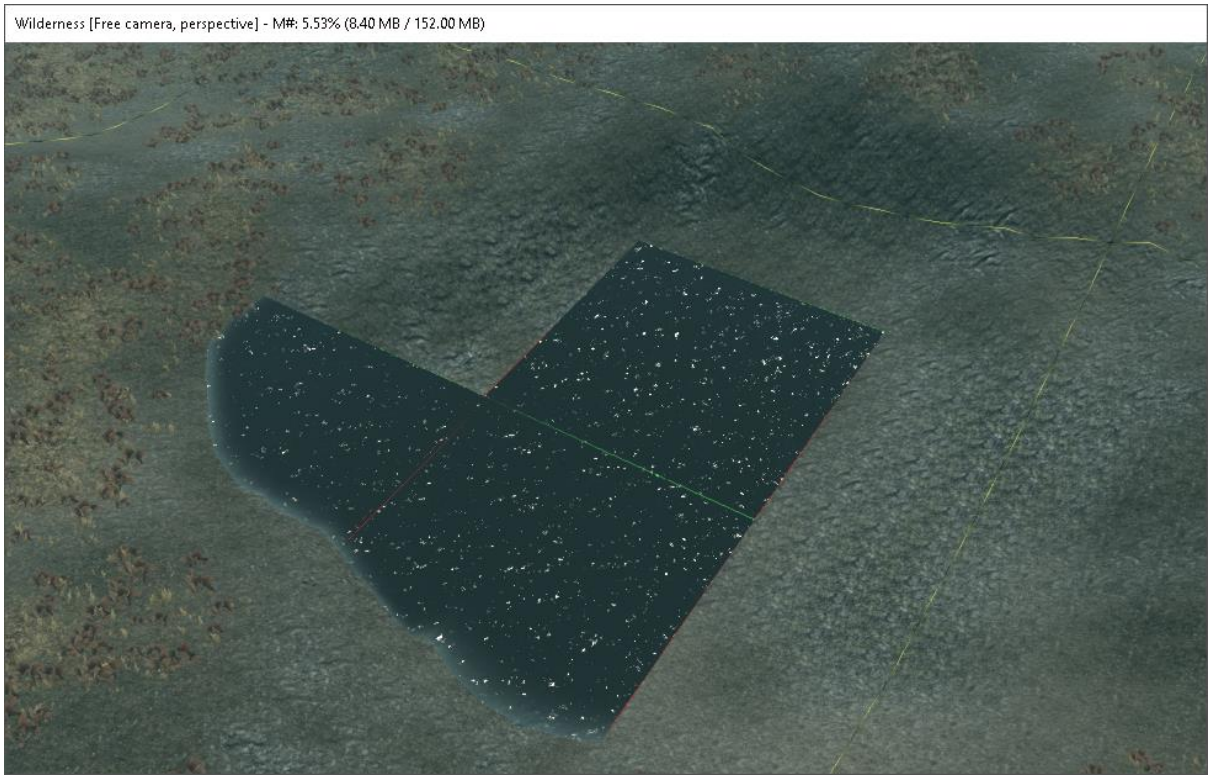


Figure 129 - Snapping multiple water planes together instead.

Another thing we can do is change the water type of the ocean layer on a per cell basis. Generally you'd want to do this for large bodies of water like lakes or swamps.

In the Cell View, right-click on the cell you want to change the water type for and select Edit.

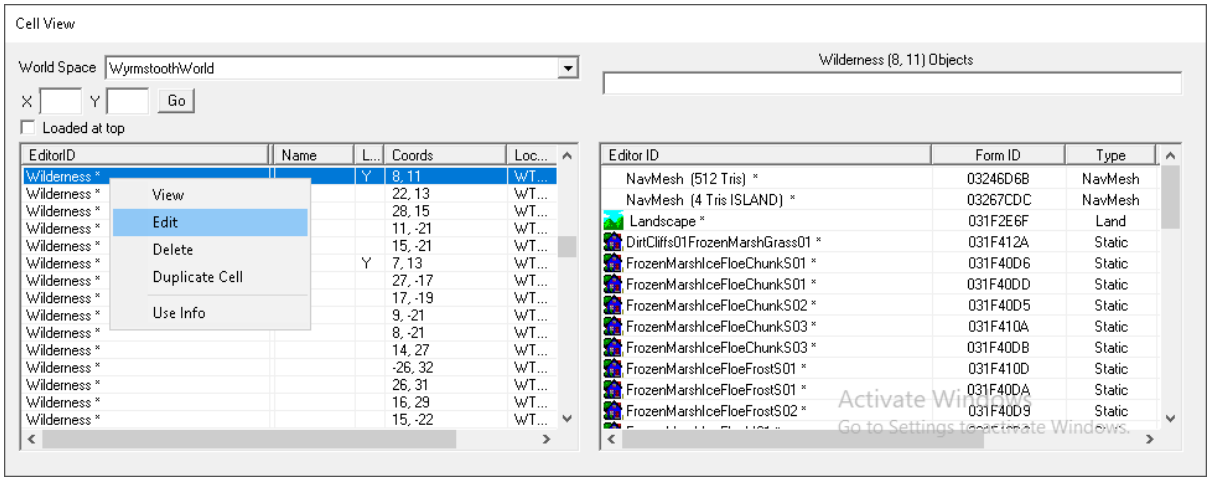


Figure 130 - Editing cell properties.

Change the Water drop-down. In my example I set it to MurkyWater.

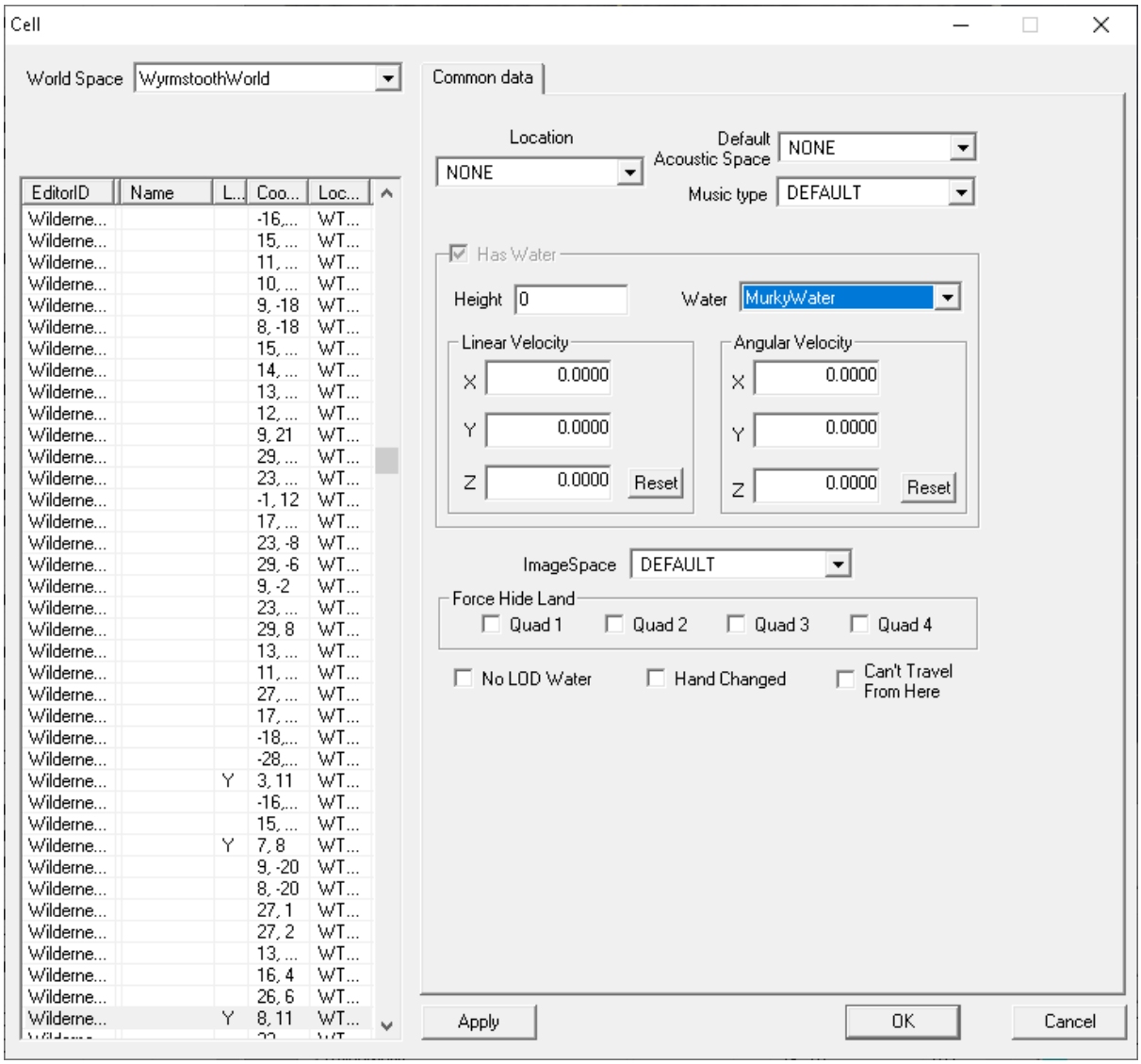


Figure 131 - Changing the water type of a specific cell.

Click OK.

And here's a screenshot of a cell with its water type changed:

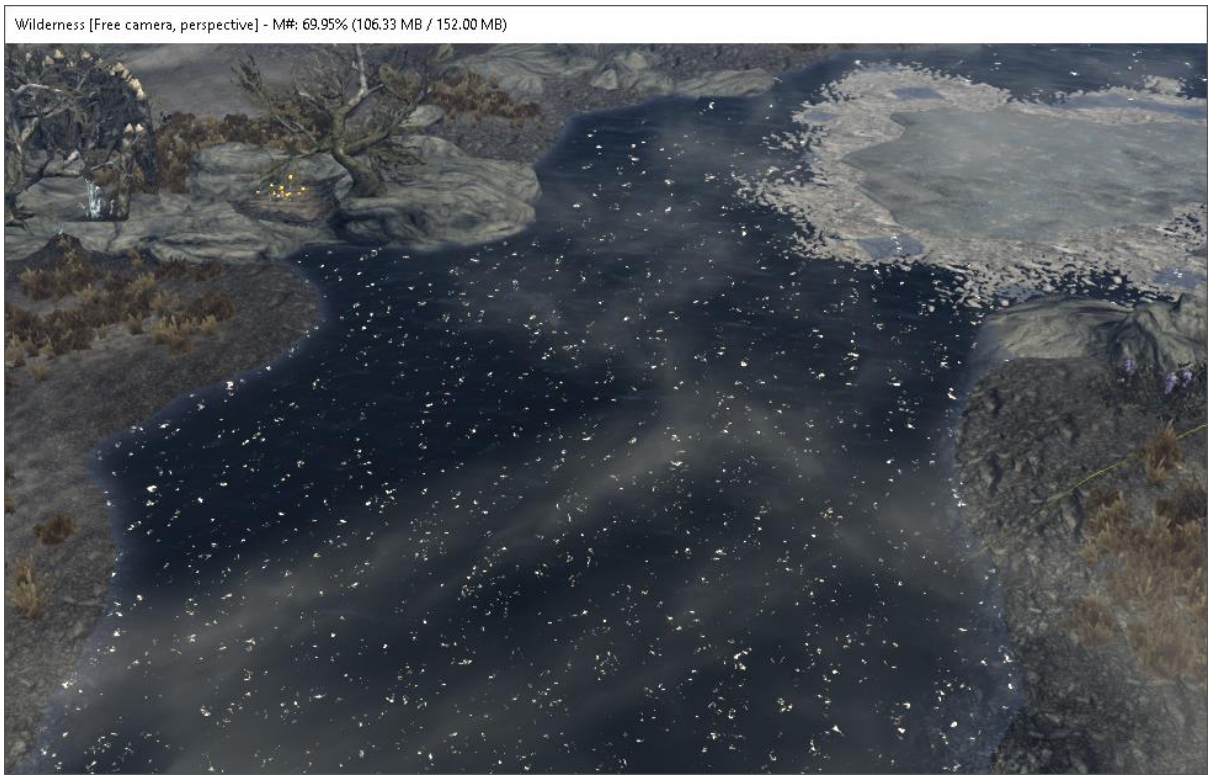


Figure 132 - Water type for cell has changed.

Note: If your lake or river spans multiple cells, you'll need to change the water type in each cell.

If your river or lake borders the ocean, you'll need to be a bit creative with how you cover up the transition from the water type you're using to the default ocean water type.

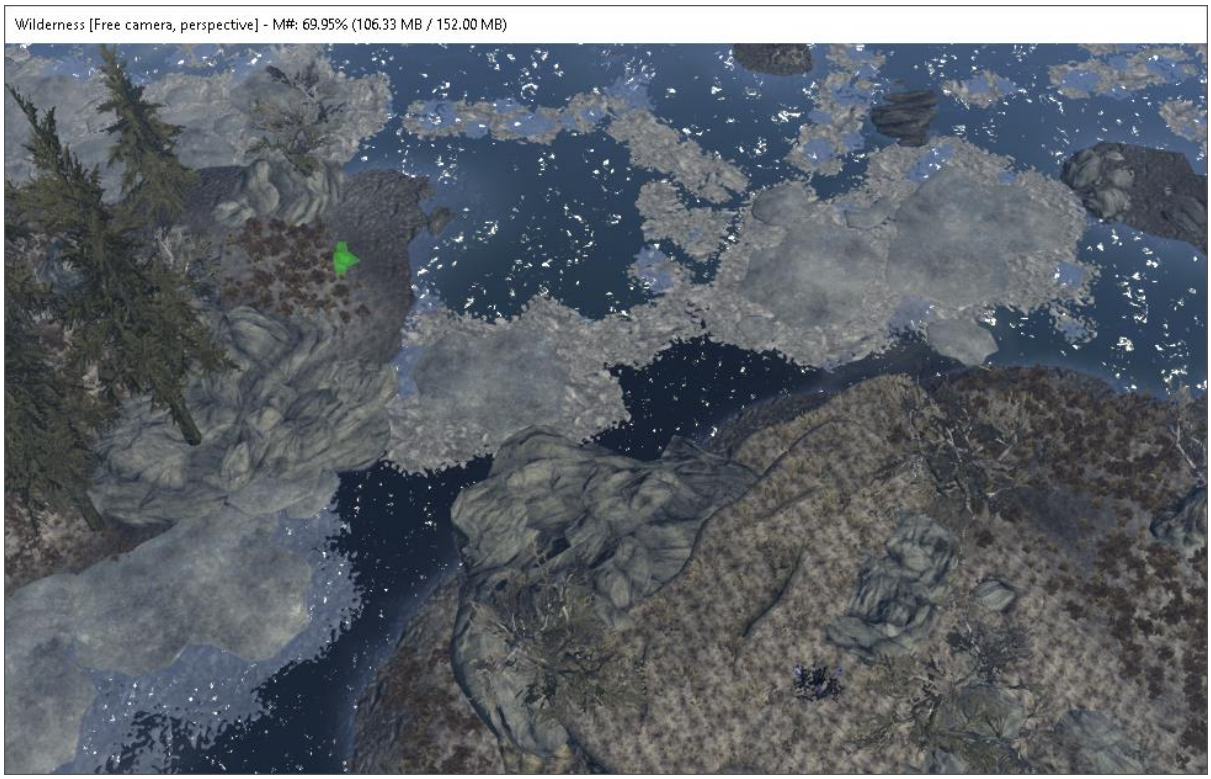


Figure 133 - Where two different water types meet.

In the screenshot above, I added some ice over the top of the cell border to cover the visible seam between two different water types.

Here's that screenshot again, but with the ice objects removed to show you what that seam between two different water types along the cell border looks like:

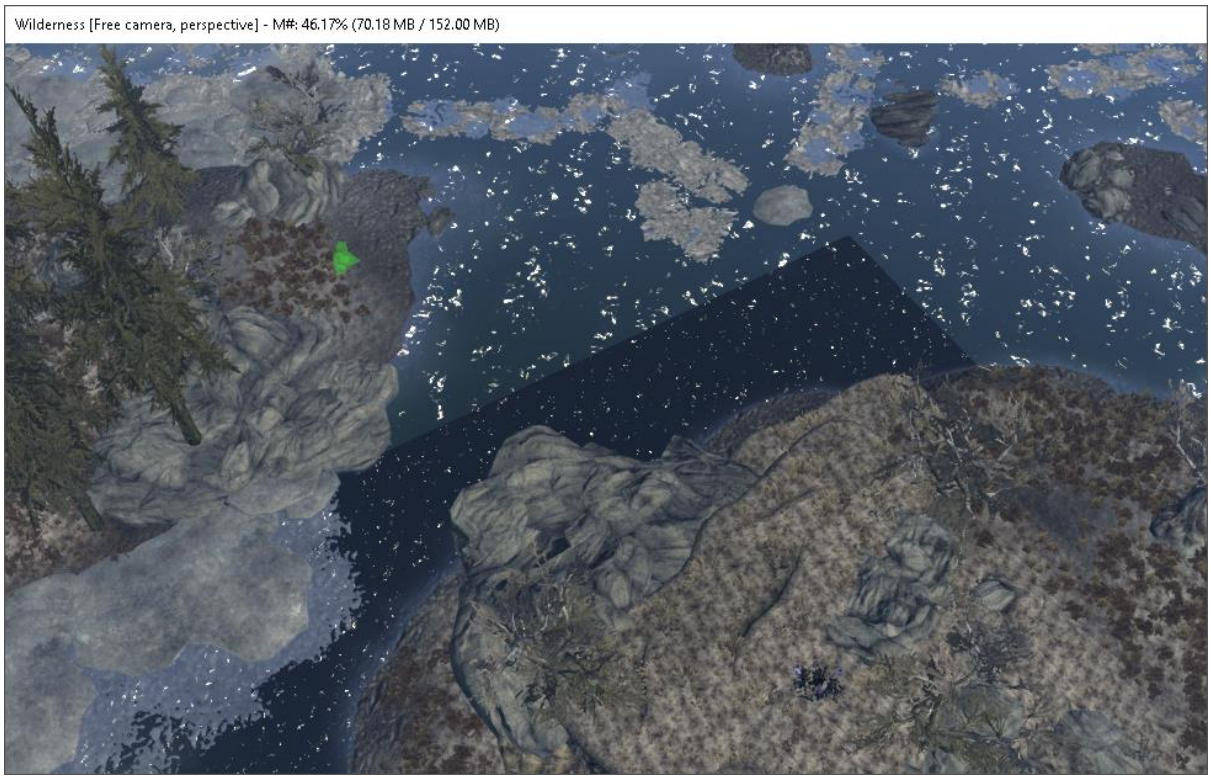


Figure 134 - Ice removed.

You can also change the height of the water in cell properties.

In the following example, I set Height to -6000.

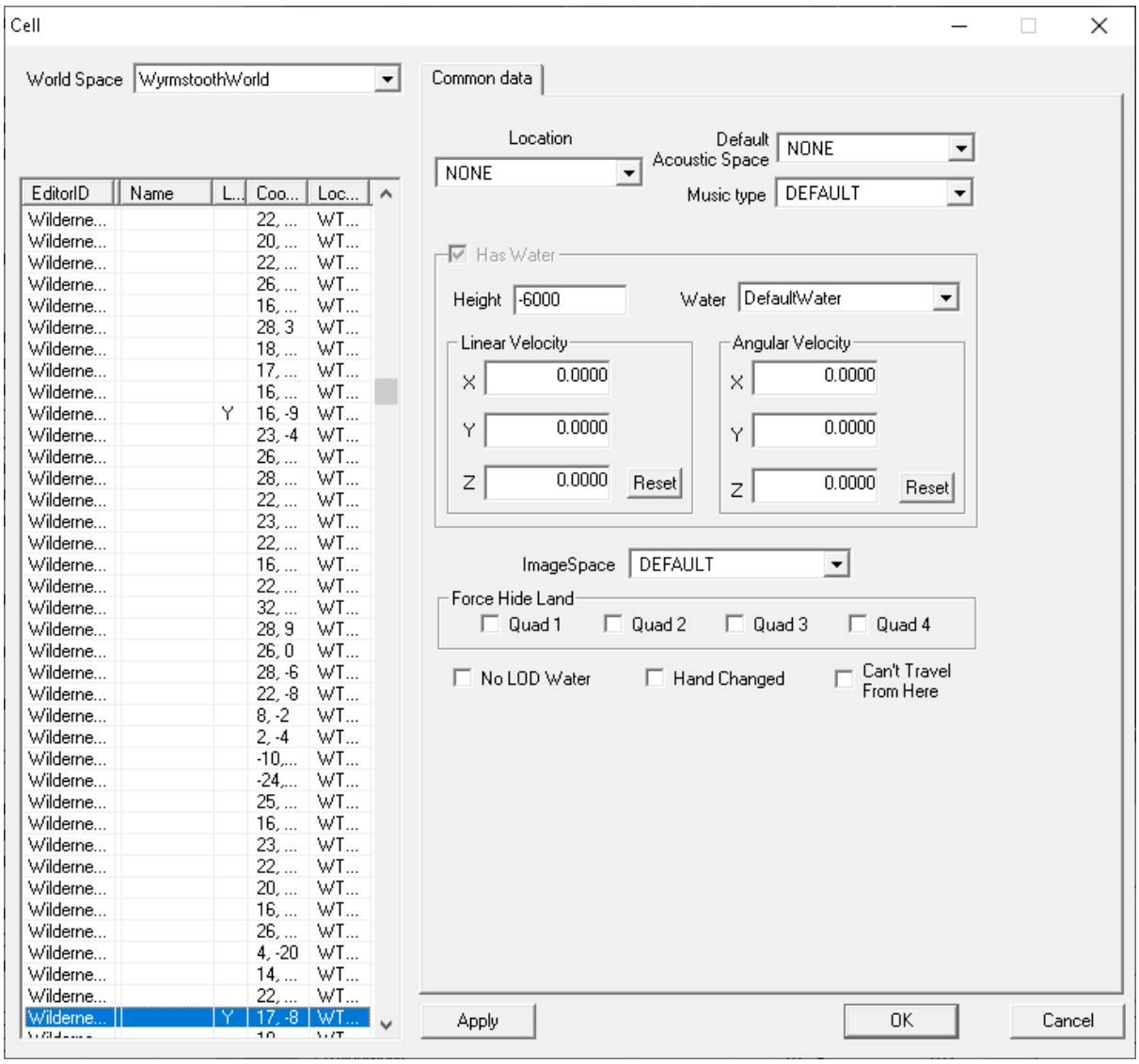


Figure 135 - Lowering a cell's water.

By lowering a cell's water below the default sea-level, I can have areas of the map that are below sea-level but are not underwater.

For example, on the left hand side of the following screenshot, I have a deep pit that the player can walk down into. This pit is below sea-level so I had to reduce the cell's water height by -6000 units.

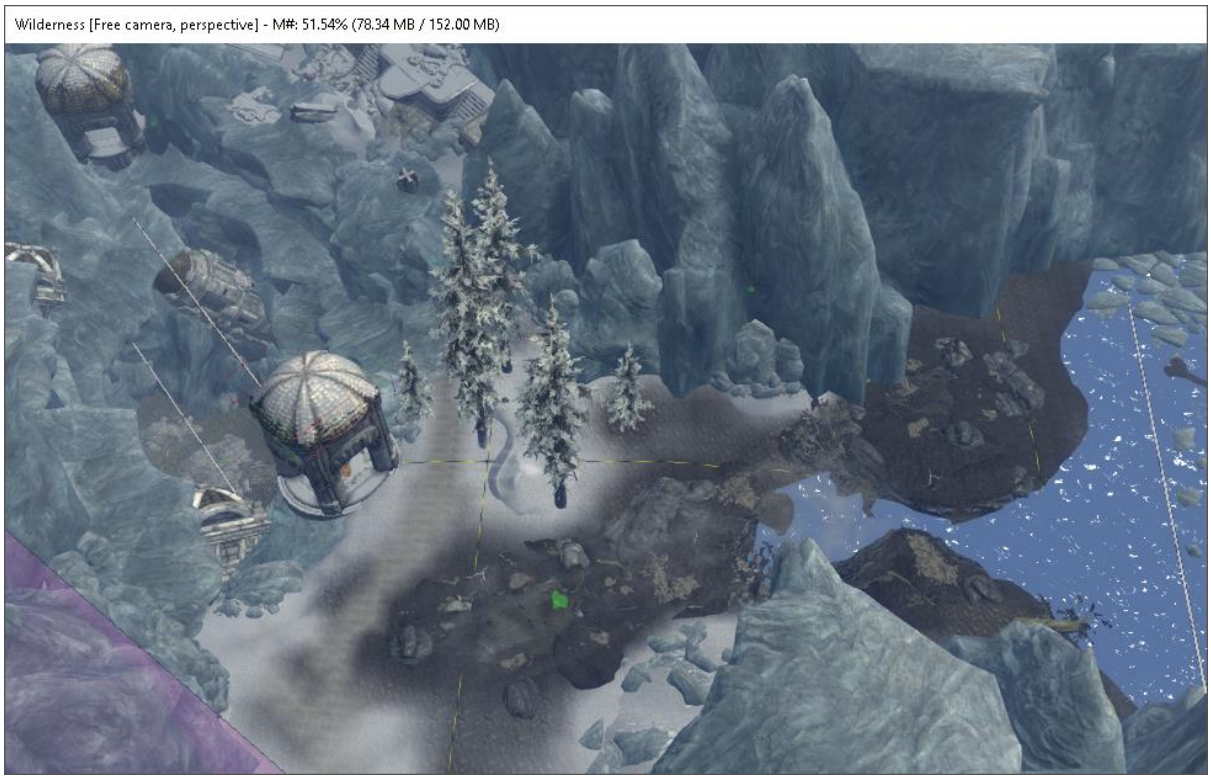


Figure 136 - An area of the map below sea-level.

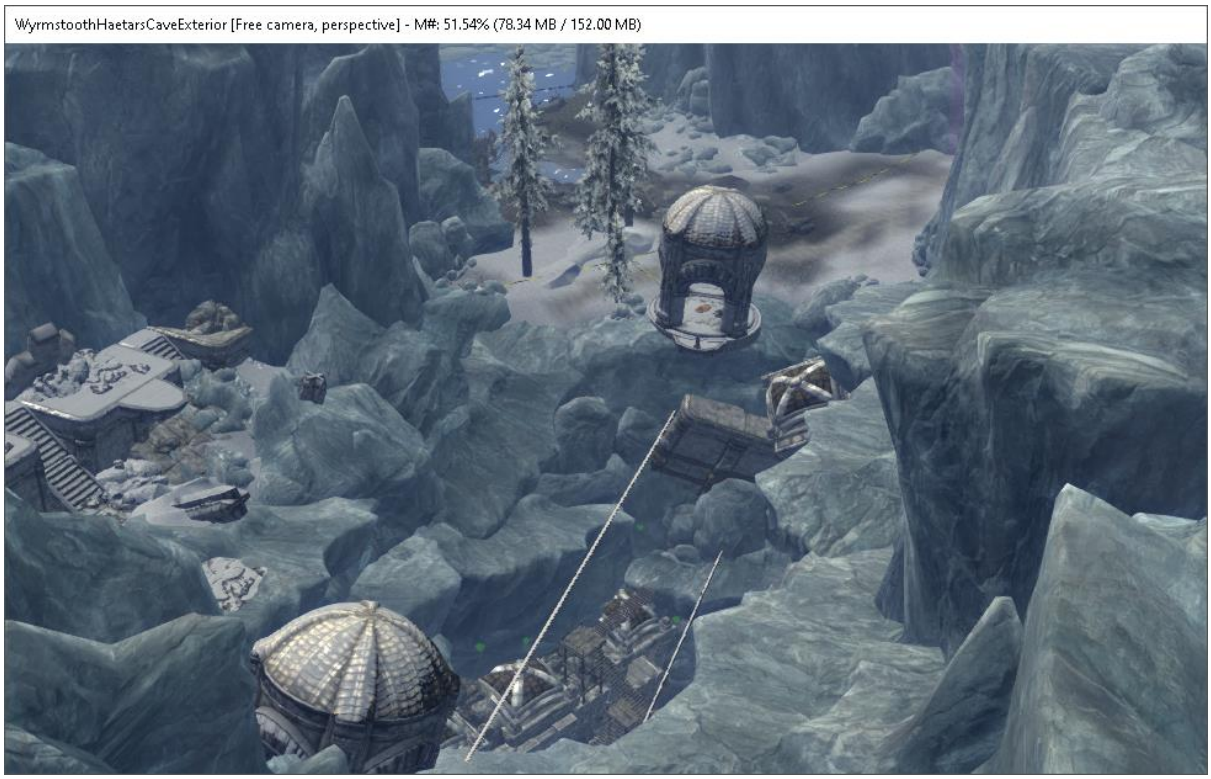


Figure 137 - The pit from another angle.

SETTING UP WATER FLOW AND CURRENTS

Skyrim.esm includes several water planes for creeks and rivers for different flow directions, e.g.: Water1024CreekFlow, Water1024CreekFlowSE and Water1024CreekFlowSW.



Figure 138 - Water planes with different flow directions.

But what if we need a water plane to flow in a very specific flow direction, or with a stronger current? We'll need to create our own.

Note: In this section I'll be talking about water flow and currents. Just so we're on the same page, when I mention water flow I'm talking about the direction of the water shader on the surface of the water plane. When I mention current I'm talking about the direction that NPCs or physics-enabled objects would be pulled in if they fell in.

To do so, firstly browse to WorldData > WaterType.

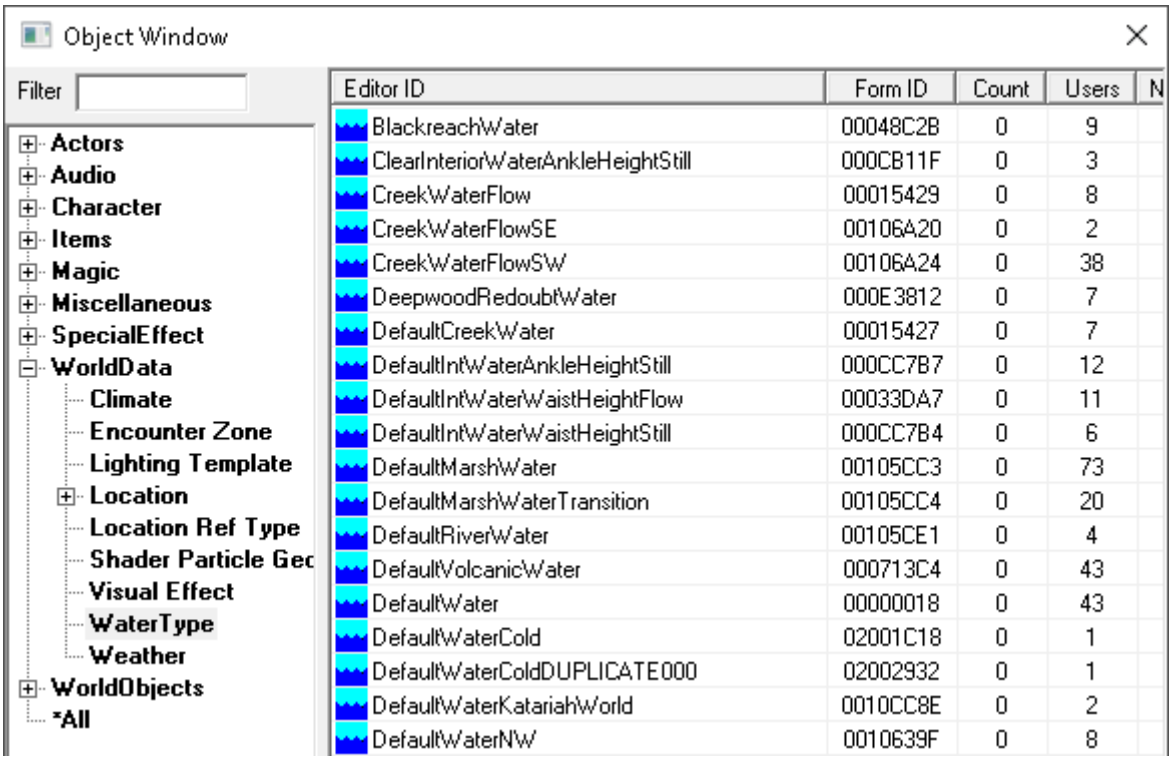


Figure 139 - Default water types.

For this example, I want to specify the direction of a flowing creek.



Figure 140 - A sample creek, with water flowing from right to left.

Right now, in the example above, I'm using Water1024Creek. The water flows from right to left, but I want to change this direction.

Right-click on CreekWaterFlow and select Duplicate.

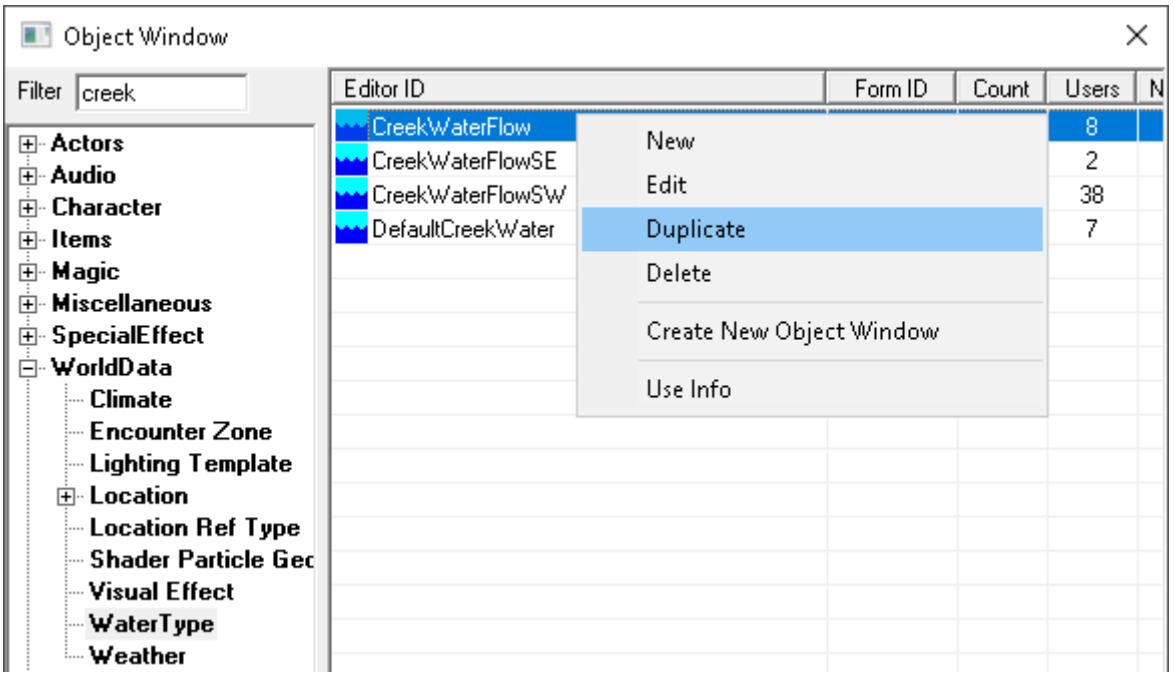


Figure 141 - Duplicating an existing water type.

Rename your water type. In my example, I just named it CreekWaterFlow with a 'WT' prefix.

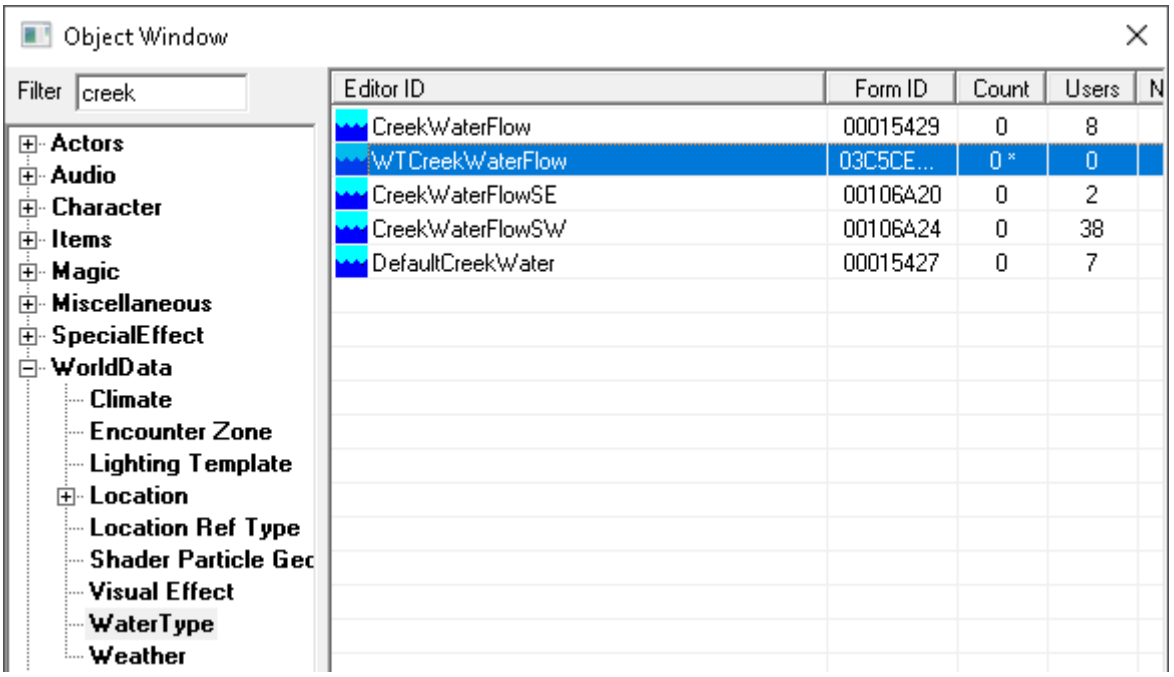


Figure 142 - Renaming our new water type.

Go to the Noise Properties tab.

Set the Wind Direction slider for Noise Layer One to 360.

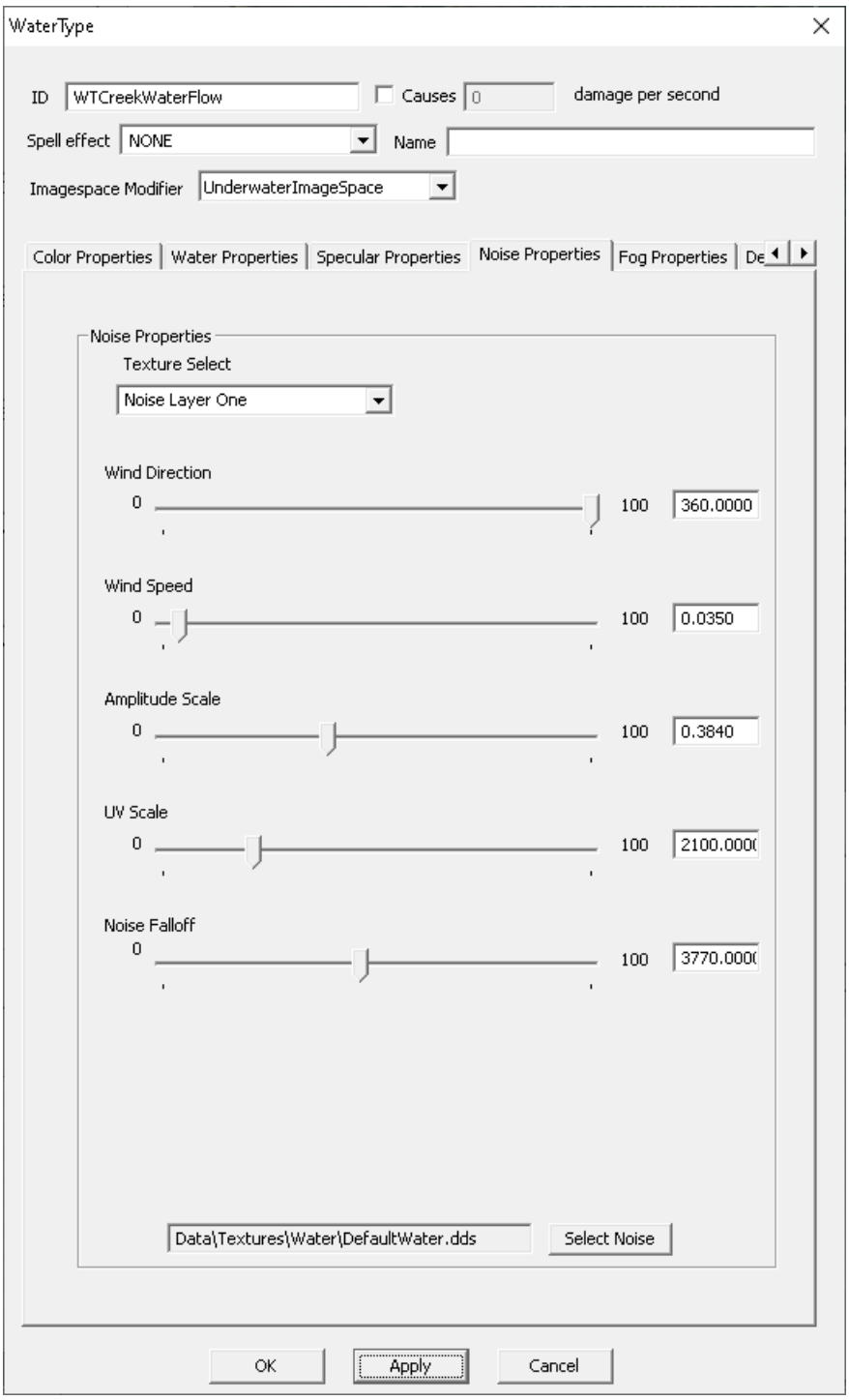


Figure 143 - Noise Layer One direction.

Set the Texture Select drop-down to Noise Layer Two.

Set the Wind Direction to 300.

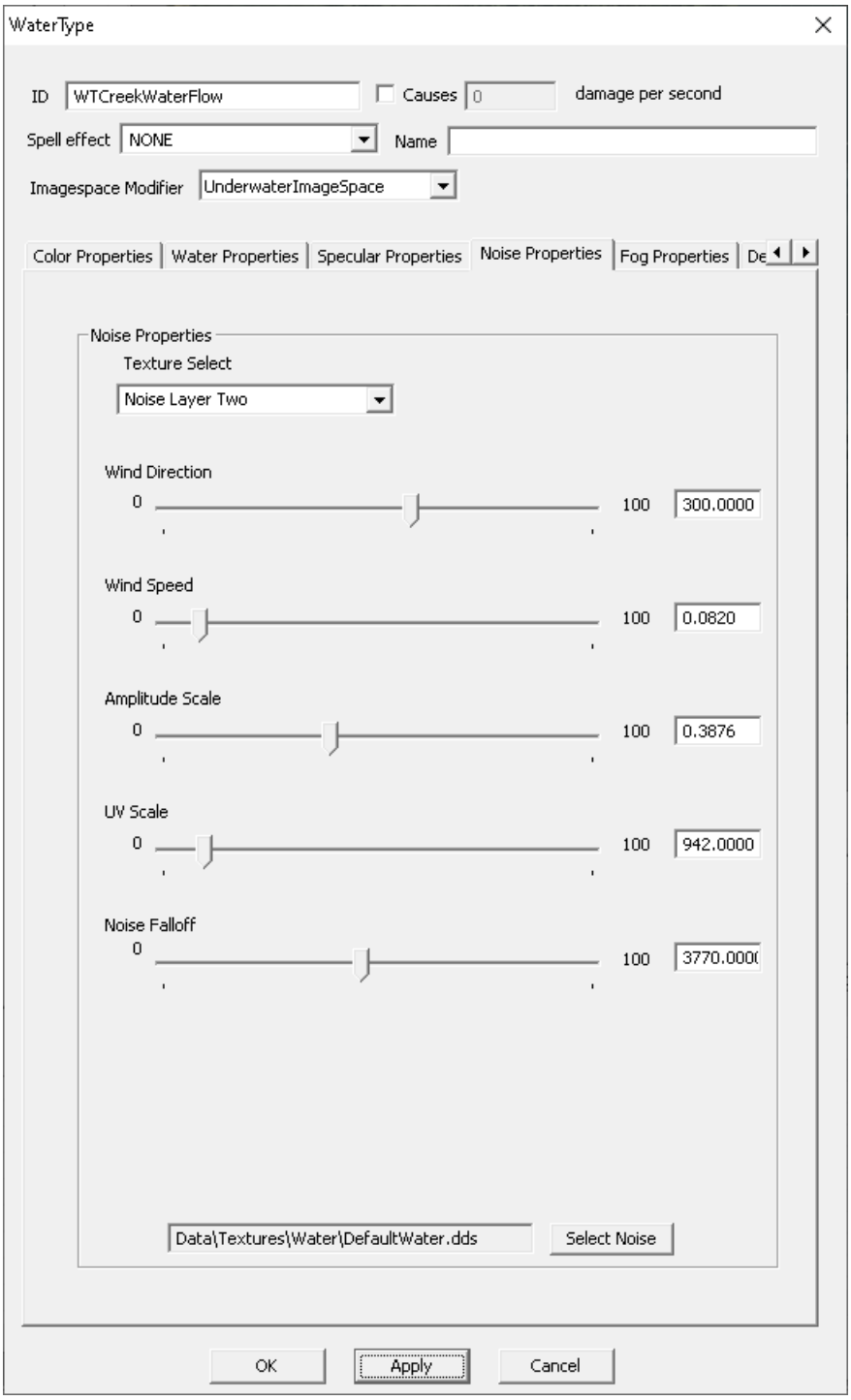


Figure 144 - Noise Layer Two direction.

Lastly, set the Texture Select drop-down to Noise Layer Three.

Set the Wind Direction to 300.

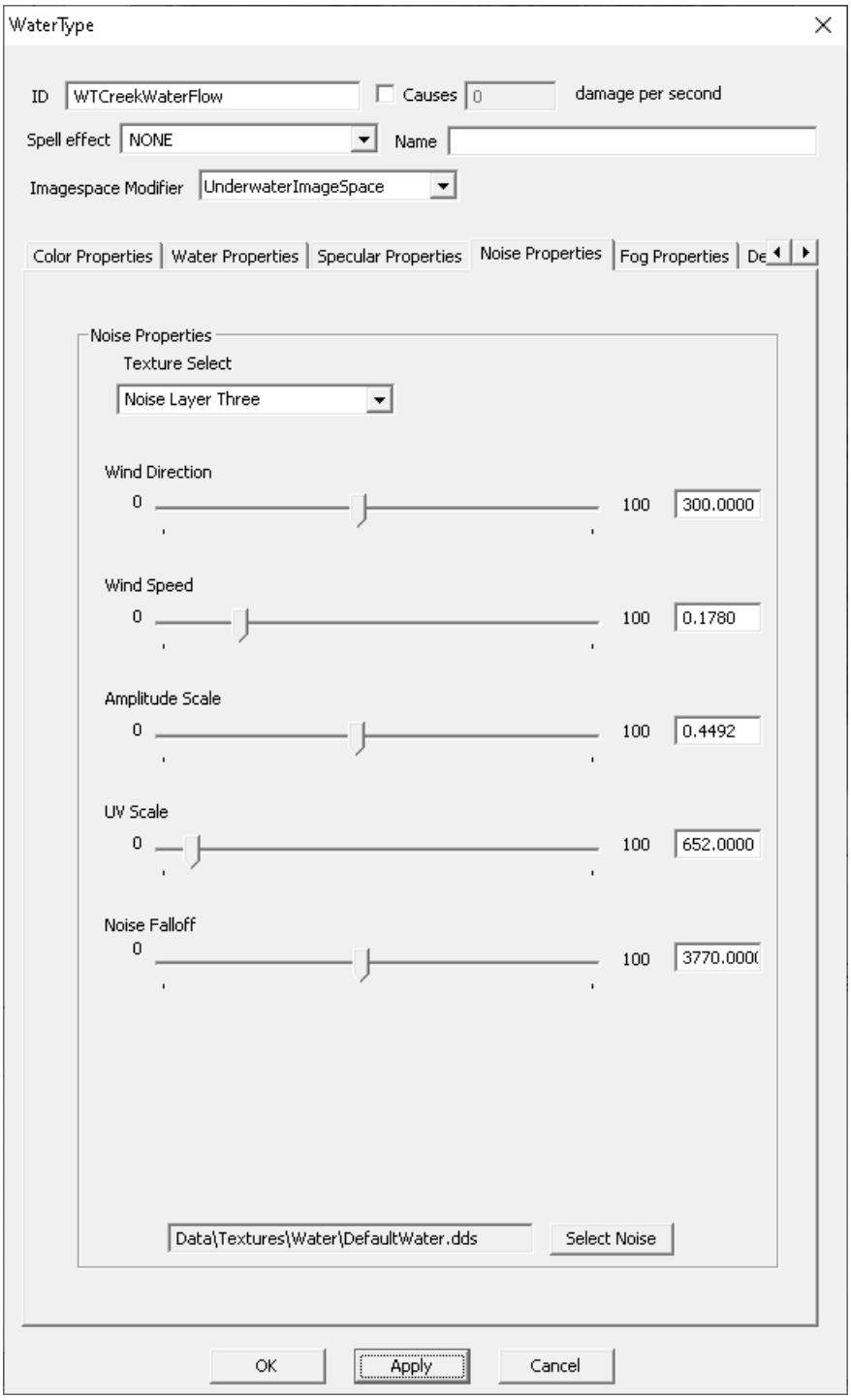


Figure 145 - Noise Layer Three direction.

Click OK to close the WaterType properties.

Next, we need to create a new water plane activator.

In the Object Window, go to WorldData > Activator.

Filter by 'creek'.

Right-click on 'Water1024Creek' and select Duplicate.

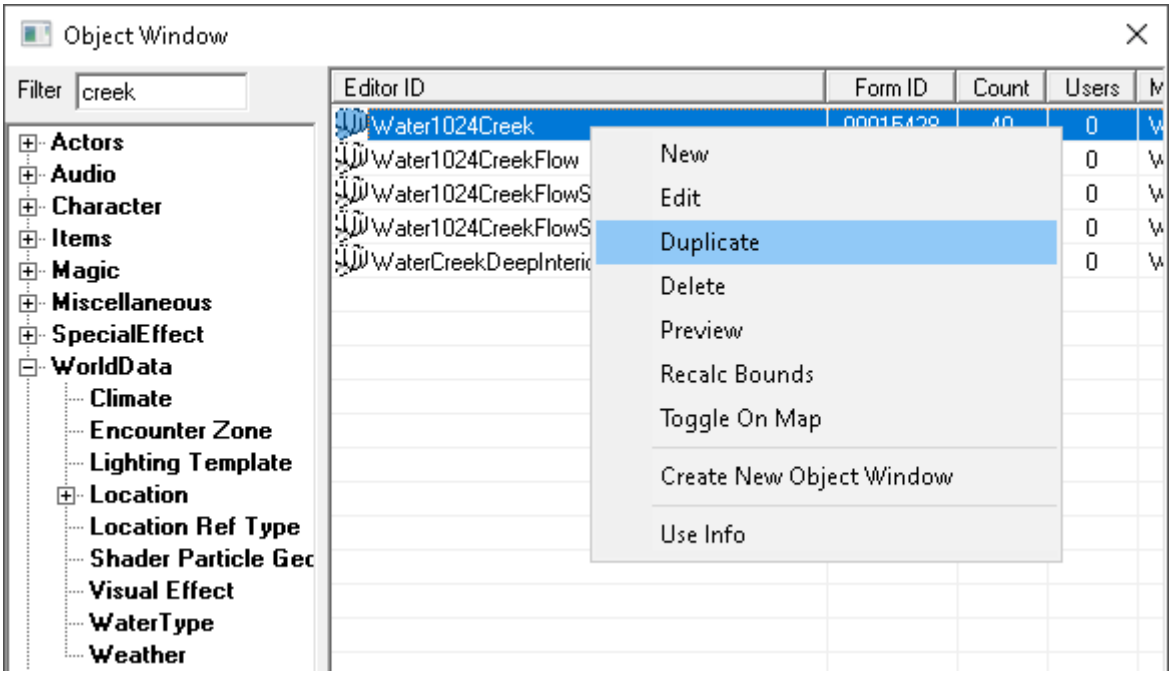


Figure 146 - Duplicating Water1024Creek.

For my example, I just named it Water1024CreekFlow with a 'WT' prefix.

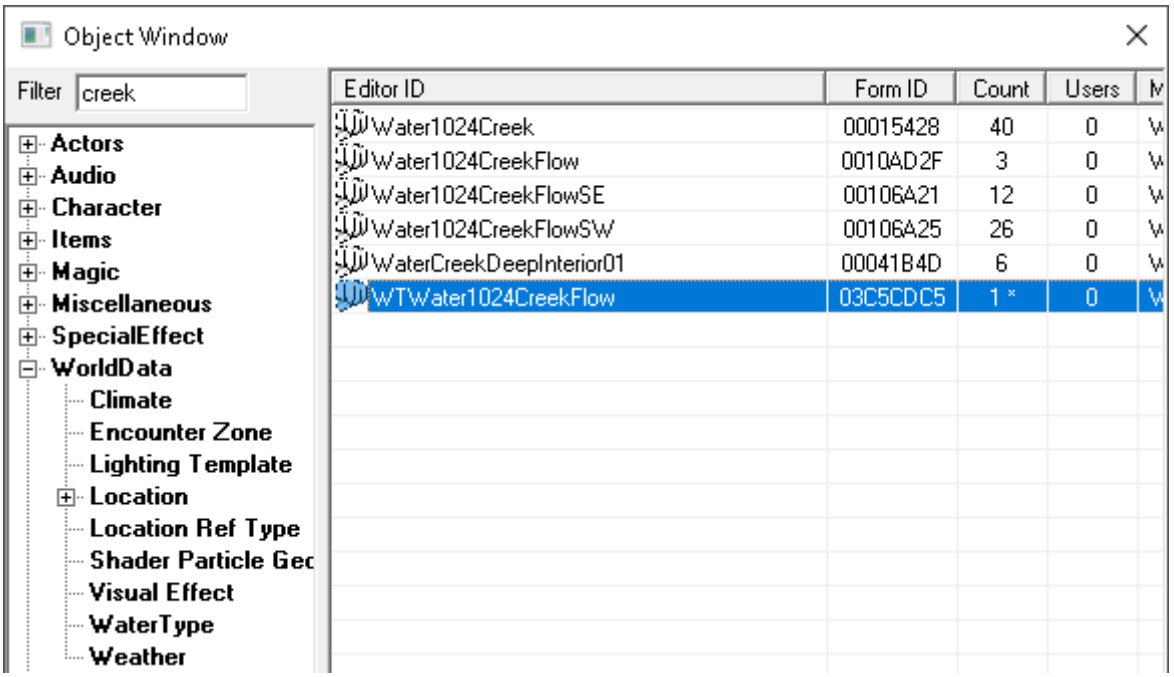


Figure 147 - Renaming the water activator.

Click on the water plane in the render window and press CTRL+F to bring up the Search & Replace screen.

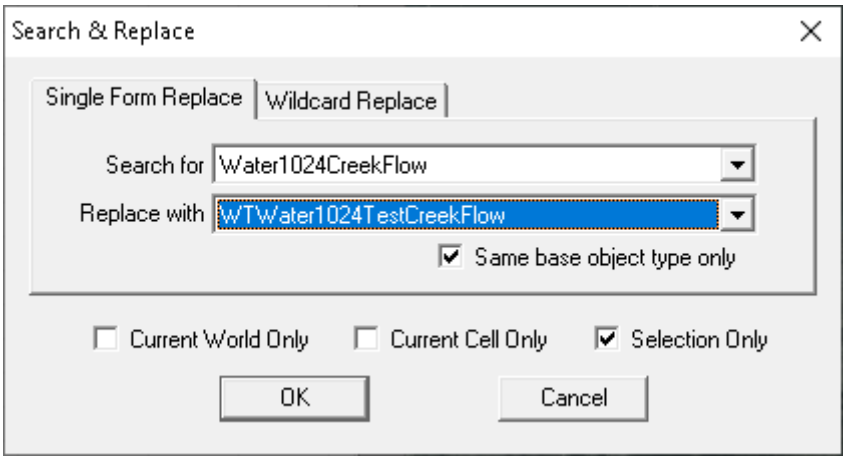


Figure 148 - Changing the water plane.

In the 'Replace with' drop-down, choose the water plane you just created. In my example that's going to be WTWater1024TestCreekFlow.

Click OK.

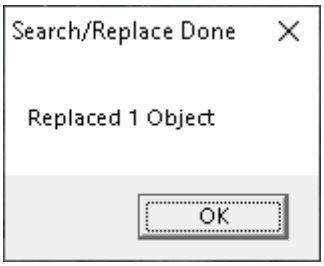


Figure 149 - Search/Replace Done.

Click OK again to the Search/Replace Done pop-up.

Press 'F5' to refresh the render window.

Obviously it's a bit hard to tell from a static screenshot, but the direction of the creek has now reversed and is flowing from left to right.



Figure 150 - Creek is now flowing in the opposite direction.

Now that our water activator is in place, go back to WorldData > WaterType in the Object Window.

We can make further adjustments to the water type properties and use the Apply button to see our changes update in the render window without having to close the properties or use 'F5' to refresh.

For example, if we decrease the Fog Amount in the Fog Properties tab to 0.3000 and click Apply, our creek will now appear a lot clearer in the render window.



Figure 151 - The creek with a reduced Fog Amount.

Alright, so let's say we want our creek to run a bit faster.

Go back to the Noise Properties tab.

For Noise Layer One, set the Wind Speed to 0.0680.

For Noise Layer Two, set the Wind Speed to 0.1090.

And for Noise Layer Three, set the Wind Speed to 0.1090.

Press Apply.

You should see the speed of the creek water update in the render window.

For more information about each field, check out the WaterType article on the Creation Kit wiki.

Now that we've changed the direction our creek is flowing, let's also change the direction of the current.

In this example I'm using a water plane to create the creek, so double-click or right-click on the water plane and select Edit.

Go to the Water Current tab.

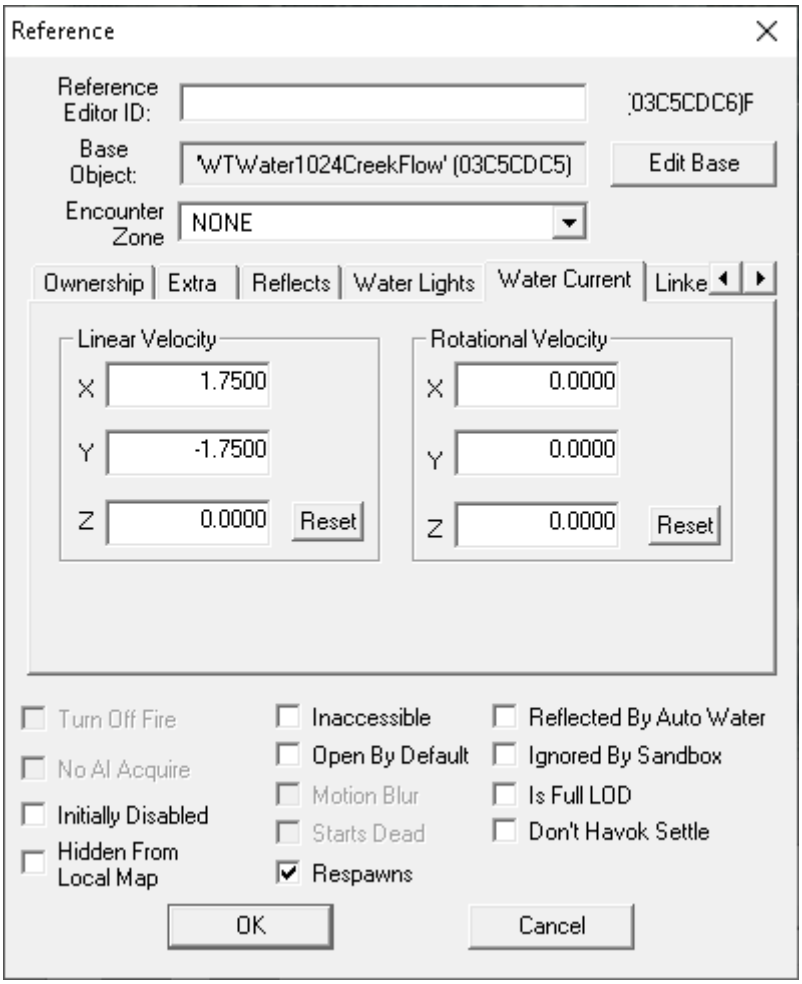


Figure 152 - Setting a water current.

For my example, under Linear Velocity, I'm going to set X to '1.7500' and Y to '-1.7500' to make the water current flow in the same direction as the waves.

The higher the numbers you set under Linear Velocity, the stronger the current.

Generally speaking you shouldn't modify the Z value under Linear Velocity.

Determining what values to set might take some trial and error, but you can use the move gizmo as a guide. With the water plane selected, press 'E' to bring it up.

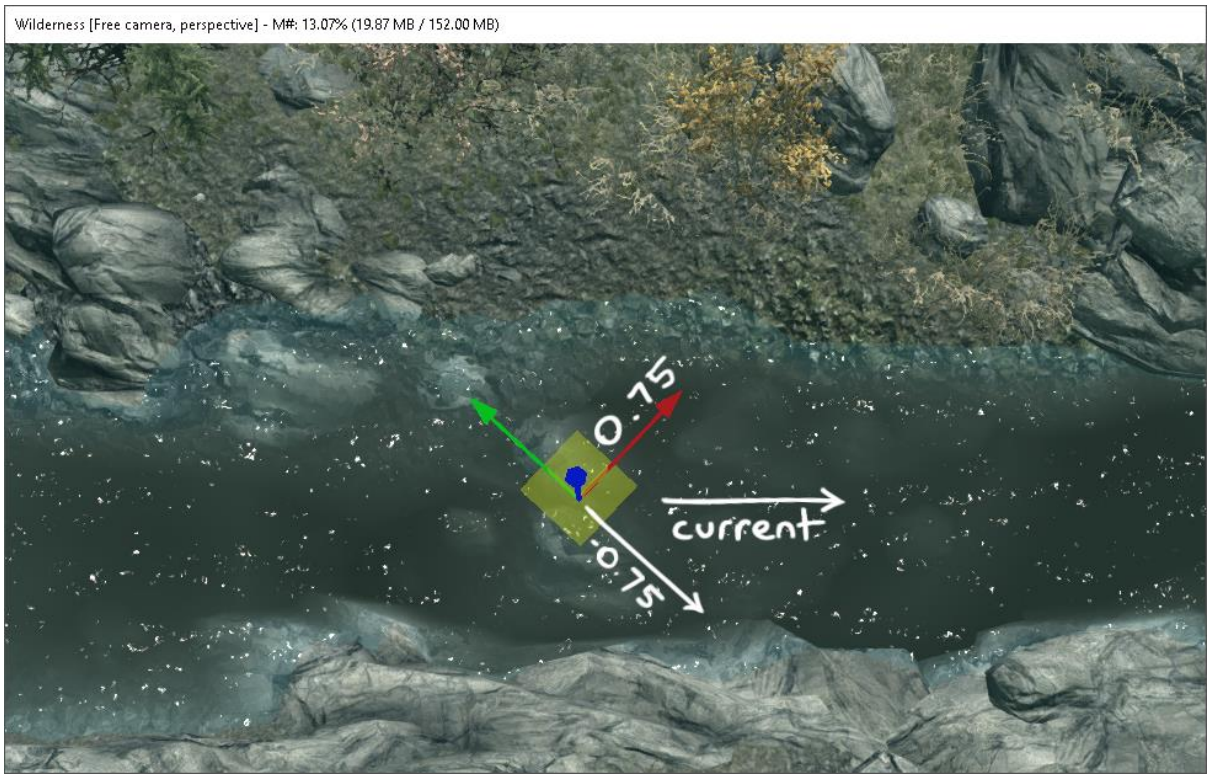


Figure 153 - World coordinates on the water plane.

So basically I want the water current to move from left to right, same as the water flow.

To do this I set X to 1.7500 because it's pointing towards the direction I want the water to go, and Y to -1.7500 because it's pointing away from the direction I want the water to go.

These two values are combined to make the current travel in the right direction.

Finally, I tested the water current out in-game and confirmed I was floating down the creek in the right direction.



Figure 154 - Floating down the creek.

If you wanted to change the current of the cell water instead, right-click on the current cell in Cell View and select Edit.

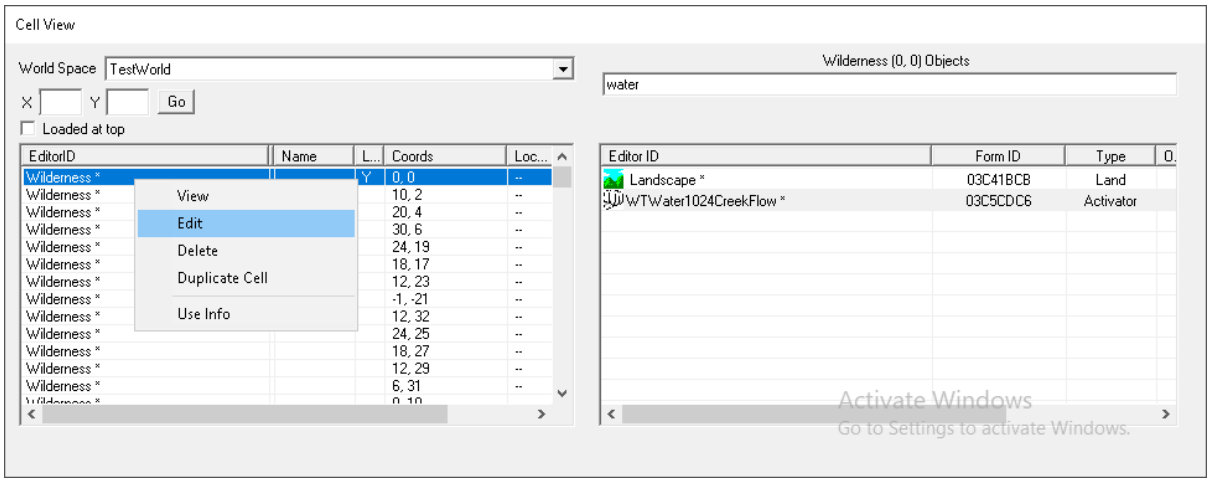


Figure 155 - Editing a cell's properties.

Like the Water Current tab in a water plane reference's properties, you can set the direction and strength of the water current under Linear Velocity.

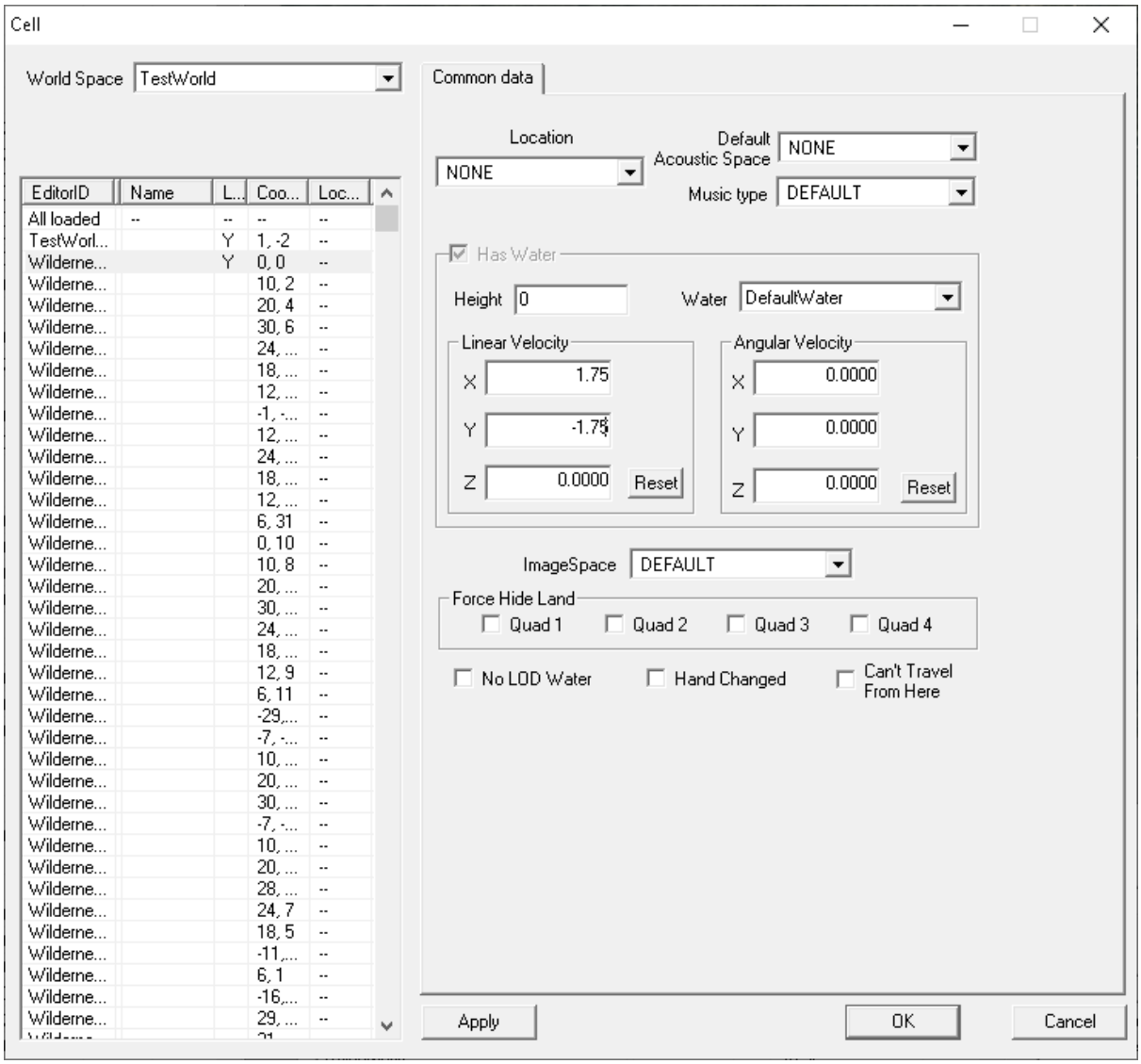


Figure 156 - Setting a current on the cell water.

Then click OK to close the cell's properties.

ADDING REGIONS

Regions are a collection of world space cells. They can be used to create separate biomes with different weather types and ambient sounds, and so on.

For example, in Wyrmstooth I created 'WyrmstoothRegion' to act as a border region to prevent players from swimming out too far into the ocean. This is the region with the red border in the screenshot below. Within 'WyrmstoothRegion' I also defined 'WyrmstoothForest', 'WyrmstoothMarsh' and 'WyrmstoothSteampools' to cover the different biomes in my world space.

Regions can also be used to automatically populate parts of your world space with static objects.

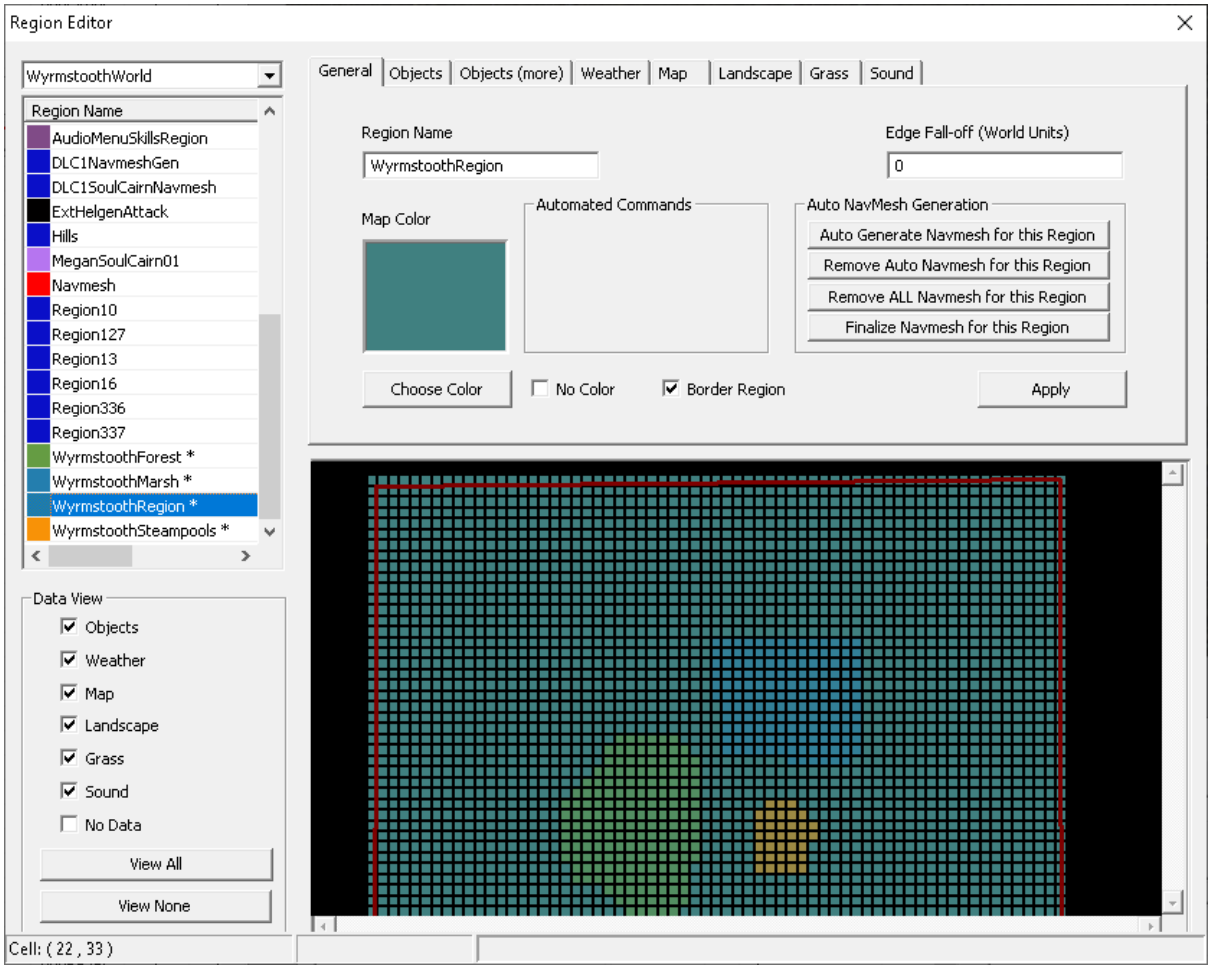


Figure 157 - An example of a world space with multiple regions.

Important: Every cell in your world space should be in a region.

The size of your border region should be defined in multiples of 8. For example, -32,32 to 32,-32, or -64,64 to 64,-64 depending on how large you plan to make your world space.

Go to World > Regions.

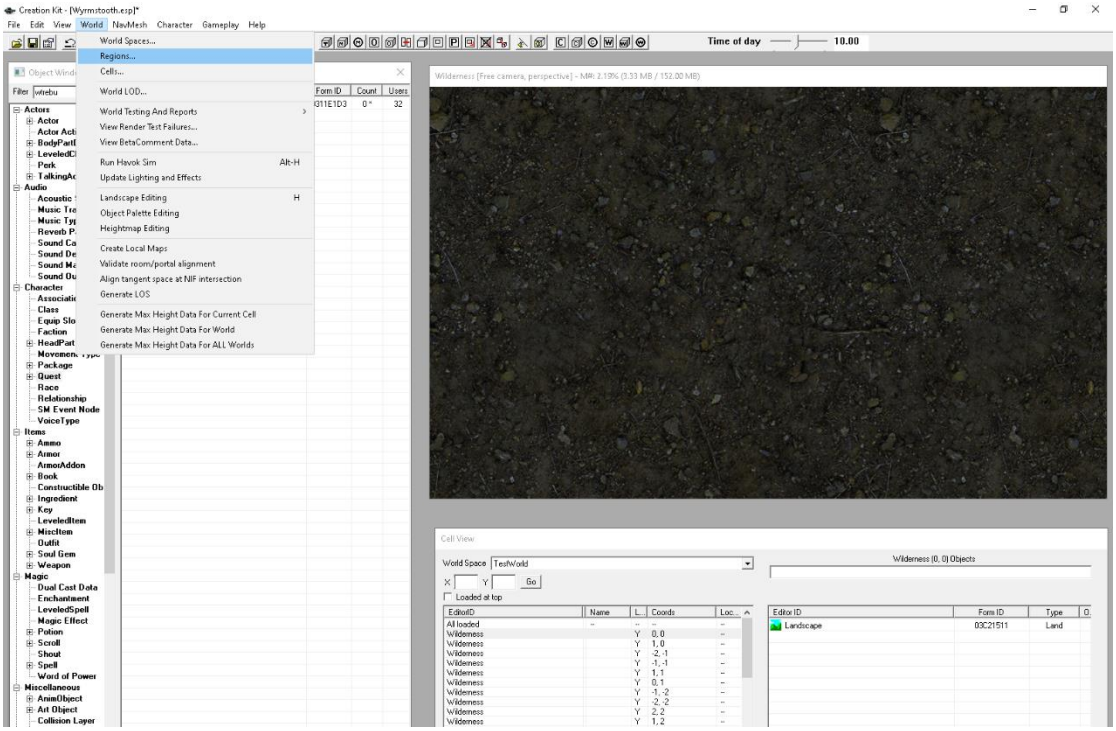


Figure 158 - Regions in the World menu.

In the drop-down at the top-left of the Region Editor, select the world space you want to create a new region for.

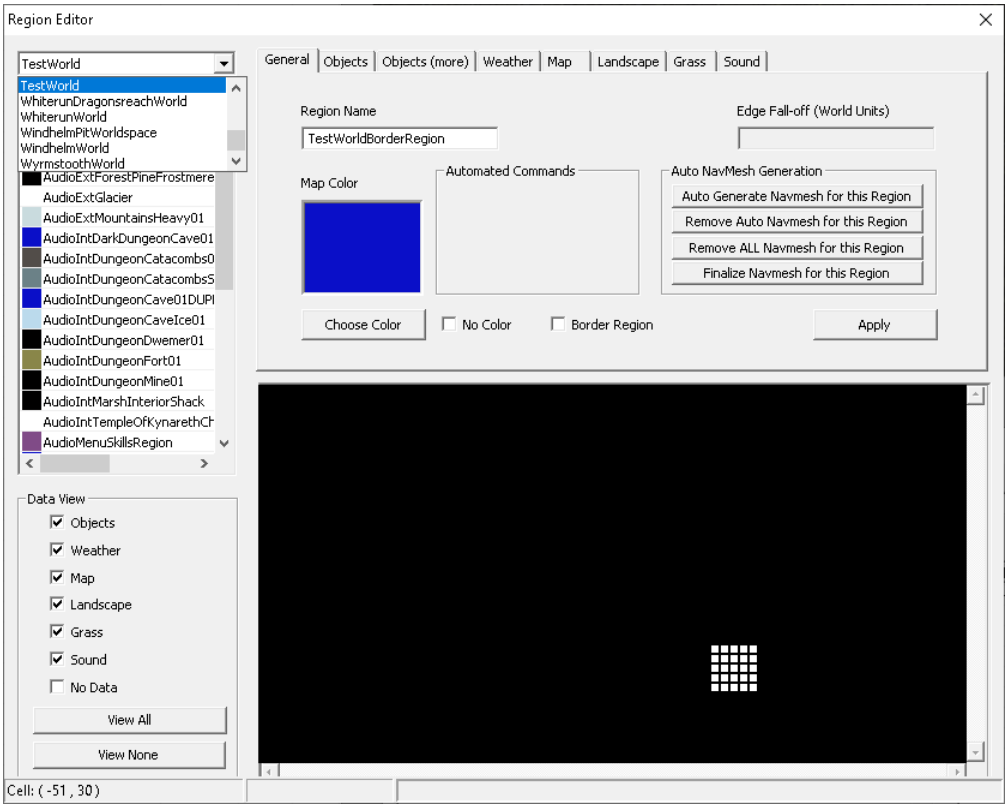


Figure 159 - Selecting the world space in the Region Editor.

Right-click on one of the regions listed and select New Region.

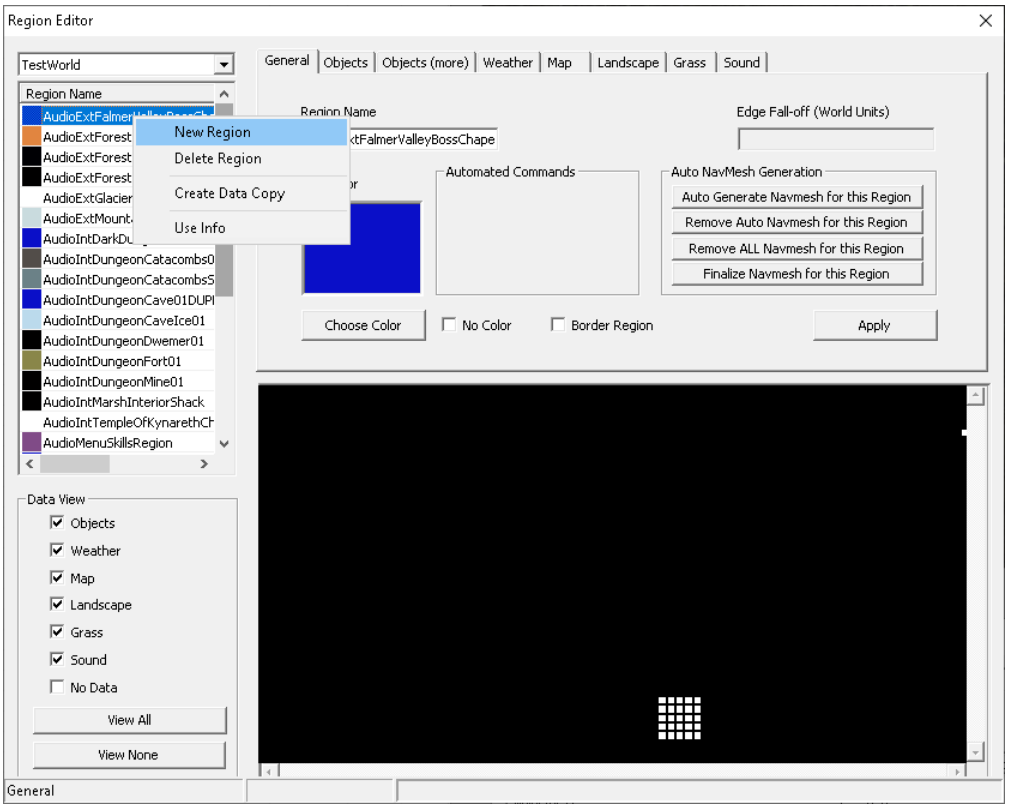


Figure 160 - Creating a new region in the Region Editor.

Under the General tab, enter a Region Name.

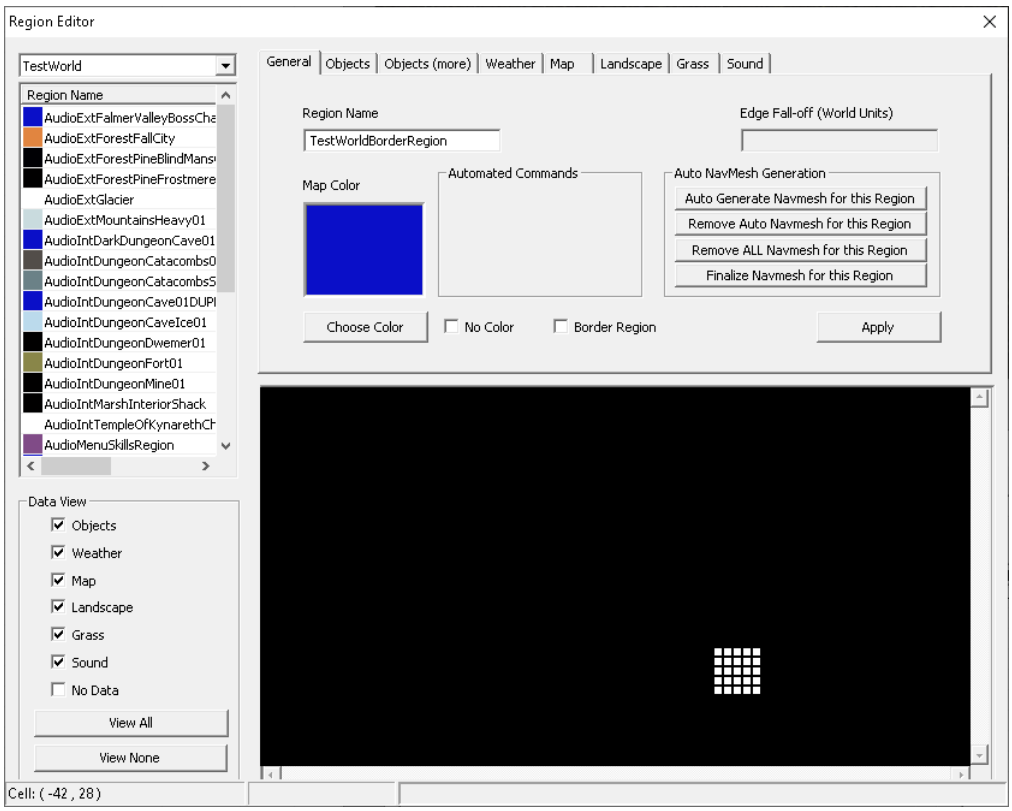


Figure 161 - Naming the new region.

Below the cell properties is the cell view. The white squares represent existing cells the Creation Kit has already created automatically when you loaded your world space in the render window. The black squares represent sections of your world space for which no cell has yet been generated.

We'll need to create a world space sized in a multiple of 8 for the sake of later on generating our land LOD, so let's start by defining a border region.

You can pan around the cell view by holding down the Spacebar while moving your mouse around. The cell you're currently hovered over will be shown at the bottom left.

In the black area, find cell -32,32 and left click. You should see a small 'x' appear.

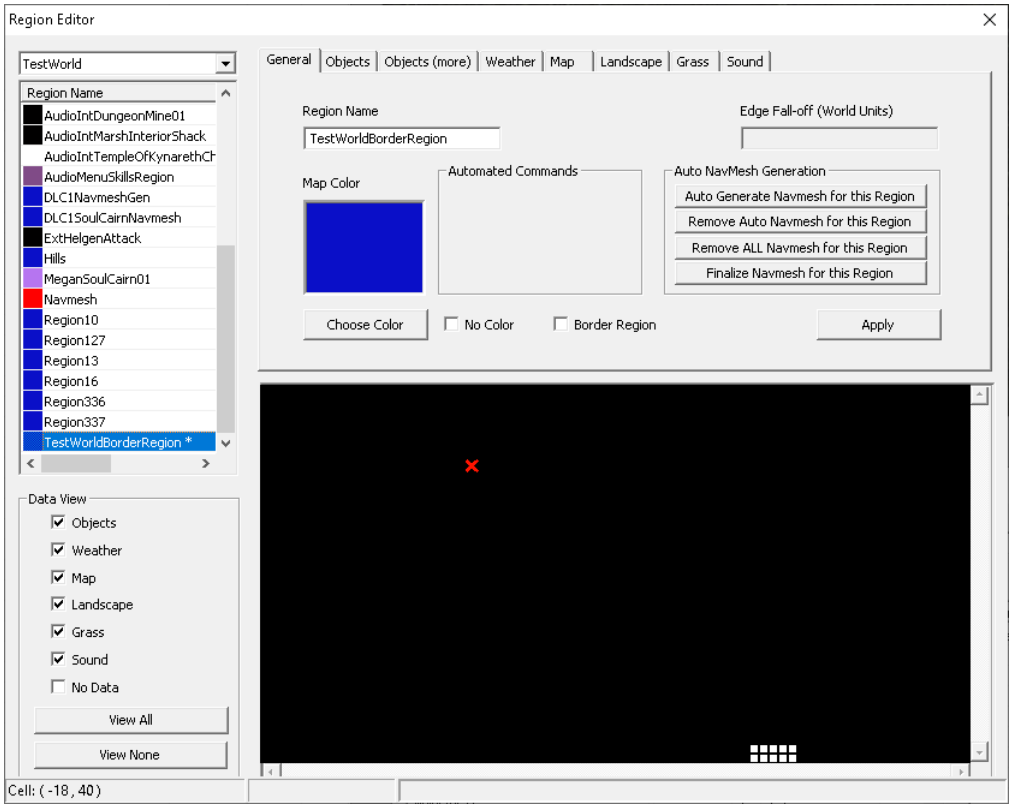


Figure 162 - Marking cell -32,32.

Next, find cell 32,32 and left-click on it to place another ‘x’. A line should be drawn between the two.

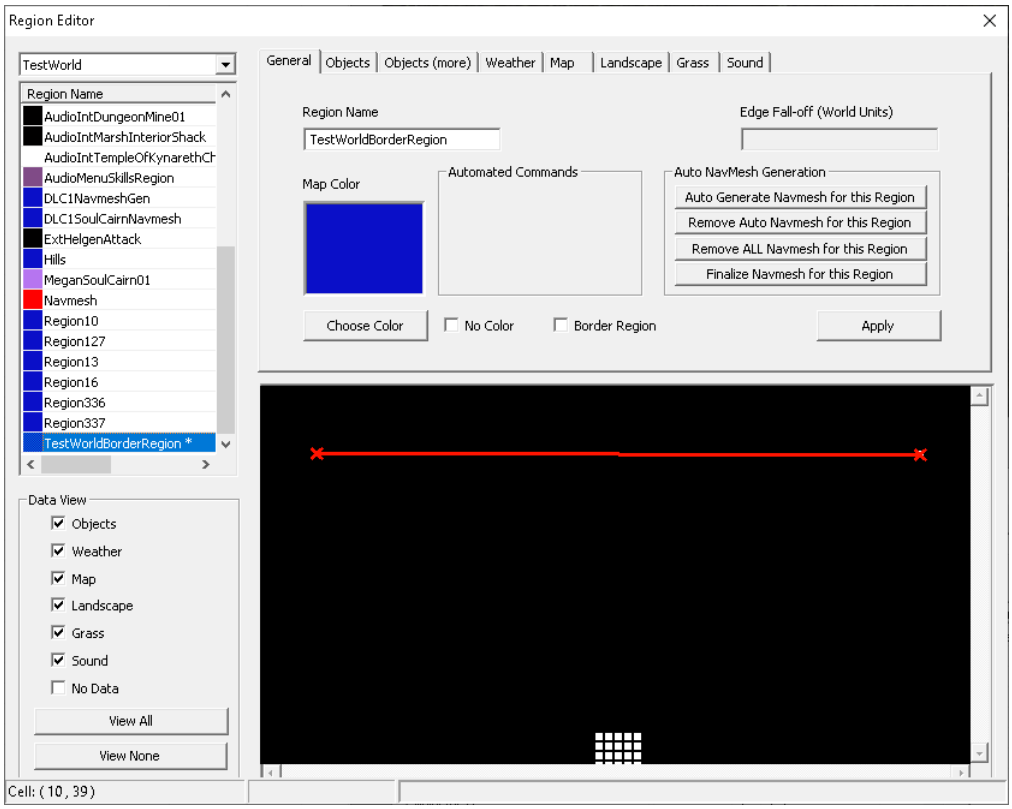


Figure 163 - Continuing the region border.

Continue down and left-click on 32,-32 then to the left and left-click on -32,-32. Finally, to close our square, left-click once again on -32,32.

Click Apply. To refresh the view, click on another region in the list then go back to your region. The cell view should now show new cells generated in-between the points we defined.

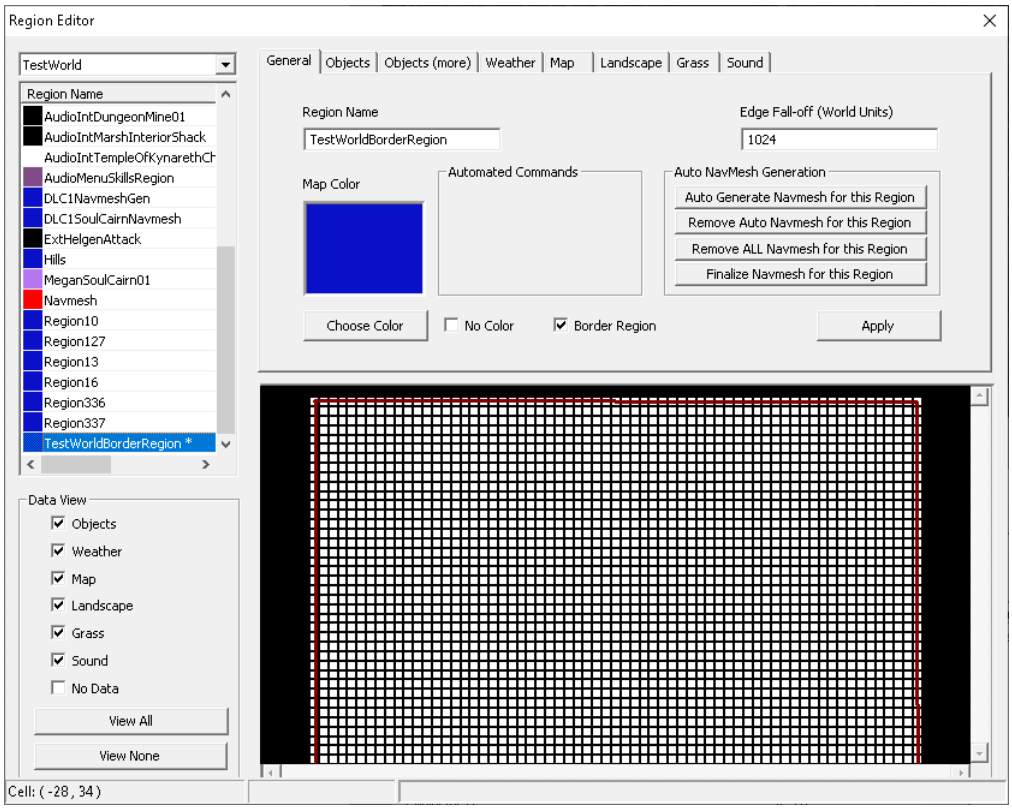


Figure 164 - New cells created within the defined border.

Click on the Border Region tickbox to mark this region as a border region.

Click Choose Colour and give your region a colour to differentiate it from other regions we'll be adding.

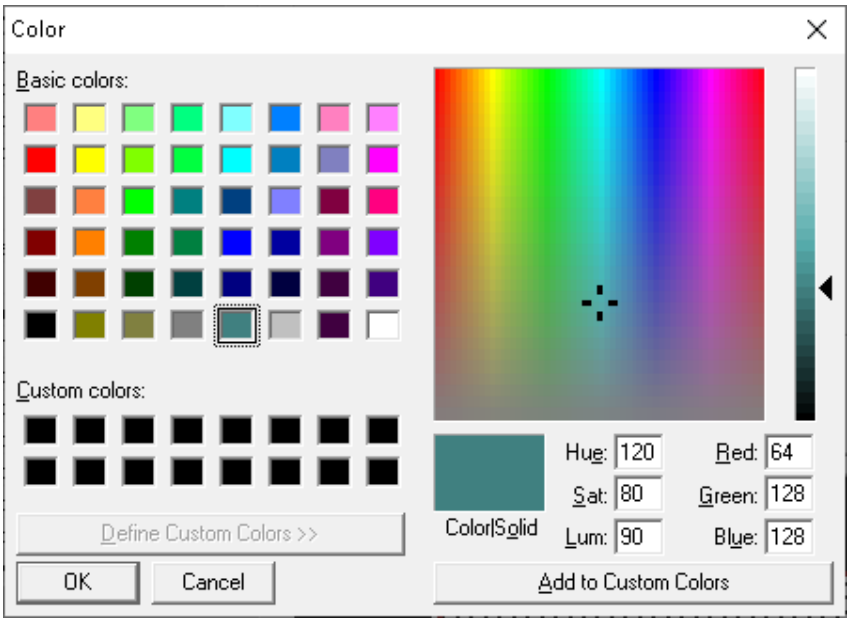


Figure 165 - Region colour.

Go to the Weather tab and tick 'Enable this type of data'.

If your region is nested within another region, tick 'Override'. Since we're working on a border region it isn't necessary.

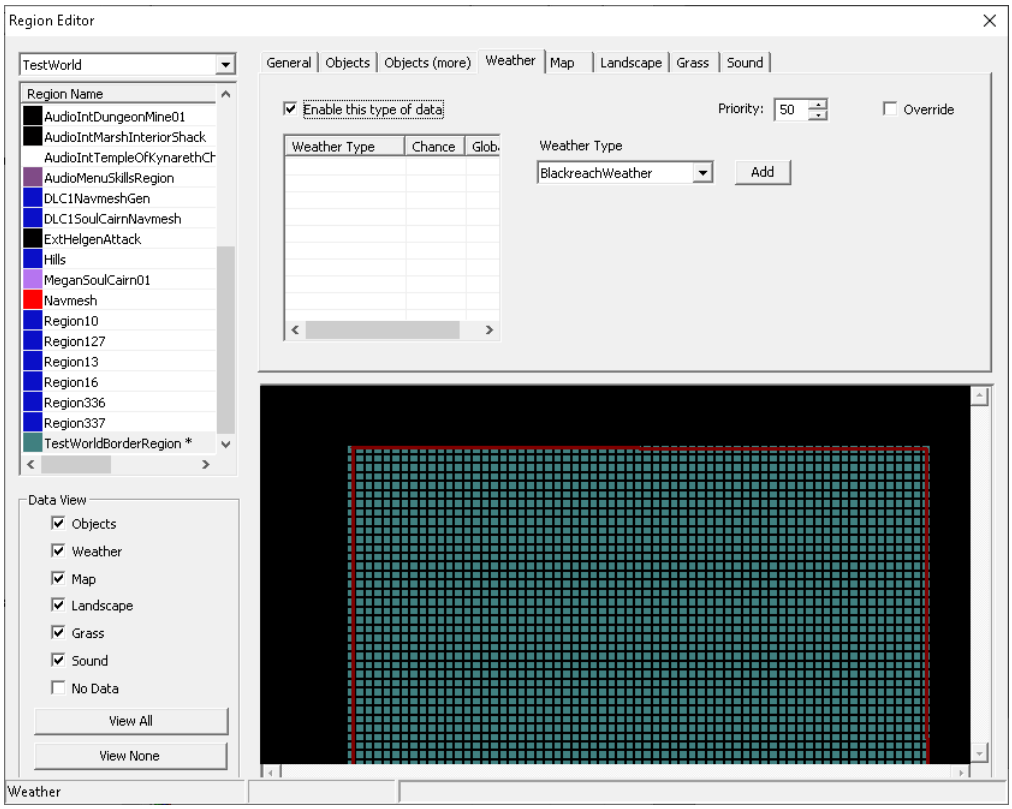


Figure 166 - Enabling region-specific weather.

To start adding weather types, choose the Weather Type you want to add in the drop-down menu and click Add. For this example I'm adding SkyrimCloudy.

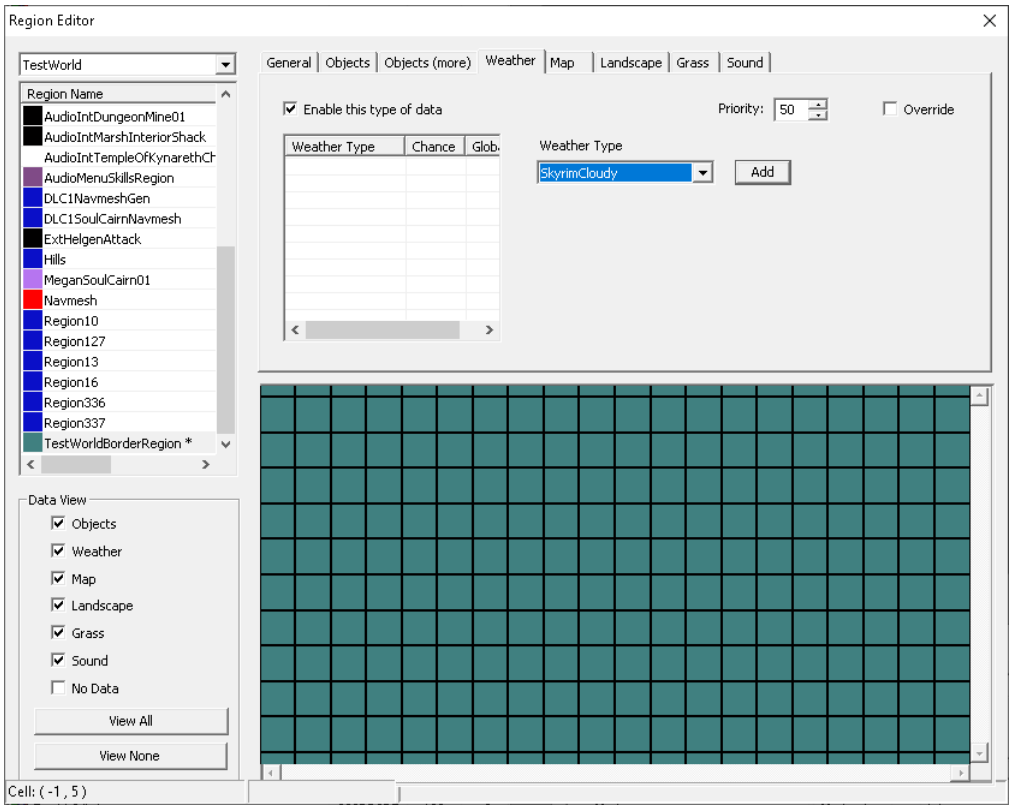


Figure 167 - SkyrimCloudy weather type selected.

When you add a weather type, the Select Form window pops up. We can just close it, no need to select anything.

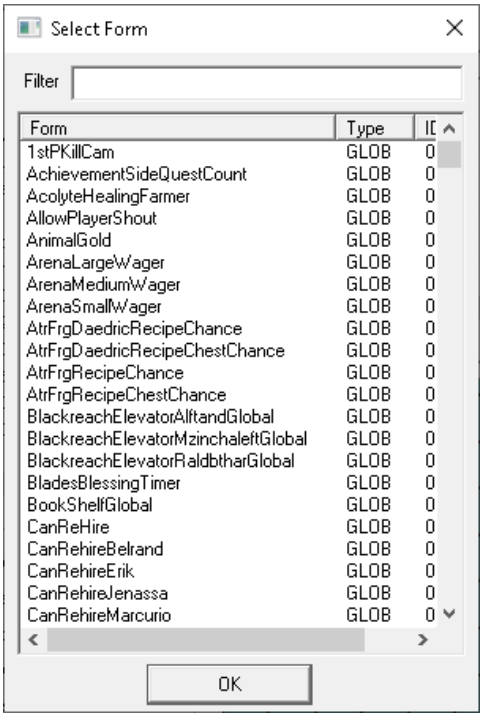


Figure 168 - Just ignore the Select Form window.

Double-click on the weather type in the list to modify its chance in the Chance column.

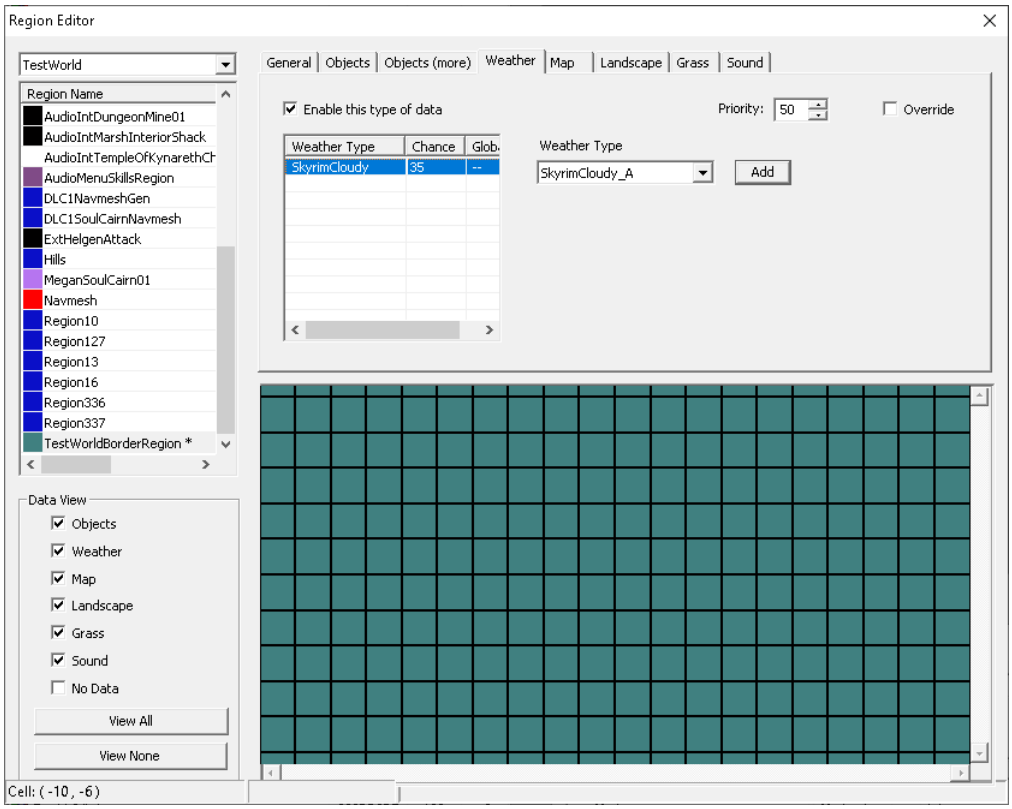


Figure 169 - Modified the chance of a specific weather's regional occurrence.

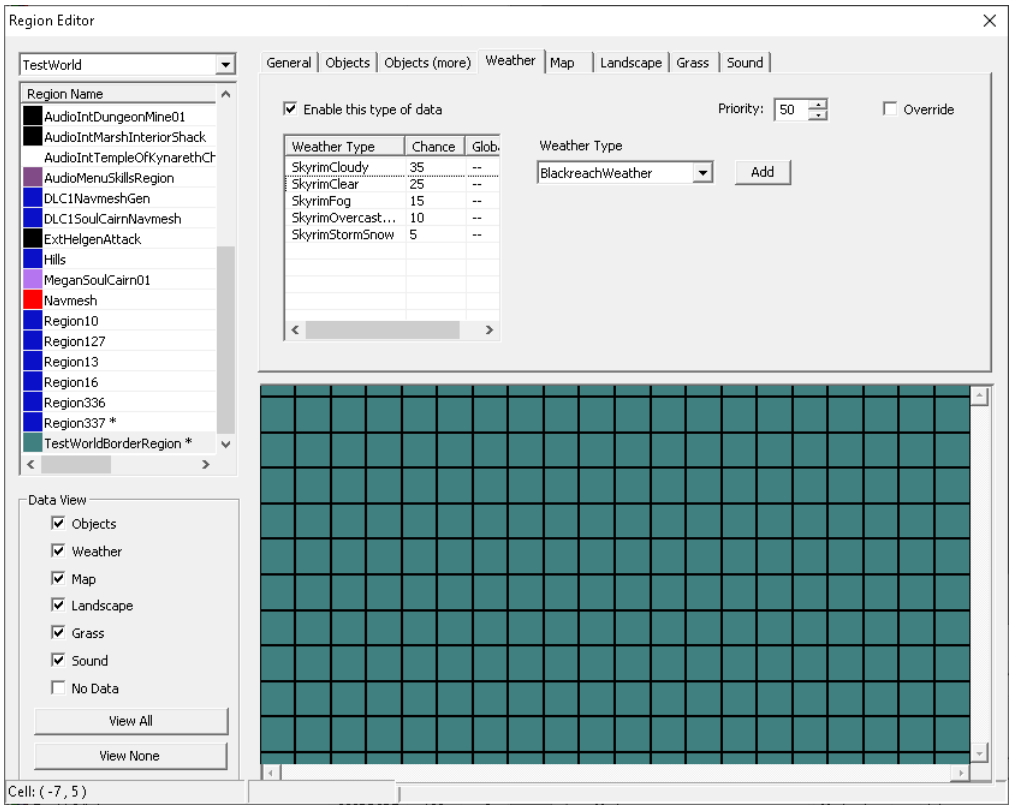


Figure 170 - Added more weather types.

Go to the Sound tab and tick 'Enable this type of data'.

Note: If your region is nested within another region that already has sound data enabled, the 'Enable this type of data' tickbox should already be ticked but will appear greyed out as per the screenshot below.

To add new sounds, in the Object Window navigate to Audio > Sound Descriptor and search for the sound you want to add, such as AMBrWindForestPineGustB01, then drag-and-drop it into the sound list in the Region Editor.

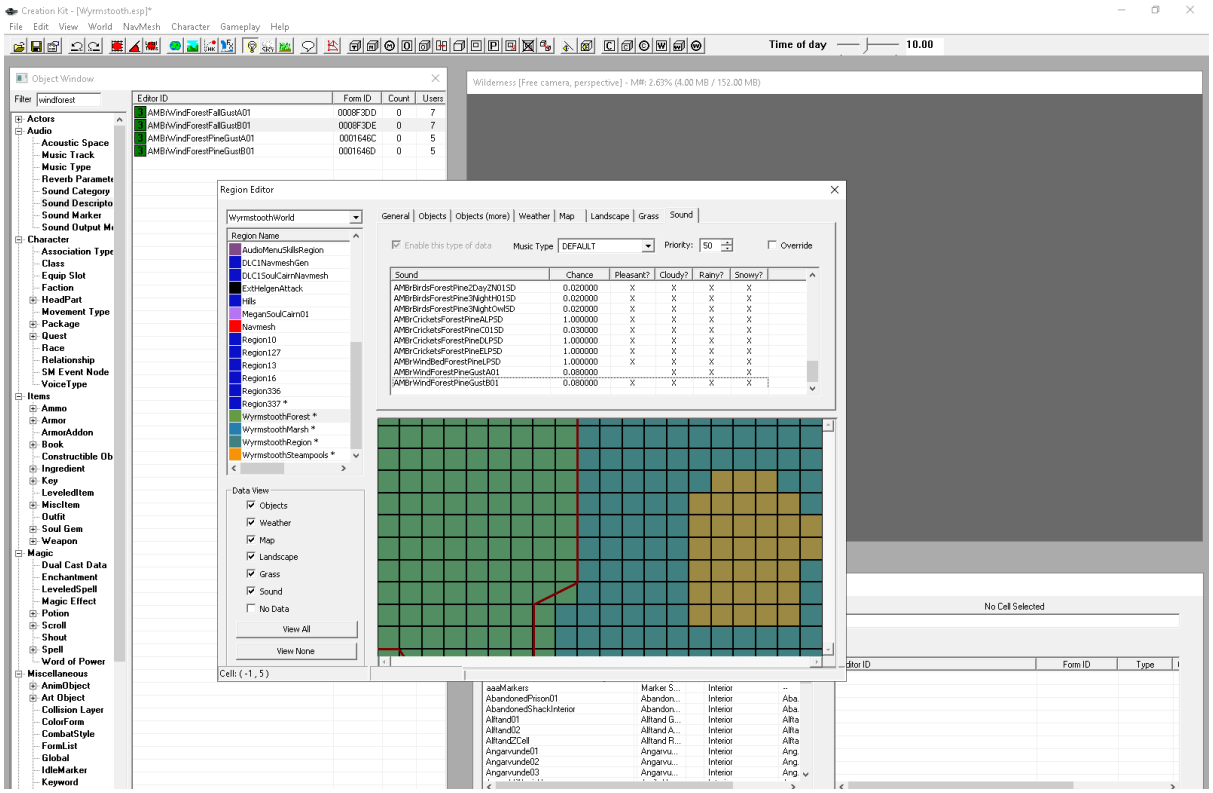


Figure 171 - Adding regional ambient sounds.

The screenshot above shows the list of sounds I added to the WyrmstoothForest region.

You can modify the chance of hearing a sound by double-clicking on its chance value to bring up the following form:

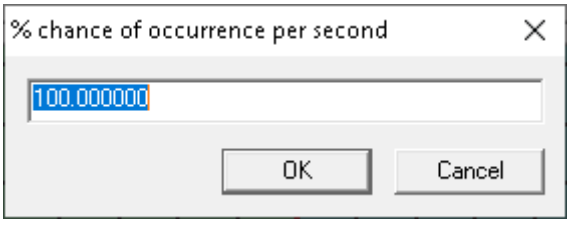


Figure 172 - Modifying the chance of hearing a sound.

Similarly, you can mute certain sounds during specific types of weathers by double-clicking on the 'X' in the corresponding column to make it disappear. For example, in the screenshot above I had muted the AMBrWindForestPineGustA01 sound during pleasant weather. Otherwise, during other types of weather, it has an 8% chance of being played each second.

Next, go to the Map tab and tick 'Enable this type of data'.

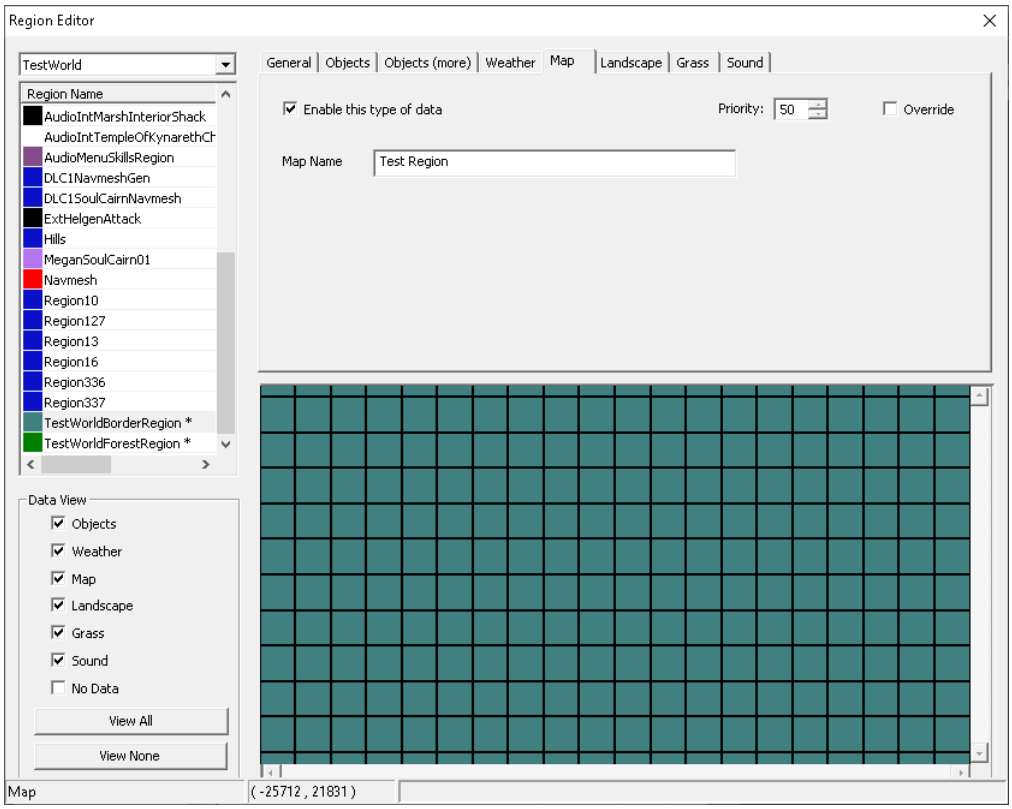


Figure 173 - Region map name for save files.

For the Map Name field, enter in the name of the region. The Map Name defined here will be shown when loading a save in the load save menu if you saved in a wilderness cell.

Let's create another new region. For this example I just called it TestWorldForestRegion.

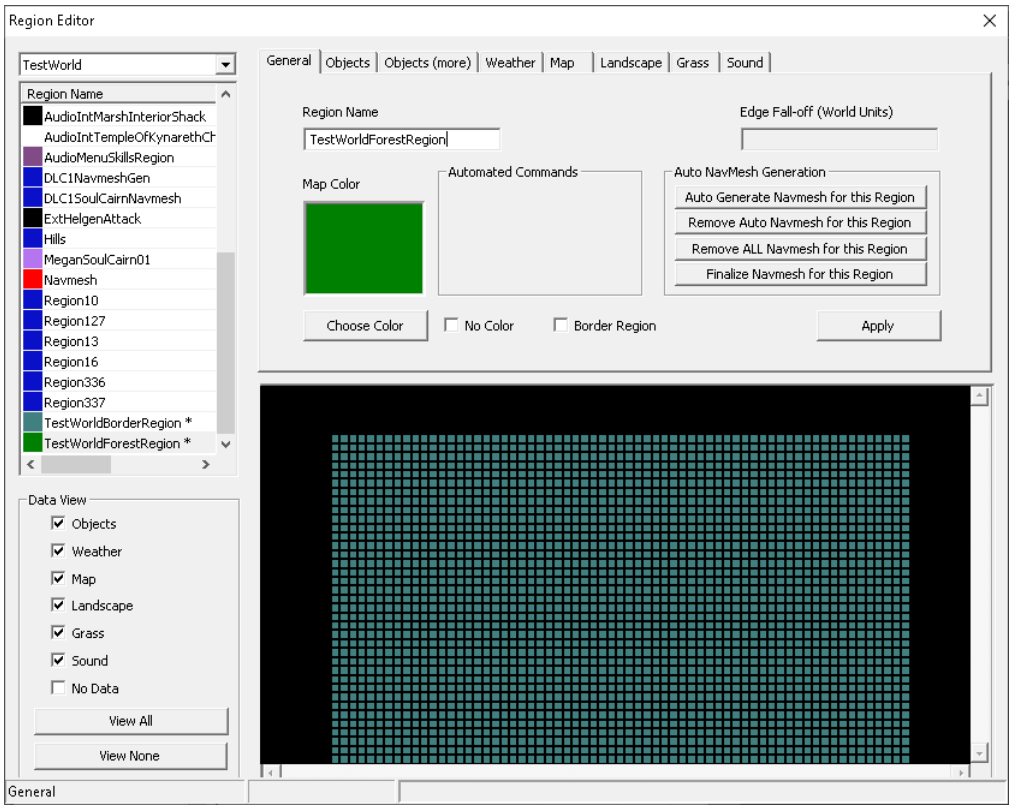


Figure 174 - Added a second region.

Draw the region borders for the new region in the cell view. Make sure it's a closed shape.

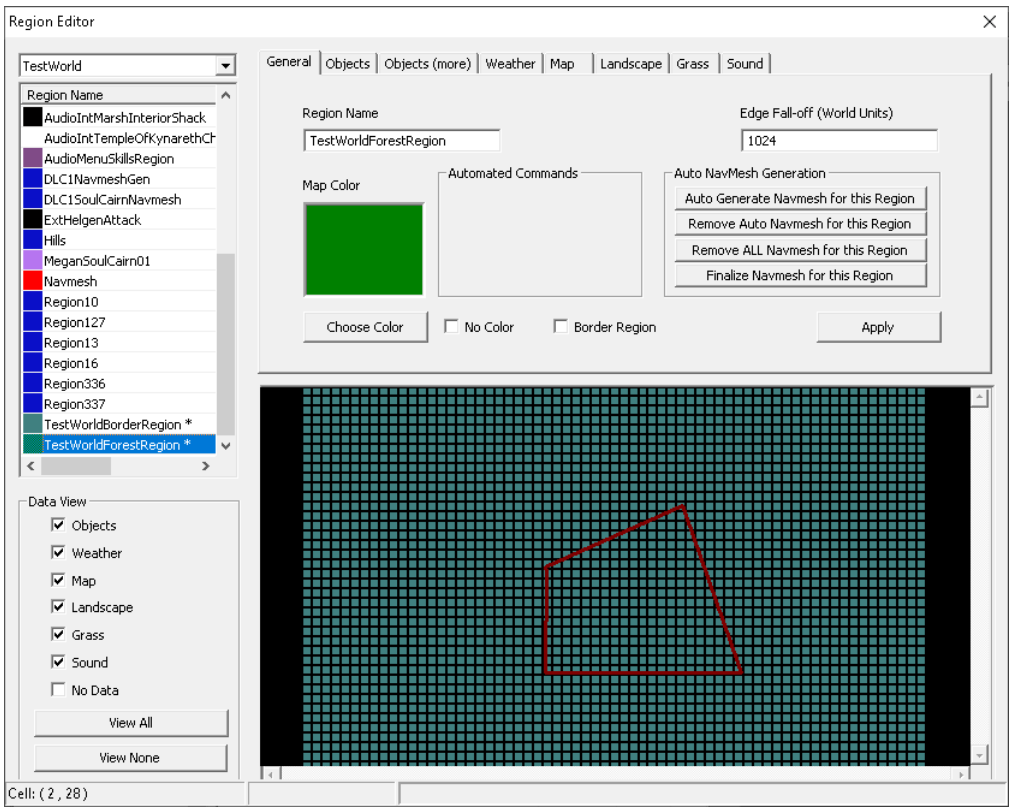


Figure 175 - Drew the border of the second region.

Go to the Weather tab. Add the weather types you want this region to experience and enter in their % chances.

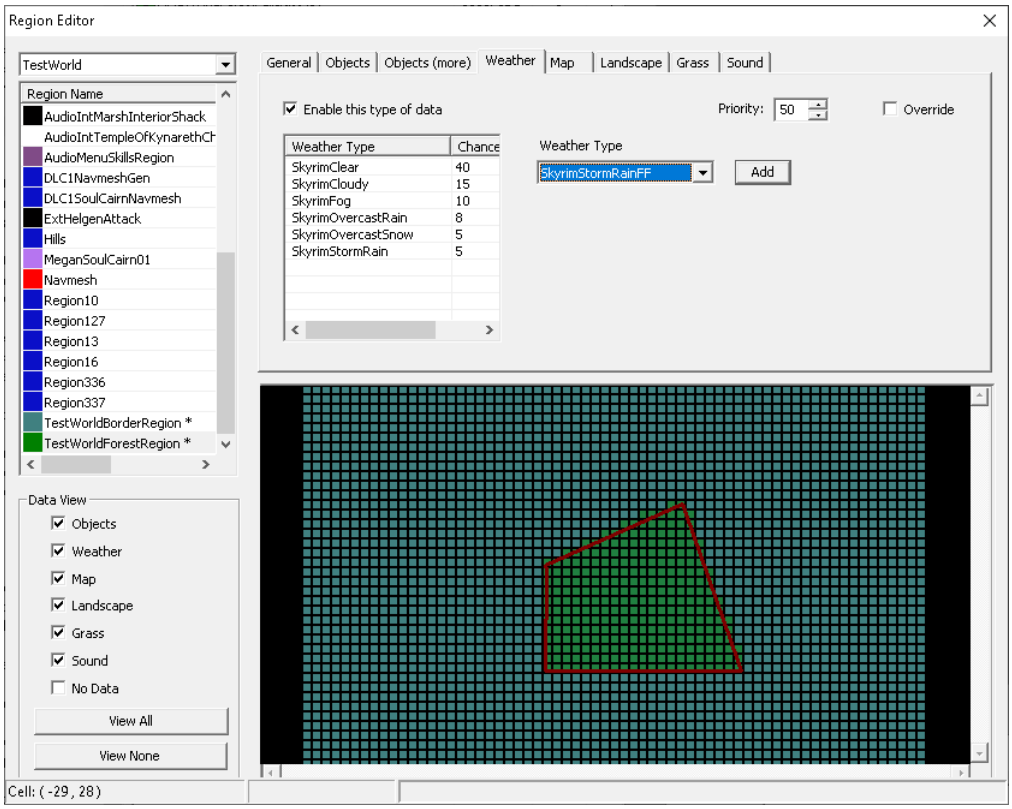


Figure 176 - Cells assigned to the new region.

The cells within the border of the new region should change colour.

You can add extra points to the shape of the region border by holding down CTRL and left-click dragging on the red border to add or subtract cells.

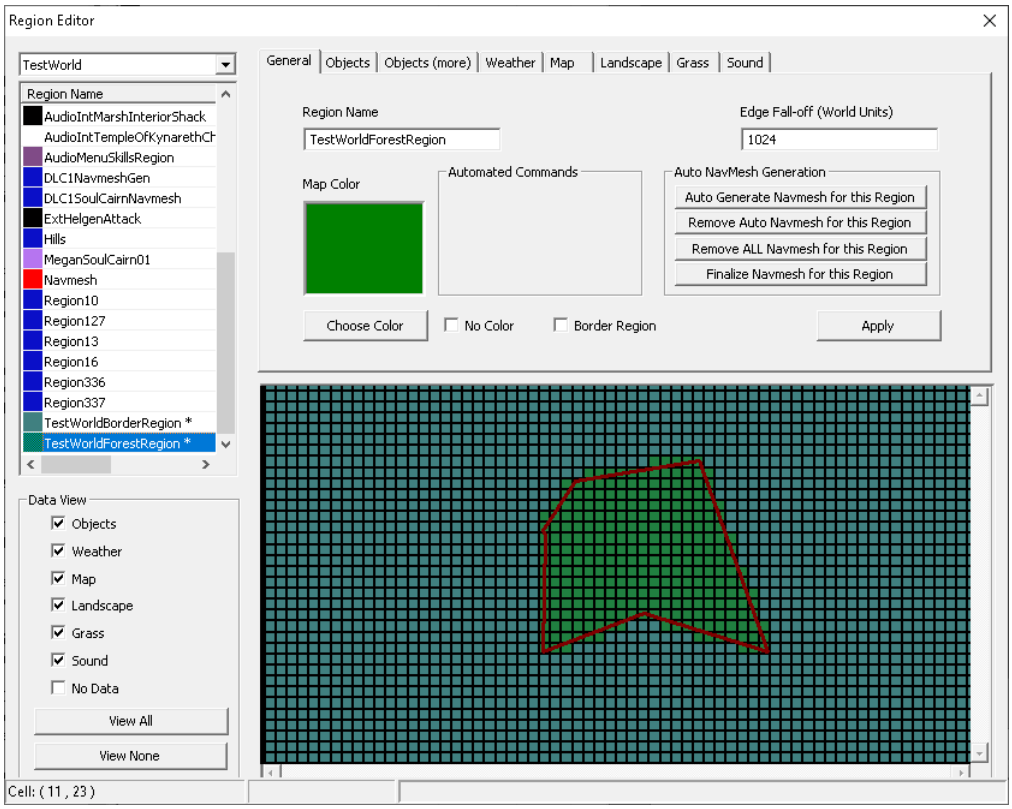


Figure 177 - Region border redrawn.

REGION GENERATED TEXTURES AND OBJECTS

Automatic texture painting and object generation is a fairly complicated topic. In this section I'm going to walk you through a few simple examples to give you enough of an idea how all of this works

The Objects and Objects (more) tabs can be used to automatically texture and/or clutter a region with objects such as trees, bushes and rocks based on conditions like slope and height. Objects and textures can be parented together to quickly form islands of detail, such as clustering small rocks around large boulders, or clustering thickets and bushes around trees.

Note: To be perfectly honest I would recommend texturing and cluttering your world space manually. Despite how tedious that sounds, you're most likely going to need to rework the positions of most of these automatically generated objects and textures. Still, you might find these steps useful as a place to start off from.

Let's have a look at auto-texturing our landscape first.

For this first example I created a small region (-4,-4 to 4, 4) and named it TestWorldAutoTexture.

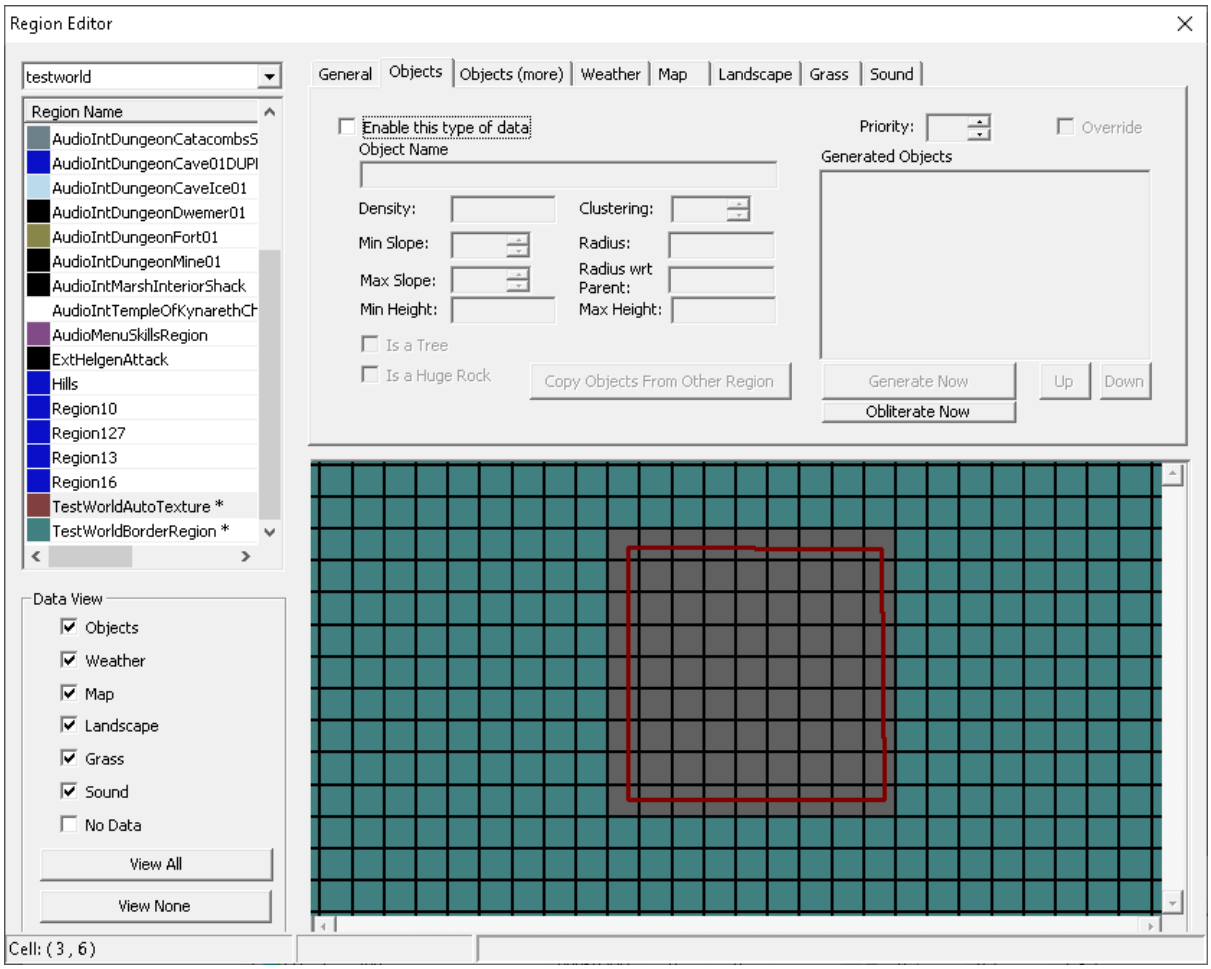


Figure 178 - TestWorldAutoTexture region.

For reference, here's what our landscape looks like without any textures applied.

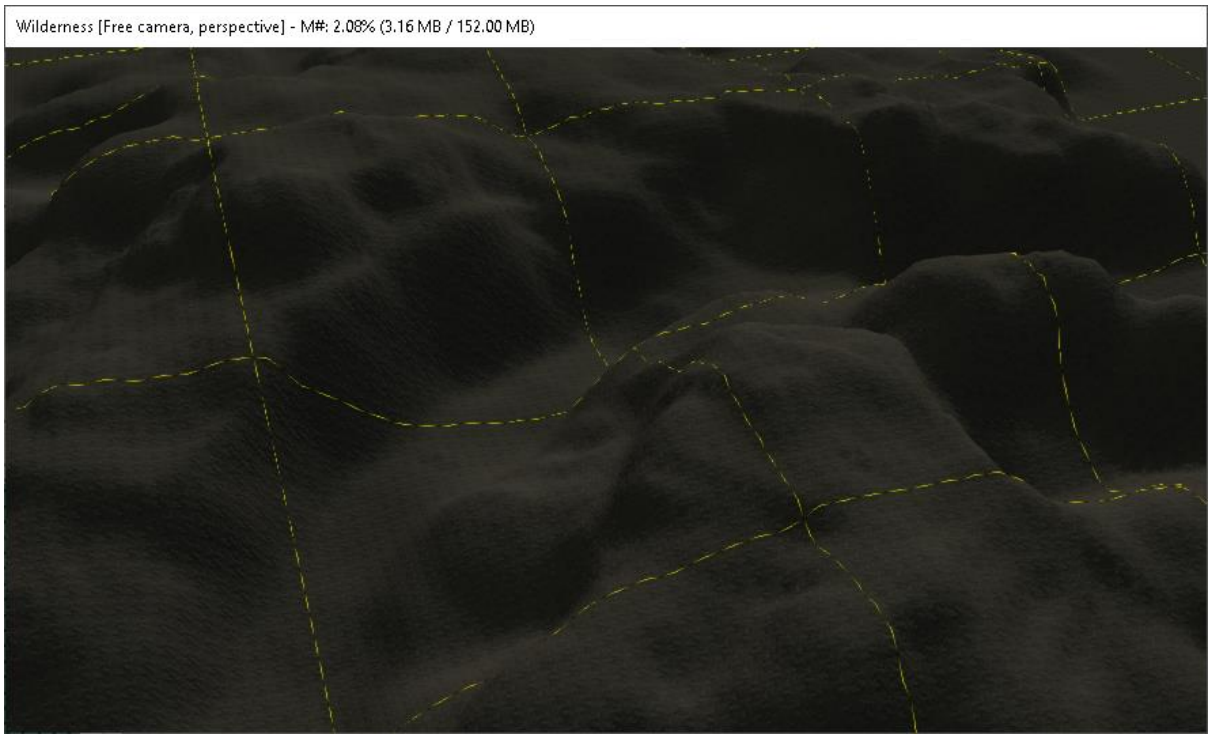


Figure 179 - Sample landscape without textures.

Go to the Objects tab and tick 'Enable this type of data'.

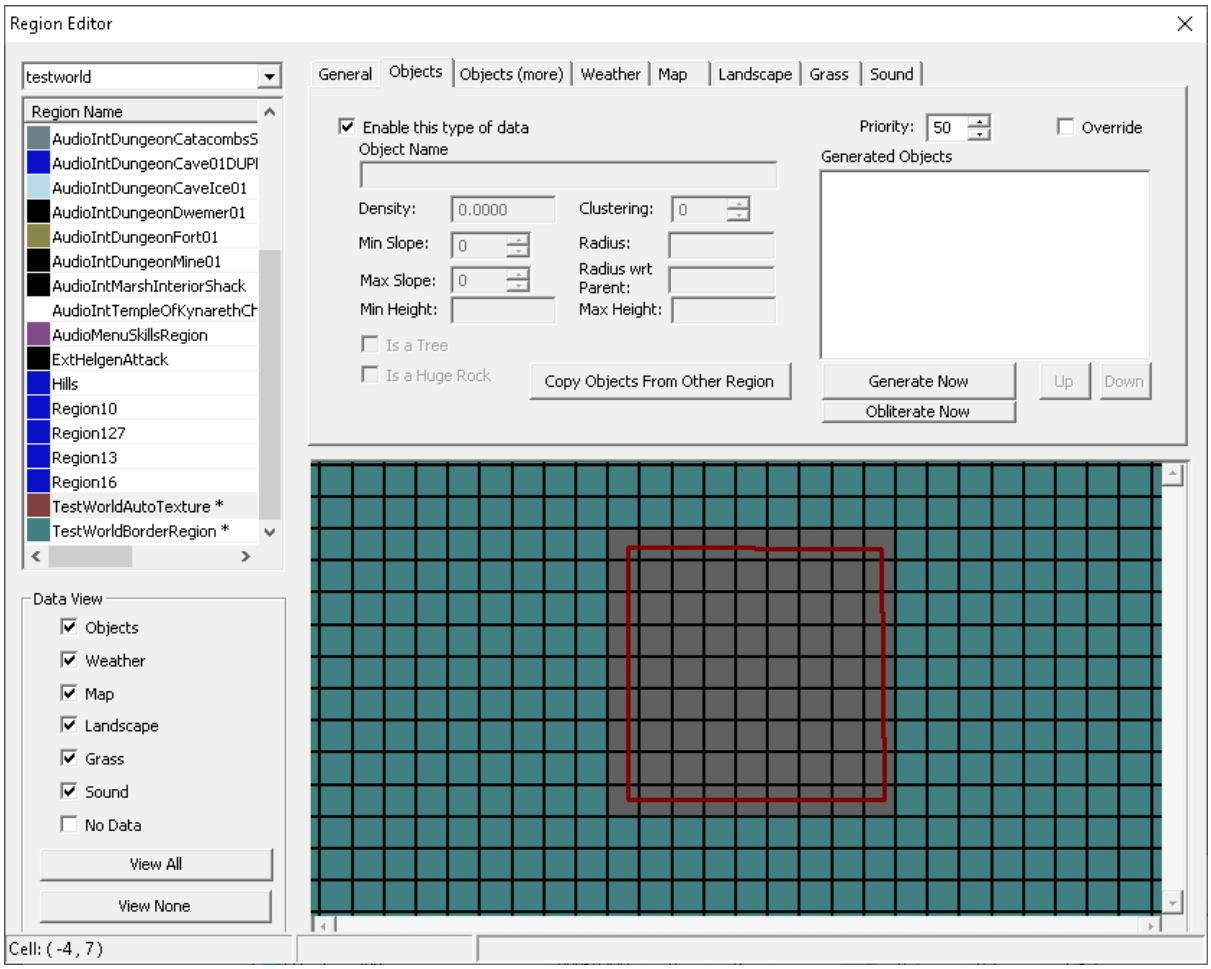


Figure 180 - Enabling object generation.

We'll get into automatic object population shortly, but for now let's concentrate on adding textures first.

In the Object Window go to Miscellaneous > LandTexture and select a texture to add to our region.

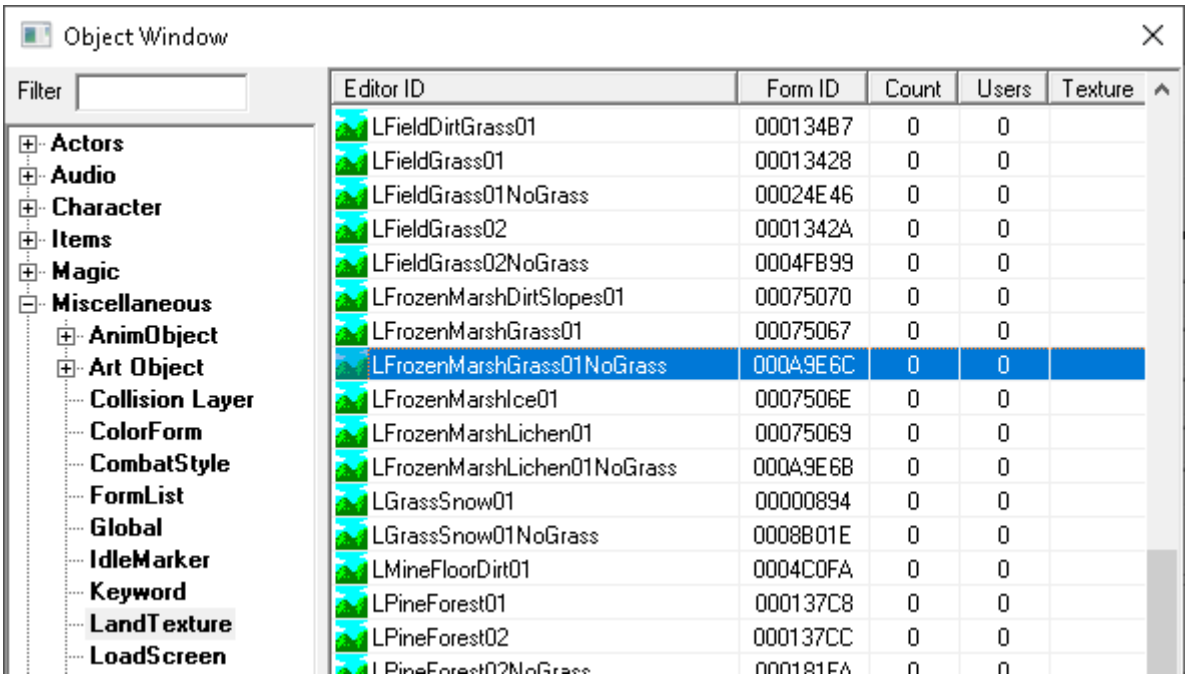


Figure 181 - Selecting a texture to automatically paint on our landscape.

For this example I'm going to add LFrozenMarshGrass01NoGrass.

Drag and drop the landscape texture into the Generated Objects list in the Region Editor.

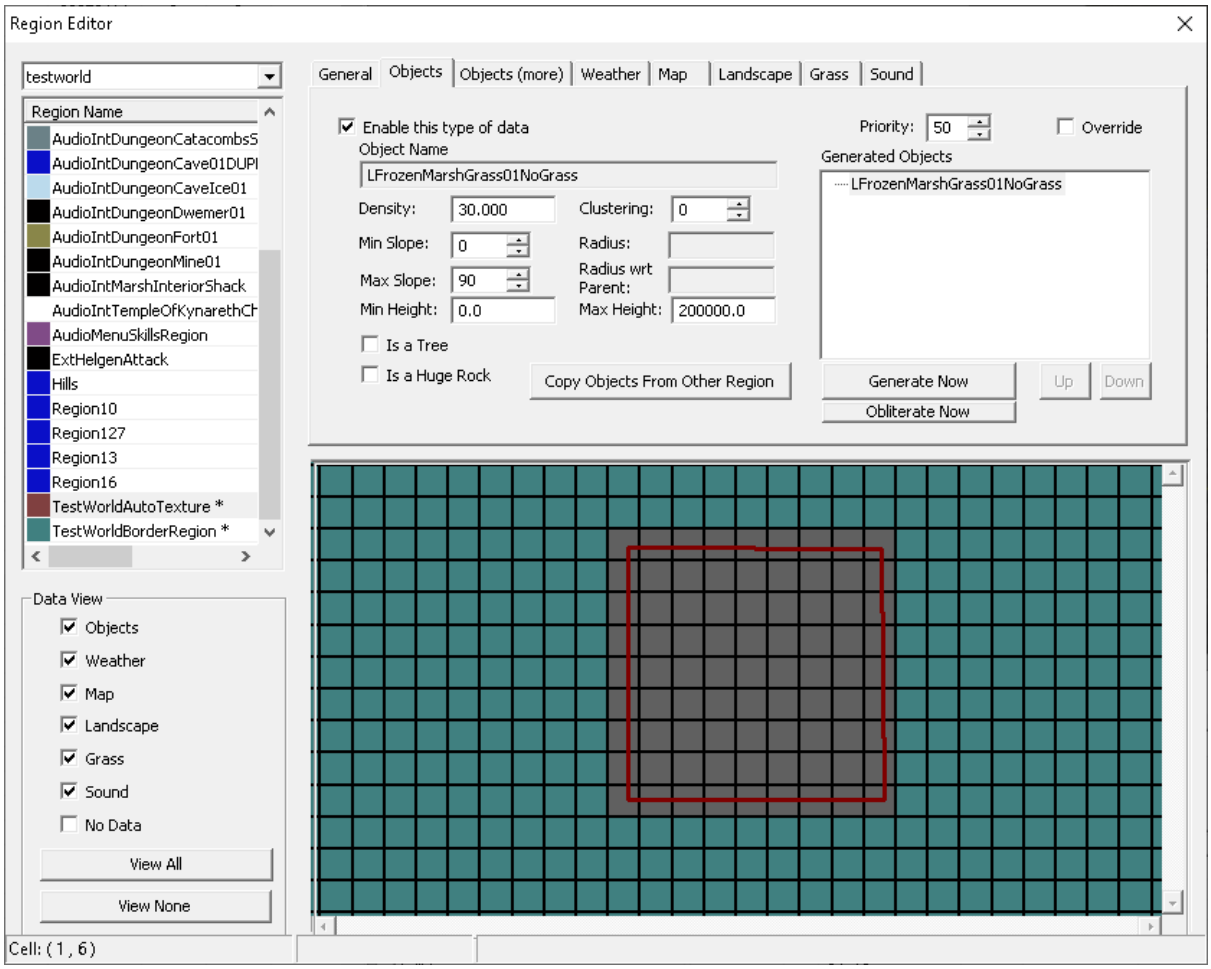


Figure 182 - Landscape texture added to Generated Objects list.

Set the Density to 80, the Max Slope to 20 and the Min Height to -200000.

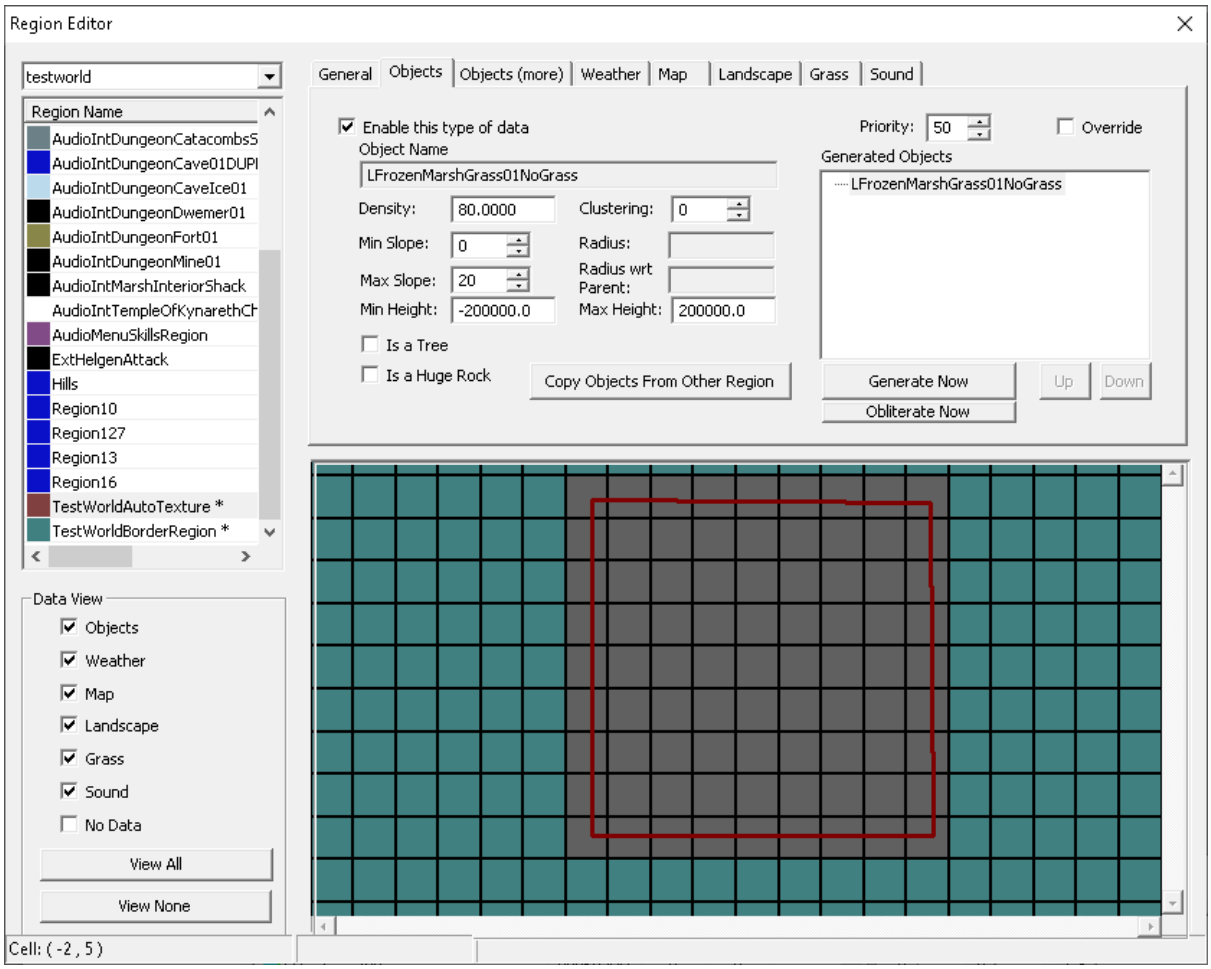


Figure 183 - Setting the generation parameters for LFrozenMarshGrass01NoGrass.

To see what this looks like so far, click on Generate Now.

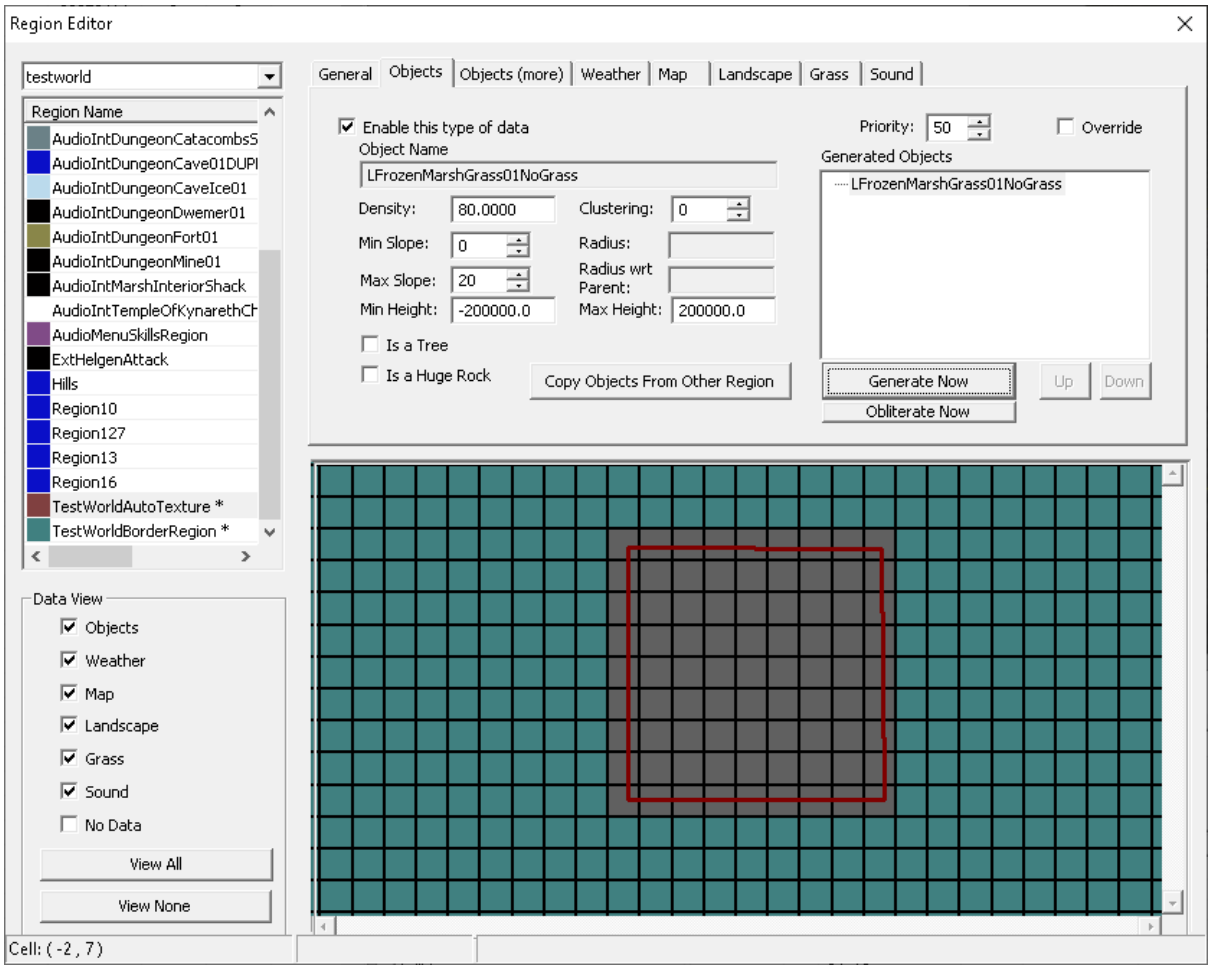


Figure 184 - Generate Now.

Click Yes to the confirmation pop-up.

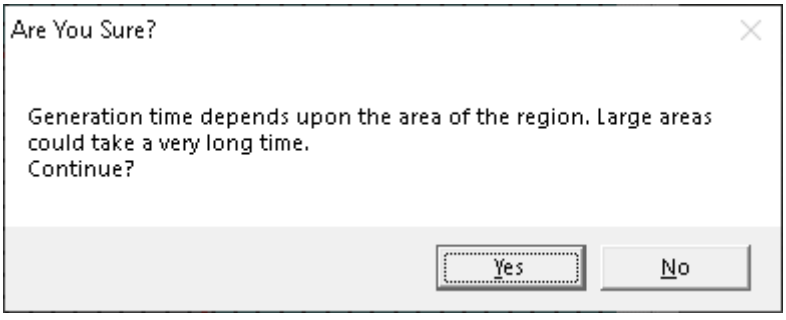


Figure 185 - Are You Sure? popup.

Click OK to the Generation Complete pop-up to dismiss it.

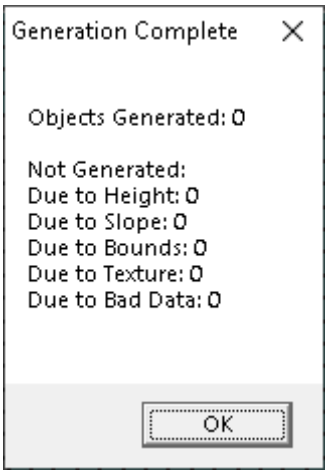


Figure 186 - Generation Complete pop-up.

Here's what our landscape looks like now:

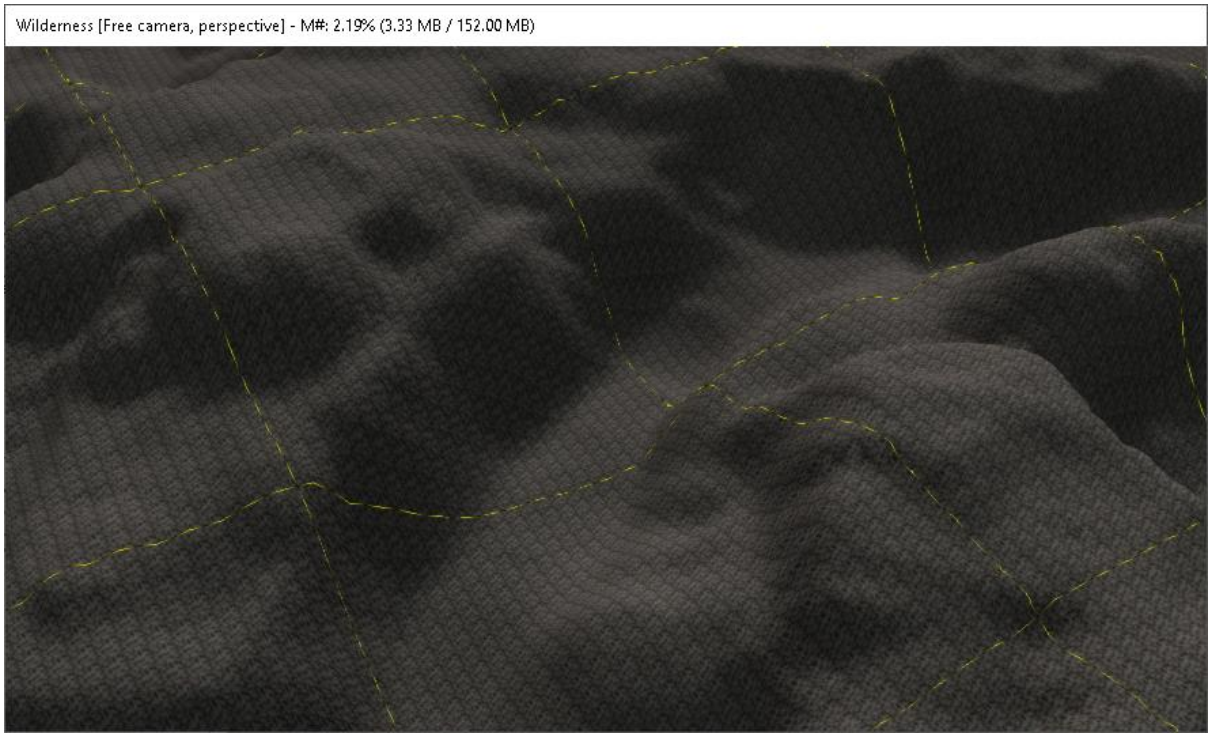


Figure 187 - Landscape automatically painted.

Despite setting a Max Slope, it looks like it painted the entire landscape. So what gives? The problem here is that we only defined one landscape texture.

If you open the Landscape Edit tool and press 'I' to look at landscape textures, you'll see what I mean.

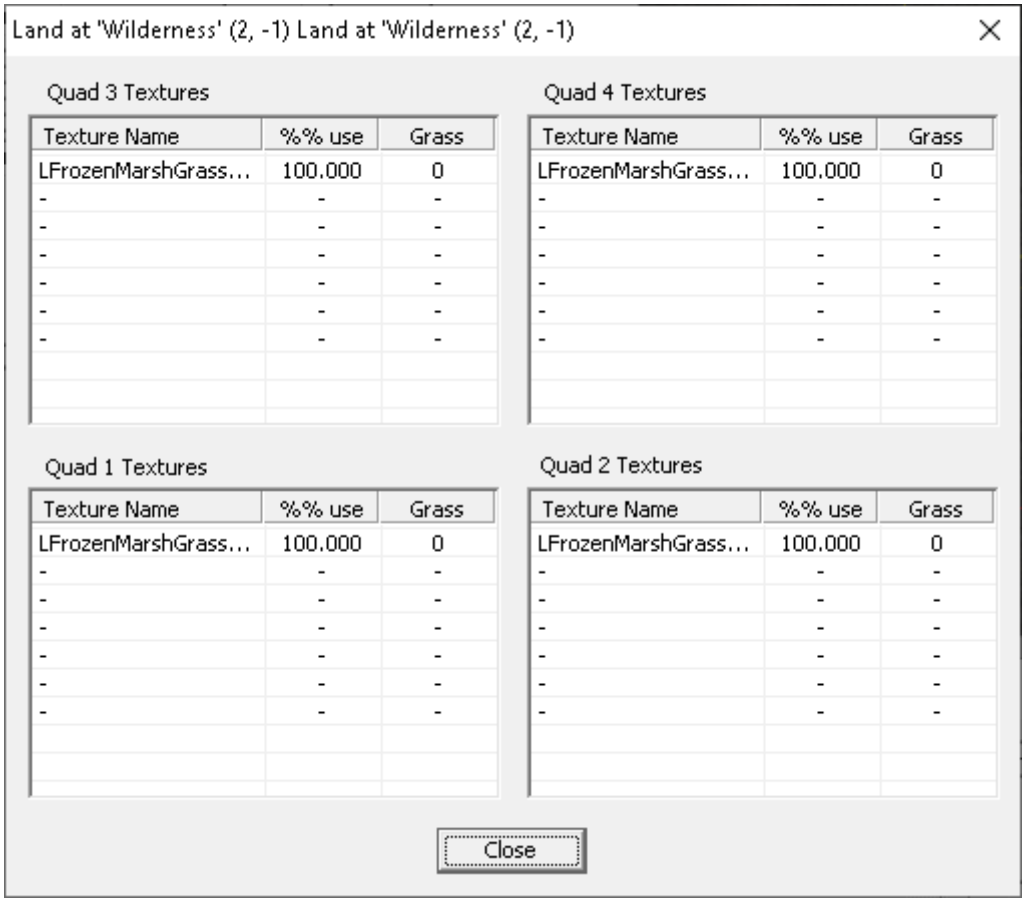


Figure 188 - Landscape textures.

LFrozenMarshGrass replaced LAND_DEFAULT, and seeing as though LAND_DEFAULT covers the entire landscape by default, LFrozenMarshGrass ended up covering the entire landscape too.

Important: Each cell needs at least one texture!

To rectify this, we need to specify a base texture that will be painted below everything.

For this example I added LDirt01. I moved it above LFrozenMarshGrass01NoGrass by highlighting it in the Generated Objects list and clicking on the Up button. I set Density to 100, Min Height to -200000 but left Max Slope at 90.

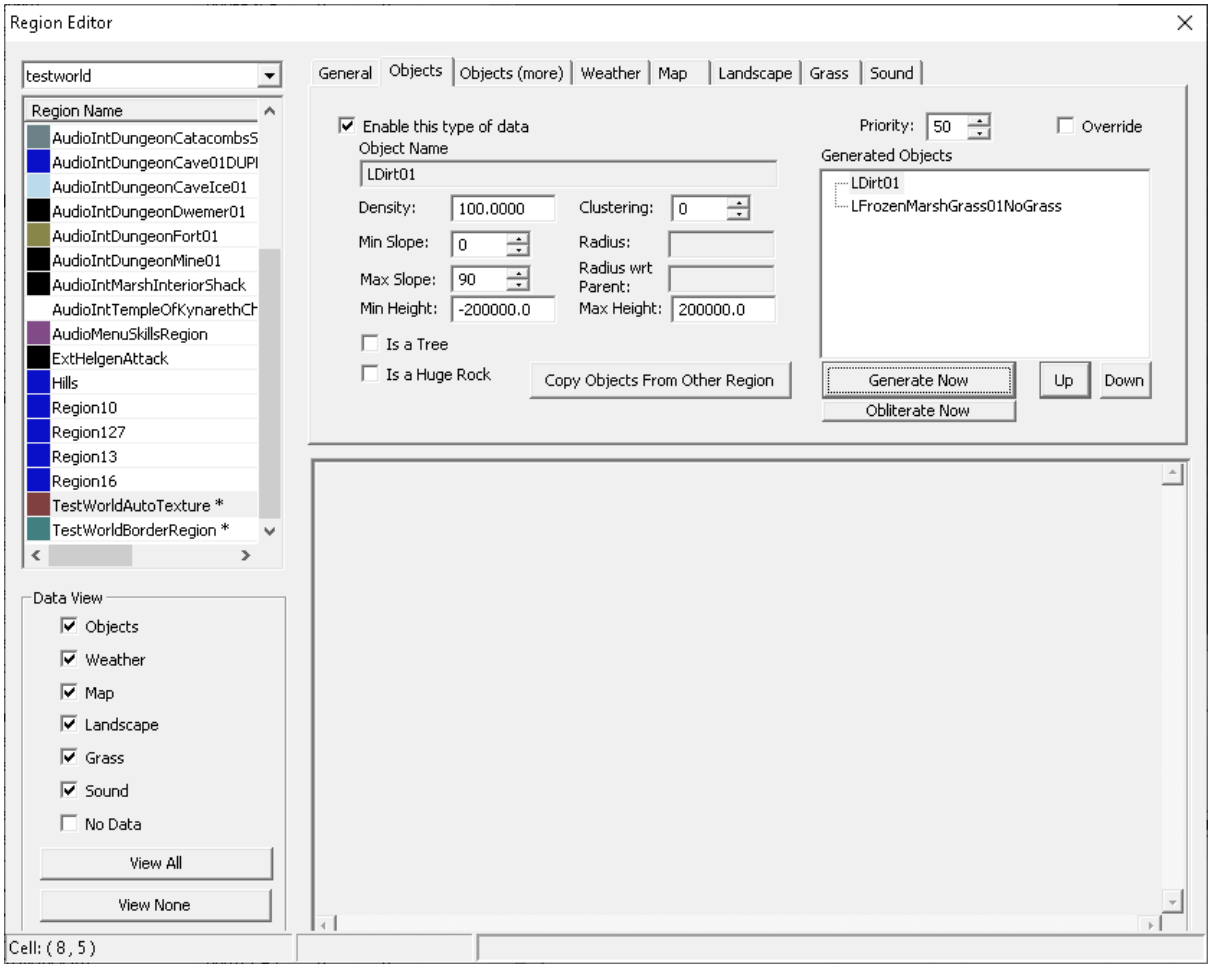


Figure 189 - LDirt01 added to Generated Objects list.

Click Generate Now again to regenerate landscape textures.

Note: The Obliterate Now button won't remove landscape textures. Landscape textures will be repainted when we click Generate Now.

Here's our landscape after adding LDirt01.

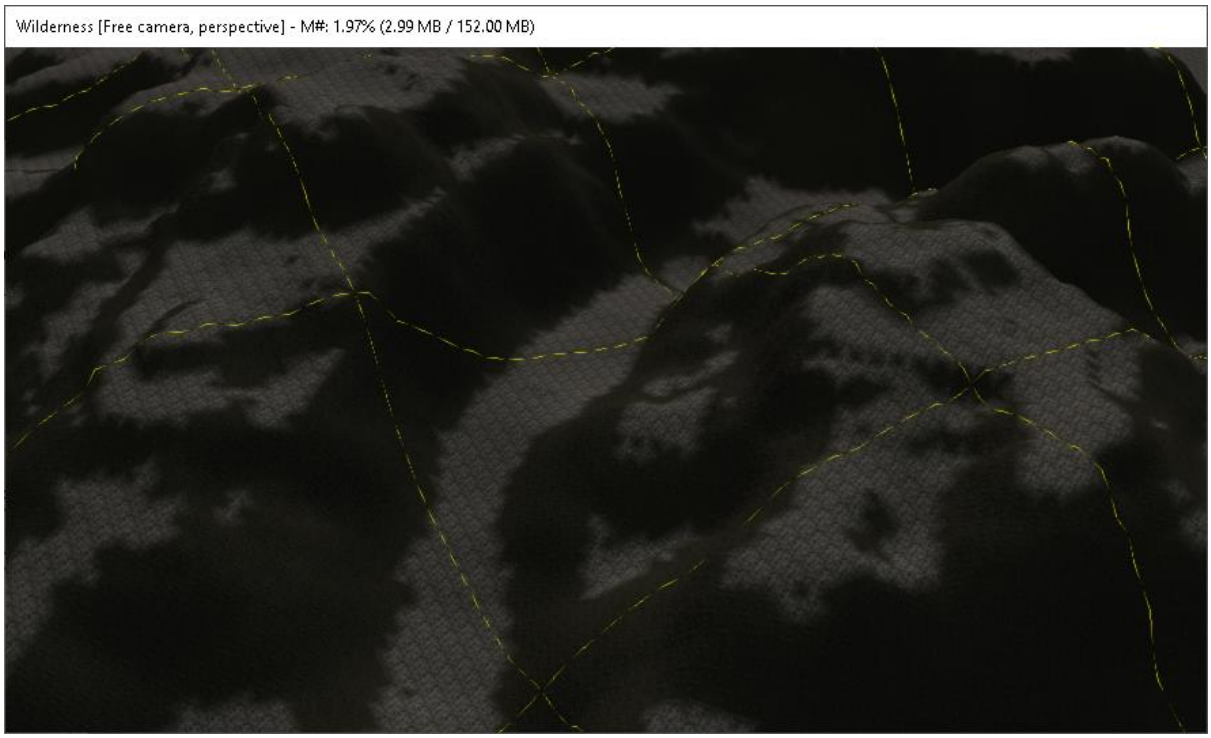


Figure 190 - Landscape after adding LDirt01.

As you can see, LDirt01 is now covering 100% of the landscape with LFrozenMarshGrass01NoGrass painted on only the flat areas.

Let's add one more texture to only the steeper slopes.

I added LRocks01NoRocks but this time set Min Slope to 40 and left Max Slope set to 90. Density was also set to 100 to ensure it covers LDirt01 on steep slopes.

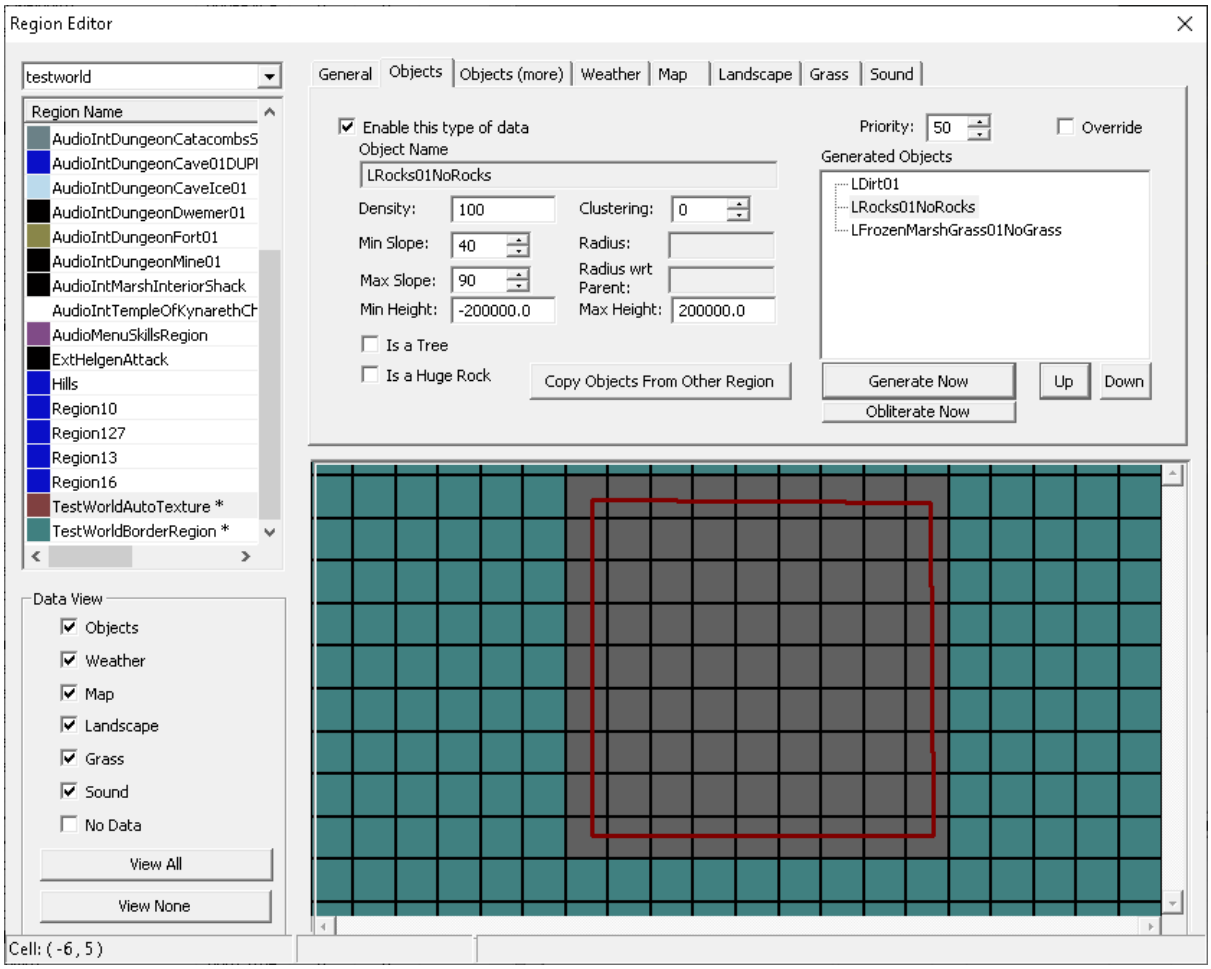


Figure 191 - LRocks01NoRocks added to Generated Objects list.

Again, click Generate Now to regenerate landscape textures.

As you can see, LRock01NoRocks was painted on the steep hill sides.

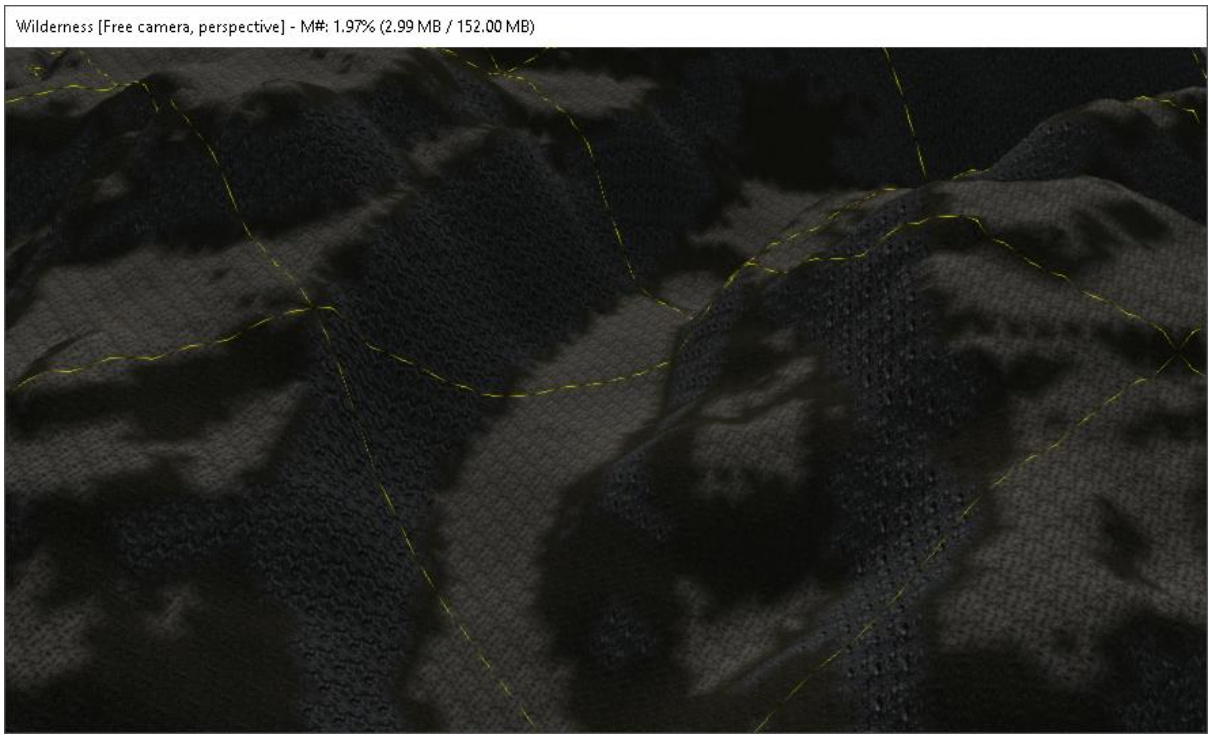


Figure 192 - Landscape after adding Rock01NoRocks.

To try and break up these textures and add some visual variety, we can add child textures by nesting them within the textures we've already added. We just need to be mindful of the number of landscape textures we've already added.

Remember: We shouldn't add more than 6 landscape textures.

If you drag and drop a new texture onto an existing texture in the Generated Objects list, it'll nest it as a child texture.

So in the following example I nested LFrozenMarshLichen01NoGrass below LFrozenMarshGrass01NoGrass, so wherever LFrozenMarshGrass01NoGrass is painted, LFrozenMarshLichen01NoGrass will have a chance to be painted as well.

To ensure it doesn't completely overwrite the parent texture, I reduced its density to 50 but kept the same Min Slope and Max Slope values as the parent.

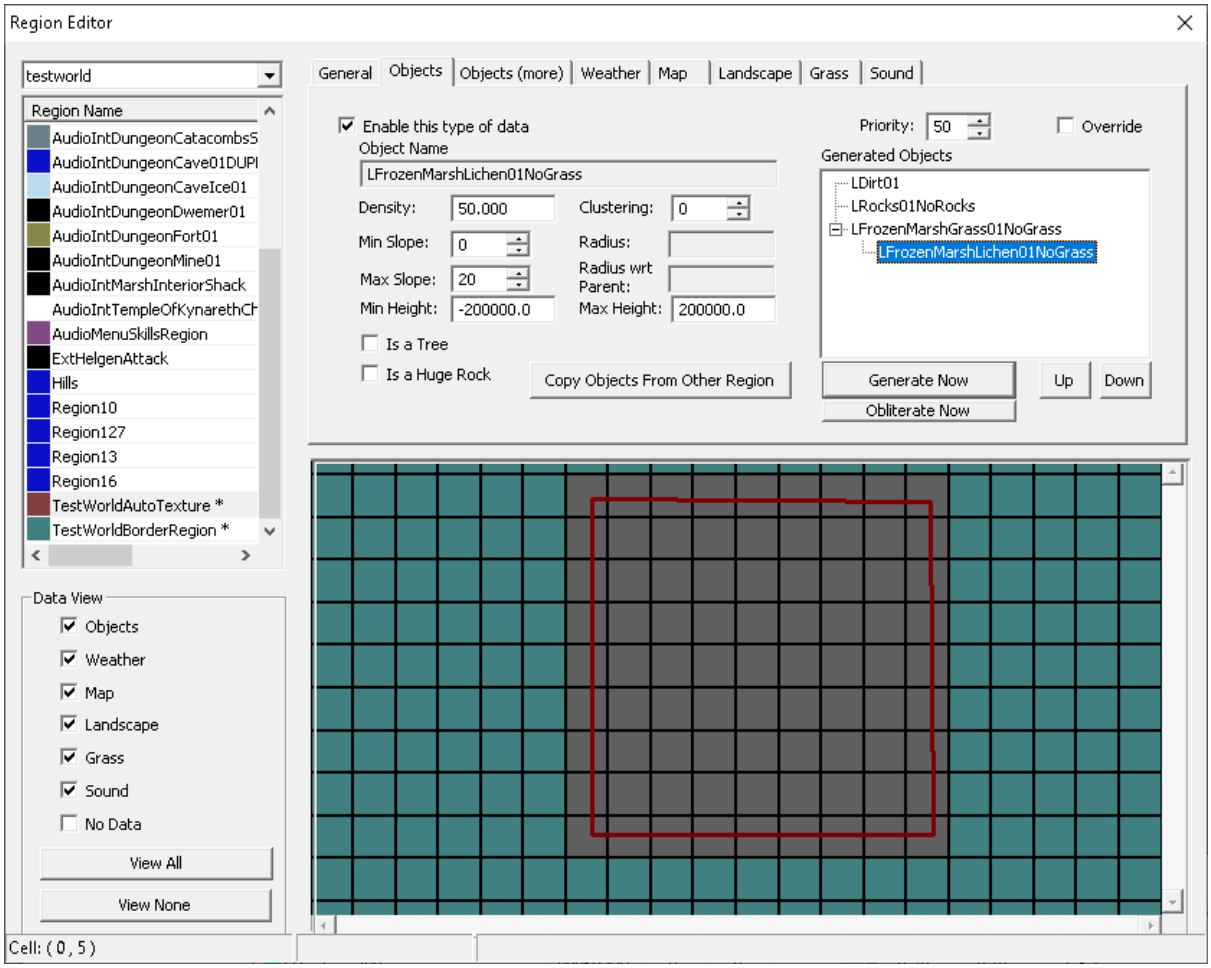


Figure 193 - LfrozenMarshLIchen01NoGrass added to Generated Objects.

And here's our end result:



Figure 194 - LfrozenMarshLichen01NoGrass blended with LFrozenMarshGrass01NoGrass.

As you can see, LFrozenMarshLichen01NoGrass is being painted over LFrozenMarshGrass01NoGrass, blending with the parent texture.

To paint the child texture in a blotchier manner, we can increase the Clustering value.

In the following example I increased Clustering to 50.

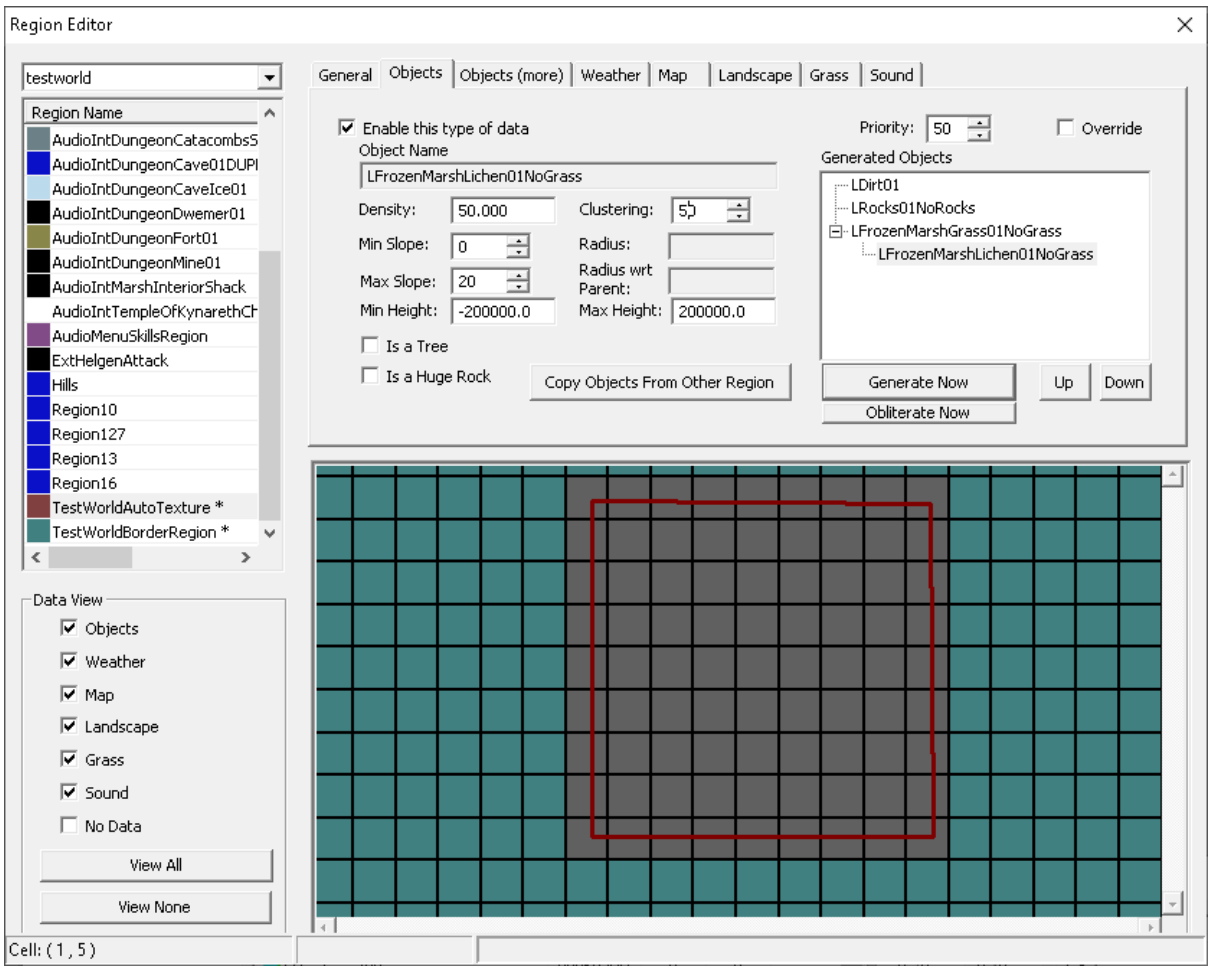


Figure 195 - Increasing the Clustering value.

As you can see, LFrozenMarshLichen01NoGrass is now spread around more randomly.



Figure 196 - LfrozenMarshLichen01NoGrass with Clustering set to 50.

We can think of increasing the Clustering value as being the same as reducing the size of a random noise mask. The higher the value the noisier the distribution of the texture will be.

I'll be looking at how we can generate objects randomly across our sample region next.

Note: I'd recommend starting with a small region first to test out the density and distribution of auto generated objects. You can always expand that region afterwards once you're satisfied with how objects are being populated then generate again to fill a wider area.

In the Object Window, go to WorldObjects > Static.

For this example I'm going to start by dragging and dropping RockL01, RockL02 and RockL03 into the Generated Objects list in the Region Editor.

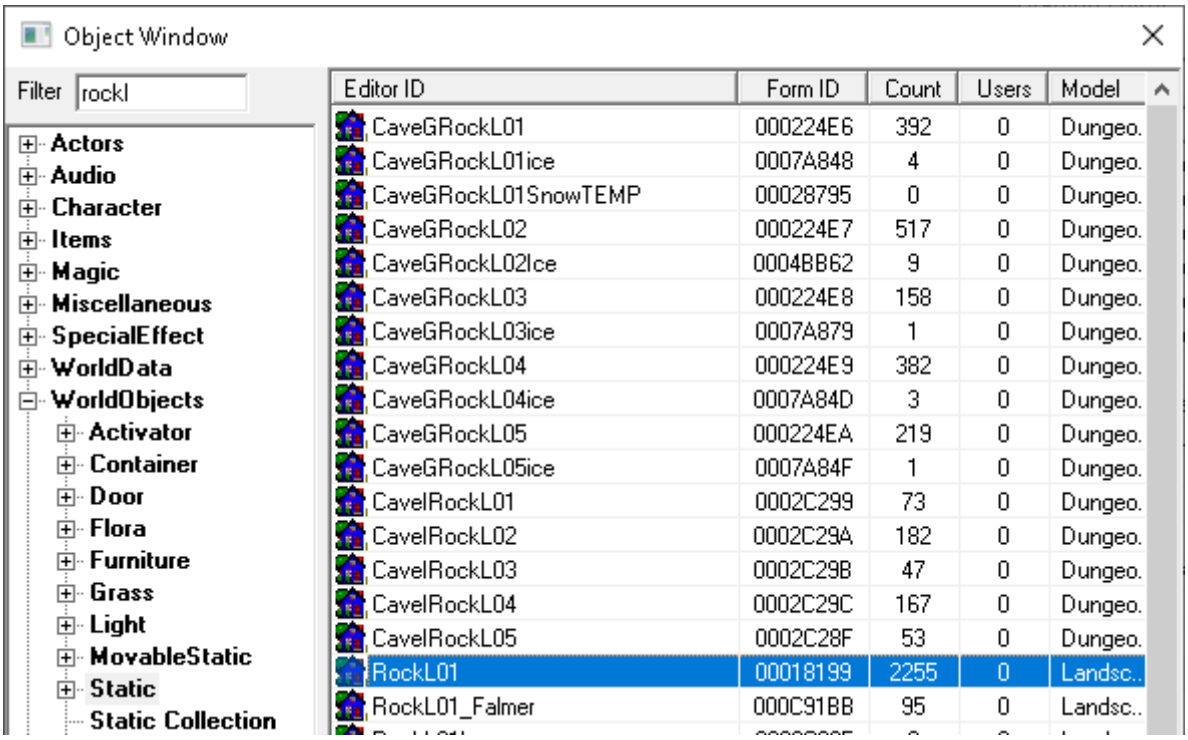


Figure 197 - Static rocks.

For each rock, set Min Height to -200000, tick 'Is a Huge Rock' then click on Generate Now.

Note: 'Is a Huge Rock' will allow smaller objects to be populated on top of it.

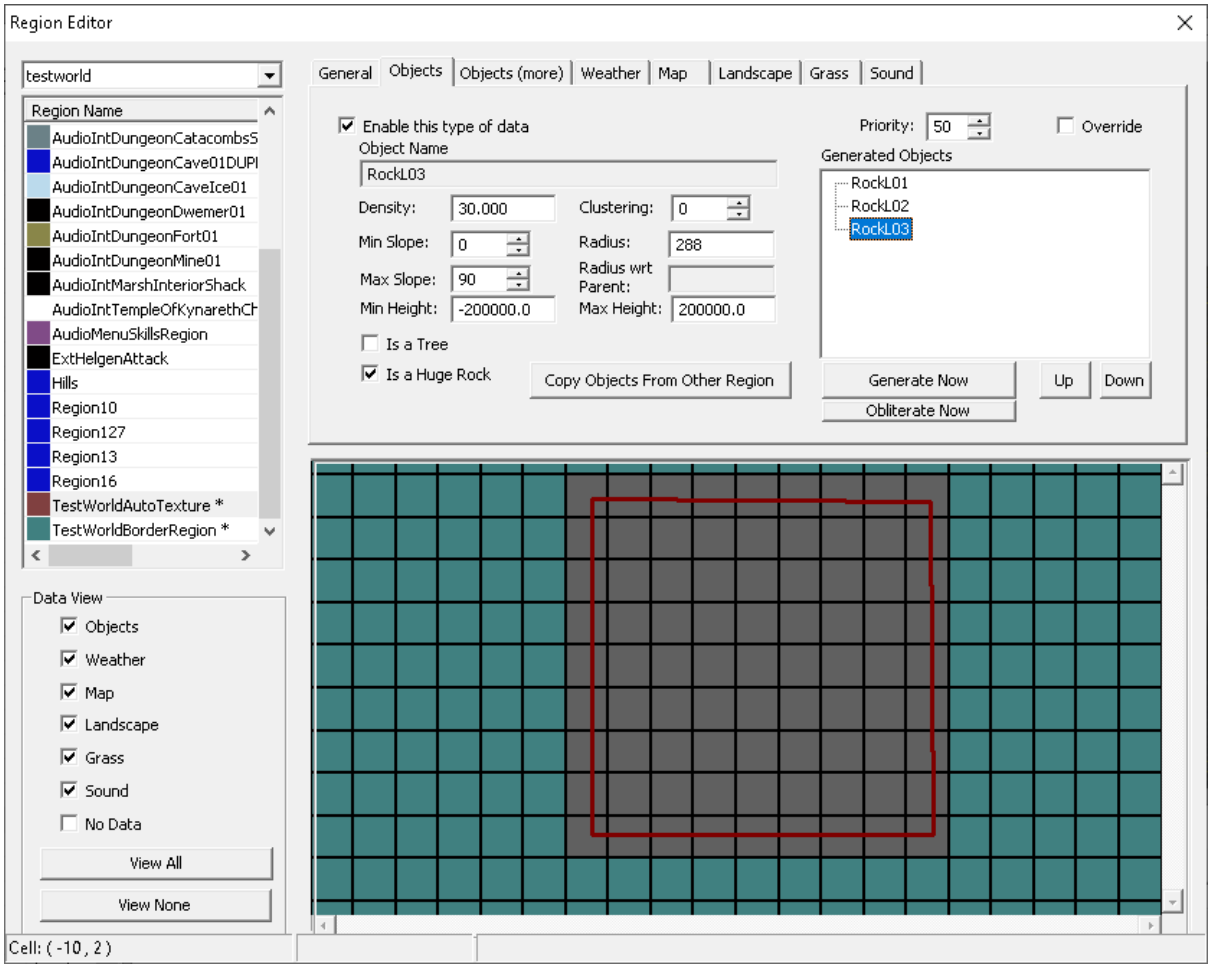


Figure 198 - Populating the region with three different types of rock.

Click 'Yes' to the confirmation pop-up.

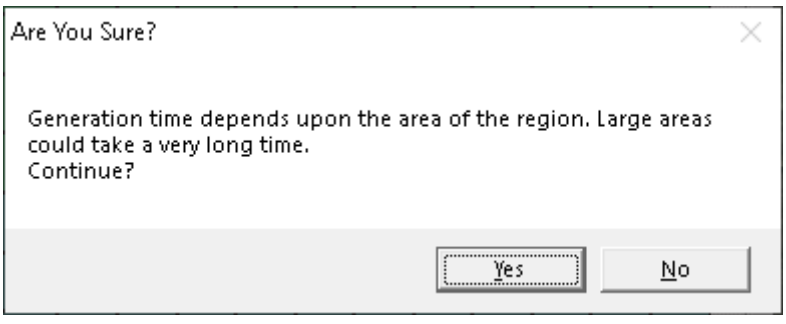


Figure 199 - Auto generate confirmation.

Here's what our distribution looks like with these default settings:

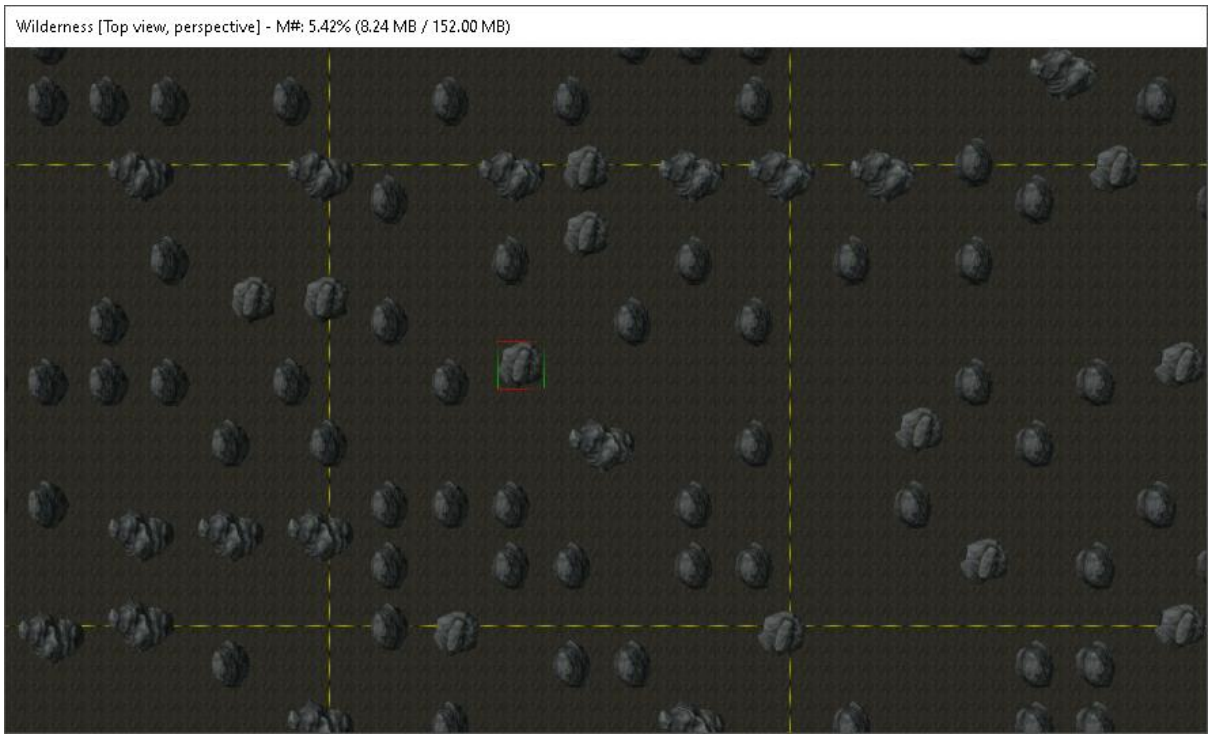


Figure 200 - Object distribution.

You'll notice that all the objects are facing the same direction. We'll fix this shortly but for now let's take a look at the different ways we can control the distribution of objects in the region.

Click Obliterate Now to remove the objects generated in this region. Click 'Yes' to the confirmation pop-up.

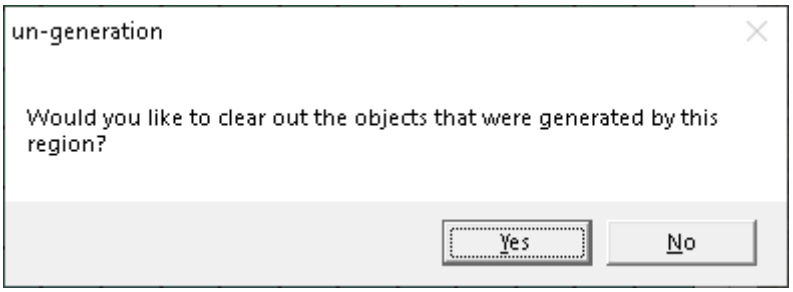


Figure 201 - Auto generate obliteration confirmation.

Let's look at increasing the Density of RockL02 next and see what that looks like. Click on RockL02 in the Generated Objects list to select it and set the Density field to 60.

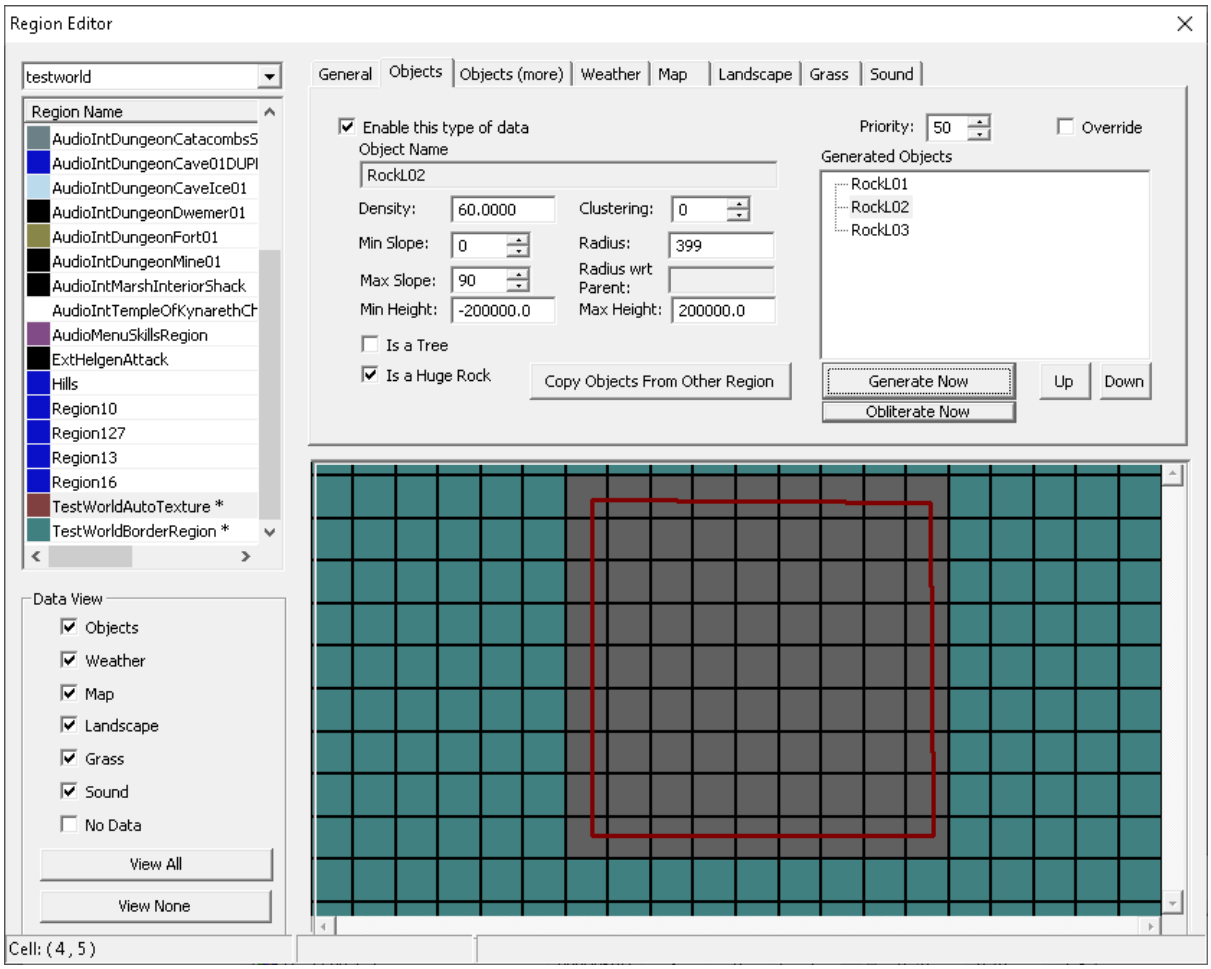


Figure 202 - Increasing RockL02 Density value.

The Radius for RockL02 by default is 399. This field determines how close this object could potentially be placed next to other instances of itself.

Reduce the Density value on RockL01 and RockL03 to 10 then click Generate Now.

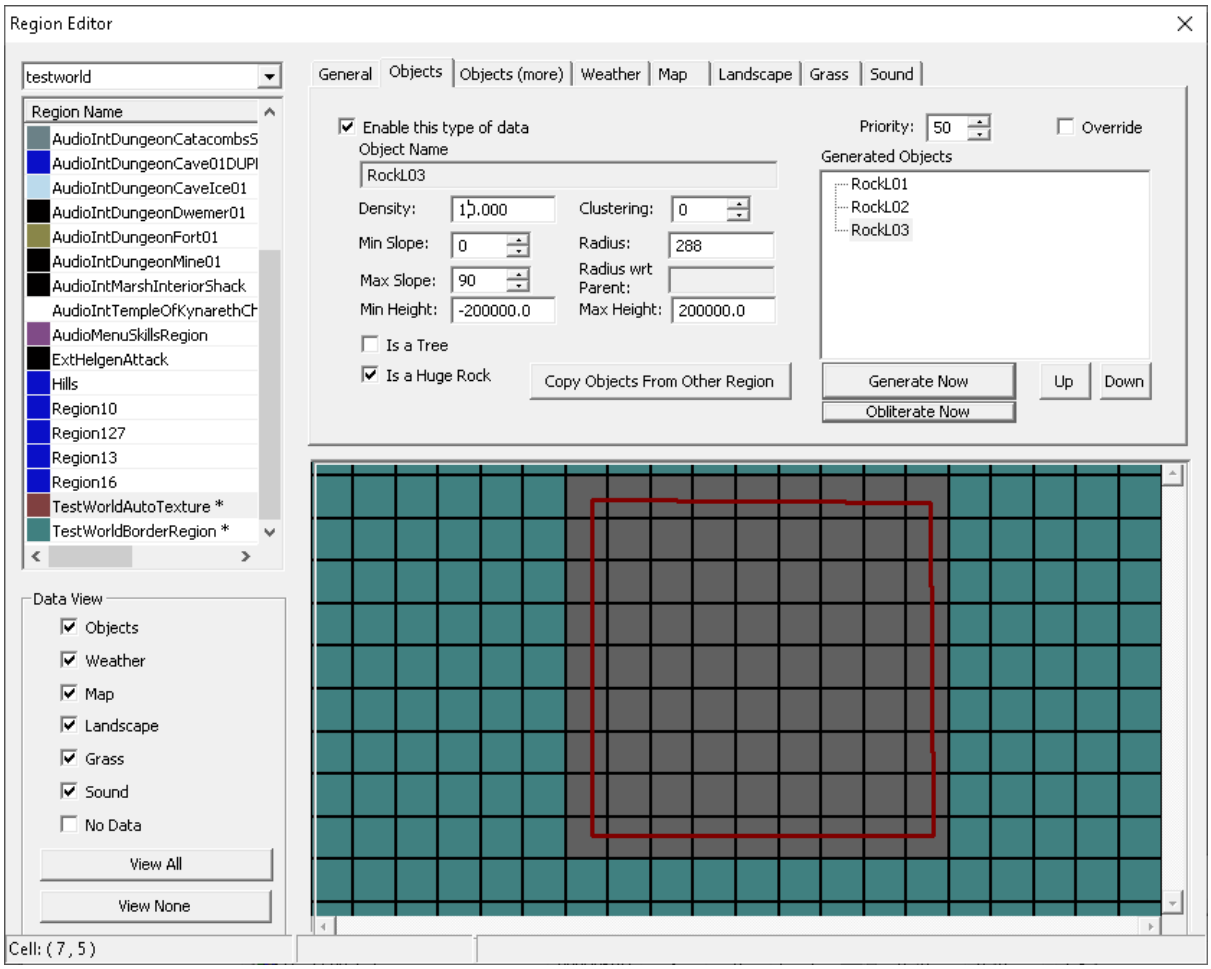


Figure 203 - Decreasing the RockL01 and RockL03 Density value.

Important: Ticking the Override tickbox will force this region's object generation settings to take precedence over the object generation settings from other regions that a cell may also be within.

As you can see there are now a lot more instances of RockL02 than RockL01 and RockL03.

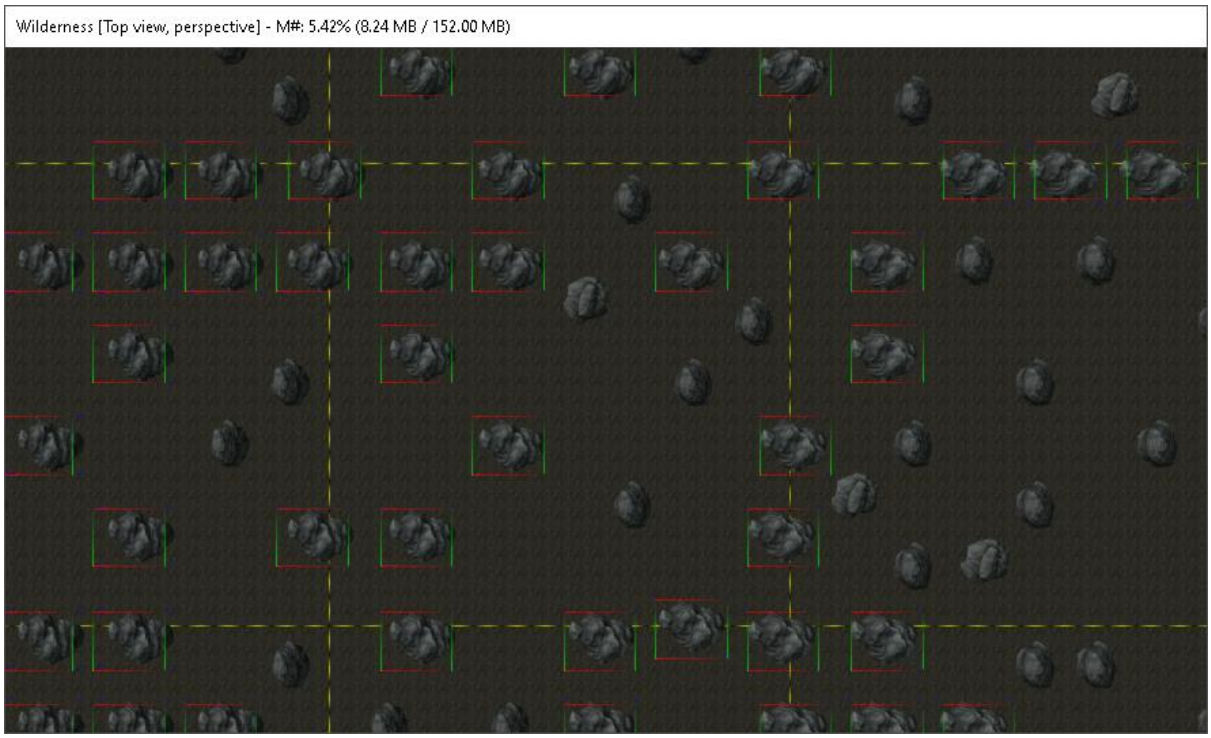


Figure 204 - More instances of RockL02.

To control the likelihood that an object will be placed close to another instance of its type, increase the Clustering value.

By default the Min Slope is set to 0, allowing an object to be placed on flat surfaces. The default Max Slope is 90 meaning the object may be generated on steep cliff faces.

Min Height controls the lowest elevation this object may be populated at. By default that's 0.0. Since our landscape is technically below the ocean surface, we need to reduce this number which is why I set it to -200000.

Similarly, Max Height controls the maximum elevation that an object can be populated at. The object won't be populated above the height set in this field. By default that will be 200000.0. If you only want an object to be generated below the ocean surface you would need to set this field to 0.0.

Next let's look at setting up child objects.

From the Object Window I'm going to drag and drop RockS01, RockS02 and RockS03 onto RockL02. For this example I'm going to remove RockL01 and RockL03 for now.

Set the Min Height on RockS01, RockS02 and RockS03 to -200000 and the Density to 10.

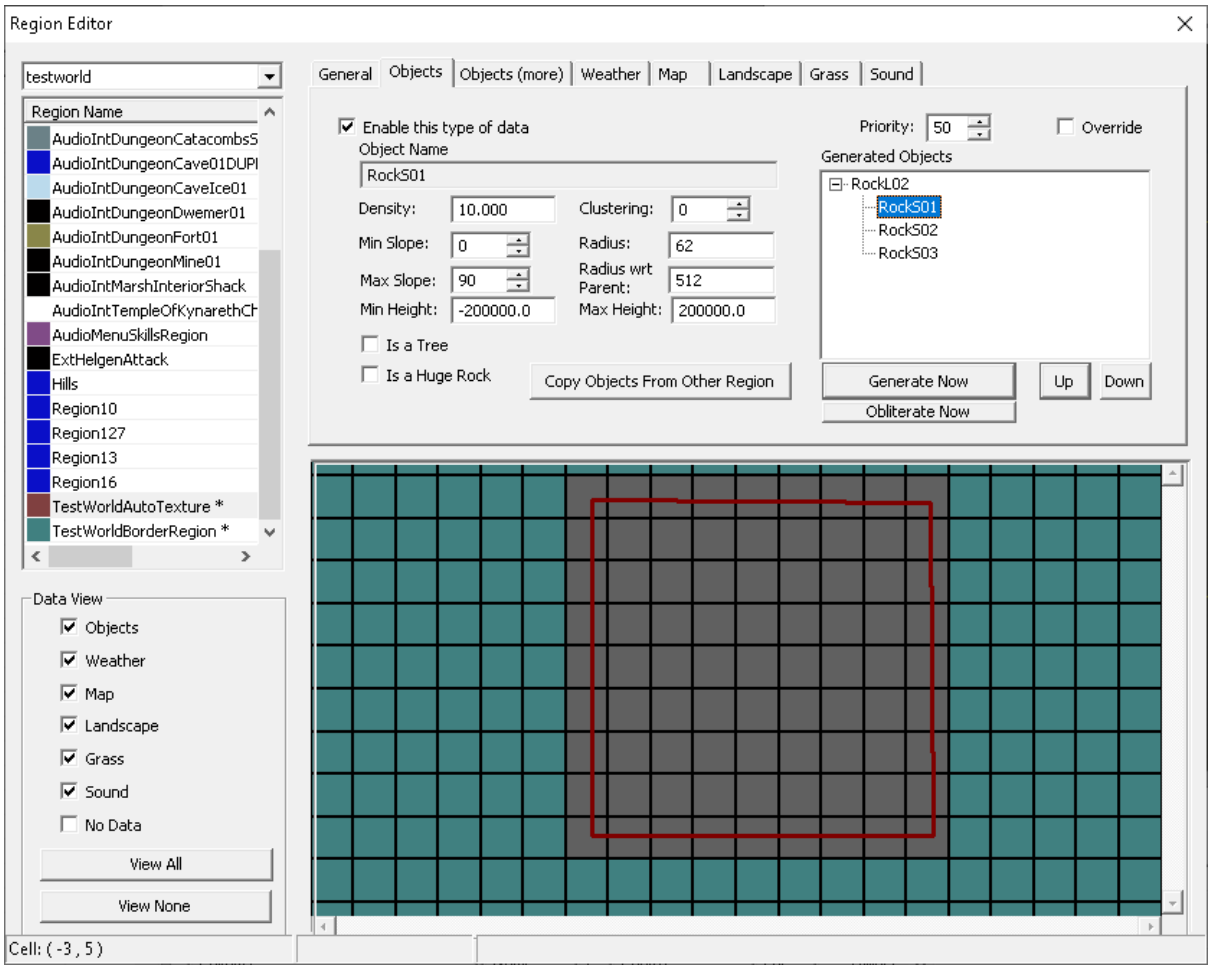


Figure 205 - Adding child objects.

To get a better idea of what's going on I'm going to reduce the Density of the parent object RockL02 to 5 before generating.

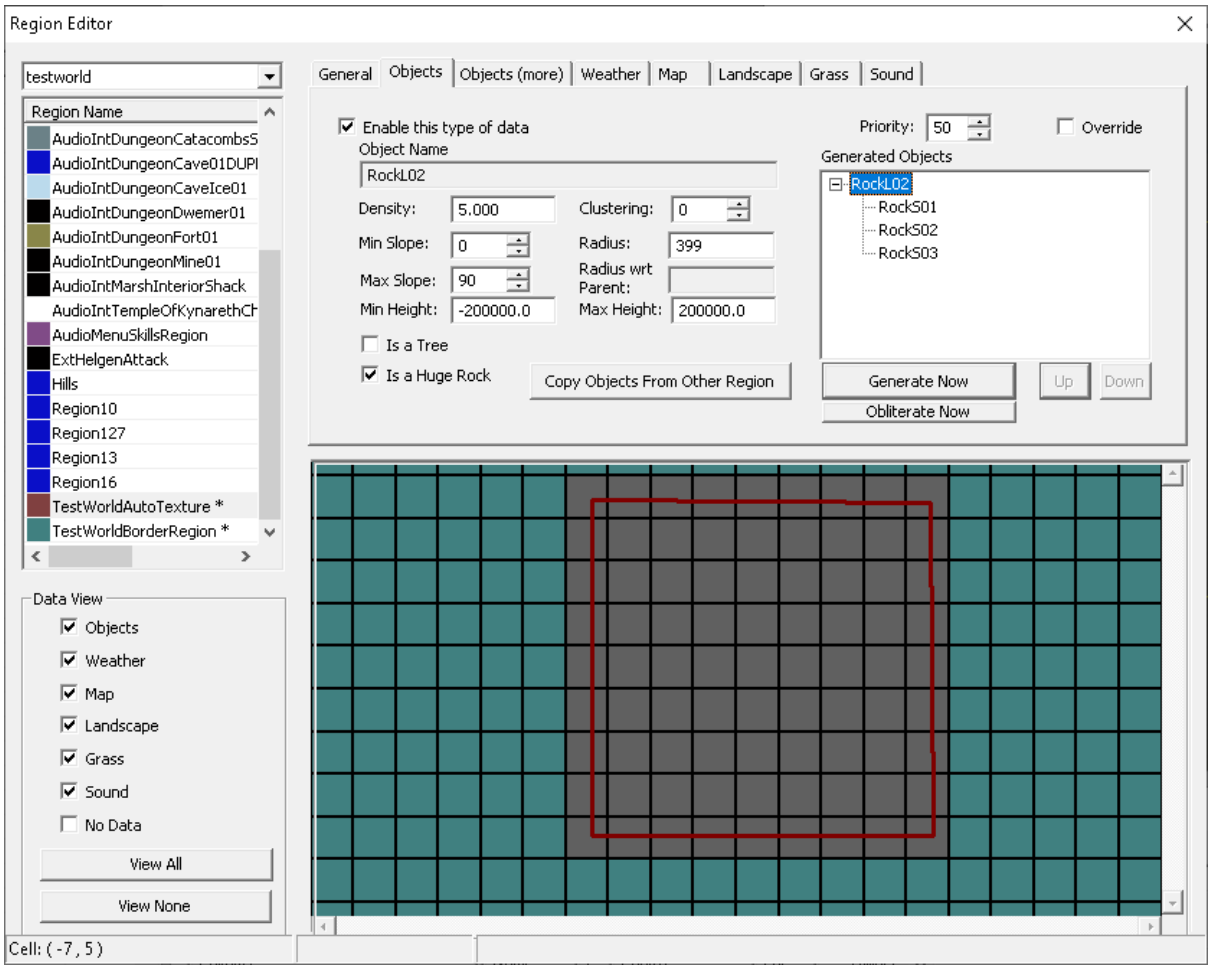


Figure 206 - Reducing the Density of RockL02.

And here's what we get. As you can see, the small rocks are being automatically populated around the large rocks.

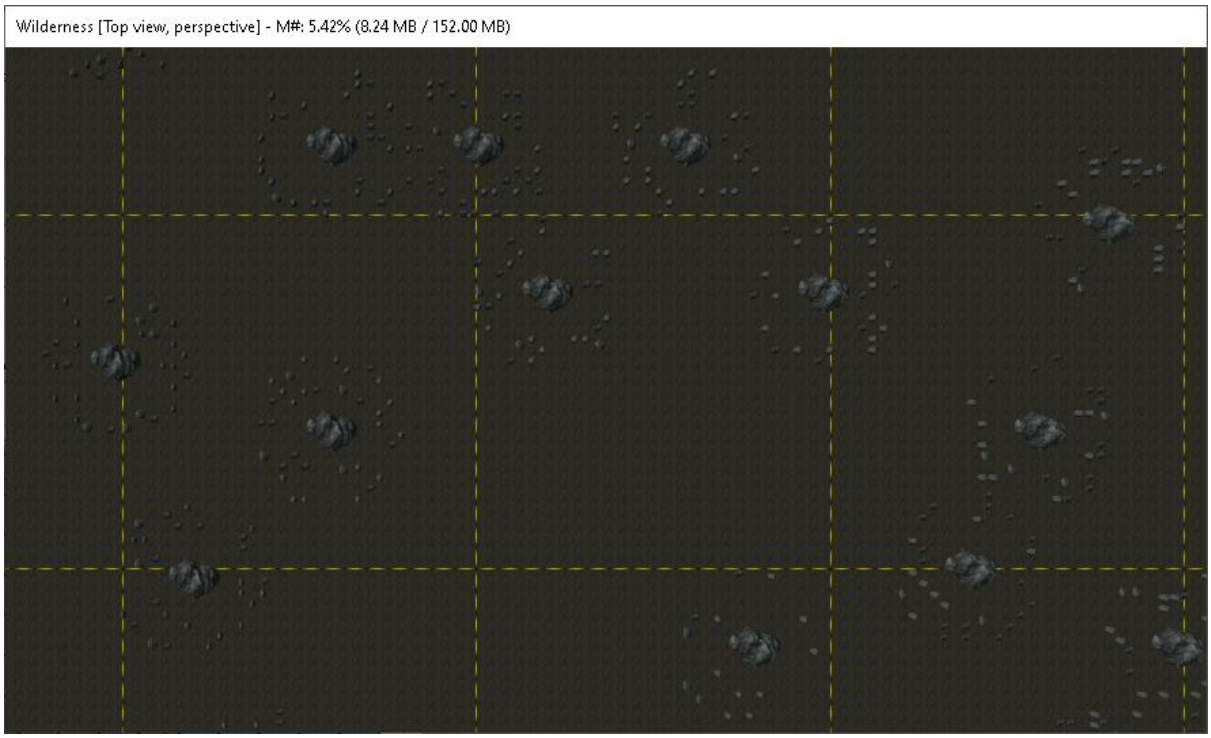


Figure 207 - Small rocks generated around the large rocks.

Radius controls the minimum distance this object can be from another instance of its type.

On the other hand, the 'Radius wrt Parent' field controls how far away from the parent the children object can be generated.

For RockS01, RockS02 and RockS03 set 'Radius wrt Parent' to 1024 then generate again.

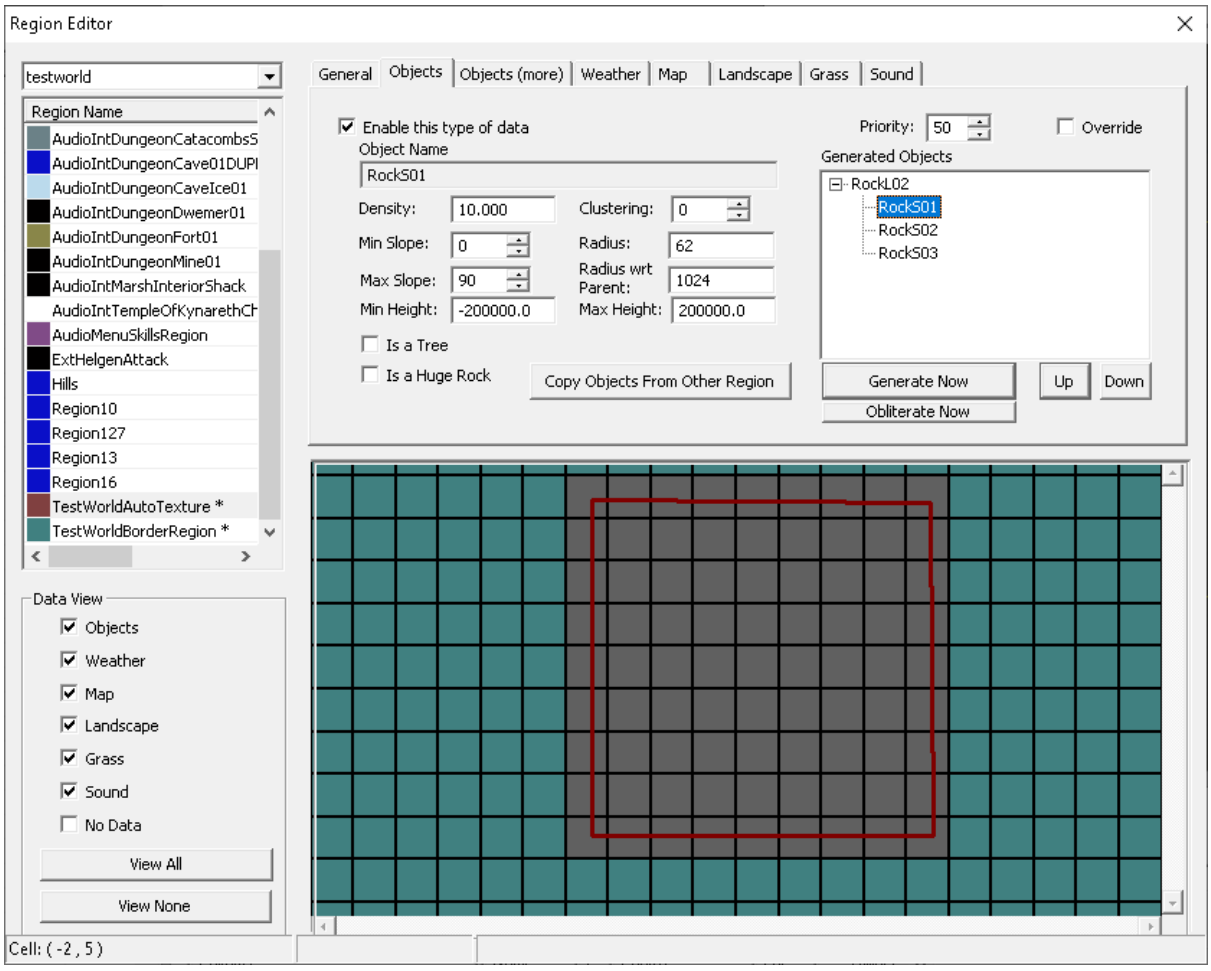


Figure 208 - Increasing the Radius wrt Parent field.

So as you can see this has increased the distance from which child objects will be generated from the parent, which has also resulted in many more child objects being generated.



Figure 209 - More child objects generated around parent.

Here's a look from a closer angle:

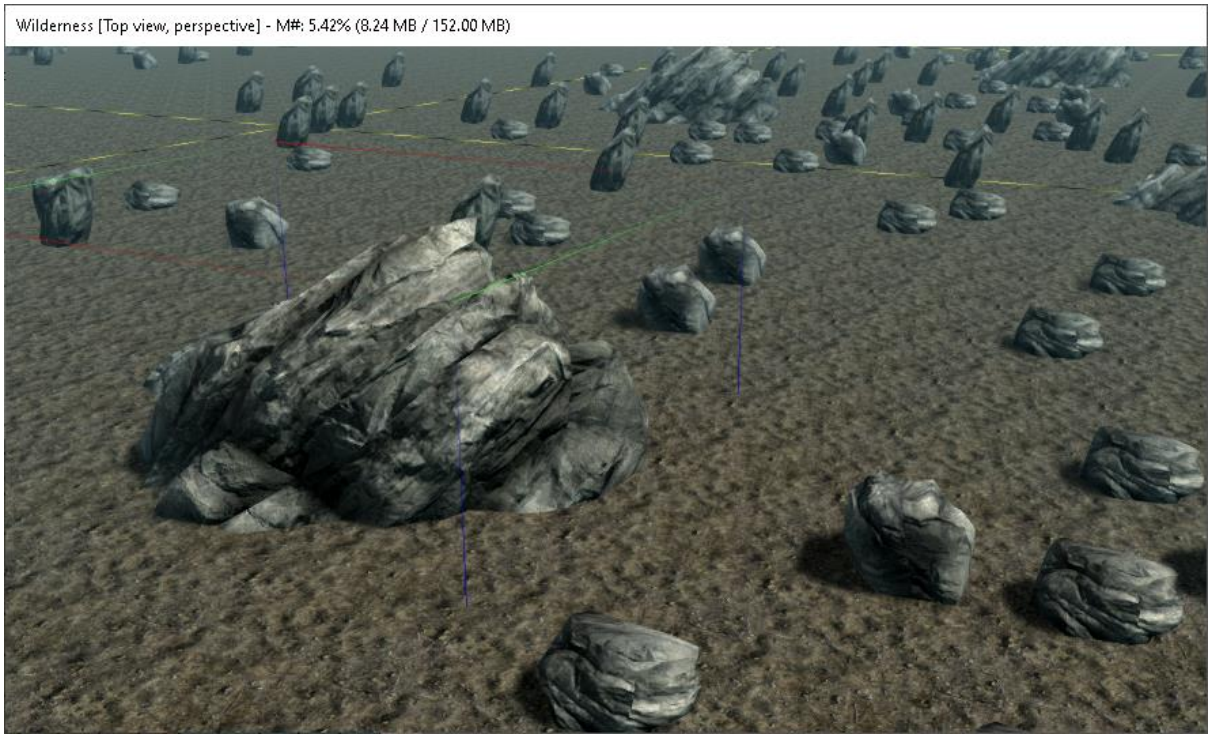


Figure 210 - A closer look at the auto generated rocks.

All these rocks look too uniform so let's do something about that next.

To add some random variance to these objects we'll need to switch over to the Objects (more) tab.

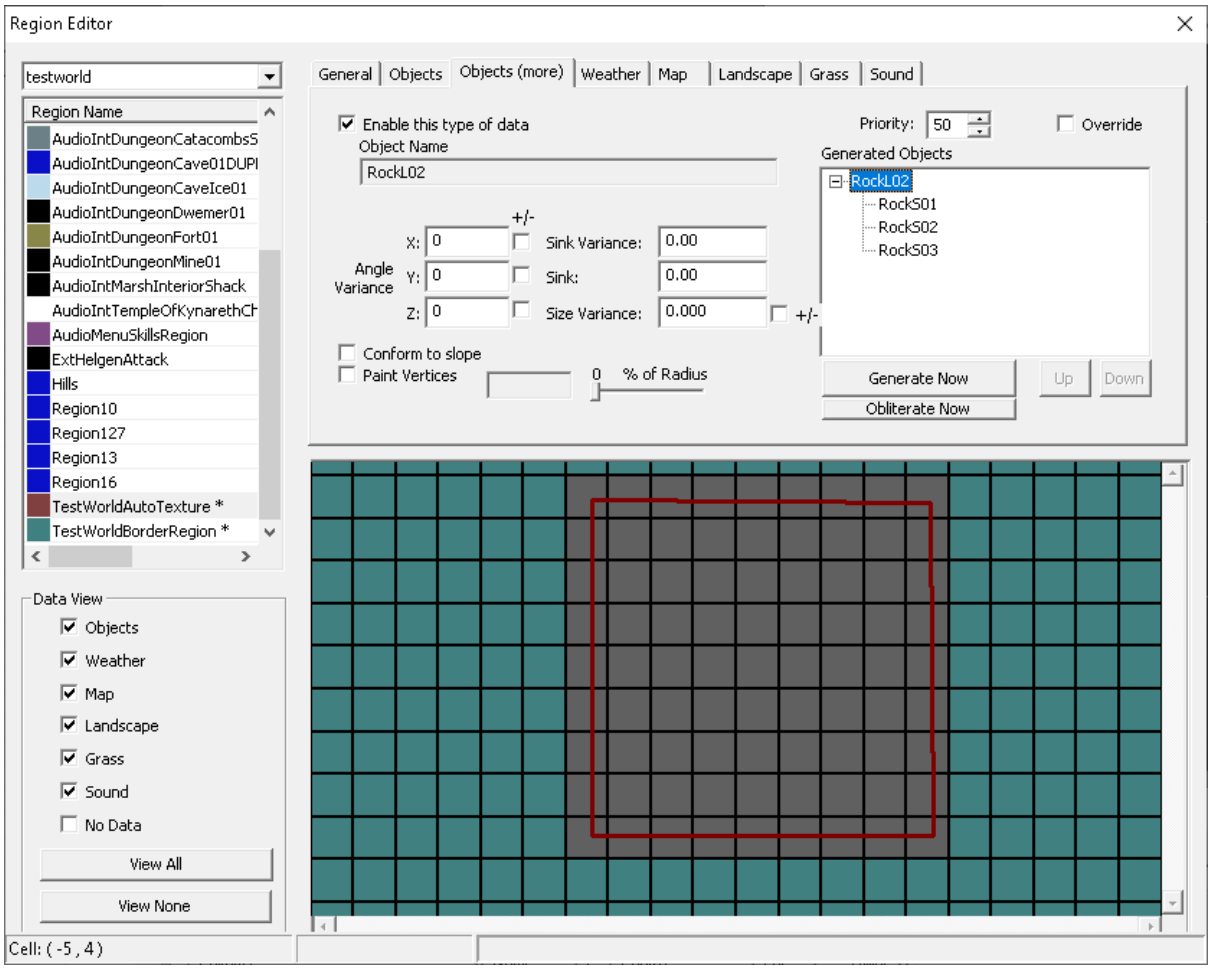


Figure 211 - Objects (more) tab.

To add some random rotation on the Z axis I set Z to 180.

I also set the Size Variance field to 0.400 allowing the scale of the object to randomize in a similar manner. Ticking the '+/-' button will allows negative values, i.e.: -0.4 to 0.4 instead of just 0 to 0.4.

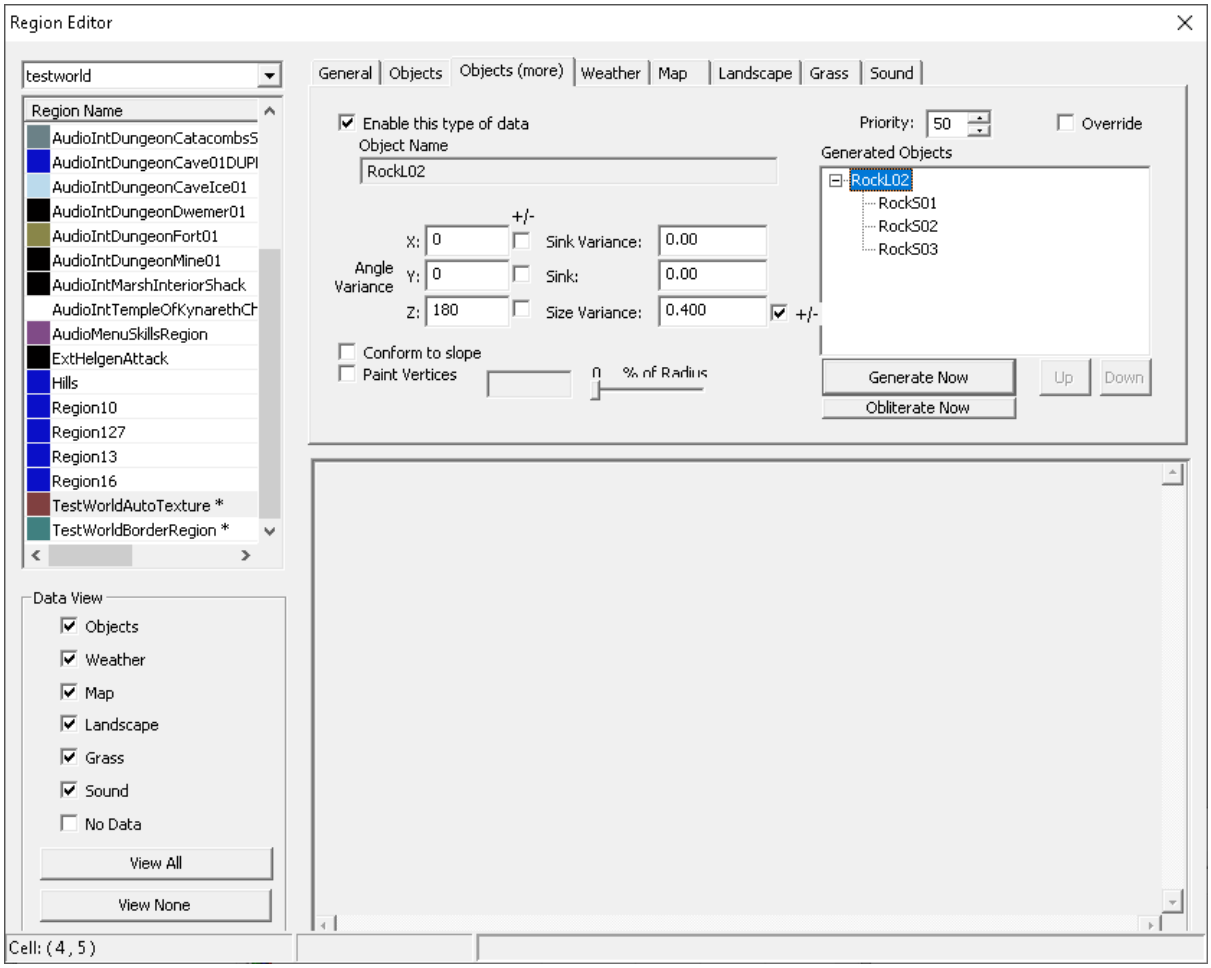


Figure 212 - Added Angle and Size variance.

After generating, we can see that each object is now facing a different direction and object scaling isn't as uniform as before.



Figure 213 - Objects with a little bit of variance.

Sink controls how far into the ground a generated object will be placed at, with Sink Variance adding some randomization to this amount.

Conform To Slope rotates the object in correlation to the slope it's placed on.

Let's try adding some trees next for more variety.

In this example I added TreeAspen01 and TreeAspen02.

I set the Min Height to -200000, increased their Density to 60 and reduced the Radius to 128. I also ticked 'Is a Tree' on both of them.

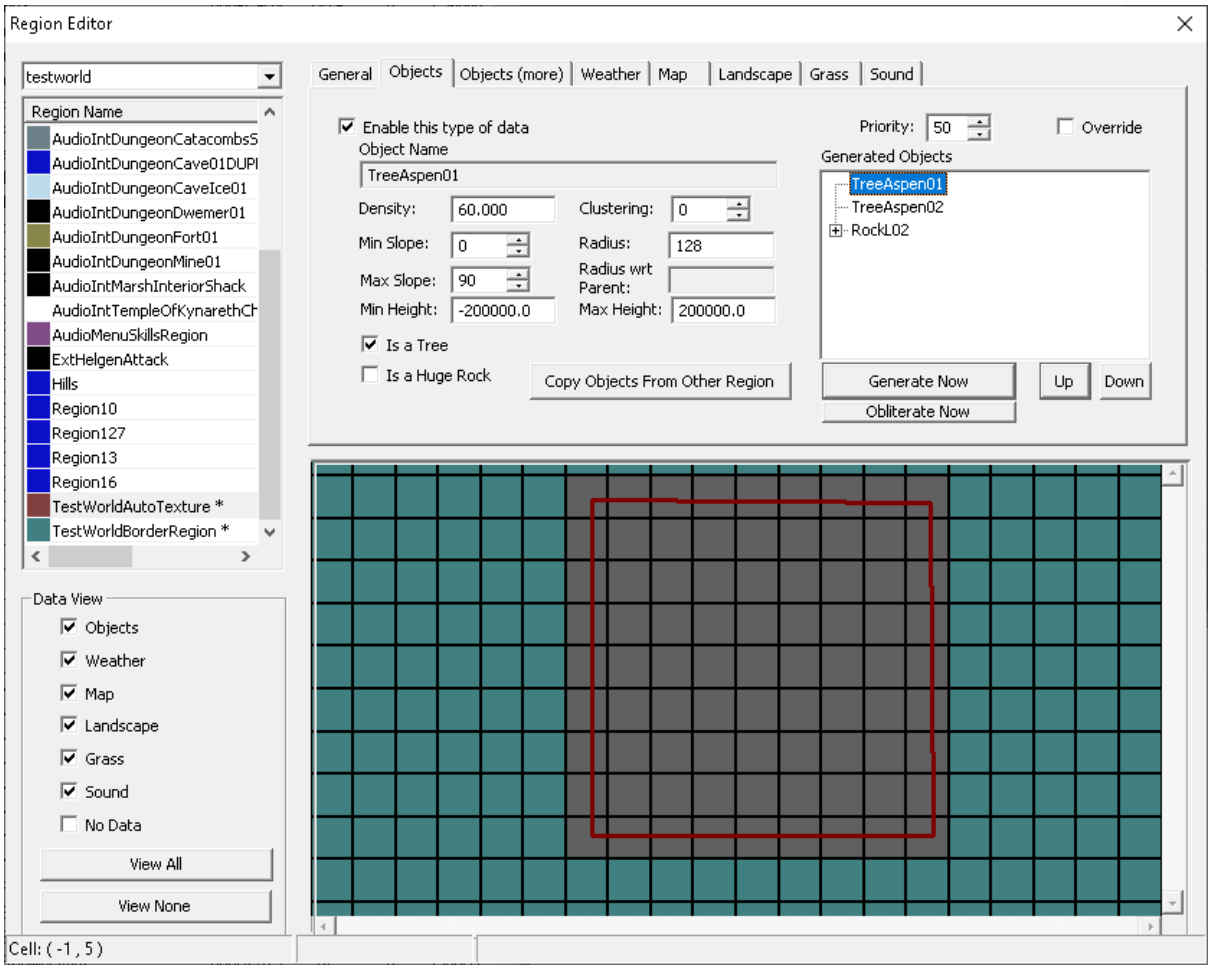


Figure 214 - Added aspen trees to Generated Objects list.

After generating here's what we have:

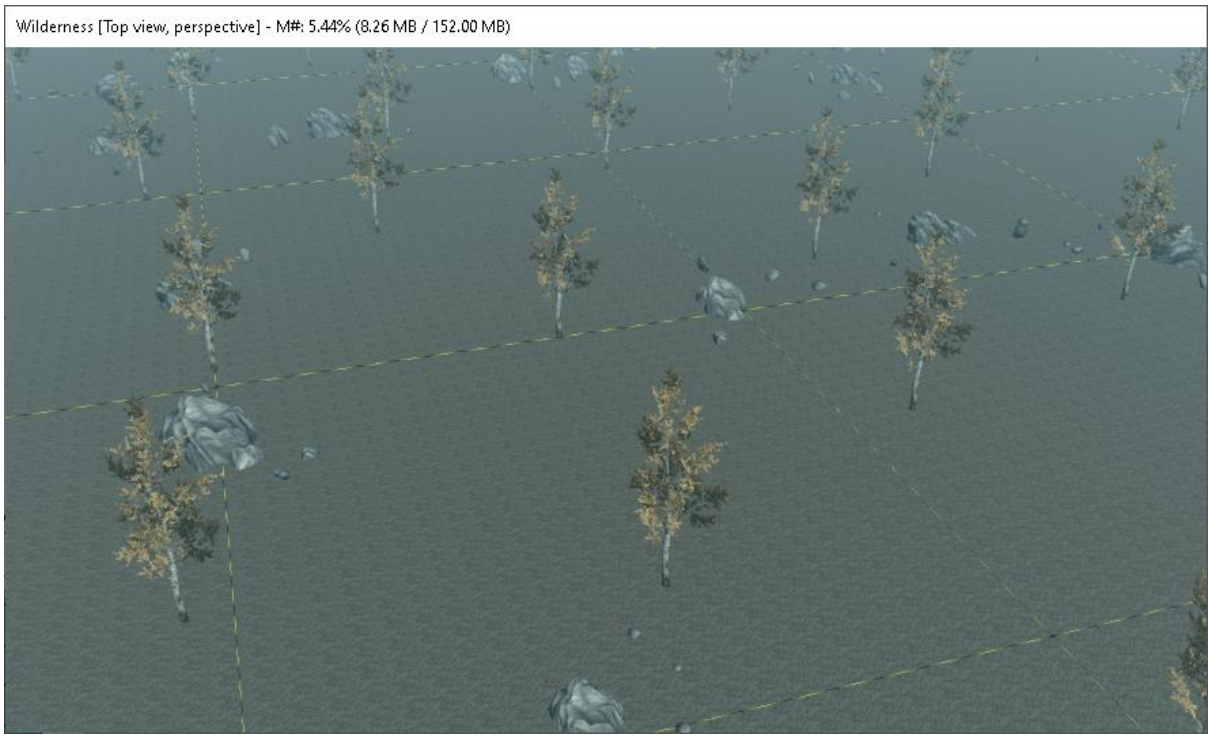


Figure 215 - Aspen trees generated on landscape.

The landscape looks a bit bare beneath them so let's add some thickets next.

From the Object Window I nested TreeThicket01 beneath each tree.

I set Min Height to -200000, Density to 60 and Radius wrt Parent to 1024.

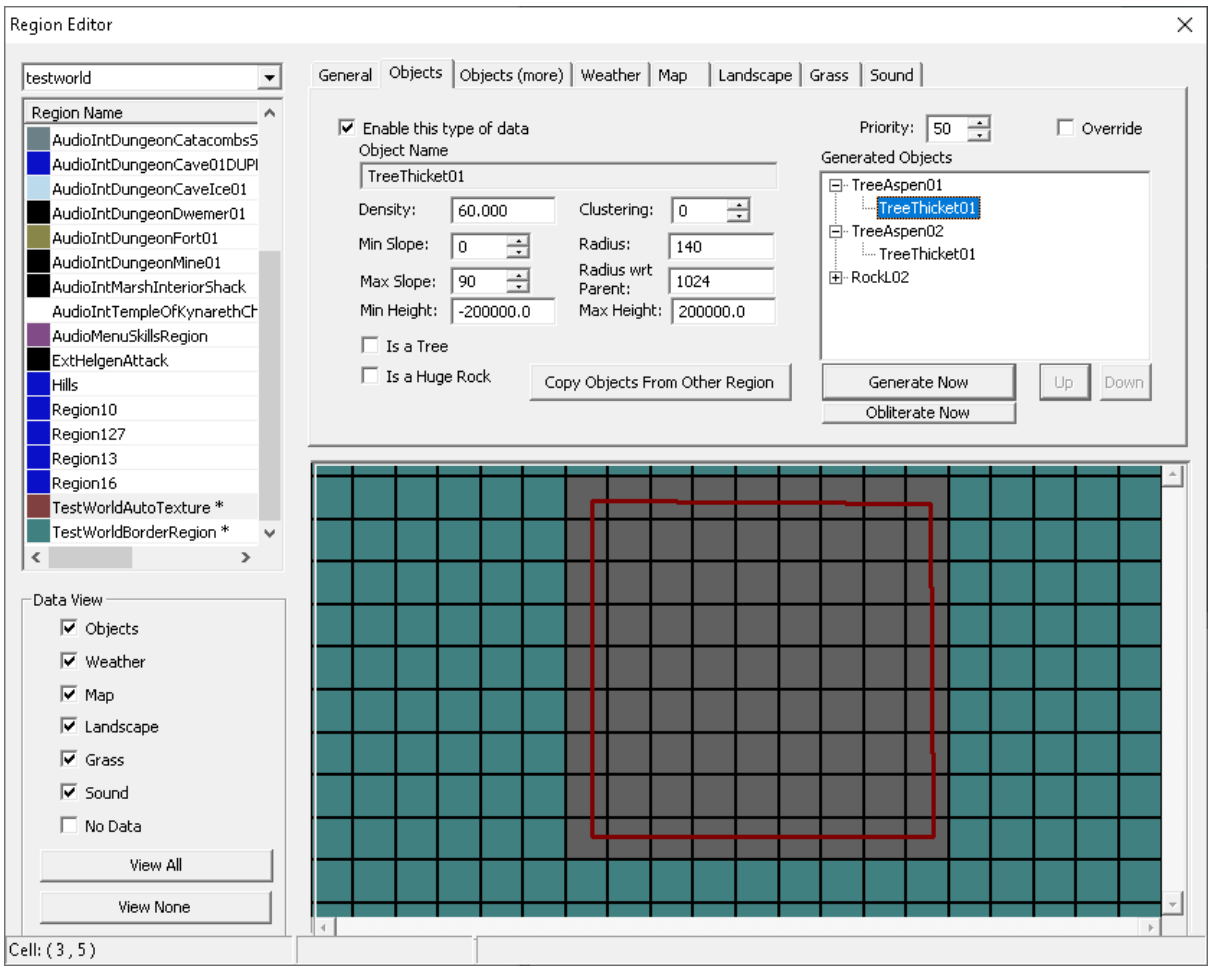


Figure 216 - Adding thickets.

You should now see thickets appear around the trees.

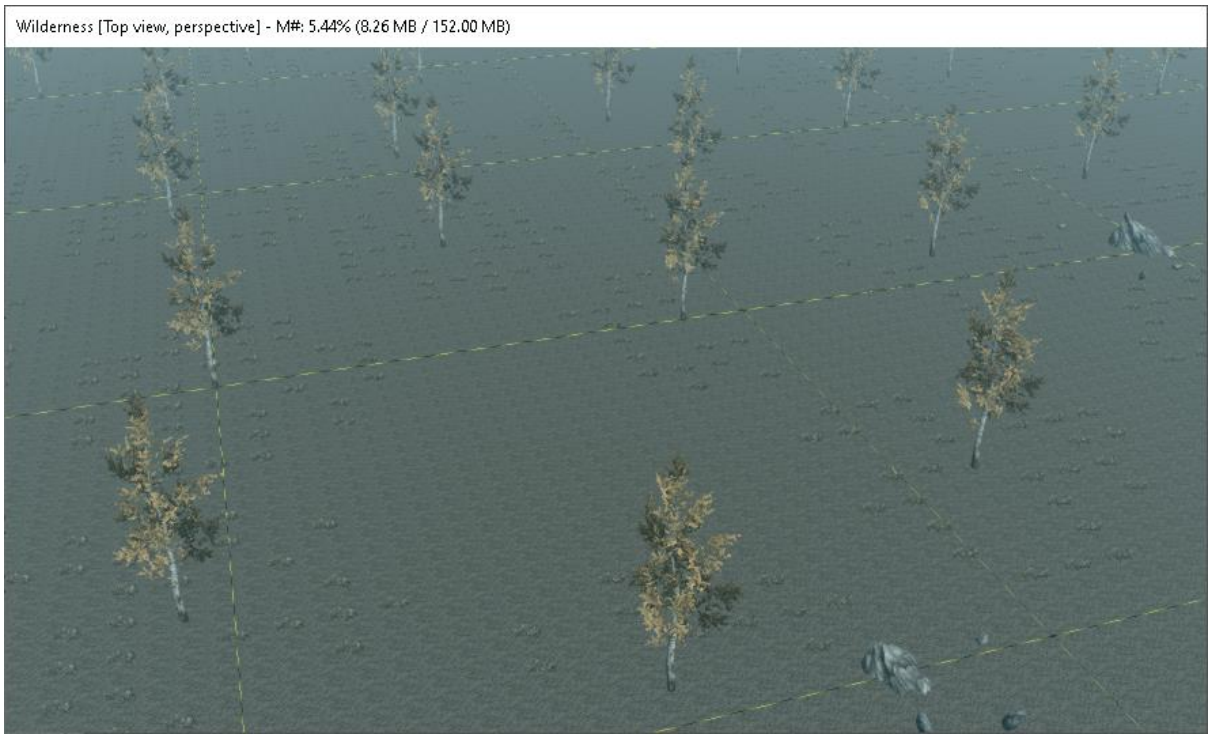


Figure 217 - Thickets appearing around the trees.

To fix the uniformity, let's try adding some variance next.

Set the Angle Variance for Z to 180, and the Size Variance to 0.200 then generate again.

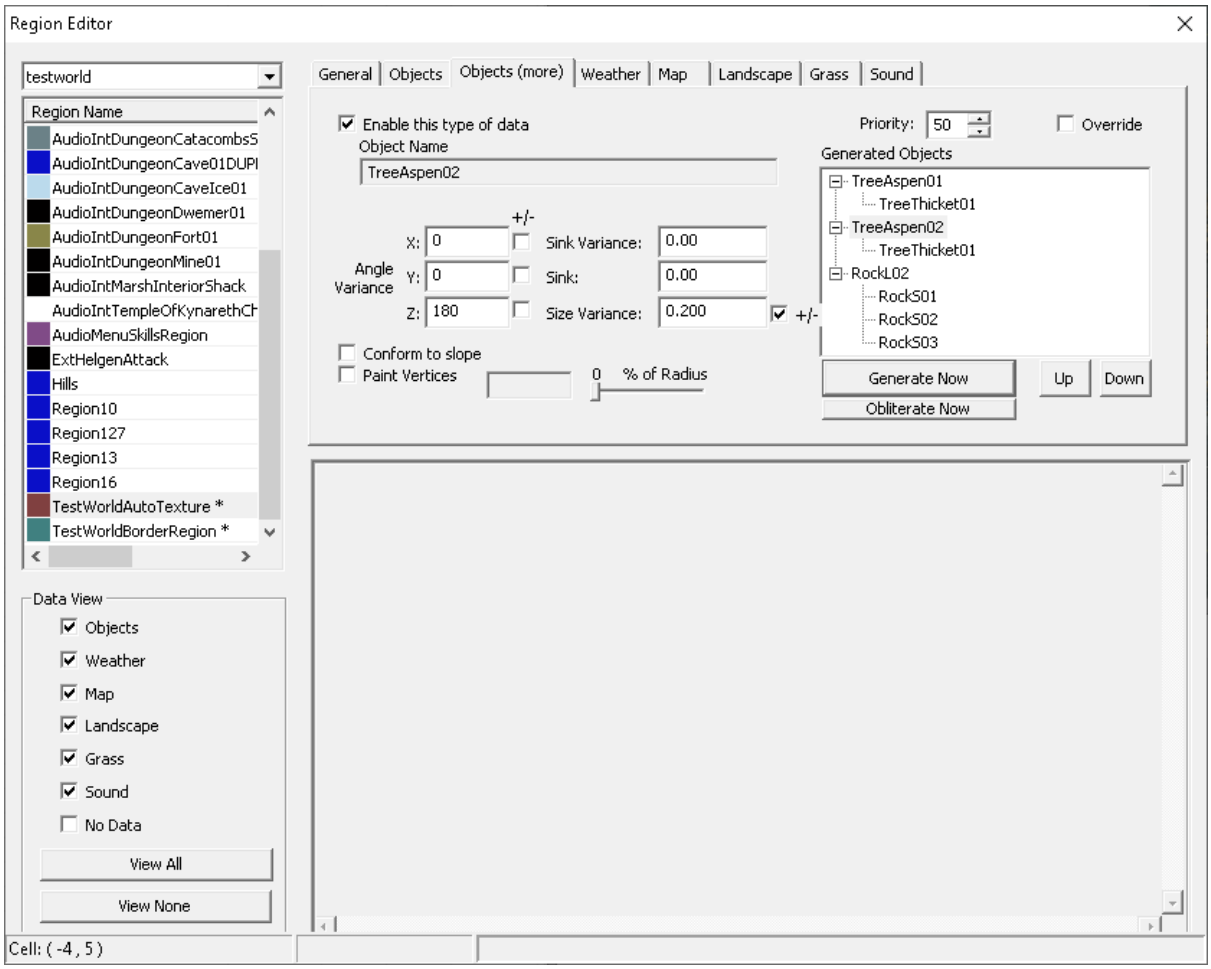


Figure 218 - Adding variance to the aspen trees and their thickets.

Interestingly, while the scale variance affected the aspen trees and the thickets, it does not appear that the angle variance had any effect. The angle variance is, however, still affecting RockL02 and the RockS01, RockS02 and RockS03 child objects.

Note: This might be a bug? If anyone can let me know, I would appreciate it.

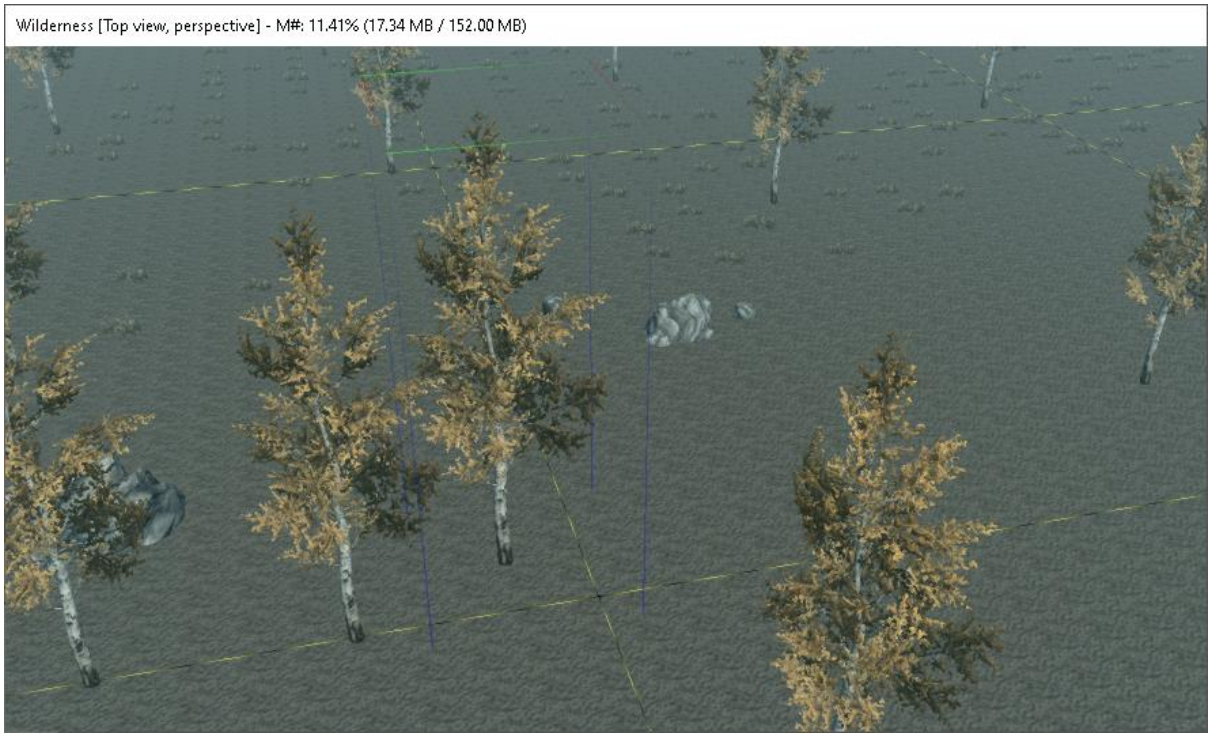


Figure 219 - Scale variance works on trees, but angle variance doesn't for some reason.

It also seems like Obliterate Now sometimes doesn't fully obliterate all parent trees...

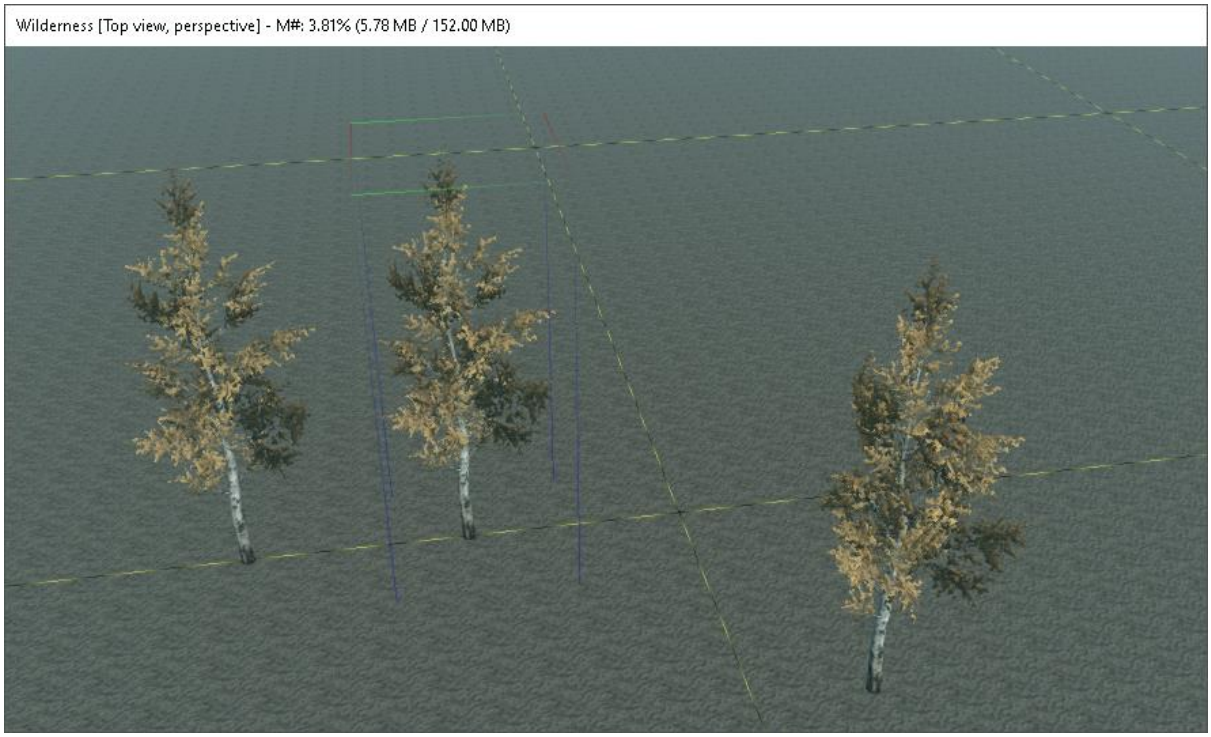


Figure 220 - Aspen trees not being fully obliterated.

Lastly, let's have a look at nesting textures below objects.

Nesting a texture below an object will cause that specific texture to be painted around instances of the parent object upon generation.

Important: Don't nest objects under textures.

To test this out I'm going to set the landscape texture LFallForestLeaves01 as a child of both aspen trees. I'm also going to paint LRocks01NoRocks beneath every instance of RockL02 by nesting it below RockL02.

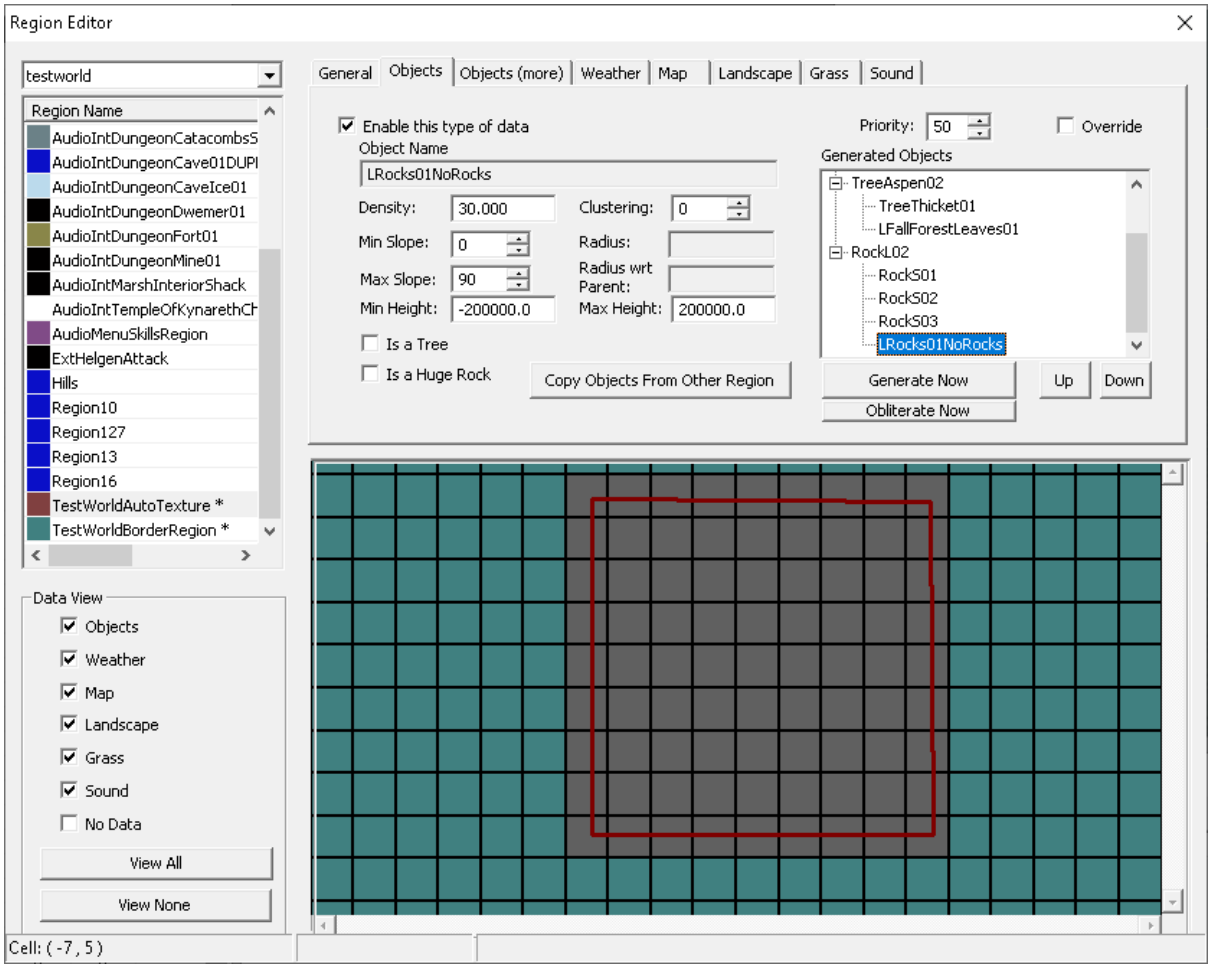


Figure 221 - Nesting landscape textures as children of parent objects.

After generating, our landscape looks like this:

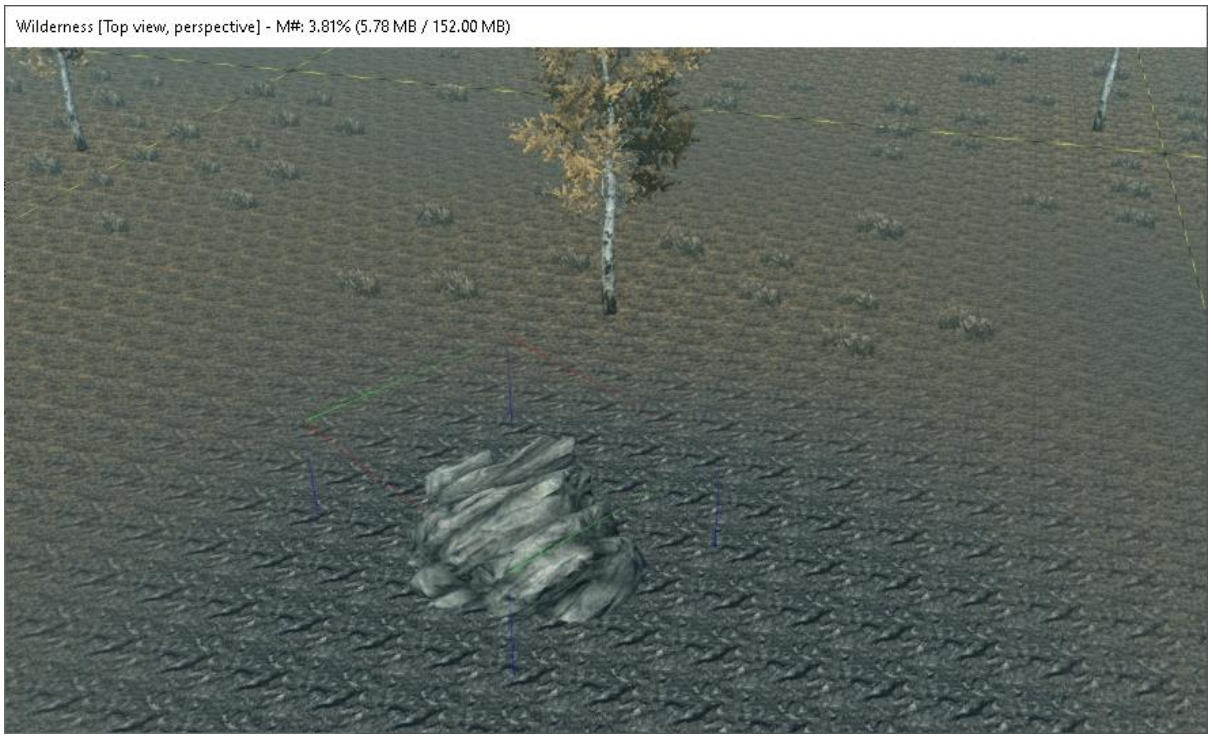


Figure 222 - Nested landscape texture.

We can control the size of the texture radius by adjusting the '% of Radius' slider.

By default the '% of Radius' slider is set to 0.

The percentage that the '% of Radius' slider is comparing against is the radius of the parent object. The closer to 100%, the smaller the radius of the texture.

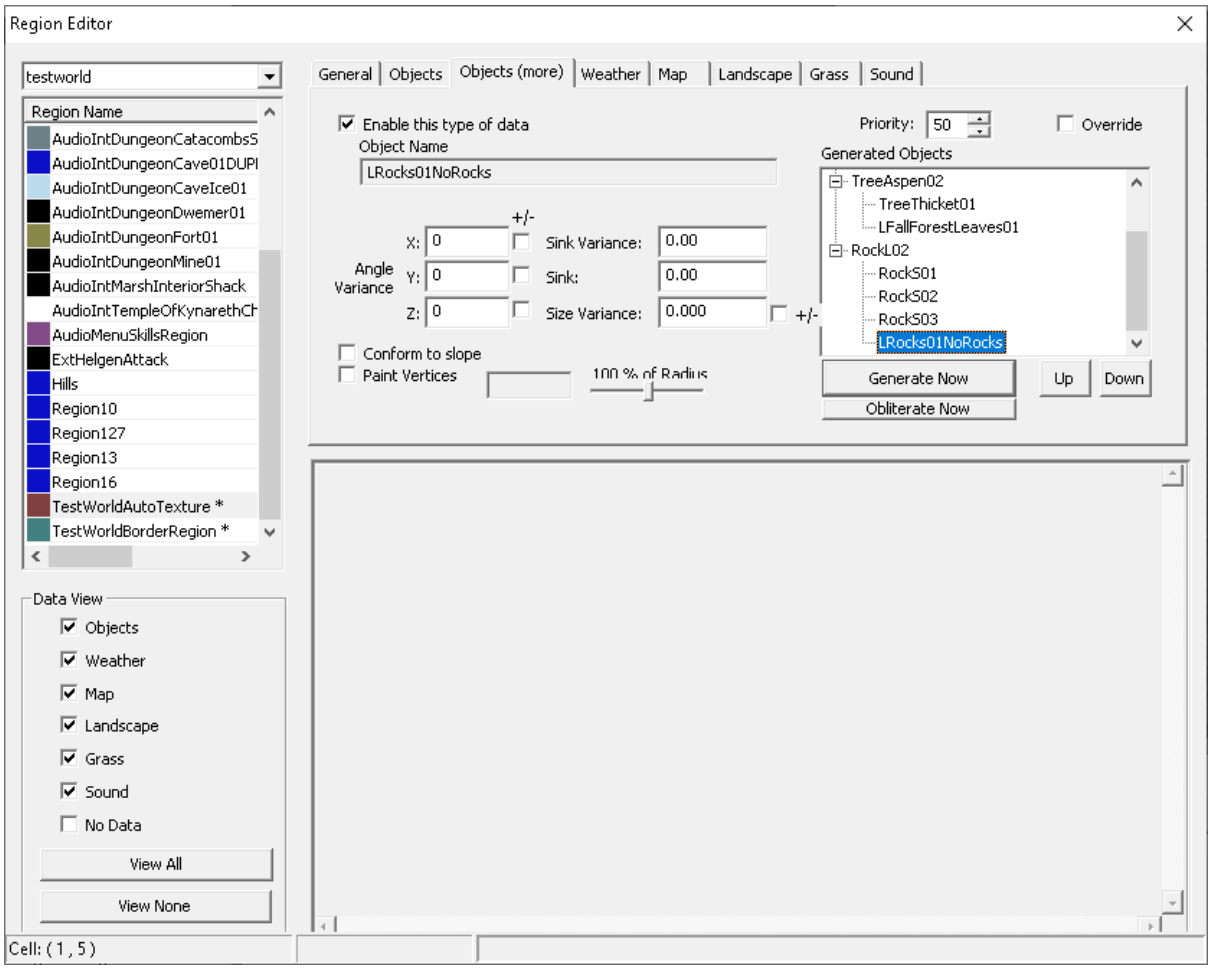


Figure 223 - % of Radius.

After setting the '% of Radius' slider to 100% for LFallForestLeaves01 and LRocks01NoRocks the textures beneath the parent objects no longer extend as far out.



Figure 224 - Smaller texture radius around parent objects.

Increasing this value above 100% will again increase the radius of the texture around the parent object.

Important: You can combine this technique of nesting textures below parent objects with landscape texture generation discussed at the start of this section, just make sure you stick to 6 different types of textures or less otherwise you're going to end up painting vertex colours.

As far as I can see, none of the base game regions of the Skyrim world space have any textures or objects set up to auto populate, though they might've removed them from the Region Editor after generation.

For more information on what each field does, refer to the Regions article on the Creation Kit wiki.

SETTING UP LOCATIONS

Locations are also used by the radiant quest system. They can be used to attach radiant markers and NPCs with radiant flags to a specific location.

Generally speaking, for each settlement, bandit camp, animal den, dungeon, etc. that we create, we'll need to create a location record for it and assign it to the interior and/or exterior cells relevant to that location.

In the Object Window, go to WorldData > Location.

To create a new location, right-click on one of the existing locations and select New.

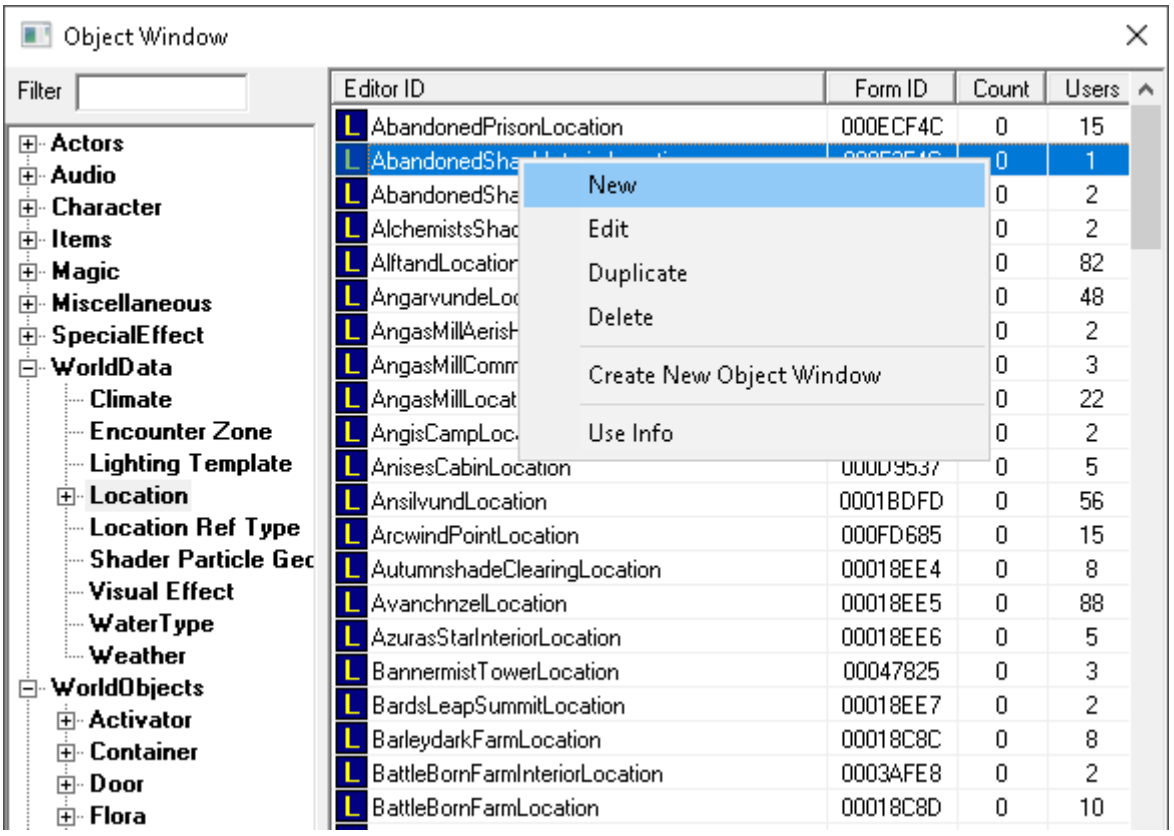


Figure 225 - Creating a new location.

Enter in the location ID. For my example, I'm going to use this location for a town named Stonehollow, so I'll set the ID to StonehollowLocation with the 'WT' prefix.

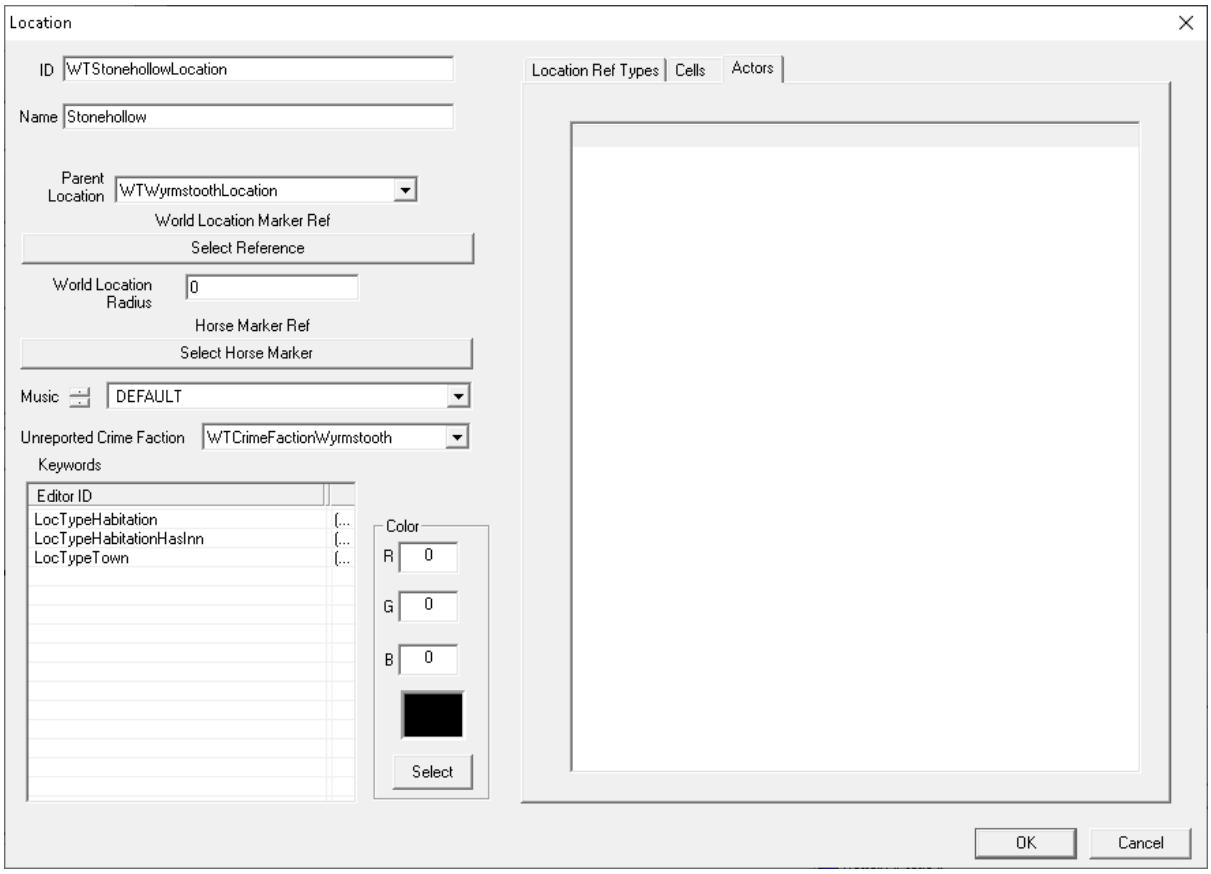


Figure 226 - Location properties.

I set the Name field to Stonehollow and the Unreported Crime Faction to WTCrimeFactionWyrmstooth which is the crime faction I use for the Imperial controlled locations on Wyrmstooth. For enemy-controlled dungeons you generally won't need to set an unreported crime faction.

Keywords are used by radiant quests and random encounters and are used to describe that location.

To add a keyword, right click in the keywords list and select Add.

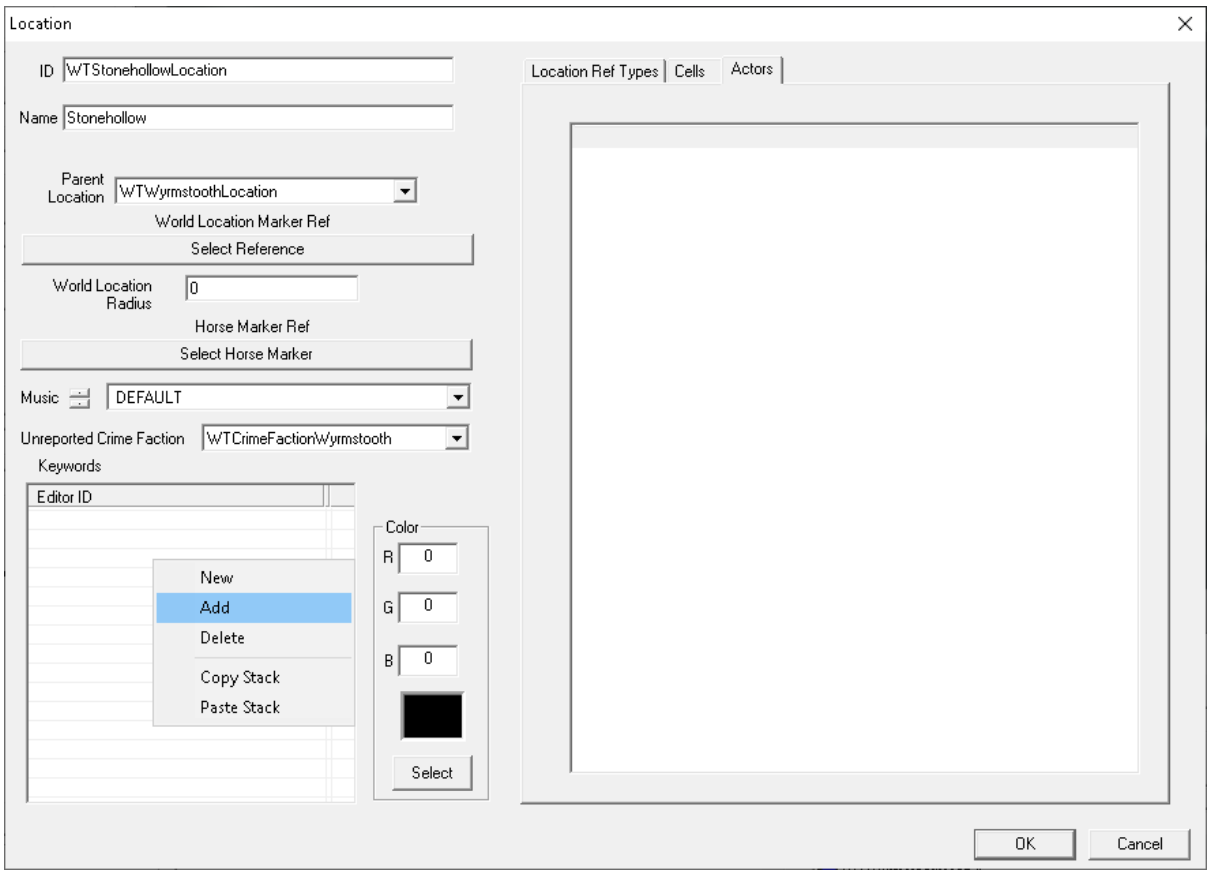


Figure 227 - Adding keywords.

Select the keyword and click OK to add it to the keywords list. Most keywords you'll want to assign are prefixed by 'LocType' so filter your search by that.

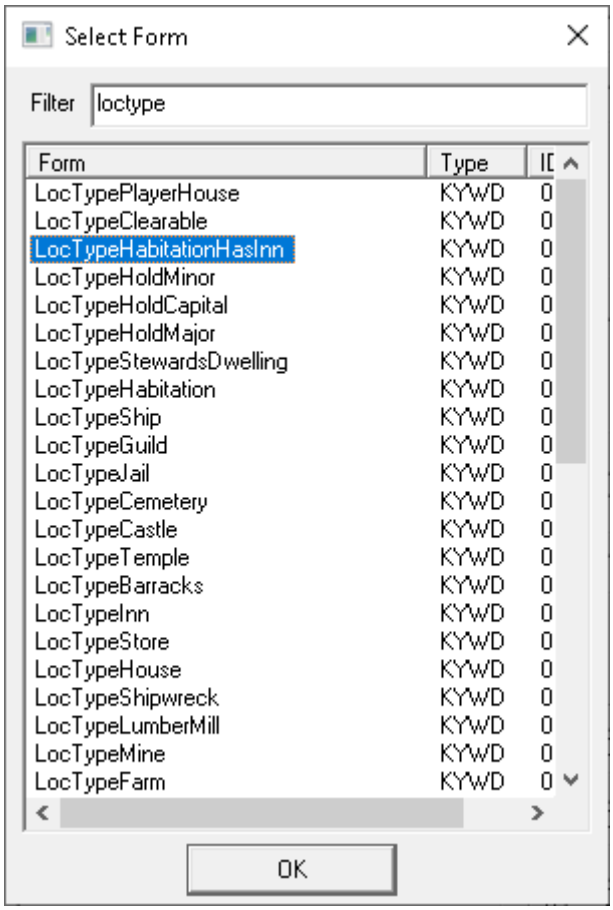


Figure 228 - Keywords list.

For my example, I added LocTypeHabitation, LocTypeHabitationHasInn and LocTypeTown. These keywords are used by most small settlements in the base game such as Riverwood.

I would recommend taking a look at the keywords added to locations similar to the one you're setting up to see what kind of keywords you may need to assign if you want the radiant quest system to use your new location.

Important: If you want your location to be clearable, add the LocTypeClearable keyword. Without this keyword, your location won't be flagged as 'cleared' on the map screen even if every boss NPC associated with that location has been defeated.

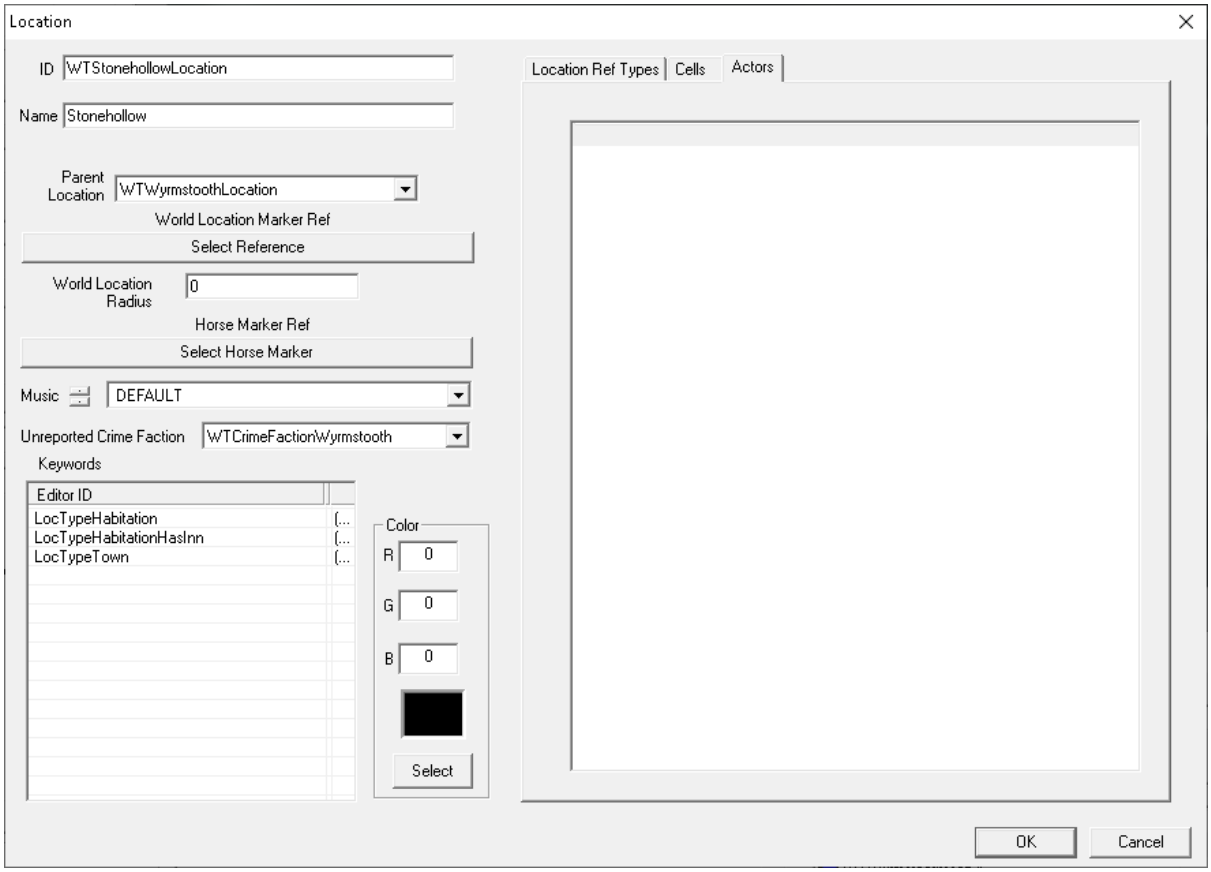


Figure 229 - Keywords added to location.

Click OK to exit out of the location properties.

Now to assign our new location to a cell.

Firstly, if you haven't already, you can rename a Wilderness cell by left-clicking on a cell you've highlighted or by pressing 'F2'.

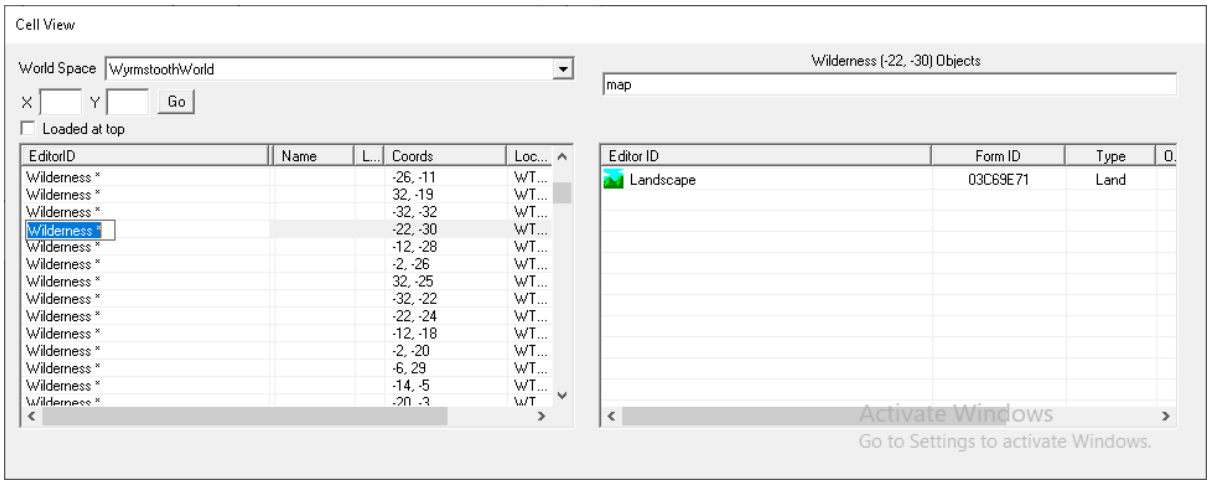


Figure 230 - Renaming a wilderness cell.

In the Cell View window, right-click on a cell you want to assign to the new location and select Edit.

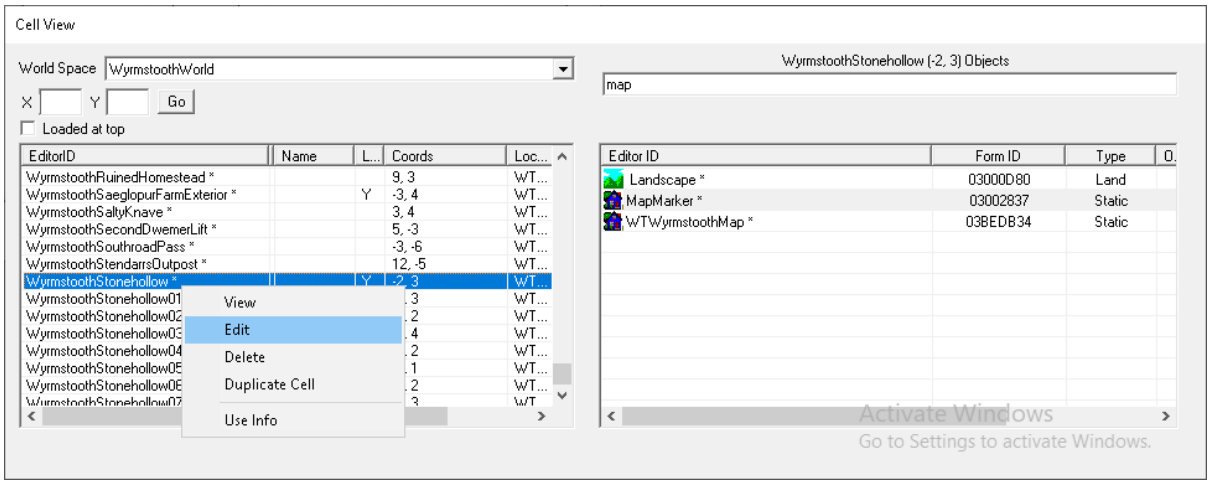


Figure 231 - Editing a cell.

You can specify a location in the Location drop-down.

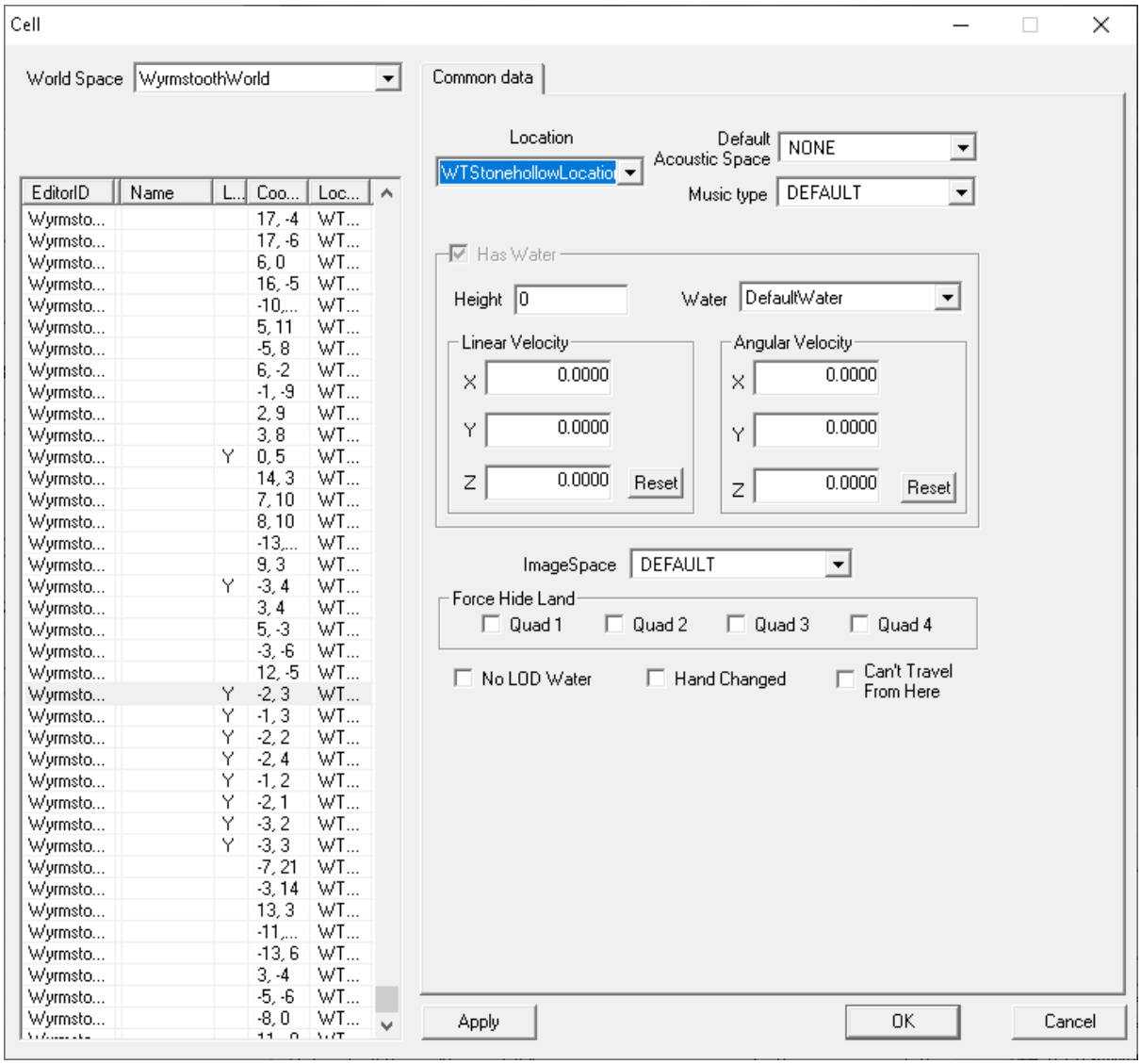


Figure 232 - Location assigned to cell.

If your location spans multiple cells, assign your location to those cells as well. You can assign a location to multiple cells including exterior world space cells and interior cells.

Click OK to close out of the Cell properties.

Locations can be nested by setting the Parent Location.

For example, Hulgard and Svenja's house is part of the Stonehollow settlement, so the Parent Location drop-down is set to WTStonehollowLocation. Internally, WTHulgardSvenjaHouseLocation will be treated as part of WTStonehollowLocation.

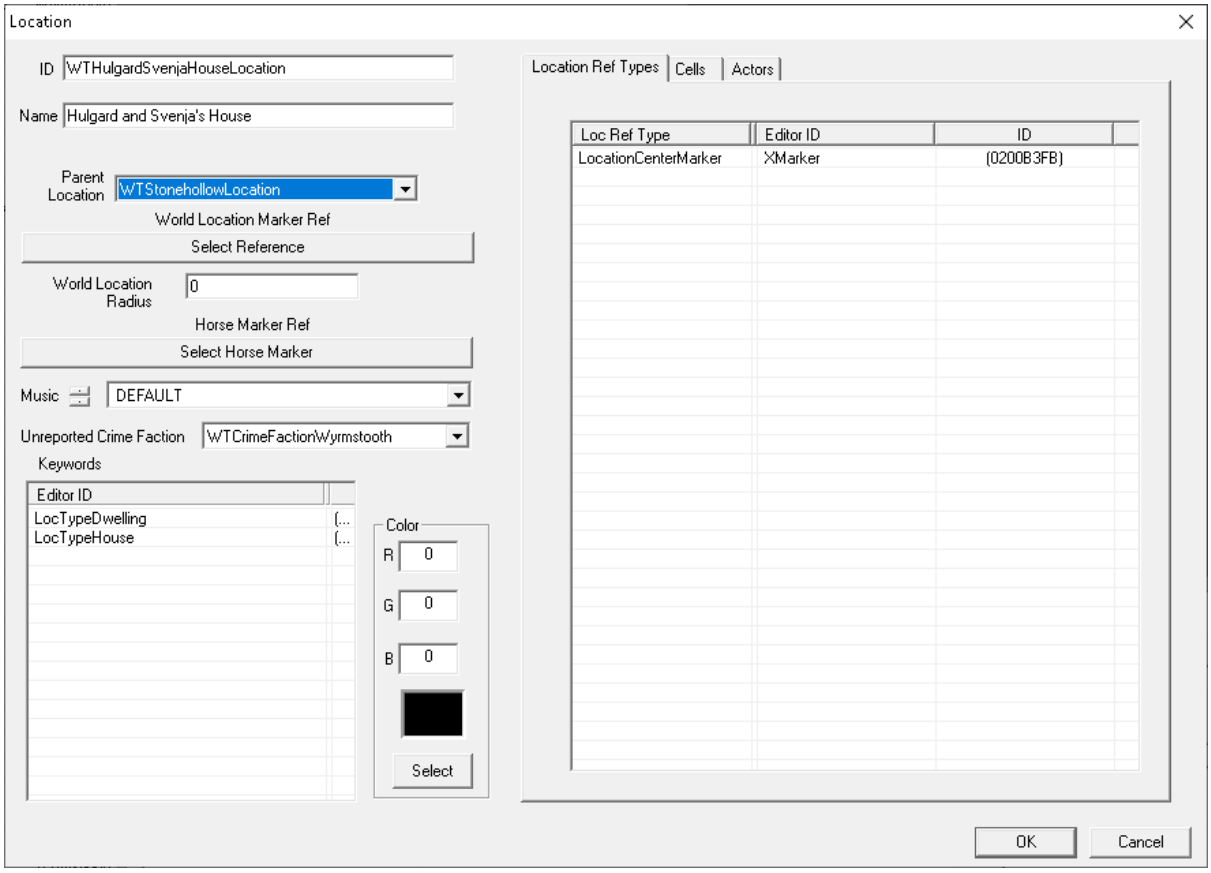


Figure 233 - Hulgard and Svenja's House location.

If you go back to the Location properties for the location you just created, you should now see a list of assigned cells in the Cells tab and a list of any NPCs in those cells in the Actors tab.

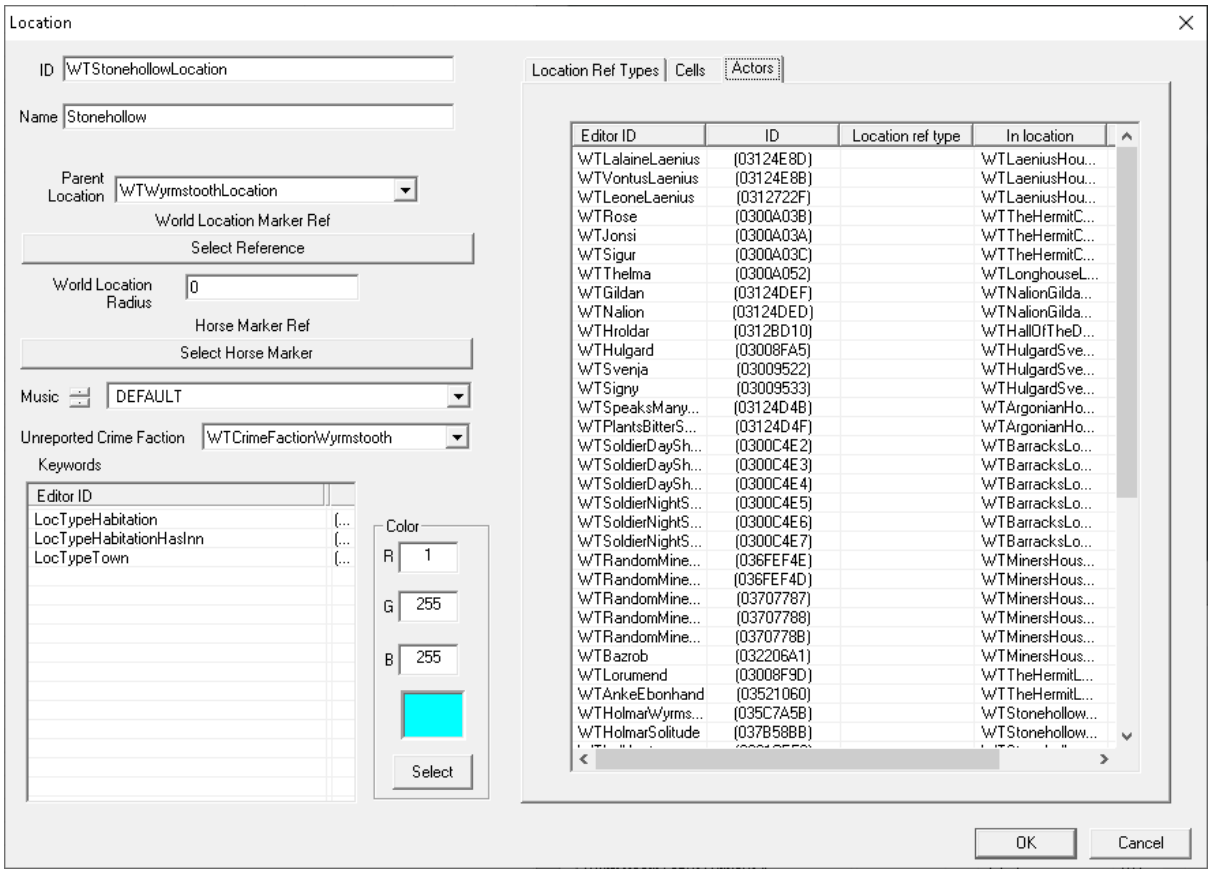


Figure 234 - Tabs are now populated with data from assigned cells.

SETTING UP ENCOUNTER ZONES

Encounter Zones are generally used to define the minimum and maximum levels of levelled list-generated enemies at a specific location. They can also be used to set a location to never respawn and to automatically set NPC or faction ownership of all items and containers in cells assigned to the location specified.

I created encounter zones for each clearable location on Wyrmstooth, such as dungeons, caves, or enemy controlled camps.

To create a new Encounter Zone, in the Object Window go to WorldData > Encounter Zone.

Right-click on one of the existing encounter zones and select New.

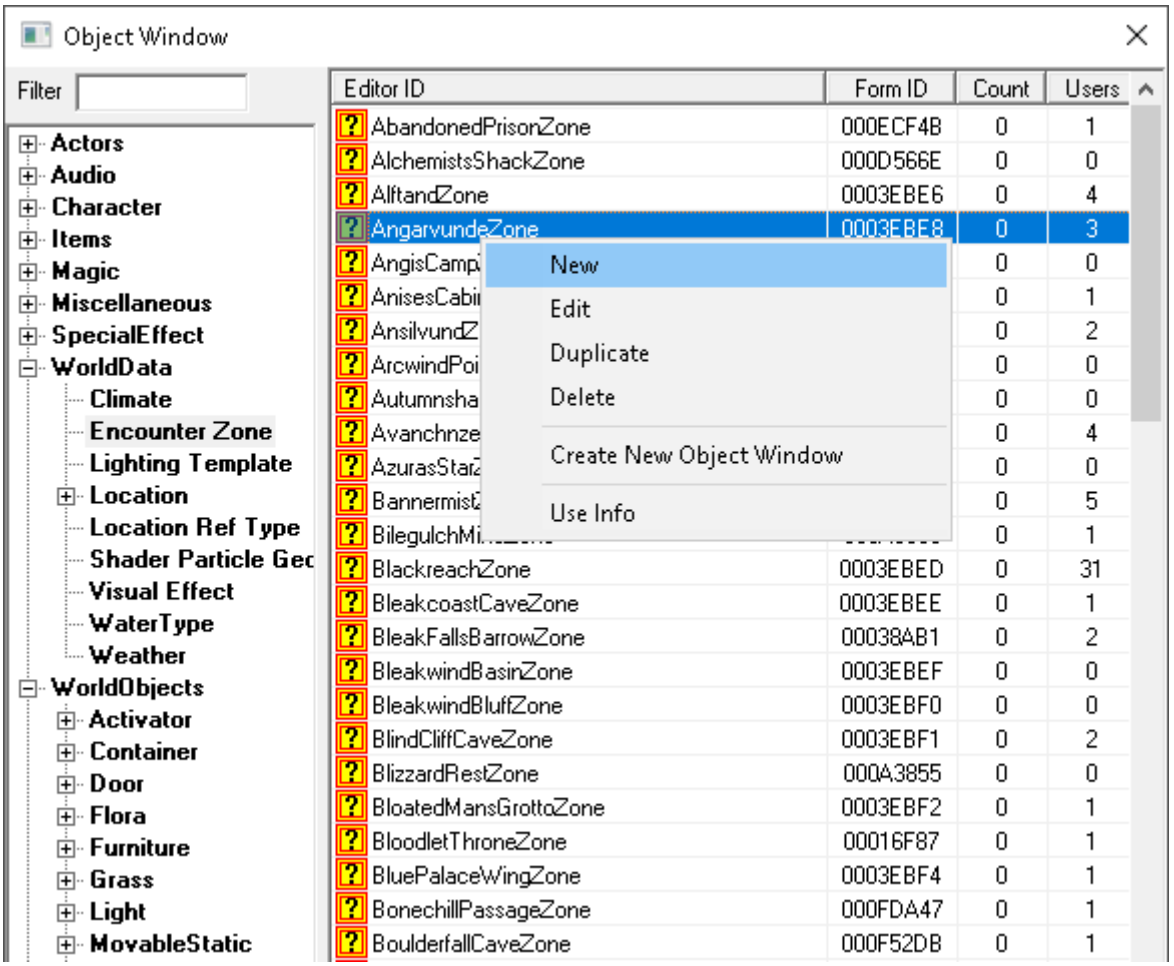


Figure 235 - Creating a new encounter zone.

Set the encounter zone ID. For my example I'll be creating an encounter zone for a cave called Coldwave Crescent.

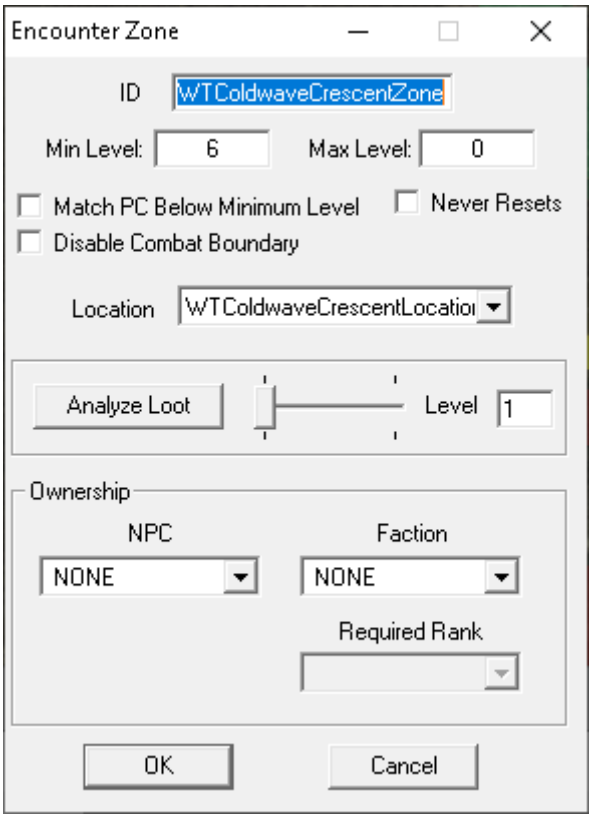


Figure 236 - Encounter zone settings.

I set the Minimum Level to 6. This will affect enemies spawned in using a levelled list.

If you wanted to increase the difficulty of the enemies spawned in a dungeon, you could increase the Min Level. Alternatively, if you want to prevent a dungeon from becoming too difficult, set a Max Level. Bleak Falls Barrow for example has a Max Level of 20. Draugr deathlords start spawning in at around player level 30 meaning you should never see a deathlord spawn in Bleak Falls Barrow due to this Max Level cap.

Set the Location drop down to the location you want to use this encounter zone for.

If you wanted to set ownership on all items and containers in the cells that use this encounter zone, you can specify that under the Ownership section. For my example, I'm going to keep these set to NONE as this cave will only contain hostile NPCs.

Ticking 'Never Resets' will prevent any cells that this encounter zone is assigned to from resetting. These kinds of encounter zones are assigned to player homes to prevent containers from resetting.

Diasble Combat Boundary allows hostile NPCs to leave this area to pursue the player if they are in combat with the player.

Match PC Below Minimum Level allows enemies spawned in using a levelled list to match the player's level if the player's level is below the Min Level specified.

Let's look at assigning this encounter zone to a cell next.

In the Cell View window, right-click on the cell you want to add the encounter zone to and select Edit.

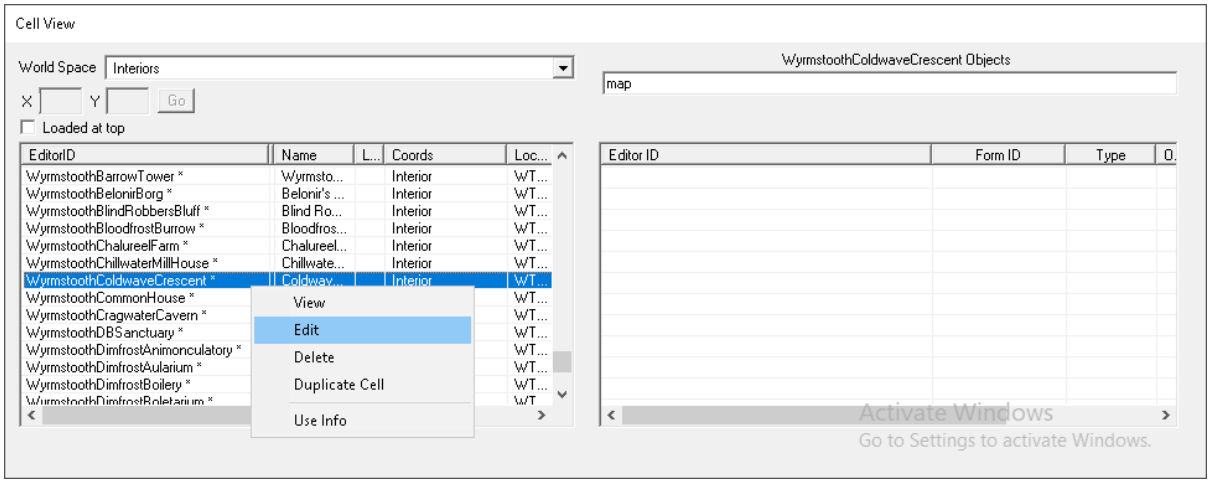


Figure 237 - Editing a cell.

Go to the Interior Data tab and set the Encounter Zone drop down to the encounter zone you created.

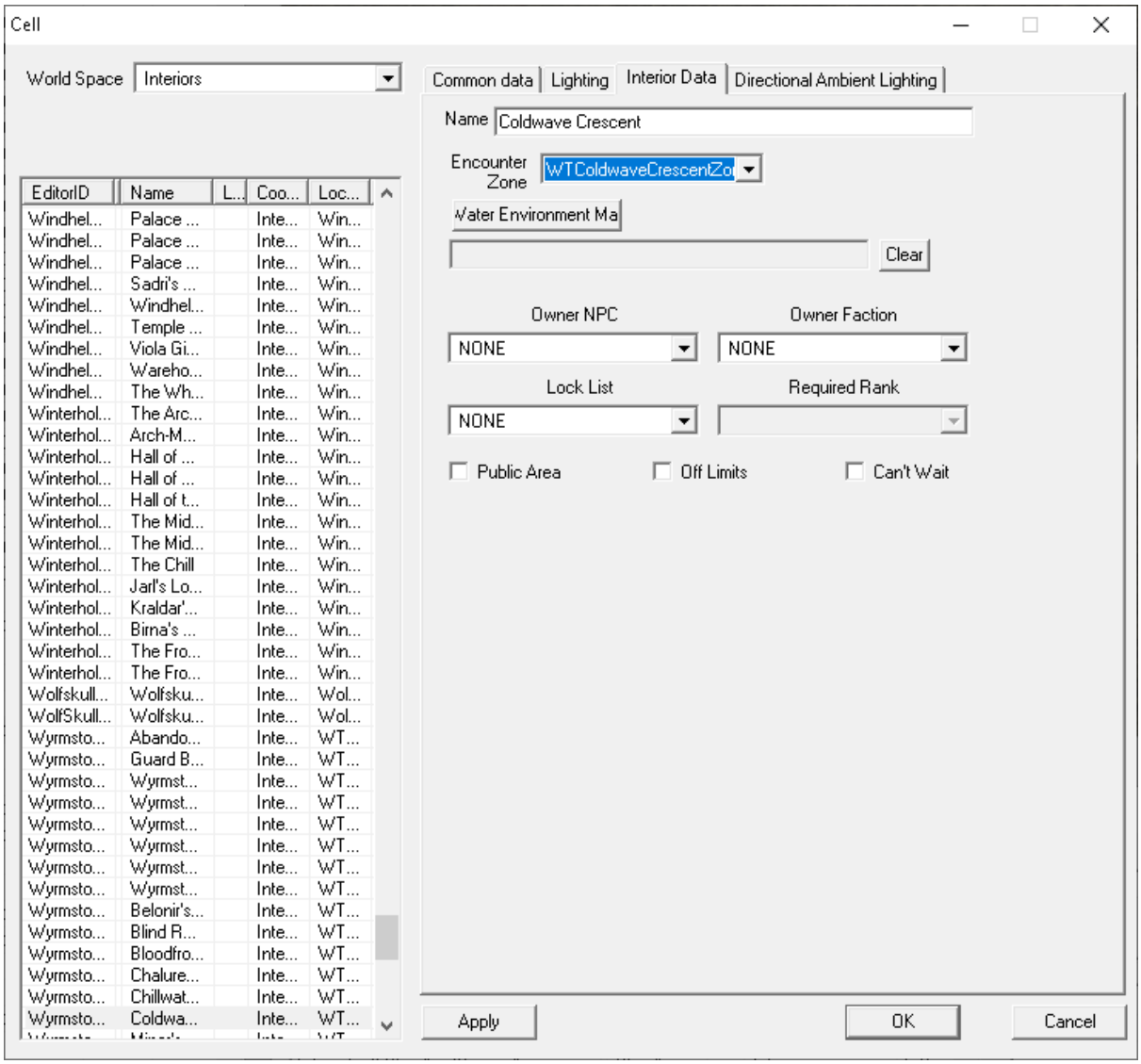


Figure 238 - Adding an encounter zone to a cell.

Click OK to close out of Cell properties.

For more information see the Encounter Zone article on the Creation Kit wiki.

ADDING NAVMESH

Navmesh is used to determine where NPCs can and cannot go. It also determines where NPCs can find cover during combat and allows NPCs to traverse between doors in interior cells and doors in exterior world space cells.

Simply put, any surface that an NPC should be able to walk on or swim over needs to be covered with navmesh.

Once navmesh has been added to a cell, it needs to be finalized before NPCs can use it. If you make any changes to any parts of the navmesh after finalization, you will need to finalize it again.

Navmesh is made out of vertices, edges and faces.

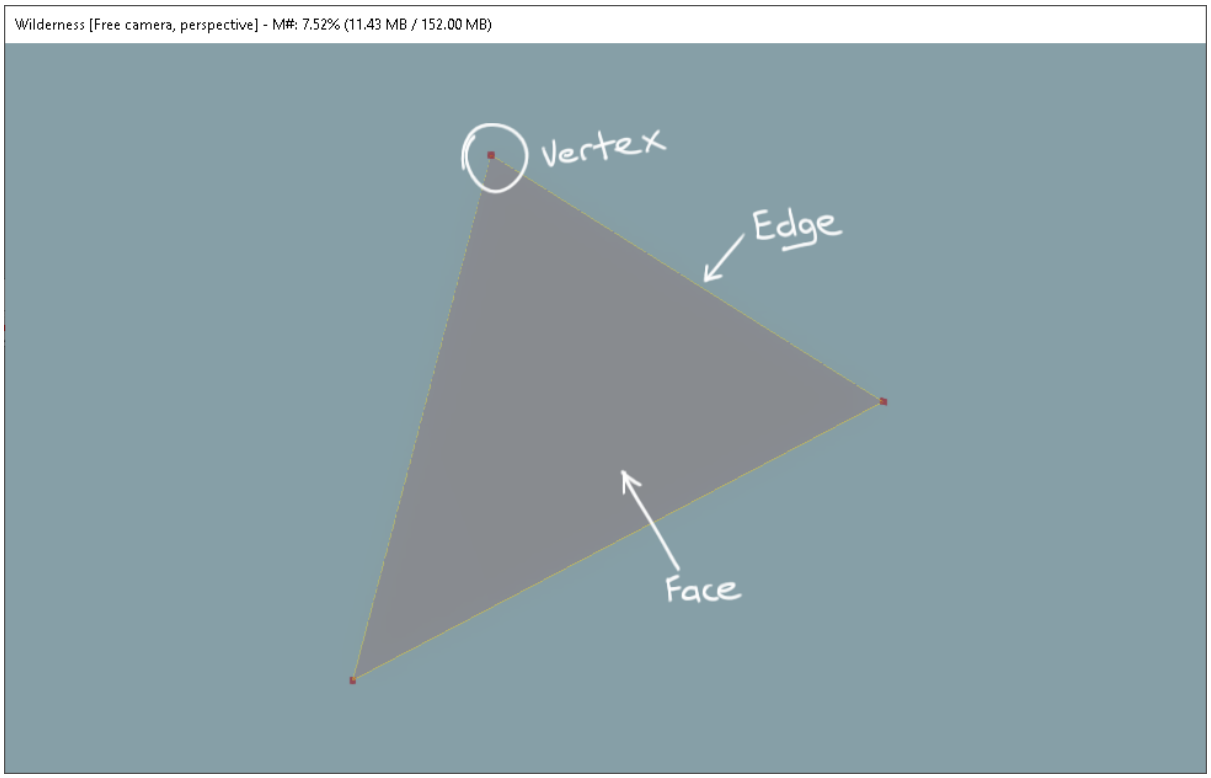


Figure 239 - Composition of a navmesh triangle.

To enter navmesh mode, click on the Navmesh button in the toolbar.

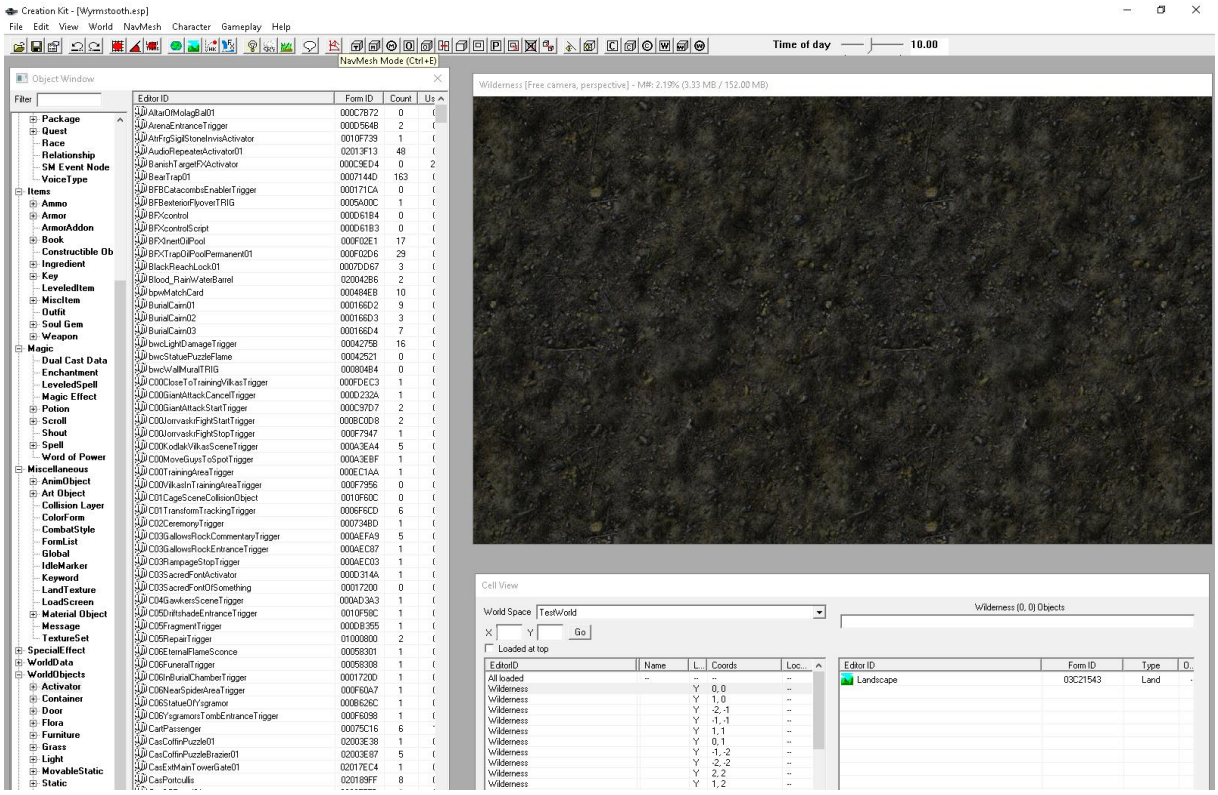


Figure 240 - Navmesh mode button.

When you enter navmesh mode, the navmesh toolbar will appear.



Figure 241 - Navmesh toolbar.

Navmesh is made of vertices that join together to form triangles. To begin manually creating navmesh, hold down CTRL and begin right-clicking in places where you want to place vertices.



Figure 242 - A navmesh triangle.

Once you've placed 3 vertices down they'll join together automatically while you're holding down CTRL.

You can move existing vertices, edges or faces around after placing them by left-click dragging the currently selected vertices, edges or faces.

Selected vertices, edges or faces are highlighted in green. To deselect, click on a surface that isn't covered by navmesh.

You can select multiple vertices, edges or faces by holding down CTRL.

To change the draw mode, press W. You can toggle between non-transparent, semi-transparent, and navmesh-only views. Non-transparent mode won't show navmesh occluded by other objects, semi-transparent mode will allow you to see the navmesh through the level, and navmesh only mode will hide everything else except navmesh.

For a list of navmesh shortcut keys, refer to the Navmesh Cheat Sheet.

If you keep right-clicking with CTRL held down, more triangles will be created automatically as you continue adding vertices.



Figure 243 - Adding more triangles to navmesh.

You can join three separate vertices together to form a triangle by highlighting them all and pressing 'A'.

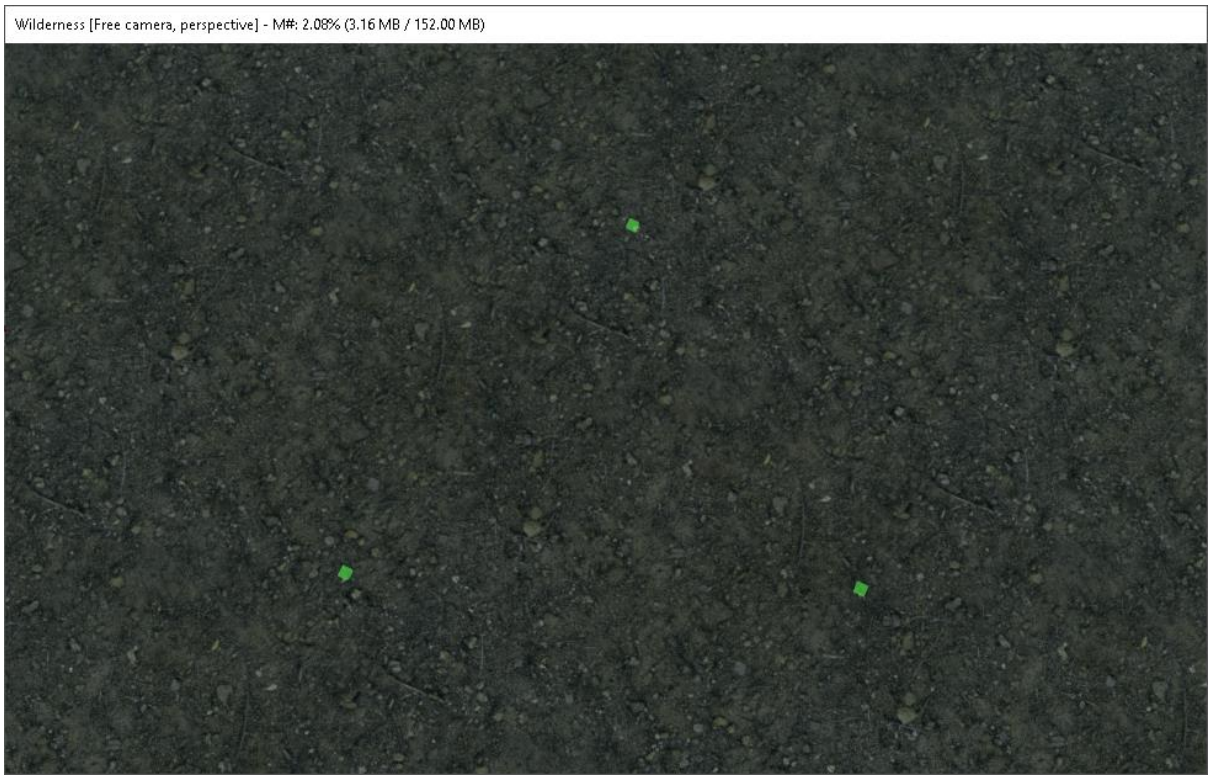


Figure 244 - Joining separate vertices.

You can also join four separate vertices together by selecting them and pressing 'A' to join them together.



Figure 245 - Joining four vertices.

Important: Don't overlap navmesh like this:



Figure 246 - An example of bad overlapping navmesh. Avoid doing this!

There's nothing wrong with having a navmeshed pathway like a bridge above navmeshed terrain, but navmesh that's closely overlapping as per the screenshot above will result in errors and should be avoided.

If you need to select an entire navmesh segment, click on one triangle and press 'F'. This will select all adjoining faces.



Figure 247 - A navmesh island selected.

Note: You can move navmesh around like an object by left-click dragging highlighted vertices, edges or faces, but in most cases you won't want to do that as navmesh will need to conform around objects and to your landscape.

To delete the currently selected vertices, edges or faces, press Delete.

Here's an example of what navmesh around an object should look like.



Figure 248 - Navmesh around a rock.

When finalizing navmesh, sometimes extra vertices may be added automatically, splitting up large triangles, but try and be mindful of the complexity of your navmesh topology as it will affect the overall size of your mod.

Important: Navmesh should conform to the surface that an NPC will be walking on.

Also, make sure vertices are not placed in the air! This can happen if you try and navmesh over fog volumes. To bring a vertex, edge or face to the ground, select it and press 'F' to make it fall to the nearest surface. You can also manually raise or lower a vertex, edge or face by holding down Z as you left-click drag it to constrain it to the Z axis.

When navmeshing a road, it's important to keep the vertices to either side of the pathway. Having the topology of the navmesh flow with the road should help prevent NPCs tracking off the road when traveling to a particular location.

Here's an example of what a navmeshed road should look like:

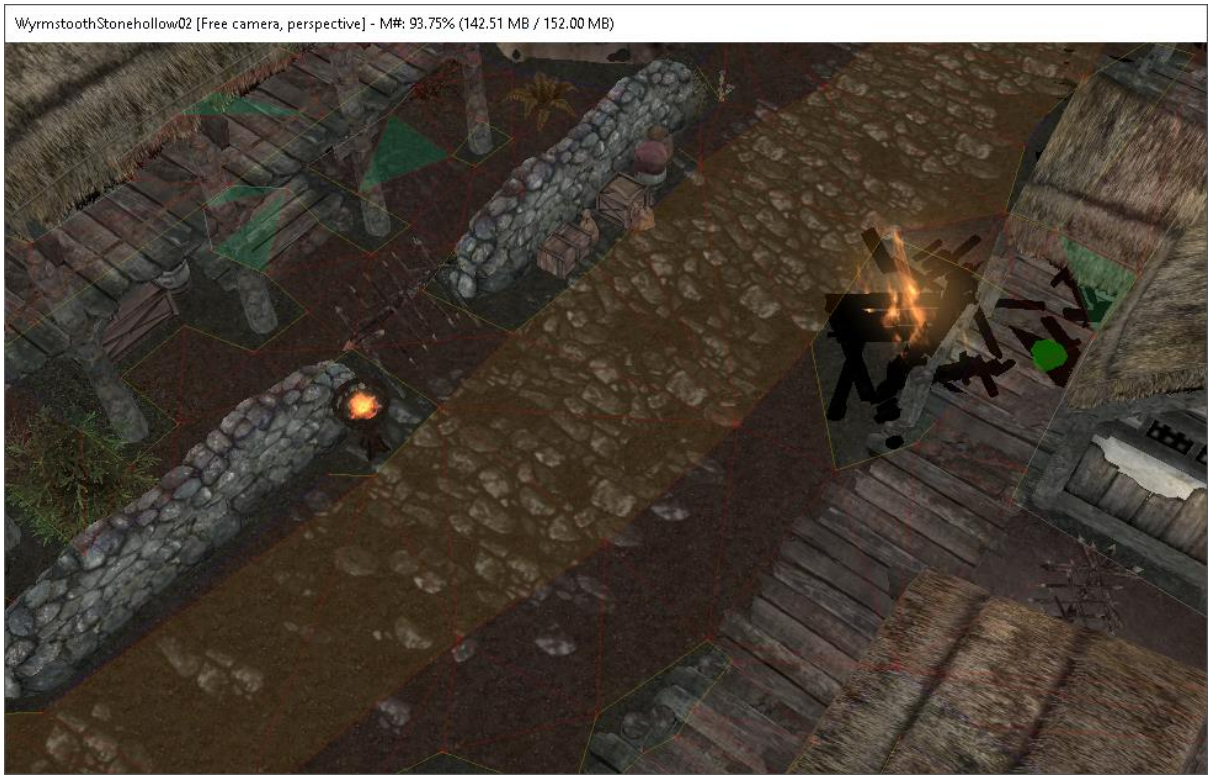


Figure 249 - An example of a navmeshed road.

The yellow navmesh over the road indicates a preferred pathway. You can turn a regular navmesh triangle into a preferred pathway triangle by selecting it and pressing 'P'. The colour of the triangle should turn from red to yellow.

NPCs traveling from one location to another will try and do so via the nearest preferred pathway if one exists.

To allow NPCs to travel from an interior cell into an exterior world space cell via door markers, you'll need to make sure a navmesh triangle is placed under each door marker. When you finalize the navmesh, the triangle under the door marker should turn green as per the screenshot below:



Figure 250 - A green navmesh triangle under a door marker.

Important: You'll need to have a green navmesh triangle under both door markers.

Navmesh can't across cell borders, nor can you join vertices together from different cells. To join the navmesh from two separate cells together you'll need to join navmesh vertices along both sides of the border before finalizing as per the screenshot below:

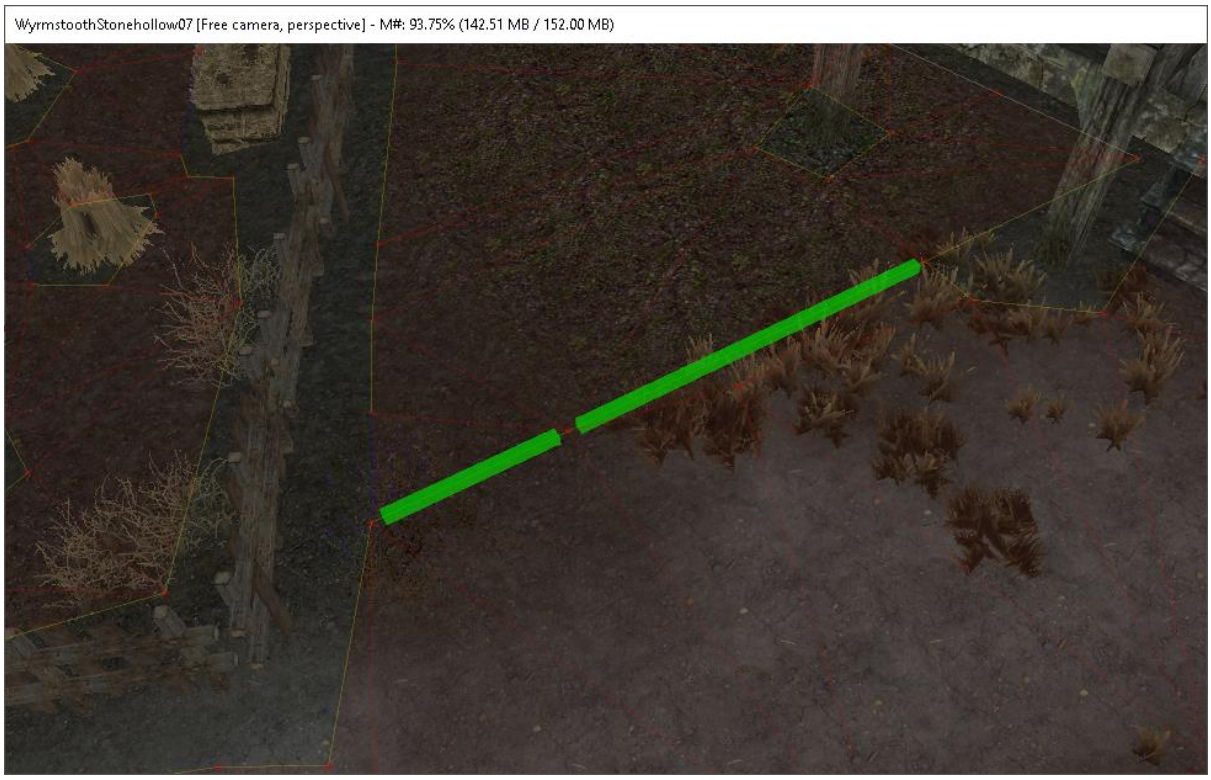


Figure 251 - Joining navmesh between world space cells.

Navmesh edges that have been successfully joined together along a cell border will be marked in bright green after you finalize.

Press 'B' to show cell borders in the render window.

Here's a closer look at the navmesh at the cell border:



Figure 252 - Border vertices pulled apart.

In the screenshot above I pulled apart the vertices along the border to demonstrate how the vertices of the navmesh vertices need to be closely overlapping.

Important: If the vertices are too far apart, the navmesh edge along the border won't turn bright green when you finalize and NPCs won't be able to cross that point from one cell to another.



Figure 253 - Border vertices joined back together again.

In the screenshot above, I moved the vertex back to where it was.

Navmesh beneath the ocean or a water plane should be marked as water triangles. To do so, select the faces beneath the water and press 'O' to turn them into water triangles.

Water triangles appear blue in the render window.

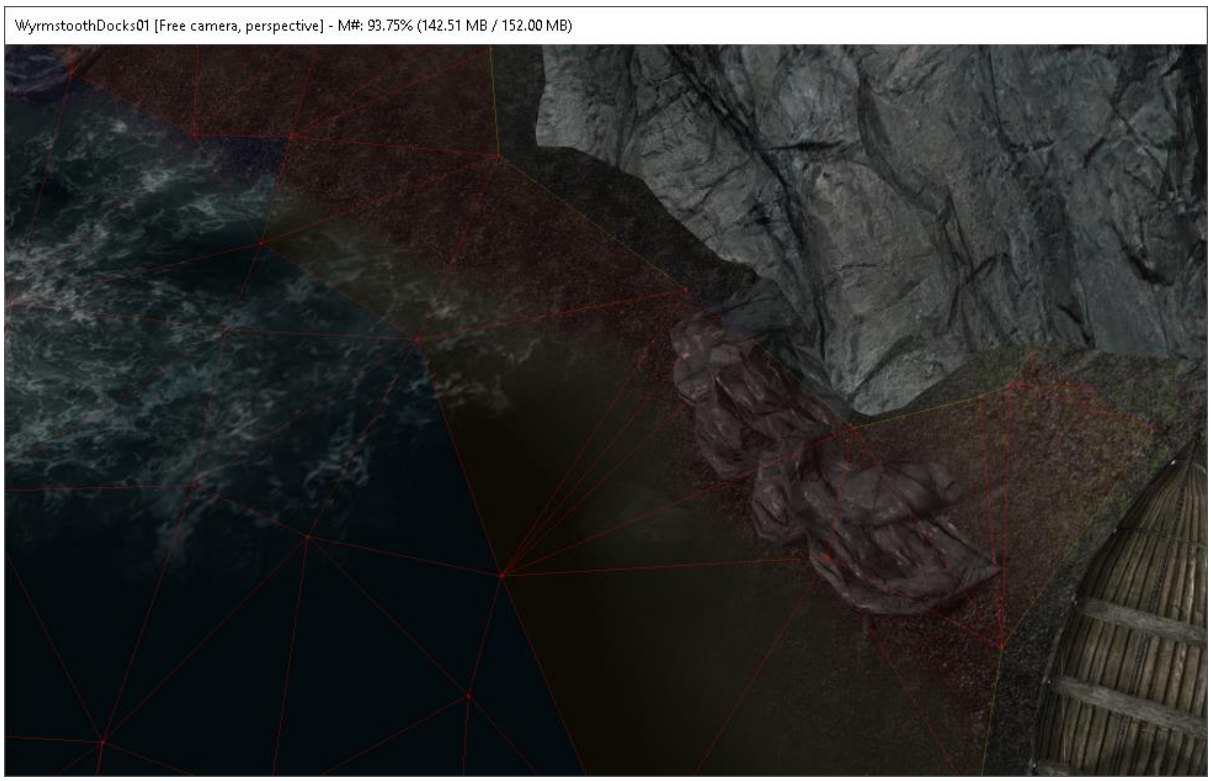


Figure 254 - Water triangles at a coastline.

Terrestrial NPCs should *try* to avoid pathing through water triangles when fleeing or pursuing an enemy.

Creatures that live in the water, such as slaughterfish, should also avoid crossing into non-water triangles.

Press the Find Cover Edges button in the navmesh toolbar to automatically mark navmesh edges that may provide NPCs with cover during combat.



Figure 255 - Navmesh cover.

Navmesh cover will be marked in blue. Edges near walls and fences are usually automatically flagged as cover edges.

Note: You can manually define which edges are cover edges. Click on the navmesh edge to select it then press 'E' to bring up Edge Cover properties.

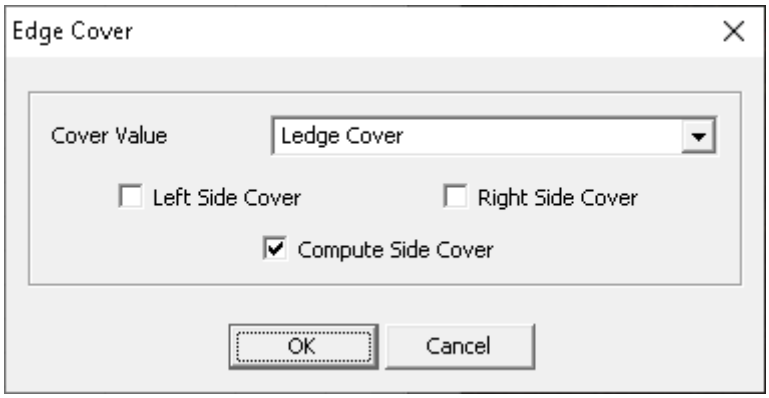


Figure 256 - Manually defining cover.

If you can't select a navmesh edge, press 'G' to toggle edge selection on or off.

Before finalizing, it's important to check for errors. You can do so by going to Navmesh > Select Triangle By Index. This will check the current cell for navmesh errors.

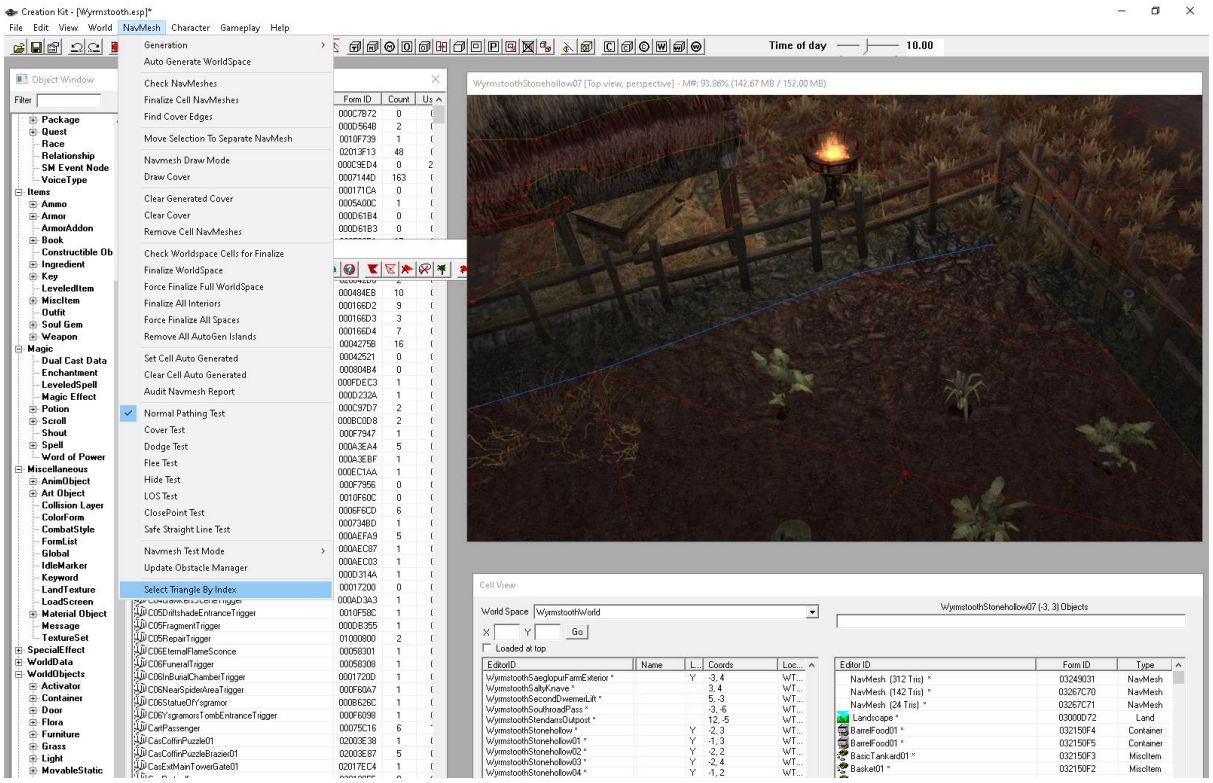


Figure 257 - Select Triangle By Index.

Click 'Check NavMesh' to check for navmesh errors.



Figure 258 - Select triangle.

If a navmesh error is found, the offending triangle will be highlighted automatically and shown in the render window.

You'll need to fix the error by hand, usually by deleting the bad triangle and recreating it. Sometimes you'll also need to remove some of the triangles around it too and rework the surrounding topology if the error persists.

Once the bad triangle has been fixed, click Next Warning and continue until no more warnings are found.



Figure 259 - No Warnings Found.

Once all navmesh errors have been corrected, you'll see 'No Warnings Found'.

When the Creation Kit loads a mod, any navmesh errors found will be listed in the EditorWarnings.txt file, so it's important to check that file after loading your mod in the Creation Kit whenever you're working on navmesh.

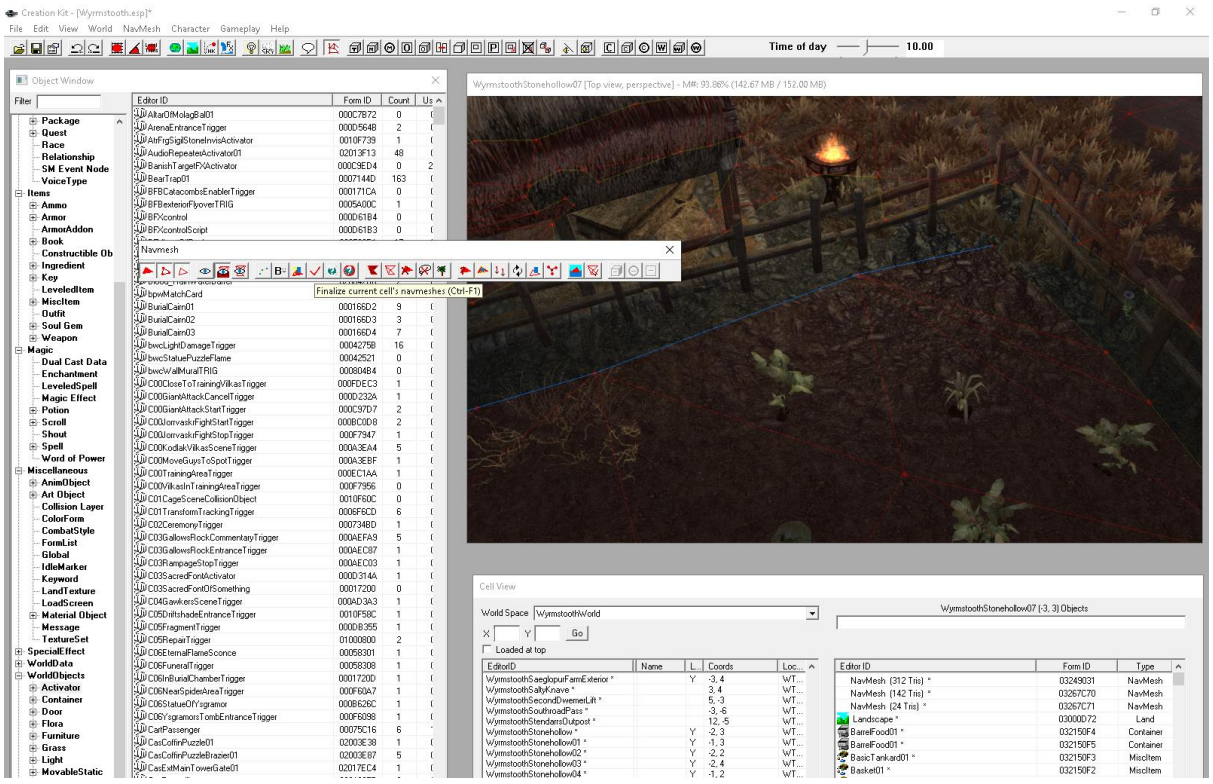


Figure 260 - Finalizing navmesh.

To finalize navmesh, click on the Finalize Cell button in the navmesh toolbar.

Important: There are several methods to automatically generate navmesh for a specific cell or even an entire world space, but to be honest I would actually recommend just navmeshing by hand despite how incredibly tedious that might sound.

Automatic navmesh generation tends to create a lot of 'navmesh islands' which are sections of navmesh that aren't joined to anything, and you'll need to spend time deleting them or joining them together so NPCs don't end up getting stuck if they happen to somehow enter one.

Automatic generation also tends to create a lot of inefficient sunburst vertices. A sunburst vertex is a single vertex that a bunch of navmesh edges connect to as per the screenshot below. You should delete these and retopologize your navmesh manually.



Figure 261 - An example of an inefficient sunburst vertex.

The topology of your navmesh in open areas of your world space should be a bit more evenly distributed like this instead:

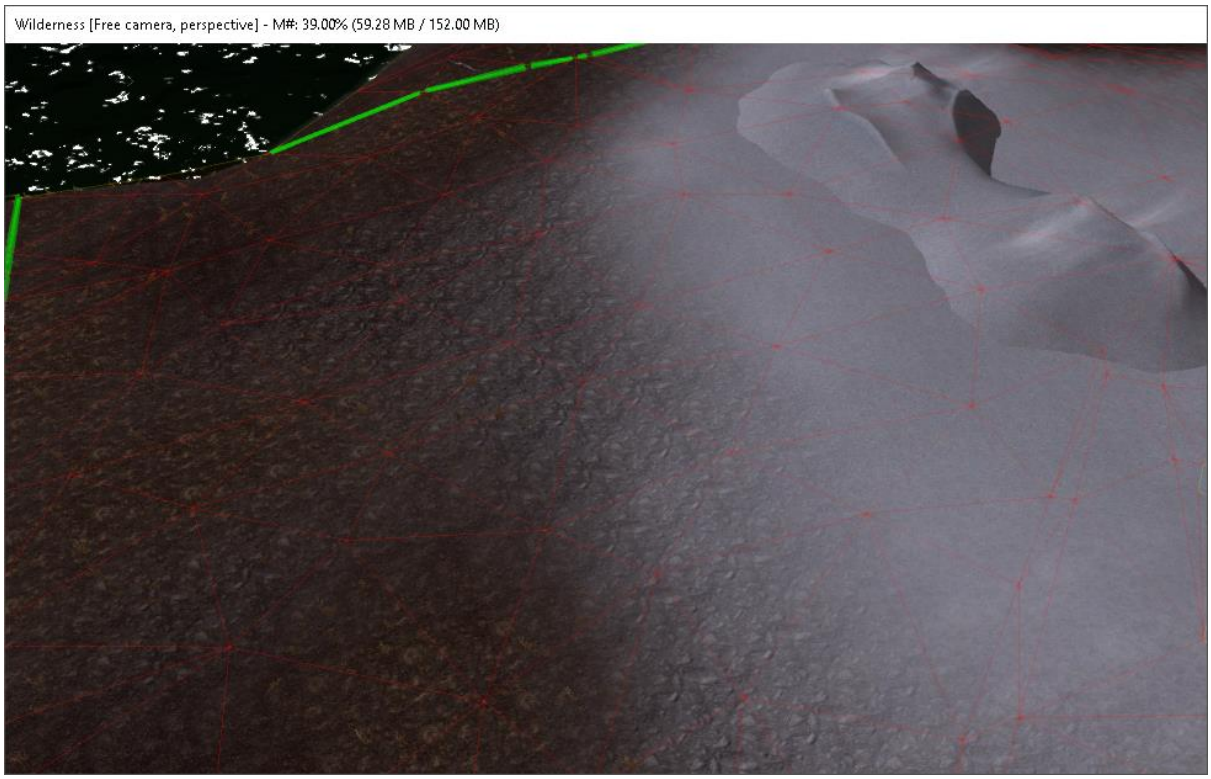


Figure 262 - An example of slightly better navmeshing.

While Auto Generate World Space does have a recast generation option, it lacks the same settings that Recast-Based-Generation has to control where exactly navmesh is automatically placed. The trade-off however is that you will need to run this manually in each cell of your world space that requires navmesh.

Note: If you really need to automatically generate navmesh, I'd recommend using Recast-Based-Generation on a cell-by-cell basis rather than Auto Generate World Space.

Go to NavMesh > Generation > Recast-Based-Generation.

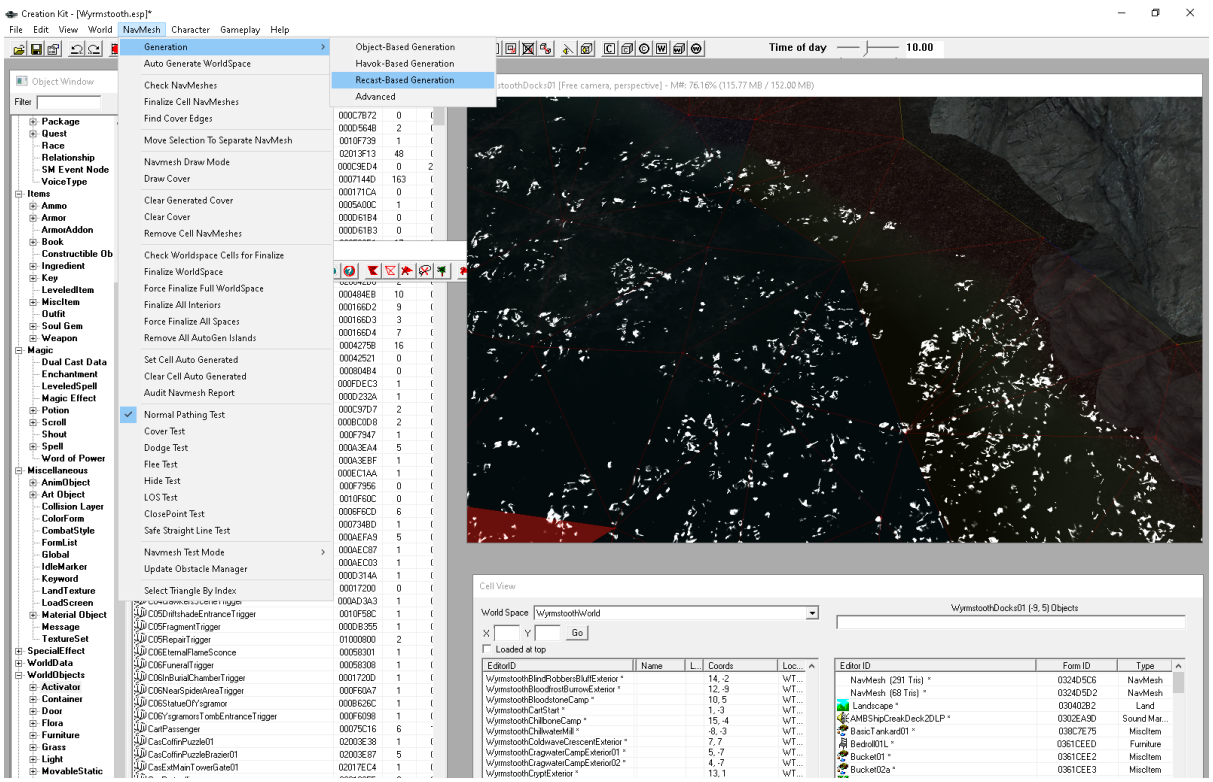


Figure 263 - Recast Based Navigation menu.

Keep the settings at their default, but change Agent Radius to 8.0000 before clicking OK. If too many small navmesh islands are created, try setting the Agent Radius back to 32.0000.

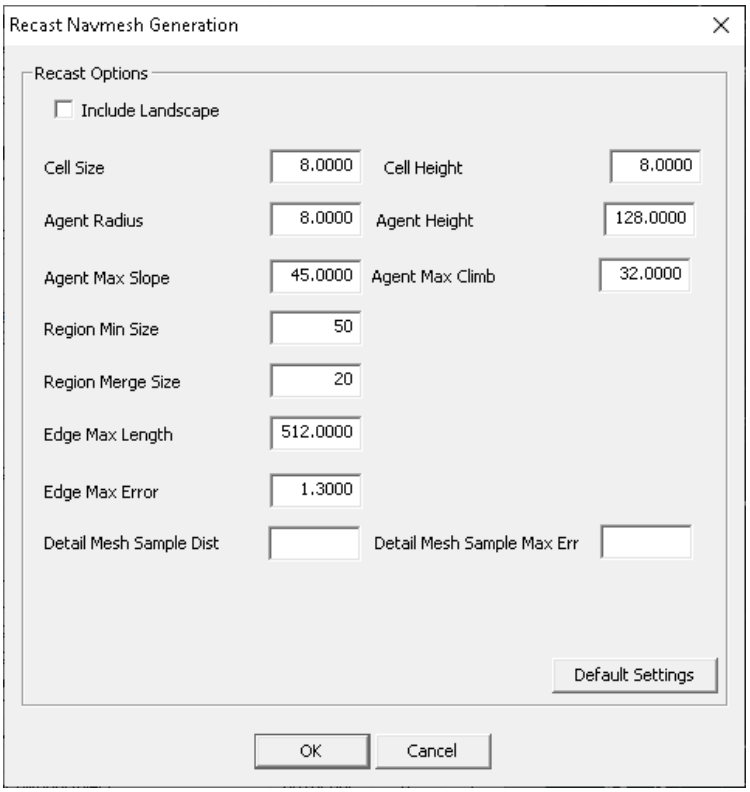


Figure 264 - Recast Navmesh Generation.

ADDING OCCLUSION GEOMETRY

Occlusion planes and occlusion cubes are used to improve performance by hiding objects. They are usually hidden within large structures such as buildings, castle walls, and sometimes within mountain ranges to cull objects that the player should not have direct line-of-sight to.

There are two types of occlusion geometry; occlusion planes and occlusion cubes. Occlusion planes are two-dimensional and can only be resized on an x/y axis. Occlusion cubes on the other hand are three dimensional.

In the following screenshot, the rock will be occluded based on the player's current position and won't be rendered until the player moves around the occlusion cube and regains line-of-sight to it.

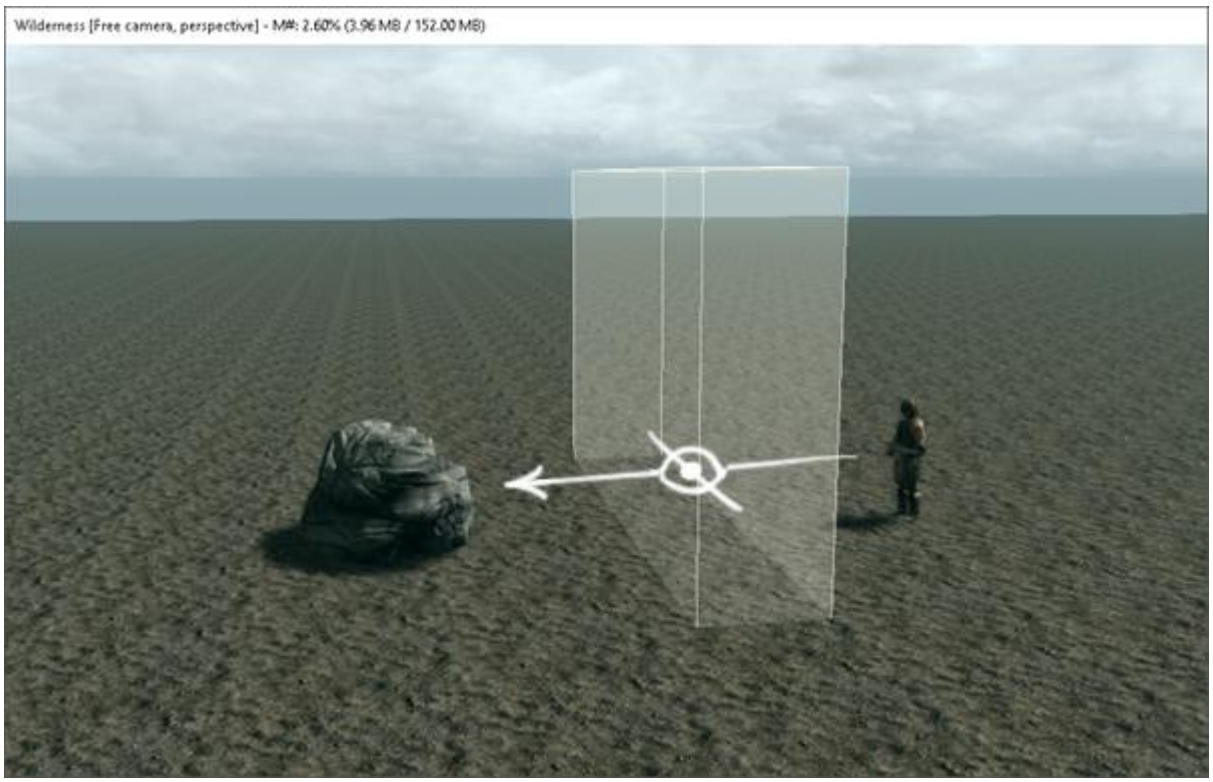


Figure 265 - The rock is occluded based on the player's current position.

Note: The occlusion geometry itself is invisible to the player.

Occlusion geometry can be created around a selected object. In this example I'm selecting one of the town houses in Stonehollow.

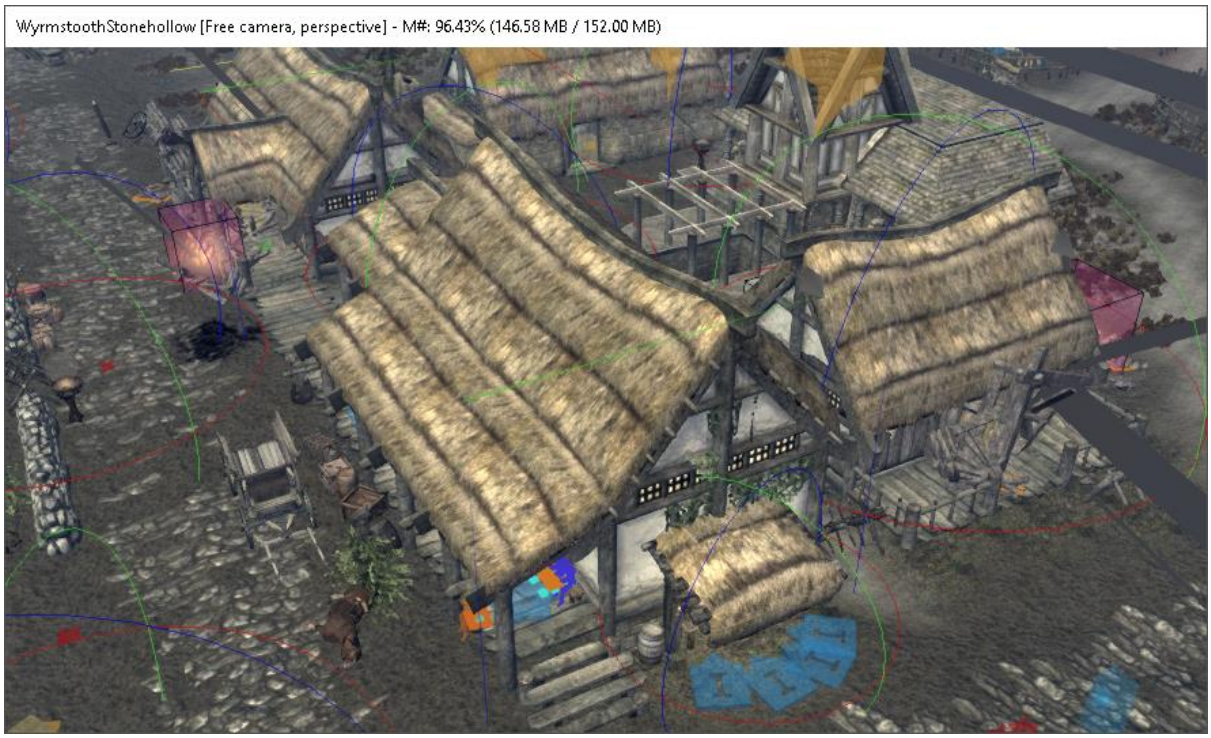


Figure 266 - The object to add occlusion geometry to.

Next, click on the Occlusion Cube button in the toolbar to fit an occlusion cube around the object. The occlusion cube should show up as semi-transparent white cube in the render window.

If nothing appears, press 'M' to toggle markers. If the occlusion cube still isn't showing, go to View > Show/Hide Window and make sure 'Occlusion Planes' is ticked.

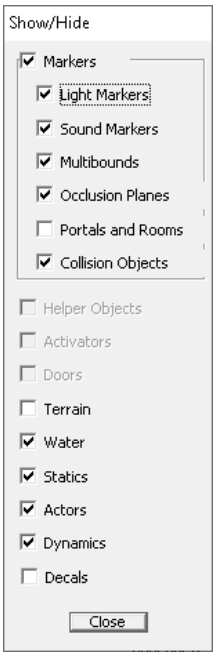


Figure 267 - Show/Hide Window.

The resize gizmo should be automatically enabled when an occlusion cube or occlusion plane is added, but if it isn't, press '2' to toggle it on or off.

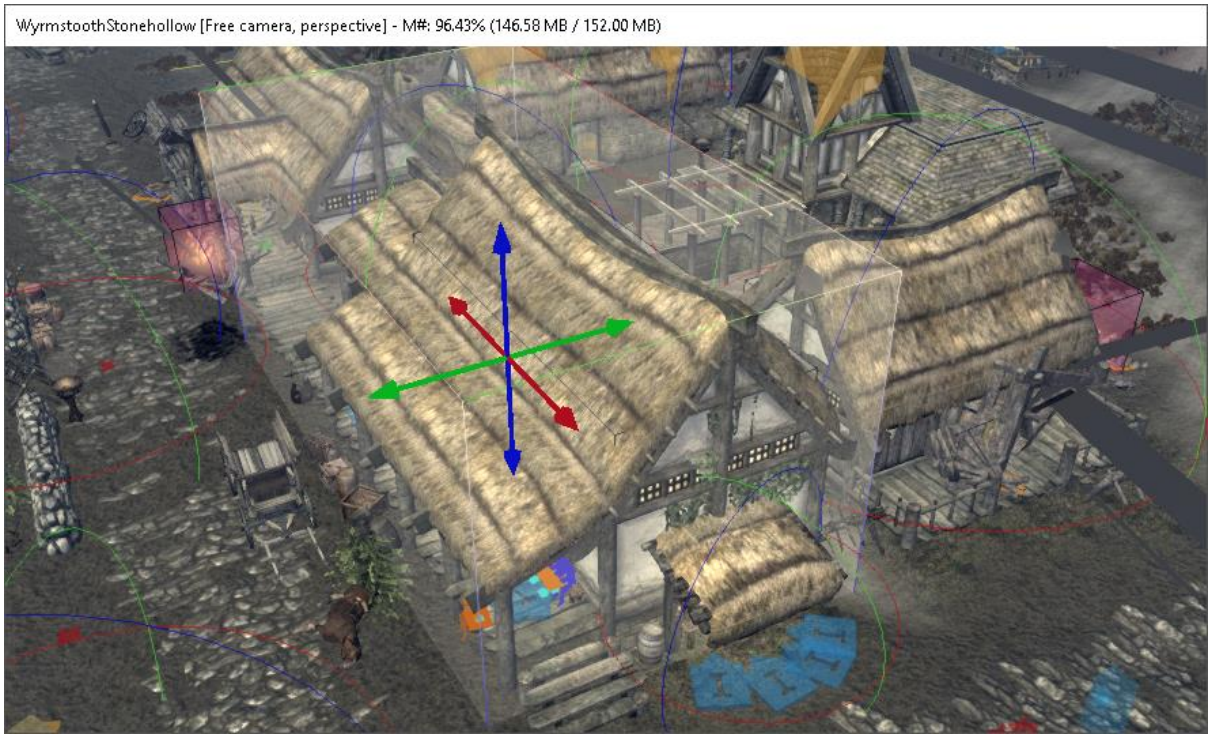


Figure 268 - Occlusion cube.

Important: Use the resize gizmo to resize the occlusion cube so that it's hidden within the geometry of static objects. You should avoid having any part of the occlusion geometry sticking out.

In my example below, I lowered the occlusion cube below the part of the house where the roof starts so it isn't sticking out. I also extended it into the ground a bit to ensure ground-level objects behind the house are being occluded properly when the player walks around it.

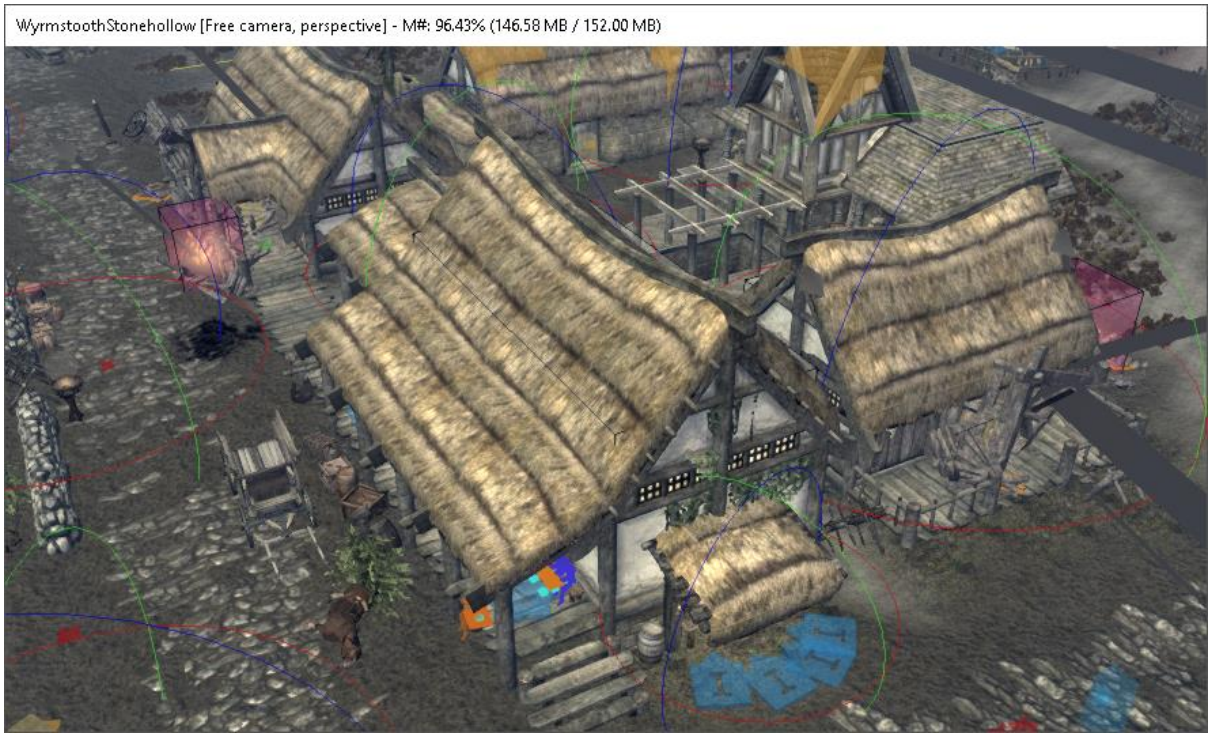


Figure 269 - Occlusion cube fitted to a farm house.

If I raise the house up into the air you can see how the occlusion cube was positioned.



Figure 270 - House lifted up to show occlusion geometry.

If we take a look under the ground plane, you can see how far down I extended the occlusion cube.

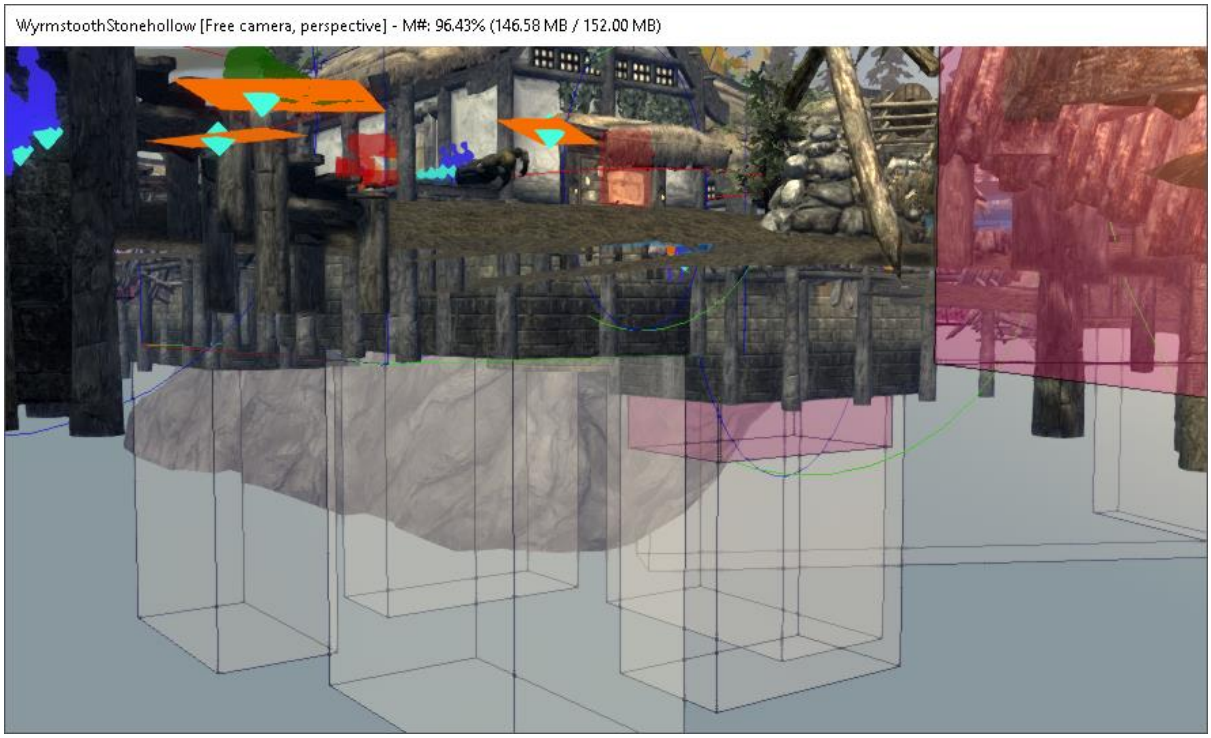


Figure 271 - Occlusion cube extended into the ground.

Occlusion geometry can be used in interiors as well to help occlude portions of a cell that are difficult to box in with roombounds.

Important: Don't add too many occlusion cubes or occlusion planes. The game needs to calculate what objects to occlude, so having a lot of occlusion geometry in a small concentrated area can actually be deleterious to performance.

Once you've added occlusion geometry, I'd recommend walking around in-game with an FPS counter like Fraps to confirm FPS gains instead of losses. If performance is worse after adding occlusion geometry, you might need to rework it or remove it.

In the next example you can see how I hid an occlusion cube within the walls of Fort Moonwatch to occlude geometry outside the walls when the player walks into the courtyard.



Figure 272 - Fort Moonwatch wall.

And there's the occlusion cube hiding in the fort's walls.



Figure 273 - Wall piece raised to reveal occlusion cube within.

CLUTTERING A WORLD SPACE

The art of level design could be a separate book unto itself. For this section I'm going to try and keep things terse by showing you how I approached this task during the development of Wyrmstooth with some specific examples. If you're looking for information on how to set up region-based automatic cluttering, see the section on Region Generated Textures and Objects.

Note: Personally I would recommend cluttering your world space manually by hand versus region-based automatic cluttering. Even though that means hand-placing every single object, in my opinion it's easier to make a more organic-looking landscape. That said, you may find automatic cluttering useful for placing some initial objects down to help you get started.

Before you begin, know what you want to achieve. Whether that's from a rough sketch of the world space, a list of ideas, photos, and/or professionally drawn concept art. I would strongly recommend having something to reference from first.

Important: Cluttering a world space is probably the most time-consuming part of making a new lands mod. The last thing you want to do is delete what you've already made because you changed your mind about the design or tone you're trying to achieve.

Level design usually begins at the blocking out phase. Blocking out is the process of determining the playable area, shape and flow of a 3d map. In other game engines such as Unreal this is usually done with proxy geometry but in Skyrim we have the benefit of already having assets ready to be used.

I would recommend you begin blocking out your world space by starting with the biggest pieces first. Start with mountain peaks and mountain cliffs, then work down in scale leaving the smallest objects to last. Lakes and rivers can also be useful in dividing up your world space.

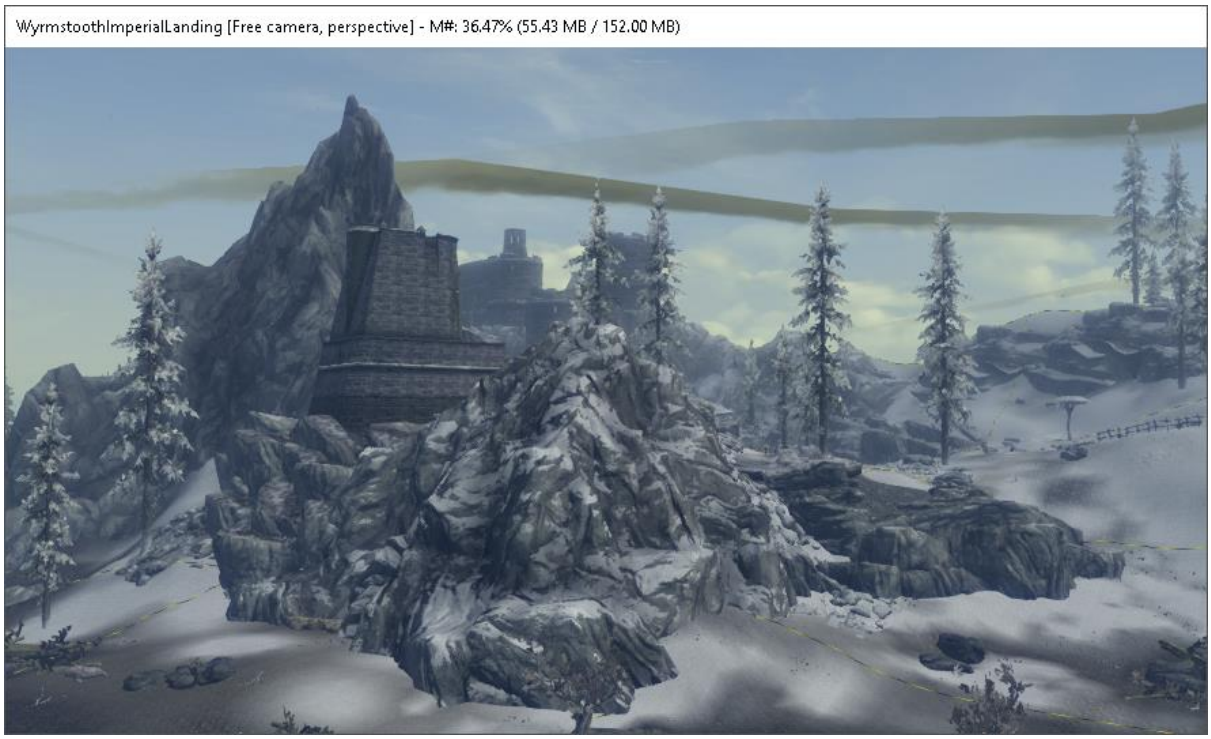


Figure 274 - Mountain peaks and mountain cliffs.

You should approach level design as a series of 'passes'. Your first pass might involve placing mountain peaks, mountain cliffs, lakes and rivers to split up your world space into distinct regions. The next pass after that might involve making a start on the largest points of interest on your map such as cities, towns and villages - even if it's only building placement at this stage. Then work down in scale from there.

Important: By splitting up a seemingly insurmountable task into a series of subtasks, you make the work more manageable.

If you're working with a team of level designers, split your world space up into a series of regions first to divide the work between yourselves. Refer to the section titled Version Control and Merging Changes to collate your work back into a single master file.

In the screenshot below you can see where I placed mountains to loosely separate the Wyrmstooth world space up into several distinct regions, such as the Western Forests, the Steampools at the center of the map, the Northern Marsh, the Eastern Glacier, and the ring of snow-capped mountains around the centre.

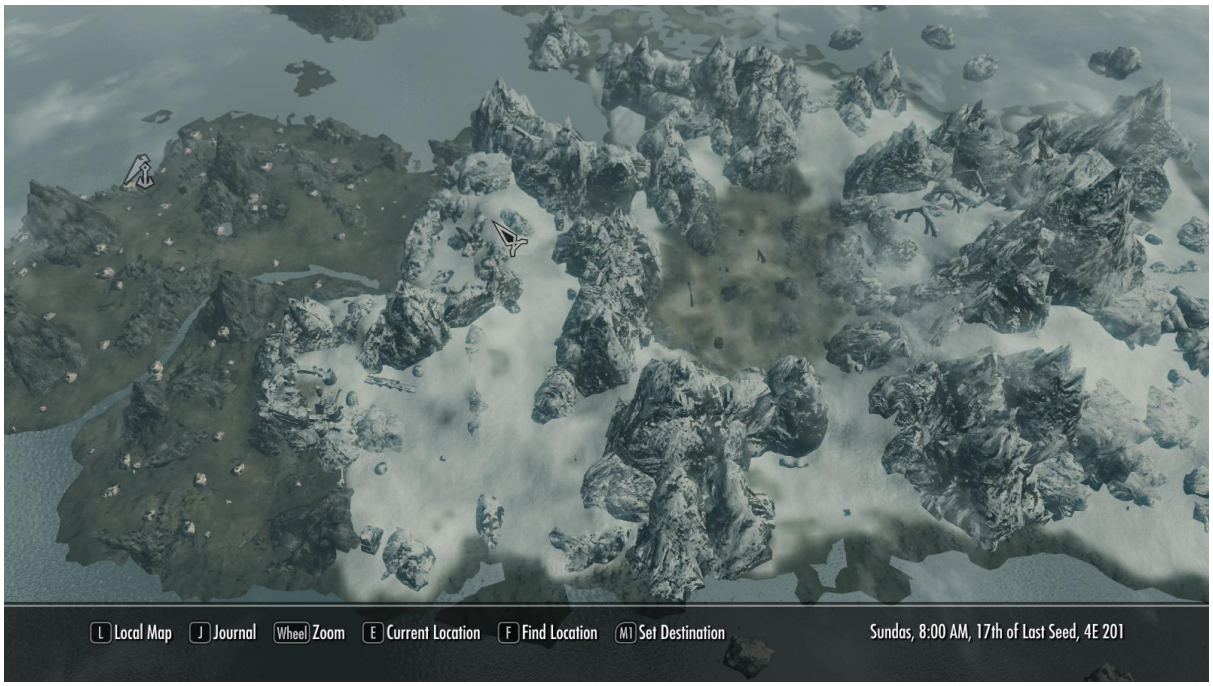


Figure 275 - Splitting a world space up into several distinct regions.

Mountains will also help reduce the amount of overall cluttering you'll need to do by covering a lot of your map! ☺

Note: When placing mountain ranges, try and avoid straight lines. Try and be mindful of how your world space is going to look in the map screen once you've built LOD.

Take some inspiration from real-world maps when working out the layout of your world space.

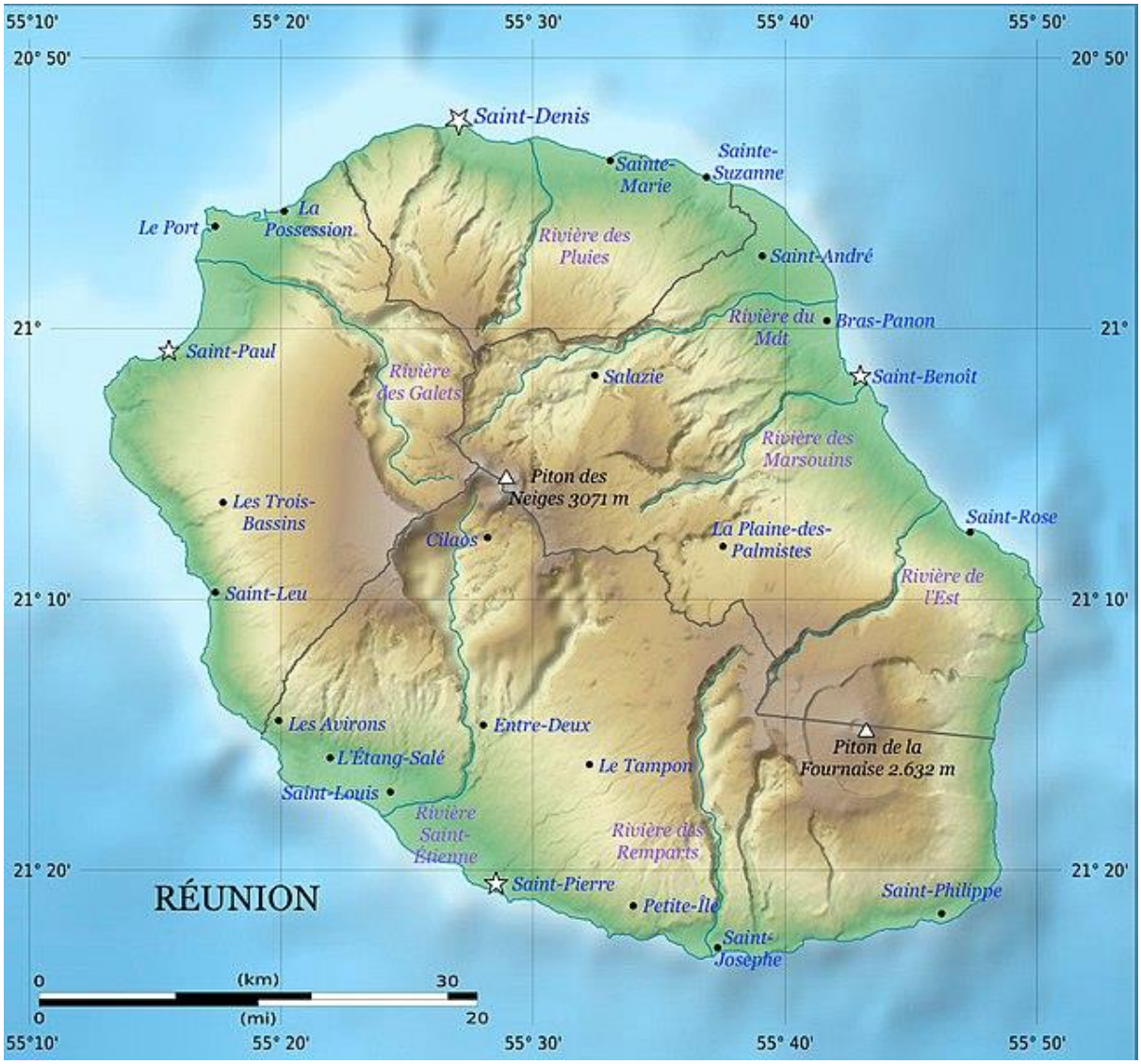


Figure 276 - Map of Reunion Island.

For Wyrmstooth, I took some inspiration from real world volcanic islands such as Reunion Island.

Medium sized objects such as the MountainCliffSlope and MountainTrim variants can be useful for blending mountain peaks and mountain cliffs with the ground as per the screenshot below:



Figure 277 - MountainTrim and MountainCliffSlope variants used to cover the base of a mountain.

Generally you'd want to cover the intersection between a mountain peak or mountain cliff and the ground layer. Rather than having mountains simply stick out of the ground, I like to surround mountains with smaller details to simulate erosion and rock falls.

Over time, rocks and boulders would've broken away from cliff sides and mountain peaks, coming to rest in the valleys or plains below. MountainTrim variants are also useful for quickly adding smaller rock detail.

Holding down the 'Z', 'X' or 'C' keys while right-click dragging an object in the render window will constrain rotation to an axis. Giving these objects a little bit of rotation will help make them look a bit more visually distinct from each other. I would also recommend adding some random scaling. You can toggle the scale tool on or off by pressing '2'.

You can also change an object's scale in its reference properties which you can get to by double-clicking or right-clicking on the object in the render view and selecting Edit.

You can change the Scale value under the 3D Data tab.

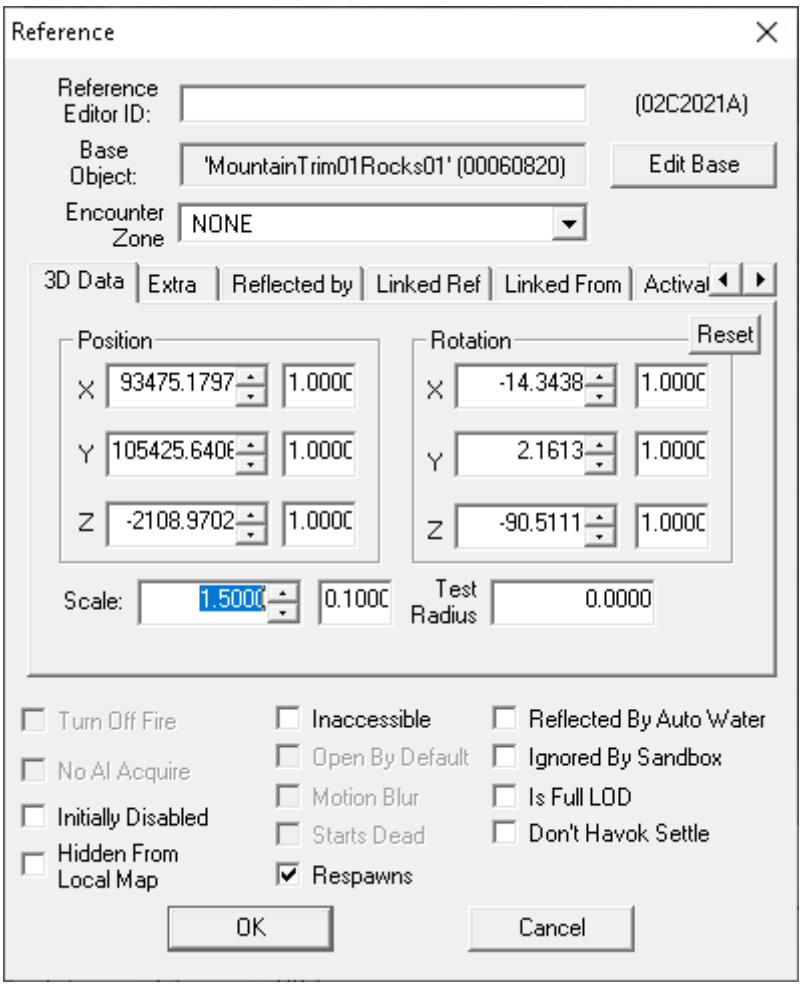


Figure 278 - Changing the scale of an object.

I would avoid increasing scale beyond 2.0000 to prevent stretching the textures too much.

Next let's have a look at adding roads.

In the Object Window, filter by 'road'.

When adding roads, I like to start with the larger pieces first, such as RoadStraight02, RoadCurveLong45R01 or RoadSCurveR01. Leave some gaps in between these pieces.

Once those are down, fill the gaps in-between with smaller road pieces such as RoadChunkL03, RoadChunkM01 and RoadChunkM03.

This will help make the road look like it has been worn down with time.

Road pieces are very flat so you might have trouble winding a road over a hill with the larger road pieces. Use the smaller road pieces for that instead.



Figure 279 - Old worn road.

Use the landscape edit tool to raise the landscape up over the sides of the road to make the seam between the land and the road less regular.

One trick for hiding the seam between the landscape and the road is to place rocks, thickets or bushes over it to make it less noticeable.



Figure 280 - The road to Wyrmstooth Barrow.

In snowy landscapes, you can even cover portions of the road with the landscape itself to simulate snow accumulation.

Now that we've divided our landscape up a bit, let's start filling the wilderness in-between.

The main technique I use for doing this is to develop 'islands of detail'. Maybe start with a big object like an old tree stump and surround it with smaller objects like rocks, bushes, thickets or smaller trees.



Figure 281 - An island of detail around an old stump.

You can then fill in the areas between these islands to blend them together.



Figure 282 - An island of detail around a tree.

Large objects like DirtCliffsIsland01PineForest01 or TundraStreamBend01PineForest02 can also help seed islands of detail.

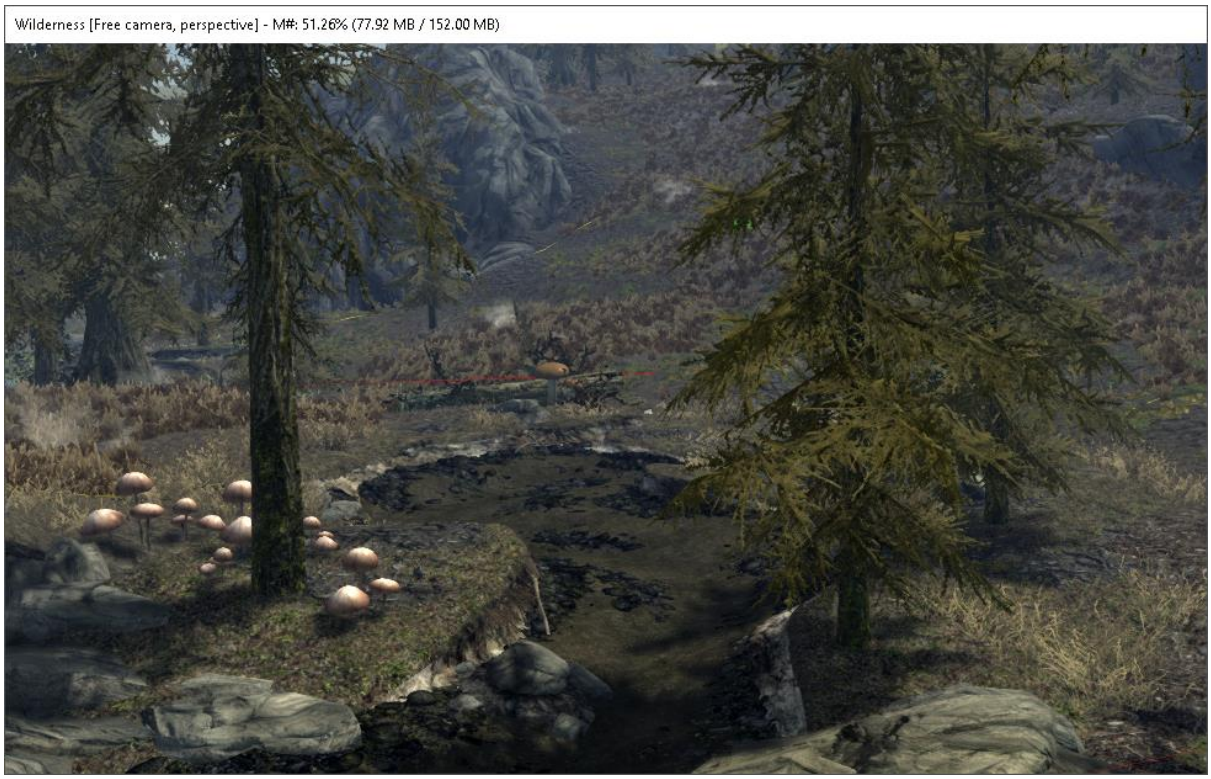


Figure 283 - TundraStreamBend01PineForest02.

TundraStreamBend01PineForest02 is normally used to create small streams, but it can also help add variety to your landscape without water too. Maybe a stream once flowed here but dried up long ago.

Important: Let the details you add to a landscape tell a story of their own. This concept is called Environmental Storytelling.

Consider the fact that Skyrim isn't a fast-paced game and players have the freedom to explore the world at their own leisure. Good environmental storytelling helps build the player's immersion with your world space.

For coastlines I like to use TundraPond01Coast1 and blend it with the LCoastBeach01 texture to create these muddy pools where creatures such as mud crabs may reside. I also added driftwood along the shore, perhaps recently washed ashore by the tides.

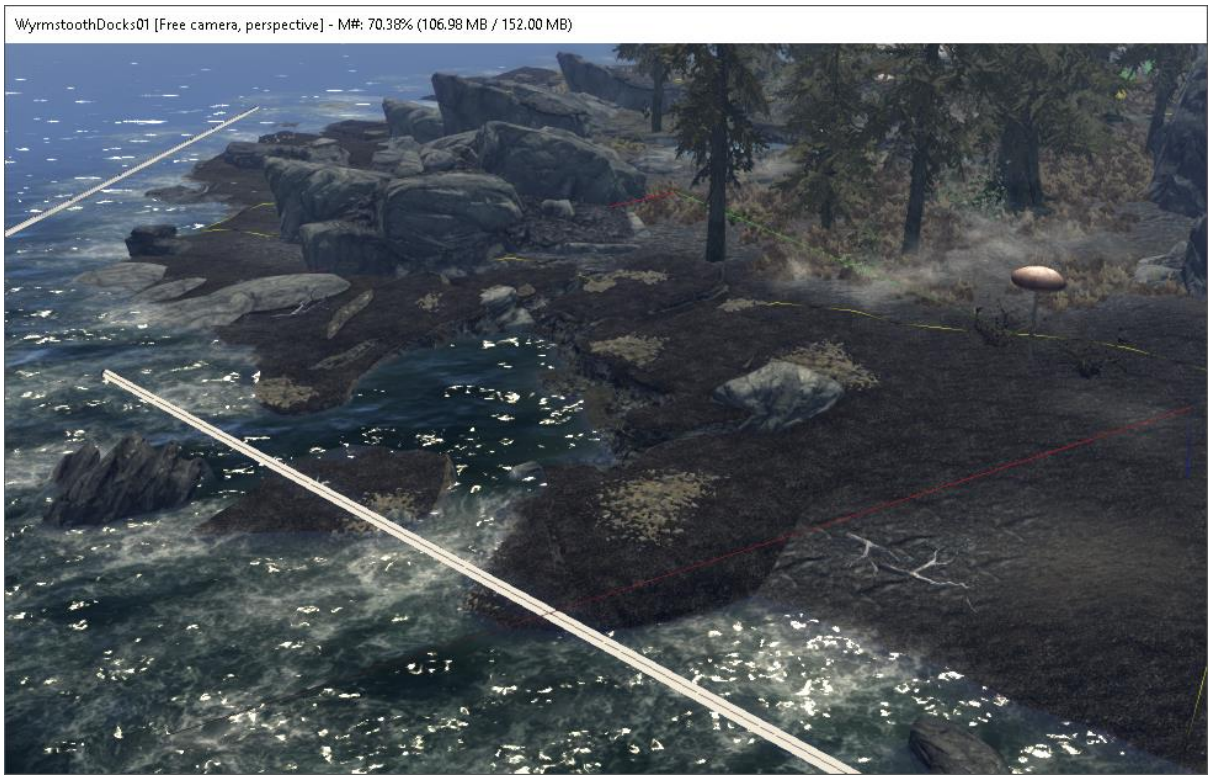


Figure 284 - Coastal pools.

Some objects have several different variants with different textures or shader effects like snow or sulfur. For example, for placing rocks in water use the Wet variant. E.g.: RockL01Wet.

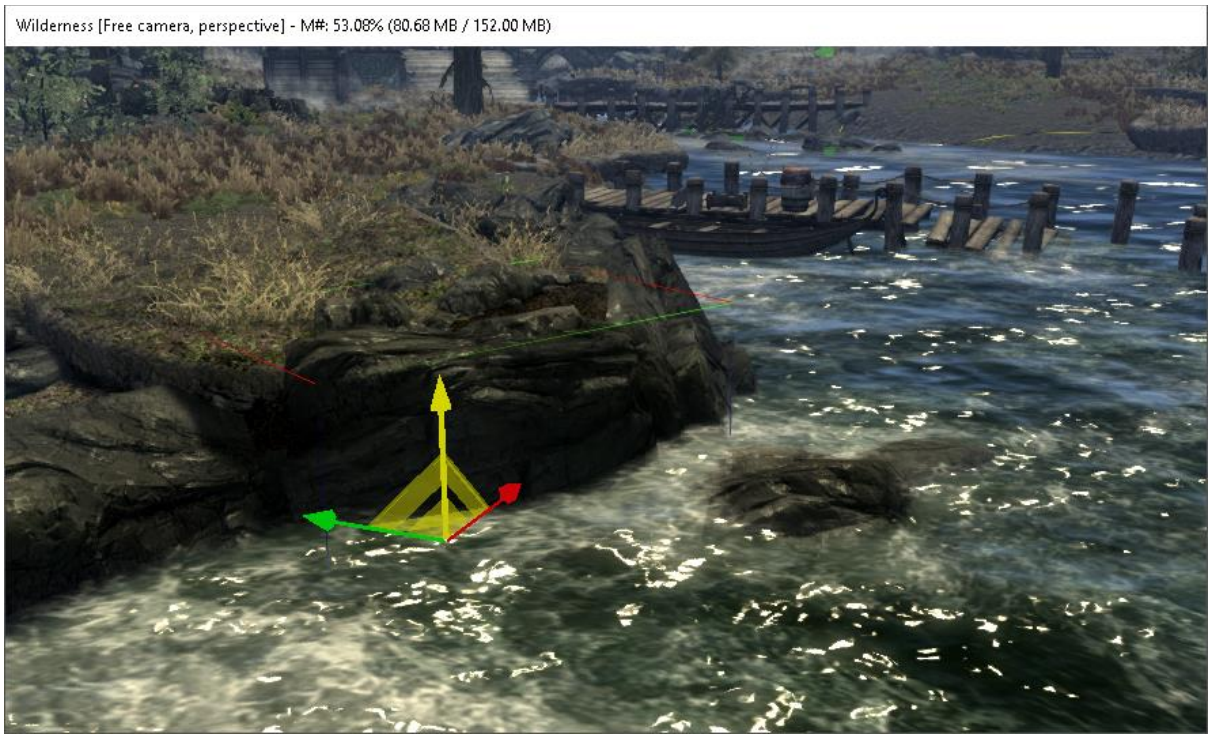


Figure 285 - RockL05Wet.

Medium sized objects such as DirtCliffsIsland01PineForest02 can be useful for adding some unevenness to the landscape. The problem however is that grass is only populated on the landscape.

To try and fix this you can use thickets to cover these bald areas.



Figure 286 - TreeThicket01 covering DirtCliffsIsland01PineForest02.



Figure 287 - Blending automatically generated grass with manually placed thickets.

Don't forget to clutter the ocean floor if your world space has a coastline.

Start with rocks and boulders like MountainTrim02CoastOceanFloor01, RockPileL04CoastOceanFloor01 or MountainTrim01CoastOceanFloor01 then add smaller objects around then such as RockPileM02CoastOceanFloor01 or RockPileL02CoastOceanFloor01.



Figure 288 - Ocean floor cluttering.

In the screenshot above, I painted the LCoastOoeanFloor01 texture in between the rocks to place areas of kelp along the ocean floor.

A few underwater shipwrecks can be found along the coast. Barrels and crates that were lost in these wrecks can also be found scattered along the ocean floor for the player to loot. Nordic barnacles can be found attached to the sides of these old wrecks, and slaughterfish use them to lay clumps of eggs in sheltered nooks.

Adding points of interest can help break up the monotony of the wilderness. These may be small camps, ancient ruins, remnants of houses that have been destroyed, and so on.

Here are some points of interest I added around Wyrmstooth to give you some ideas:

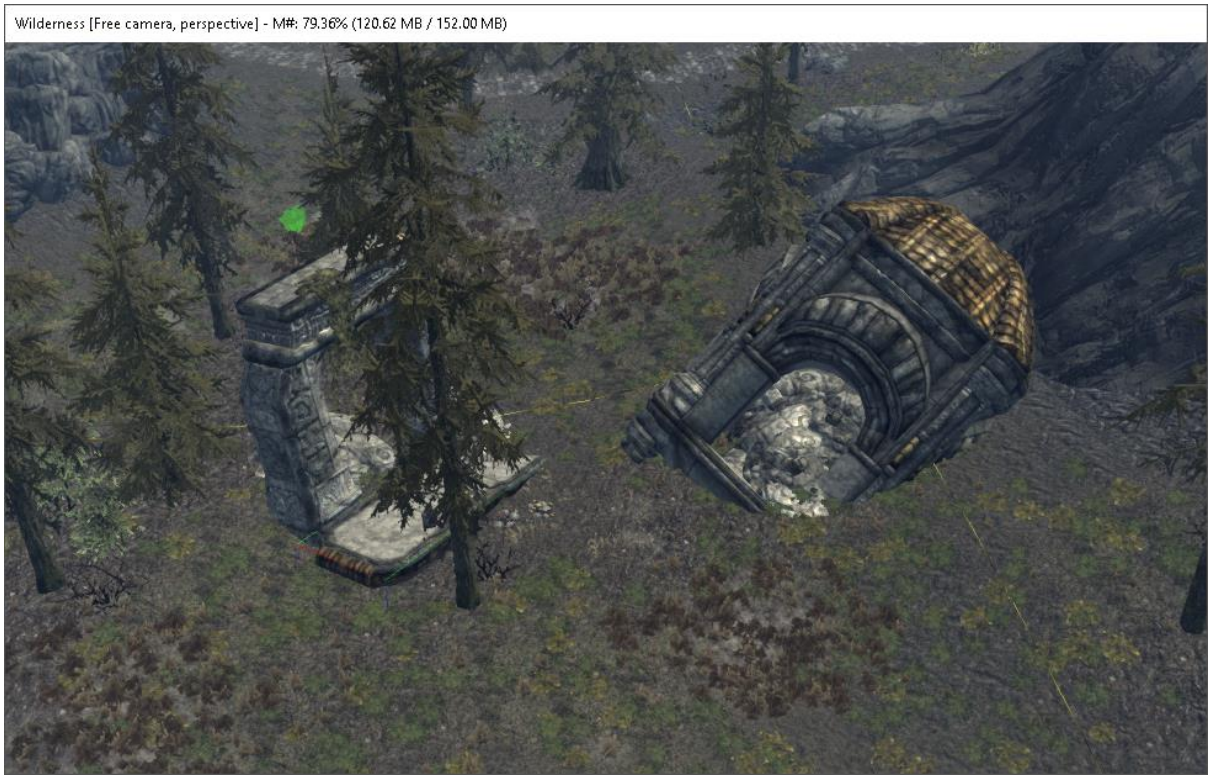


Figure 289 - A small dwemer ruin.

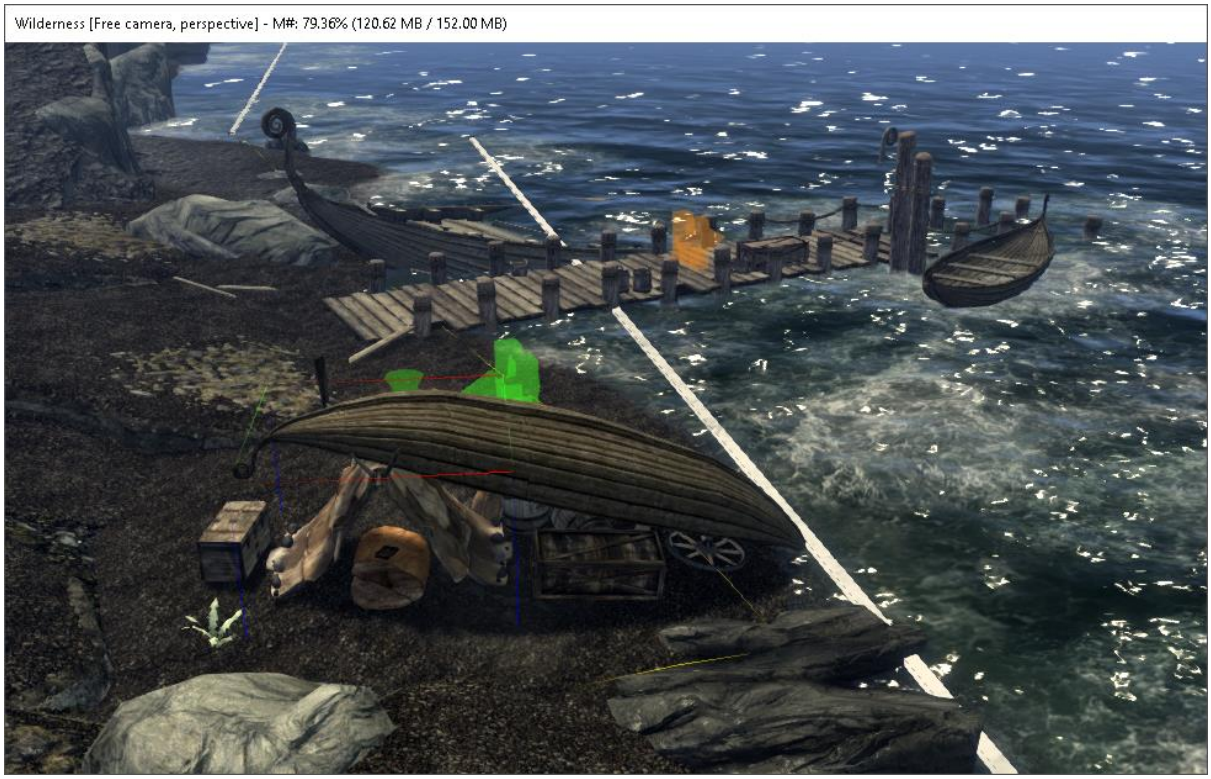


Figure 290 - A small dock and camp on the shore.

In the screenshot below, I repurposed WHshorttower to make a monument for a grave.



Figure 291 - Grave of an unknown warrior.

Don't be afraid to repurpose existing assets.



Figure 292 - A boat covering a make-shift shack.

As humid air is forced up the side of a mountain it condenses into clouds. We can simulate this effect by manually adding in these clouds. This can also be used to help exaggerate the height of the mountains, making them seem like they're poking through the cloud layer.

Filter by 'INV_' to get a list of all cloud shapes that you can manually place.



Figure 293 - Clouds along a mountain side.

Clouds placed at ground level may look like a bank of thick fog, but it's not a volumetric effect.



Figure 294 - Clouds at ground level.

Normally used in dungeons, you can repurpose the FXMist variants to add a thin layer of ground-level fog.

FXMist variants are found under WorldObjects > MoveableStatic in the Object Window.



Figure 295 - FXMistLowSmokey01HalfVis.

To add a rolling fog effect, I scaled up a customized version of FXVolumeMist. FXVolumeMist is a generator that spawns 'fog' geometry as a particle effect.



Figure 296 - FXVolumeMist.

To add birds to the sky I used FXbirdOPreyNoNestACT which is found under WorldObjects > Activator under in the Object Window.

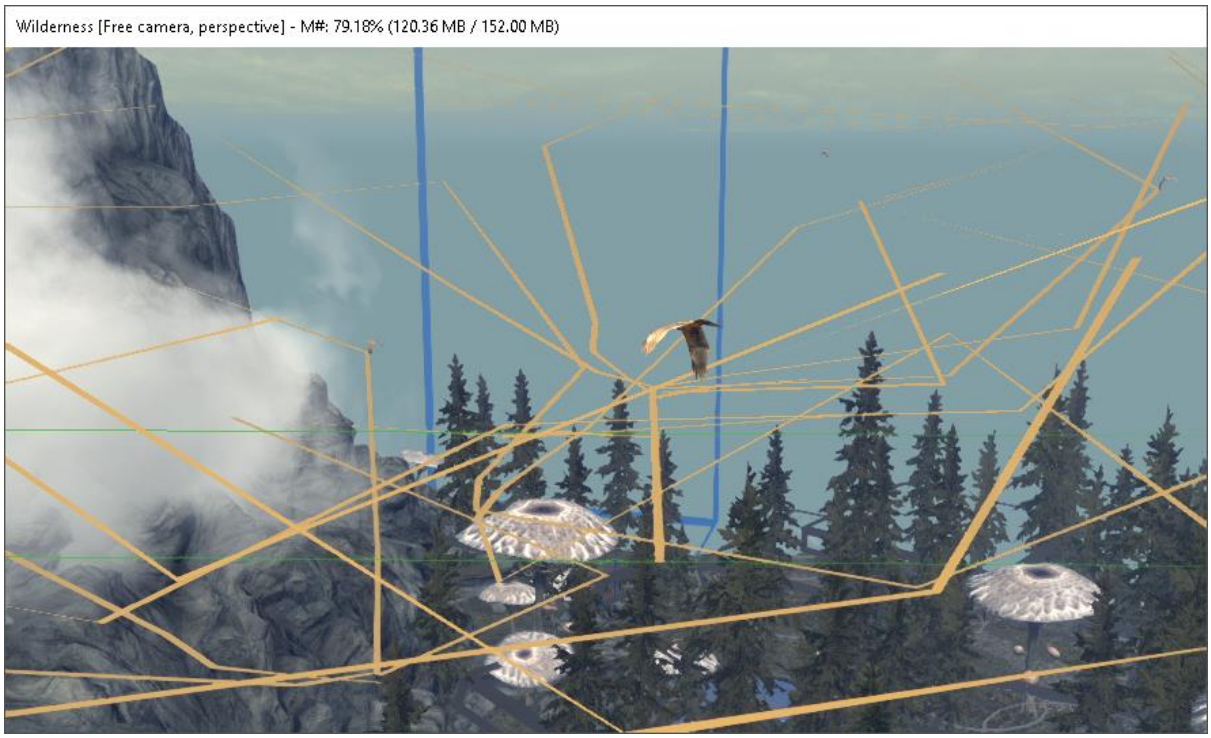


Figure 297 - Birds.

These are simple objects that follow the yellow outlined path.



Figure 298 - More birds.

When placing waterfalls, build up the sides of the waterfall with rocks to help make them look a bit more believable.

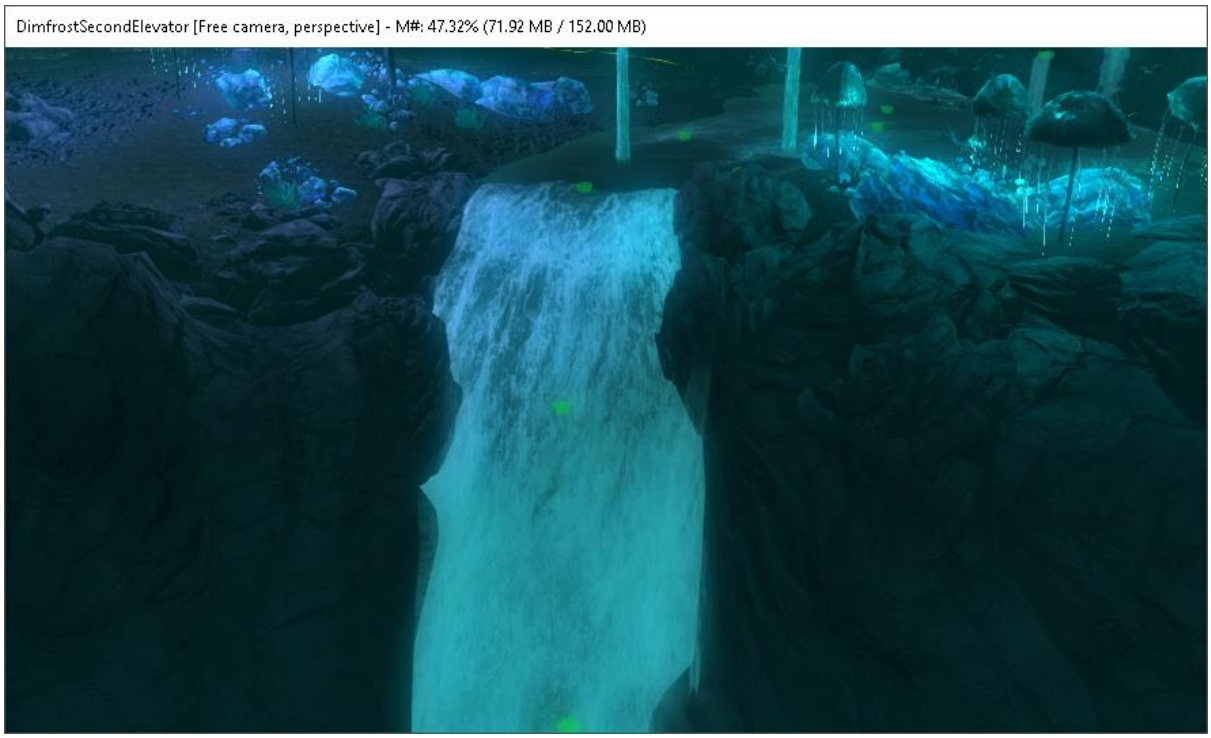


Figure 299 - A waterfall in Dimfrost.

This is to help simulate the fact that over the centuries, the force of the water would've begun to cut away at the cliff side.

Use FXWaterfallSkirtTallFront to simulate the mist at the bottom of a waterfall.

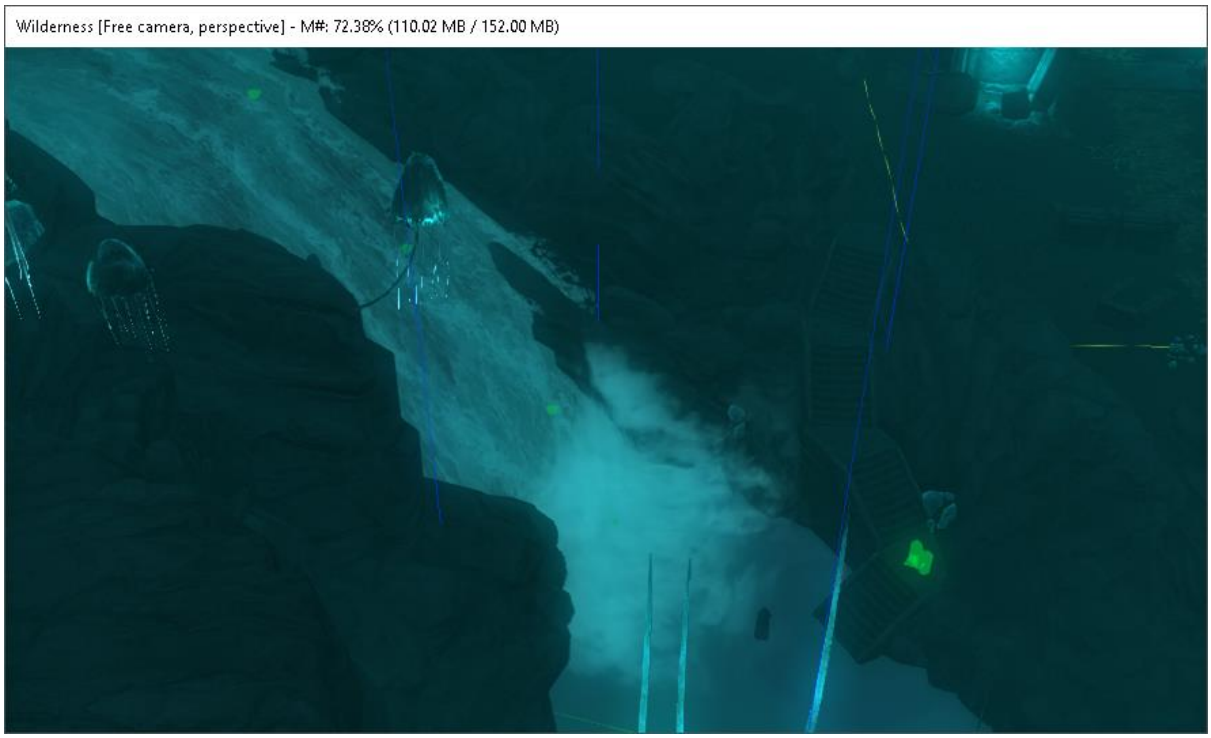


Figure 300 - FXWaterfallSkirtTallFront.

Important: As you clutter the world space, it's important to test in-game fps.

Just keep in mind: the more objects you add to each cell, the more geometry the game engine has to render, and the choppier your framerate will become.

You can use a program like fraps to display an fps count.

I aimed to keep in-game fps close to or at 60 fps.



Figure 301 - A screenshot from fraps showing the fps counter.

There are ways to mitigate low framerates by hiding objects from being rendered if they are obstructed from view.

See the sections on Adding Occlusion Geometry and Generating TVDT Occlusion for more information.

CHAPTER 2: MARKERS

ADDING IDLE MARKERS

We can use idle markers to influence an NPC's idle behavior. Typically they're used to set up patrol points or to make NPCs play idle animations at specific spots.

A list of idle markers can be found in the Object Window under Miscellaneous > IdleMarker.

Note: NPCs can use idle markers in two different ways; by being linked to one directly, or through a package.

For example, NPCs with a sandbox package will randomly choose nearby furniture and idle markers to interact with. If we place a BrowseIdleMarker marker in front of a shelf, an NPC that is sandboxing in that cell will eventually walk over to the idle marker, face the direction it's pointing, and play the IdleExamine and IdleStudy animations that are assigned to that specific idle marker.



Figure 302 - An example of a BrowseIdleMarker marker placed in an interior cell.

Most idle markers are shown in the render window as a blue square with an 'I' and an arrow pointing in the direction that the marker is facing, though some may have a custom marker model indicating how an actor will be positioned when using it.

Here's an example of some more idle markers placed in an interior cell.



Figure 303 - More idle markers in an interior cell.

This scene contains a mix of WarmHandsIdleMarker to play the IdleWarmHands and idle_A_leg_shift animations, and SweepIdleMarker to play the idleLooseSweepingStart animation.

By default, idle markers can be used by anyone. But you can add NPC or faction ownership to them to ensure they can't be used by anyone else.

Double-click or right-click on an idle marker in the render window and select Edit.

Go to the Ownership tab.

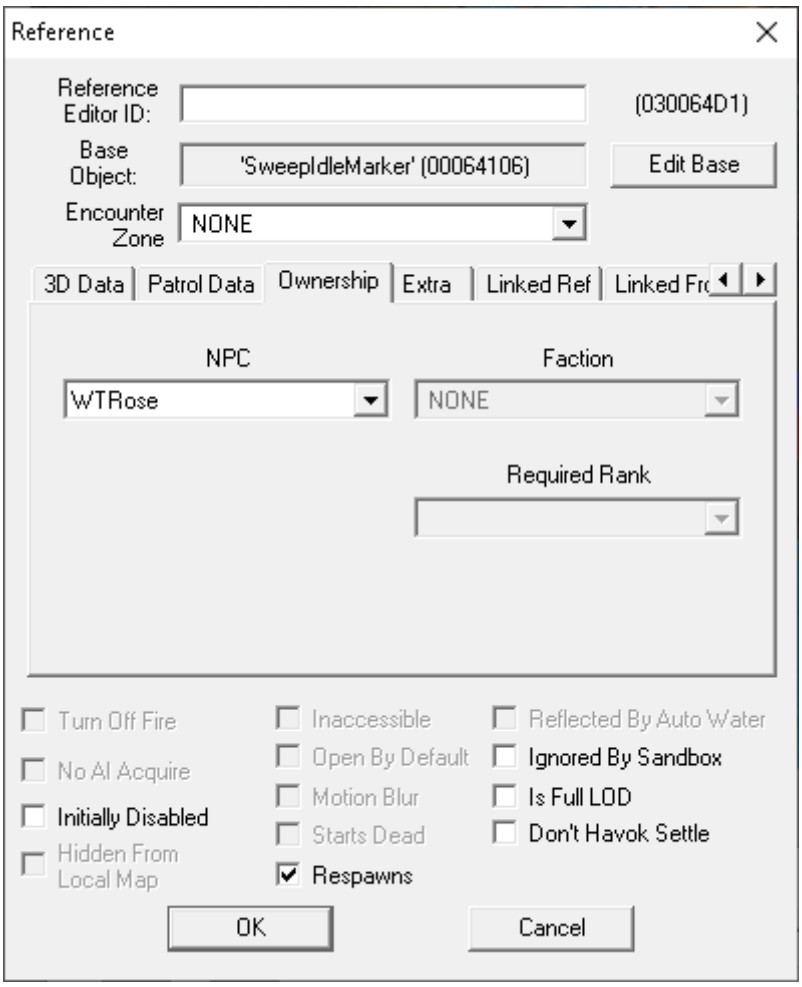


Figure 304 - Setting ownership on an idle marker.

In the screenshot above, only a specific NPC can use this particular SweepIdleMarker.

Click OK to close the reference properties.

Let's say we spawned in a bandit and wanted the bandit to patrol between several different spots. We can do this by linking two PatrolIdleMarker markers together then linking the bandit to one of them.

In the screenshot below, I've placed an LvlBanditMelee1H and two PatrolIdleMarker markers for the bandit to travel back and forth between.



Figure 305 - An LvlBanditMelee1H spawner and two PatrolIdleMarker markers.

First we need to link the two PatrolIdleMarker markers together.

Double-click or right-click on one of the PatrolIdleMarker markers and select Edit.

Navigate to the Linked Ref tab, right-click in the References list and select New.

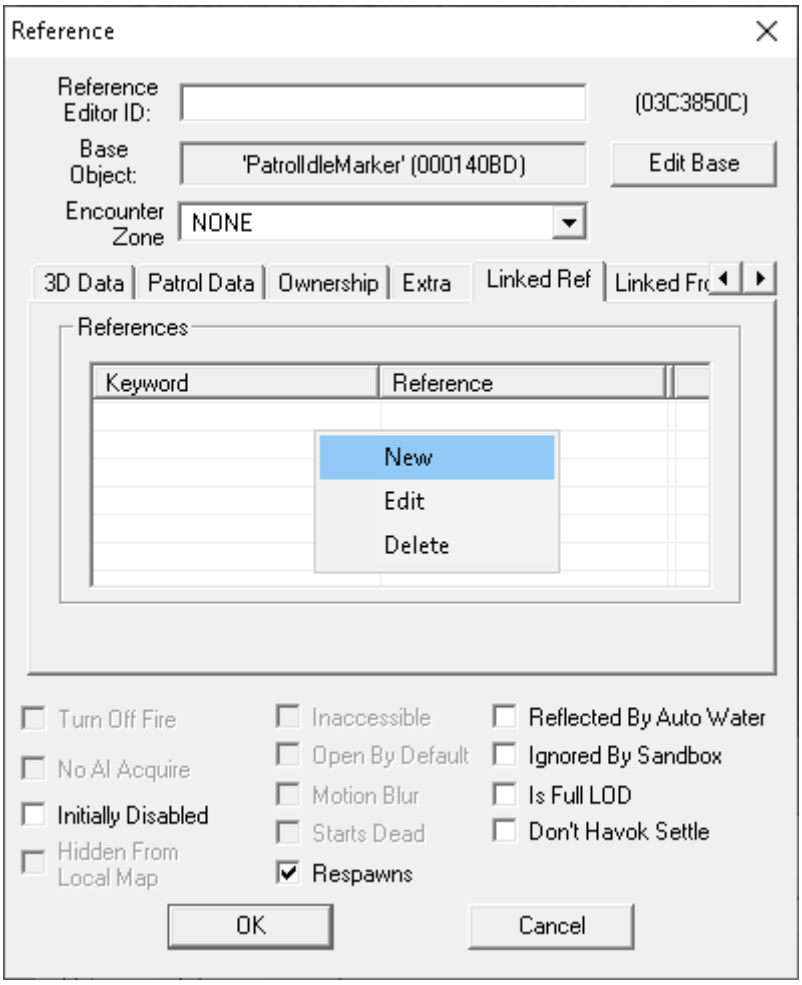


Figure 306 - Adding a new linked ref.

Click on 'Select Reference in Render Window'

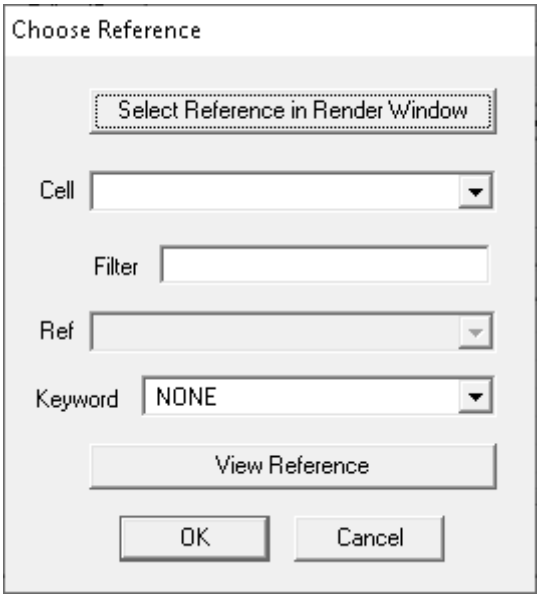


Figure 307 - Selecting the other PatrolIdleMarker marker from the render window.

This will bring up the render window and your mouse cursor will change to a crosshair. Use it to select the other PatrolIdleMarker marker by double-clicking on it.

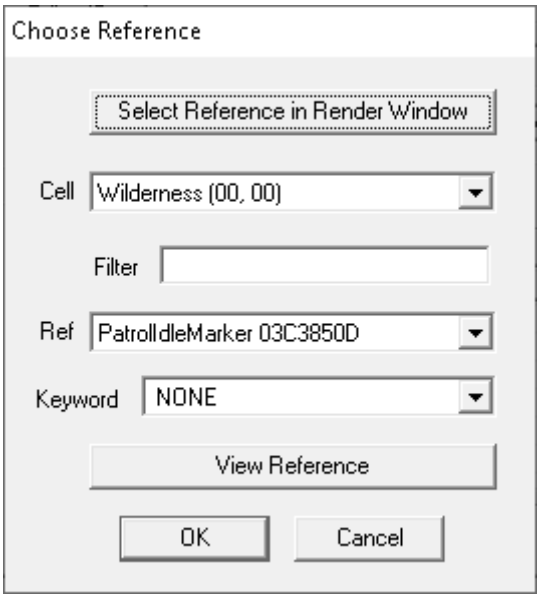


Figure 308 - Selected the other PatrolIdleMarker marker.

Click OK.

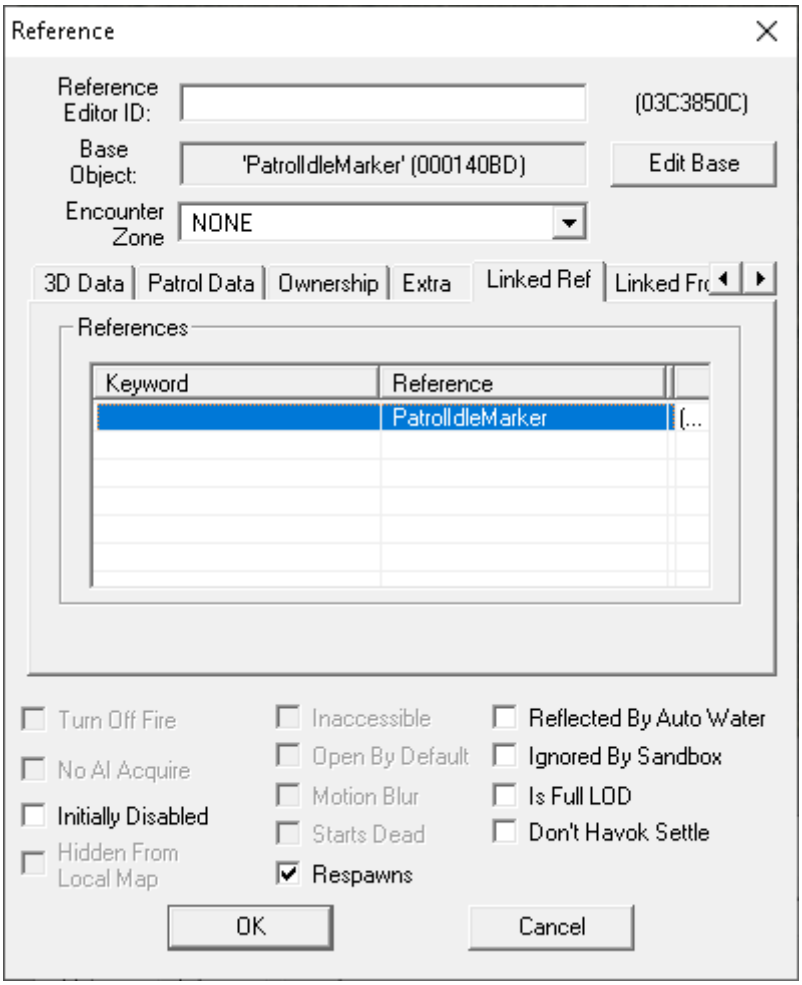


Figure 309 - The two PatrolIdleMarker markers are now linked together.

Click OK to close out of the Reference properties.



Figure 310 - Arrow showing two linked references.

In the render window you should now see an arrow from the first PatrolIdleMarker marker pointing to the marker it was linked to.

Now we need to add the bandit to these patrol points.

Double-click or right-click on the bandit spawner and select Edit.

Go to the Linked Ref tab, right-click in the References list and select New.

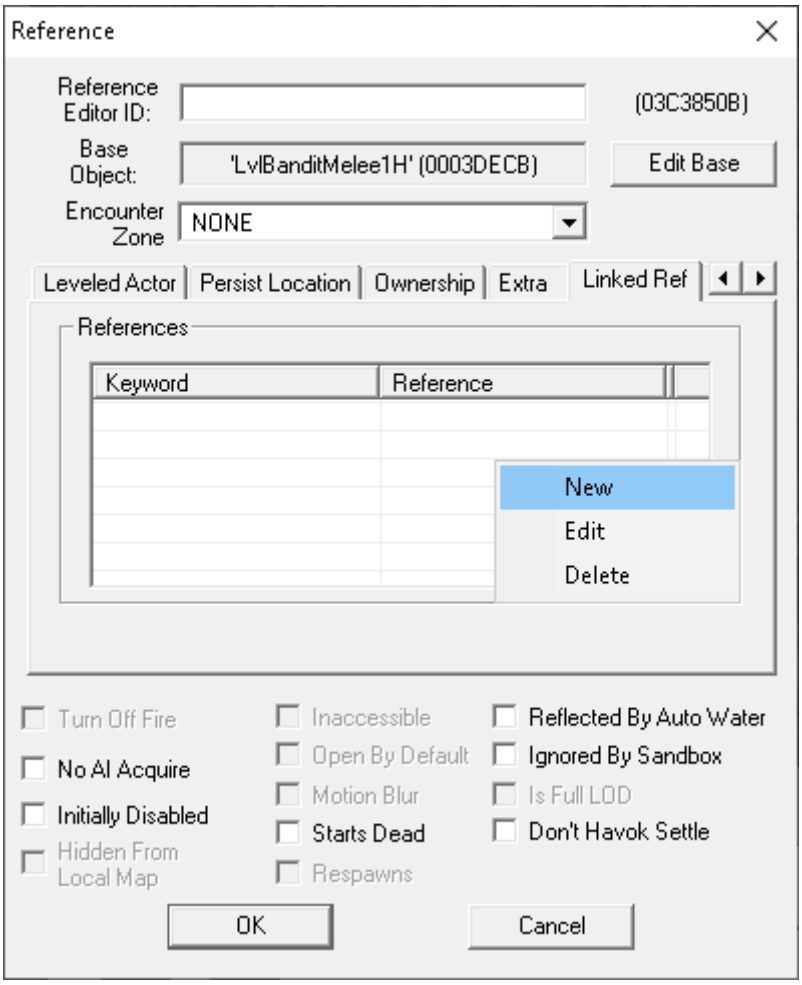


Figure 311 - Adding a new linked reference.

Click on 'Select Reference in Render Window' and select one of the PatrolIdleMarker markers then click OK.

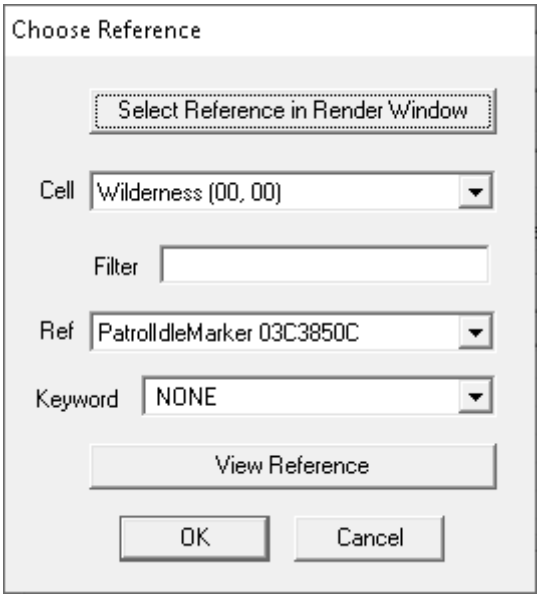


Figure 312 - Linking the bandit spawner to one of the PatrolIdleMarker markers.

Confirm the PatrolIdleMarker marker is listed in the References list then click OK to close out of Reference properties.

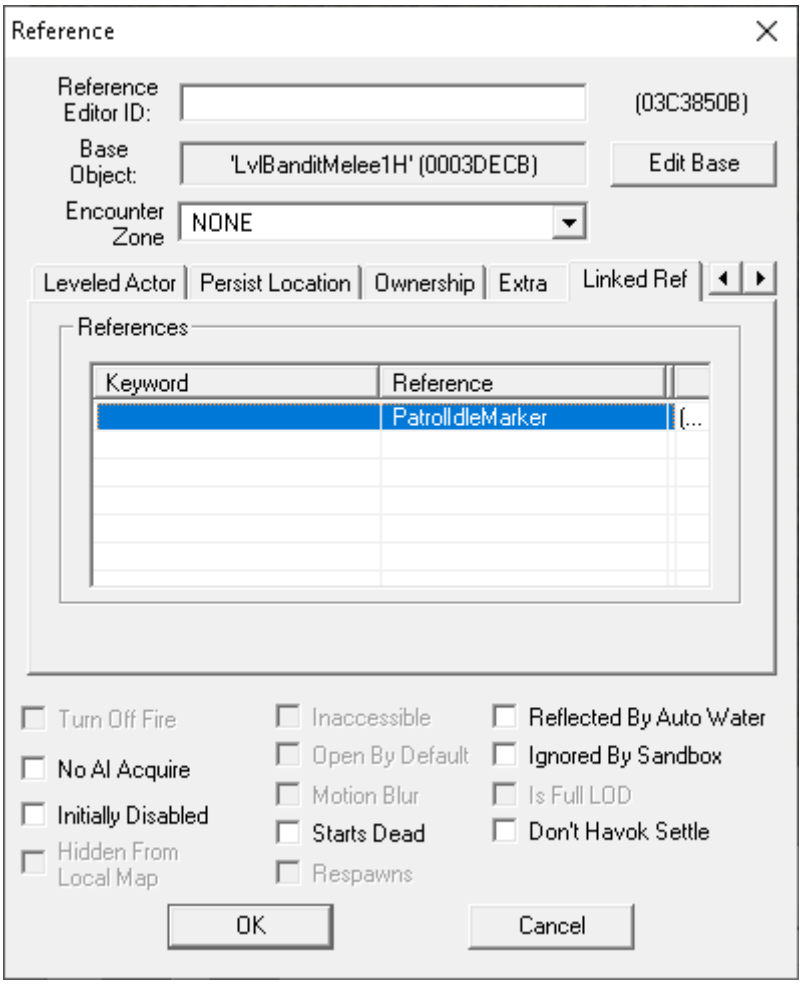


Figure 313 - LvlBanditMelee1H is now linked to a PatrolIdleMarker marker.

You should now see an arrow pointing to the PatrolIdleMarker marker that the bandit was linked to.



Figure 314 - Our bandit now has patrol points.

Right now, our bandit will walk back and forth between these two points without stopping.

We can control how much time the bandit spends at a particular patrol point by adding some idle time.

Double-click or right-click on one of the PatrolIdleMarker markers and select Edit.

Go to the Patrol Data tab.

Let's say we want the bandit to stand here for 5 seconds before moving on. Tick 'Patrol Data' and set the Idle Time to 5.0000.

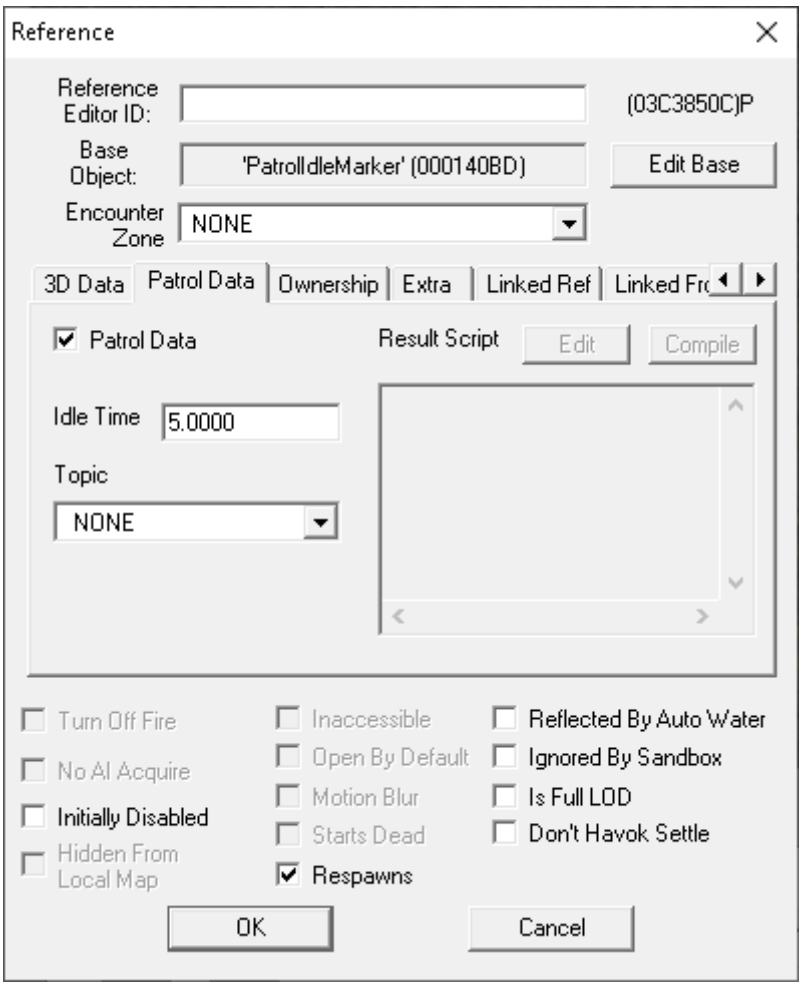


Figure 315 - Adding some idle time to this patrol point.

Click OK to close out of the Reference properties.

Repeat these steps for the other PatrolIdleMarker marker if you want to.

While technically defined as furniture, DragonPerchWordwall, DragonPerchTower and DragonPerchRockL02 can be used to let dragons land on objects.

You'll find them under WorldObjects > Furniture. Just filter by 'dragon'.

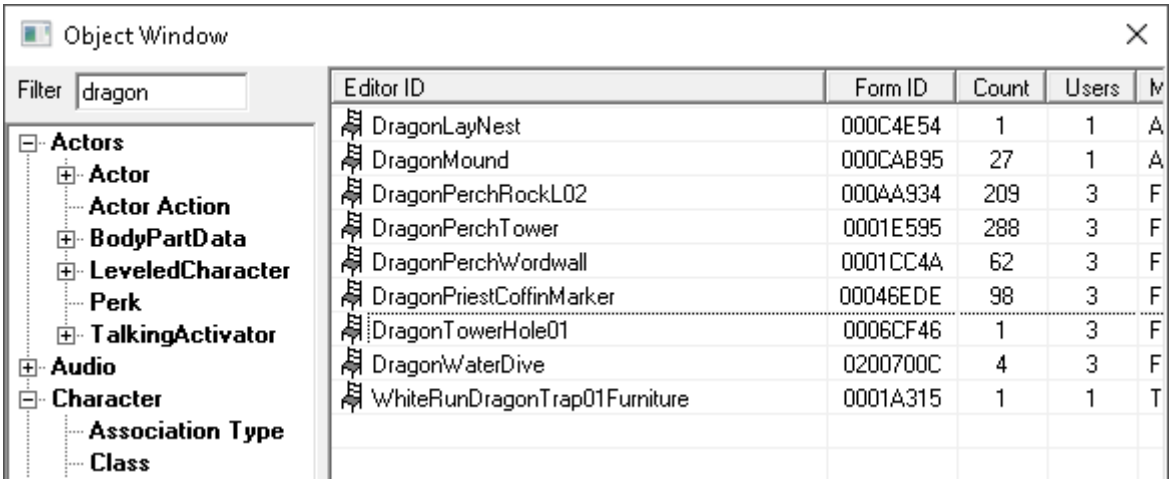


Figure 316 - A list of dragon markers.

In the screenshot below, I added DragonPerchTower to the roofs of houses to let dragons land there.

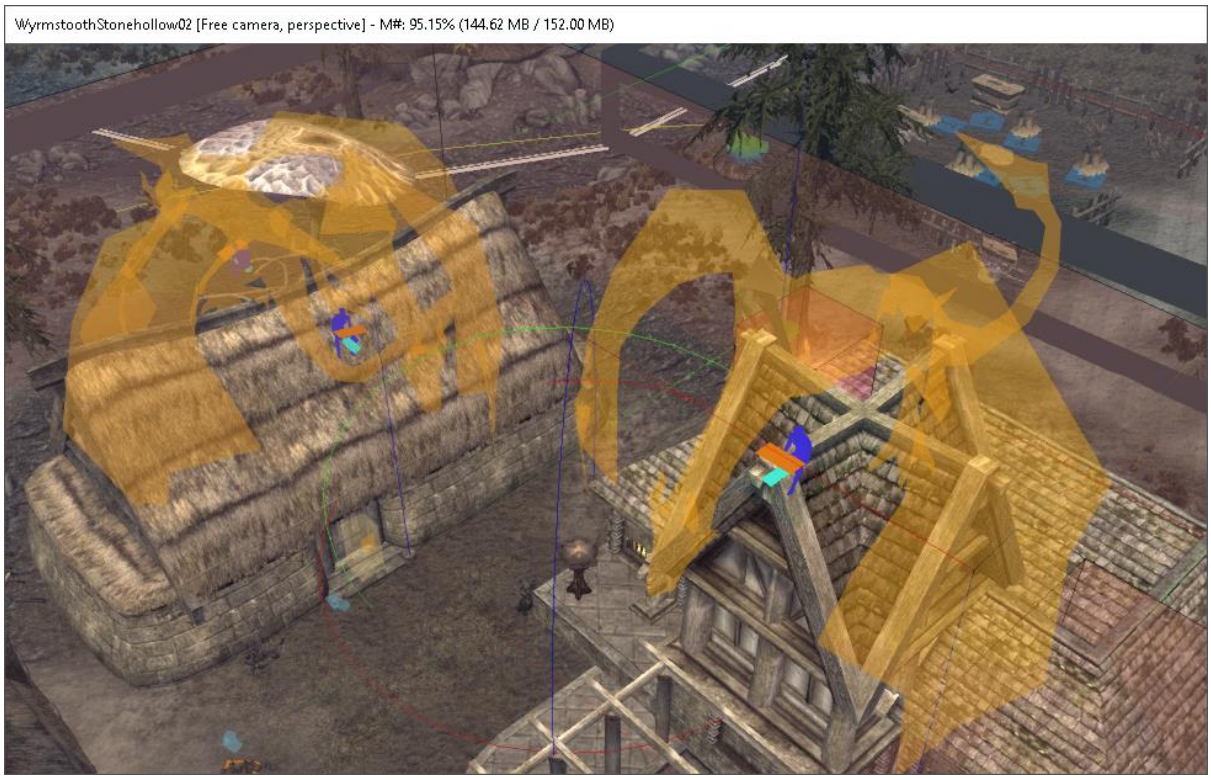


Figure 317 - DragonPerchTower added to the roofs of houses.

The screenshot below shows an example of DragonPerchRockL02 in an exterior cell.



Figure 318 - An example of DragonPerchRockL02 added to an exterior cell.

When placing dragon markers, it's important to be mindful of where the dragon's feet are going to be positioned when a dragon uses the marker. You'll want to avoid sinking the marker too far into the surface of an object below it so dragons don't clip through.

You can also find a few more idle markers listed under Furniture. For example, the screenshot below shows SoldierWallIdle. This marker can be placed against walls, wherever you want NPCs to stand in place against a wall with their arms folded.

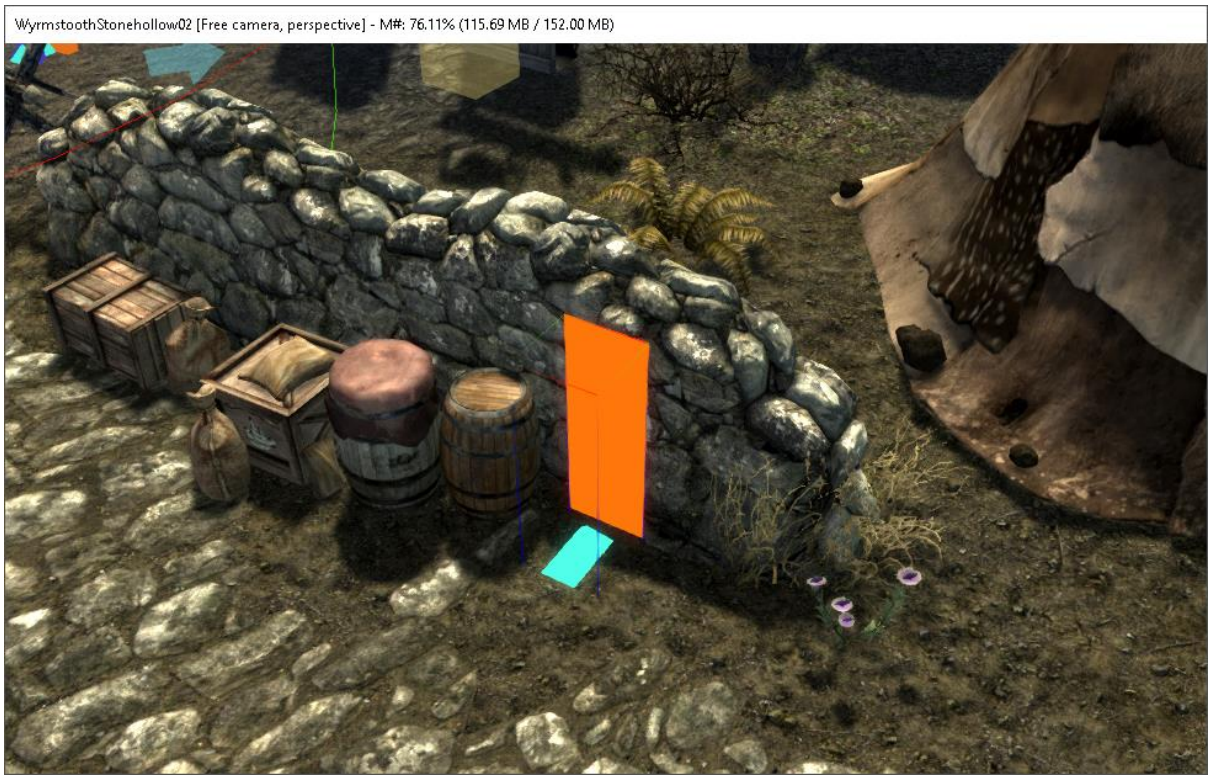


Figure 319 - SoldierWallIdle marker.

RailLeanMarker will make NPCs lean on a railing in front of them. Use the orange box to gauge where the NPC's hands will go when they use this marker.



Figure 320 - RailLeanMarker.

The three markers selected in the render window in the screenshot below are Wounded01FloorMarker, Wounded02FloorMarker and Wounded03FloorMarker.

They can be used to have an NPC lay on the ground and act wounded.

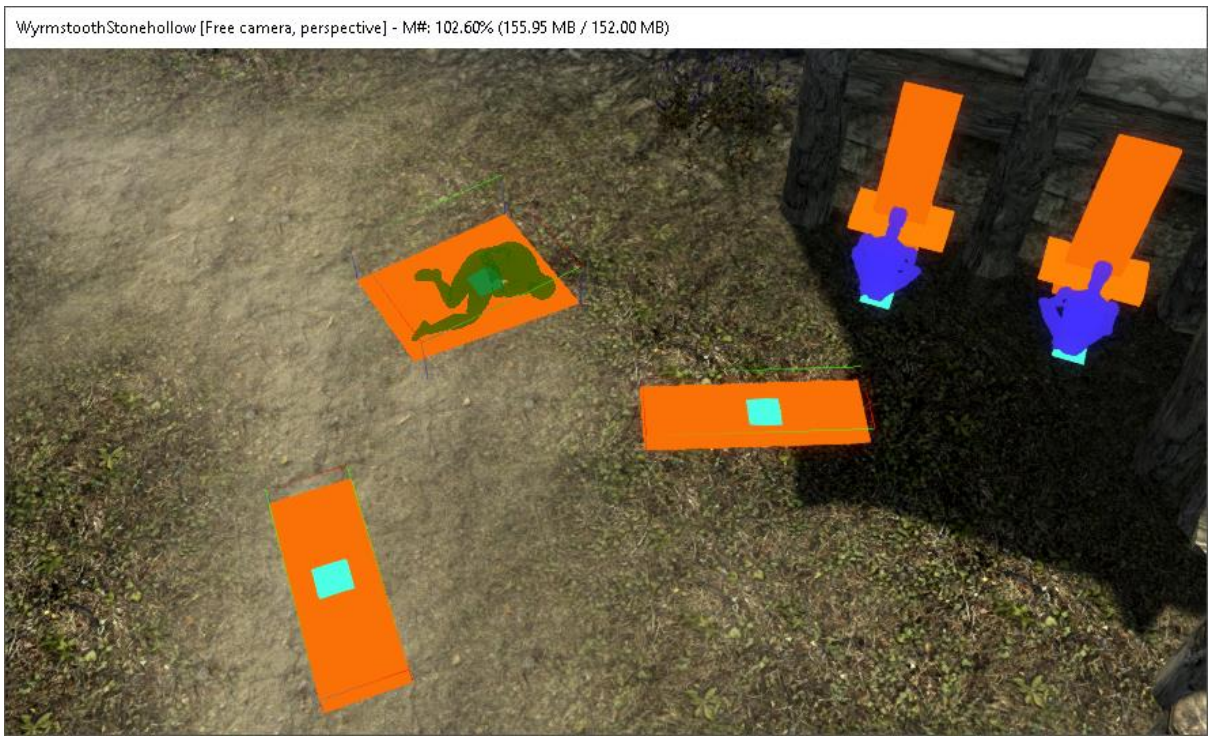


Figure 321 - Wounded markers.

At the top right we can also see a HammerWallMarker which makes NPCs play a wall hammering animation.

Animals can also use idle markers. Here's an example of DogLayingDownIdleMarker which can be used to make dogs lay down at a specific spot.



Figure 322 - DogLayingDownIdleMarker.

Another common one you'll see in dungeons is CreatureAlcoveMarker. Draugr can be linked to this marker to make them stand still in an alcove to ambush the player.

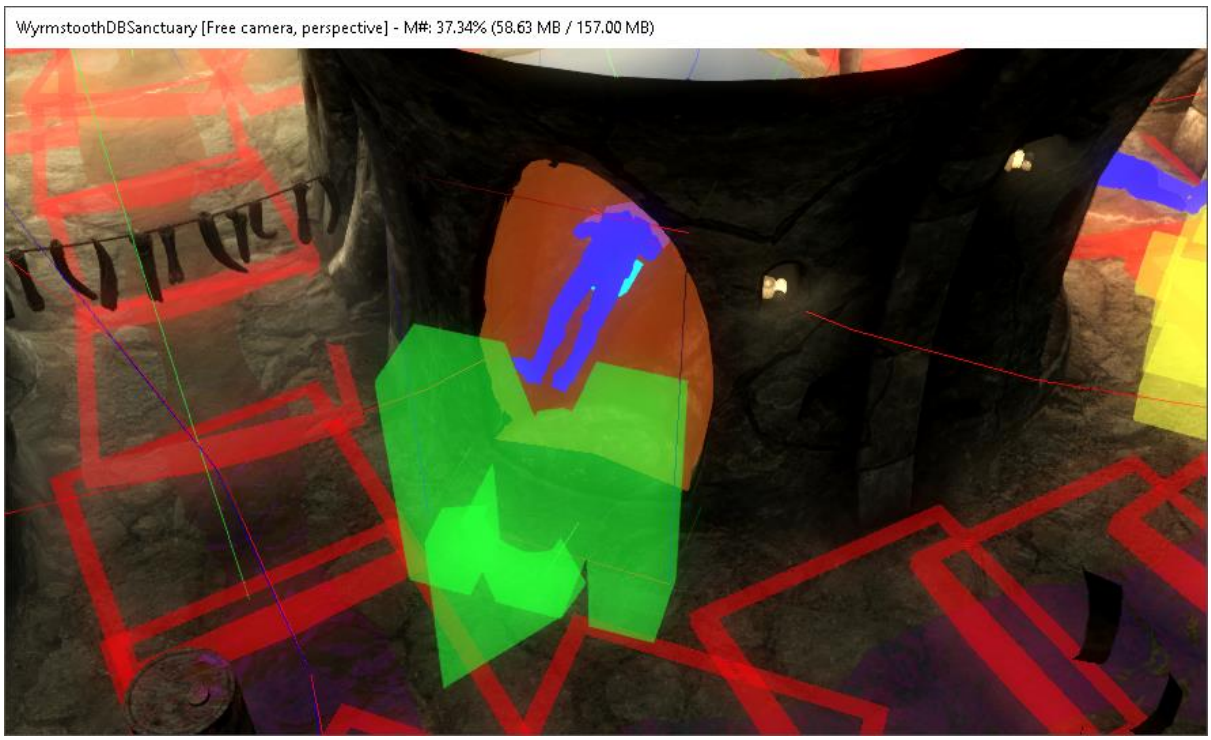


Figure 323 - CreatureAlcoveMarker.

An alternative to CreatureAlcoveMarker is CreatureAlcoveBgMarker which makes draugr lay down on a flat surface instead.

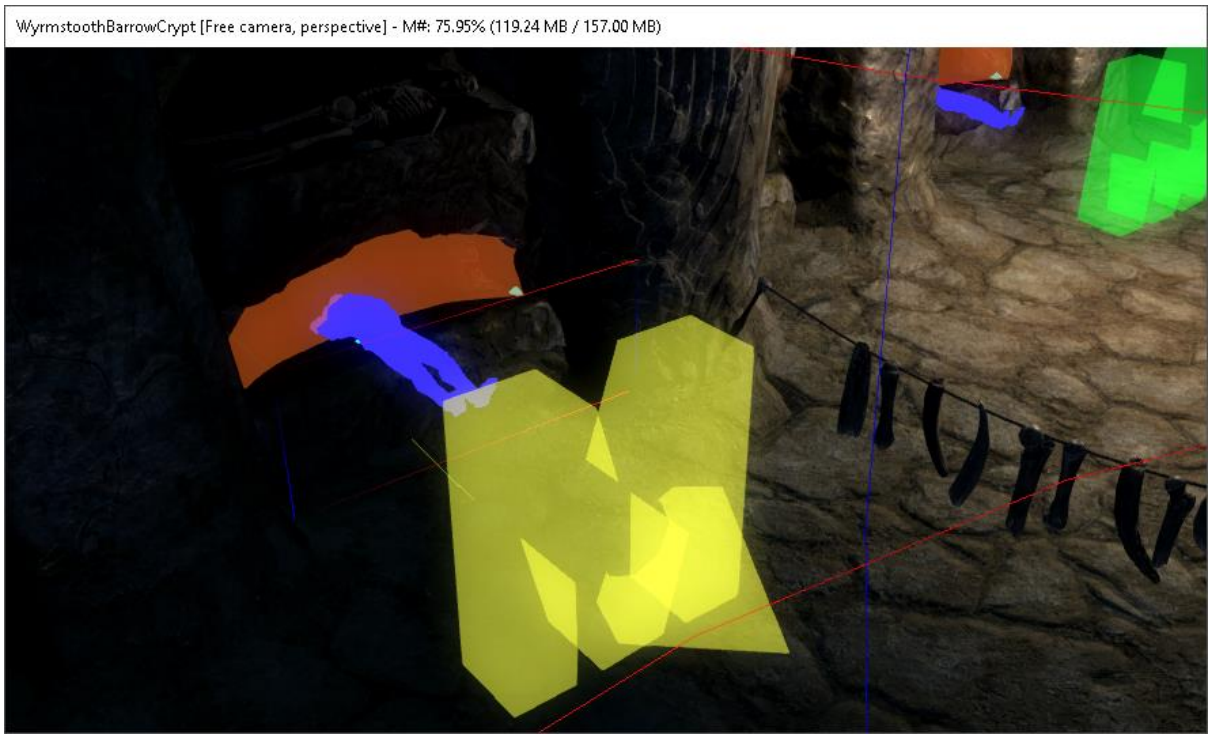


Figure 324 - CreatureAlcoveBgMarker.

For more furniture markers, just filter the Object Window by 'marker' or 'idle'.

CREATING A CUSTOM IDLE MARKER

Let's say we wanted to make an NPC play a specific animation that's not defined in one of the existing markers. We'll need to create our own.

In the Object Window, go to Miscellaneous > IdleMarker.

Right-click on an existing marker and select New.

Set an ID for this marker. In the example below, I just called it CiceroDanceMarker and added the 'WT' prefix.

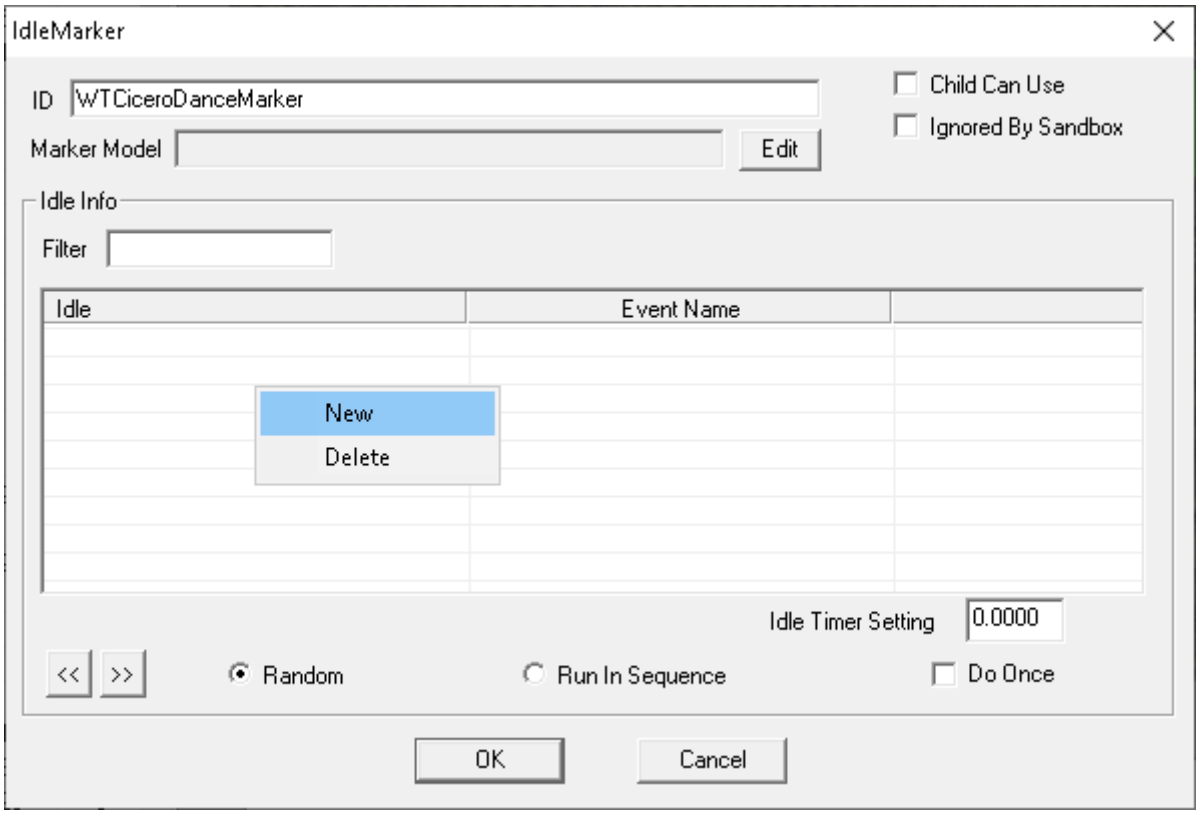


Figure 325 - Created a new idle marker.

Right-click in the Idle Info list and select New.

Search for the idle animation you want an NPC using this marker to play. For my example I'm going to add IdleCiceroDance1. To add an idle animation, double-click on it.

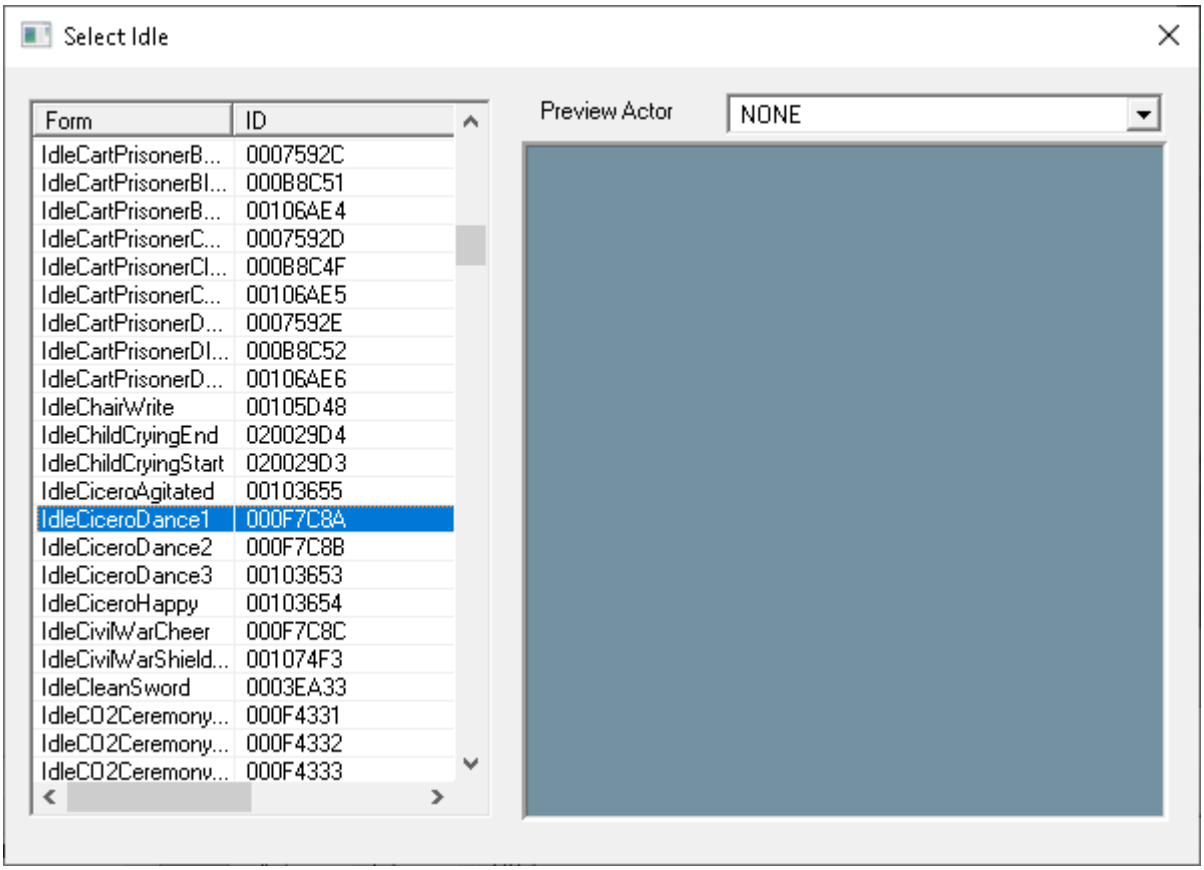


Figure 326 - Adding IdleCiceroDance1 to our new idle marker.

Confirm the idle animation has been added.

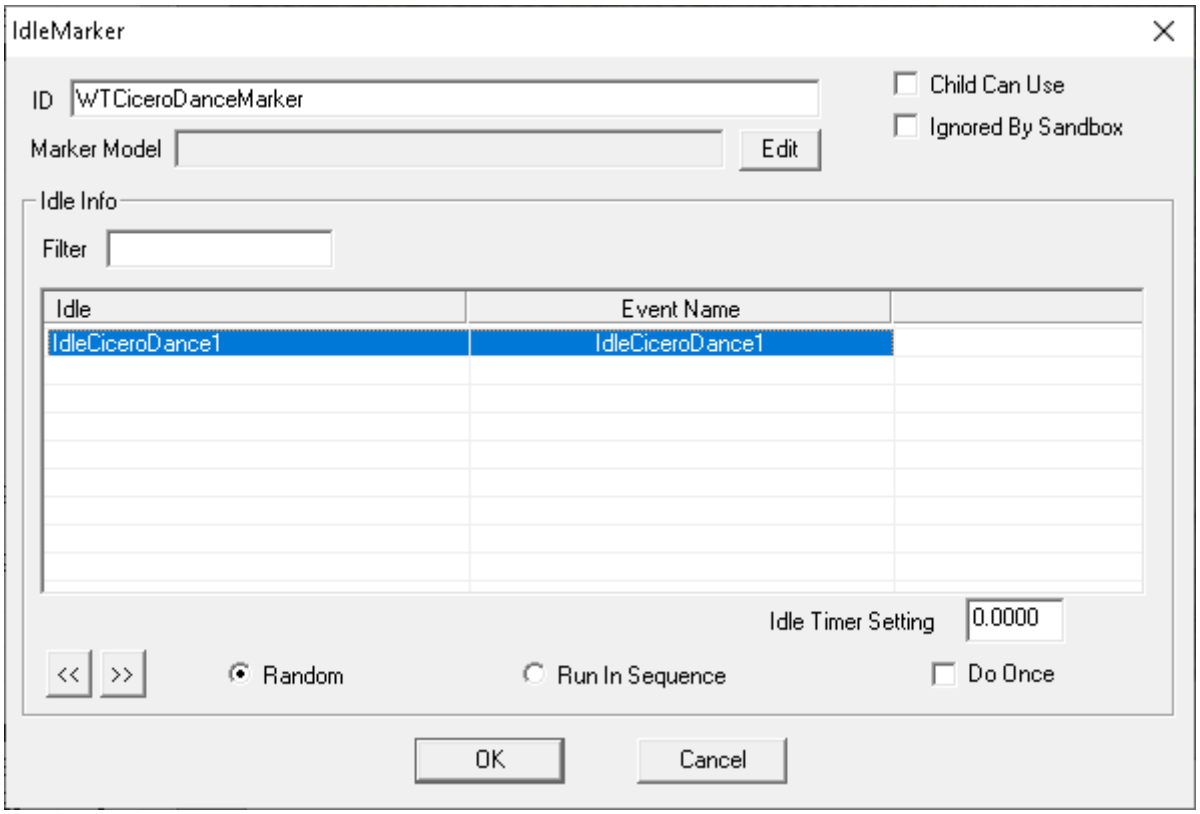


Figure 327 - New animation added to our new idle marker.

I also added IdleCiceroDance2 and IdleCiceroDance3 by repeating those steps.

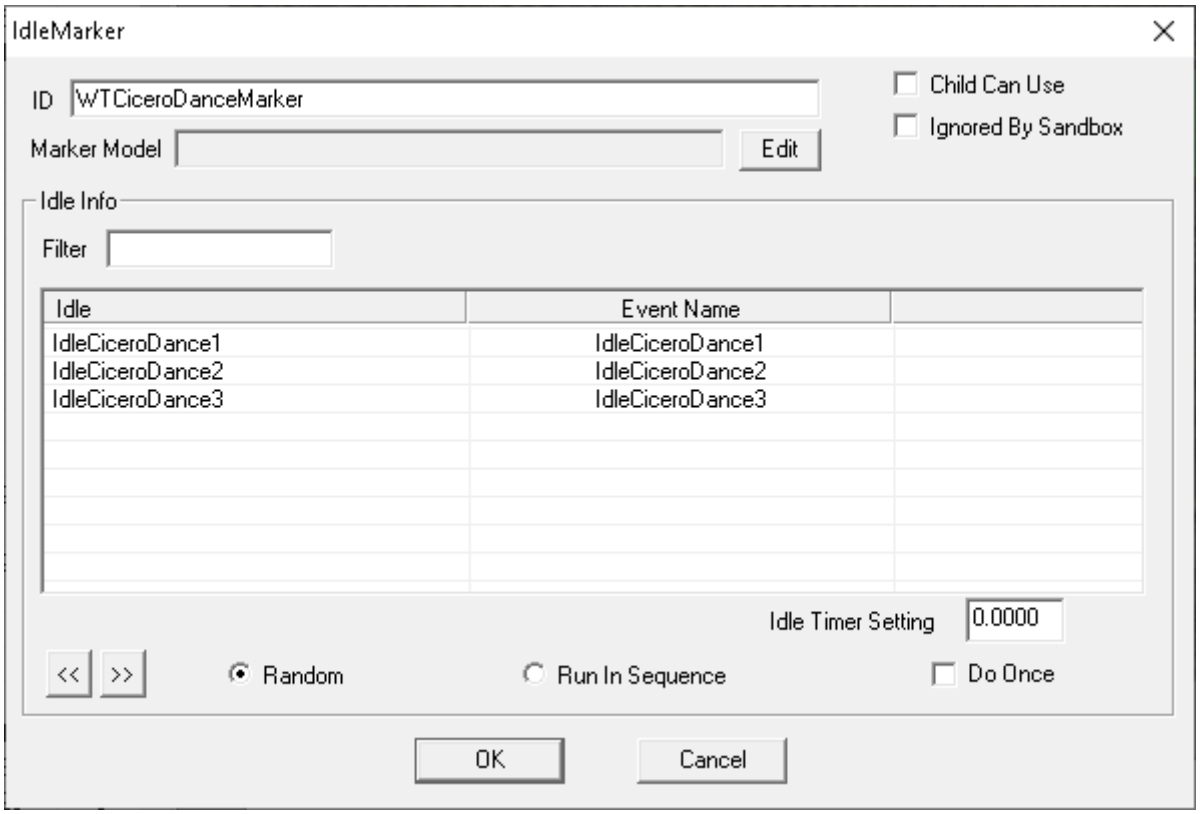


Figure 328 - Added a couple more animations to our new idle marker.

Click OK to close out of the IdleMarker properties.

We can now place our idle marker in the world.

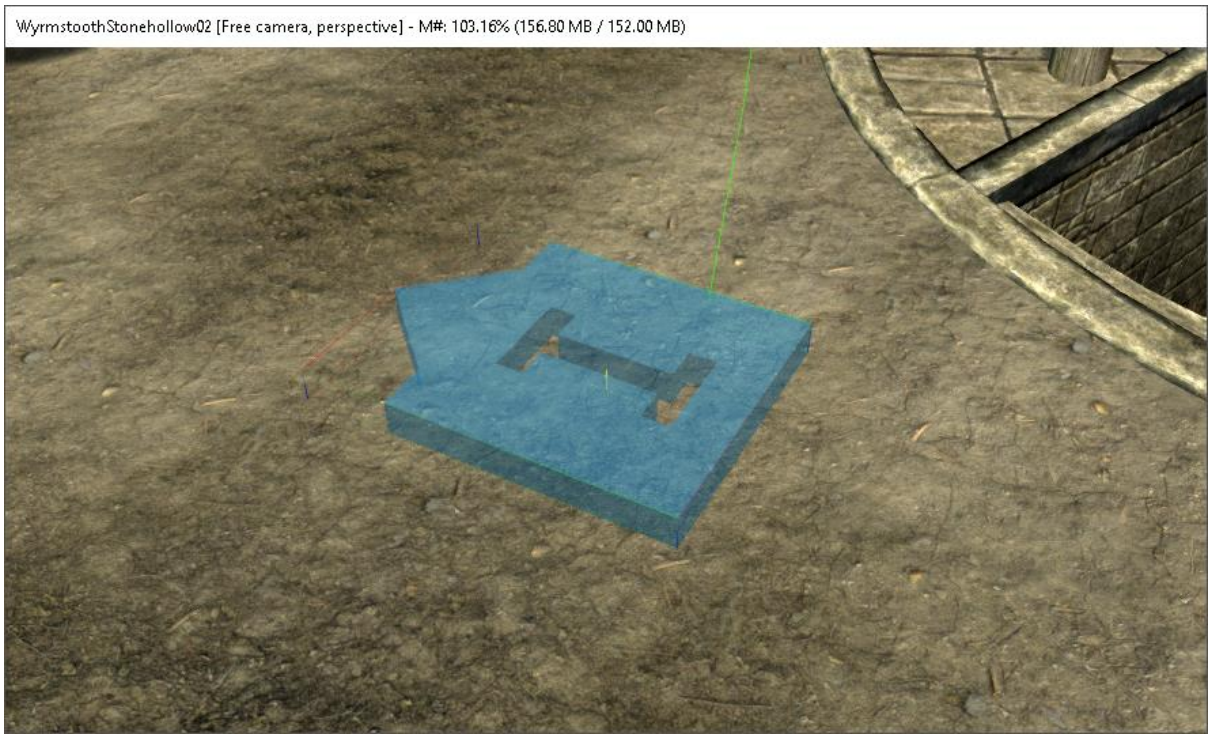


Figure 329 - Custom idle marker placed in our world space.

ADDING MAP MARKERS

Map markers mark specific locations on the world map.

In the Object Window, go to WorldObjects > Static. Filter by 'mapmarker'.

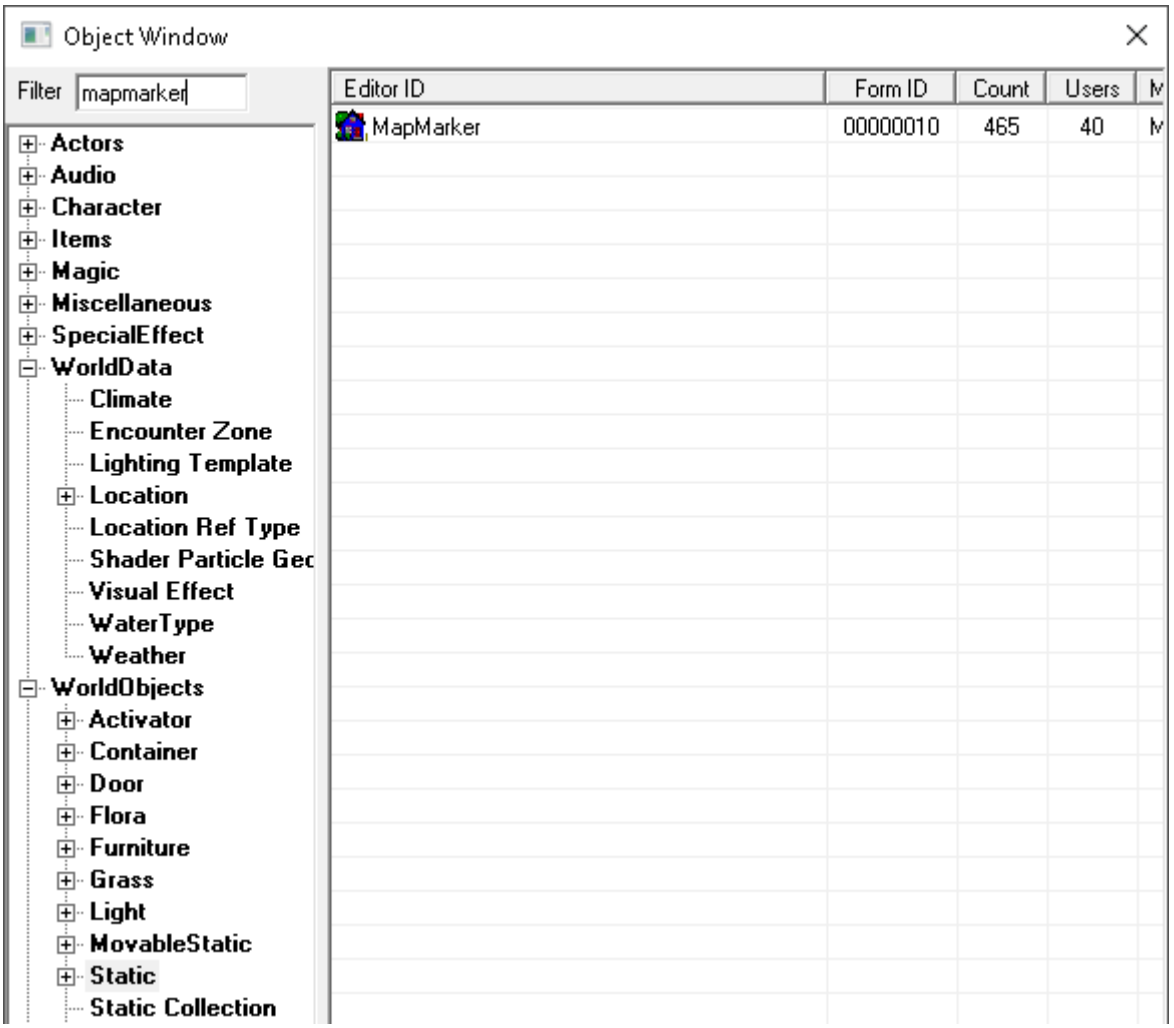


Figure 330 - Adding a map marker.

Drag and drop the MapMarker object into the render window at a spot that's central to your location.

The map marker will appear as a green square with an 'M' in it.



Figure 331 - The M map marker added to a location.

Double-click or right-click on the map marker and select Edit.

Go to the Marker Data tab.

The Name field specifies the name of the location that players will see on the map screen and when approaching the location for the first time.

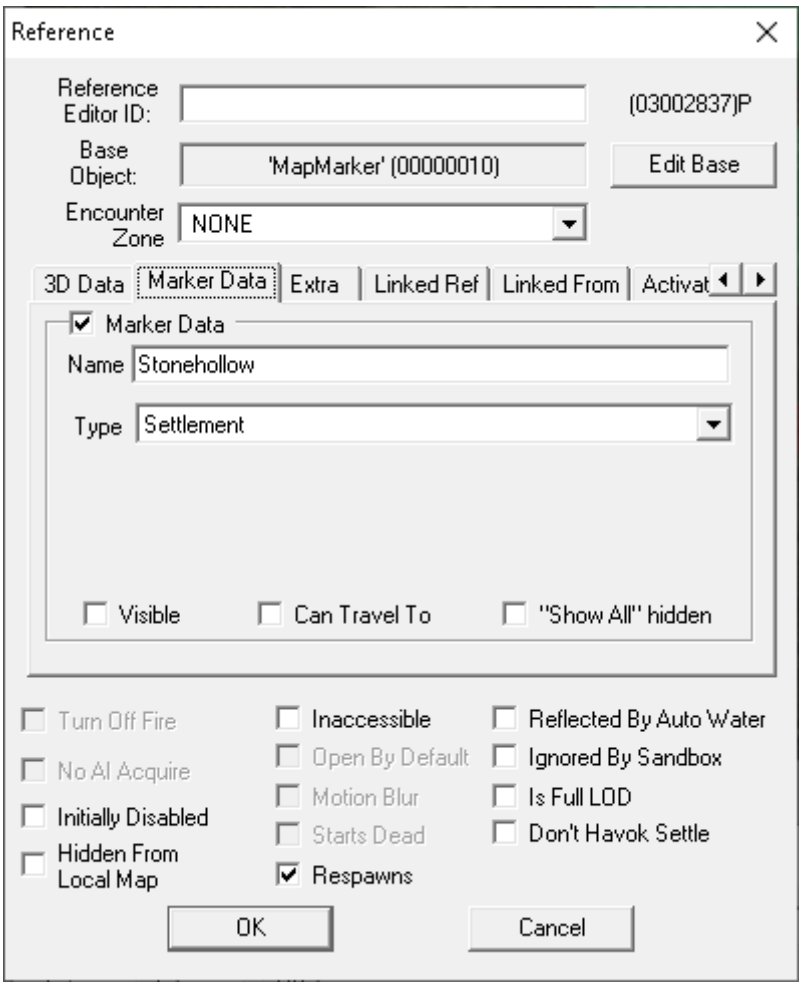


Figure 332 - Map marker data.

The Type determines the map marker icon that'll appear on the world map and on the player's compass.

Ticking Visible will make the marker appear in your map screen as soon as the game loads without needing to travel to it. If you tick Can Travel To you'll be able to fast travel to this location immediately assuming Visible is also ticked.

Go to the Extra tab.

The Radius field determines how far away from the map marker the player needs to be before they'll see the '<Location Name> discovered' message.

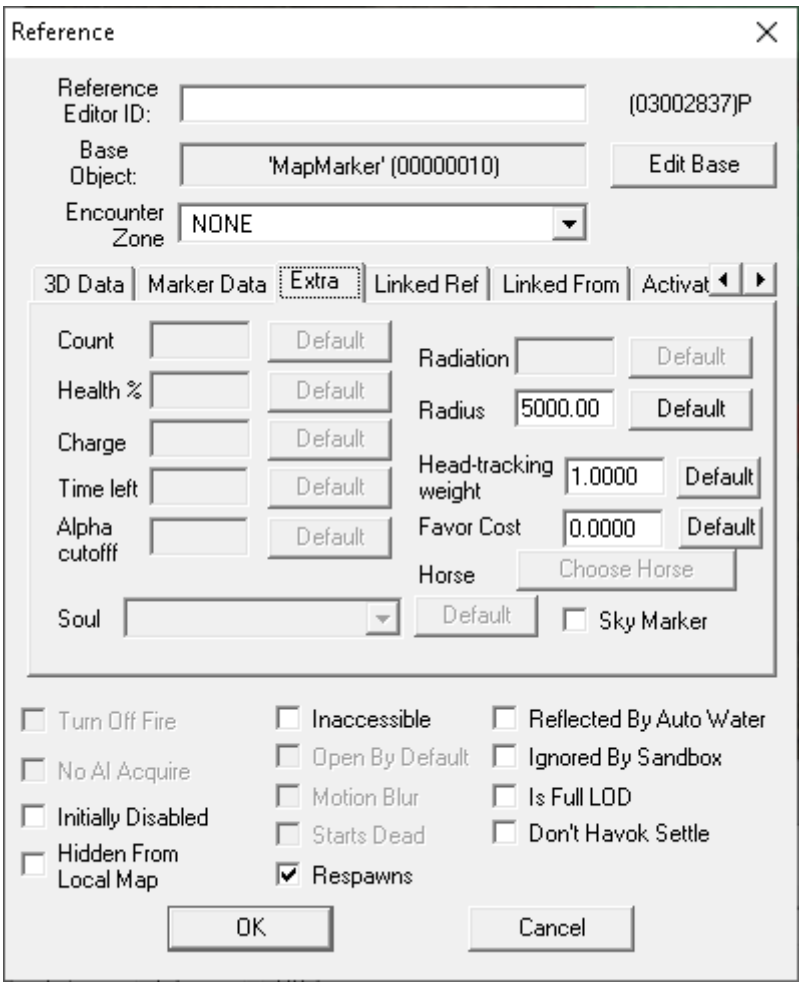


Figure 333 - Setting the map marker radius.

I increased this value to 5000.00 so players see the message as they pass through the town gate which is about 5000 units away from the map marker.

Under the Location Ref Type tab we can set the Location Reference Type to MapMarkerRefType.

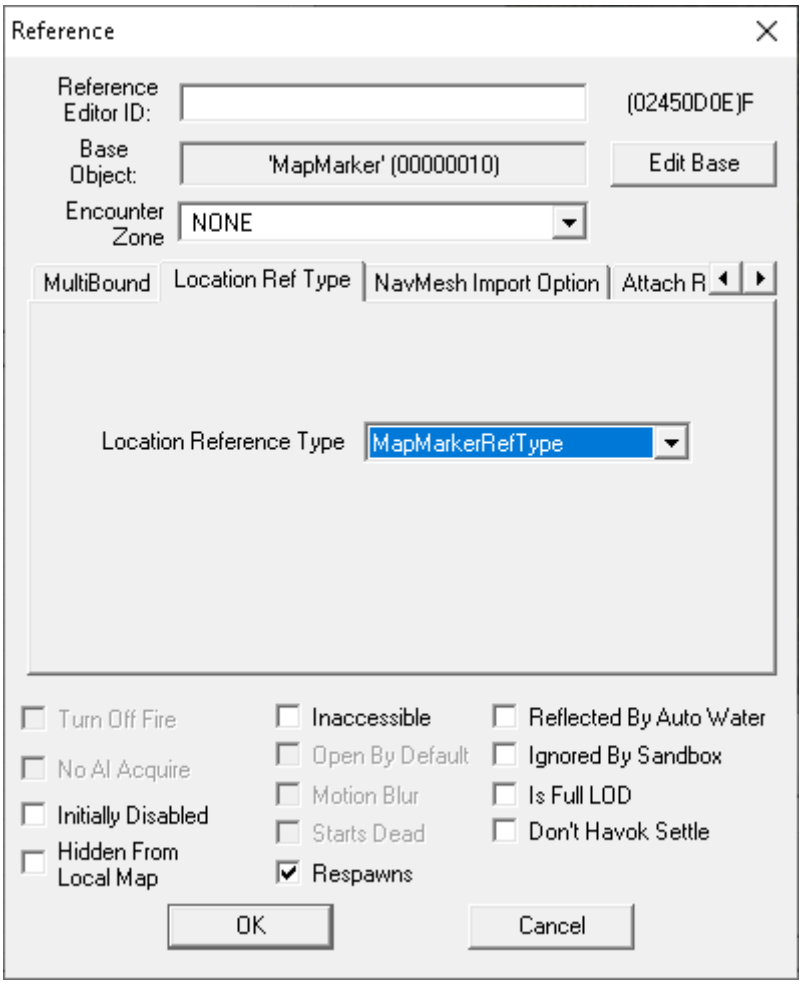


Figure 334 - MapMarkerRefType.

I'll be covering Common Radiant Markers shortly, but setting this may allow certain radiant side quests to use the location containing this map marker, depending on the types of keywords assigned to the location. Specifically Jarl bounty side quests such as Kill the Giant or Kill the Bandit Leader.

Click OK to close out of the map marker reference properties.

The map marker radius can be seen in the render window as a yellow circle.



Figure 335 - Map marker radius.

ADDING COMMON RADIANT MARKERS

Radiant markers are used by the radiant quest system to mark where quest objects or quest-related NPCs may be dynamically placed.

In this section I'm going to go over the most commonly used markers, how they're used and where they should be placed.

For setting up these markers we'll mostly be using XMarker and XMarkerHeading.



Figure 336 - XMarker and XMarkerHeading.

They're basically interchangeable, however XMarkerHeading is usually used when an object or NPC needs to be placed facing a specific direction, hence the arrow.

You can find them both in the Object Window under WorldObjects > Static. Just filter by 'xmarker'.

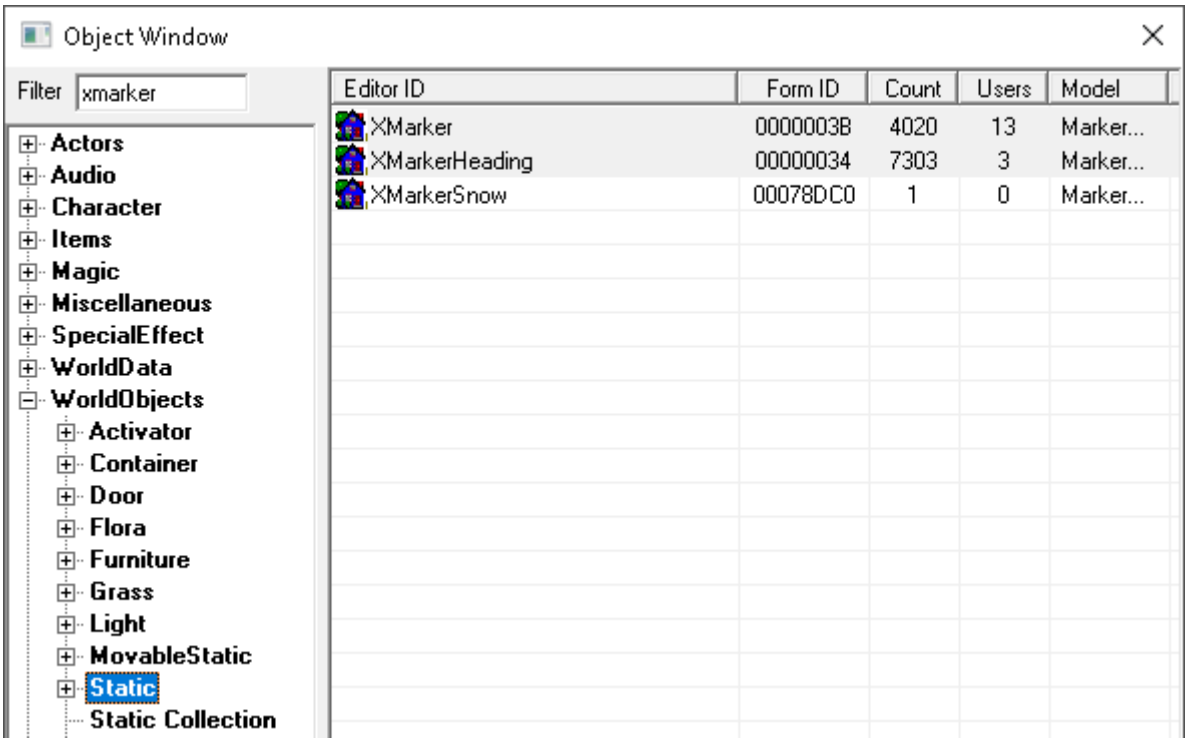


Figure 337 - XMarkers in the Object Window.

To set an XMarker as a specific type of radiant marker, double-click or right click on it and select Edit, then go to the Location Ref Type tab.

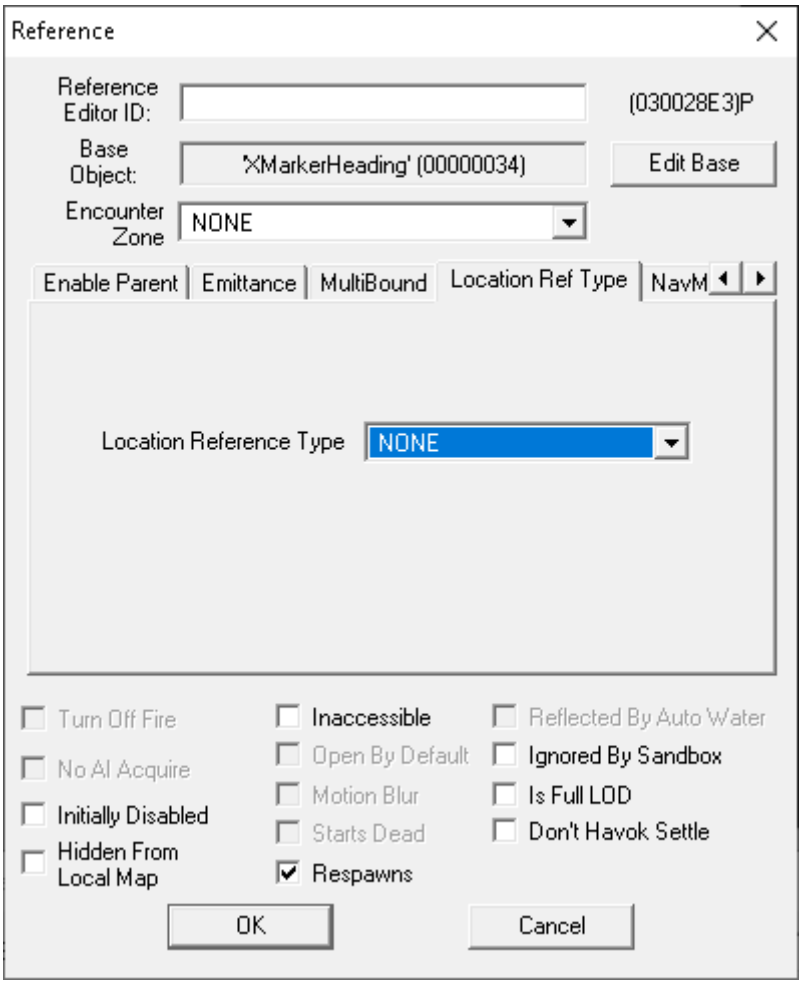


Figure 338 - Location Ref Type tab.

Set the radiant marker type by selecting it in the Location Reference Type drop-down.

All radiant markers belonging to a specific location can be found under the Location Ref Types tab in the location's properties.

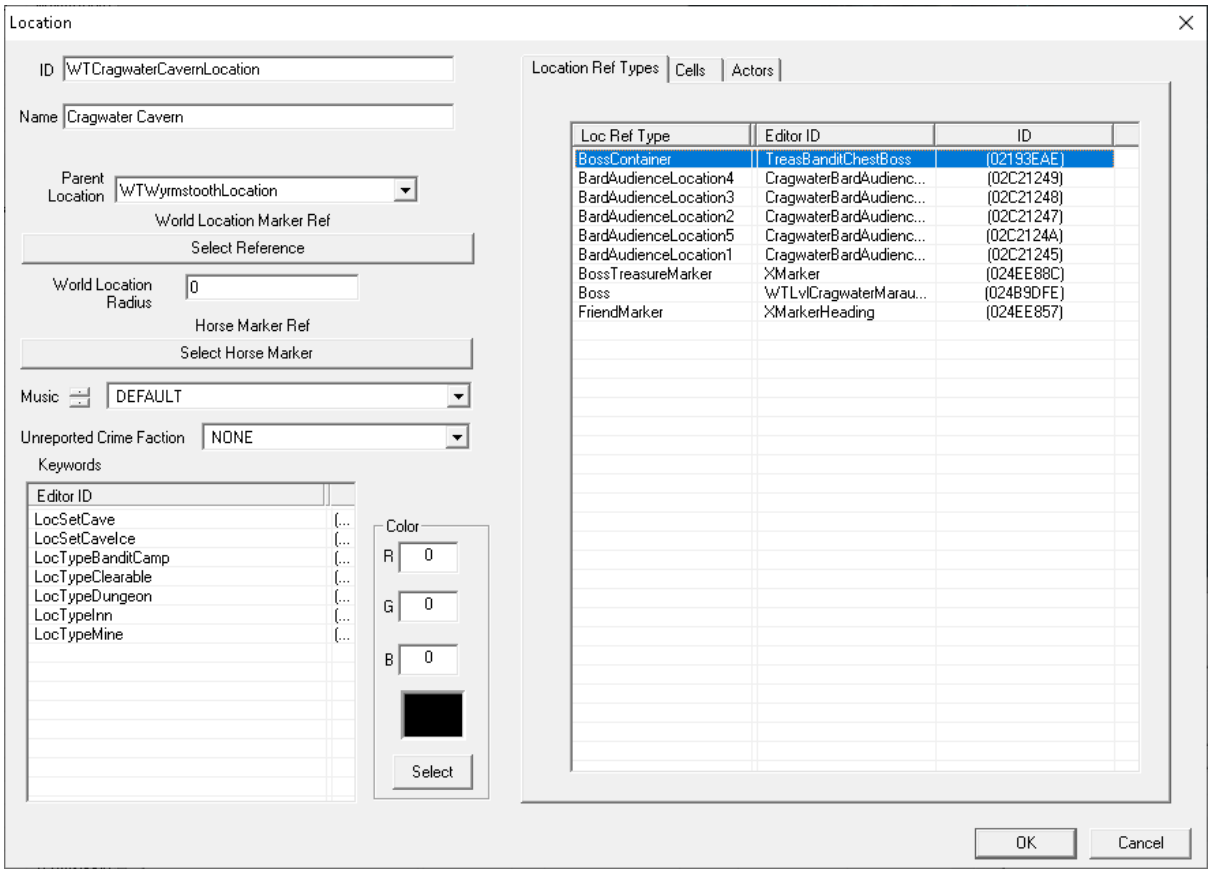


Figure 339 - Location Ref Types tab in Location properies.

LocationCenterMarker is generally used to mark the centre point of a location. Some quest packages (e.g.: random encounters) may make an NPC sandbox around this marker, or may spawn NPCs at this marker to wait for the player (e.g.: the courier).



Figure 340 - LocationCenterMarker.

BossTreasureMarker is placed on an XMarker. It's generally used by radiant quests to mark the location of a quest item in your journal.

It's usually paired with BossContainer which is assigned to a treasure chest as per the screenshot below. This is the container that a quest item may be spawned into.

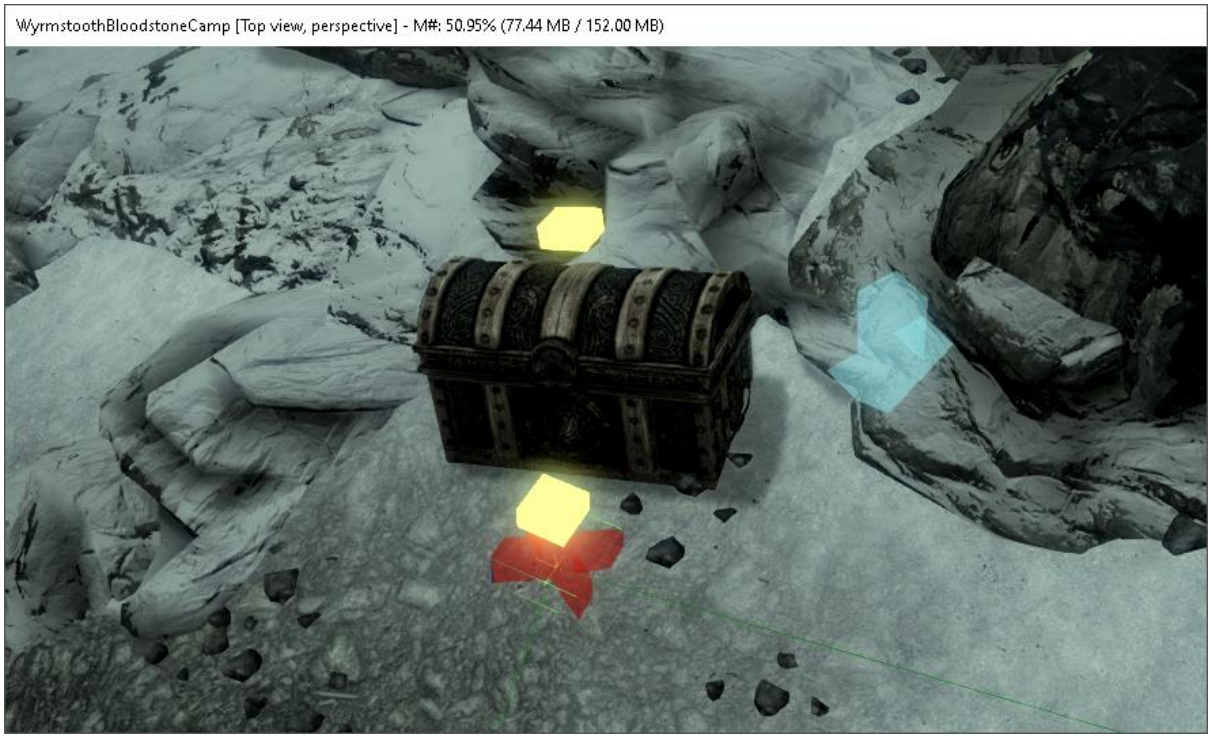


Figure 341 - BossTreasureMarker and BossContainer.

Boss is added to an actor that needs to be killed before a location can be considered cleared. A location can have multiple actors flagged as Boss, meaning they all need to be killed before a location is cleared.

Important: The location needs to have the **LocTypeClearable** keyword assigned to it.

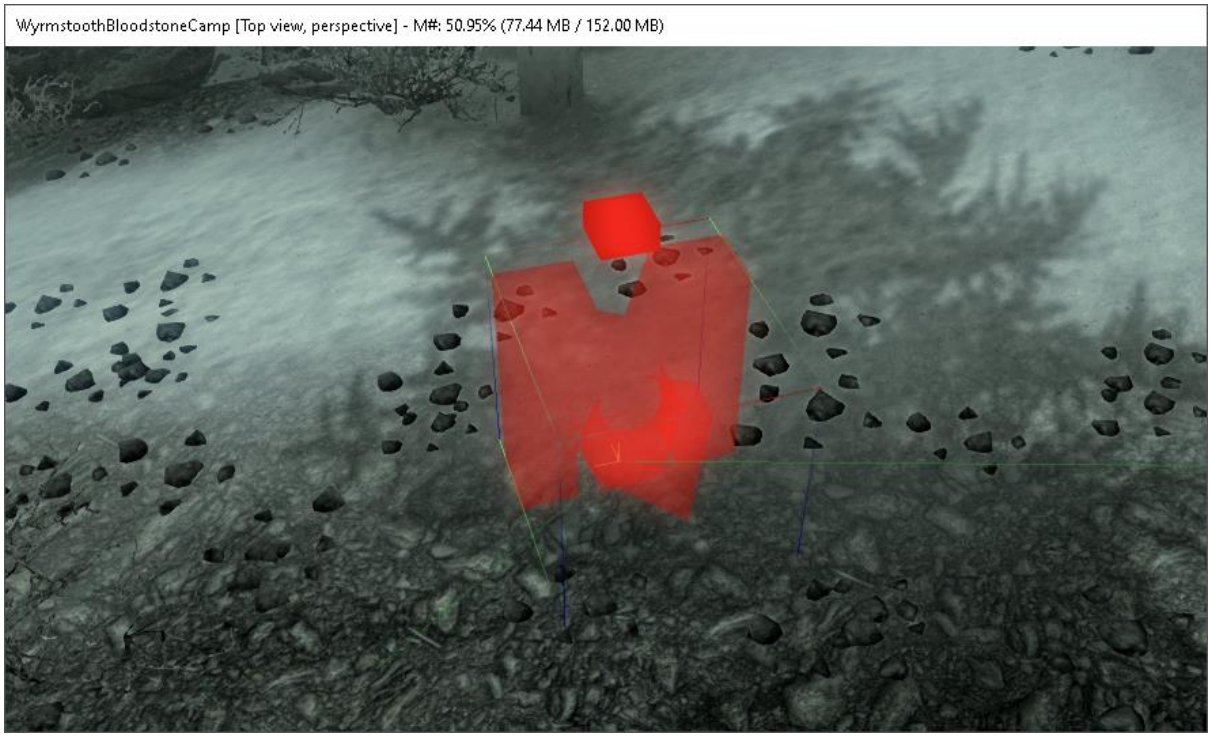


Figure 342 - Boss.

Important: Boss should only be assigned to actors.

BardAudienceLocation1, BardAudienceLocation2, BardAudienceLocation3, BardAudienceLocation4 and BardAudienceLocation5 are used to mark 5 spots where NPCs may stand when a bard begins performing.

Important: The location a bard performs at needs to have the **LocTypeInn** keyword assigned to it otherwise the performance will fail.

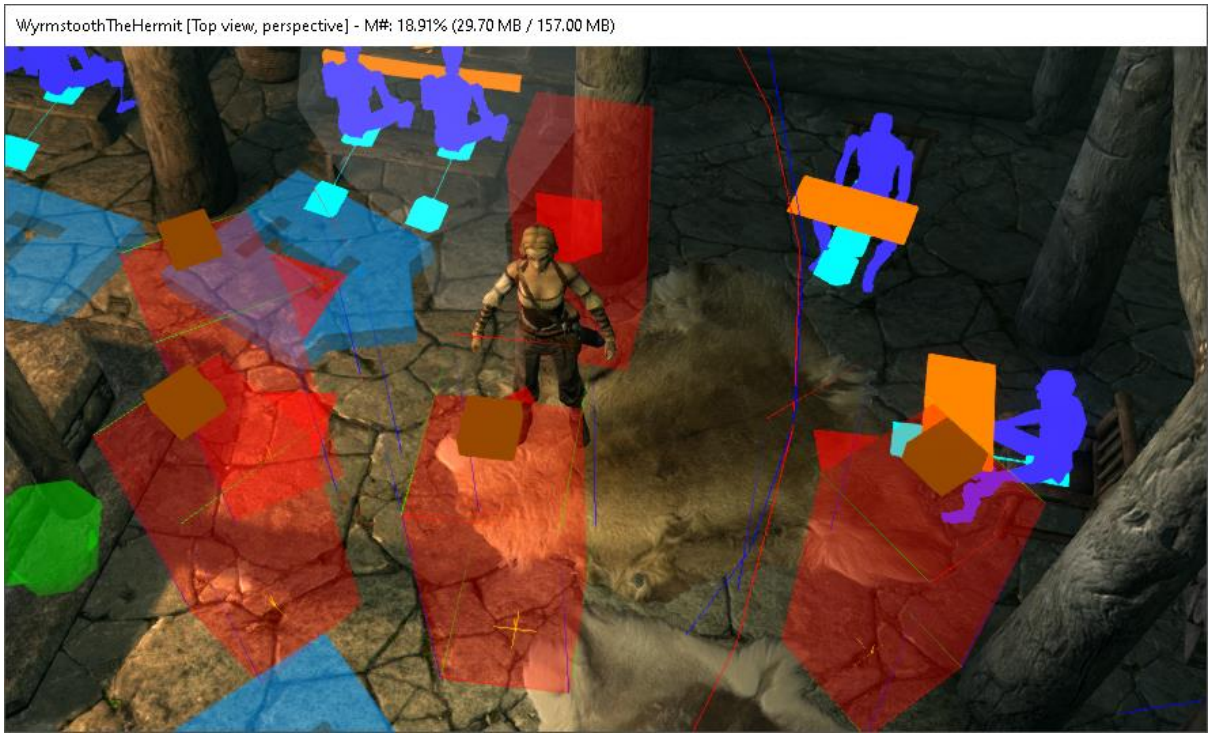


Figure 343 - BardAudienceLocation.

CaptiveMarker is used by the Companions Rescue Mission side quests. It marks the spot where an NPC is being held captive, typically in some sort of a jail cell. The NPC will stand at this spot until interacted with by the player.



Figure 344 - CaptiveMarker.

FriendMarker is used by radiant quests involving NPCs that need to wait for the player at a specific location.

Generally they're placed at a safe spot, such as near the entrance to a dungeon or a camp situation away from enemy patrols.



Figure 345 - FriendMarker.

OutsideEntranceMarker is typically placed at dungeons near the door marker in the world space cell.

NPCs that need to wait outside the entrance to a dungeon may use this marker.

This marker is also used as the spot to spawn dragons at when the next dragon event is scheduled to occur. The more you know.



Figure 346 - OutsideEntranceMarker.

Similarly, InsideEntranceMarker is typically placed in an interior cell near the door marker leading outside.

NPCs that need to wait inside the entrance to a dungeon may use this marker.



Figure 347 - InsideEntranceMarker.

TGRWealthMarker01, TGRWealthMarker02 and TGRWealthMarker03 are used to spawn items for the player to steal during the Thieves Guild The Sweep Job side quests.

Typically they're placed on shelves or on tables where there's enough room for a goblet or vase to be placed.

Important: Locations marked by the Thieves Guild should have the **TGWealthyHome** keyword.



Figure 348 - TGRWealthMarker.

TGRWealthyHomeChest is used by Thieves Guild side quests to spawn items for the player to steal.

Important: Locations marked by the Thieves Guild should have the **TGWealthyHome** keyword.

Also, you should only flag containers with TGRWealthyHomeChest.



Figure 349 - TGRWealthyHomeChest.

HouseMainDoorRefType is applied to the front door object of a dwelling. It's used by the Companions Animal Extermination side quest as well as the marriage side quest.

There's also a HouseBackDoorRefType but I don't think it's used by anything.

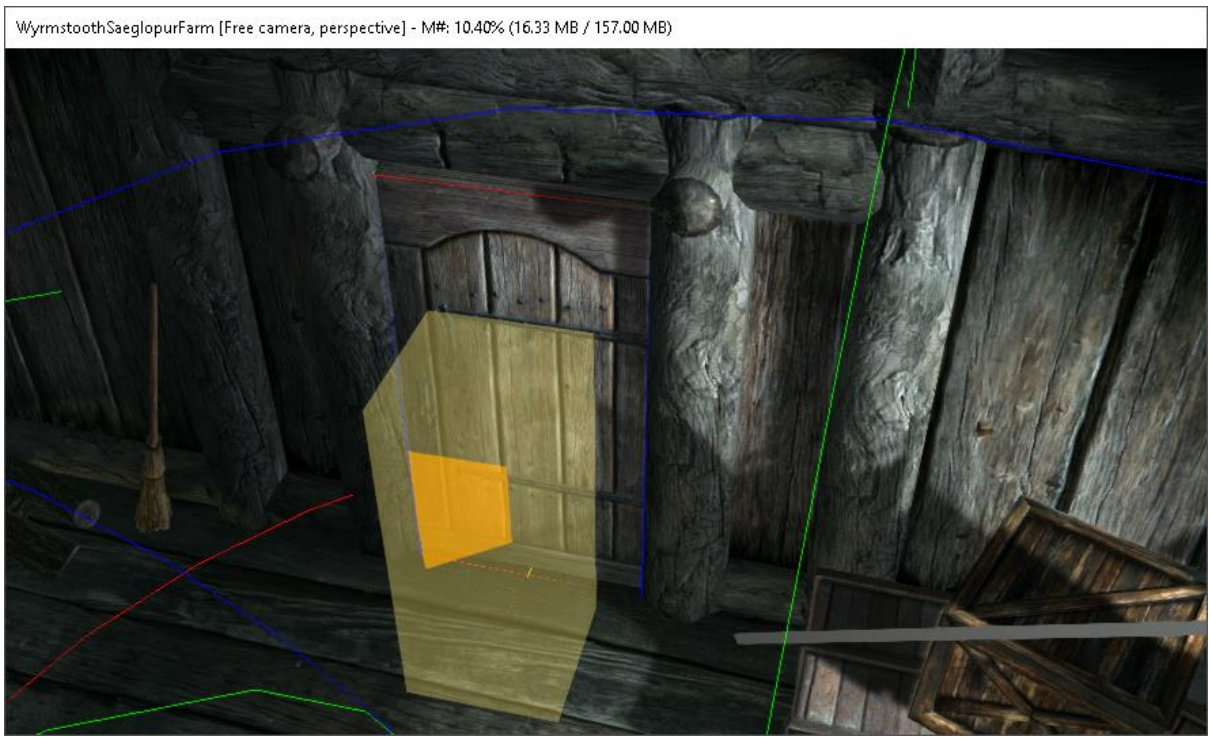


Figure 350 - HouseMainDoorRefType.

HouseContainerRefType is applied to a chest in a dwelling where an item may spawn (not related to Thieves Guild side quests). It's specifically used by the side quest That Was Always There.

Important: You should only flag containers with HouseContainerRefType.

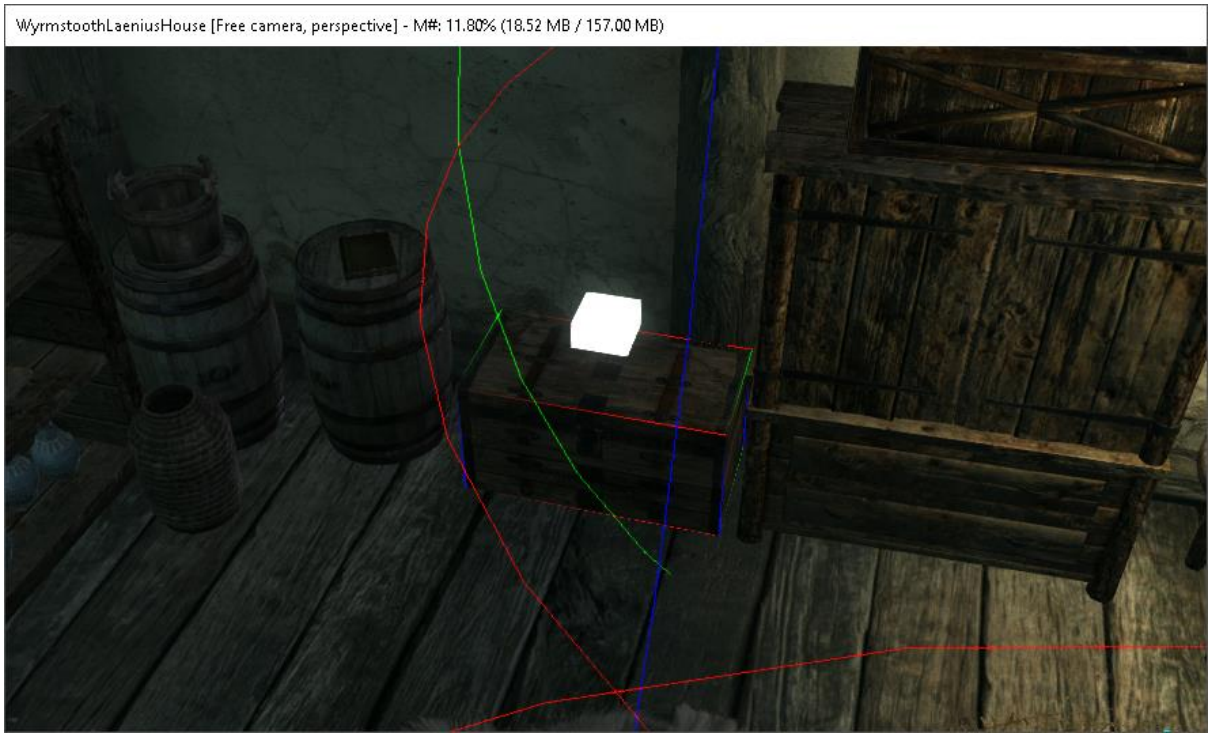


Figure 351 - HouseContainerRefType.

LocationEdgeMarker is placed near the border of a cell belonging to a particular location. It's mostly used by a bunch of civil war side quests, however random encounters do use it as well.

It's also used by the Blades Dragon Hunting and the Companions Escaped Criminal side quests.

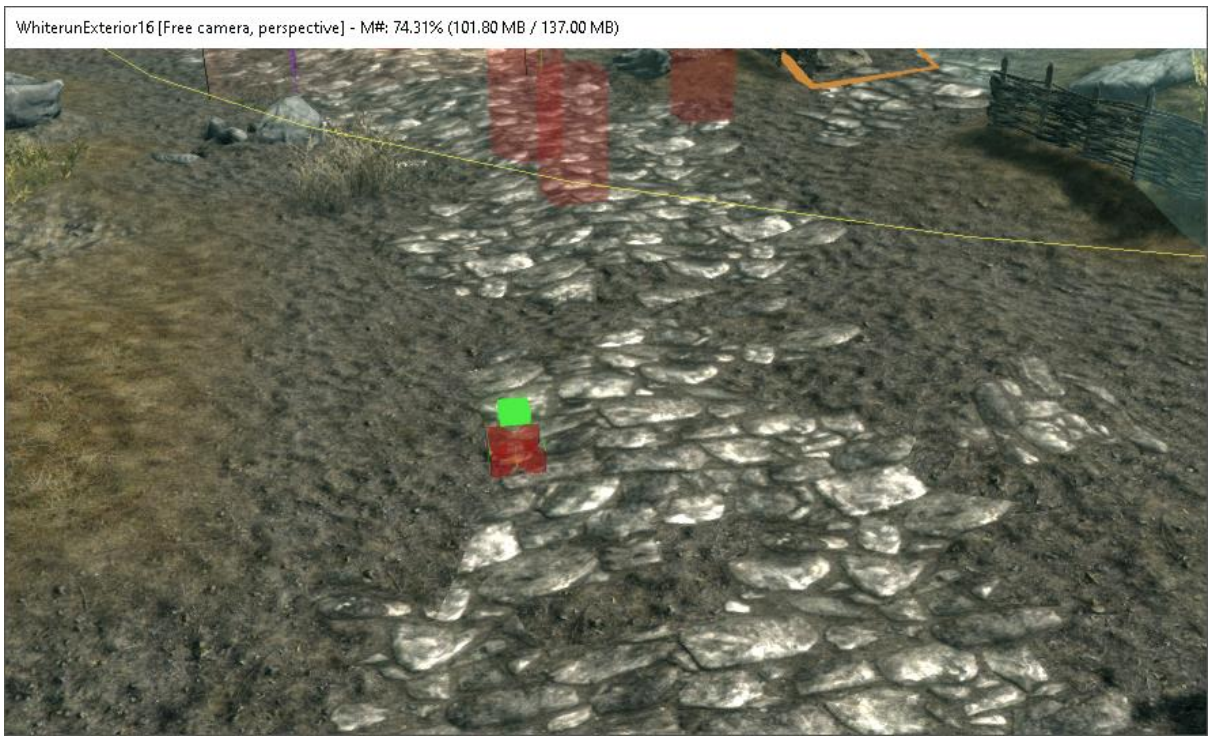


Figure 352 - Location Edge Marker.

CHAPTER 3: SOUNDS AND MUSIC

ADDING SOUNDS

Sound markers are generally used to play looping sounds at specific spots.

If you're looking to add ambient background sounds, see the section on Regions instead.

Sound markers appear as green objects as per the screenshot below:

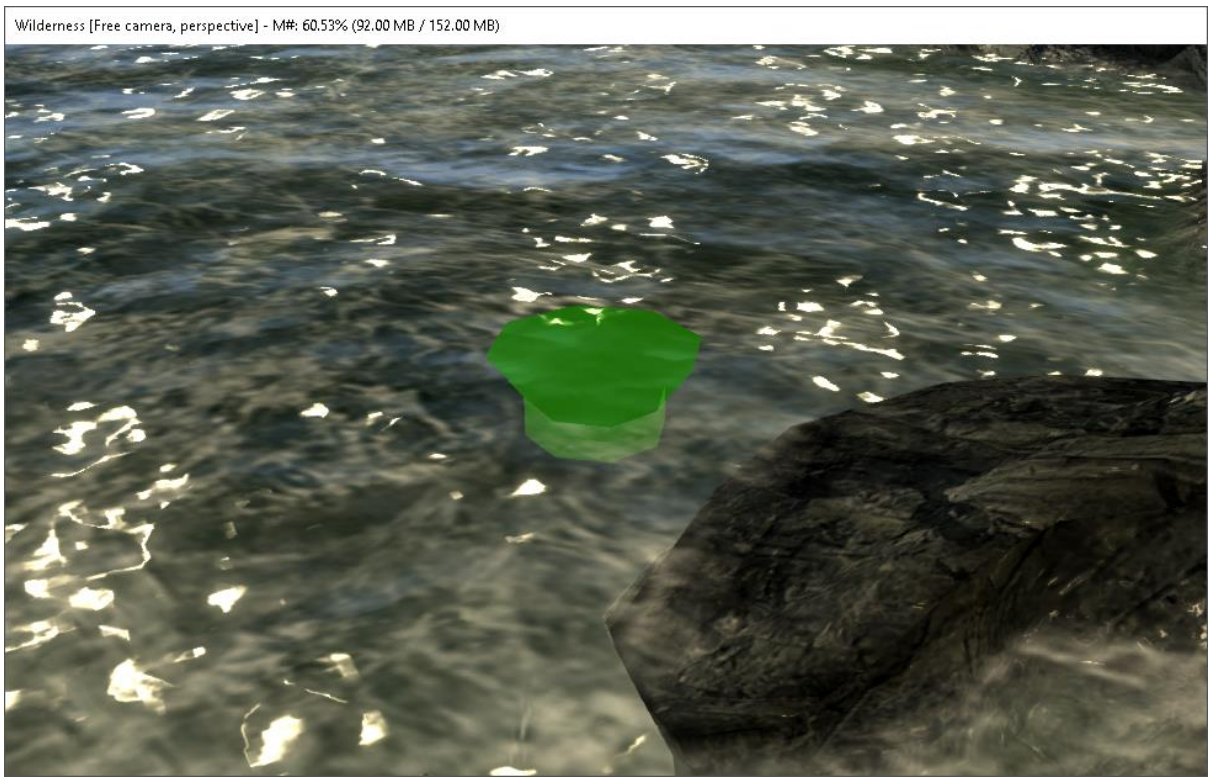


Figure 353 - A sound marker.

To see a list of sounds that you can add to your world space, in the Object Window go to Audio > Sound Marker.

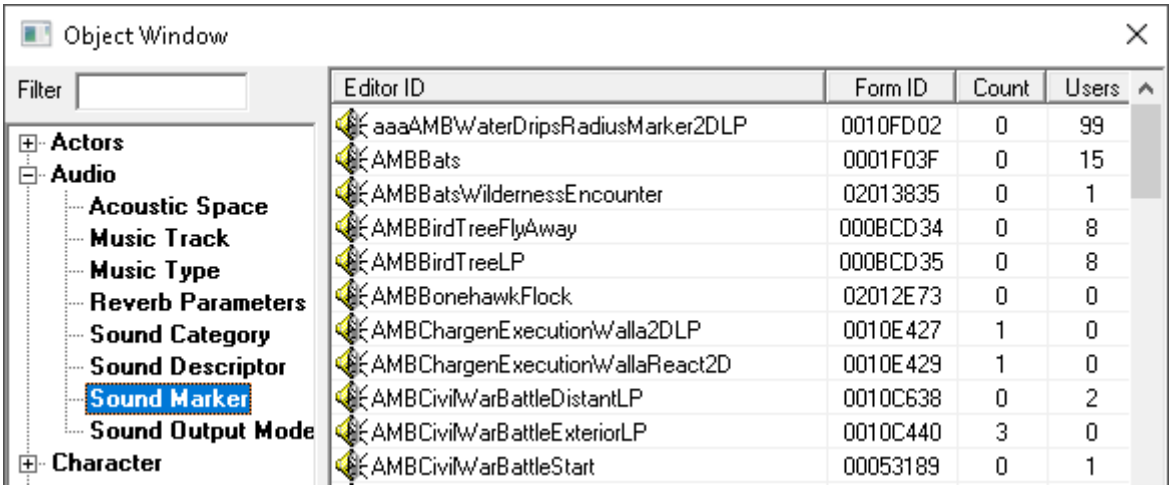


Figure 354 - List of place-able sounds.

To add a sound, drag and drop it from the Object Window into the render window.

For example, in the screenshot below, I added AMBWaterfallRapidsMedium in places where rocks break the surface of the water.

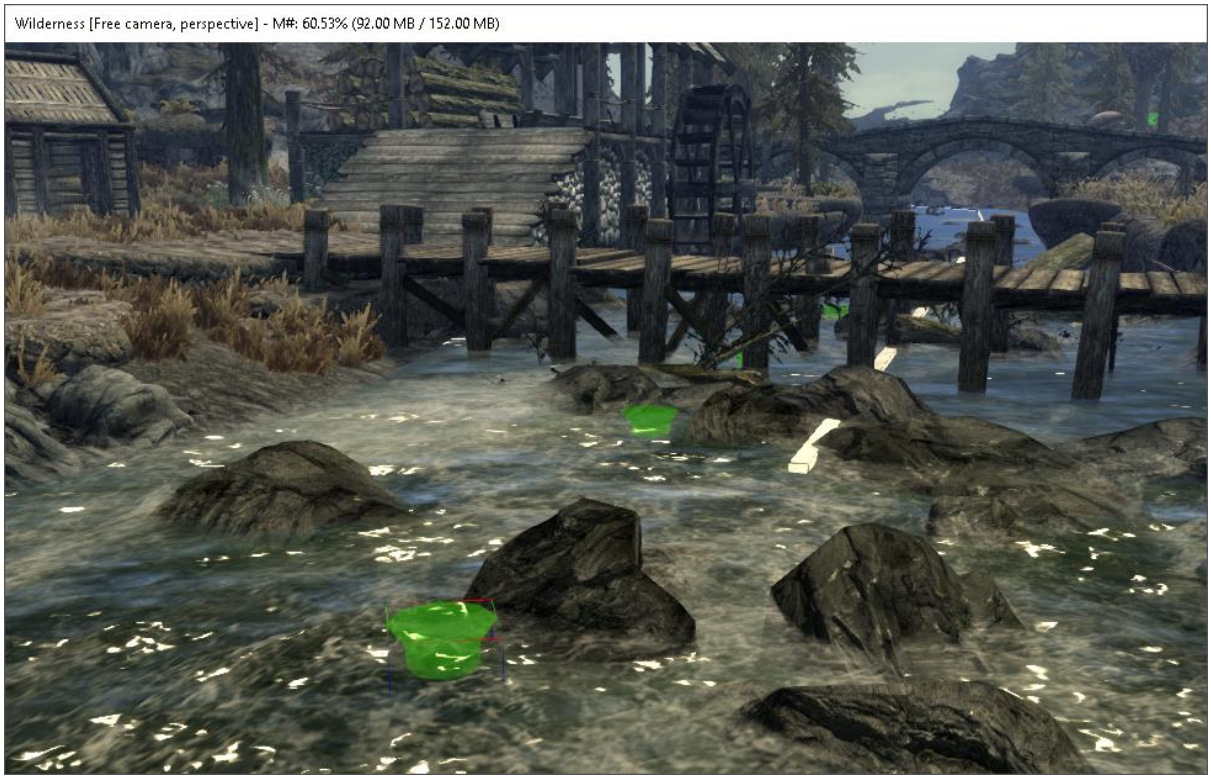


Figure 355 - Sound markers placed along a river.

But what if we wanted to add the sound of running water along a river? Or the sound of crashing waves along a beach. A better option for those cases would be to use a sound emitter.

To create a sound emitter, click on the Sound Emitter button in the toolbar.

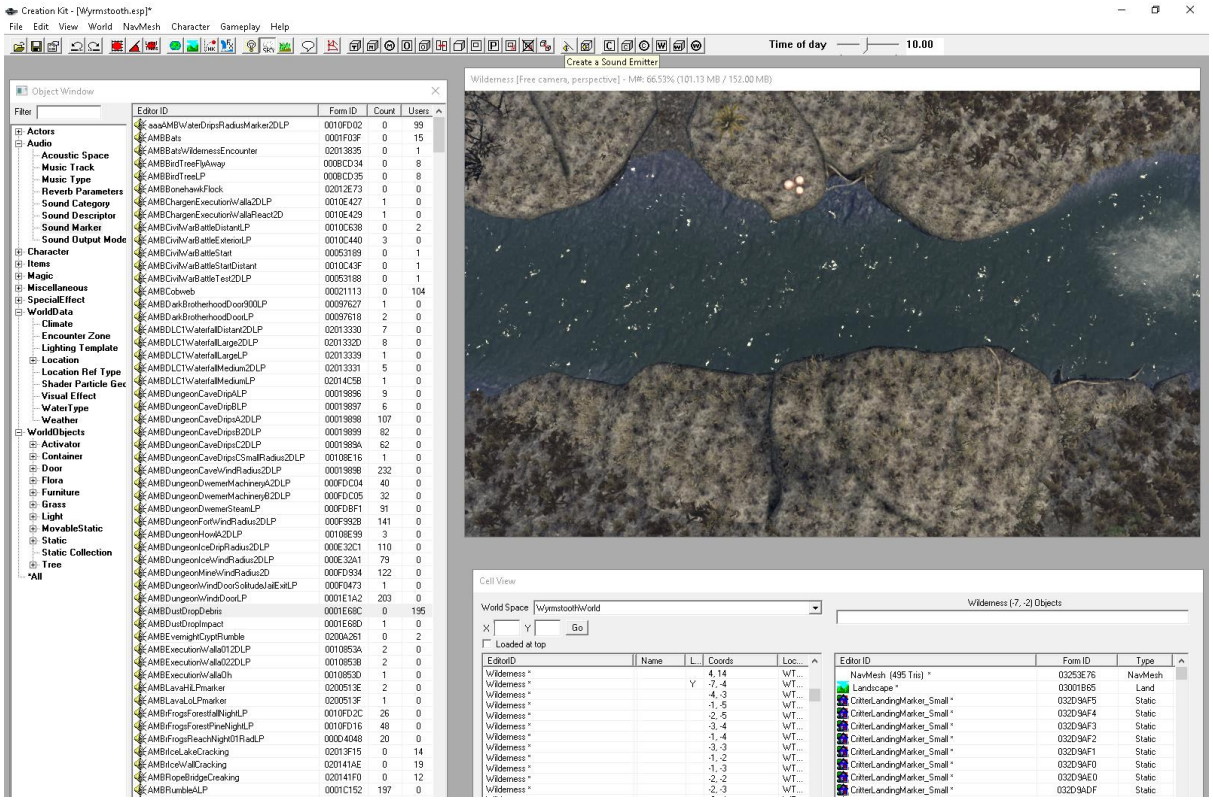


Figure 356 - The sound emitter button.

Next, click on a spot in the render window where you would like to place the sound emitter.

You'll see a Select Form pop-up. Use the Filter field to narrow down the list of sounds.

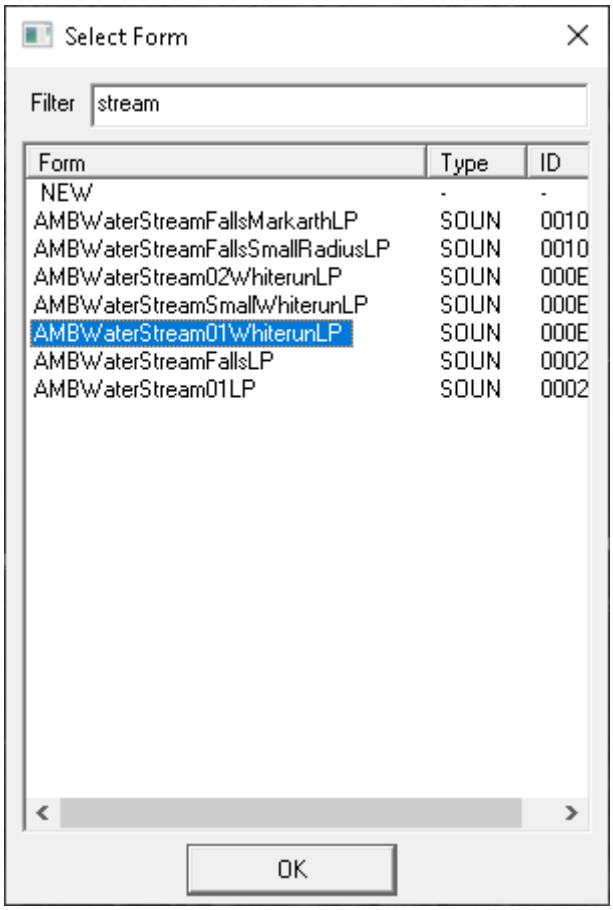


Figure 357 - Selecting a sound to place.

Select the sound you want to add and click OK.

Sound emitters appear as white bars in the render window and can be resized using the scale tool. Usually the scale tool is toggled on when you place a sound emitter, but if it isn't just press '2'.



Figure 358 - Resizing the sound emitter.

In the example above, I stretched it to fit this section of the river.

In this example, sound emitters are used to play a loop of crashing waves along the shore.

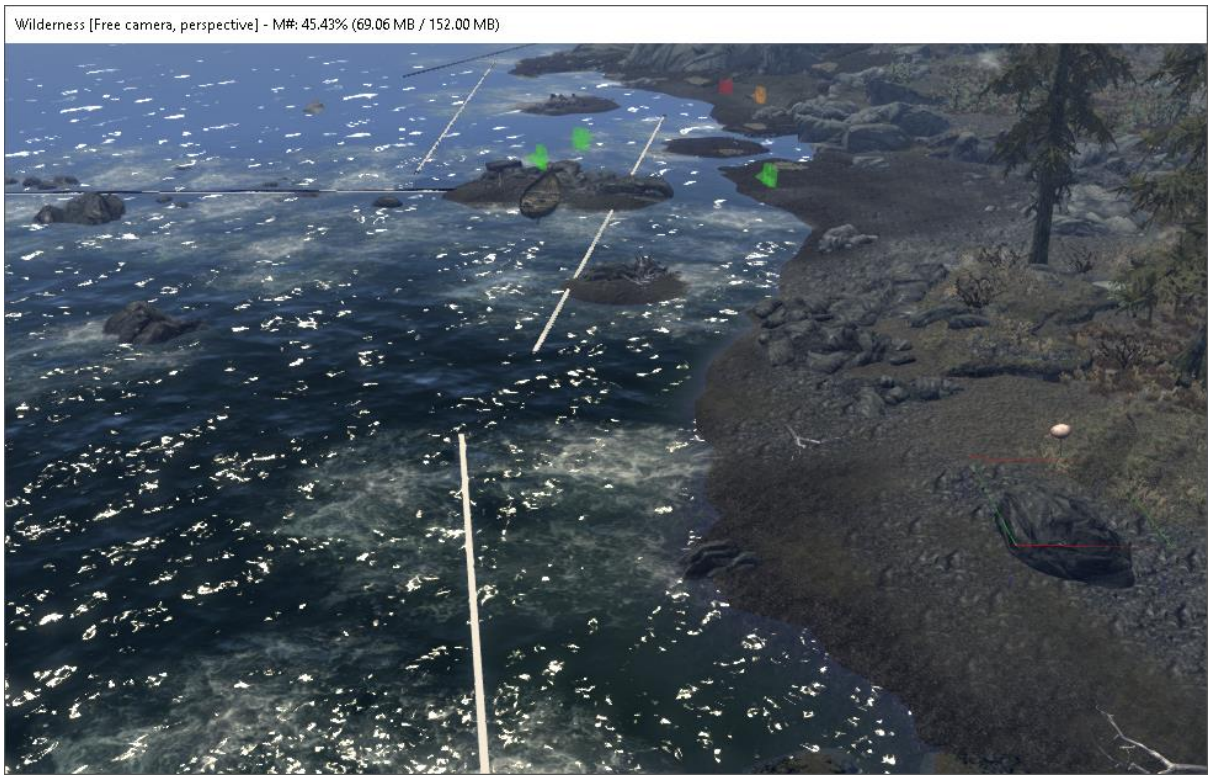


Figure 359 - Sound emitters along a beach.

Lastly, let's take a look at adding new sounds to the game.

There are a few websites that offer royalty free sound effects. The one I typically use is Freesound.org.

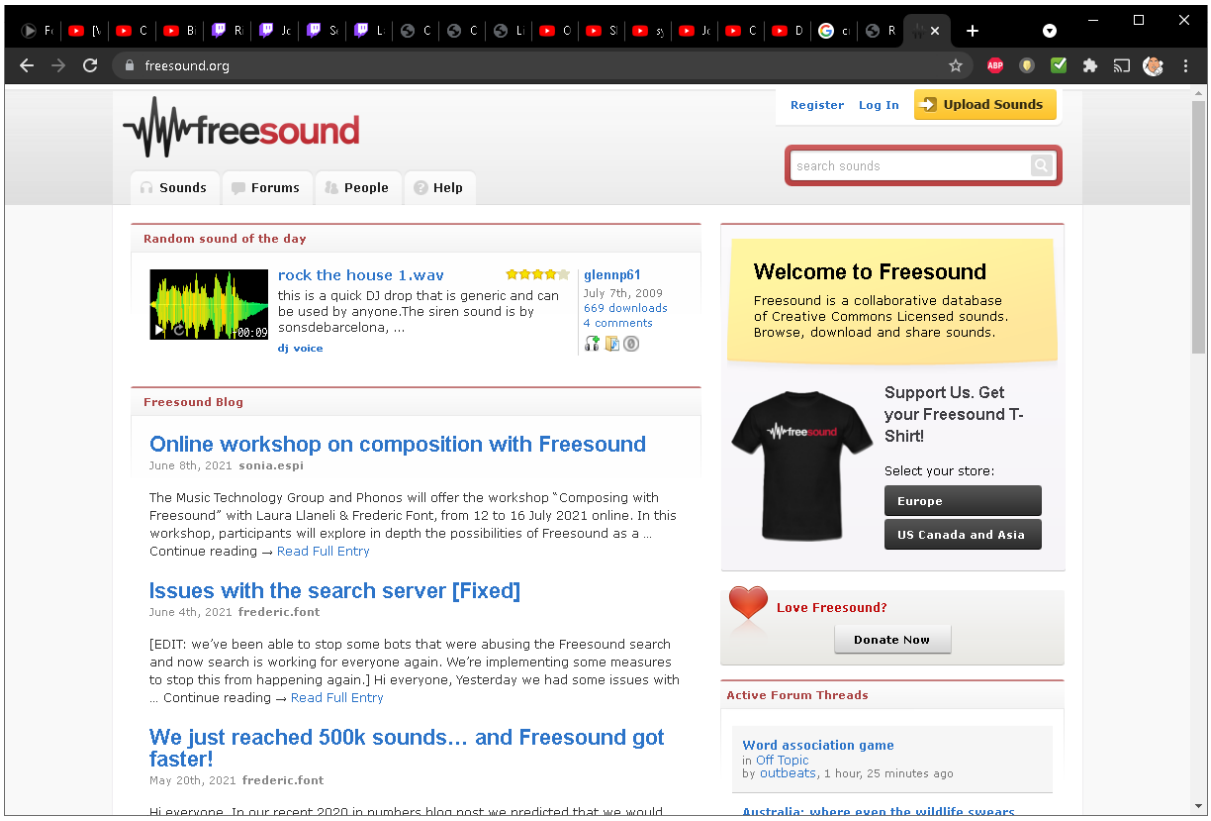


Figure 360 - Freesound.org

Just as an example, let's say we wanted to add in a looping sound of a crow squawking.

Search for 'crow' and look for a good sample.

Ideally, it should have minimal background noise and should be recorded at a quality of at least 44.1 KHz.

This one sounds good.

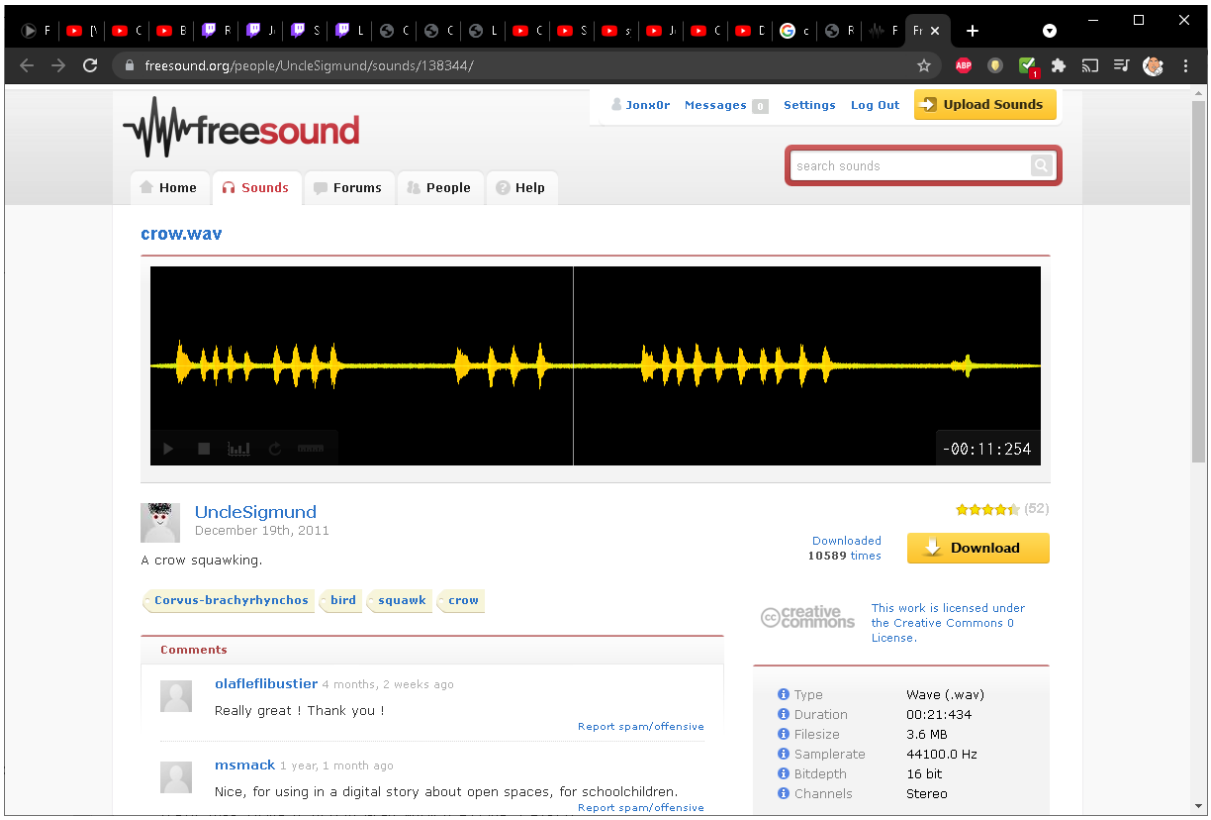


Figure 361 - Found a sample to download.

Click on the Download button to download the sound file.

To clean up the sound file, I'm going to use Audacity which you can download from their website for free.

Open the sound file in Audacity.

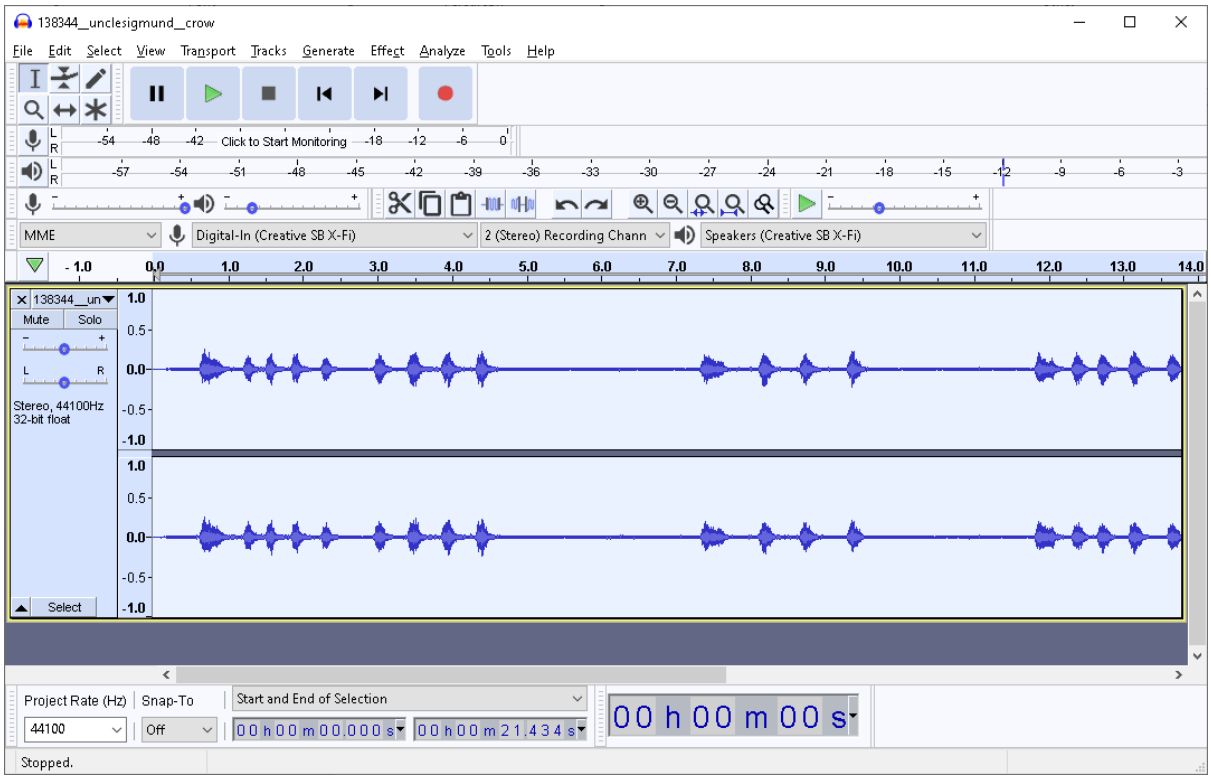


Figure 362 - Importing the sound file into Audacity.

The first thing we need to do is make this a mono track.

Click on the file name in the track and select 'Split Stereo to Mono'.

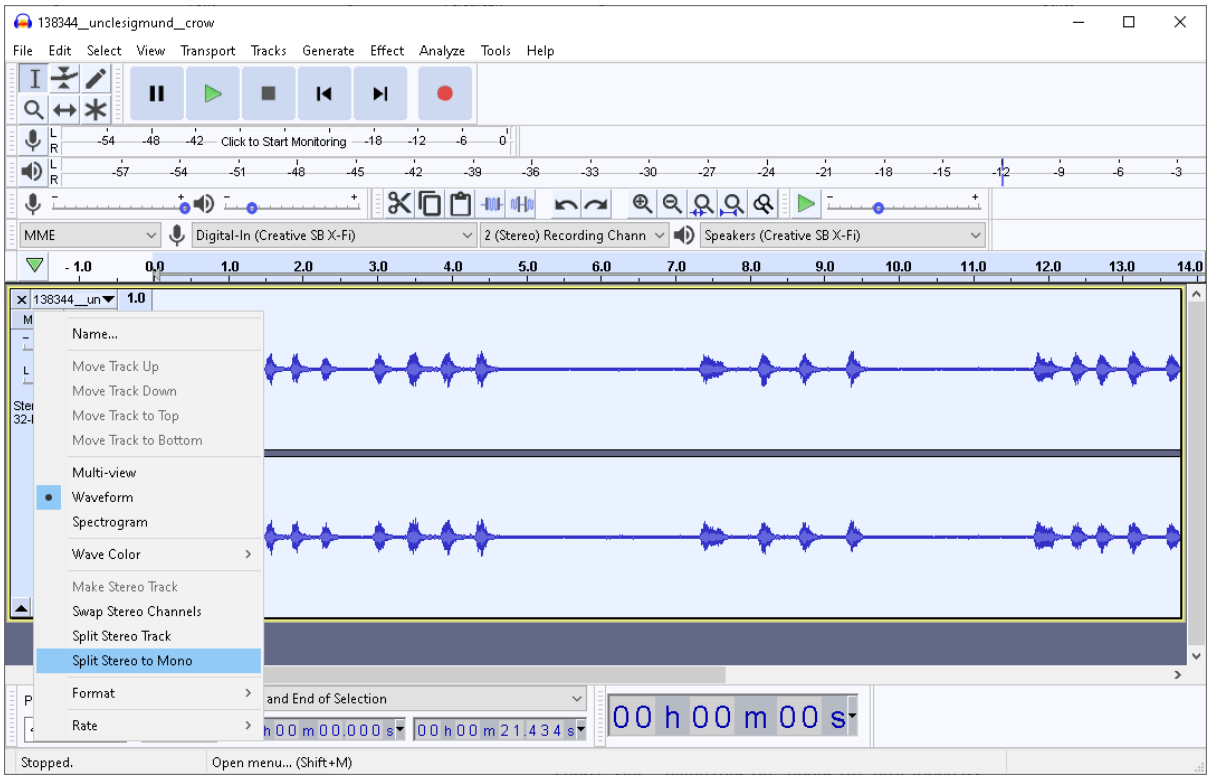


Figure 363 - Split stereo to mono.

We should now have separate tracks for our left and right channels.

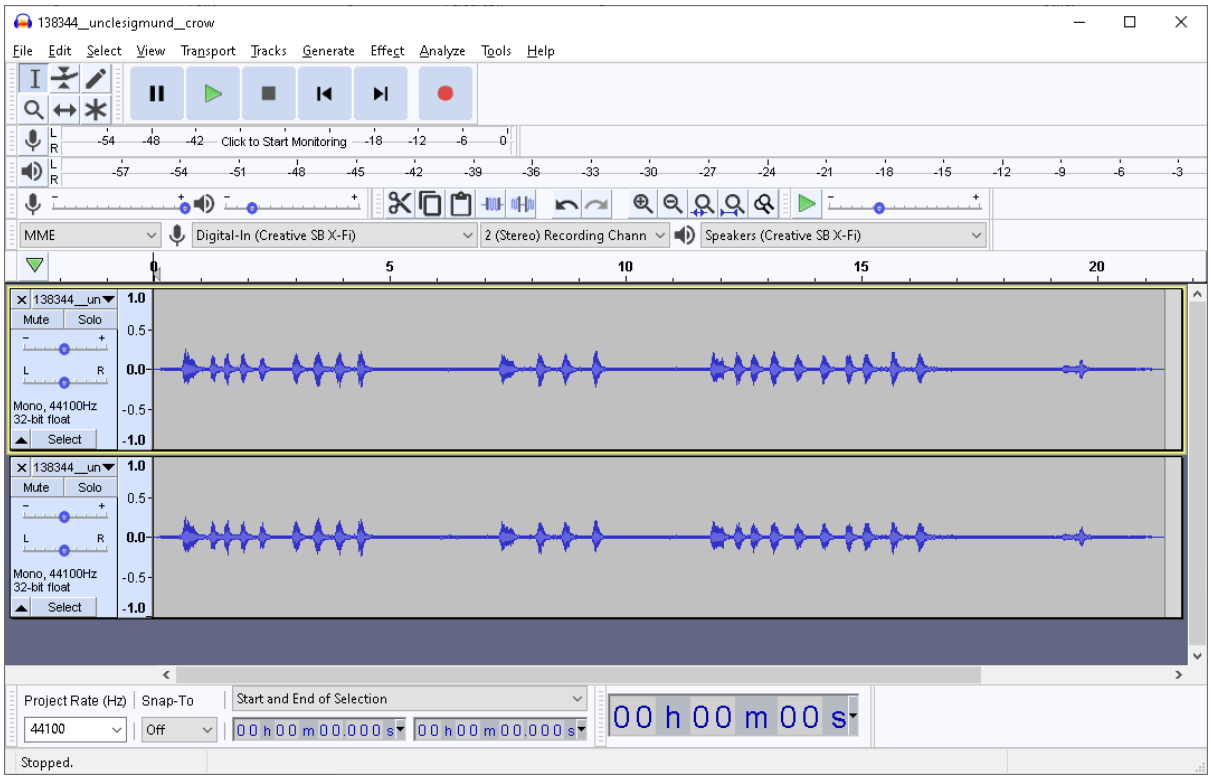


Figure 364 - Separate tracks for left and right channel.

Click 'x' next to one of the channels to remove it.

We should now be left with one channel.

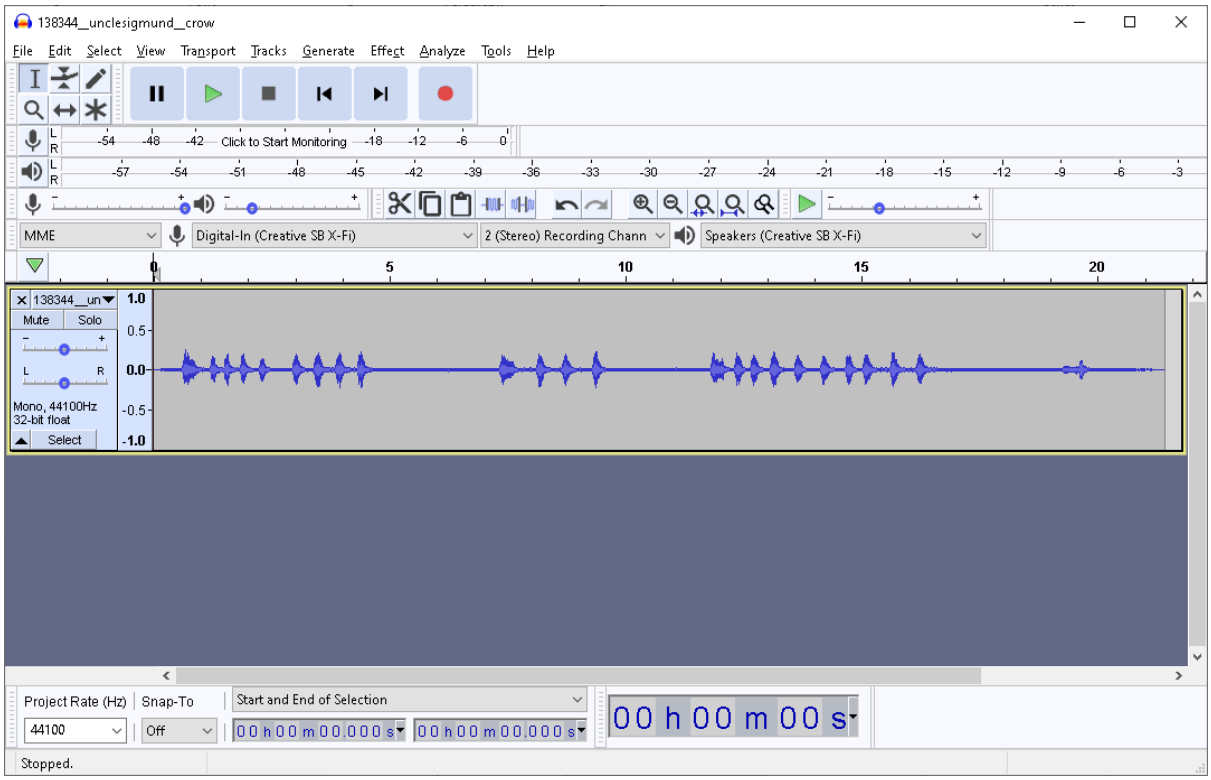


Figure 365 - One channel.

If you play the track, you'll likely notice a fair bit of noise. We'll need to remove that.

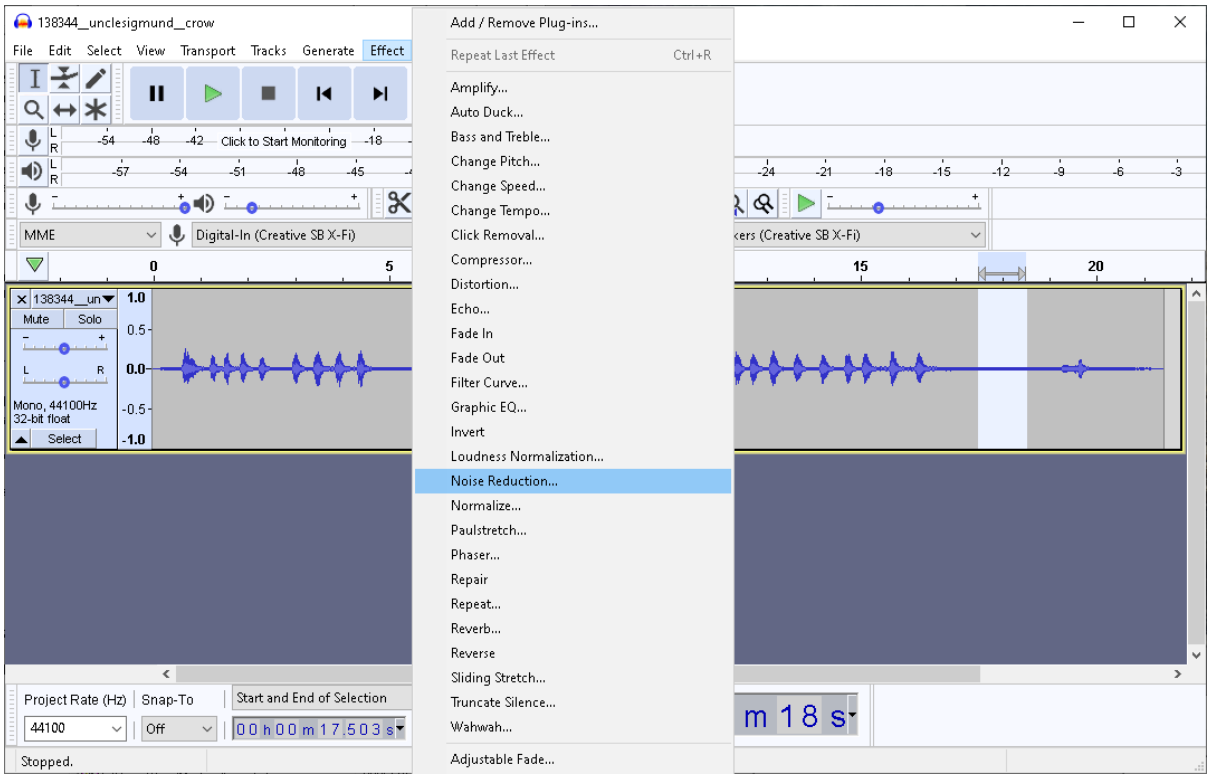


Figure 366 - Running noise reduction.

Select a portion of the background noise and go to Effect > Noise Reduction.

Click on 'Get Noise Profile'.

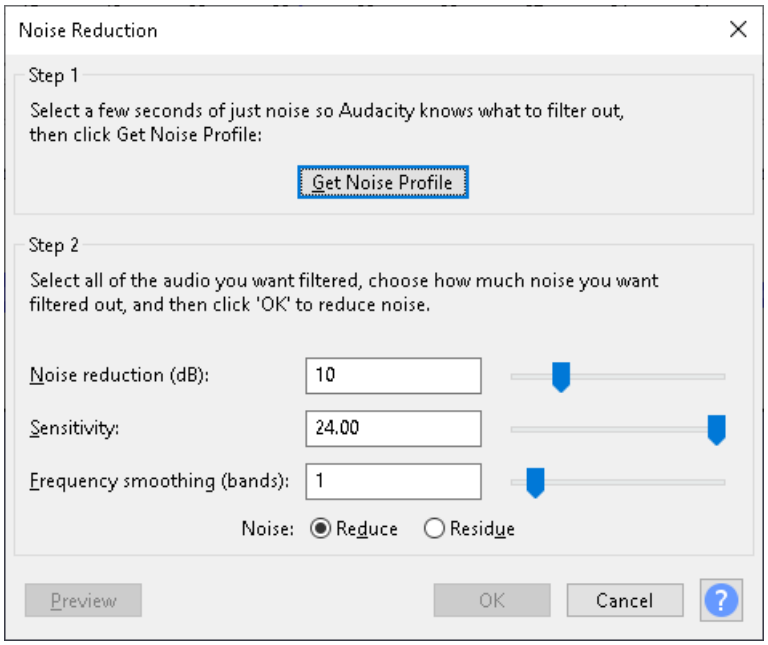


Figure 367 - Setting our noise profile.

Noise Reduction should close automatically.

Select the entire track, then go back to Effect > Noise Reduction.

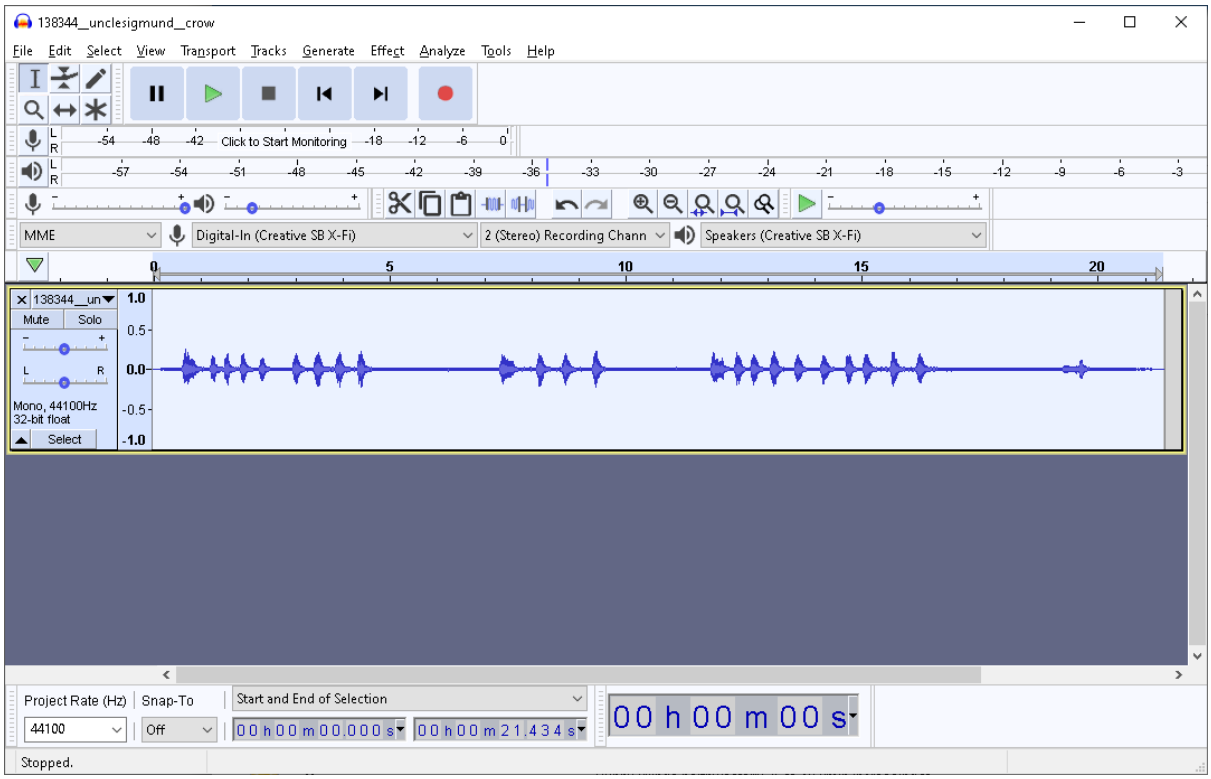


Figure 368 - Entire track selected.

Click Preview to listen to a snippet of audio with the current noise reduction settings.

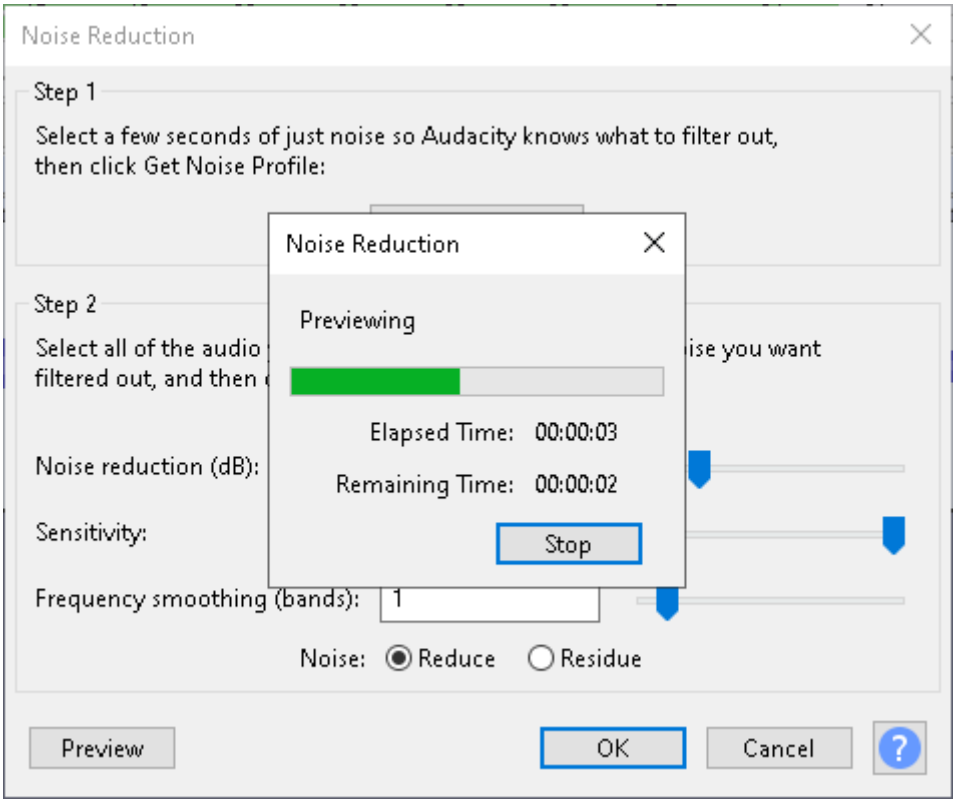


Figure 369 - Previewing noise reduction.

Increase the 'Noise reduction (dB)' slider to reduce the amount of noise. The idea is to reduce noise without distorting the audio too much.

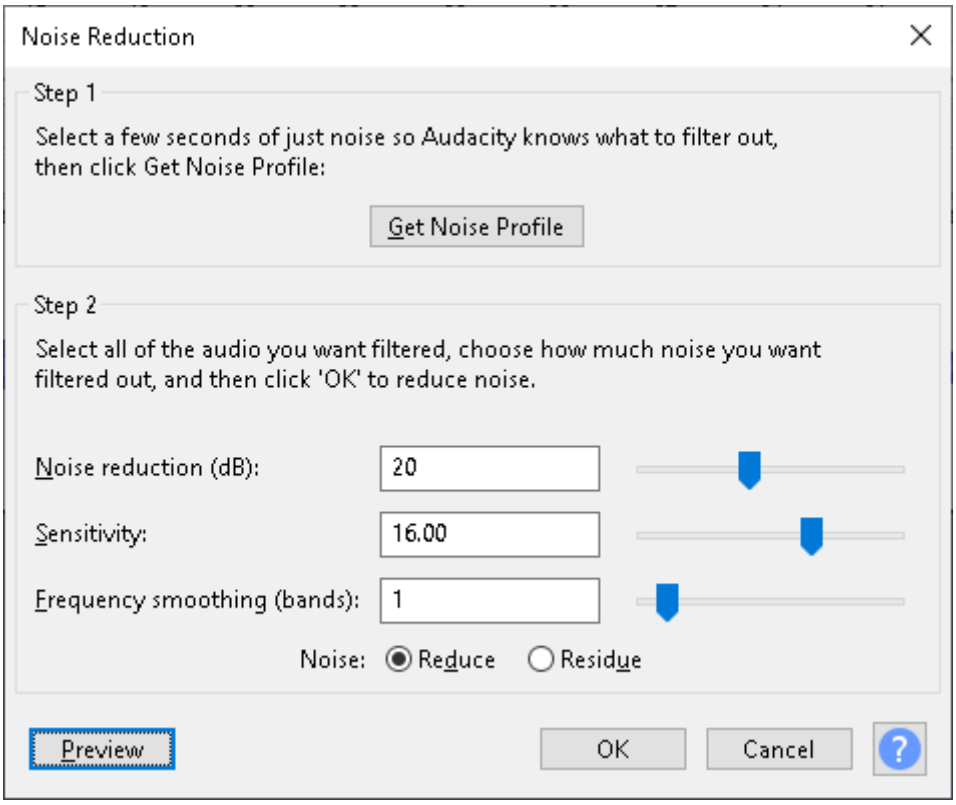


Figure 370 - Changing noise reduction settings.

When you're happy with the results, click OK.

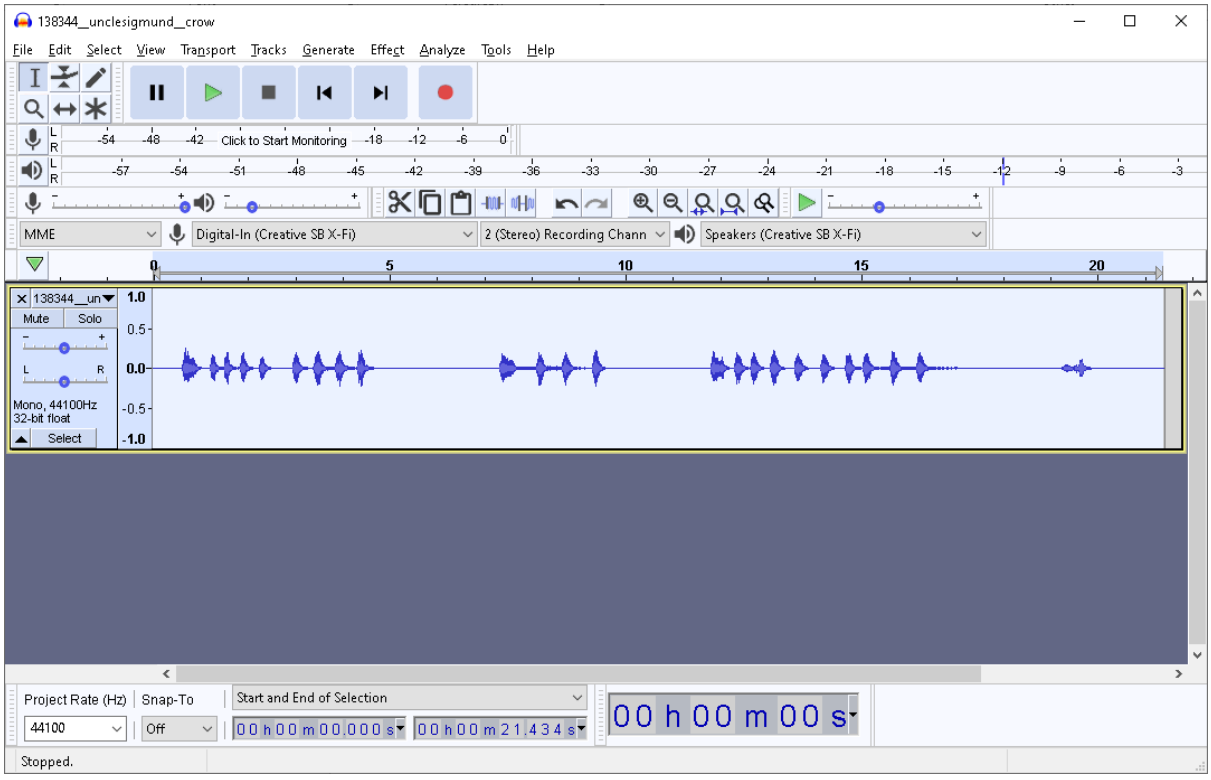


Figure 371 - Noise reduced track.

If your audio file is particularly long, you'll need to trim it. I'd recommend you try keeping looping sounds shorter than 30 seconds in length.

To export our audio, go to File > Export and select 'Export as WAV'.

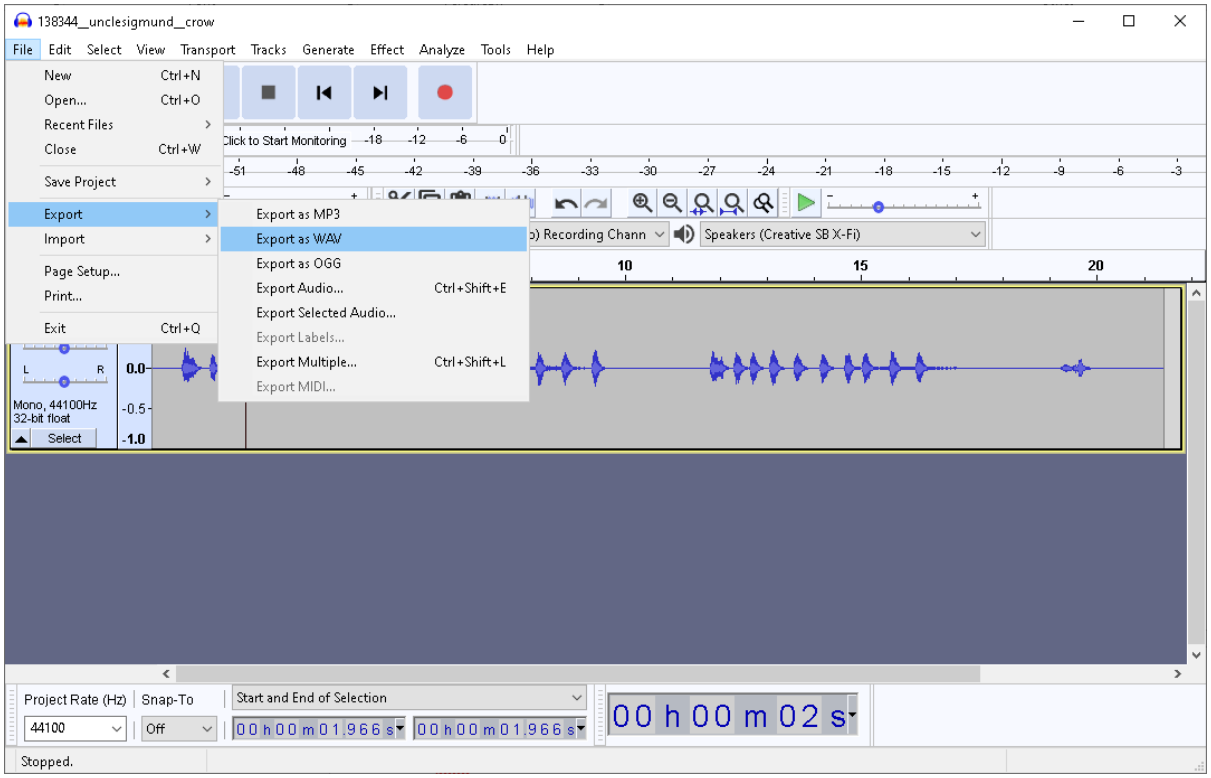


Figure 372 - Exporting audio as .wav.

I'm going to save the file to my Skyrim\Data\Sound\fx\amb folder and set the file name to 'wt_amb_crow_squawk.wav'.

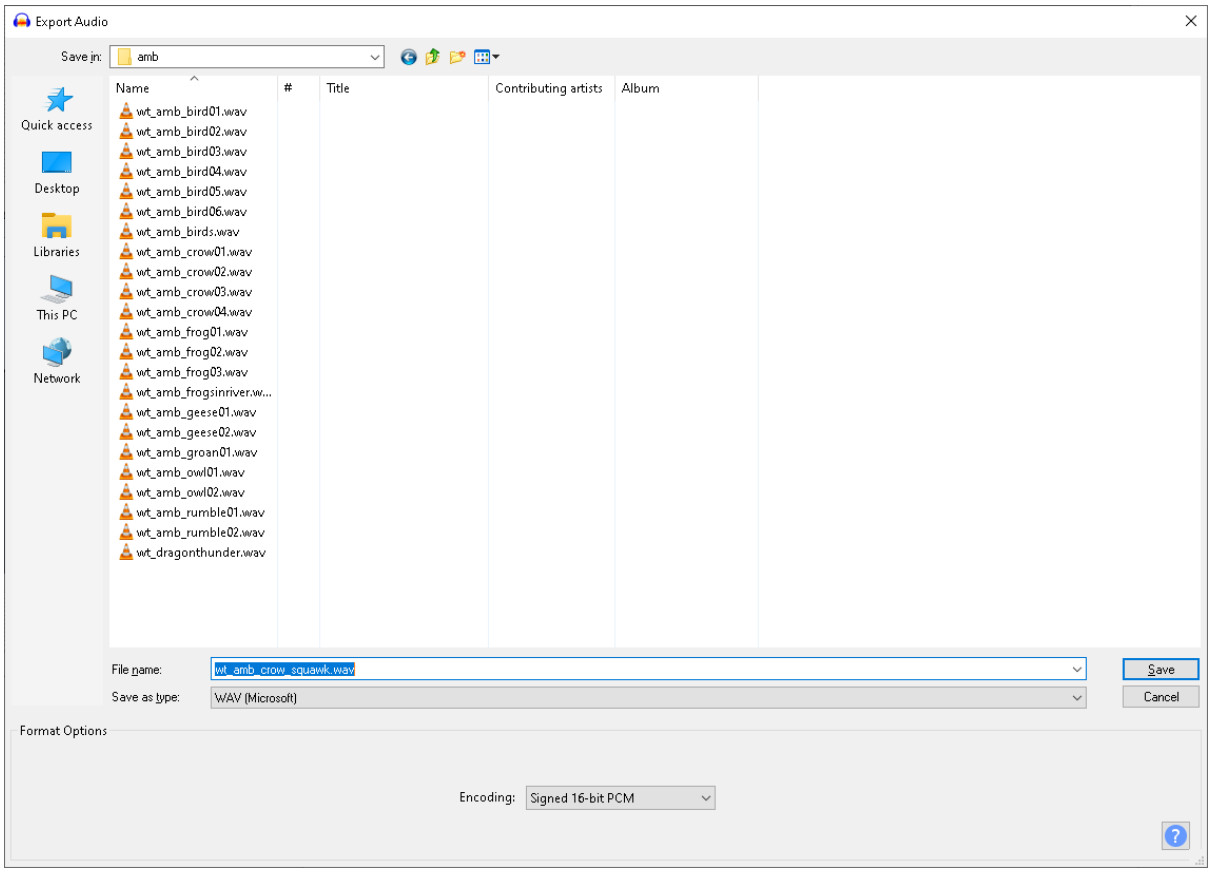


Figure 373 - Setting the file name and location.

Click Save.

Back in the Creation Kit, go to Audio > Sound Descriptor. Right-click on one of the existing sound descriptors and select New.

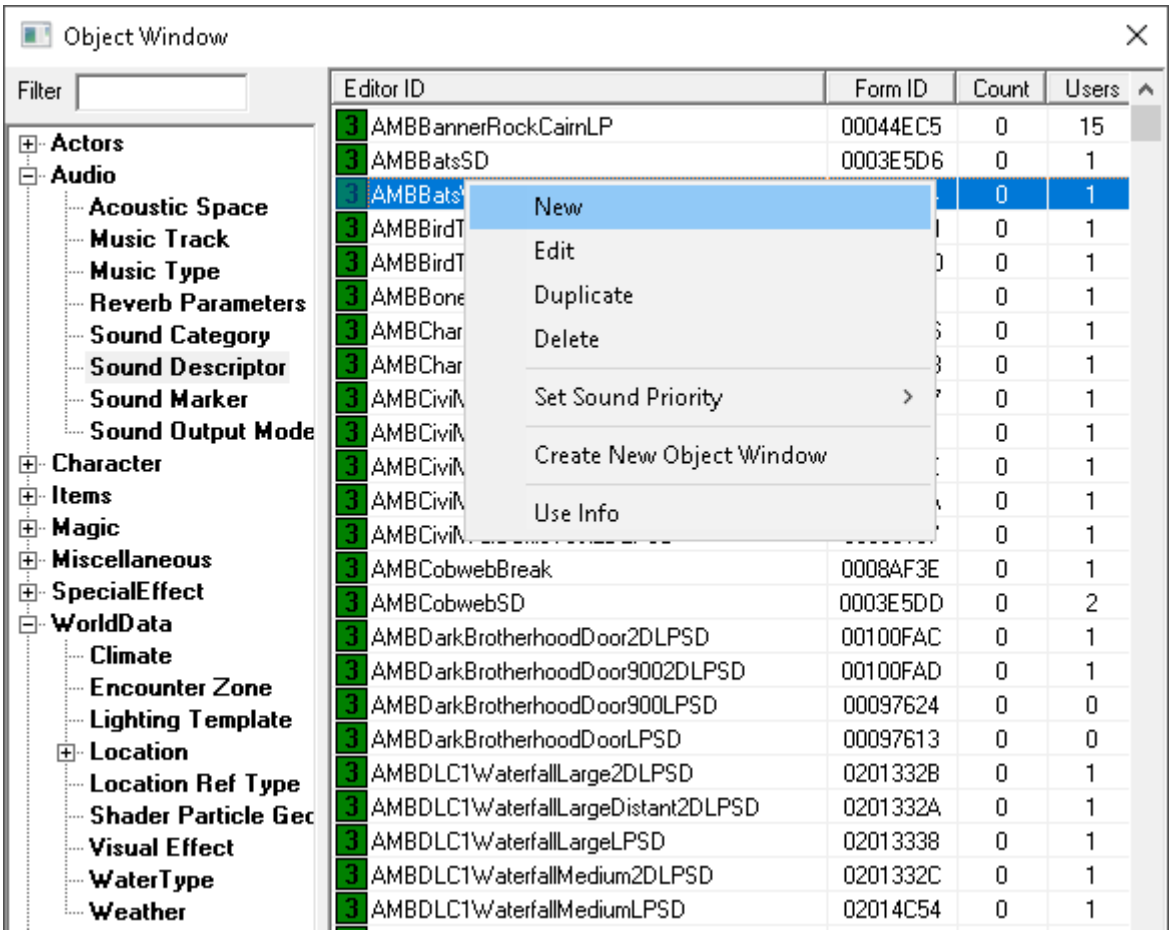


Figure 374 - Creating a new sound descriptor.

This is going to be an ambient sound, so set the Category to AudioCategoryAMB. Set the Output Model to SOMMono02000 and set Looping to Loop.

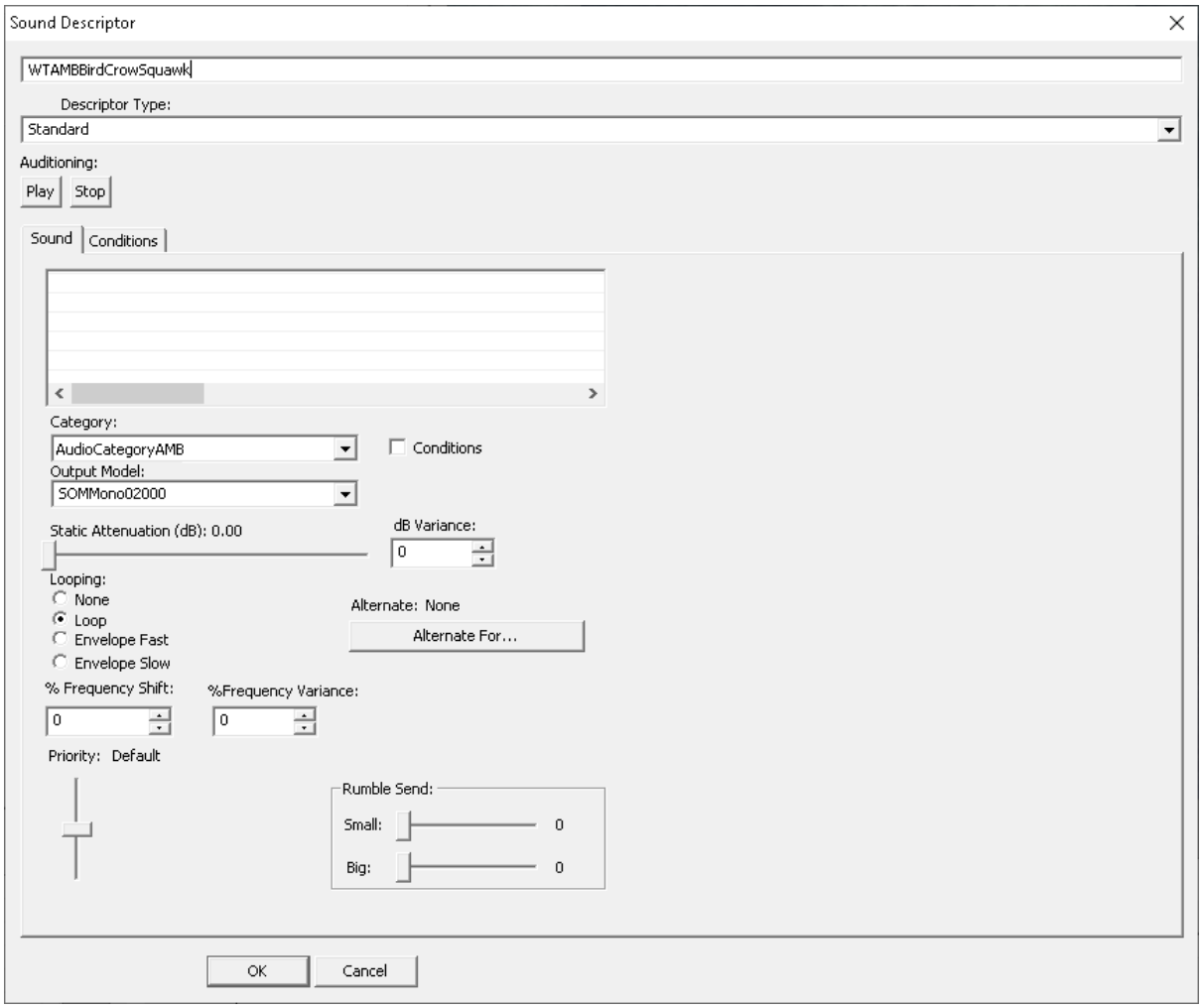


Figure 375 - Setting up our new sound descriptor.

Right-click in the sound list and select New.

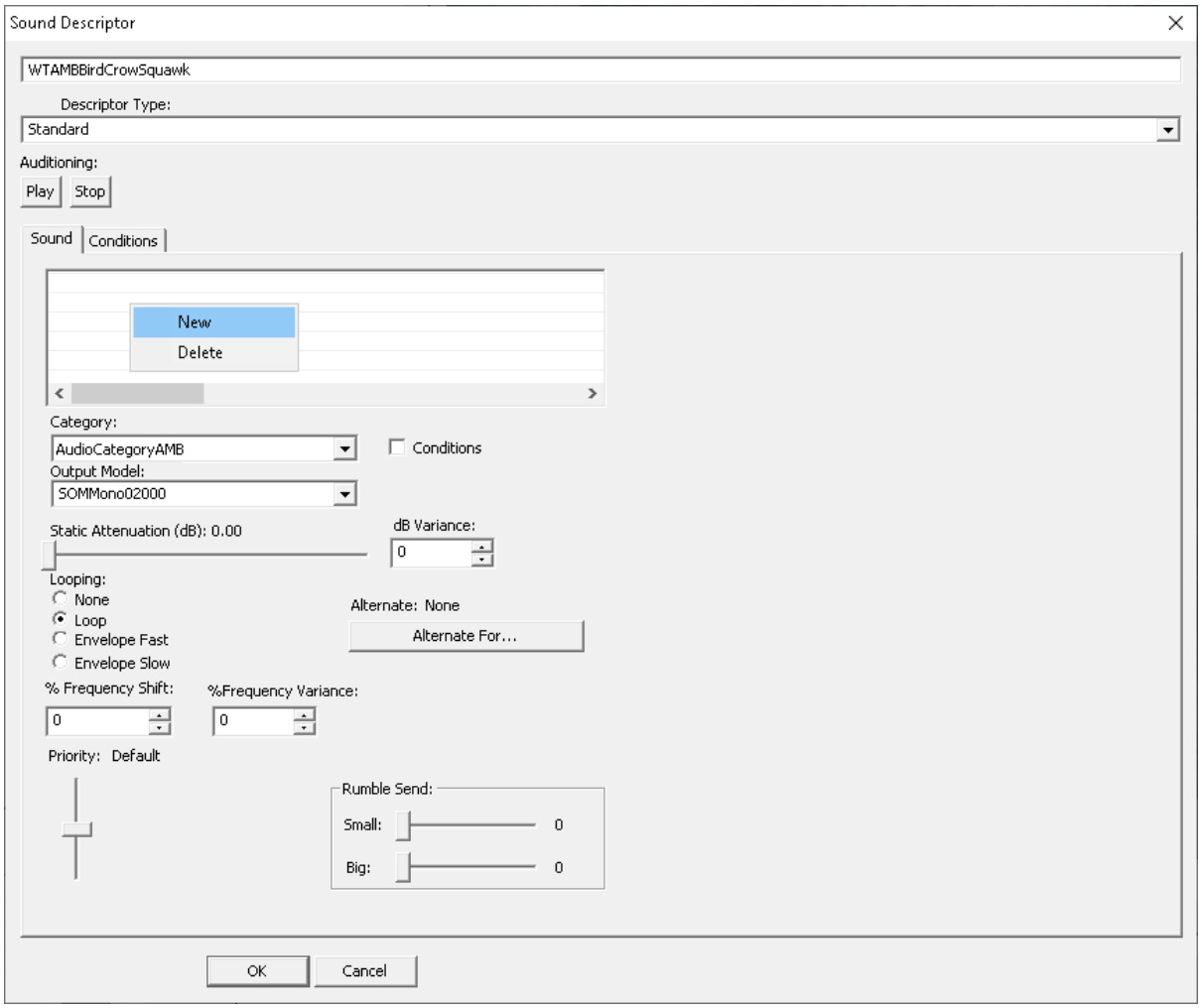


Figure 376 - Adding a .wav file to the sound descriptor.

Select the .wav file. Again, for this example that will be wt_amb_crow_squawk.wav under the Skyrim\Data\Sound\fx\amb folder.

The .wav file should now appear in the sound list.

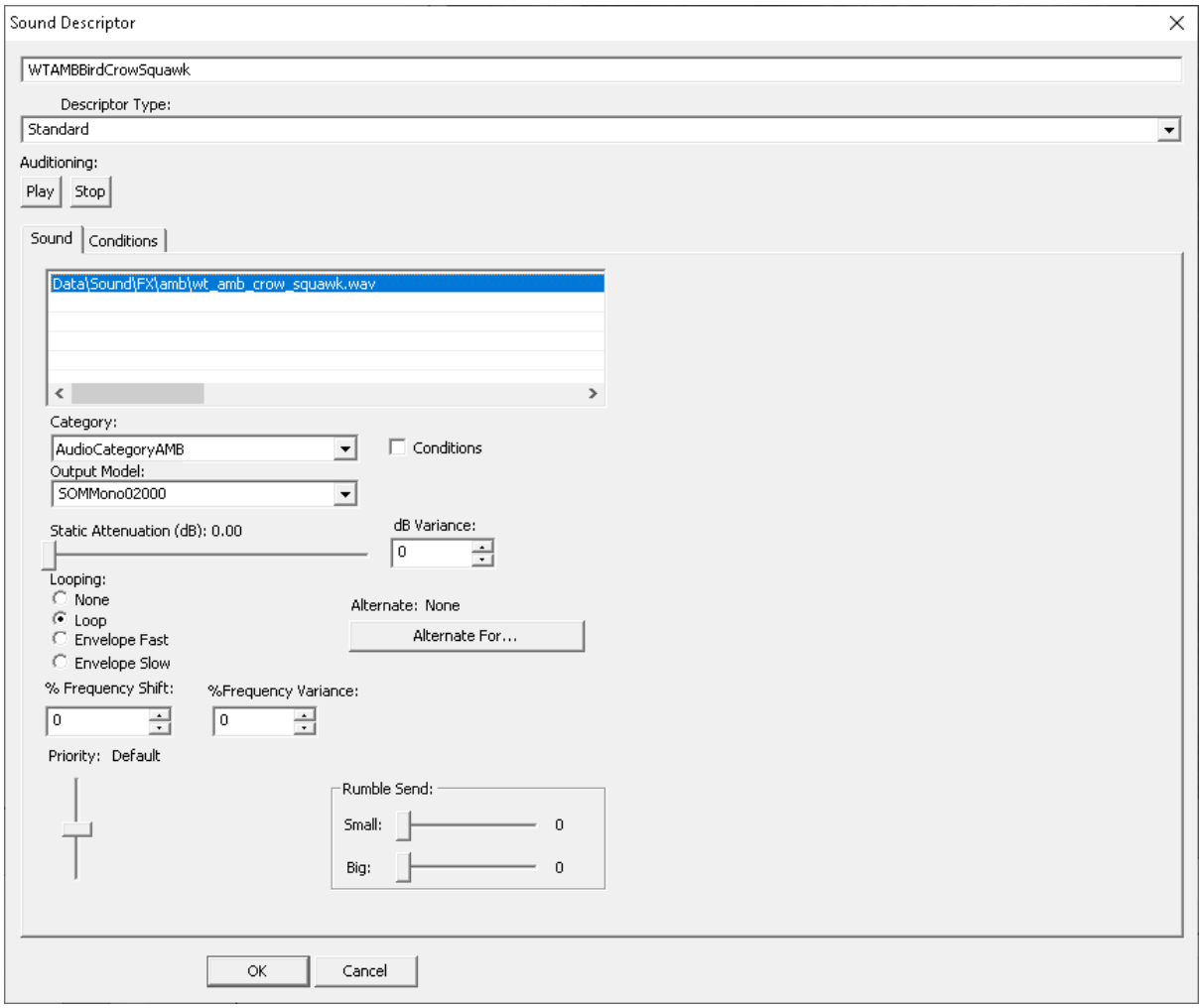


Figure 377 - Sound added to sound descriptor.

Conditions can also be added to sound descriptors. An example of a commonly used condition is to check the time of day, so certain sounds only occur during specific hours.

To add a condition, you need to tick the Conditions checkbox first.

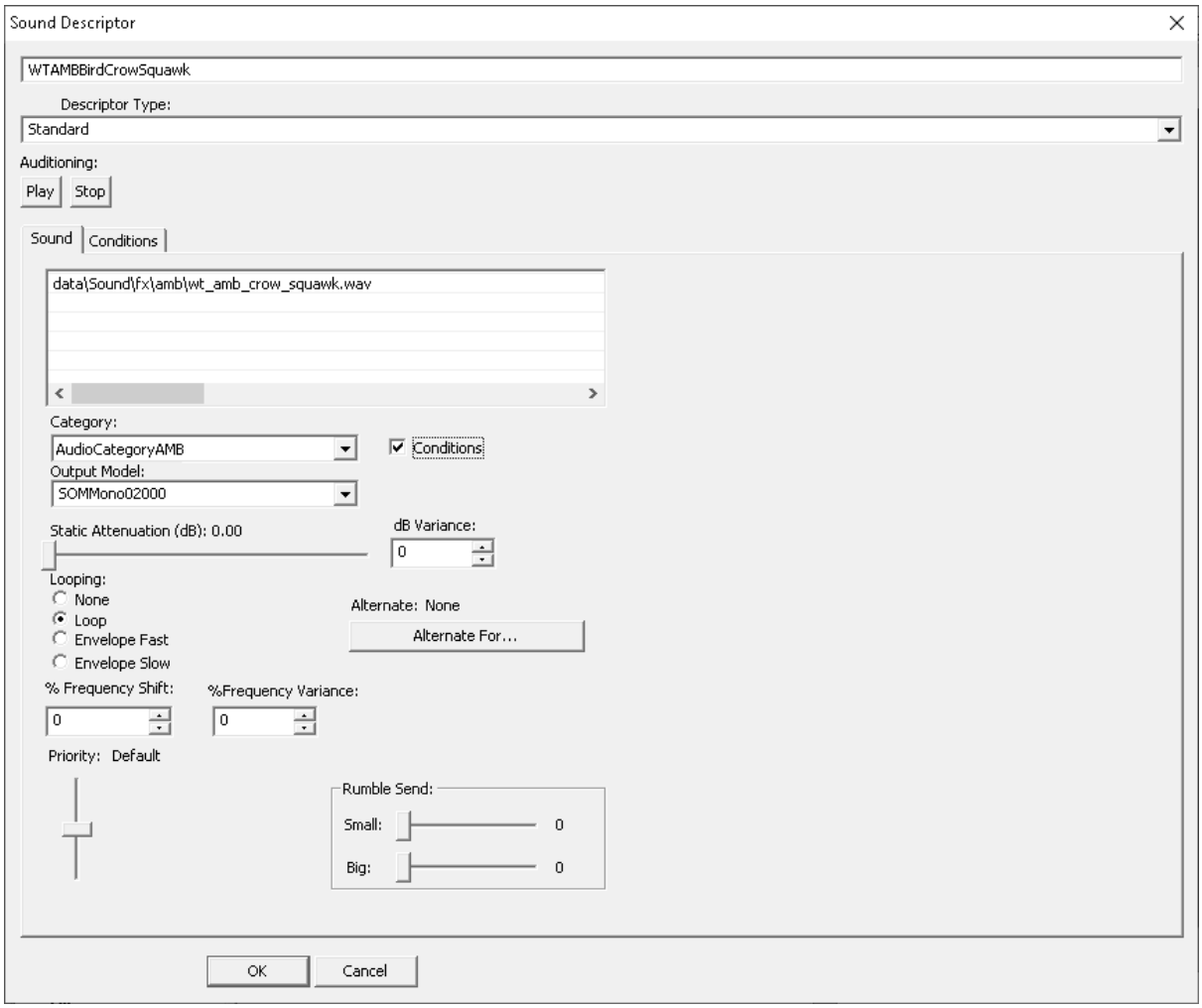


Figure 378 - Conditions enabled.

Conditions are added under the Conditions tab.

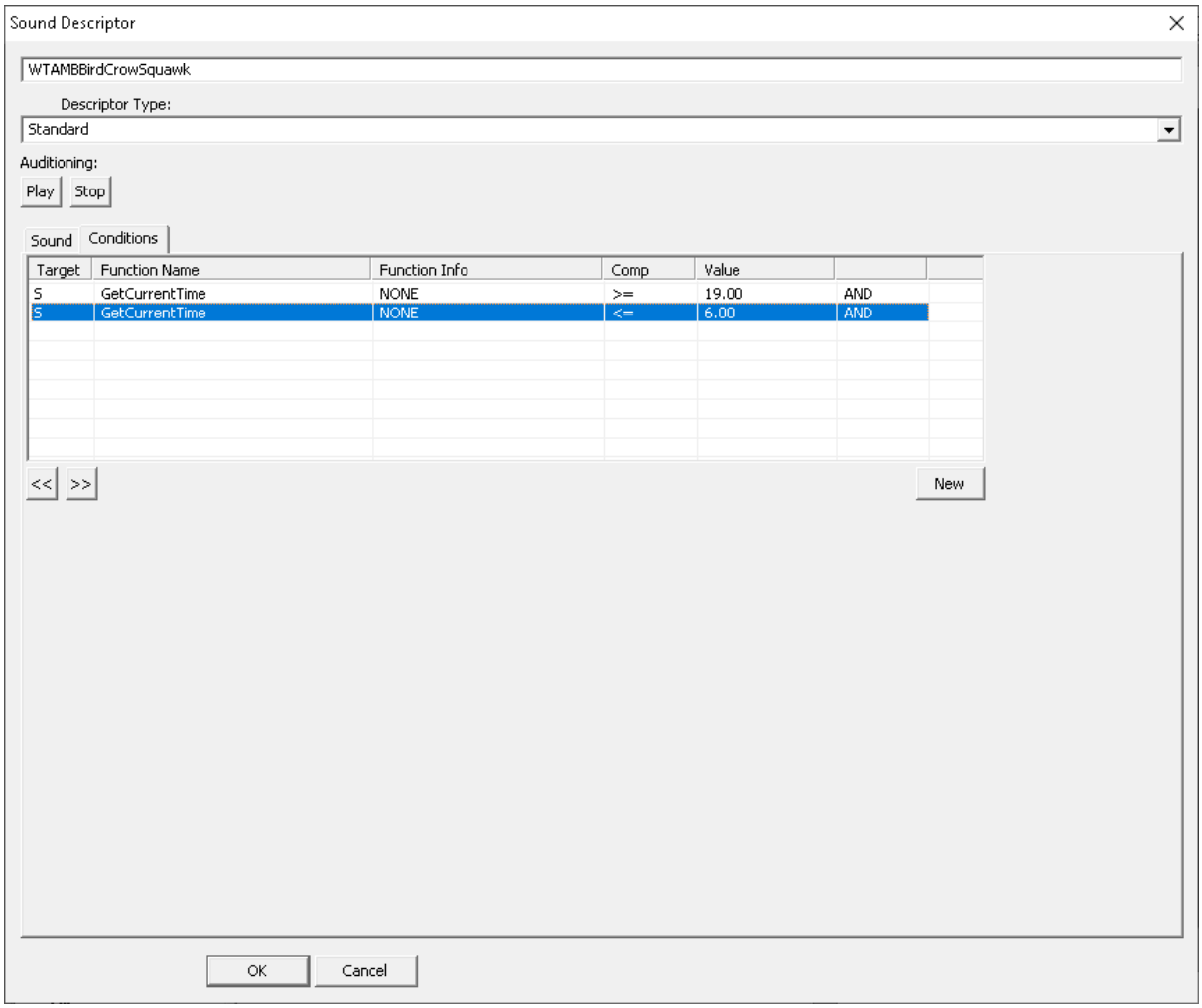


Figure 379 - Conditions added to the sound descriptor.

In the screenshot above, I added two GetCurrentTime condition functions to ensure that the crow squawking sound only plays during daytime hours.

Click OK to close out of the Sound Descriptor properties.

Important: Adding, deleting or modifying sound descriptors may cause the Creation Kit to eventually crash, so make sure you save before continuing.

To create the sound marker, in the Object Window go to Audio > Sound Marker.

Right-click on one of the existing sound markers and select New.

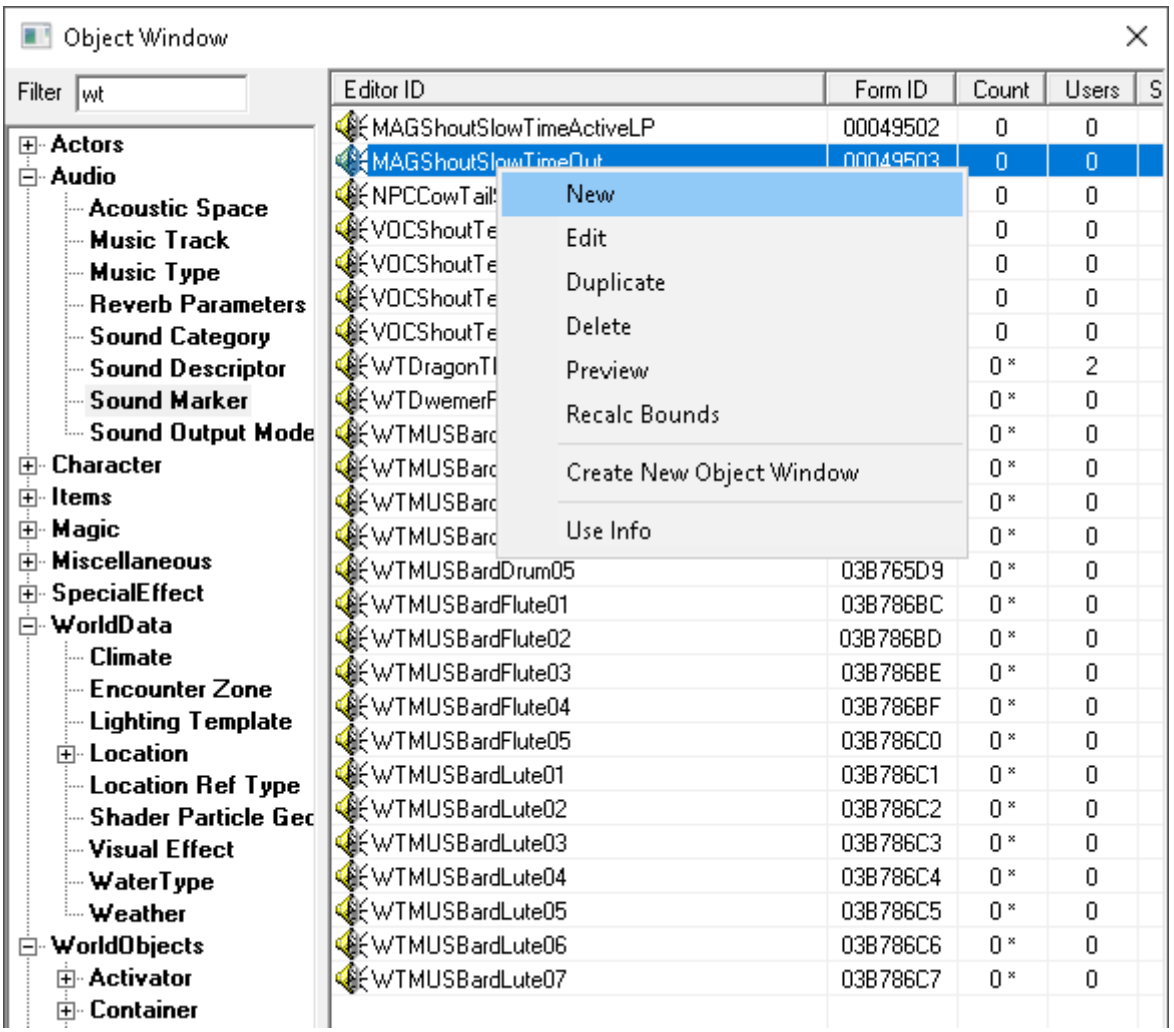


Figure 380 - Creating a new sound marker.

Select your sound descriptor and enter in an ID. I set it to match the name of the sound descriptor, but appended 'LP' to the end to indicate that it's a loop.

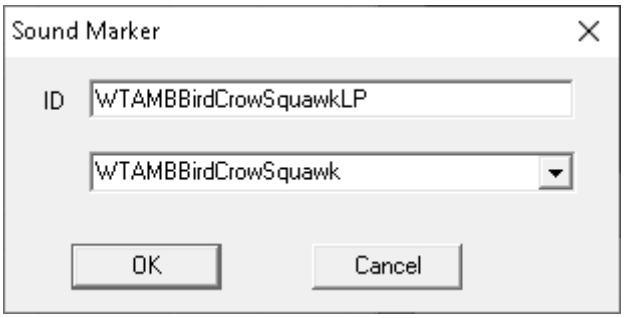


Figure 381 - Configuring the sound marker.

Click OK.

Now you can drag and drop the new sound marker into the render window to add it to your world space.



Figure 382 - The new sound marker added to a custom world space.

Alternatively, if you're setting up this sound to use as a regional ambient sound, go back to the sound descriptor and set Looping to None.

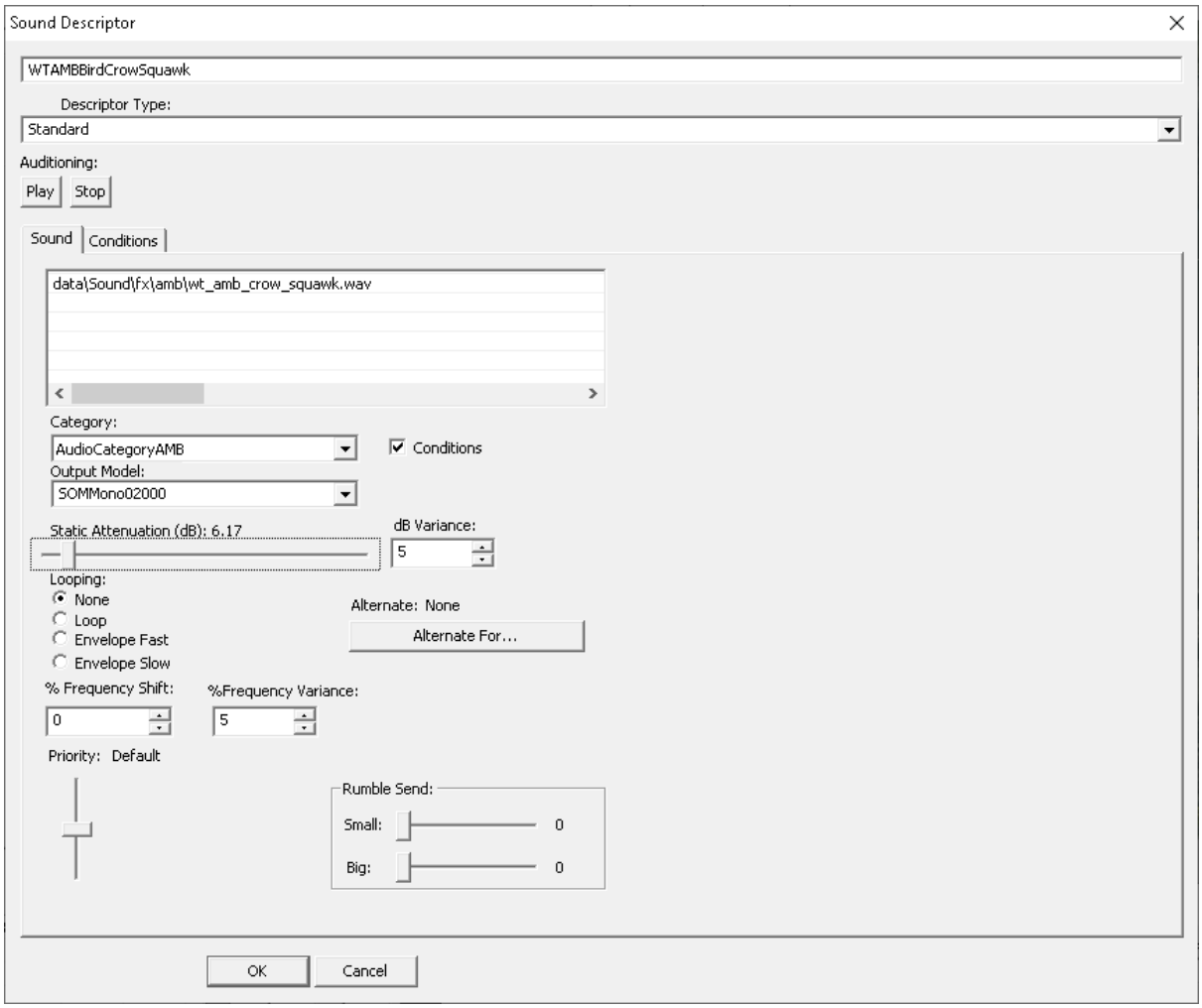


Figure 383 - Changing the sound descriptor for use as a regional ambient sound.

I also set the dB Variance to 5 and the %Frequency Variance to 5 to add some slight variation in the loudness of the sound every time it plays. You could also set a % Frequency Shift if you want to. This will affect the pitch of the sound every time it plays, but in my example I'm not going to do that.

Click OK to save these changes.

You can then drag and drop the sound descriptor into the list of sounds on the Sound tab of your Region.

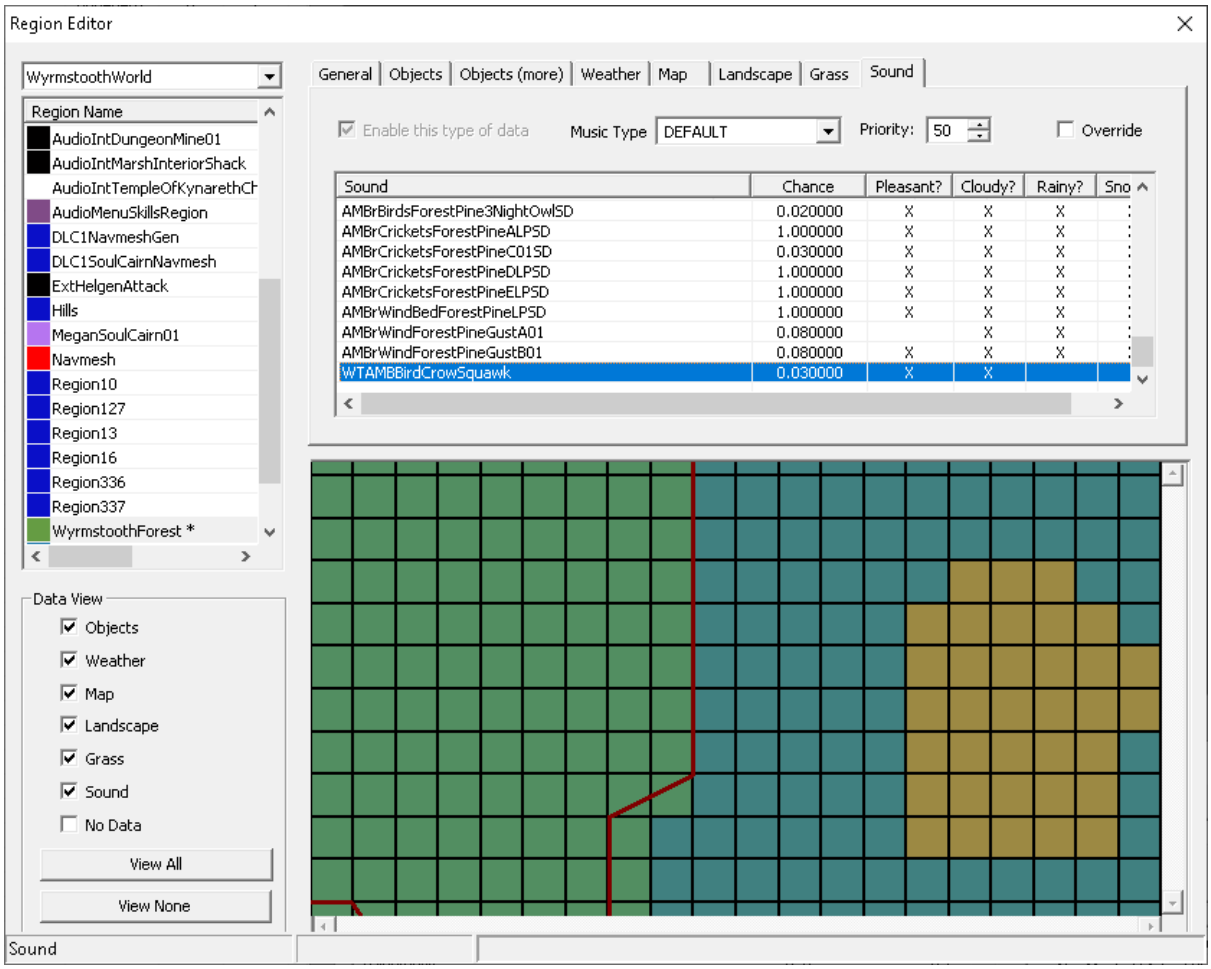


Figure 384 - Adding our new sound descriptor to a region.

See the section on Regions for detailed steps on setting them up.

ADDING ACOUSTIC SPACES

Acoustic spaces modify acoustic properties such as reverb when a player walks within one. The size of an acoustic space can be adjusted with the scale gizmo which can be toggled on or off by pressing '2'.

To add an acoustic space, click on Acoustic Space in the toolbar.

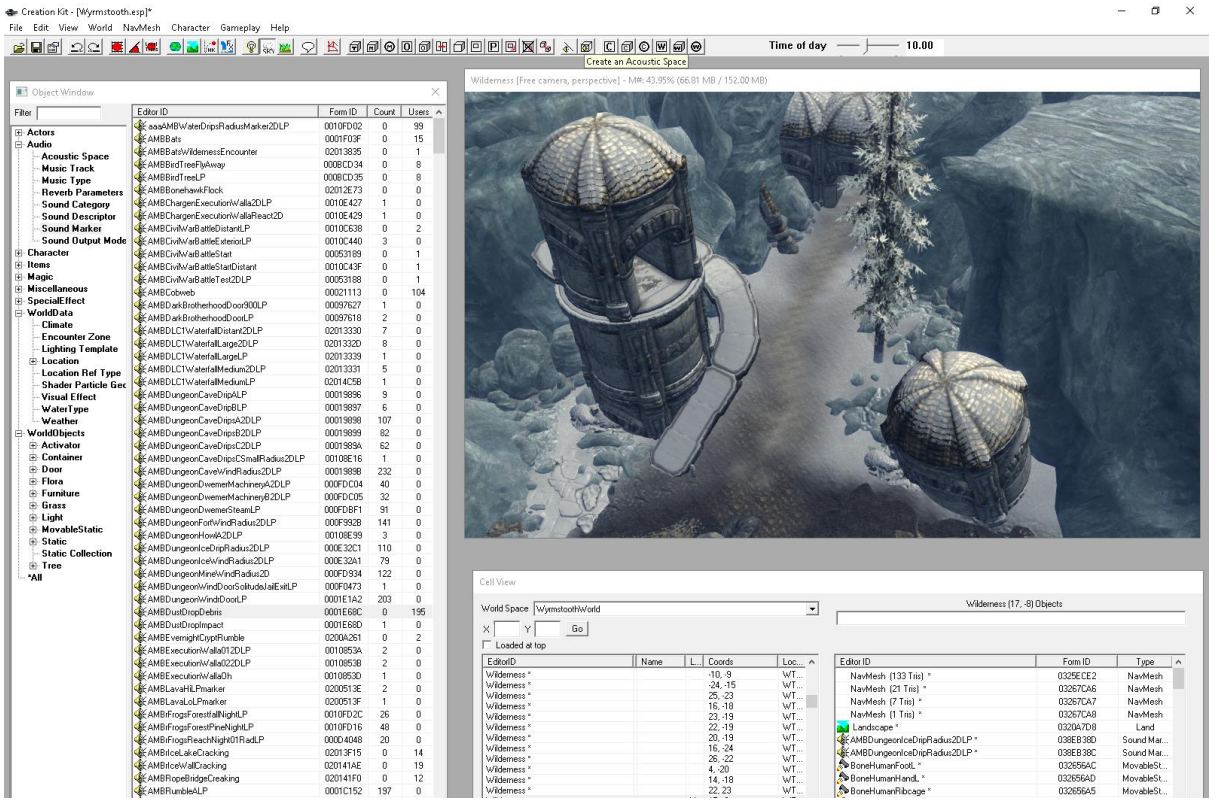


Figure 385 - Adding an acoustic space.

You'll see a Select Form pop-up. Use the Filter field to narrow down the list of acoustic spaces.

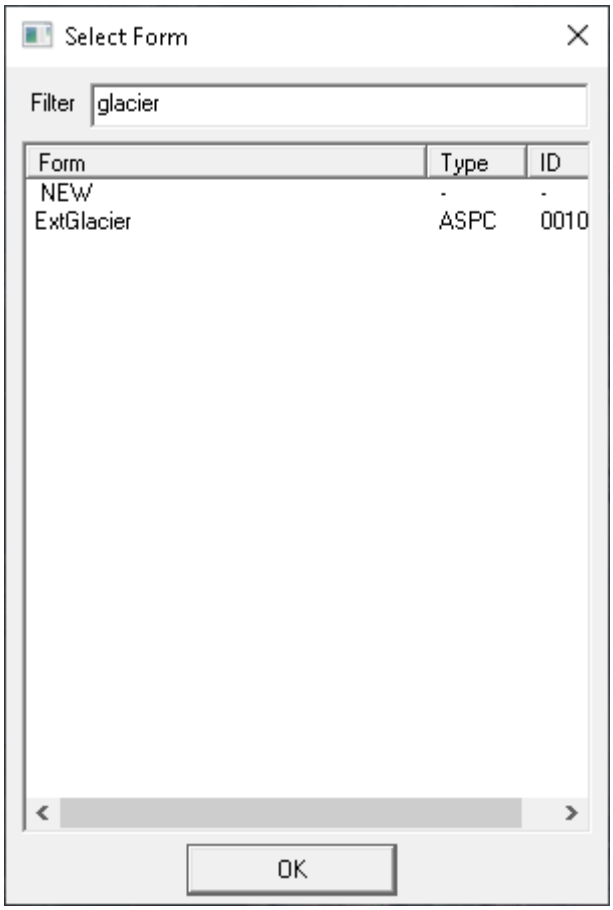


Figure 386 - Selecting an acoustic space to place.

Select the acoustic space you want to add, such as ExtGlacier, and click OK.

I like to use ExtGlacier in between glaciers, where you would expect sound to bounce around.

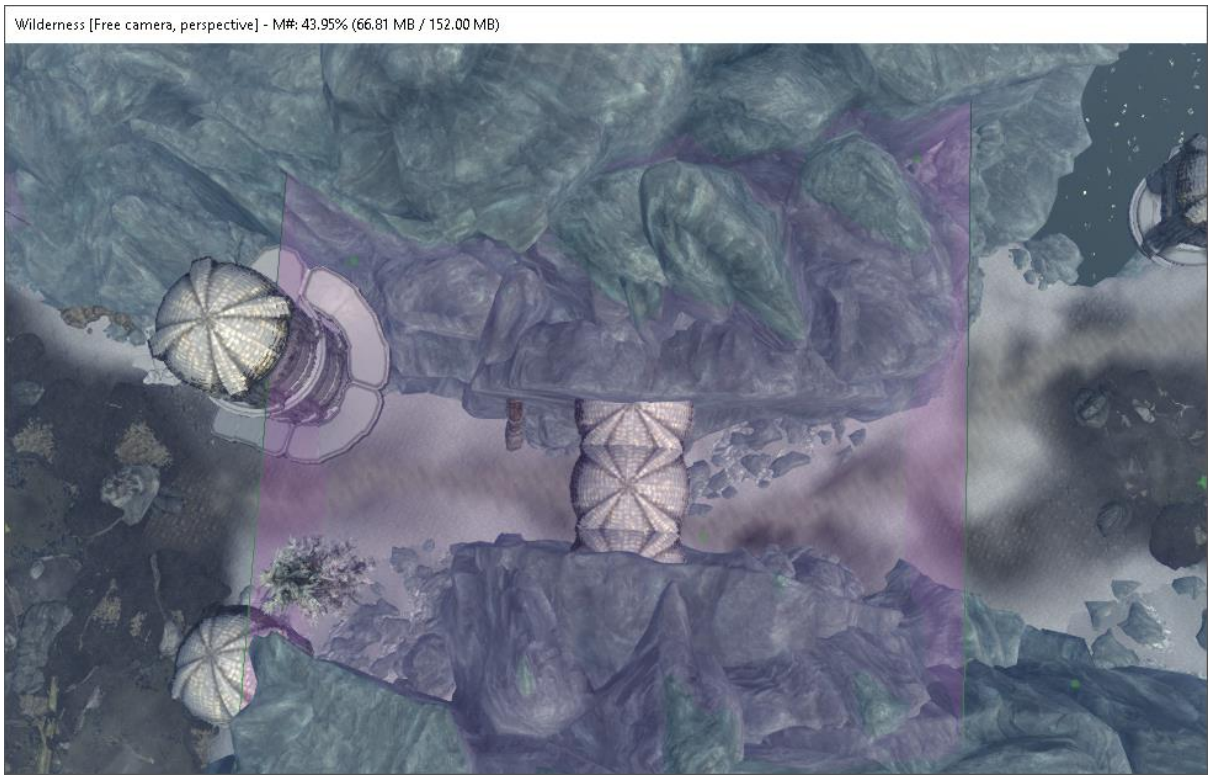


Figure 387 - An acoustic space in between glaciers.

A list of acoustic spaces can be found in the Object Window under Audio > Acoustic Space.

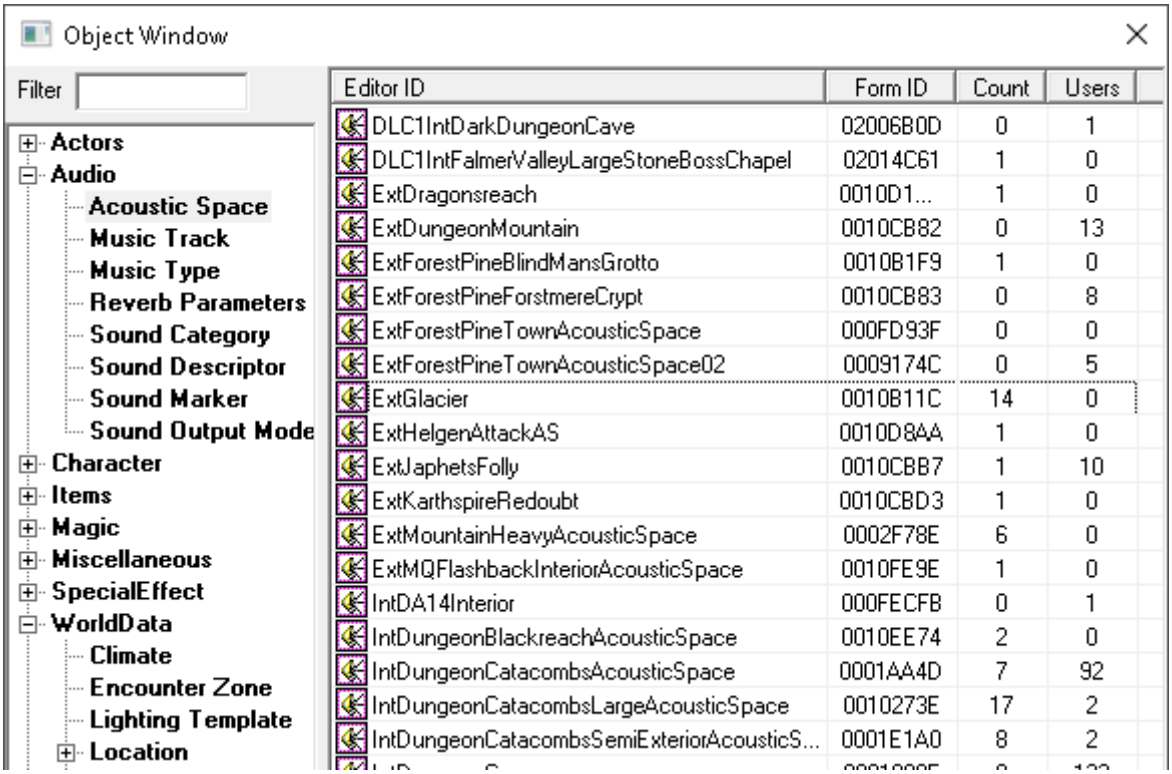


Figure 388 - List of acoustic spaces.

To add a new acoustic space, right-click on one of the existing ones and select New.

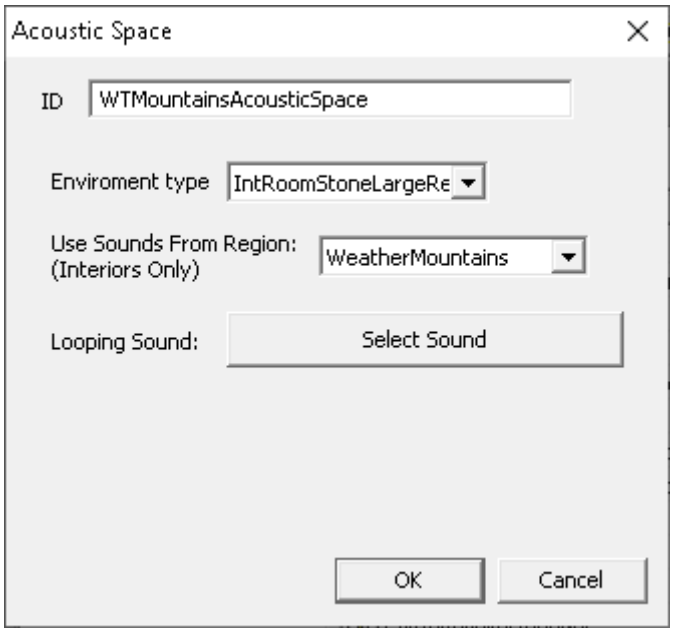


Figure 389 - New acoustic space.

Set the ID. In my example, I just set it to MountainsAcousticSpace and added the 'WT' prefix.

Say for example we wanted to add an acoustic space to a narrow mountain pass, where we'd hear sound bouncing off the cliff sides.

To accomplish this, I set the 'Environment type' field to IntRoomStoneLargeReverb and the 'Use Sounds from Region' field to WeatherMountains.

Optionally, you can add a specific looping sound by clicking on the Select Sound button, but in my example I'm not going to do that.

Click OK to create the new acoustic space.

But what if you wanted to create a specific environment type? Maybe something that would work better in between mountain peaks?

In the Object Window navigate to Audio > Reverb Parameters.

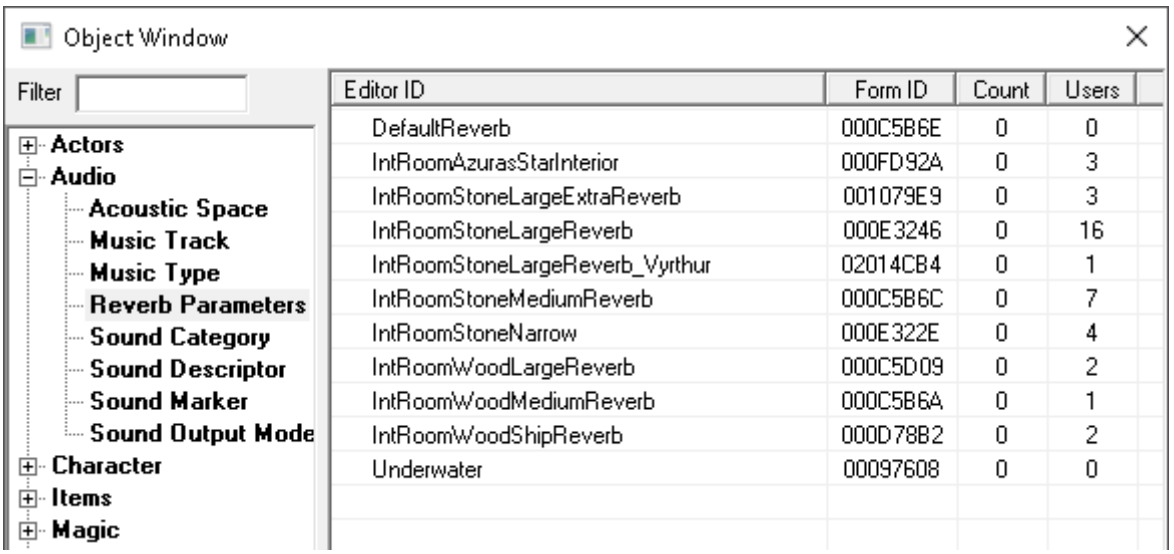


Figure 390 - Reverb parameters.

Create a new reverb parameter by right-clicking and selecting New.

Set the ID. For my example, I'm just going to call it MountainsReverb with a 'WT' prefix.

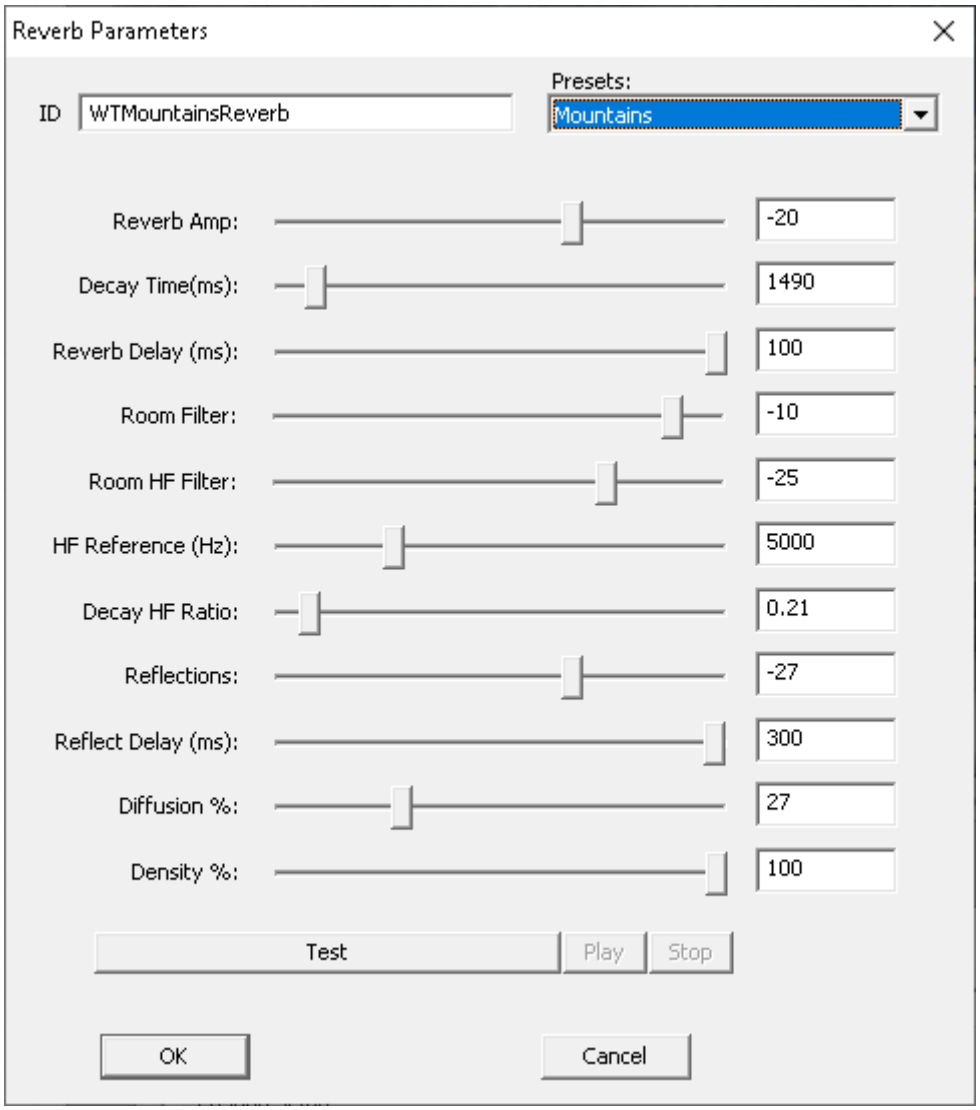


Figure 391 - New reverb parameters.

In the Presets drop-down I selected Mountains.

For a description of what each slider does, see the Reverb Parameters article on the Creation Kit wiki.

Click OK.

Now we can go back to the acoustic space we created earlier and change the 'Environment type' to the reverb parameters we just created.

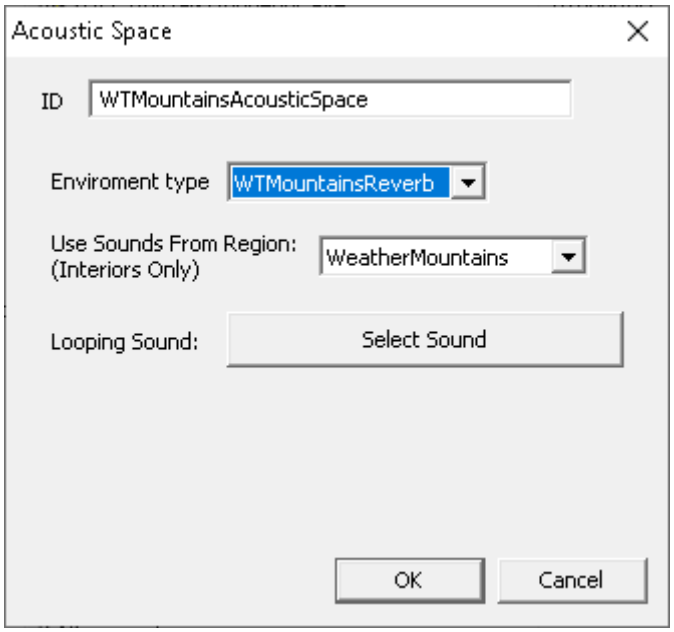


Figure 392 - Using our new reverb parameters.

Click OK to save and close.

For steps on how to set up region sounds, see the section on Regions.

ADDING MUSIC

Music can be added to a world space, a region within a world space, or to specific cells.

Music tracks need to be converted to the .xwm format first before they can be used in-game.

To convert a .wav or .mp3 file to .xwm we'll need the Skyrim Audio Converter utility which is available on Nexusmods.

Launch SkyrimAudioConverter.exe.

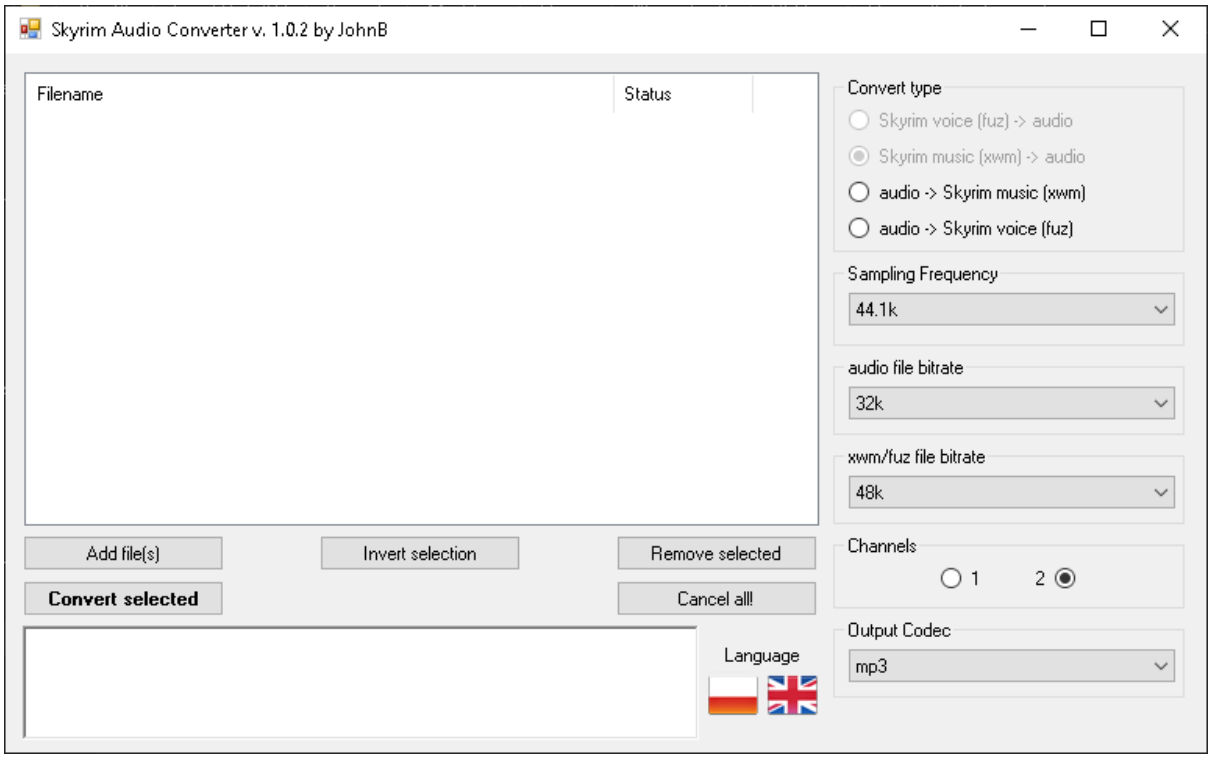


Figure 393 - Skyrim Audio Converter.

Under the 'Convert type' section, select 'audio -> Skyrim music'.

The other settings below it can be left at their defaults.

Click on the 'Add file(s)' button.

Select the audio file(s) you want to convert to .xwm and click Open.

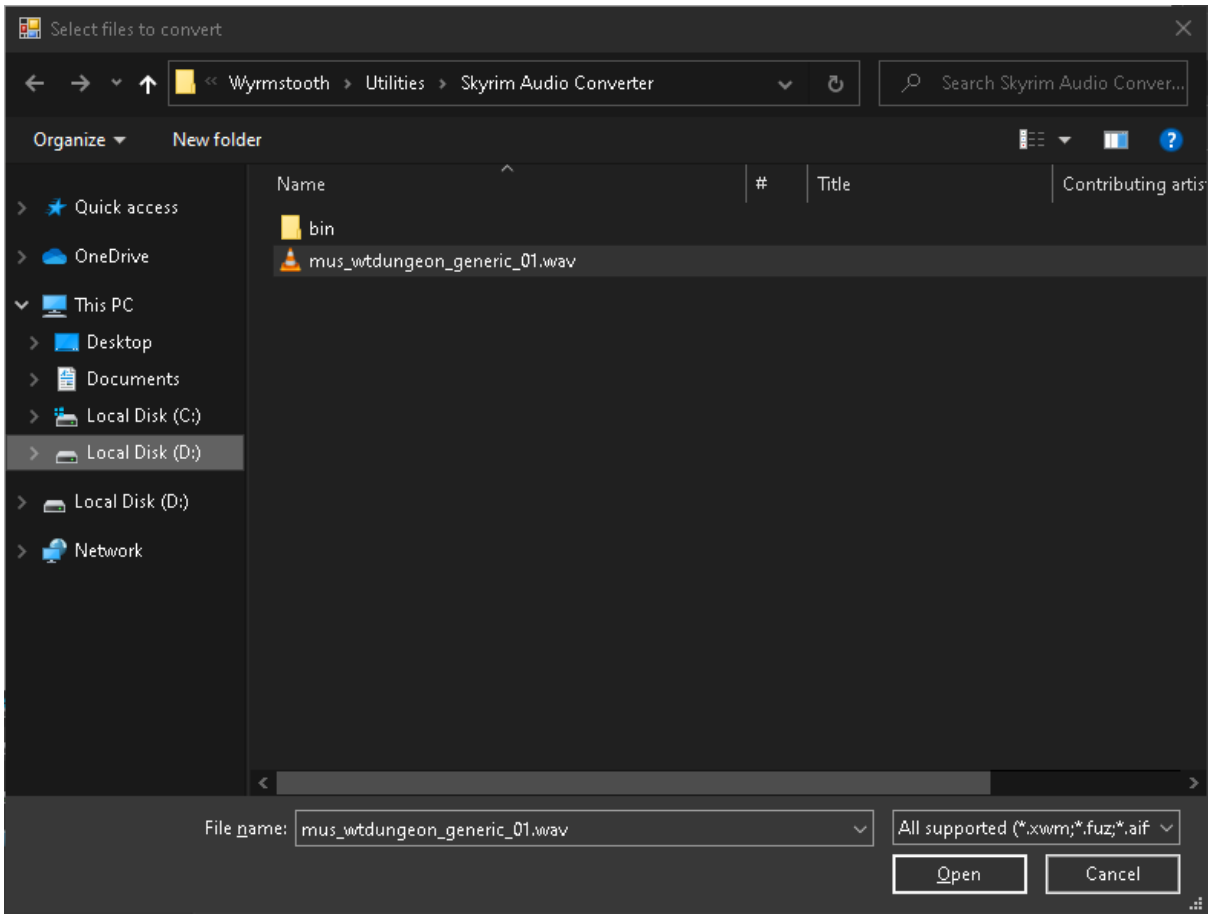


Figure 394 - Selecting the audio files to convert to .xwm.

In the example above, I'm going to be using a track composed by León van der Stadt specifically for the Wyrmstooth mod.

Confirm the audio file(s) are listed in the utility and the status shows 'Ready'.

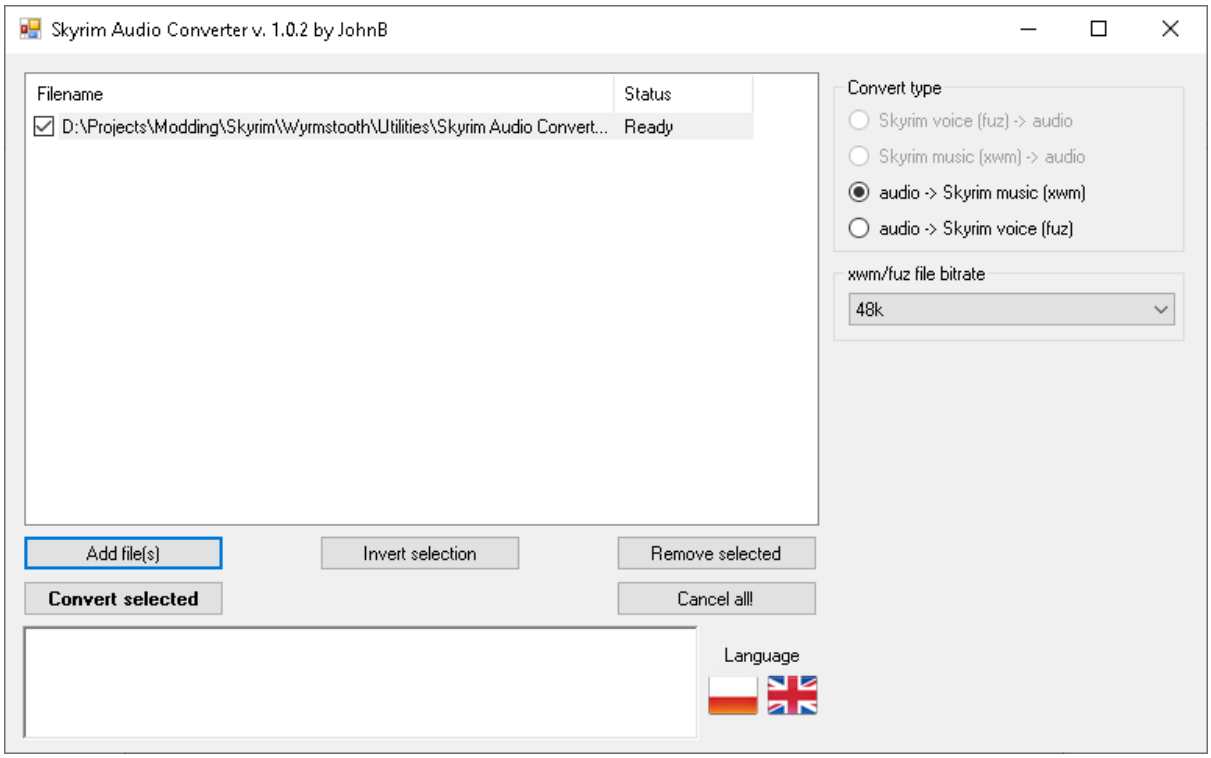


Figure 395 - Audio files to convert are listed in the utility.

Click on the 'Convert selected' button to begin converting the selected audio files to the .xwm format.

When conversion is complete, the audio files should be automatically unticked.

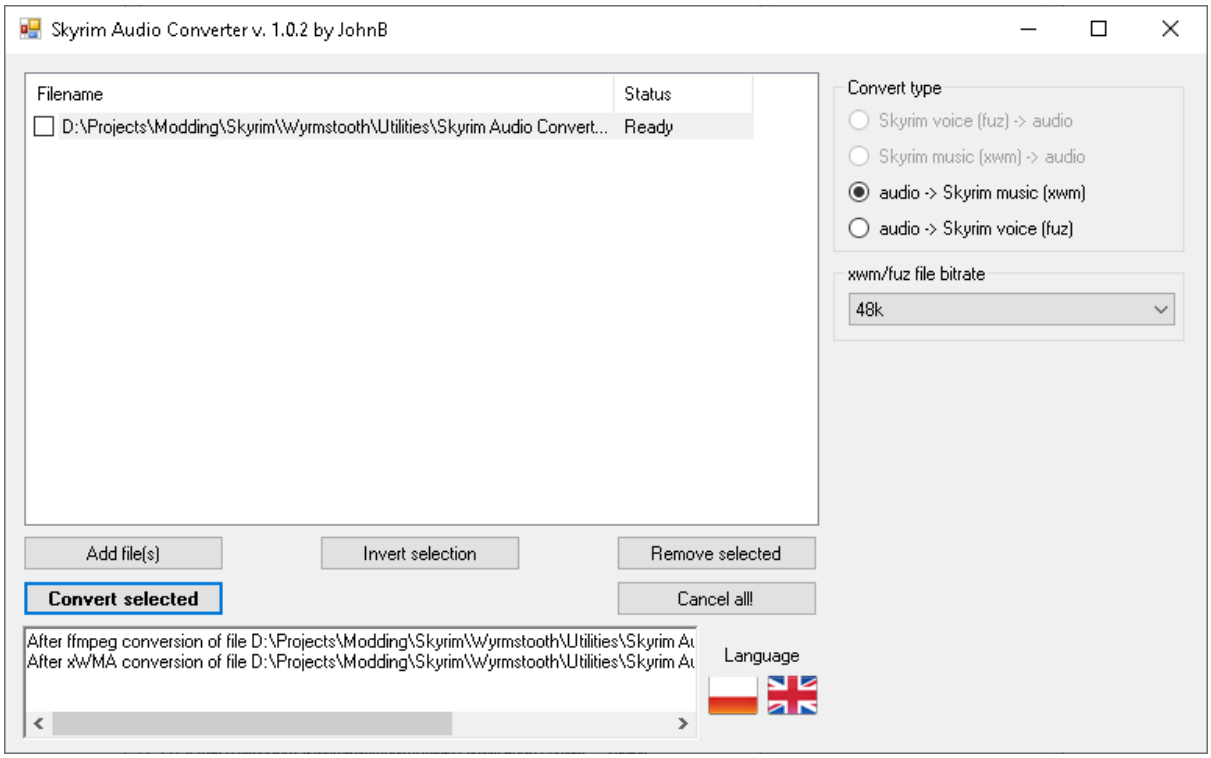


Figure 396 - Conversion finished.

Important: If your mod includes an .xwm music file and you pack it into a .bsa archive, I would recommend either not compressing the .bsa archive or use a third party utility to compress it instead. I've run into issues before where the game would crash when trying to play an .xwm music file from a .bsa archive compressed with Bethesda's archive.exe utility.

The new .xwm file(s) should be placed in the same folder you as the source audio files.

Next, we need to create new tracks for each .xwm file in the Creation Kit.

In the Object Window, go to Audio > Music Track.

Right-click on one of the existing tracks and select New.

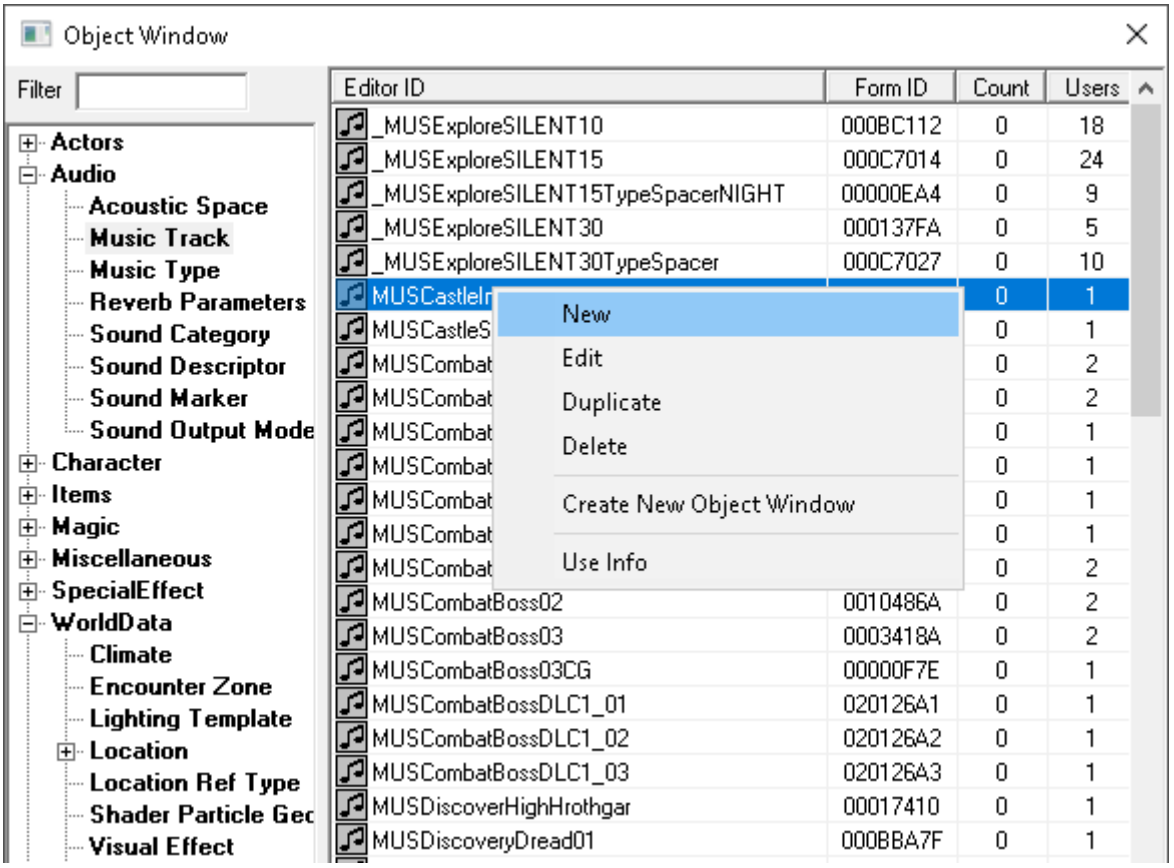


Figure 397 - Creating a new music track.

Set the Track Type to Single Track.

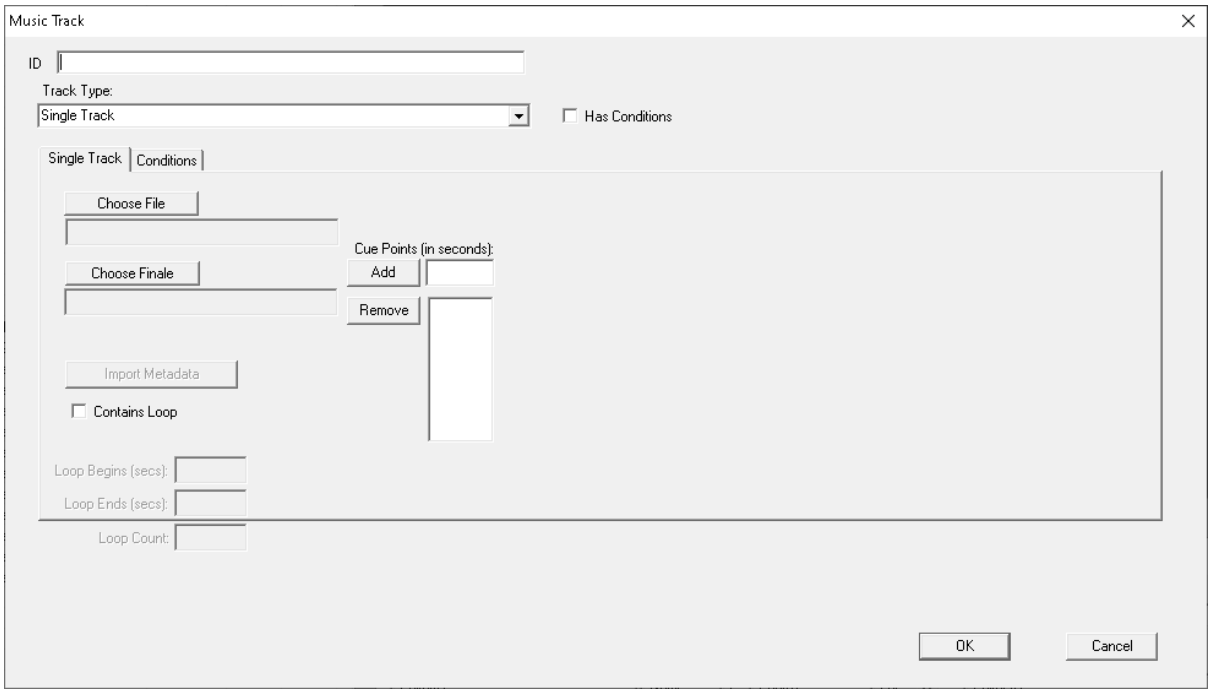


Figure 398 - A new music track.

Click on Choose File.

The file needs to be copied to your Skyrim\Data\Music directory. For example, for Wyrmstooth I put music tracks under Skyrim\Data\Music\wyrmstooth.

When choosing a file, the file extension needs to be in .wav format. If you've already converted the files to .xwm we'll need to change the file extension to .wav, add the file to the Music Track properties, then rename the extension back to .xwm.

Right-click on the music file you want to add and change its extension from .xwm to .wav.

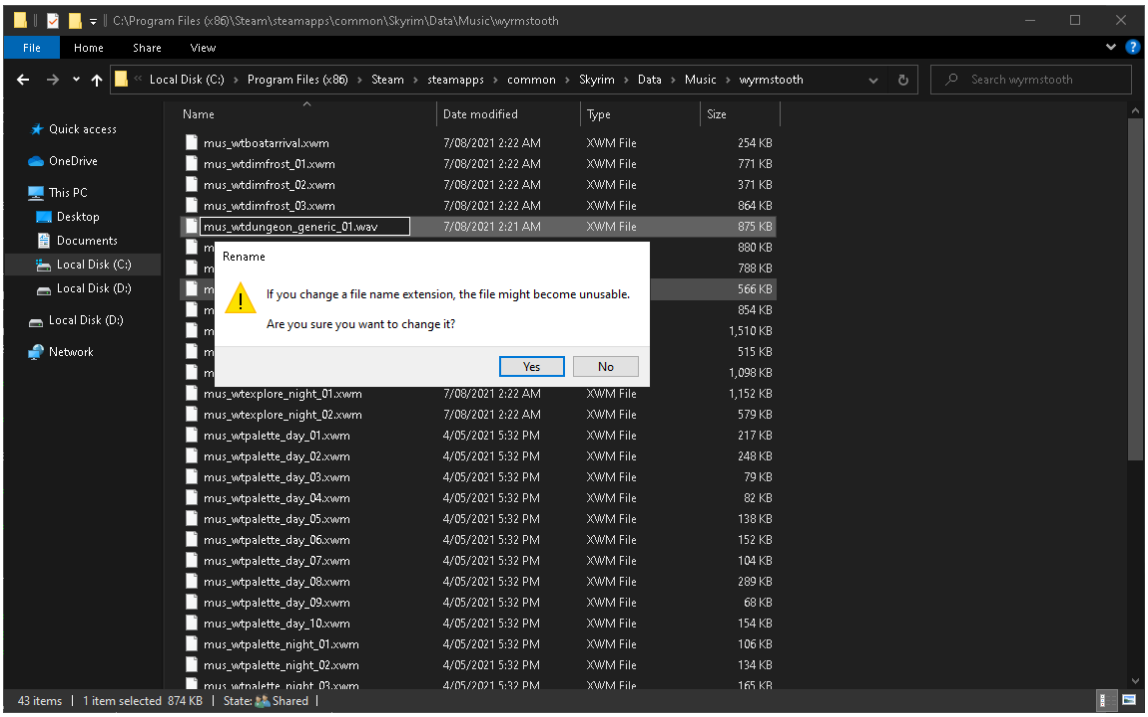


Figure 399 - Renaming the .xwm file back to .wav temporarily.

Now we can add it to the Creation Kit.

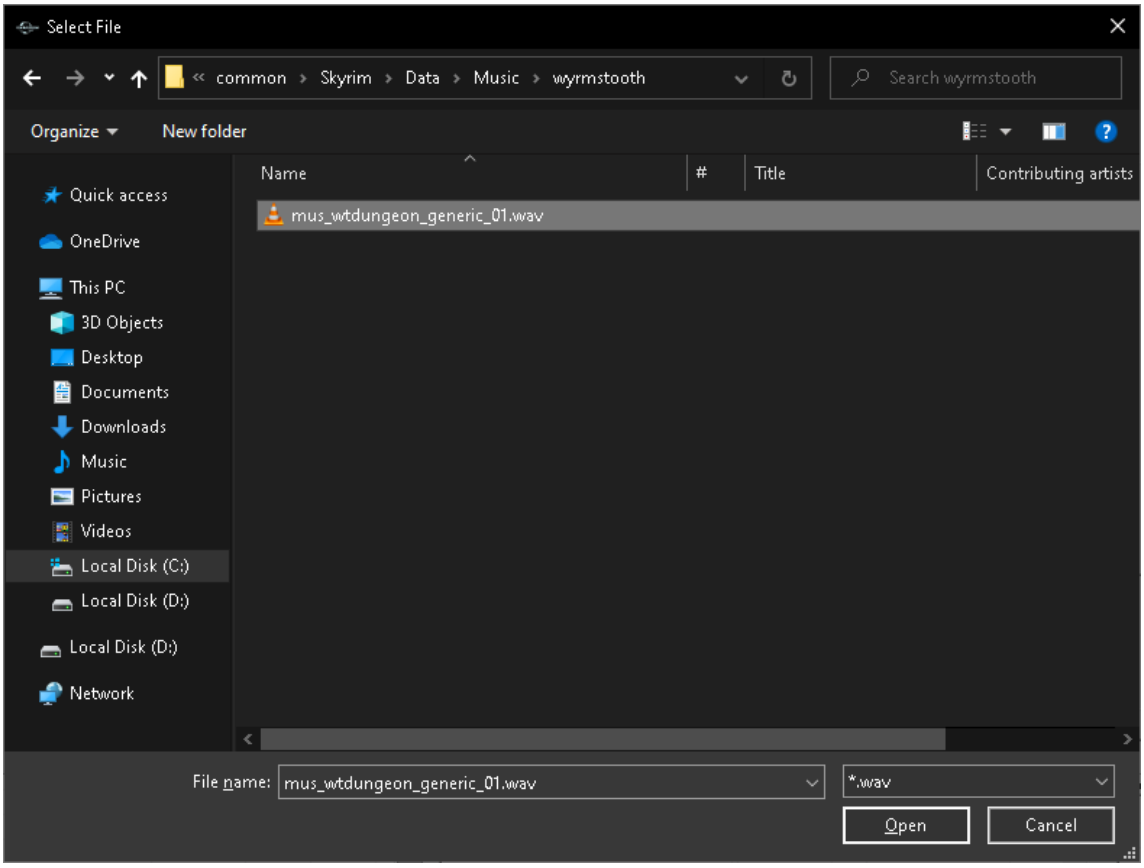


Figure 400 - Selecting the music file to add to the new music track.

Once the music file has been added as a .wav file, rename the .wav file back to .xwm.

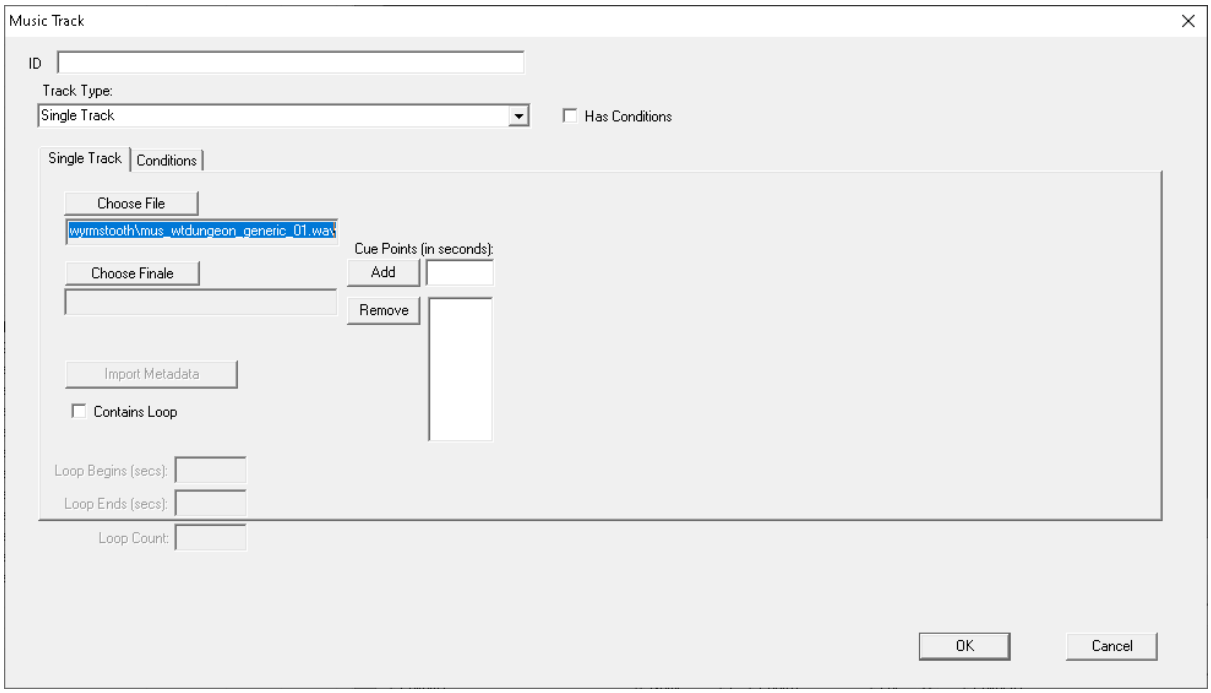


Figure 401 - Added as a .wav file in the new music track.

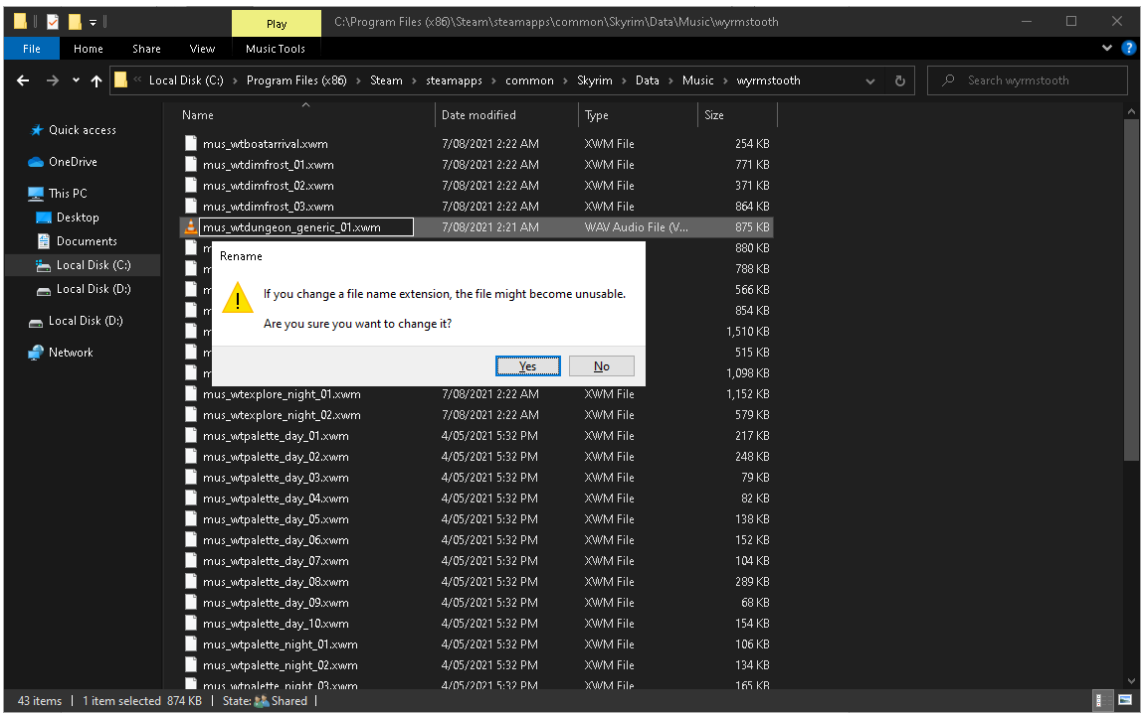


Figure 402 - Renaming the file extension back to .xwm.

Set the ID. In my example, I'm going to call the track MUSDungeon01 with the 'WT' prefix.

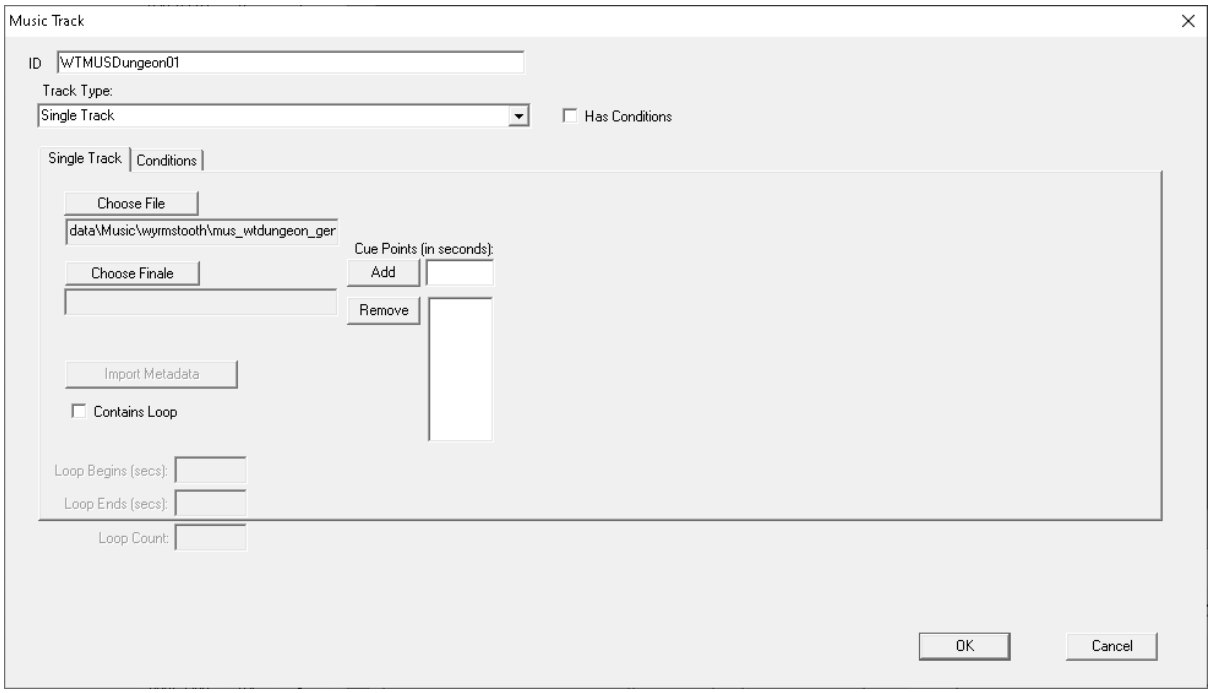


Figure 403 - Added the track ID.

If you want to add conditions to the track to only play the track under certain circumstances, tick 'Has Conditions'.

In the following screenshots, I'll be showing you a track that I set up only to play during daytime hours.

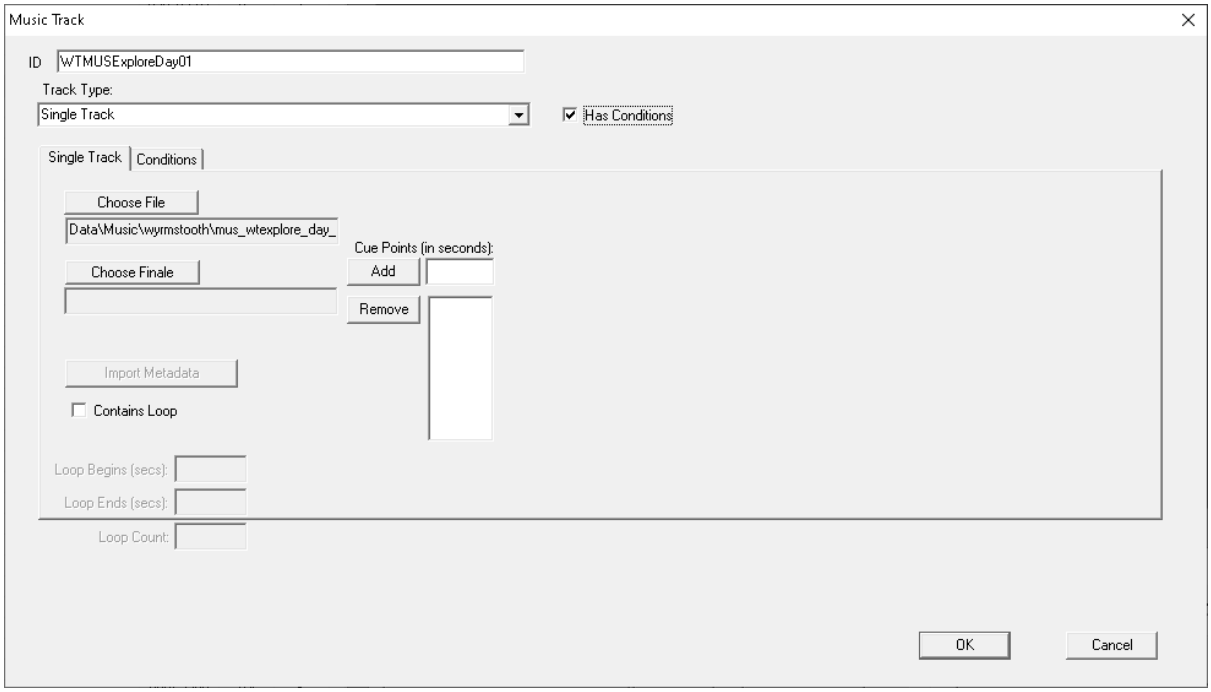


Figure 404 - Enabling conditions on a music track.

Go to the Conditions tab.

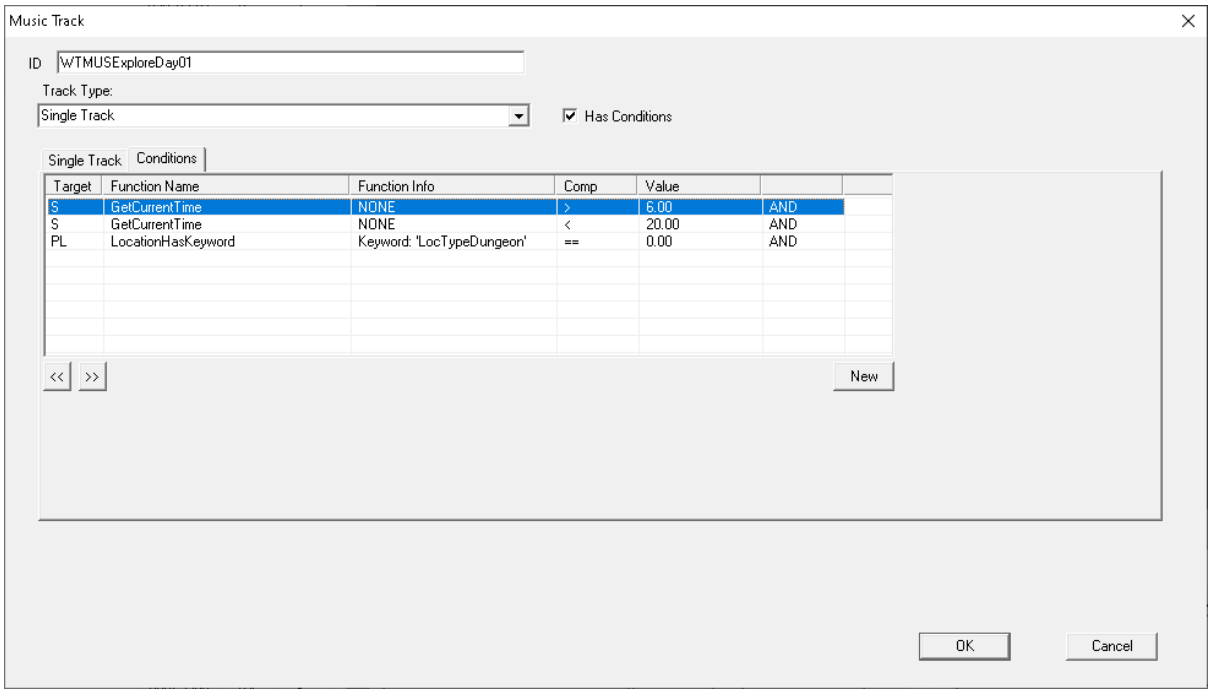


Figure 405 - Music track conditions.

In the screenshot above, I added two GetCurrentTime conditions to only play the track during daytime hours. I also added a LocationHasKeyword condition to ensure the track doesn't play in dungeons, where I want a different music track to play.

To add a new condition, right-click on the conditions list and click New.

Click OK when you're done here to close out of Music Track properties.

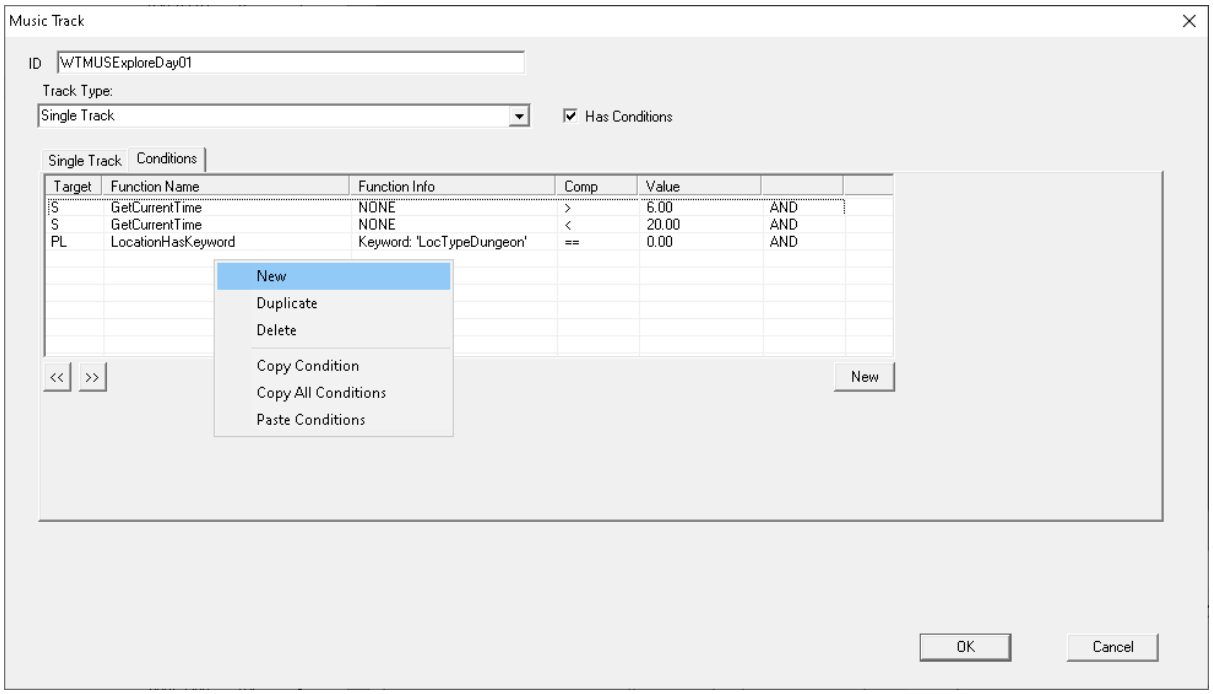


Figure 406 - Adding a new condition.

Important: It's important to mention that adding, modifying or deleting Music Types may eventually cause the Creation Kit to crash, so save your mod now before that happens.

Next, we'll need to create a new Music Type.

A Music Type is a collection of music tracks that can be referenced by a world space, region, location, or cell.

To create a new music type, in the Object Window go to Audio > Music Type. Right-click on one of the existing music types and select New.

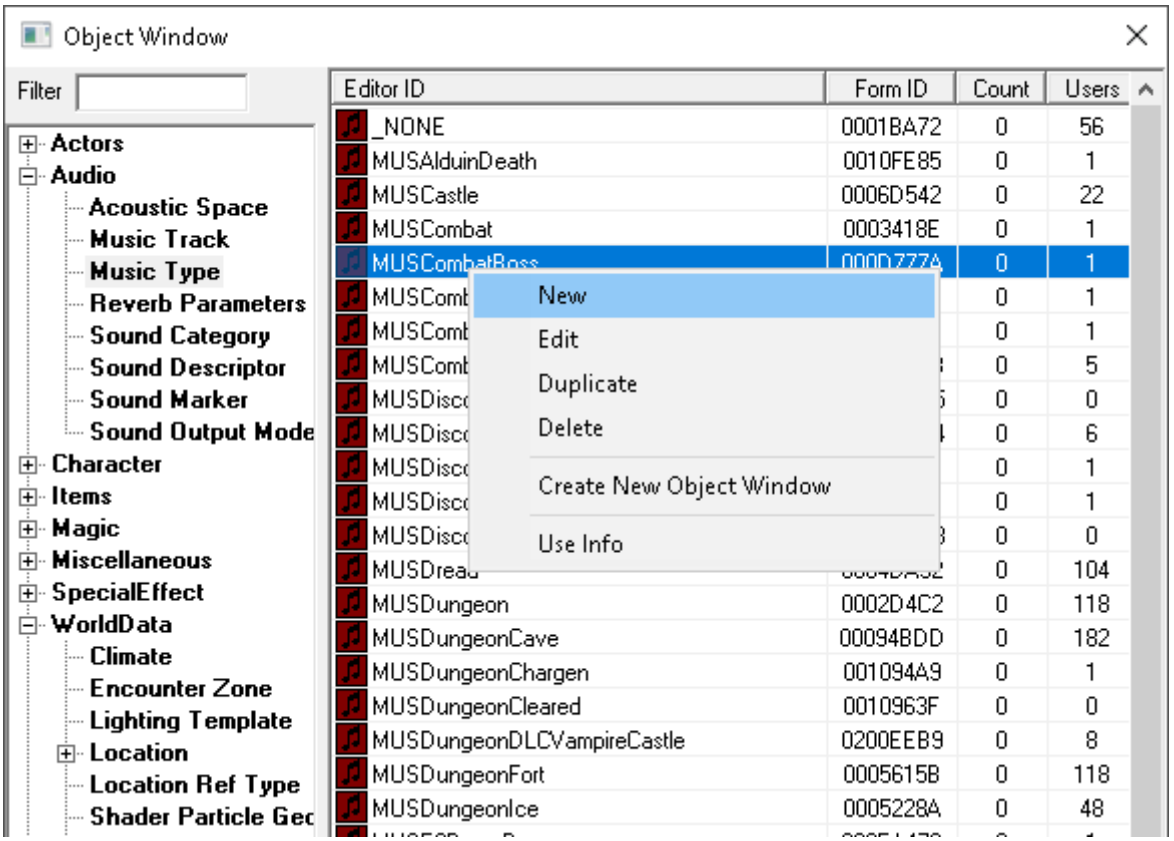


Figure 407 - Creating a new music type.

To add a music track, right click in the track list and select New.

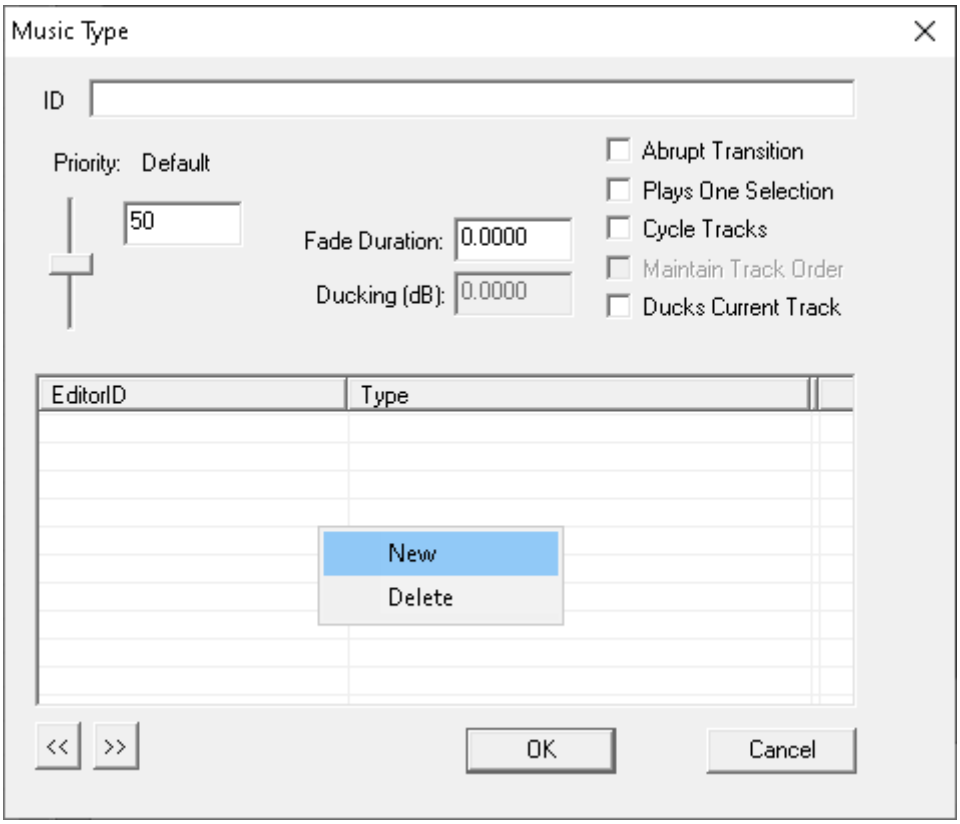


Figure 408 - Adding a new track to the music type.

Select the tracks you want to add and press OK.

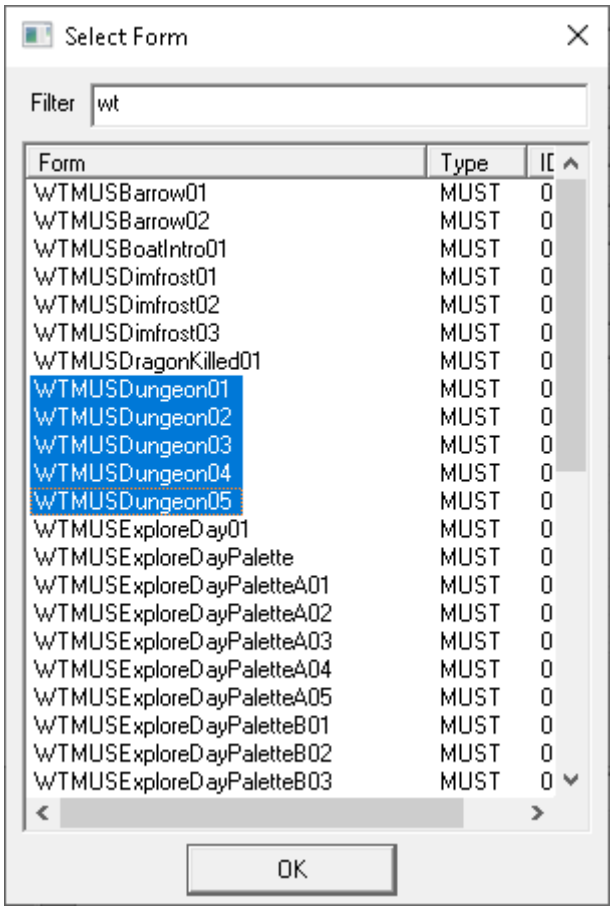


Figure 409 - Selecting the tracks to add.

Set the ID. For my example, I'm going to add it MUSDungeon and add the 'WT' prefix.

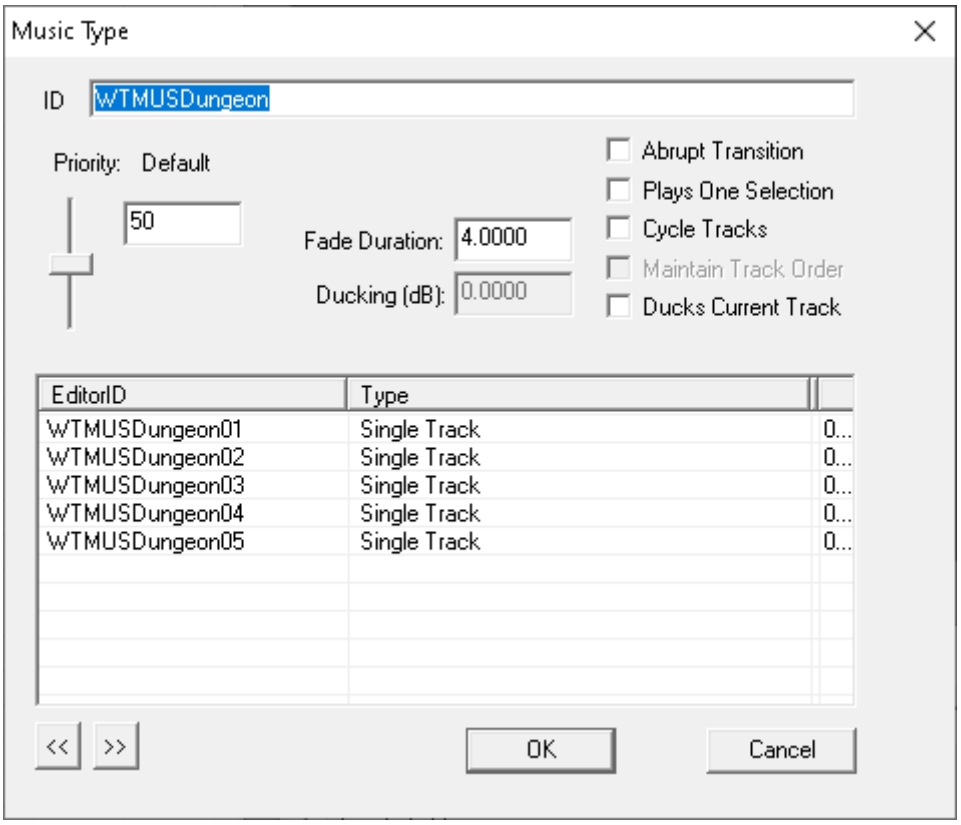


Figure 410 - Configuring the music type.

I also set the Fade Duration to 4.0000 to fade between tracks.

Click OK to close out of the music type properties.

To add a music type to a world space, go to World > World Spaces.

Select the world space you want to add it to from the list down the left hand side.

Set the music type you want to play in the Music field then click OK.

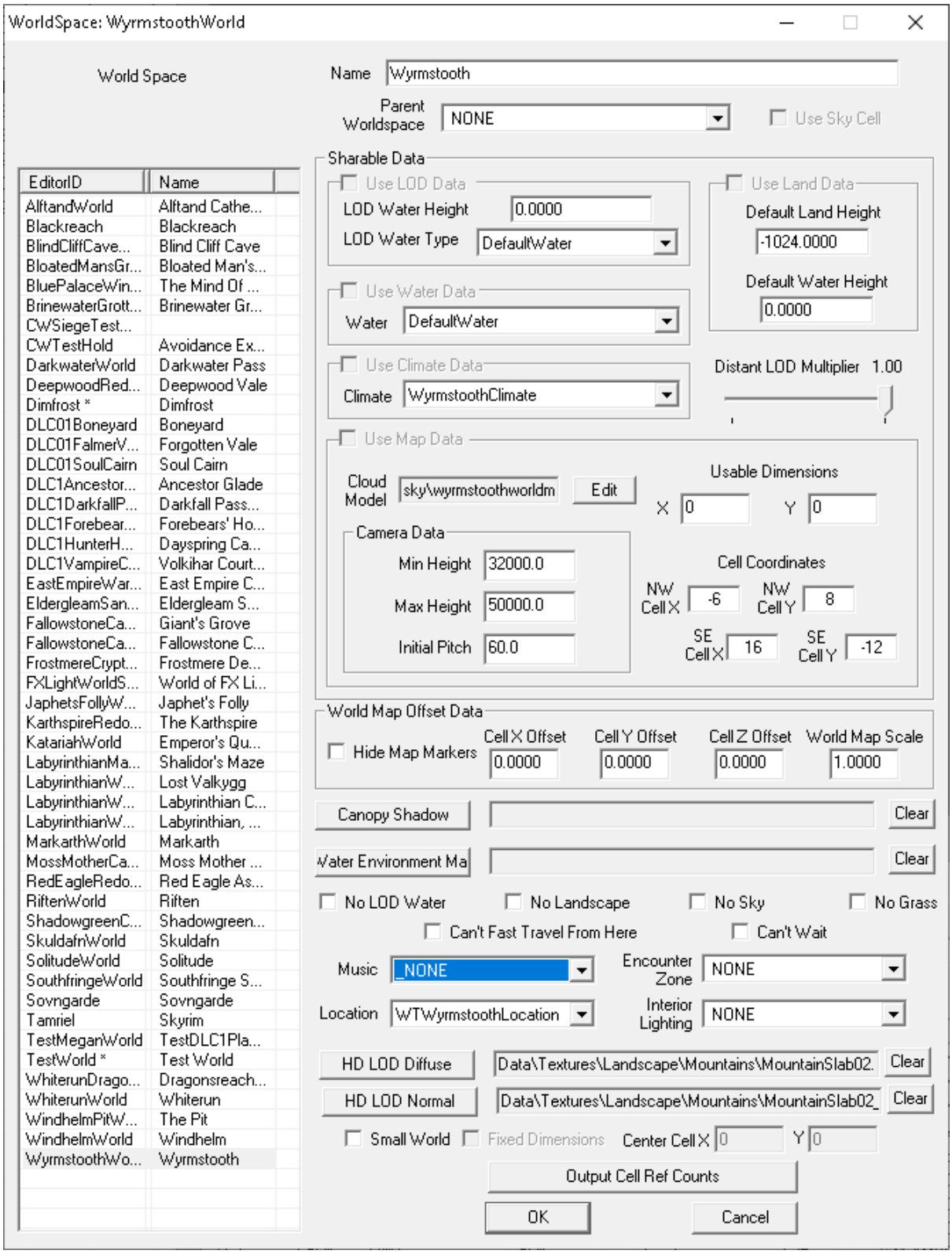


Figure 411 - Adding a music type to a world space.

To add a music type to a specific region only, go to World > Regions.

Select the world space in the drop-down menu then click on the region in the list below it.

Go to the Sound tab and set the music type you want to play in the Music Type field then click OK.

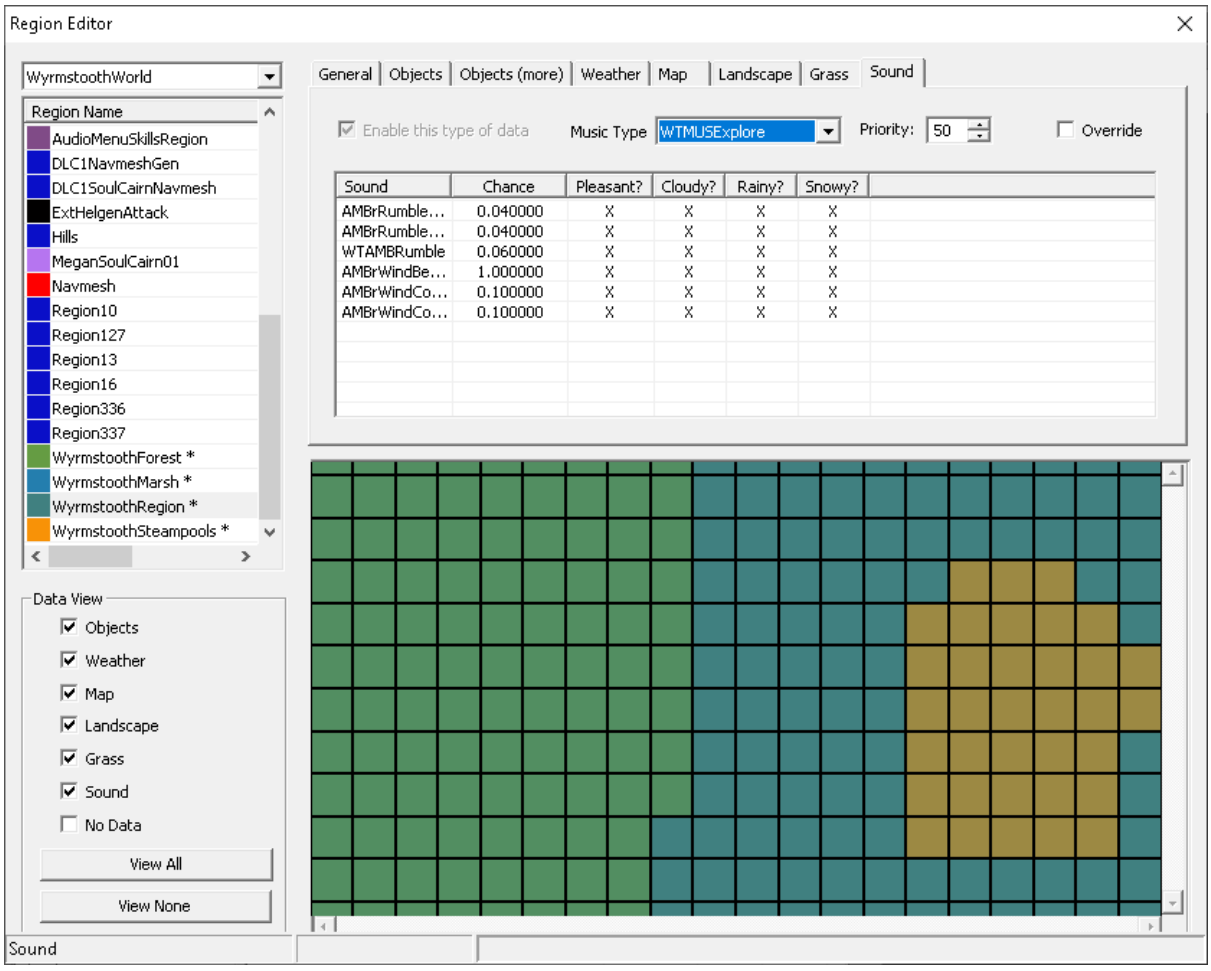


Figure 412 - Adding a music type to a region.

For Wyrmstooth, I set the day/night exploration music on the border region (WyrmstoothRegion) rather than on the world space, which is the same way the default Tamriel world space is set up.

To add music to a specific cell, right-click on the cell in the Cell View window and select Edit.

Set the music type you want to play in the Music Type field then click OK.

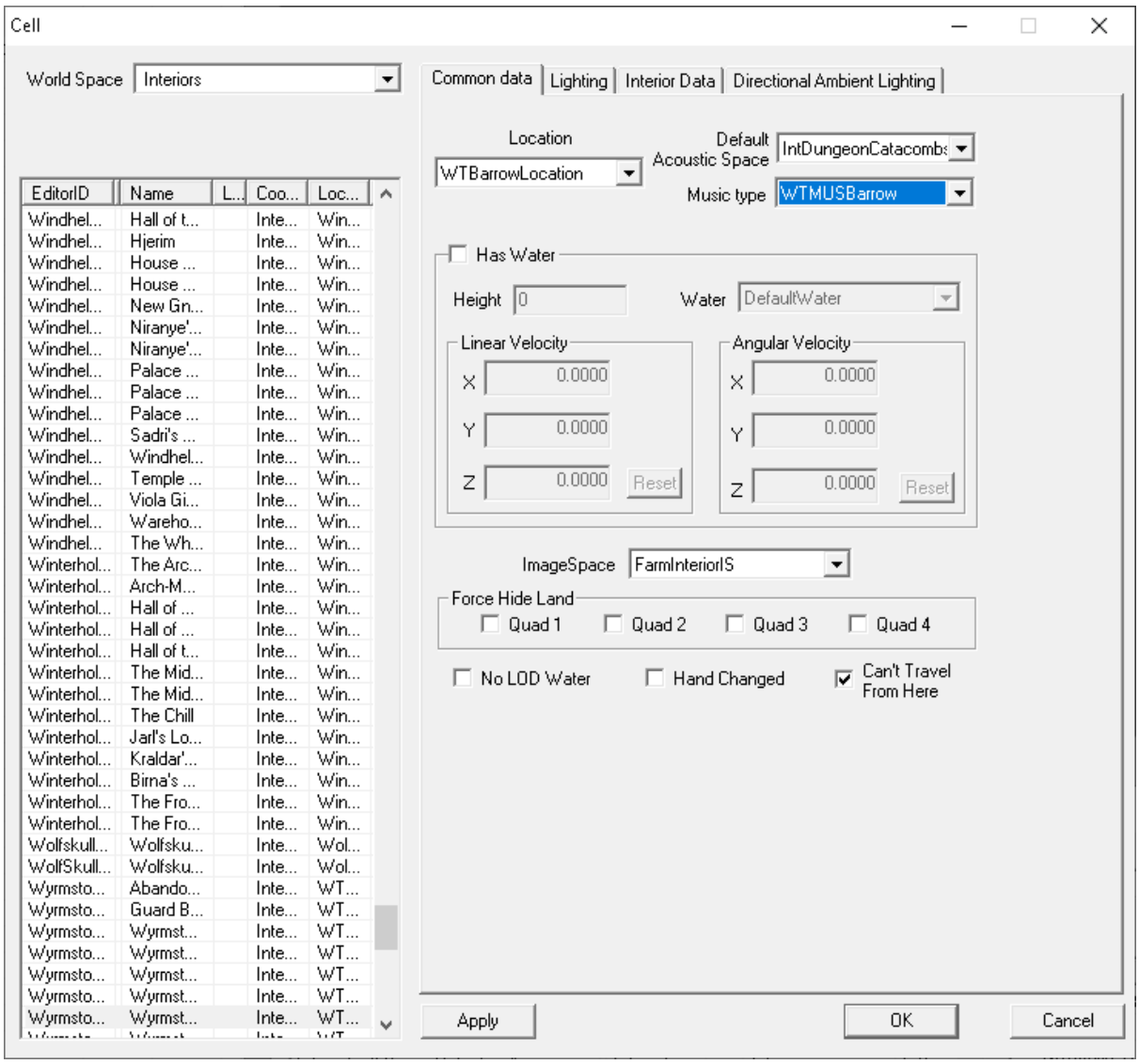


Figure 413 - Adding a music type to a cell.

In the screenshot above, I'm playing the WTMUSBarrow music type in the Wyrmstooth Barrow cells instead of the day/night exploration music to help build a different kind of mood for the player when they're exploring this dungeon.

A music type can also be assigned to a location, and will play in any cells that have been assigned to that location.

Go to WorldData > Location and open the location you want to assign a music type to.

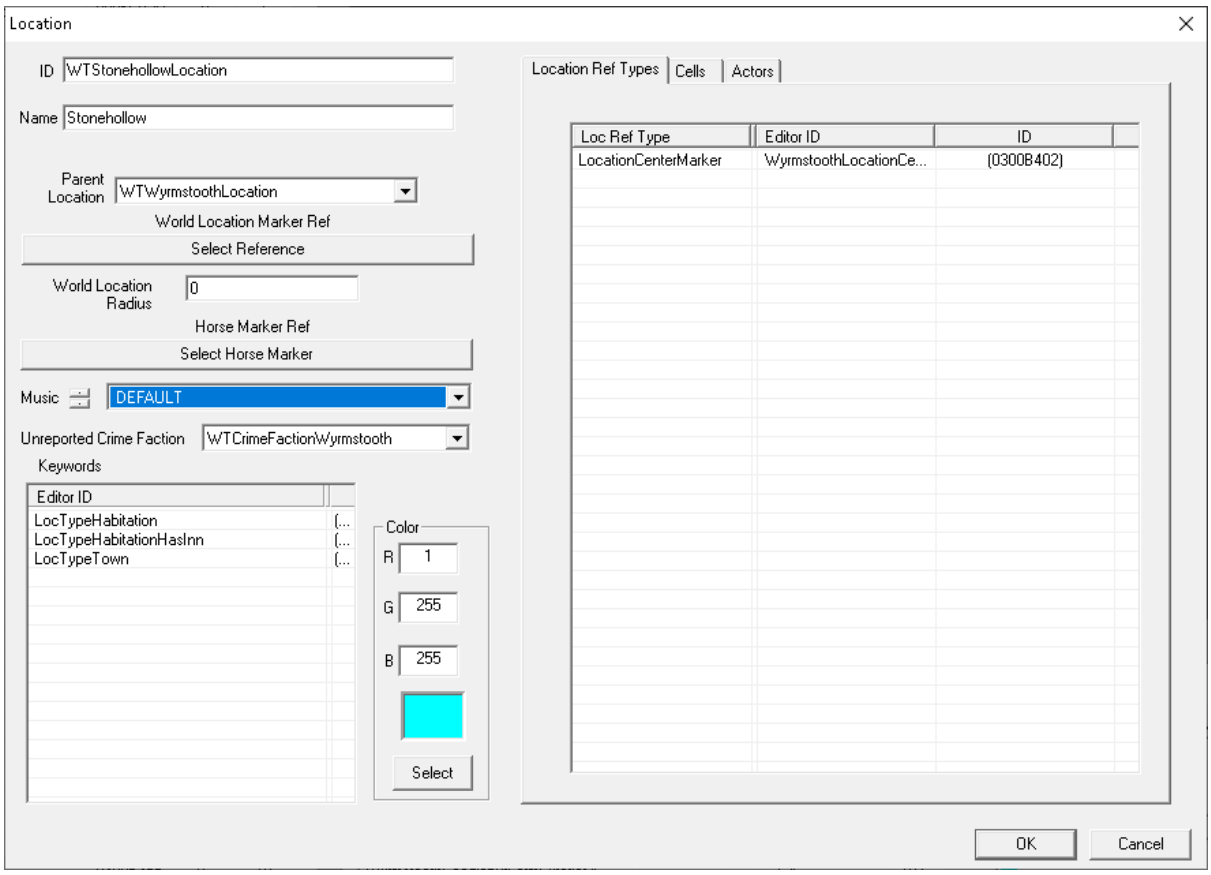


Figure 414 - Specifying a music type for a location.

The music type can be assigned in the Music drop-down.

Click OK to close out of Location properties.

Some tracks can be scripted to play at specific points in a quest. Increasing its priority and ticking 'Ducks Current Track' will play this track above any other currently playing tracks.

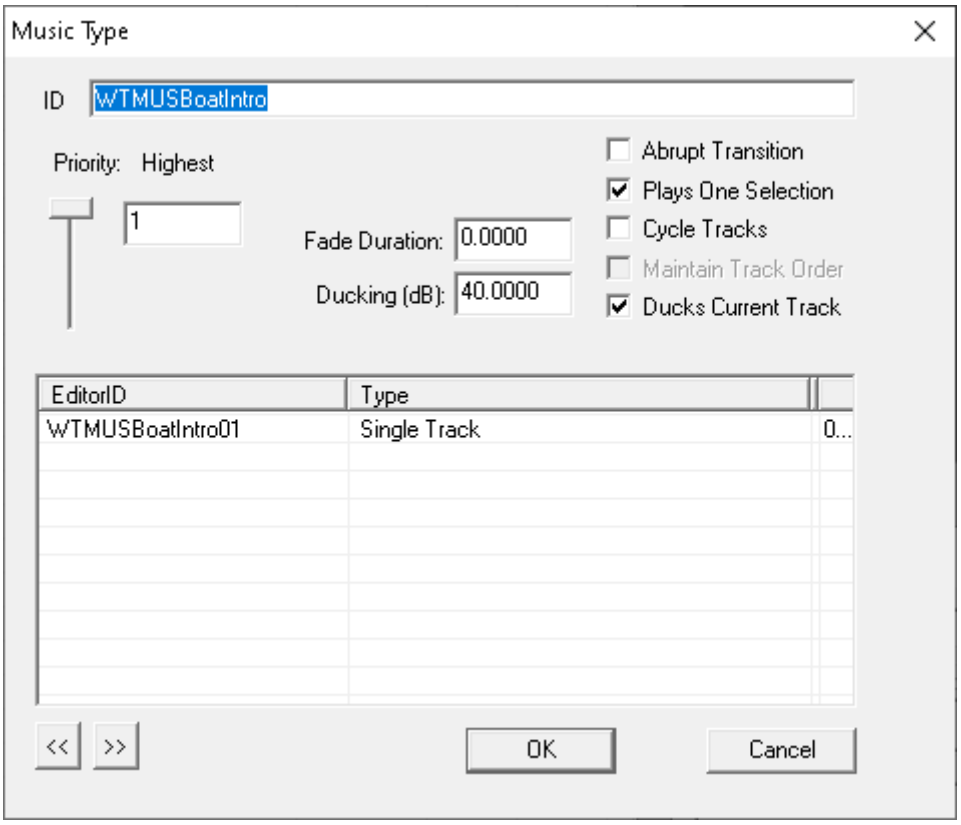


Figure 415 - Music type for a high priority scripted track.

We can then call this track using papyrus scripting. In the following example, I added BoatMusicIntro as a MusicType property on the 'Barrow of the Wyrm' quest in Wyrmstooth.

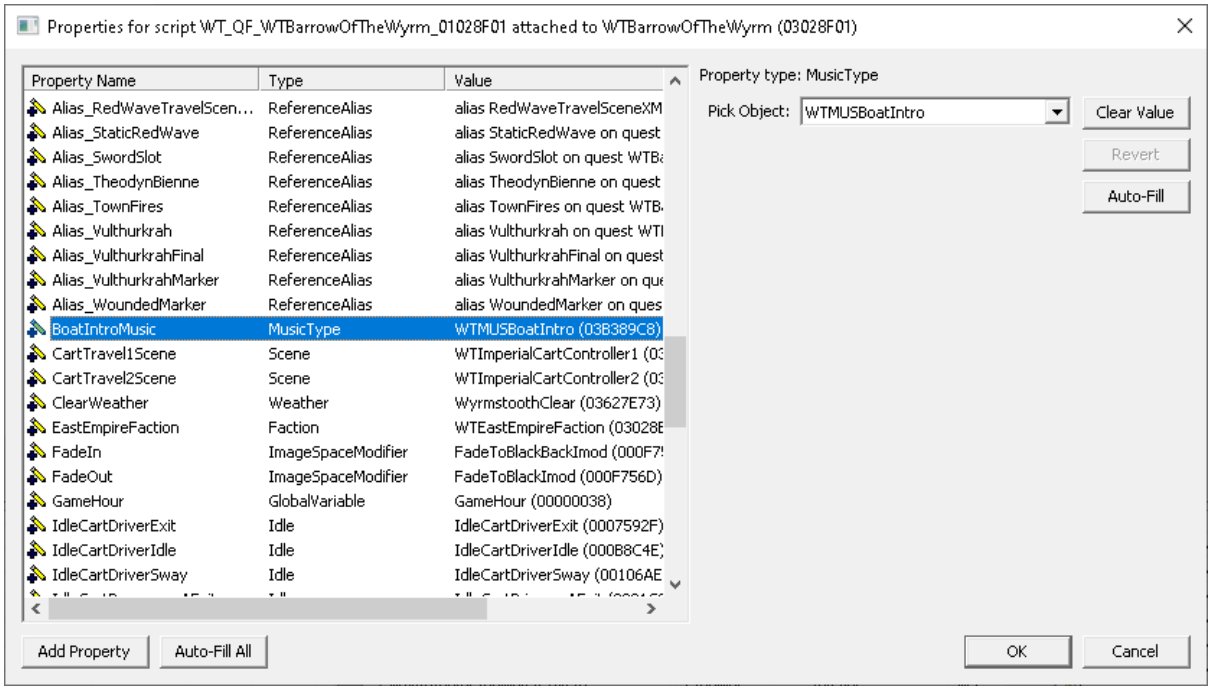


Figure 416 - Music type added to a script's properties.

I reference this music type on stage 50 with the following command:

```
BoatIntroMusic.Add()
```

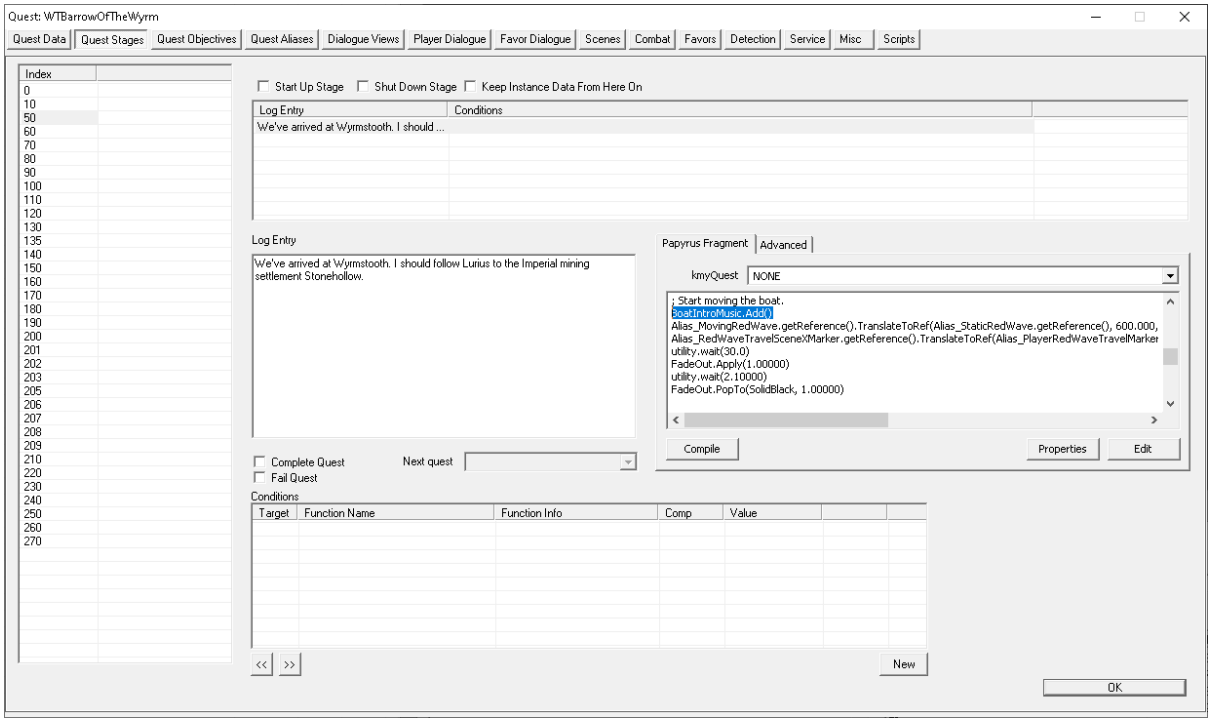


Figure 417 - Papyrus on stage 50 of Barrow of the Wyrm.

For more information on the Add() command, see the article on the Creation Kit wiki.

You can also force music to stop playing with the Remove() command.

Lastly, let's take a look at adding music triggers.

Music triggers allow us to play a track at a specific location. For example, when entering a large cave to emphasize how amazing it looks, or just before a boss fight to impose a sense of dread on the player.

First, click on an object in the render window roughly where you want to create the music trigger.

Click on Create Trigger in the toolbar.

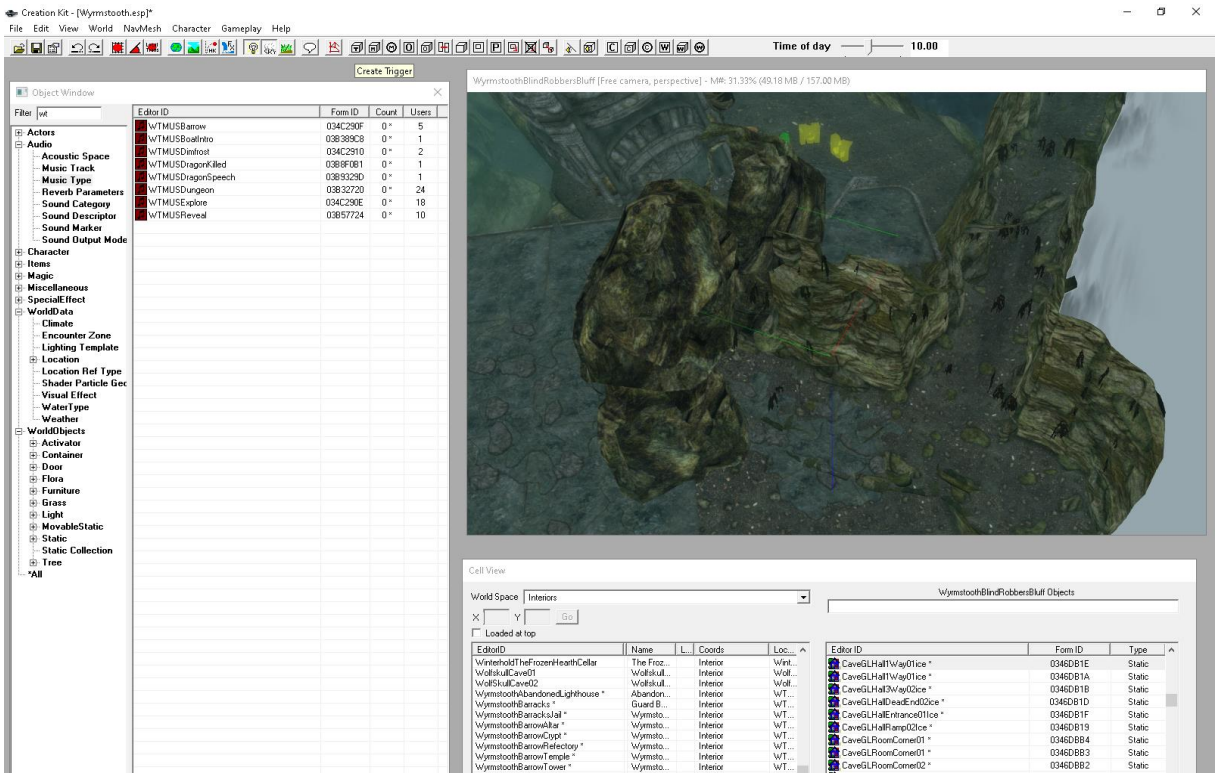


Figure 418 - Creating a new music trigger.

Filter by ‘music’ and select ‘defaultAddMusicTrigger’.

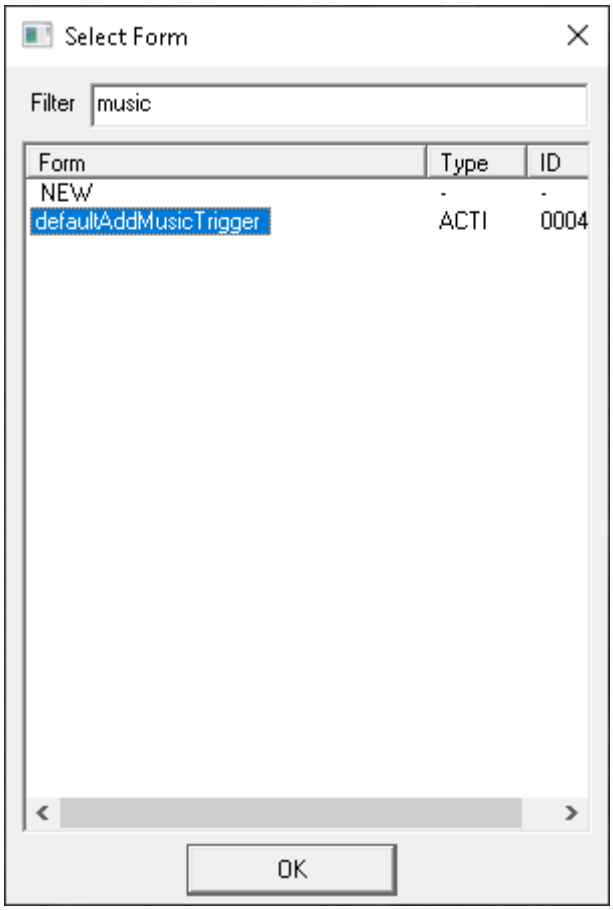


Figure 419 - Selecting defaultAddMusicTrigger.

A trigger box should appear in the render window covering the object you had selected.

Use the resize gizmo to change the width and height of the trigger box.



Figure 420 - Resizing the music trigger.

In my example, I'm adding the trigger to a pathway into a large cavern, so I'd need to make sure the trigger is wider than the pathway so the player can't walk around it. It also needs to be tall enough so the player can't jump over it.

In the Cell View window, right-click on the trigger and select Edit.

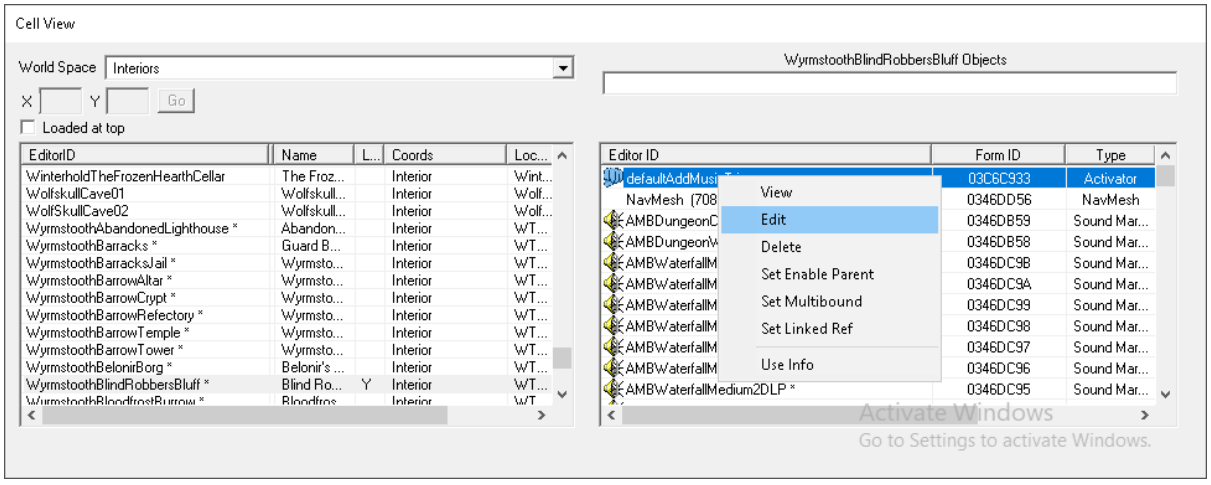


Figure 421 - Editing the music trigger's reference properties.

Go to the Scripts tab, ensure defaultAddMusicSCRIPT is highlighted then click on the Properties button.

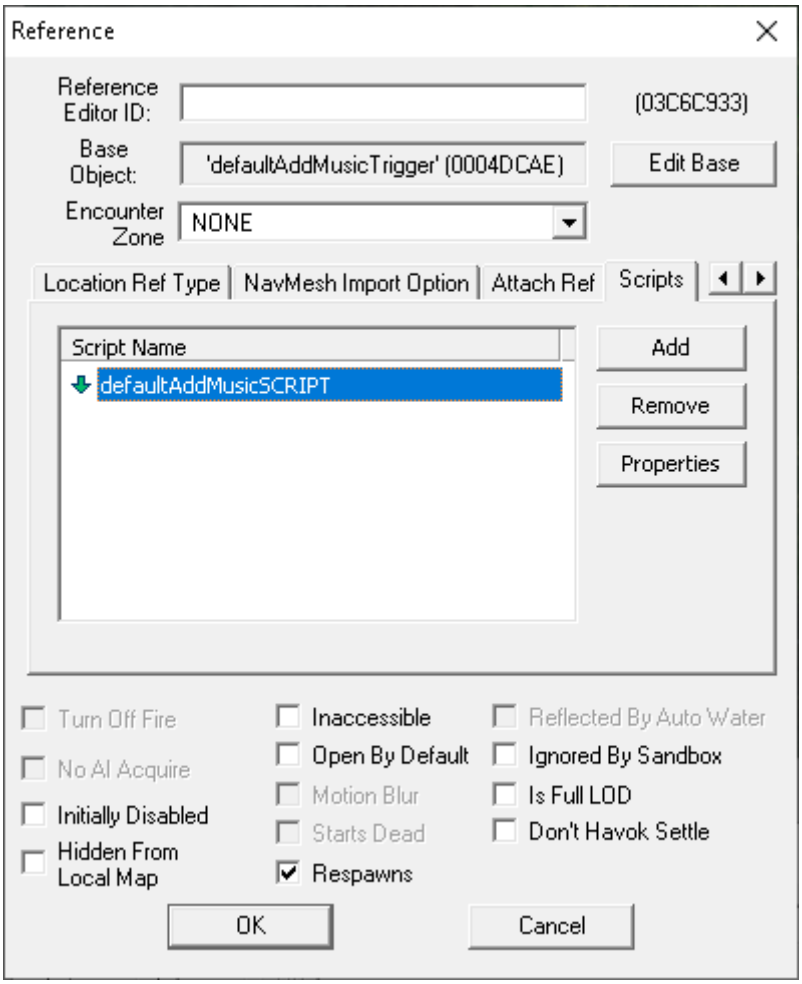


Figure 422 - Scripts tab.

Highlight myMusic and click on the Edit Value button.

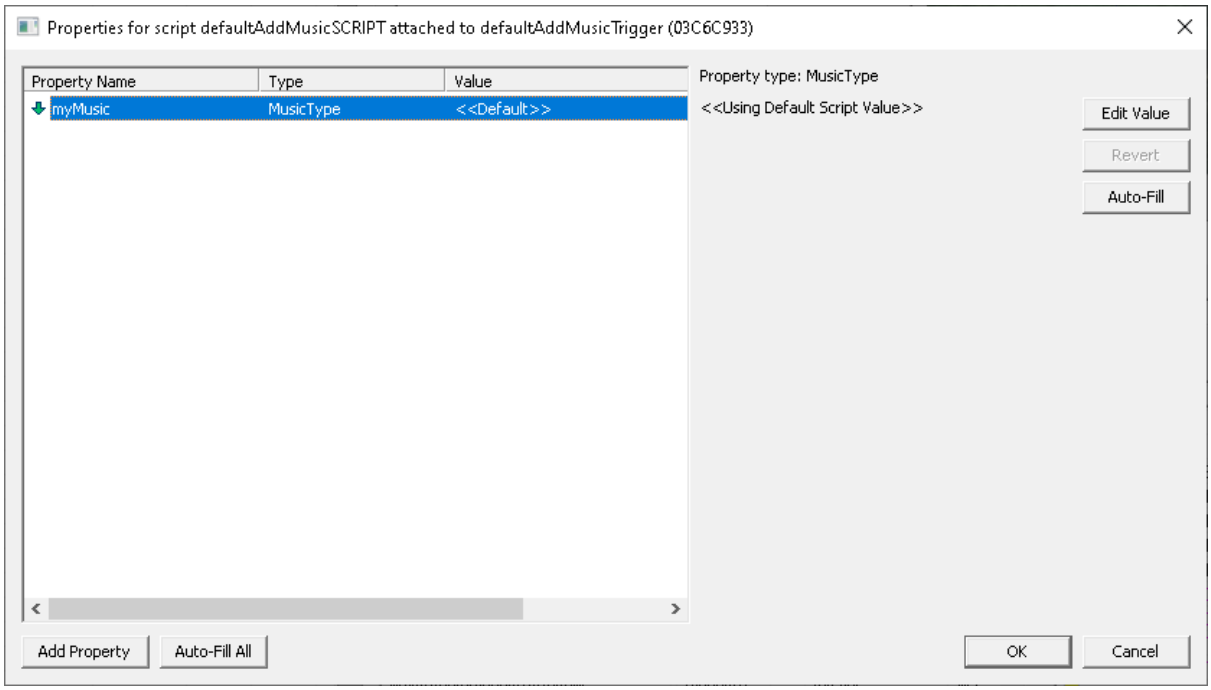


Figure 423 - Changing the music type that will be played.

Choose the music type you want to play when the player enters this trigger.

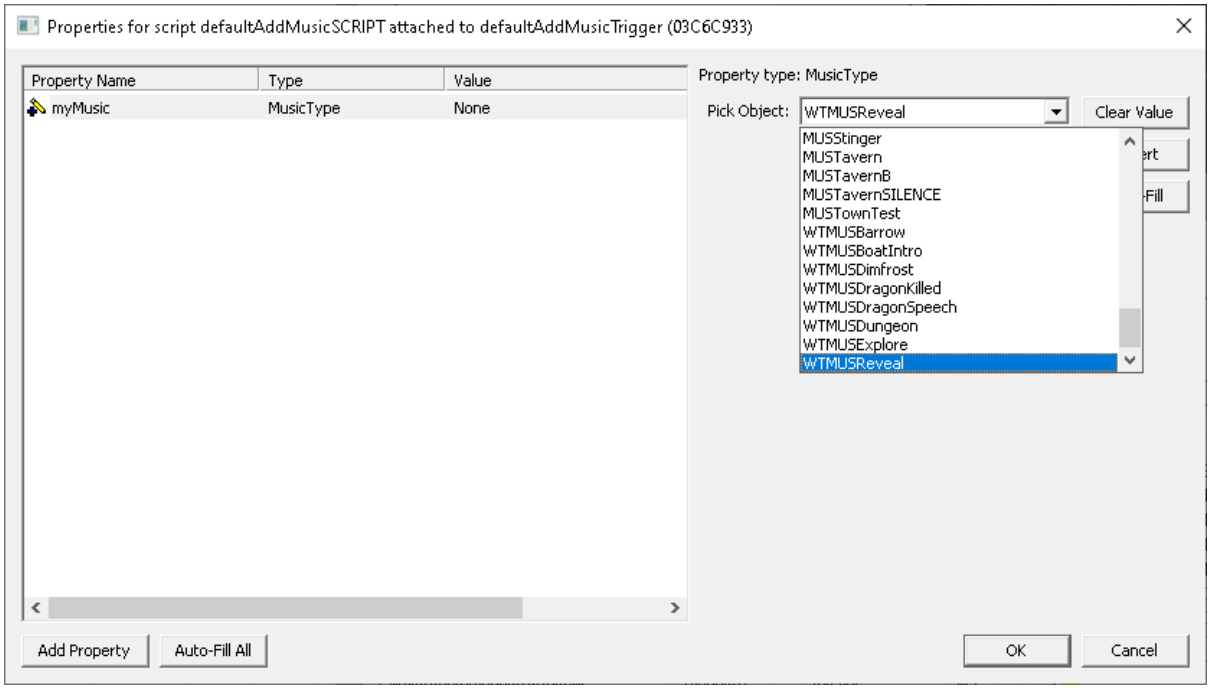


Figure 424 - Choosing the music type.

In the example above, I'm going to pick a custom music type. In the base game, MUSReveal is played when entering amazing looking places, and MUSDread is typically played before boss fights or to foreshadow that something bad is about to happen.

Click OK to close out of the script properties window.

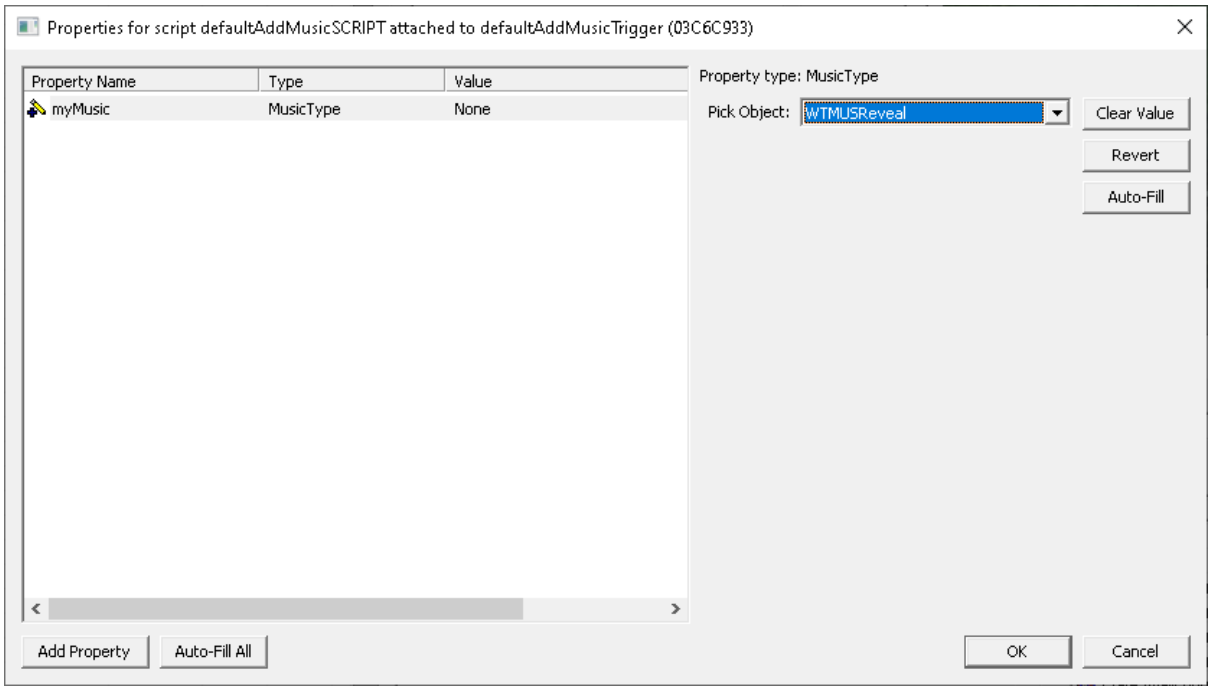


Figure 425 - Our music type has been set.

Click OK again to close out of the reference properties window.

The track you specified should now play in-game when the player passes through the trigger box.

CHAPTER 4: NPCS

CREATING AN NPC

In this section I'll be stepping you through the basics of creating a new NPC.

To create a new NPC, in the Object Window browse to Actors > Actor > Actor.

Right-click on one of the existing NPCs and select New.

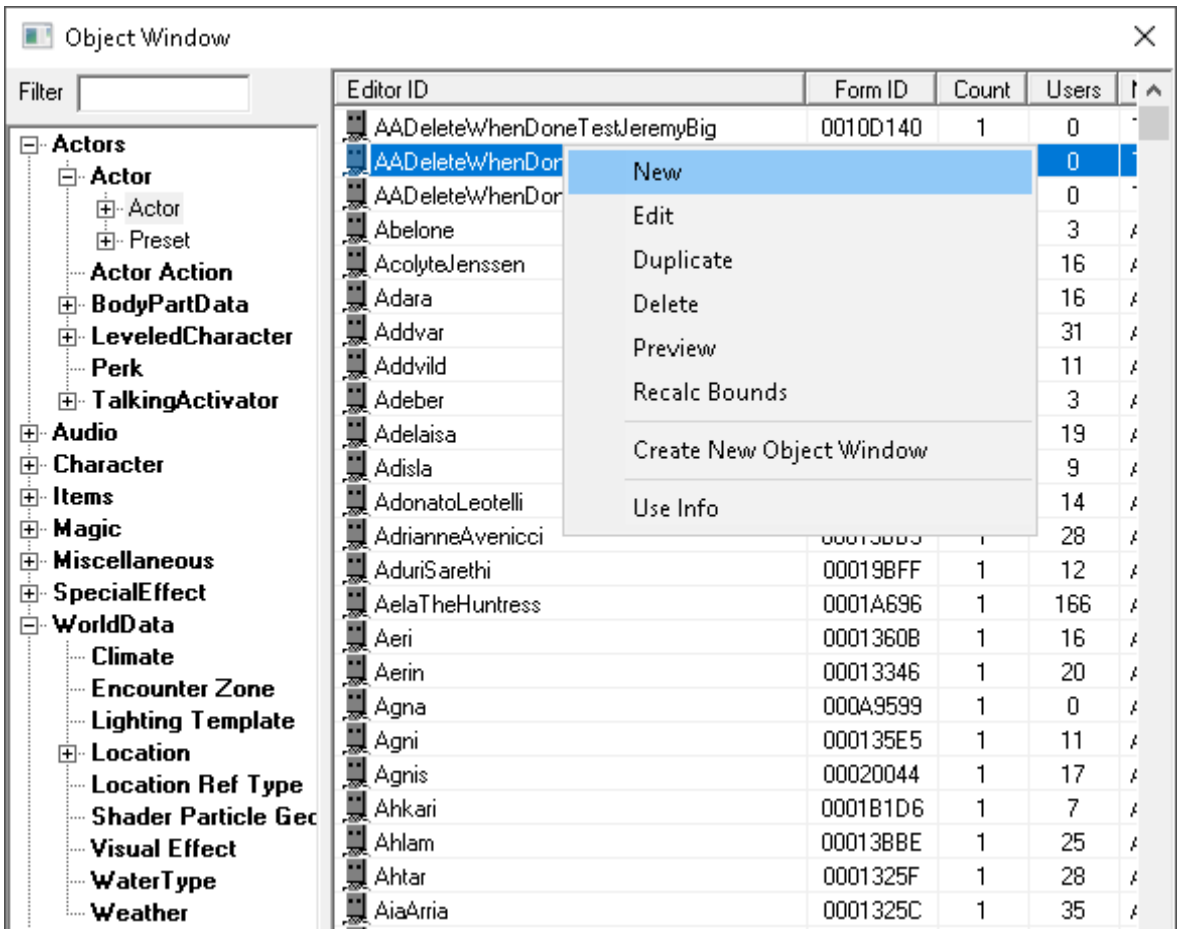


Figure 426 - Creating a new NPC.

The first thing we need to set is a unique ID for this actor base record. In the following example I set the ID to WTJornTester and set his name in the Name field to 'Jorn the Tester'. The text in the Name field is the NPC name you'll see in-game.

Let's take a look at some of the tickboxes on the left hand side first.

- Essential: Makes this NPC unkillable.
- Protected: Makes this NPC unkillable except by the player.
- Respawn: This NPC will be resurrected when the cell it's in resets.
- Unique: Allows the story manager to reference this NPC even if it has been unloaded. Since I only plan on having one instance of this NPC out in the world I'm going to tick it for this example.
- Summonable: Needs to be ticked when creating an NPC that you can summon with a spell.
- Is Ghost: Makes this NPC unhittable.
- Invulnerable: Makes this NPC unable to take damage, but can still be hit.
- Doesn't Bleed: Disables blood splatter when hit.
- Simple Actor: Disables facial animations and death and assault events.
- Doesn't affect stealth meter: Tick this for NPCs you plan on turning into a follower.

I'm going to make this character a nord, so I set the Race drop-down to NordRace.

For this example, I'm going to use the MaleNord voice type. If you plan to give this NPC unique voice acting and had previously set up a custom voice type, you would need to select that voice type here.

Ticking the Female tickbox would make this NPC use the female model instead. If the NPC is female, ticking Opposite Gender Anims would make the NPC play the male standing and walking animations instead, and vice versa.

Important: Changing an NPC's height may affect the accuracy of enemy spellcasters.

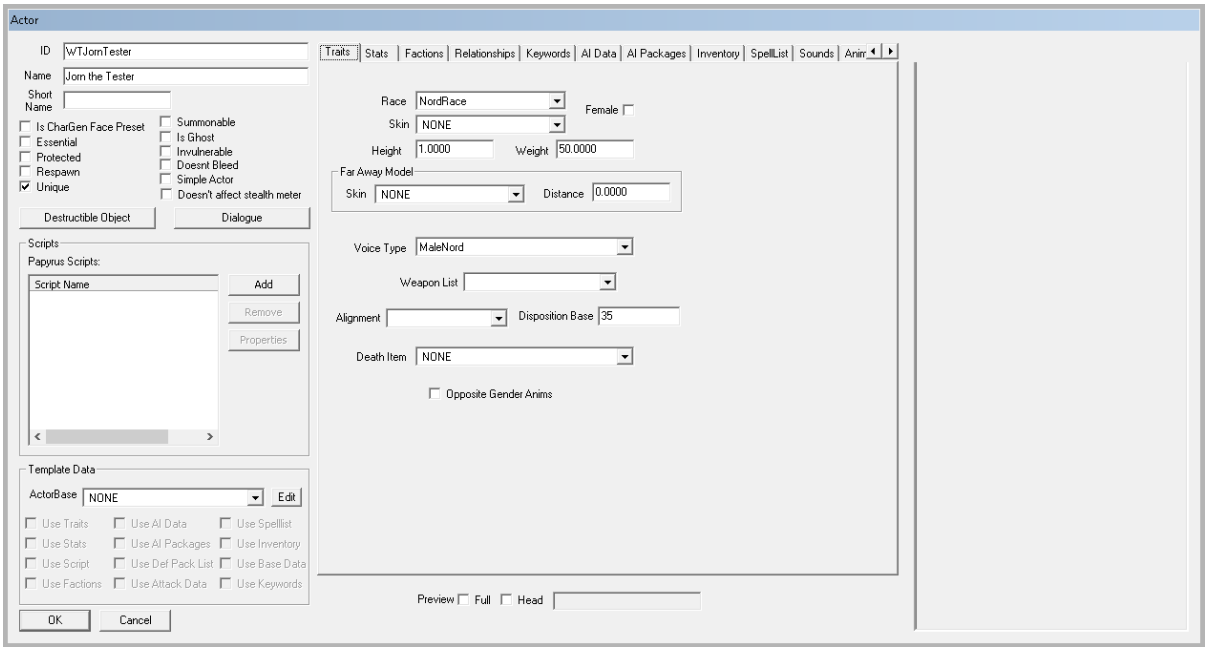


Figure 427 - Traits tab.

Go to the Stats tab.

By default the NPC's level will be set to 1. You can change this value or set the NPC's level to scale with the player's level by ticking the 'PC Level Mult' tickbox.

With PC Level Mult ticked, the Level field will become the Level Mult field; a multiplier used to calculate the NPC's level. Calc Min will set the NPC's minimum level and Calc Max will cap the NPC's maximum possible level.

For this example, I set Calc Min to 5 and Calc Max to 10, leaving the Level Mult set to 1.00.

Changing an NPC's level will affect their skills and their Health/Stamina/Magicka. You can give an NPC extra health, stamina or magicka by adding a value to the Health Offset, Magicka Offset or Stamina Offset fields.

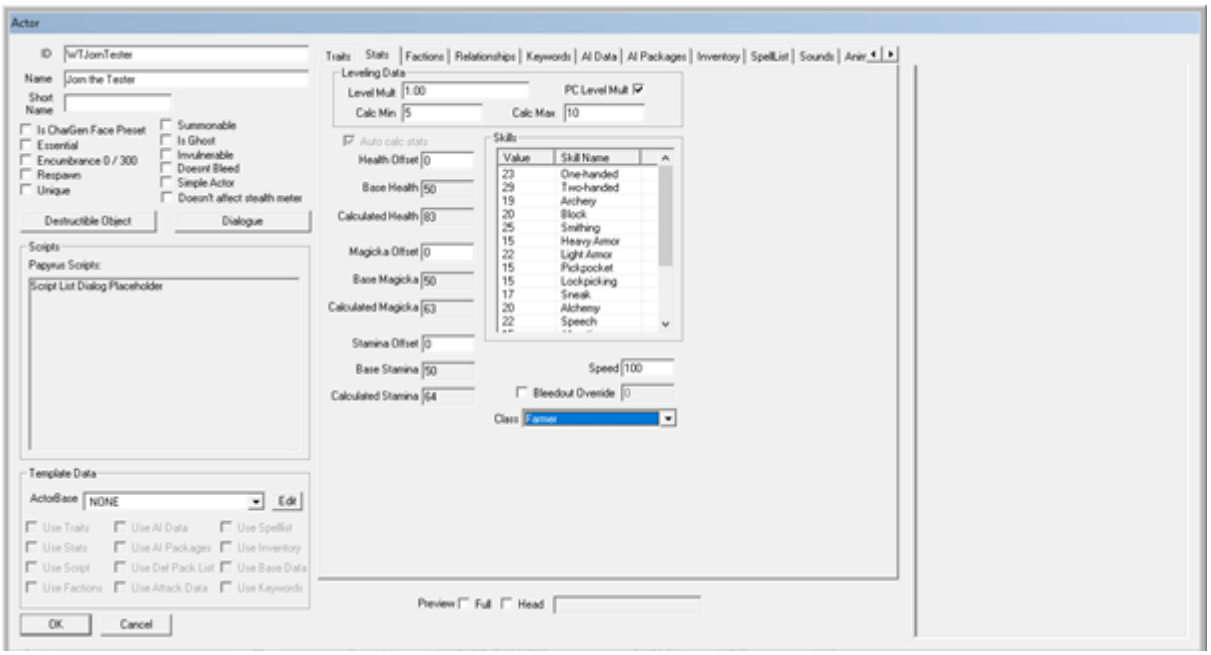


Figure 428 - Stats tab.

The Class drop-down affects the NPC's skill point distribution. For example, CombatSorceror will gain more points in magic skills per level than the CombatBarbarian class. I'm going to set Jorn's class to Farmer for this example.

Next let's look at the Faction tab.

This lists the factions that this NPC belongs to. Factions serve several purposes. They define which NPCs will aggro onto an aggressor if a member of a faction is attacked within a detectable range of another NPC of the same or allied faction.

They are also used to determine the dialogue available to an NPC. For example, merchants are members of the JobMerchantFaction faction, miners are members of the JobMinersFaction faction and followers are members of the PotentialFollower faction.

Factions are also used to link a merchant with their inventory.

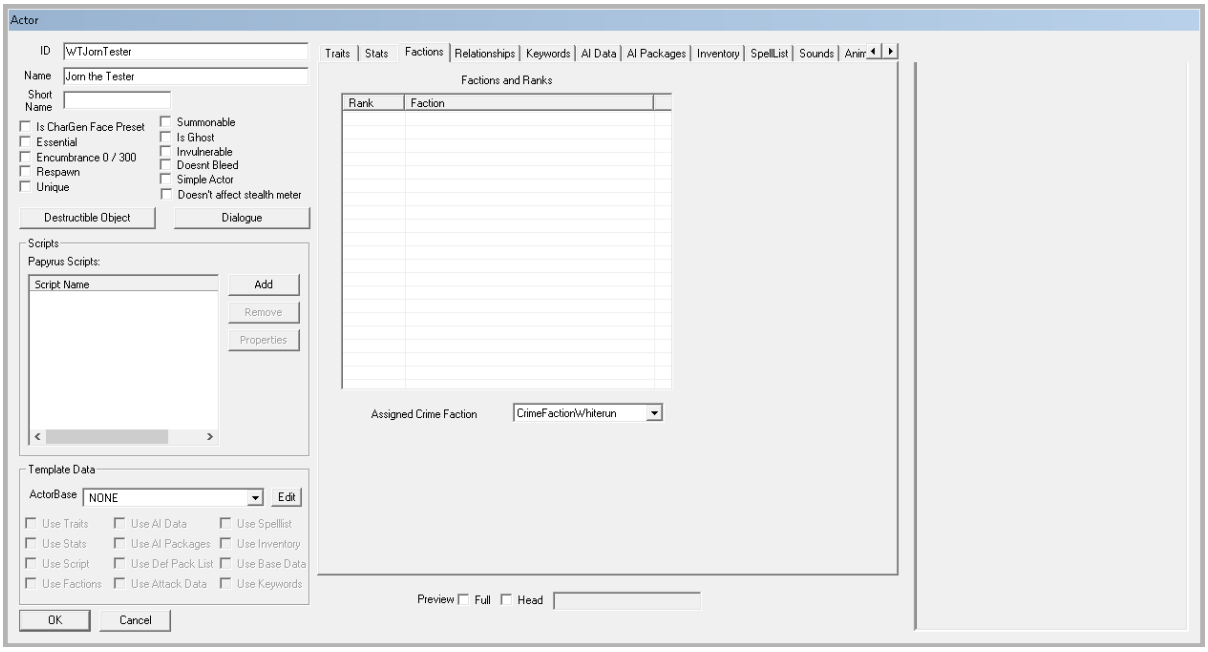


Figure 429 - Faction tab.

For this example, I'm going to add Jorn to the CrimeFactionWhiterun and TownRiverwoodFaction factions.

To add a new faction, right-click in the factions list and select New.

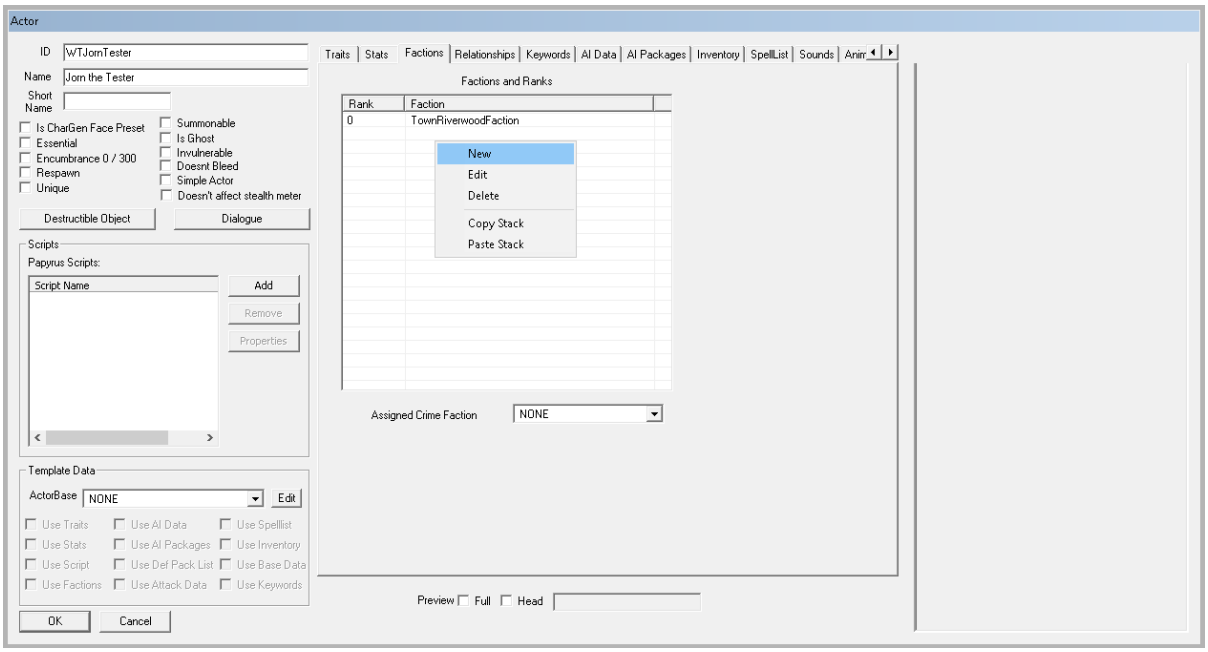


Figure 430 - Adding a faction to an NPC.

Filter by 'crimefactionwhiterun', click on CrimeFactionWhiterun in the form list and click OK.

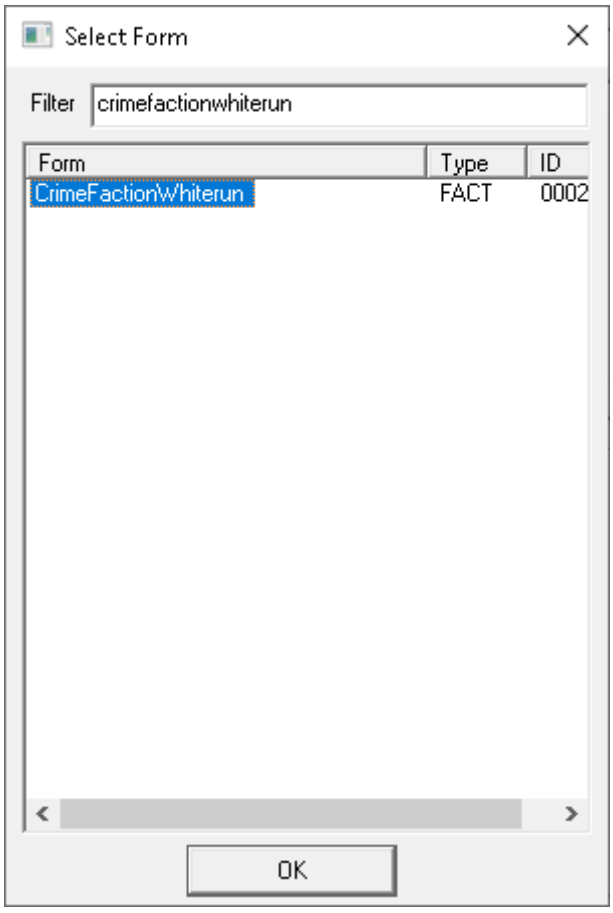


Figure 431 - Faction list.

Repeat these steps to add Jorn to the TownRiverwoodFaction as well.

Set the Assigned Crime Faction drop-down to CrimeFactionWhiterun. This will determine which hold will issue a bounty on the player if the player is caught committing a crime against this NPC.

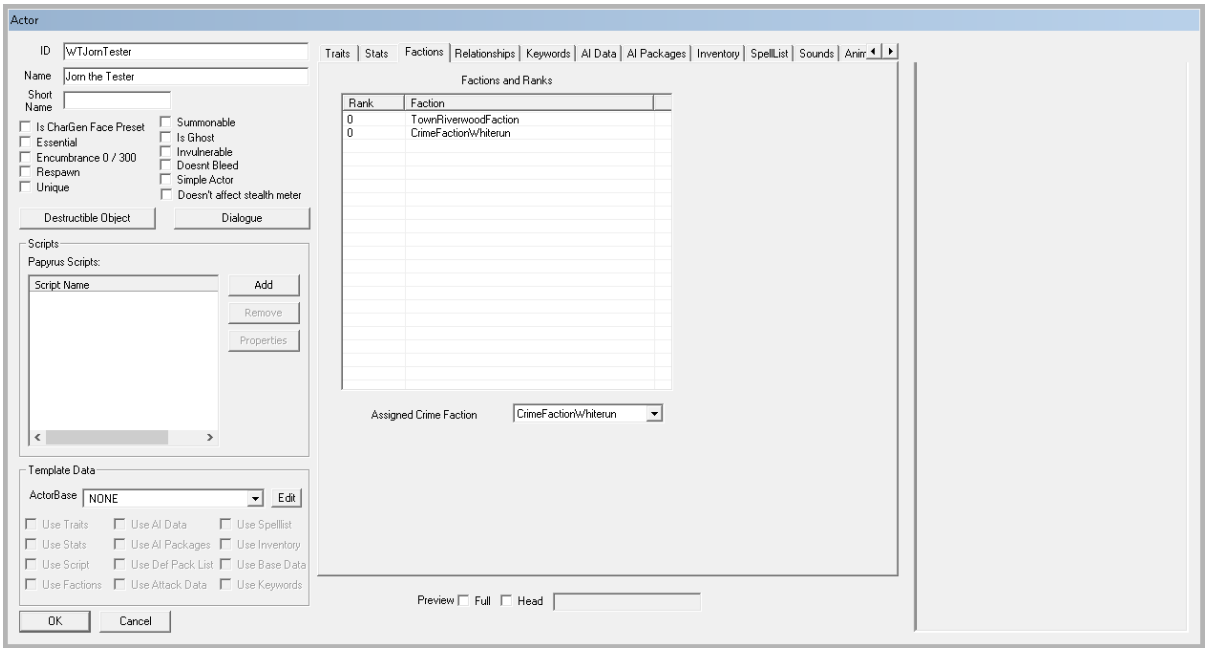


Figure 432 - Factions tab with crime faction set up.

Because Jorn is a member of the TownRiverwoodFaction, if he sees you attacking other members of this faction he may become hostile depending on how his AI will be set up in the AI Data tab later.

The Relationships tab allows you to set what kinds of relationships this NPC has to other NPCs. This affects the kind of dialogue available to the NPC. For example, an NPC may comment favourably about the player assaulting a nemesis, or may lament the passing of a loved one.

To add a new relationship, right-click in the relationships list and select New.

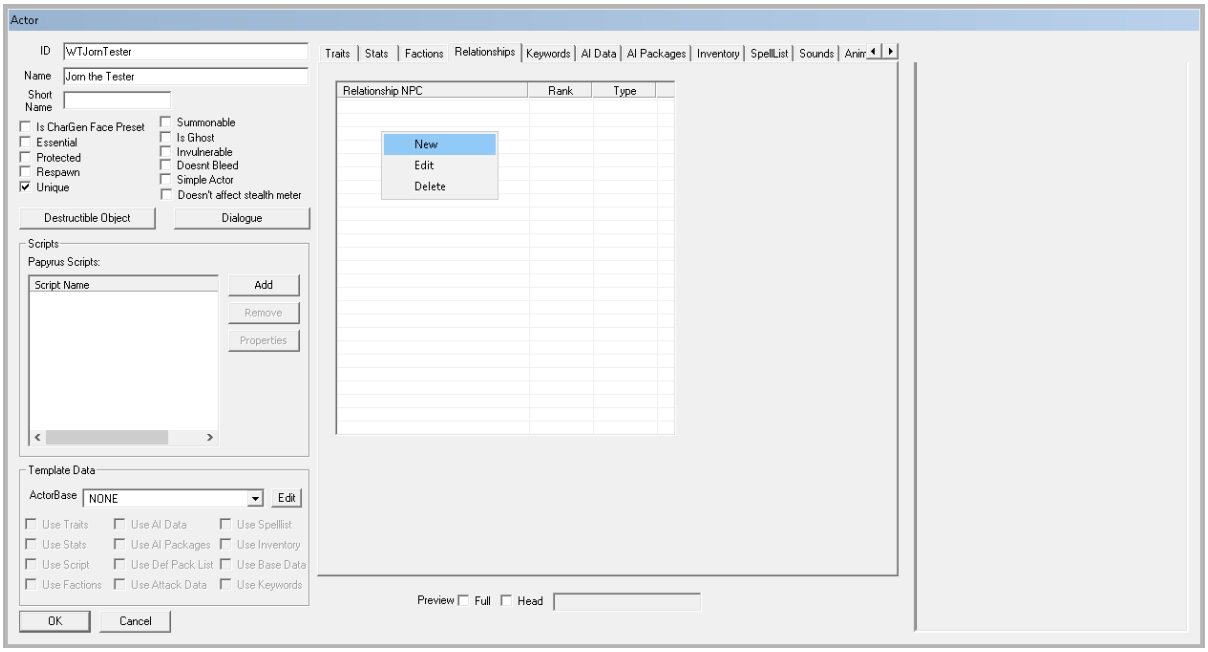


Figure 433 - Adding a new relationship.

Set the Relationship ID. This should just be the names of the two NPCs involved.

The Parent NPC drop-down should automatically be set as Jorn.

For this example, I'm going to set the Child NPC drop-down to Delphine and I'm going to set the Relationship Level to Foe.

Jorn is going to have an open disdain for Delphine and should anything happen to her, he'll likely be pleased about it.

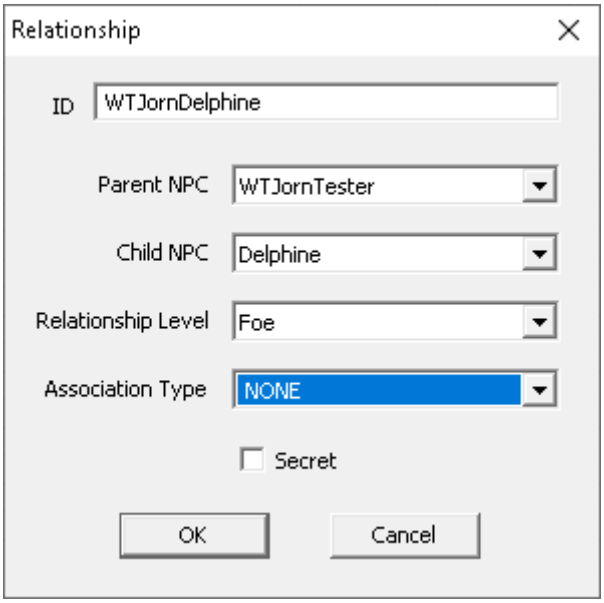


Figure 434 - Relationship properties.

Click OK to close out of Relationship properties.

Jorn's relationship with Delphine should now appear in the relationships list.

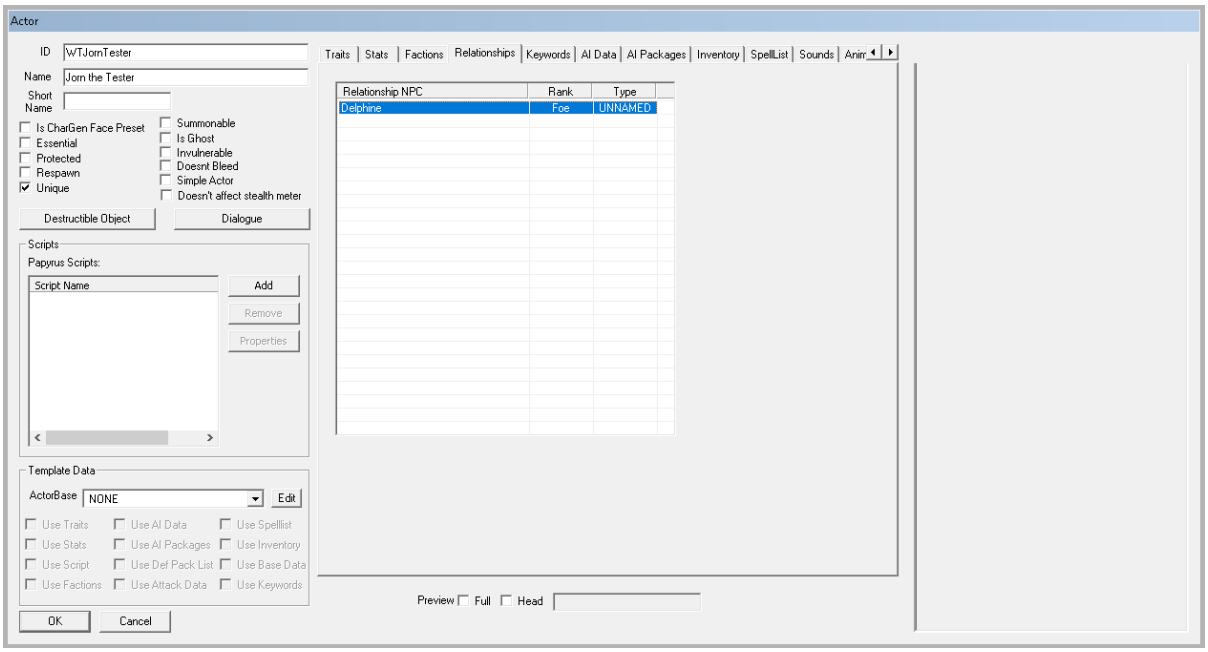


Figure 435 - Relationships tab with a relationship set up.

The Keywords tab is used to assign keywords to an NPC, which act as additional flags.

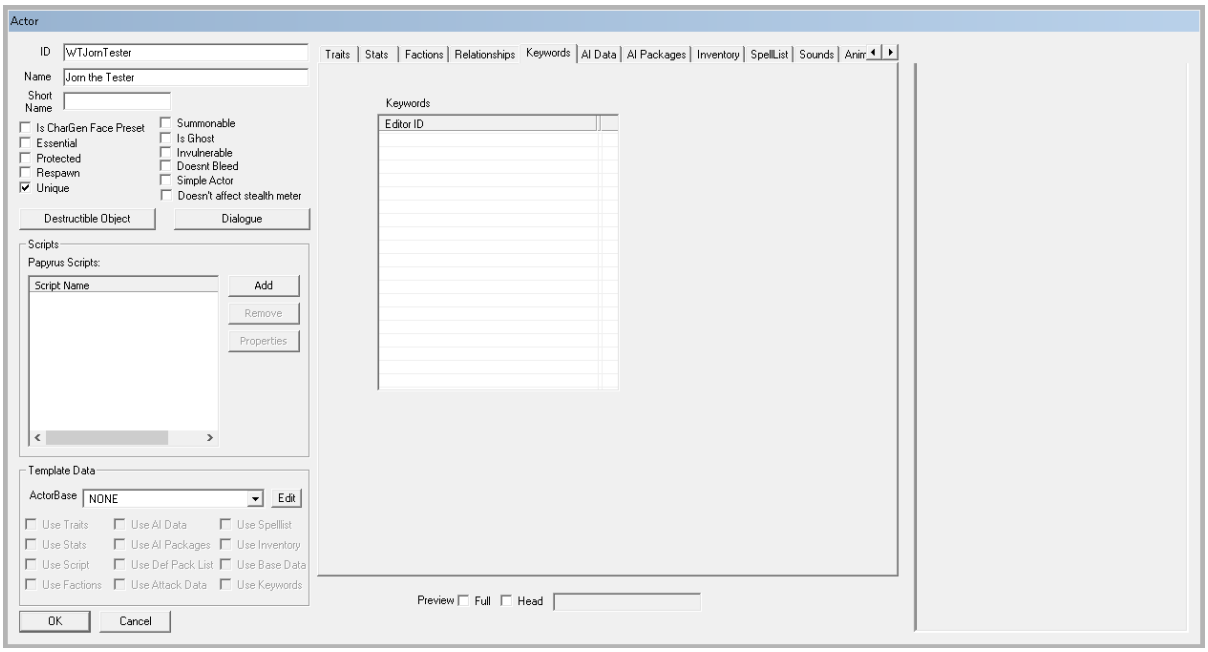


Figure 436 - Keywords tab.

For this example I'm not going to give Jorn any keywords. But if you wanted to add a keyword, right-click in the keywords list and select Add.

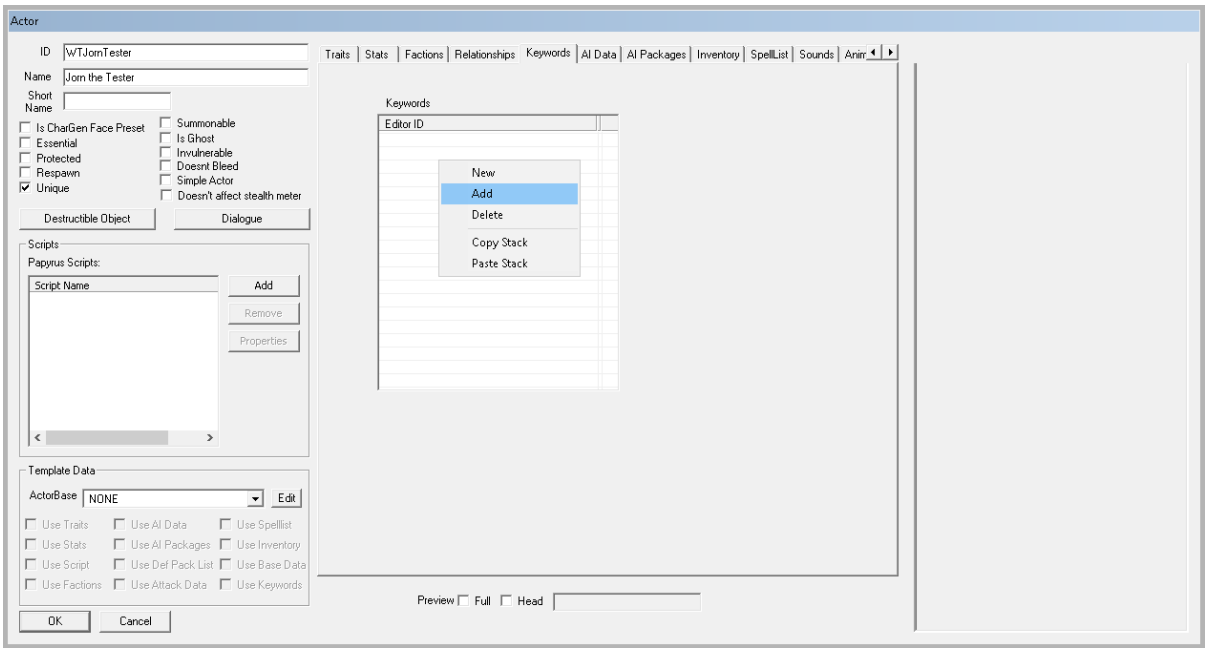


Figure 437 - Adding a new keyword.

Highlight the keyword you want to add then click OK.

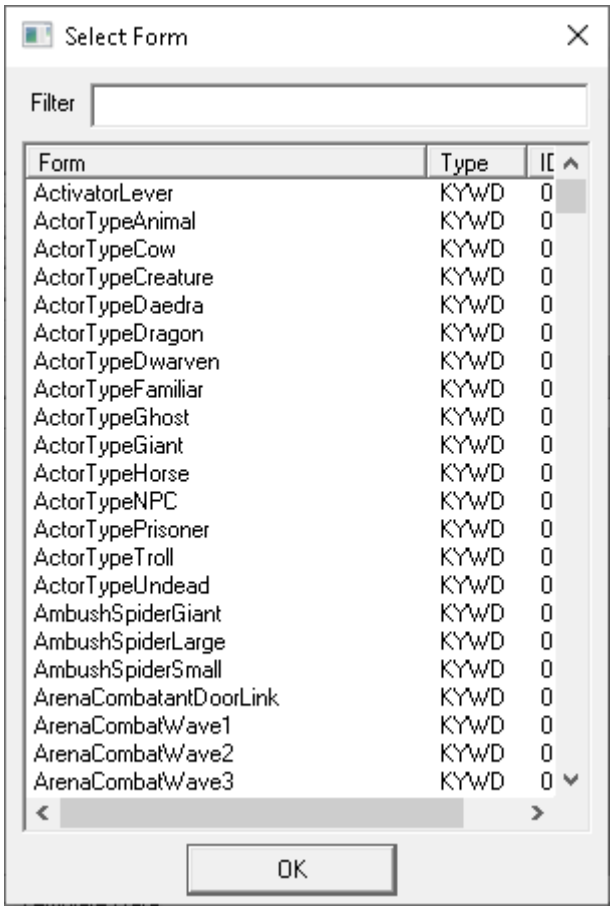


Figure 438 - Keywords list.

The AI Data tab allows you to set how this NPC will behave in combat.

The Aggression drop-down sets how this NPC reacts to the player or other NPCs that are in different factions.

Unaggressive NPCs will avoid engaging in combat unless it's to defend themselves. Aggressive NPCs will attack enemies as soon as they're detected. Very Aggressive NPCs will attack enemies as well as other neutral NPCs as soon as they're detected. Frenzied NPCs will attack anyone on sight, even members of their own faction.

The Confidence drop-down affects the likelihood that this NPC will flee during combat.

Cowardly NPCs will flee from combat as soon as they detect an aggressor. Cautious NPCs will flee if the aggressor is stronger than they are. Average NPCs will flee if they are outmatched in combat. Brave NPCs will flee from combat only if they are losing. Foolhardy NPCs will never flee from combat.

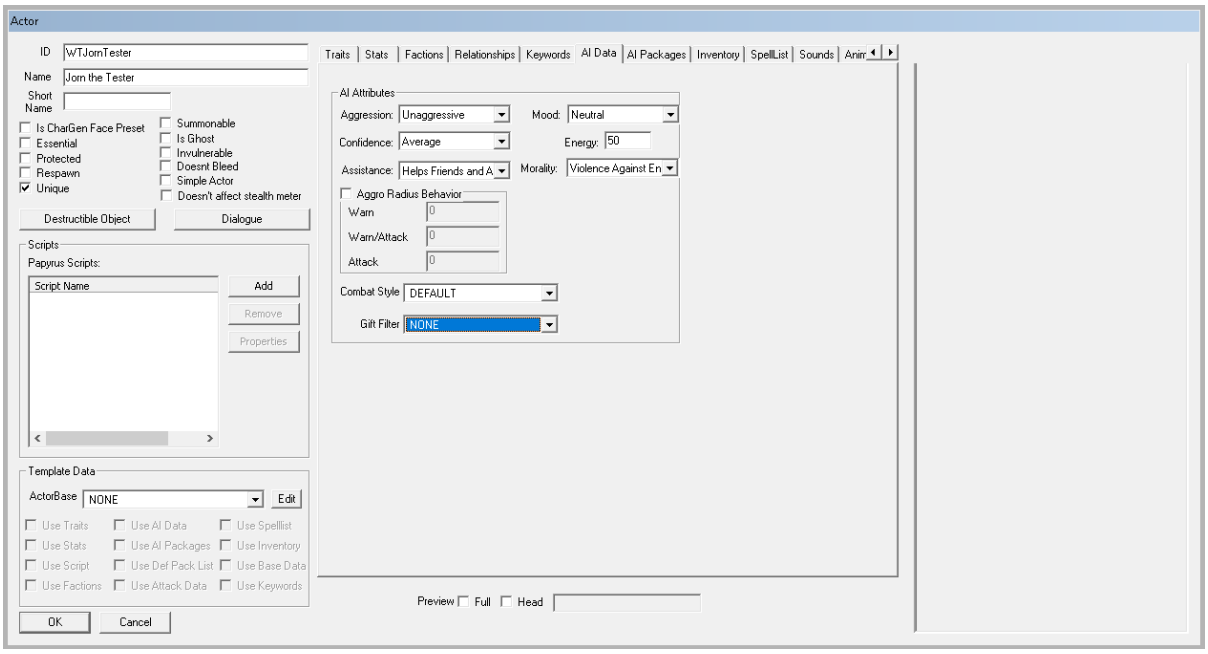


Figure 439 - AI Data tab.

The Assistance drop-down determines whether the NPC will engage in combat if they detect that a member of their faction or an allied faction is under attack.

Helps Nobody means this NPC won't join combat to help out an ally. Helps Allies means this NPC will join combat to help out members of their faction. Helps Friends and Allies means this NPC will join combat to help out members of their faction or members of allied factions.

The Morality drop-down determines which crimes this NPC will commit when directed to commit a crime by the player.

No Crime means this NPC will never commit a crime when directed to. Property Crime Only means this NPC will only steal. Violence Against Enemies means this NPC will assault other NPCs when directed to. Any Crime means this NPC will do anything that the player orders them to.

The Gift Filter and Mood drop-downs aren't used as far as I'm aware.

Combat Style is used to determine how often the NPC uses certain actions in combat, such as blocking or casting spells. You should choose a combat style that matches the NPC's primary weapon. For example, if the NPC is wielding a two-handed weapon, you should use something like csHumanBerserkerLvl1. For spellcasters, use something like csHumanMagic. Combat Styles can be found by browsing to Miscellaneous > Combat Style in the Object Window.

Lastly, the Aggro Radius Behaviour tickbox changes this NPC's behaviour based on distance to the player or to other NPCs.

If someone enters the NPC's Warn radius, the NPC will stop what they're doing and deliver a 'stay back' warning.

If someone enters the NPC's Warn/Attack radius, a warning will be given. If the player or NPC doesn't leave this radius in a timely manner, the NPC will become hostile.

Finally, anyone entering the Attack radius will be attacked by the NPC immediately.

We'll be Adding and Creating AI Packages shortly, but for now let's go to the Inventory tab.

Set the Default Outfit to FarmClothesOutfit01. This is what Jorn will be wearing when he spawns in. Outfits are collections of clothes, boots, helmets, and so on. Outfits can be found in the Object Window under Items > Outfit.

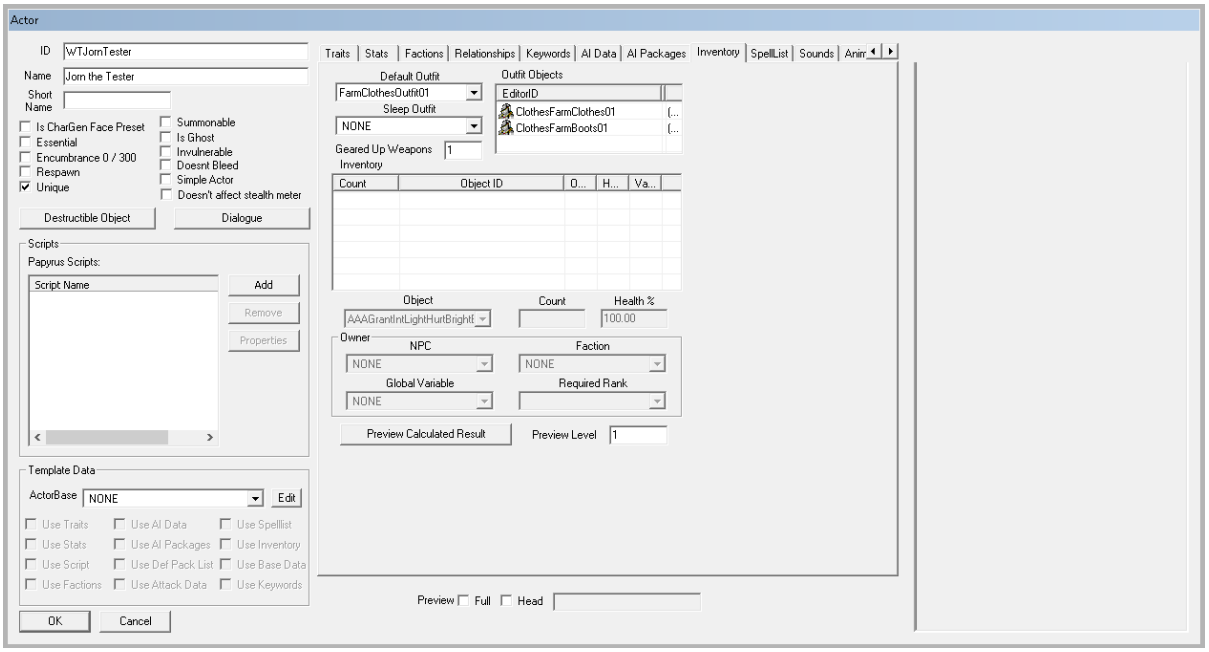


Figure 440 - Adding a default outfit.

Let's give him something to defend himself with. Right-click in the inventory list and select New.

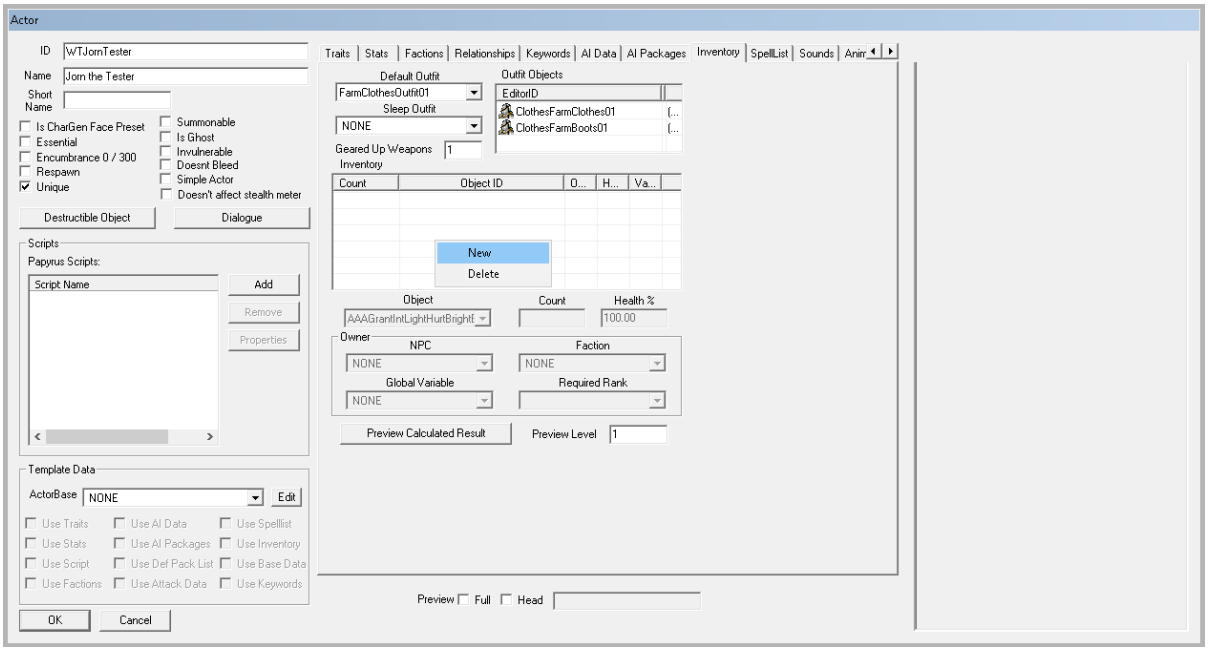


Figure 441 - Adding a new inventory item.

By default this will add AAAGrantIntLightHurtBrightBounce into the NPC's inventory. We need to change this to what we want the NPC to actually have.

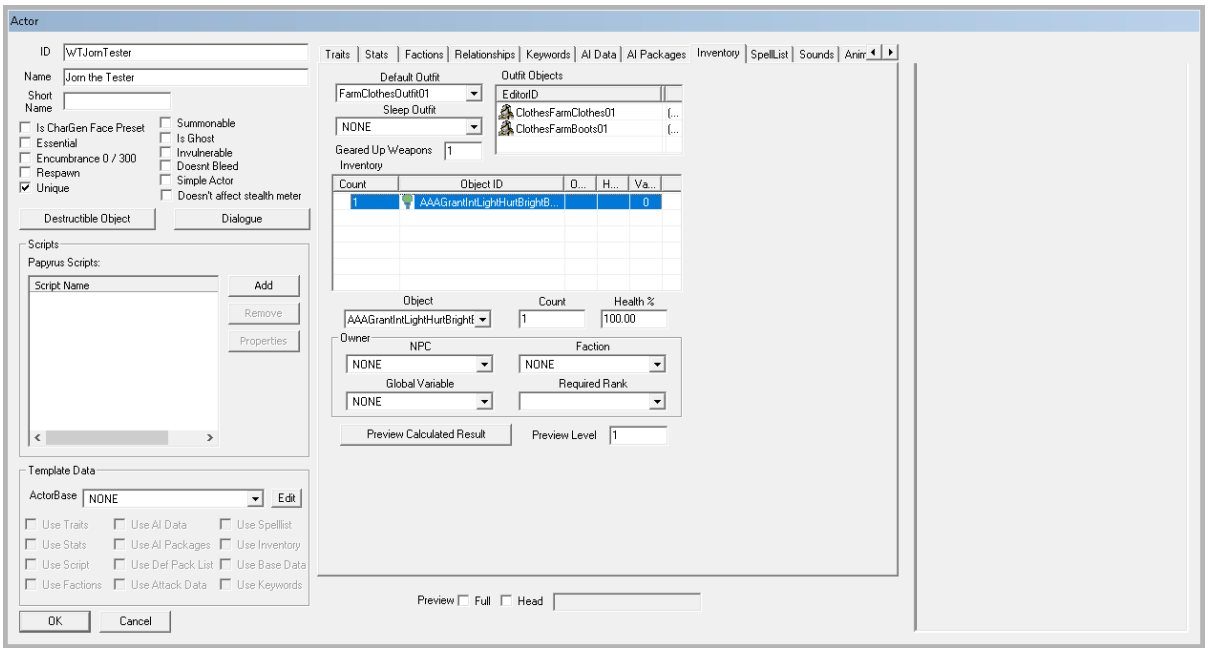


Figure 442 - New inventory item added.

Click on it to select it then set the Object drop-down to IronDagger.

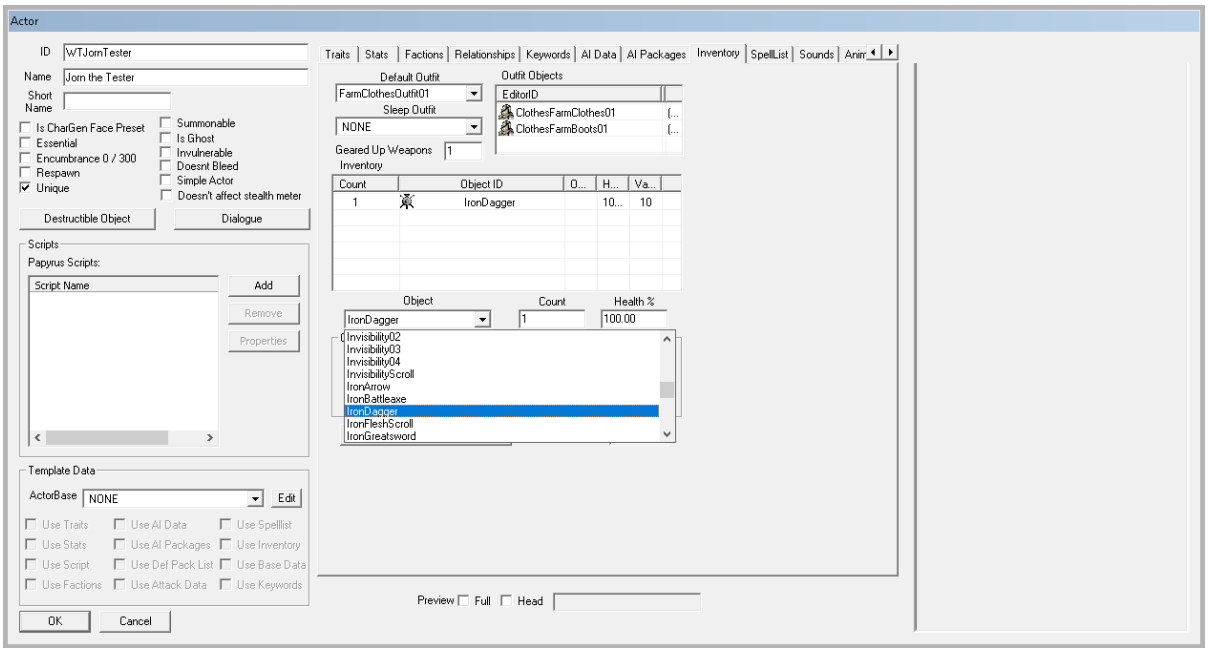


Figure 443 - Changing the first inventory item to IronDagger.

Let's also give him some random inventory items. Add another inventory item and set it to LootCitizenPocketsPoor. This will add a random item to Jorn's inventory when he spawns in. It could be a bit of gold, a low value ring, or some lock picks. Levelled items can be found in the Object Window under Items > LevelledItem.

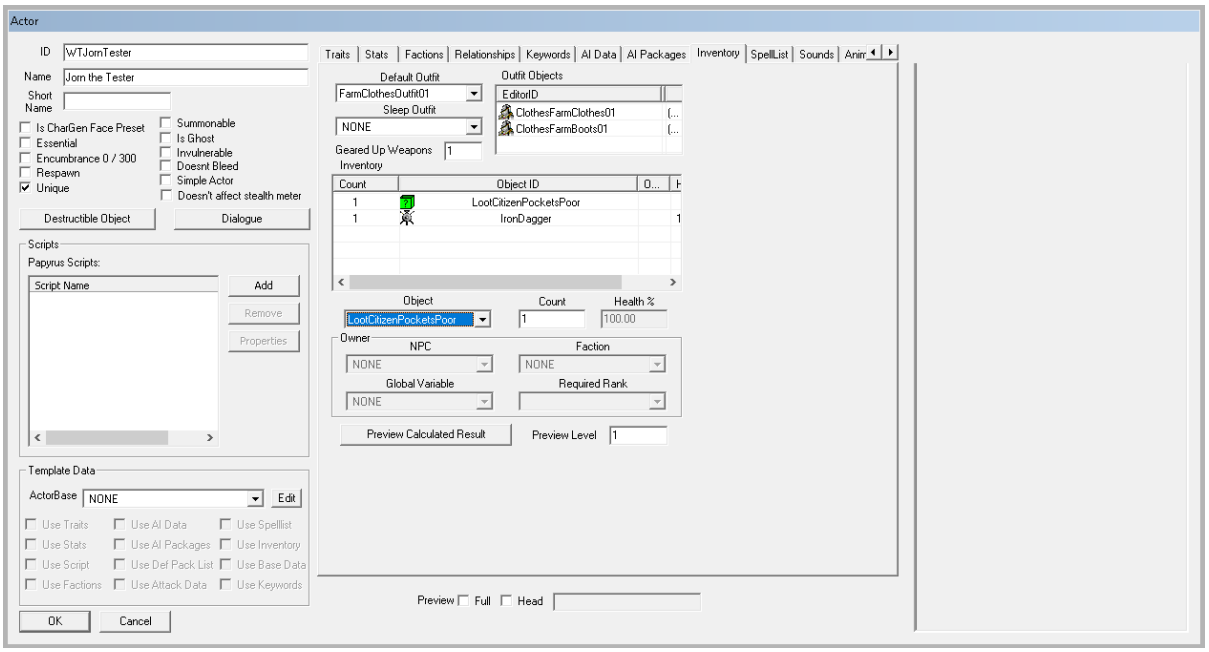


Figure 444 - Added a LootCitizenPocketsPoor levelled item.

Spells and perks can be assigned to an NPC under the SpellList tab.

To add a spell, right-click in the spell list and select New.

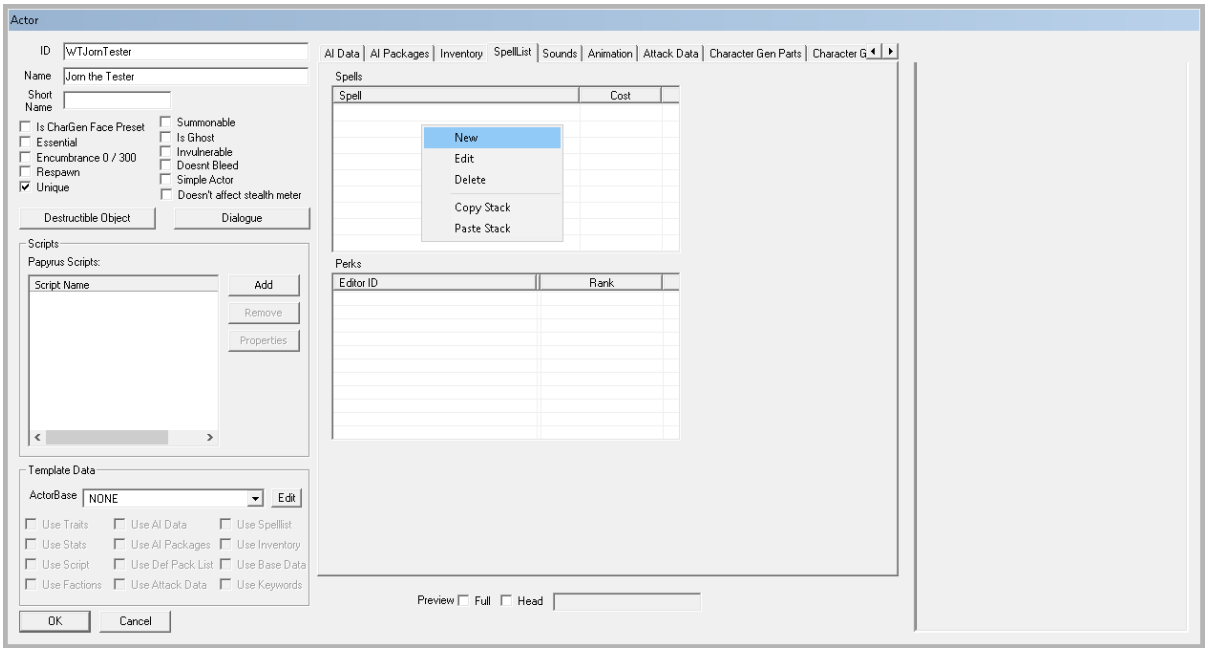


Figure 445 - Adding a spell.

Filter by 'healing', select HealingLeftHand and click OK.

You'll notice that this is different to the regular Healing spell in that the NPC will only use this spell in their left hand, allowing them to simultaneously attack with their right hand weapon.

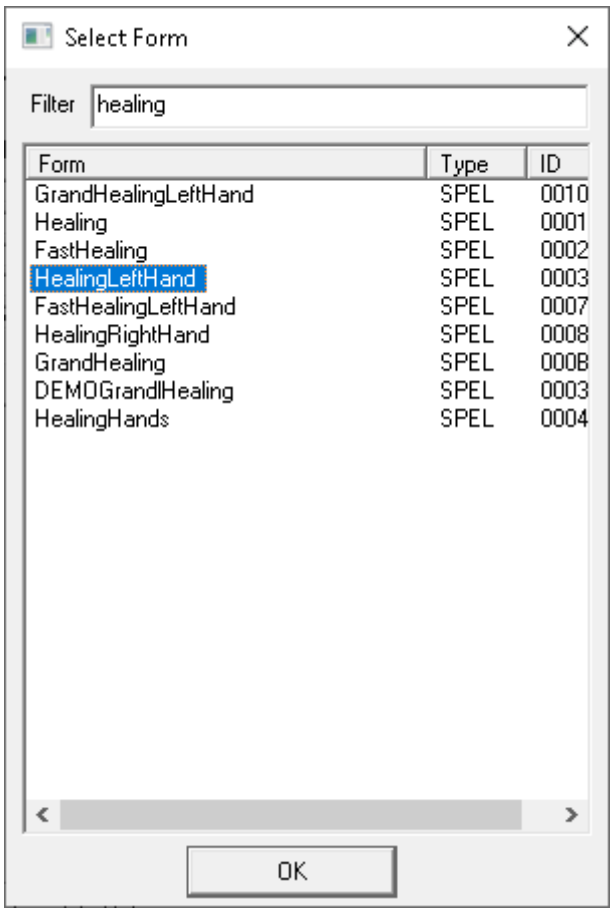


Figure 446 - Adding a left hand healing spell.

We can make this spell cost less magicka by giving Jorn the Novice Restoration Perk. To do this, right-click in the perks list and select Add.

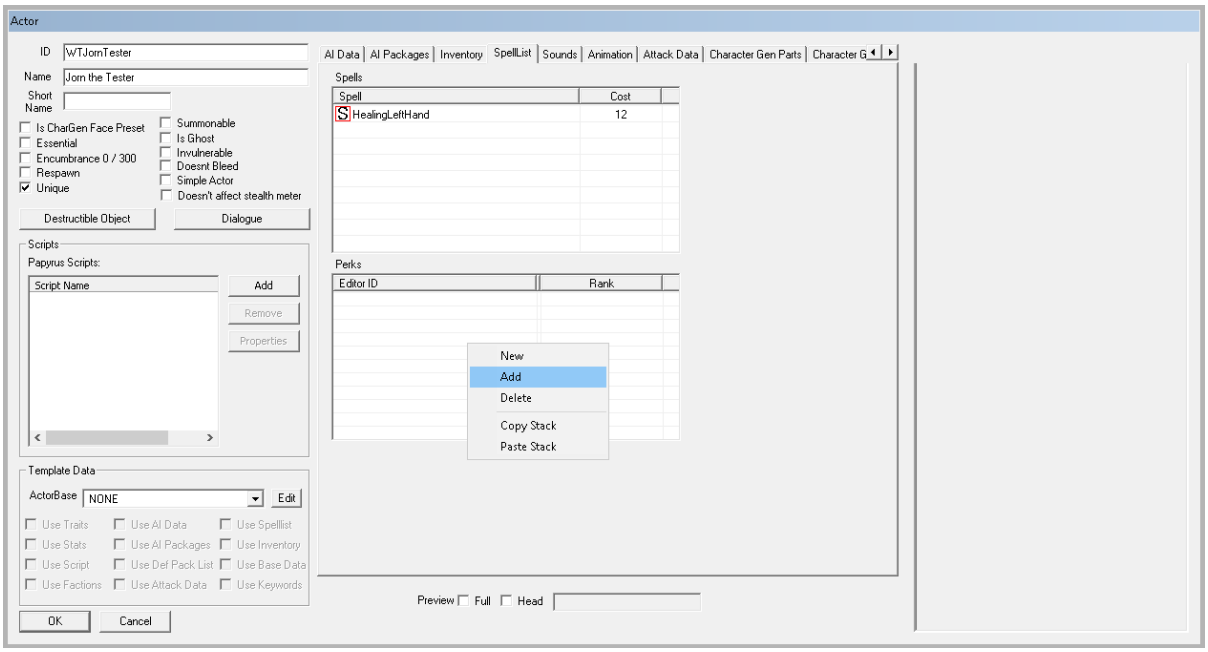


Figure 447 - Adding a perk.

Filter by 'restoration', click on RestorationNovice00 to highlight it then click OK.

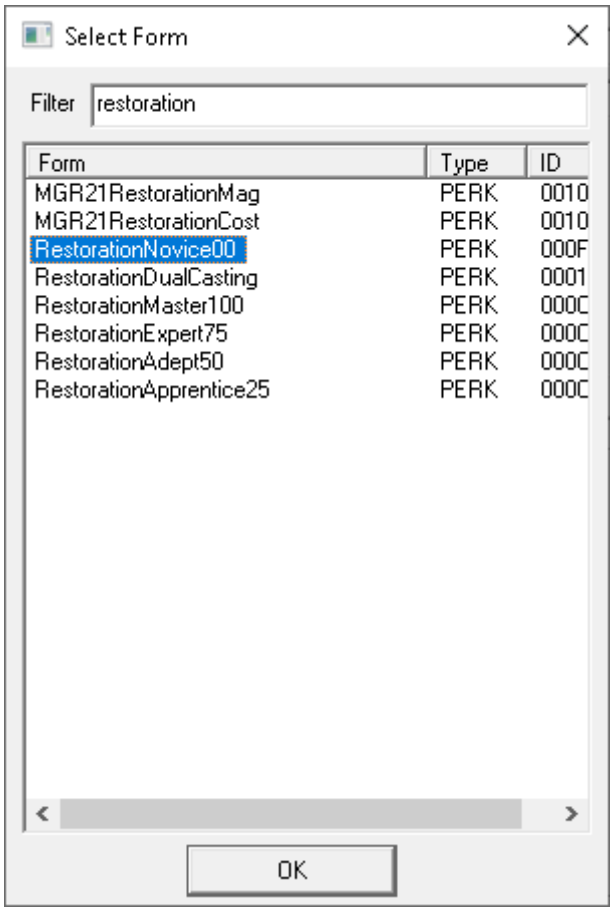


Figure 448 - Adding the Novice Restoration perk.

We can skip over the Sounds, Animation and Attack Data tabs and go to the Character Gen Parts tab next.

Here you can customize the NPC's head colours, such as hair colour, skin colour, face paint, as well as hair style.

To see the NPC's face in the preview area, tick Head at the bottom.

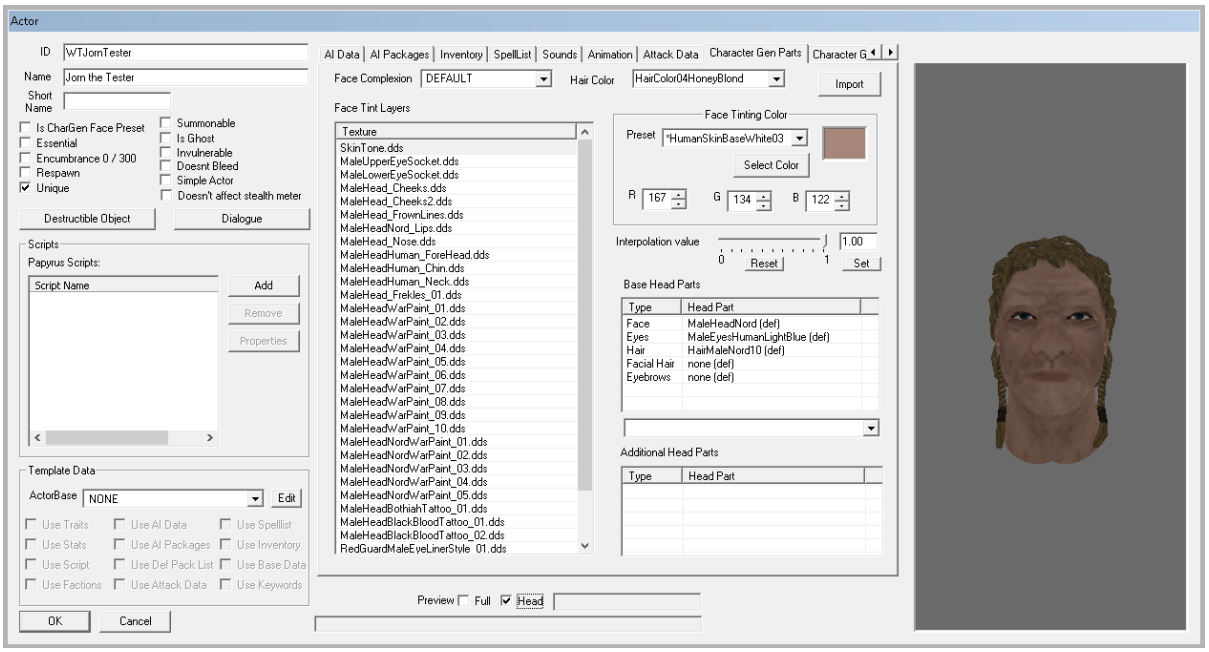


Figure 449 - Character Gen Parts tab.

Let's start customizing Jorn by changing his hair style.

In the Basic Head Parts section, click on Hair.

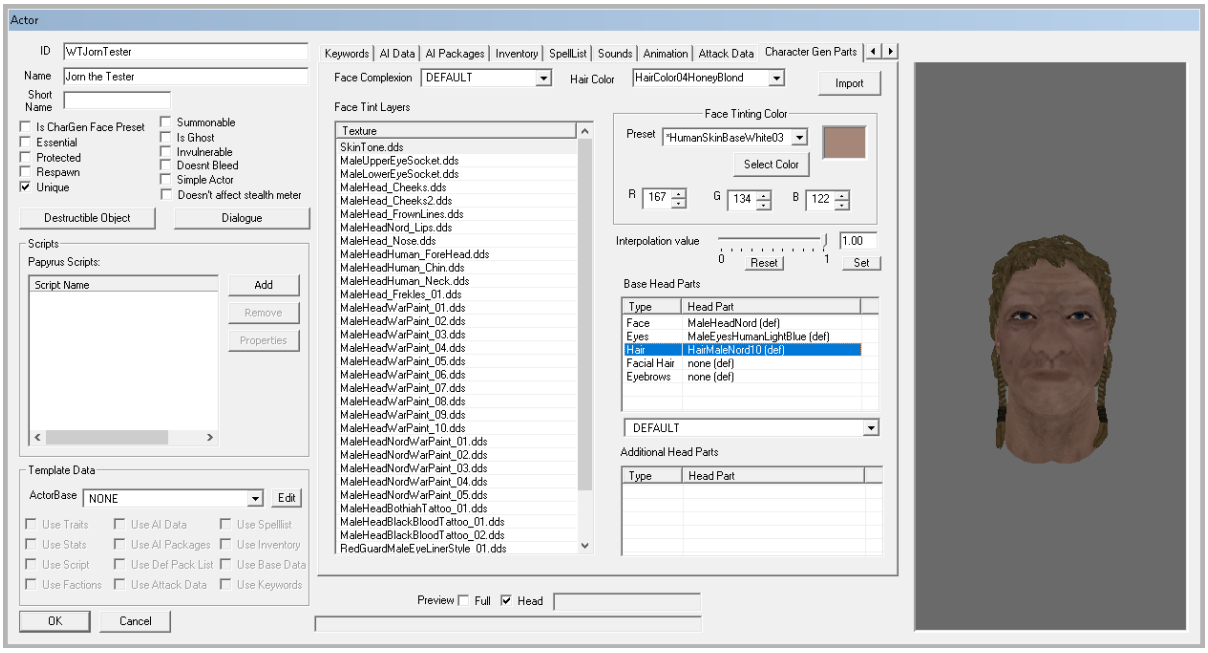


Figure 450 - Changing an NPC's hair style.

I'm going to set it to HairMaleNord14.

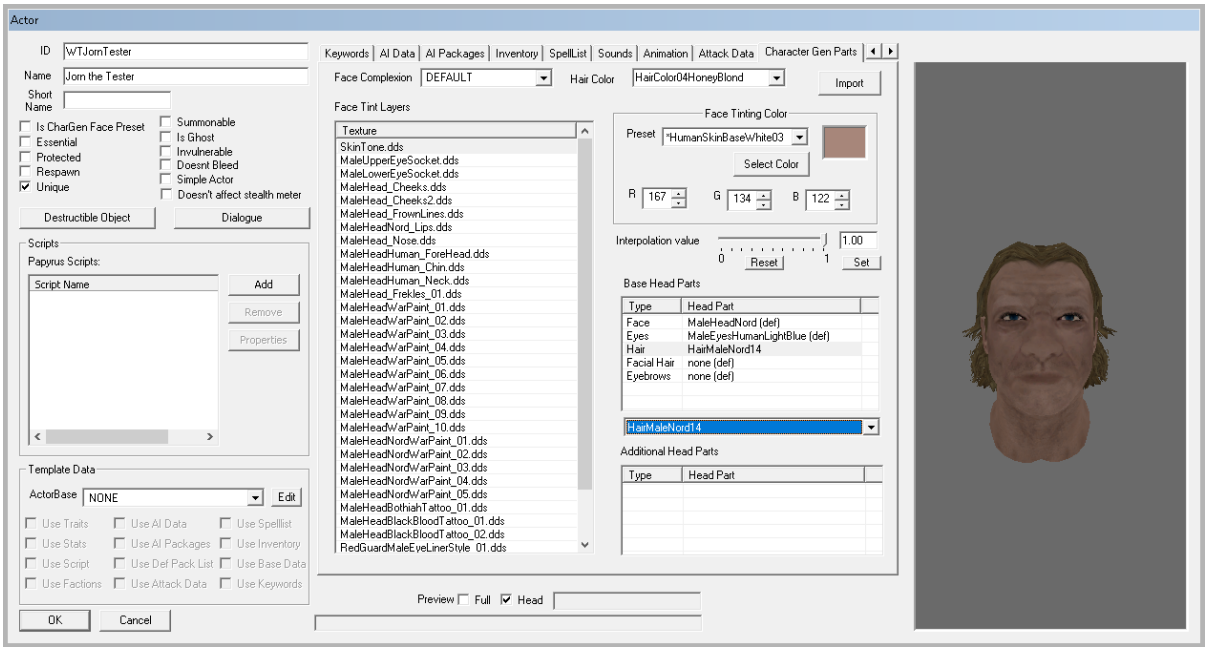


Figure 451 - Hair style changed.

To change an NPC's hair colour, change the Hair Colour drop-down near the top. In this example, I changed it to HairColour08MediumBrown.

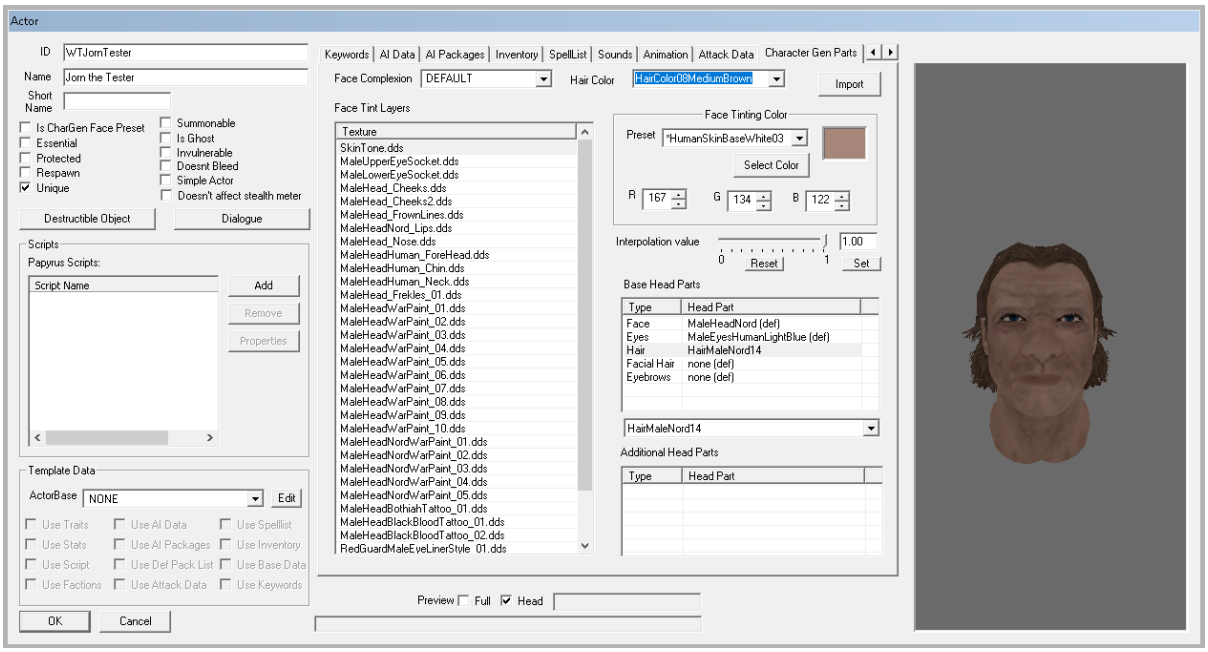


Figure 452 - Changing an NPC's hair colour.

Next, let's give him some eyebrows. In the Basic Head Parts section I set Eyebrows to BrowsMaleHumanoid02.

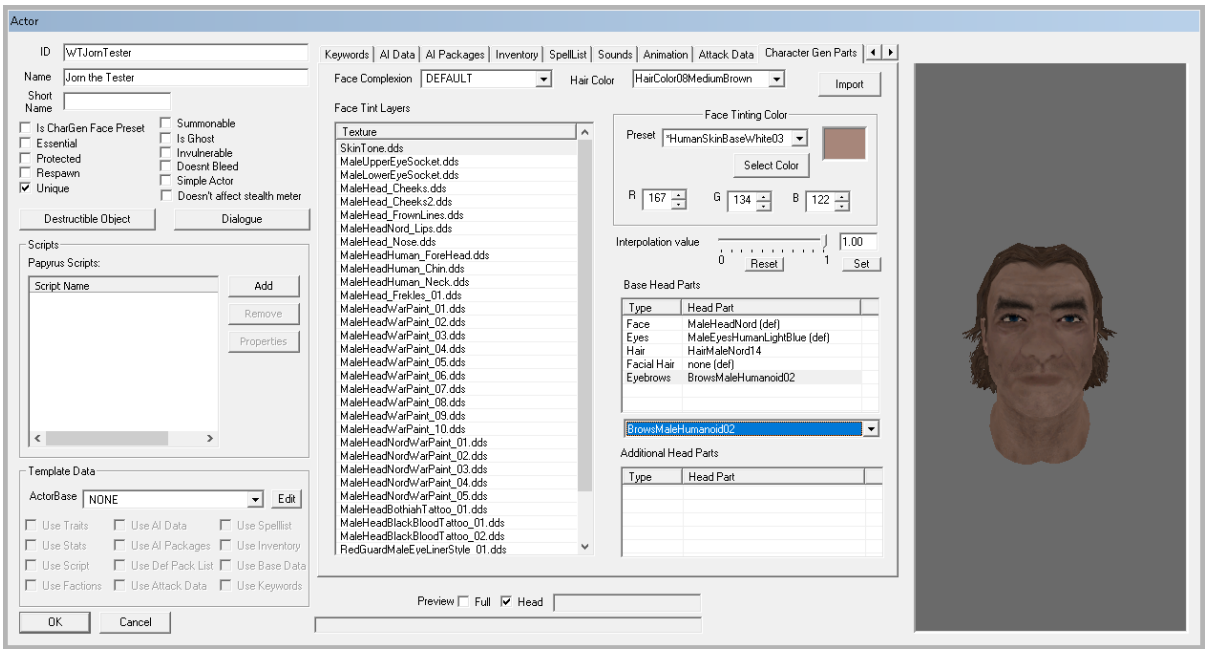


Figure 453 - Changing an NPC's eyebrows.

Give him some facial hair by setting Facial Hair to HumanBeard08.

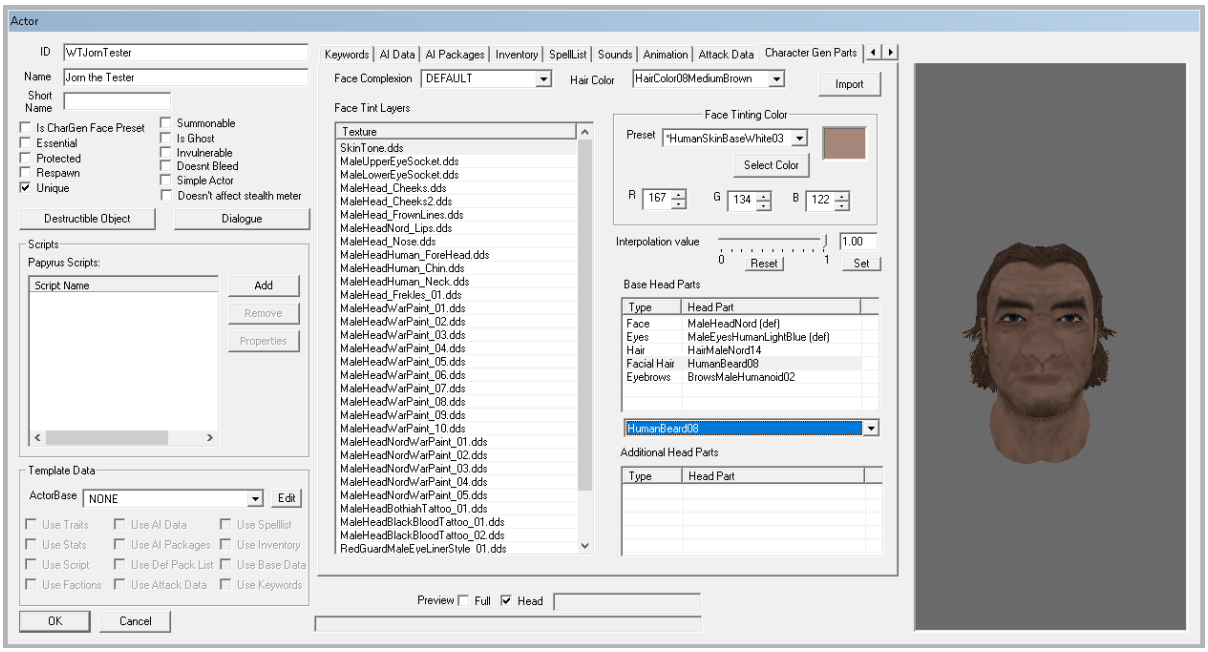


Figure 454 - Changing an NPC's facial hair.

And change his eye colour by setting Eyes to MaleEyesHumanHazel.

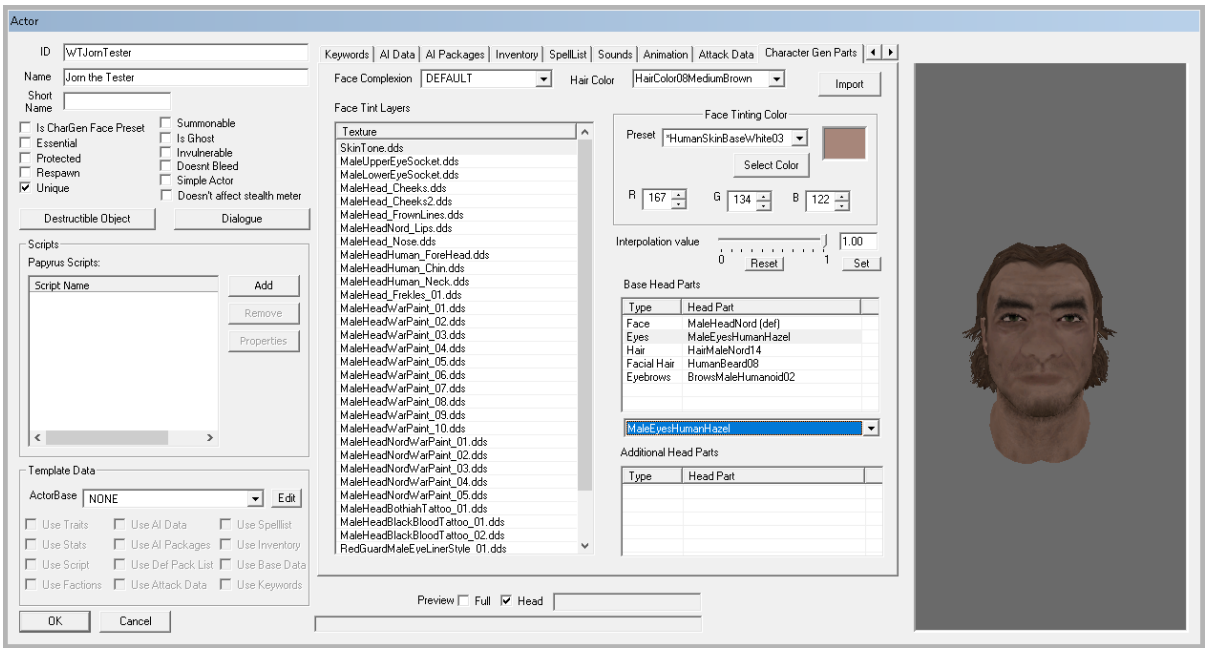


Figure 455 - Changing an NPC's eye colour.

Maybe Jorn hasn't been sleeping well lately, what with these dragon attacks. In the Face Tint Layers, click on MaleUpperEyeSocket.dds to highlight it, then under Face Tinting Colour set the Preset drop-down to BrowntTintDark. Change the Interpolation-value slider below it to about 0.40. 1.00 means the value is at full-strength, 0 would effectively hide the shading.

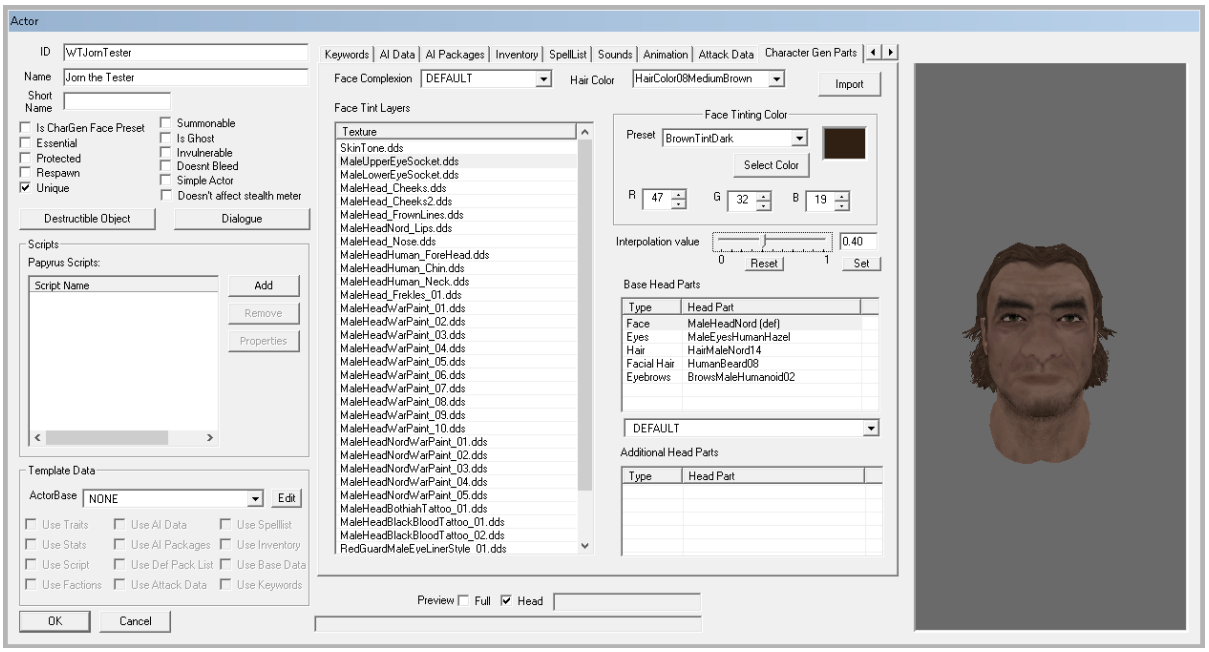


Figure 456 - Changing an NPC's face tint.

Perhaps Jorn doesn't bathe as often has he should. We can add dirt to his face by scrolling down to MaleHeadDirt_01.dds in Face Tint Layers, set the Preset to BrownTintDark then set the Interpolation Value slider to about 0.55.

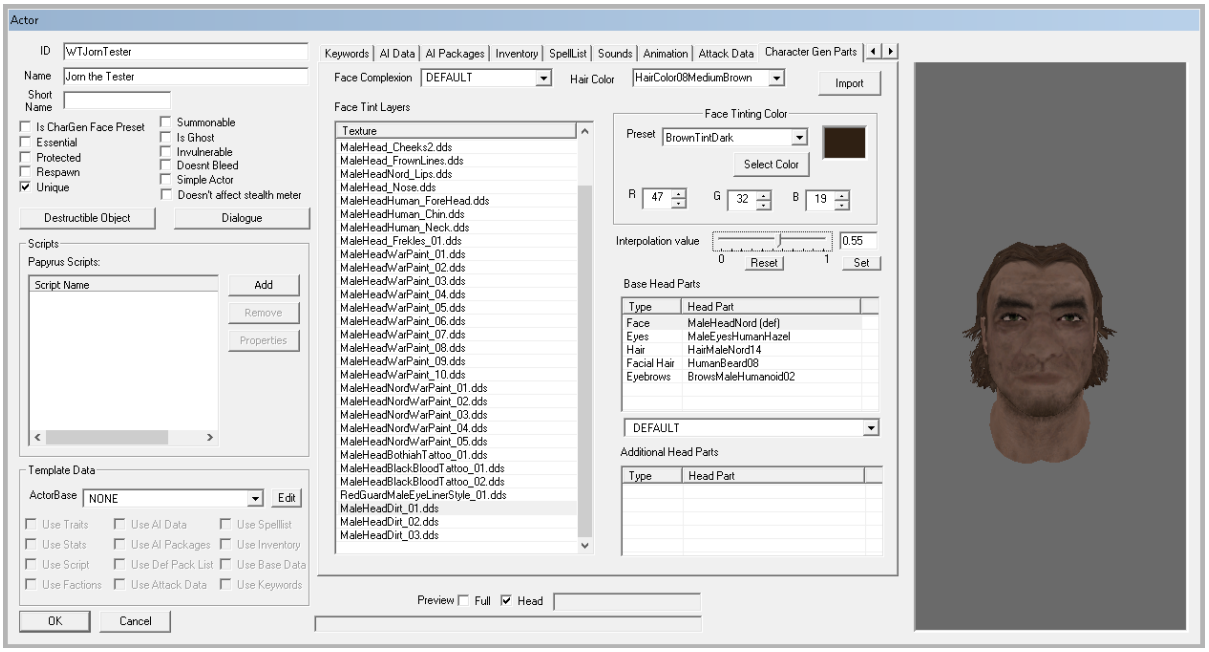


Figure 457 - Adding dirt to an NPC's face.

Note: You can choose a custom colour by clicking on the Select Colour button, but just be aware that colour selection can be a bit buggy so I generally just stick with the default presets.

Lastly, switch over to the Character Gen Morphs tab. Here we can adjust the shape of the NPC's facial features.

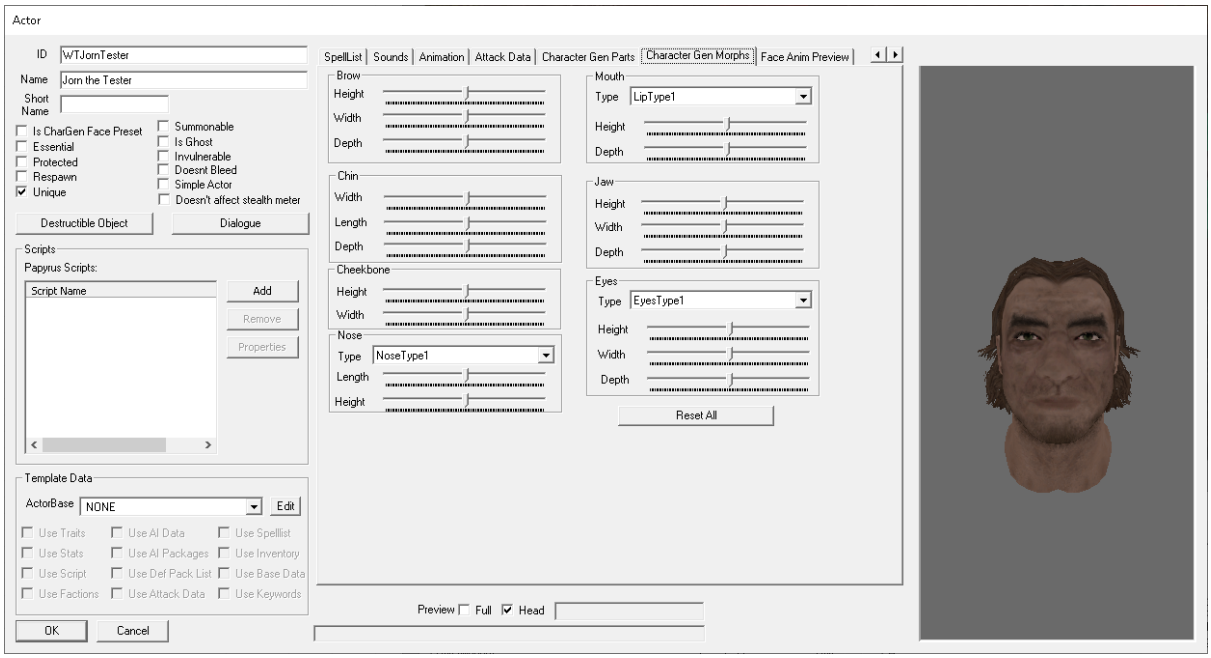


Figure 458 - Character Gen Morphs tab.

In the following screenshot, I made a few adjustments to the NPC's face. His nose was set to NoseType11, mouth to LipType11 and eyes to EyesType19.

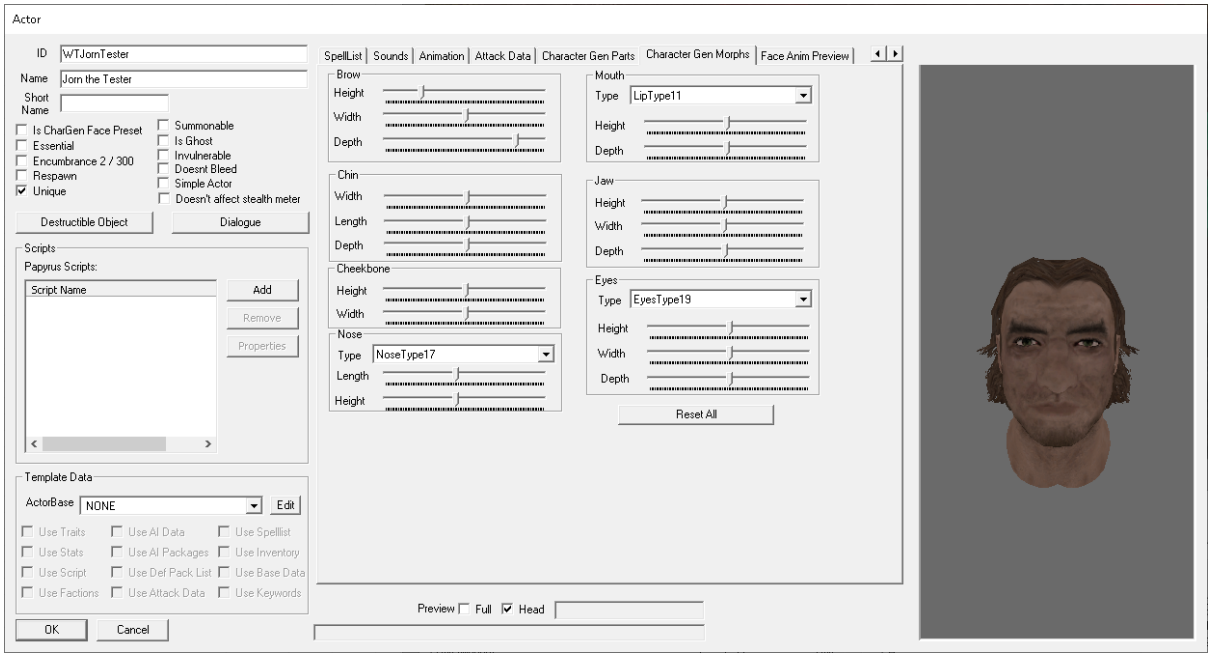


Figure 459 - NPC's face after some adjustments.

Now that our NPC is mostly done, we need to build facegen data.

To do this, click on WTJornTester in the Object Window to highlight him and press CTRL + F4.

Click Yes to the Confirm pop-up.

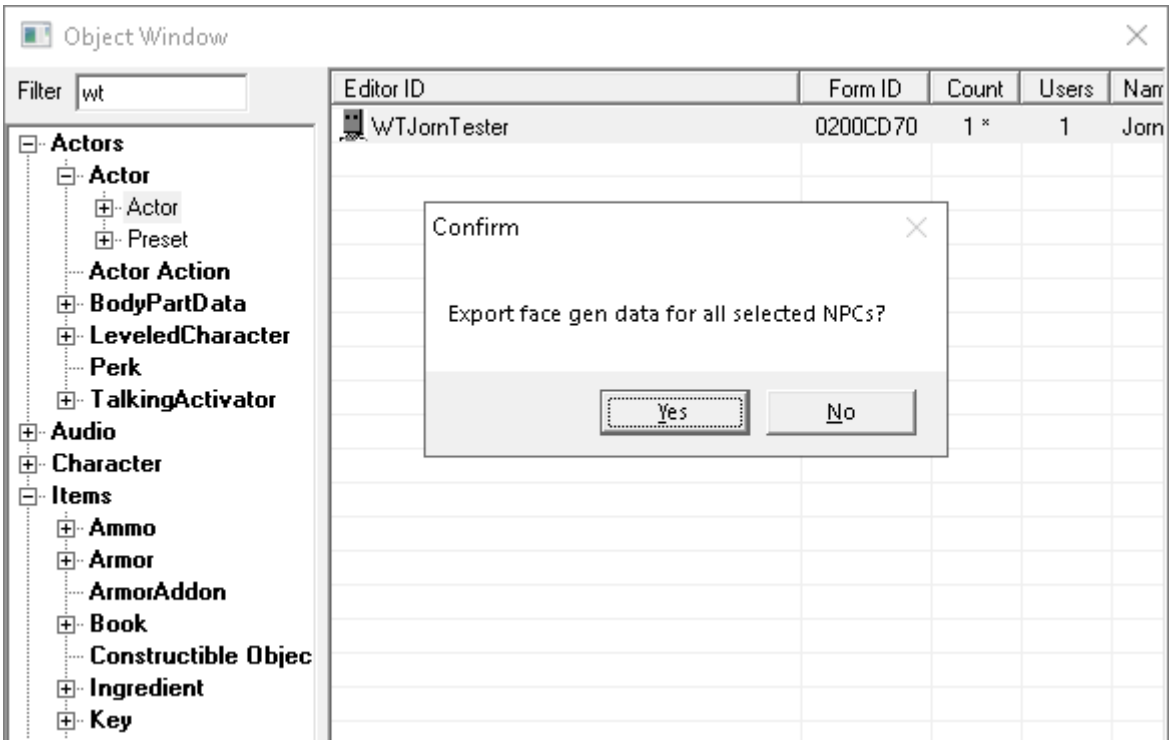


Figure 460 - Building facegen data.

Once it's finished, click OK to the Done pop-up.



Figure 461 - Done.

The mesh and texture generated can be found under Skyrim\Data\meshes\actors\Character\FaceGenData\FaceGeom\<YourModName>.esp or Skyrim Special Edition\Data\meshes\actors\Character\FaceGenData\FaceGeom\<YourModName>.esp and Skyrim\Data\Textures\actors\Character\FaceGenData\FaceTint\<YourModName>.esp or Skyrim Special Edition\Data\Textures\actors\Character\FaceGenData\FaceTint\<YourModName>.esp.

Before we place Jorn in the world, let's give him a script to clean up his body if he gets killed.

Double-click or right-click on WTJornTester and select Edit.

In the Scripts section, click on the Add button.

Important: Adding a script to an NPC may crash the Creation Kit. If you've added custom .bsa archives to SArchiveList or SResourceArchiveList in your SkyrimEditor.ini file, make sure you've extracted the .pex files from them to your Skyrim\Data\Scripts or Skyrim Special Edition\Data\Scripts folder.

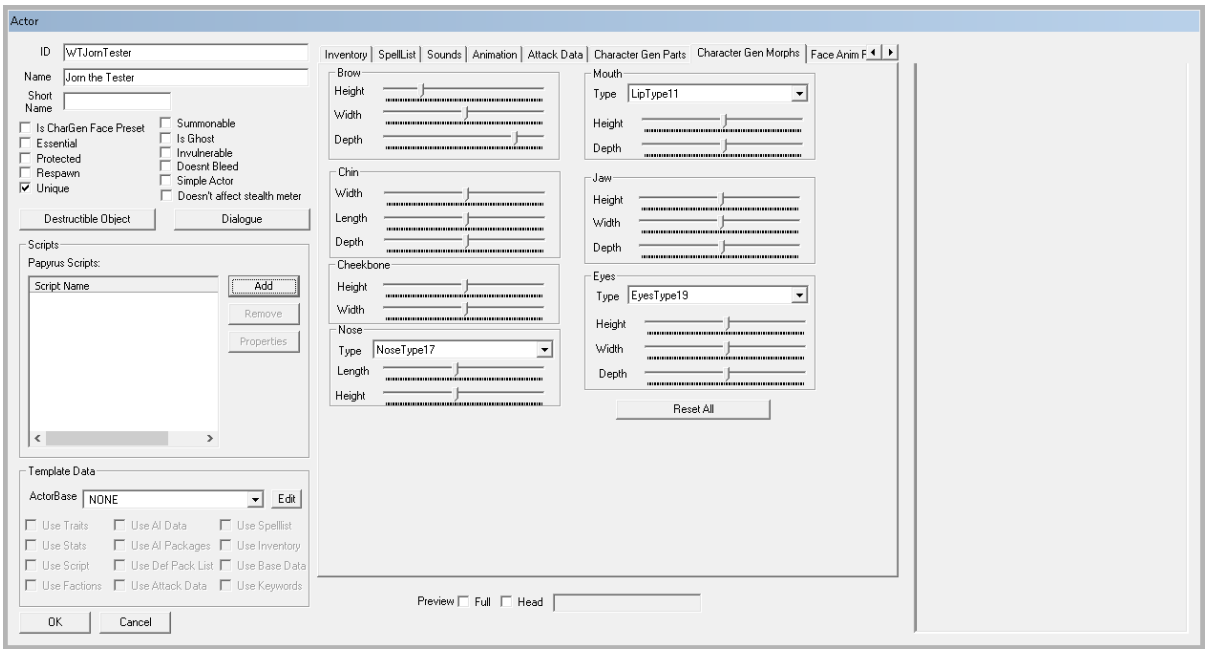


Figure 462 - Adding a script.

Filter by 'wideadbodycleanup', click on the WIDeadBodyCleanupScript script in the scripts list then click OK to add it to the NPC.

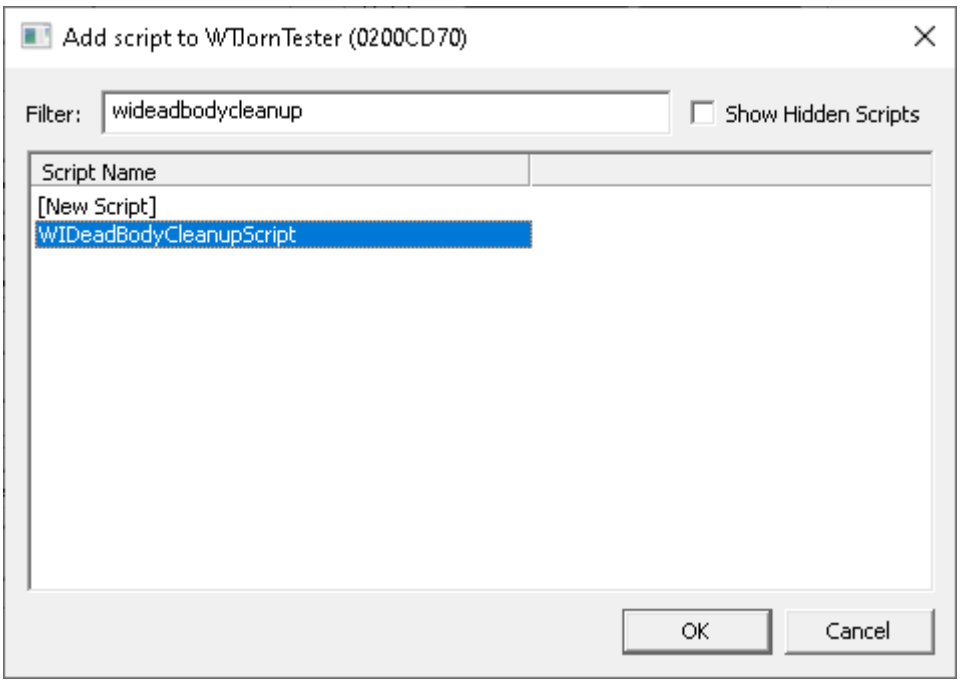


Figure 463 - Adding WIDeadBodyCleanupScript.

If Jorn had a grave, we can use this script to transfer his inventory to a coffin like some named NPCs have in major cities, but for now it's fine to just transfer his corpse to the dead body cell.

In the Papyrus Scripts list, click on WIDeadBodyCleanupScript to highlight it then click on the Properties button.

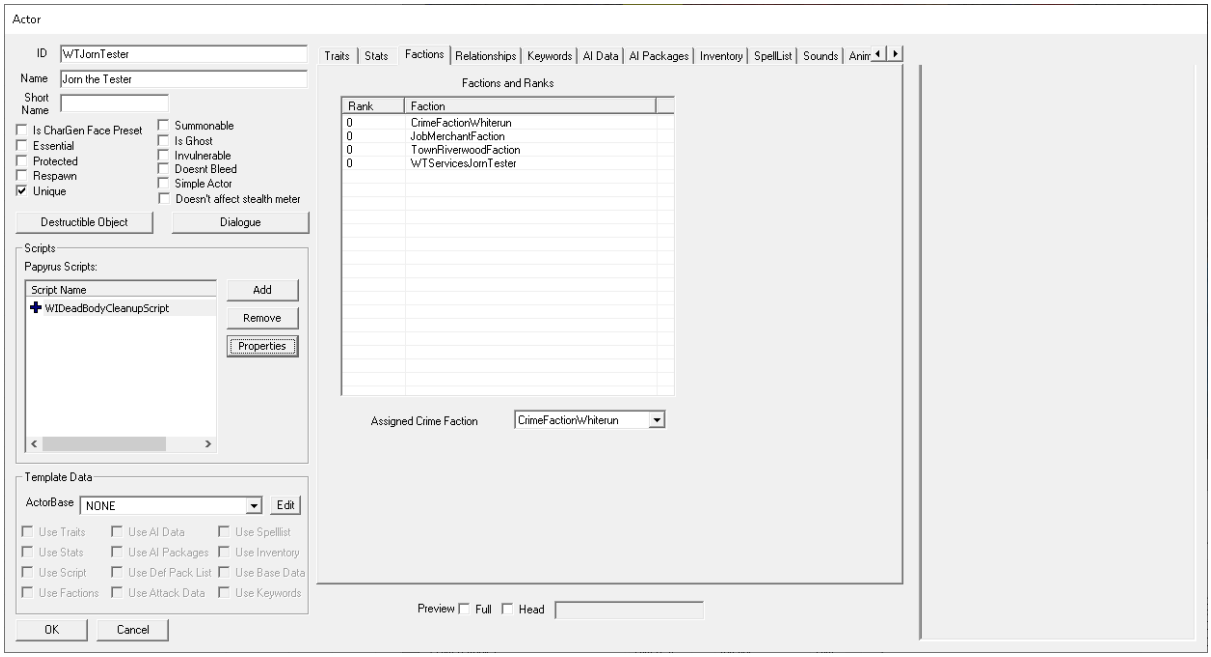


Figure 464 - Opening properties for WIDeadBodyCleanupScript.

Click on the Auto-Fill All button. Click OK to the Property Auto-Fill pop-up.

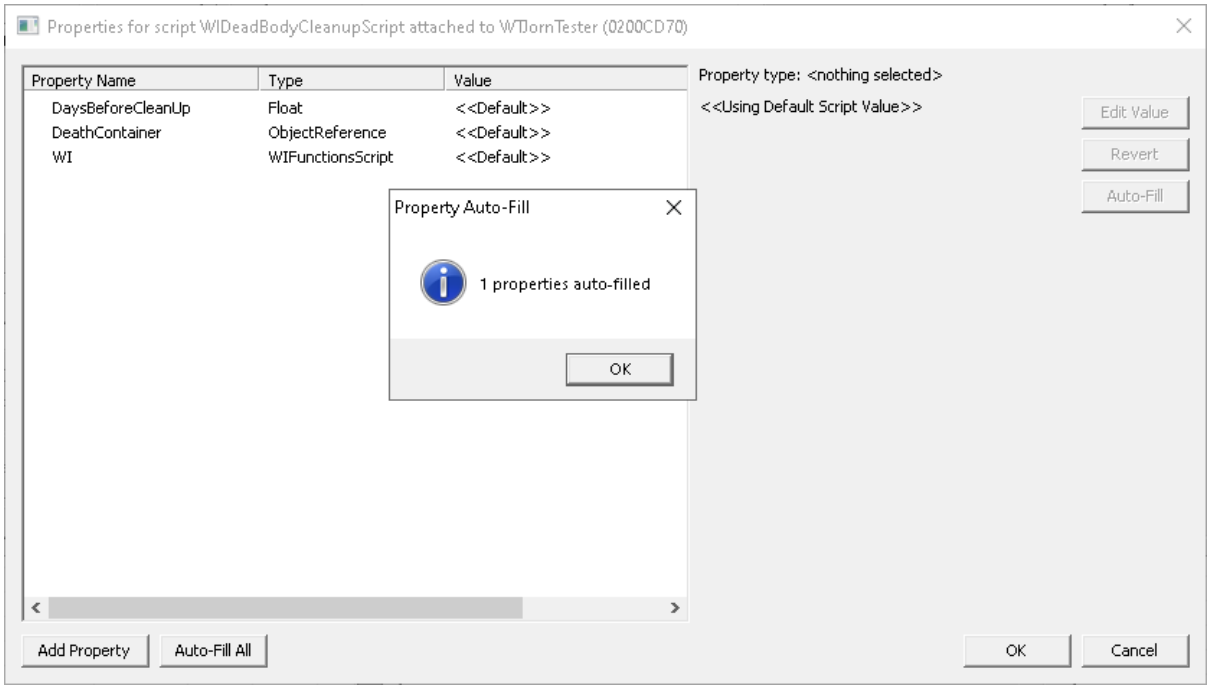


Figure 465 - Auto-fill properties.

Click OK to close out of script properties.

Lastly, let's place our NPC in the world.

Click OK to close out of Actor properties for now.

Click on WTJornTester in the Object Window to highlight him, then drag and drop him into the render window to place him into the world.

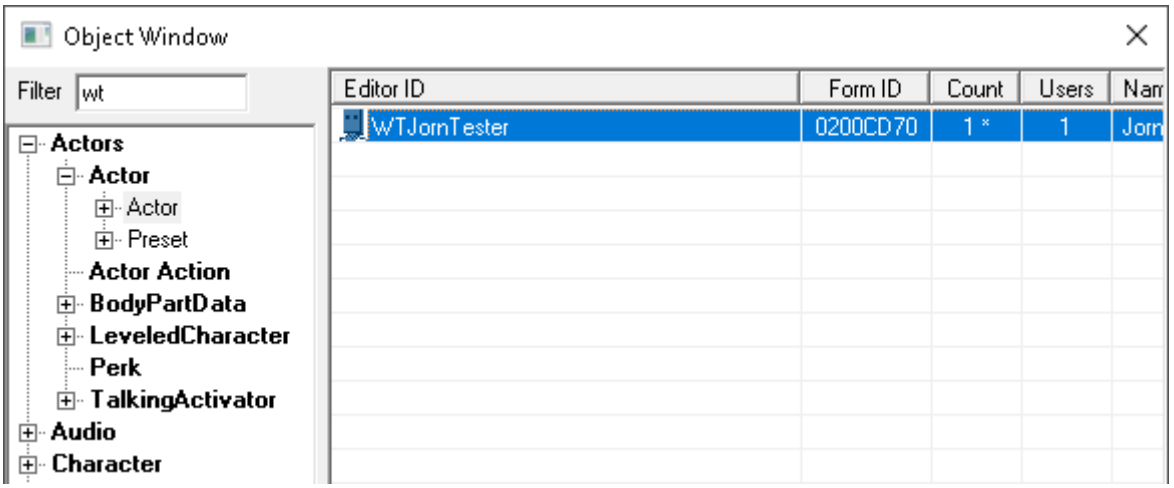


Figure 466 - WTJornTester highlighted in the Object Window.

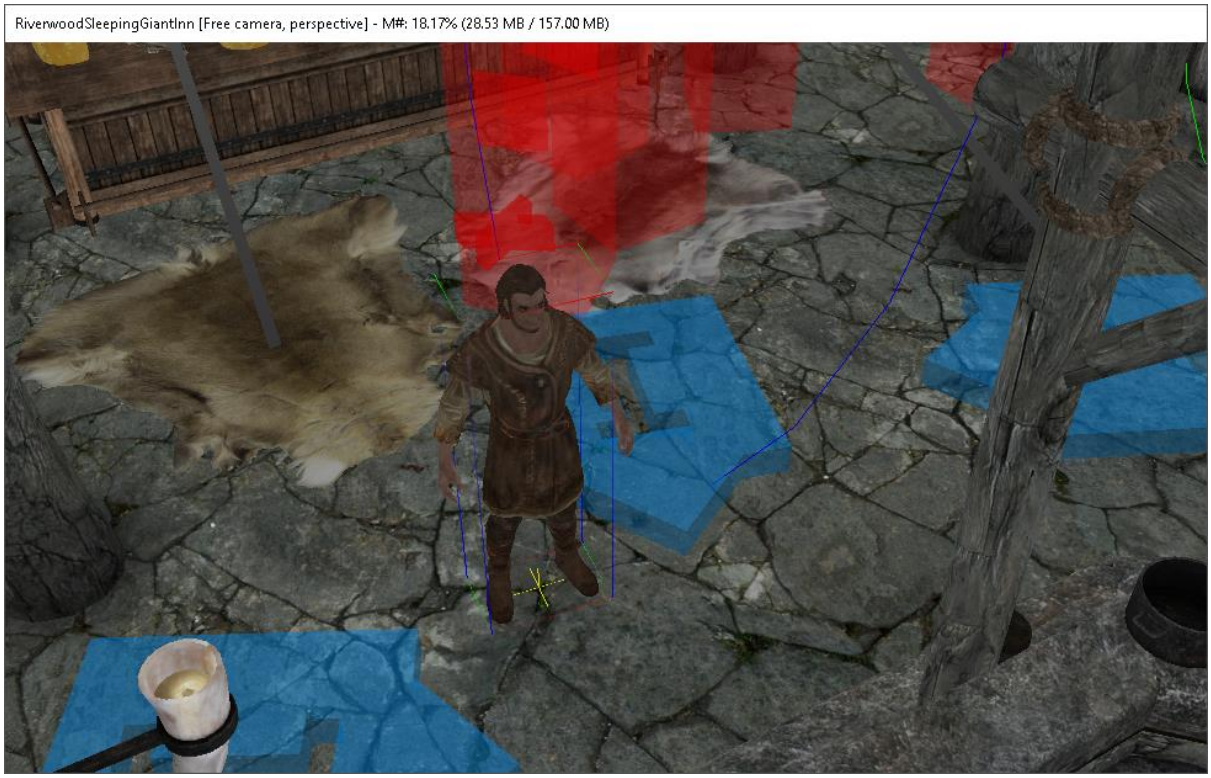


Figure 467 - WTJornTester added to the Riverwood inn.

For more information on what each field does, see the Actor article on the Creation Kit wiki.

ADDING AND CREATING AI PACKAGES

AI packages are used to set up an NPC's daily schedule. They are used to define when an NPC sleeps, eats, works or travels.

Packages are added to an NPC under the AI Packages tab.

AI Packages can be added here to give the NPC a daily schedule. In this section we're going to continue with our previous example by adding some packages to our Jorn the Tester NPC.

To add a package, right-click in the packages list and select Add.

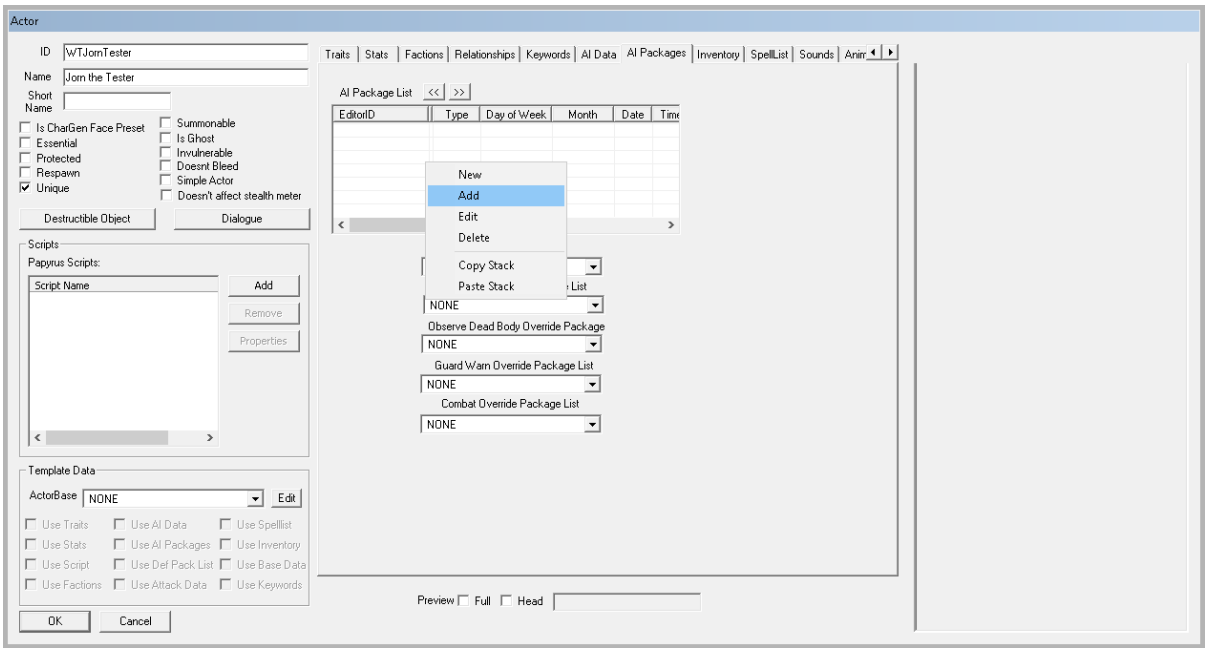


Figure 468 - AI Packages tab.

Filter by 'defaultsleep', click on DefaultSleepEditorLoc0x7 to highlight it then click OK.

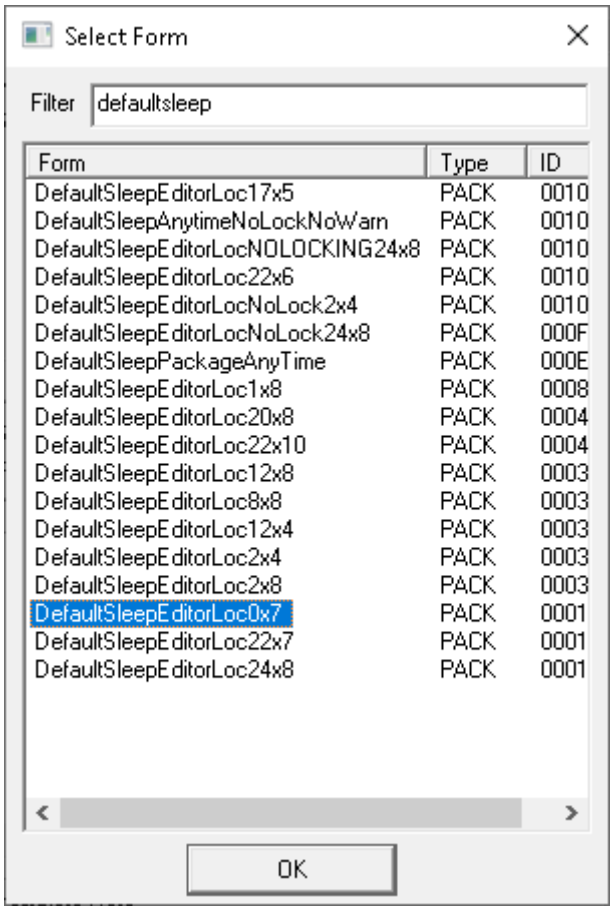


Figure 469 - Adding a sleep package.

This package will make the NPC sleep at its editor location from midnight to 7:00 AM. Editor location just means the spot at which the NPC was placed in the Creation Kit.

Let's add another package. Right-click in the packages list again and select Add.

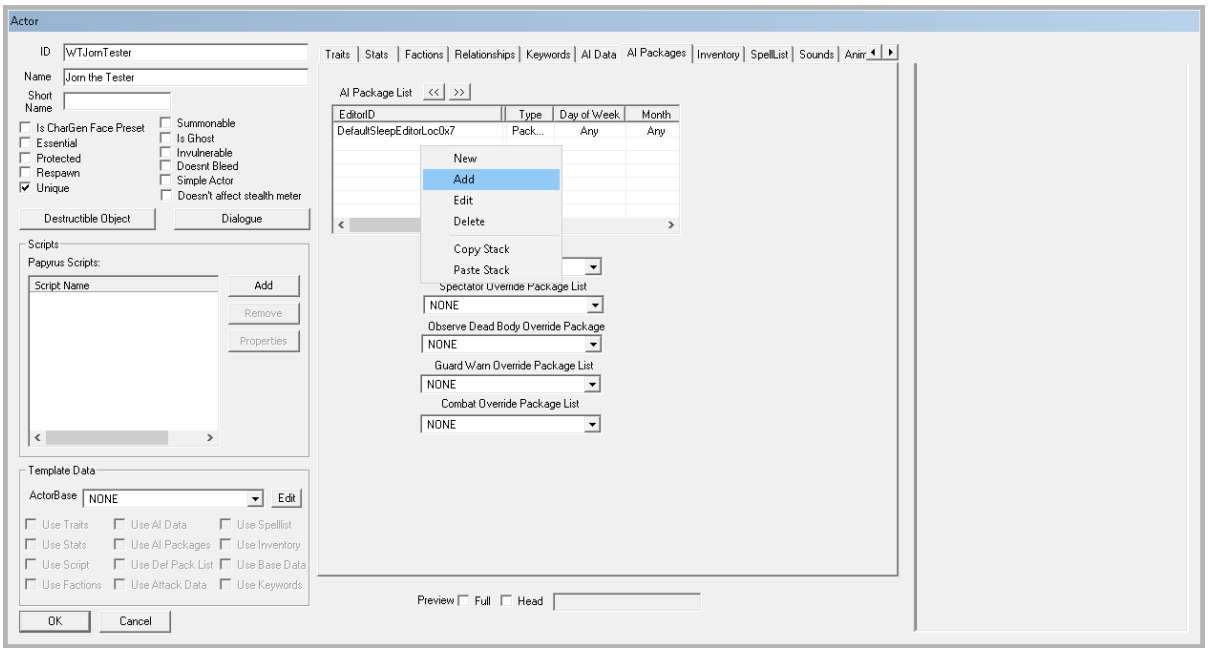


Figure 470 - Adding a second package.

Filter by 'defaulteat', click on DefaultEatEditorLoc8x1 to highlight it then click OK.

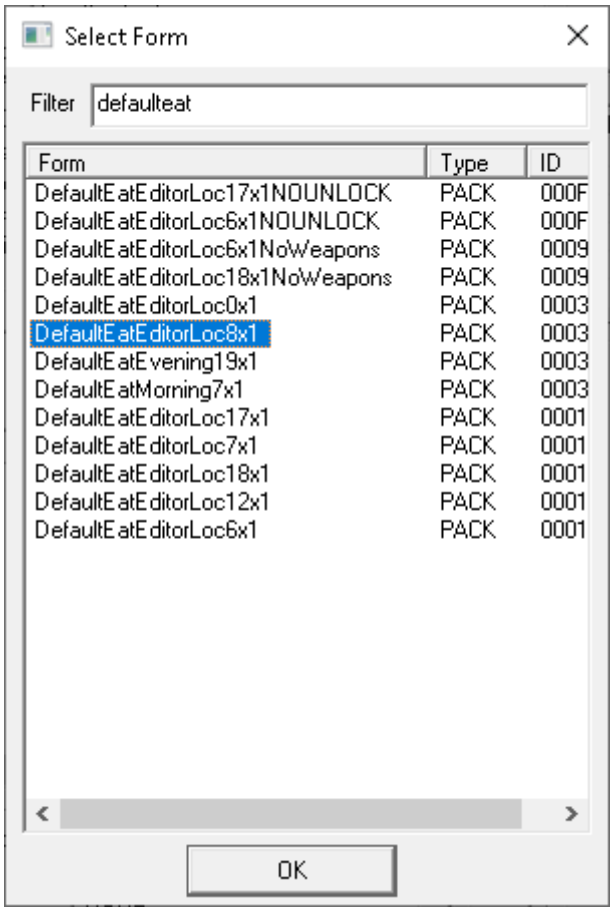


Figure 471 - Adding an eat package.

This package will make the NPC eat from 8:00 AM to 9:00 AM.

Use the '<<' and '>>' buttons to ensure DefaultEatEditorLoc8x1 is at the top of the list.

Let's add another eat package for dinner time.

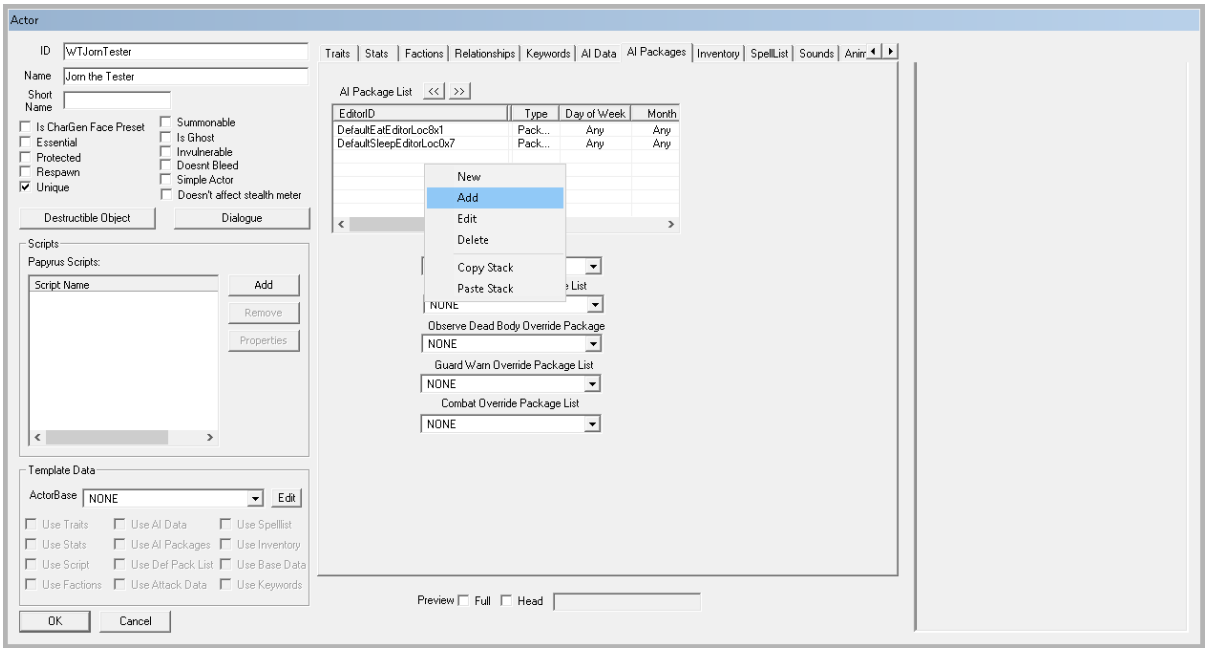


Figure 472 - Adding another eat package.

Filter by 'defaulteat' again, this time select DefaultEatEvening19x1 then click OK.

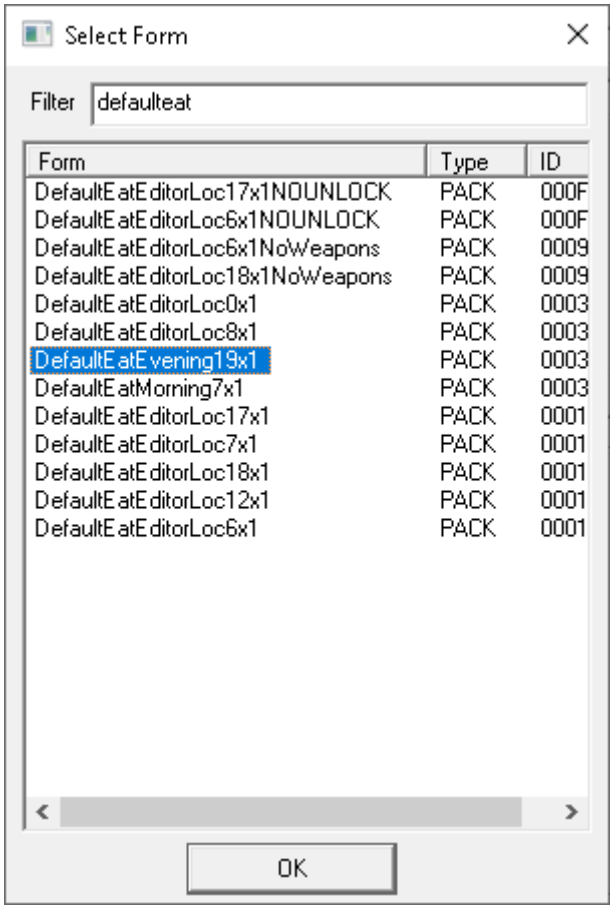


Figure 473 - Adding another eat package.

Next, let's give this NPC something to do during the day. This time, let's create a custom AI package.

Right-click in the packages list and select New.

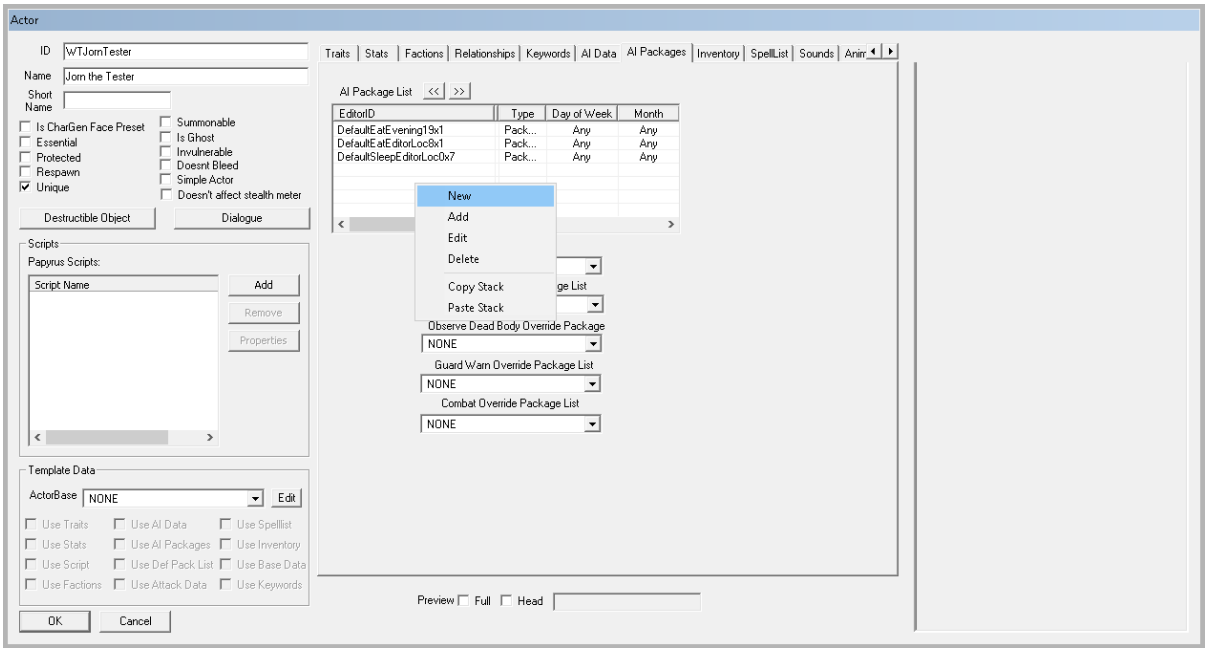


Figure 474 - Creating a new package.

Set the package ID. For this example, I went with WTJornSandboxRiverwood9x10 as I plan to have Jorn wander around outside in Riverwood from 9:00 AM until 7:00 PM.

Note: Make a habit of adding the active hours of the package to the package ID.

To accomplish this, first let's set the Package Template to Sandbox. Sandbox packages will allow the NPC to wander around and interact with idle markers and furniture within a defined radius.

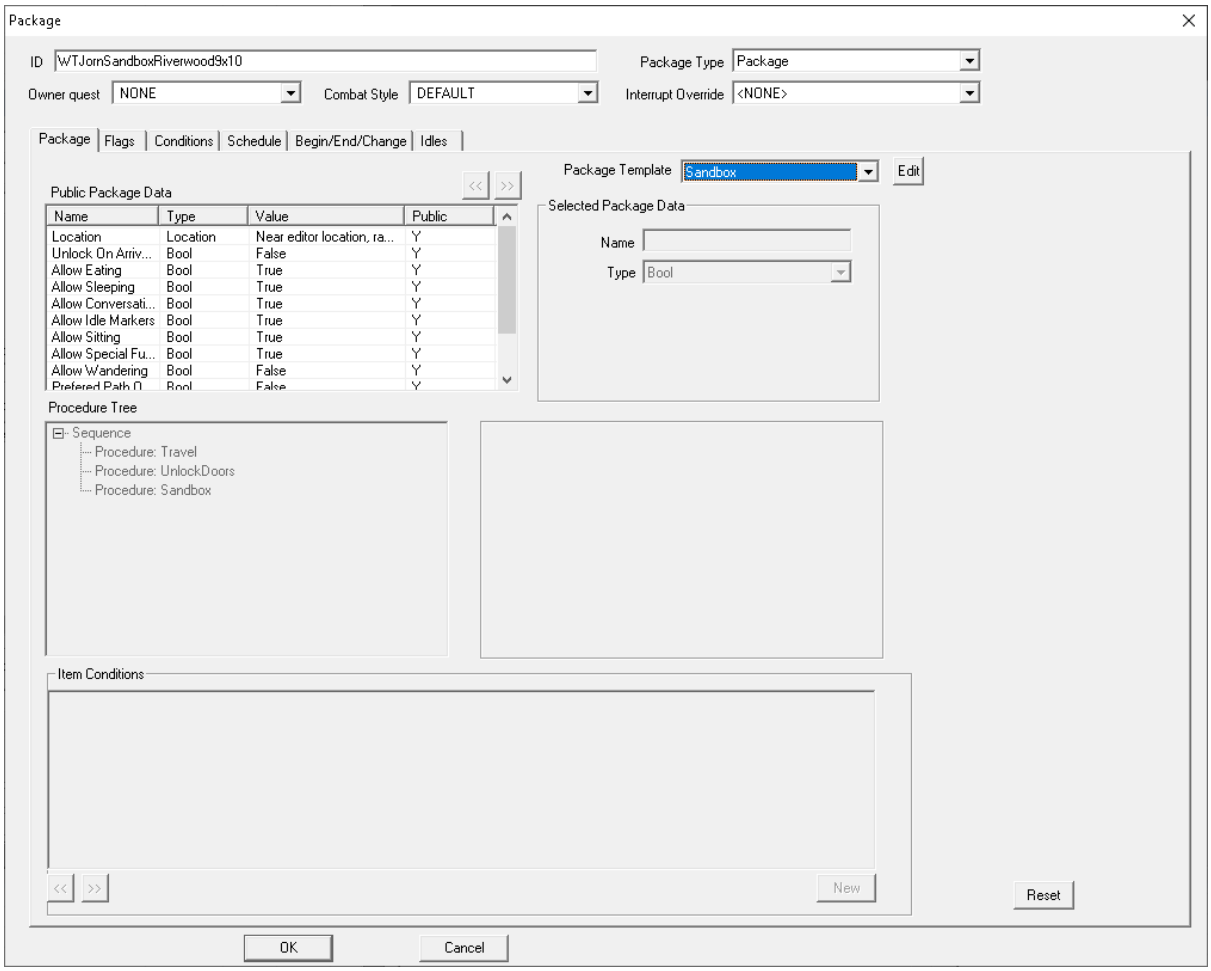


Figure 475 - Package tab.

Under Public Package Data, click on Location to highlight it, then set the Radius field to 4096.

Note: A world space cell is 4096 x 4096 units in size.

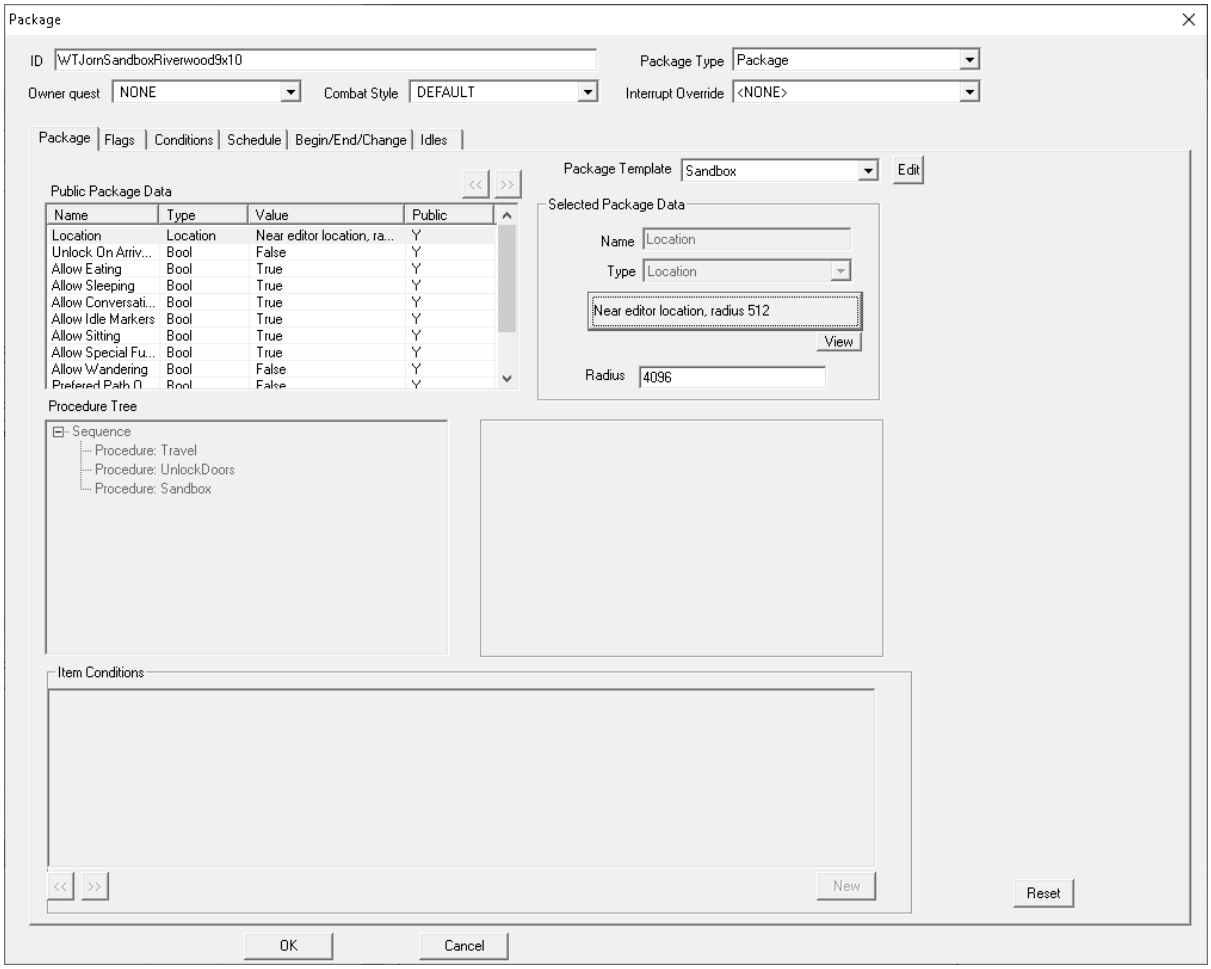


Figure 476 - Increasing the sandbox radius.

Click on the button above it that currently reads 'Near editor location, radius 512'.

Select Near Reference, then click on the Select Reference button to choose the object in the render window that the NPC should sandbox around.

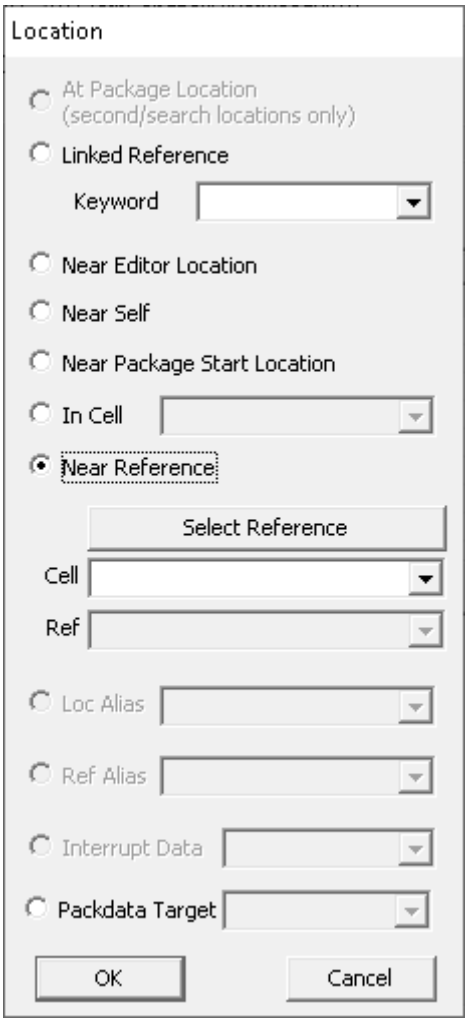


Figure 477 - Choosing the reference to sandbox around.

Select Riverwood's LocationCenterMarker marker. The LocationCenterMarker marker is the XMarker with the yellow box above it.



Figure 478 - Riverwood's LocationCenterMarker marker.

With the reference selected, click OK.

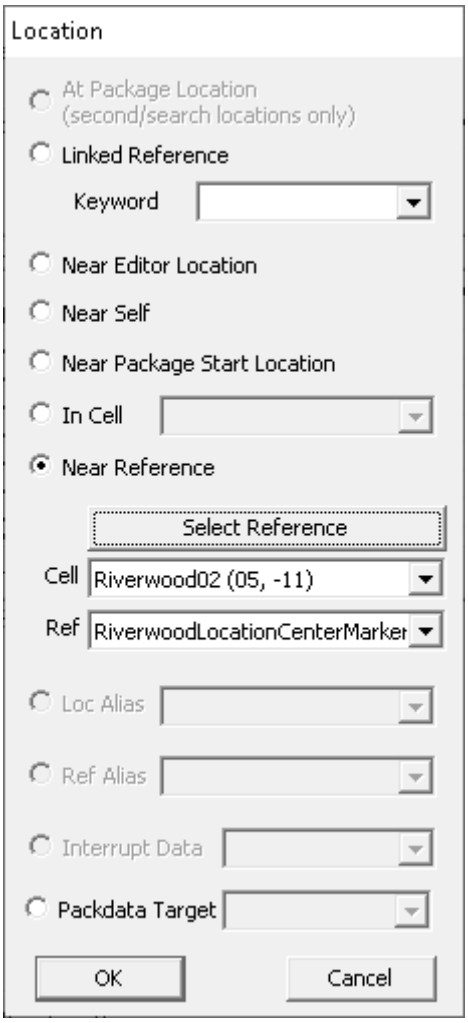


Figure 479 - Reference added to location.

Scroll down to Allow Wandering and tick the Value tickbox to enable it. This will let the NPC wander to a randomly chosen spot within the sandbox radius. If this is not ticked, the NPC will only interact with idle markers and furniture within the sandbox radius.

If you wanted to set how often an NPC changes what they're doing, increase the Energy value under Public Package Data. By default this is set to 50.

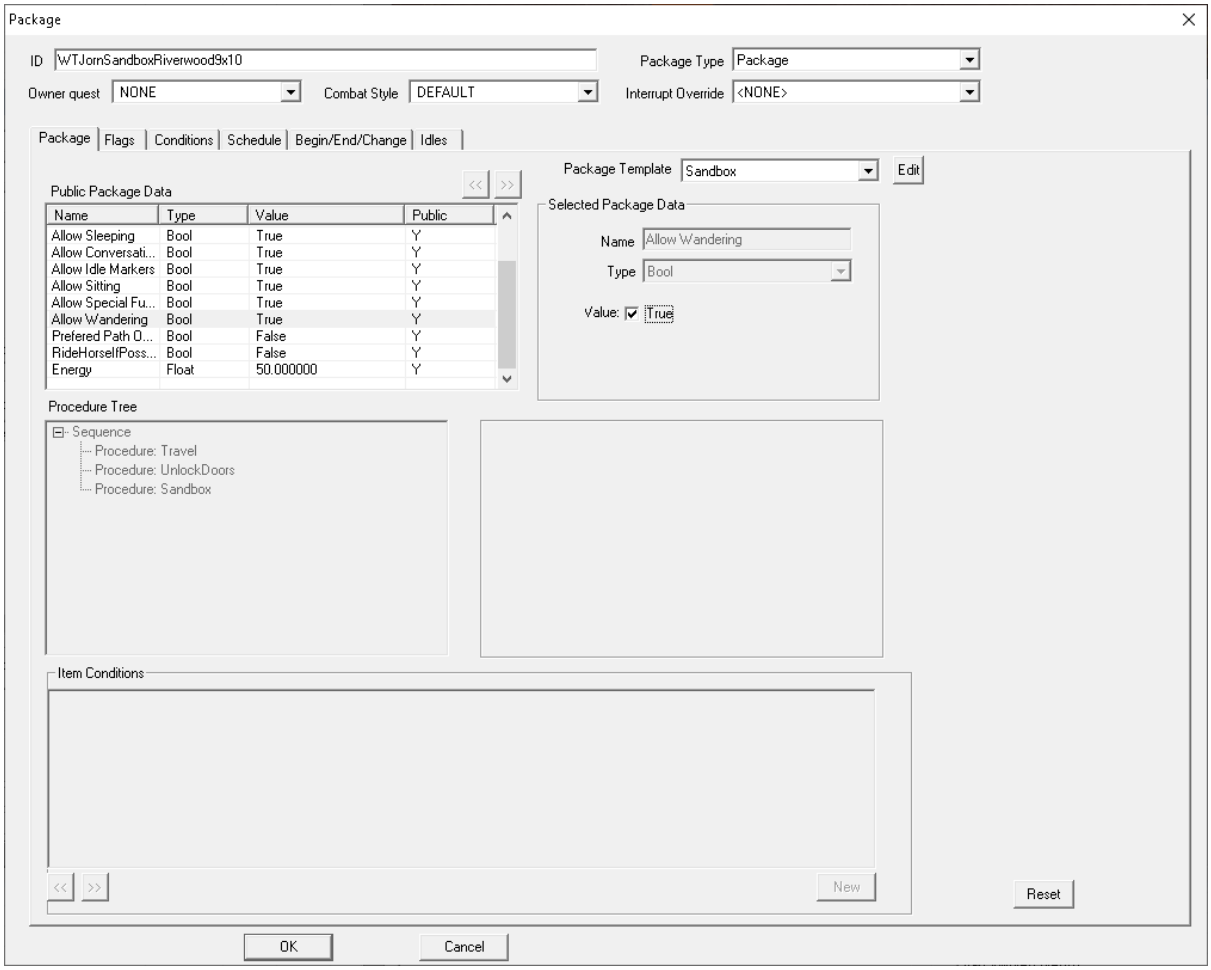


Figure 480 - Enabling Allow Wandering.

Go to the Schedule tab. Set the Hour drop-down to 9 and the Duration field to 10. The NPC will only use this sandbox package from 9:00 AM until 7:00 PM.

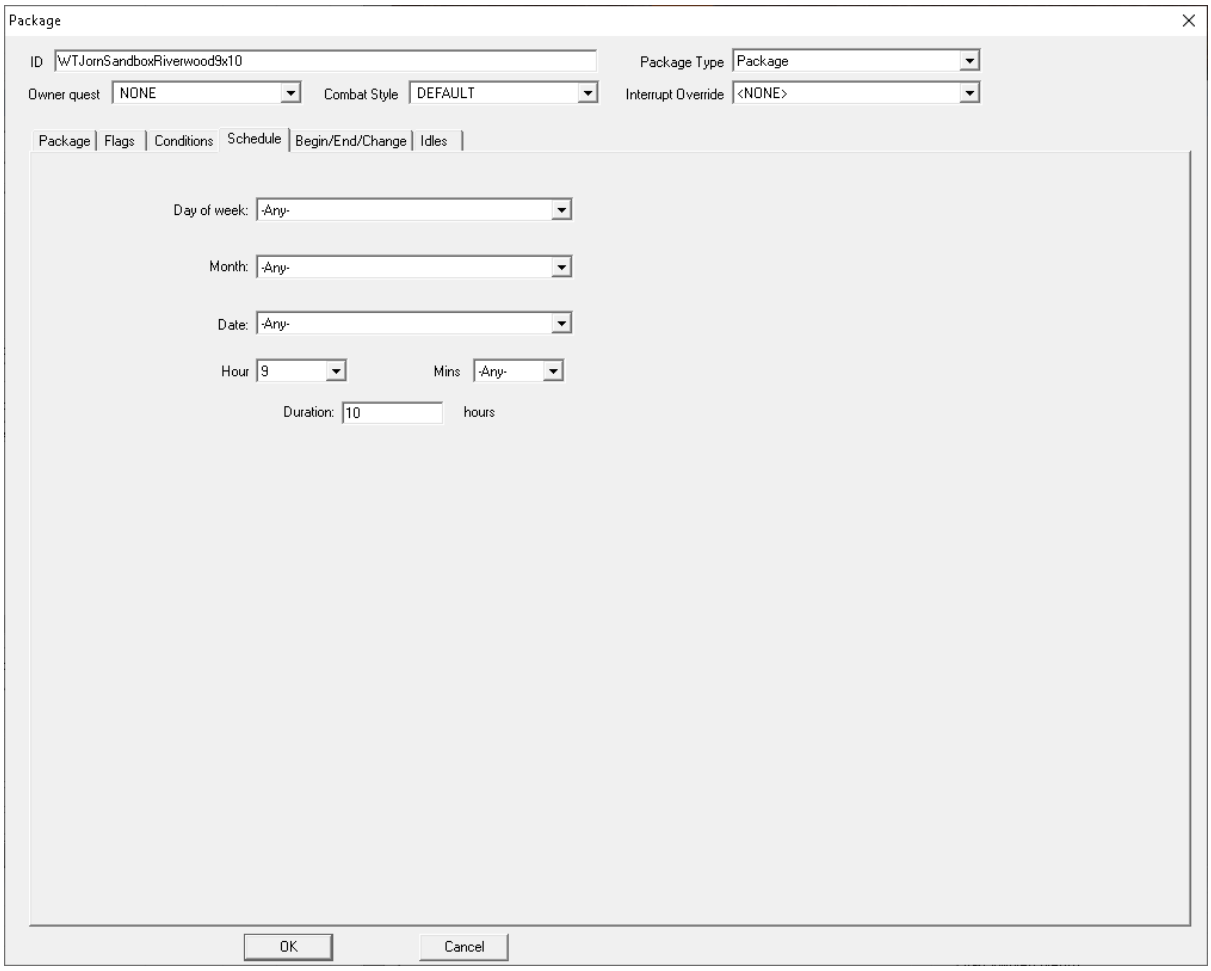


Figure 481 - Adding a start and stop time to a package.

Click OK to close out of Package properties.

Our AI Packages tab should now look like this:

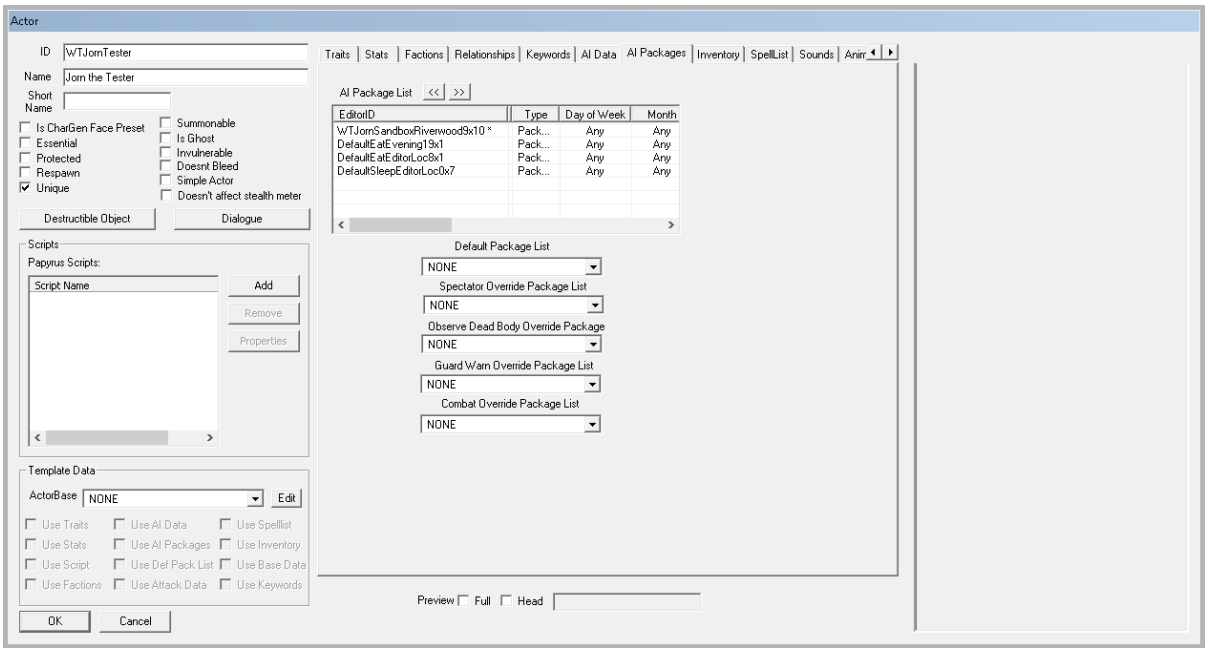


Figure 482 - AI Packages tab with four packages set up.

Again, use the '<<' and '>>' buttons to ensure WTJornSandboxRiverwood9x10 is at the top of the list.

Let's add one more package to cover the evening hours.

Right-click in the packages list and select Add.

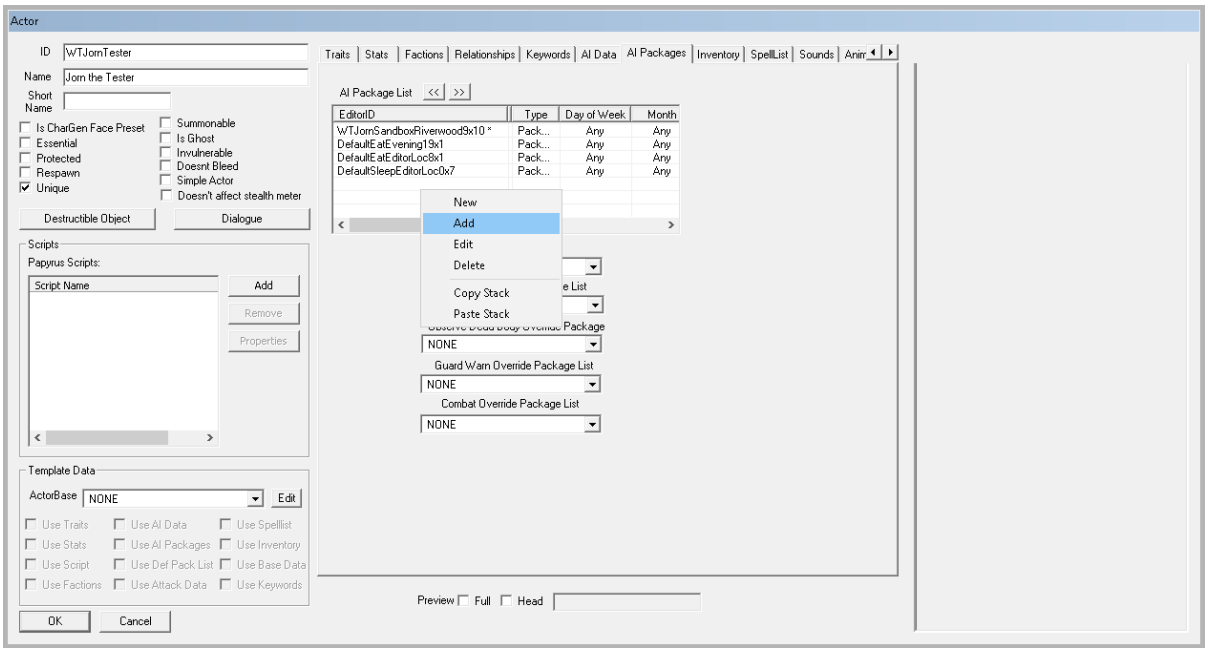


Figure 483 - Adding a fifth package.

Filter by 'defaultsandbox', click on DefaultSandboxEditorLocation1024 to highlight it then click OK.

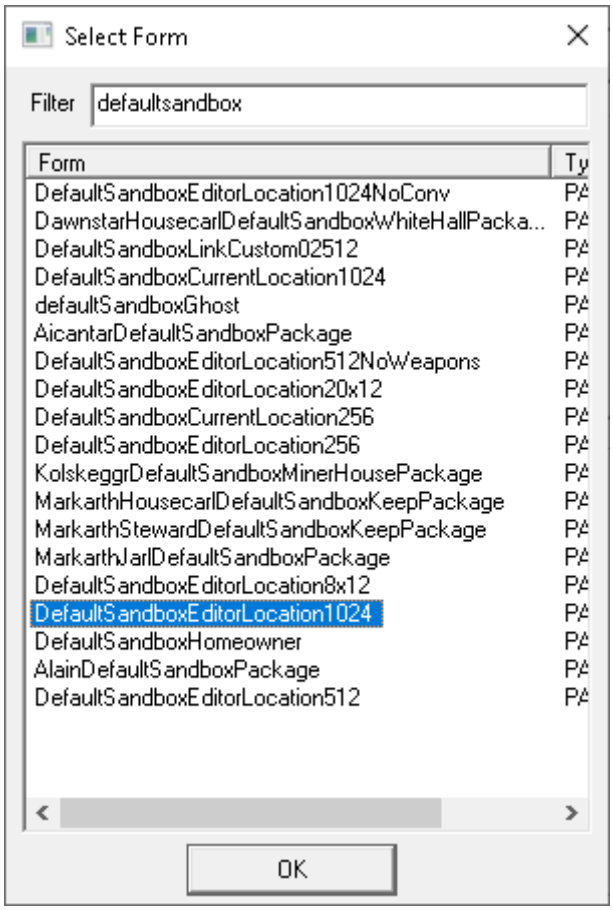


Figure 484 - Adding a DefaultSandboxEditorLocation1024 package.

Important: Packages are processed sequentially starting from the topmost package.

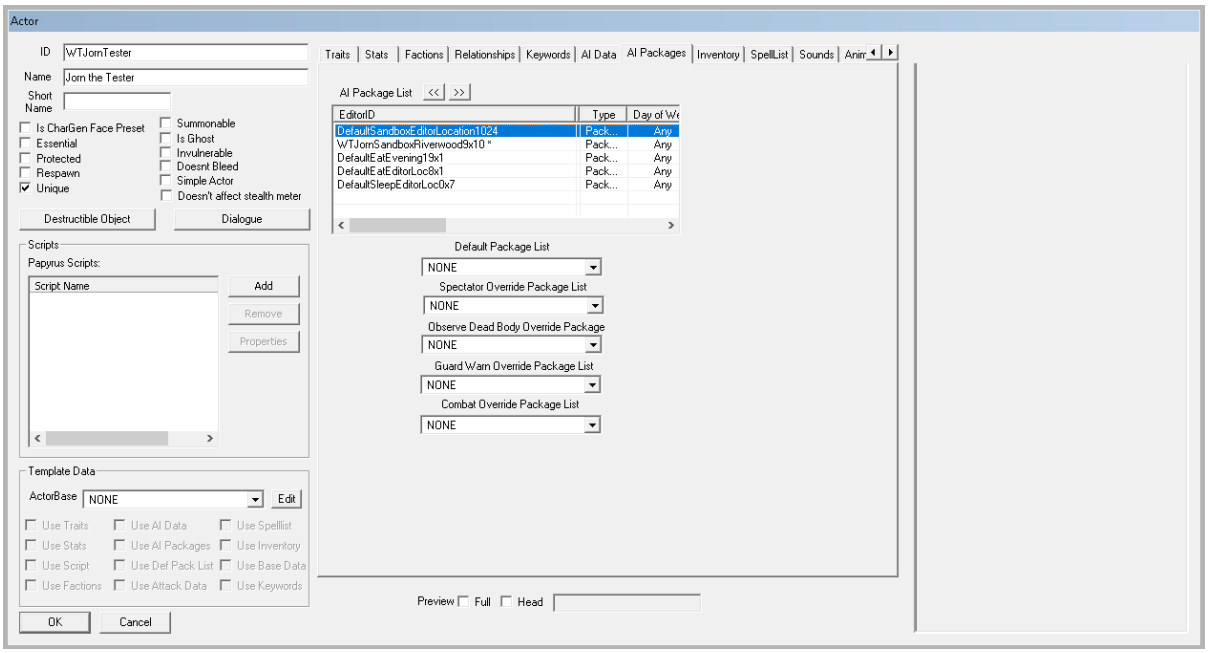


Figure 485 - AI Packages tab with five packages set up.

Click on DefaultSandboxEditorLocation1024 to highlight it, then click '>>' until it has been moved to the bottom of the packages list.

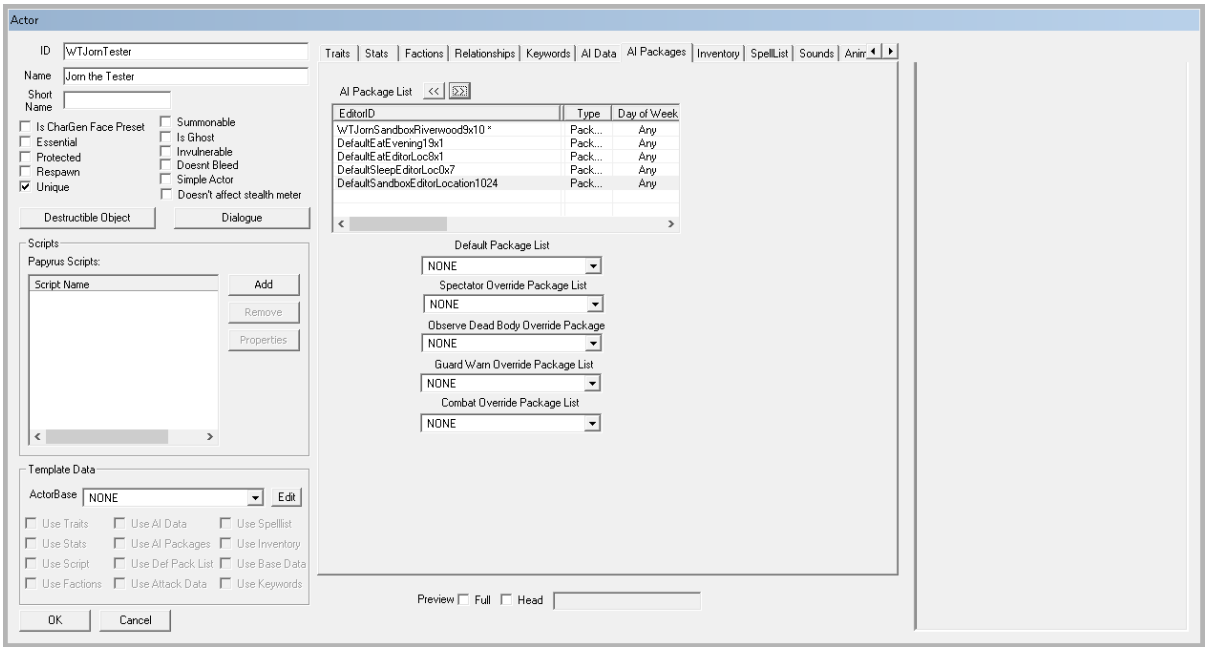


Figure 486 - DefaultSandboxEditorLocation1024 moved to the bottom of the packages list.

This will make Jorn sandbox in the Riverwood inn where we placed him in the previous section.

MAKING A MERCHANT

In this section I'll be showing you how to set an NPC up as a merchant. I'll be using Jorn the Tester, the NPC I created in the previous sections, in this example.

The first thing we need to do is make the merchant container. This container is usually placed in out of reach of the player, i.e.: under floors or beneath the ground.

The easiest way to make a new merchant container is to duplicate an existing one. In the Object Window, go to WorldObjects > Container and filter by 'merchant'.

For this example, I duplicated MerchantRiftenPawnedPrawnChest by right-clicking on it and selecting Duplicate.

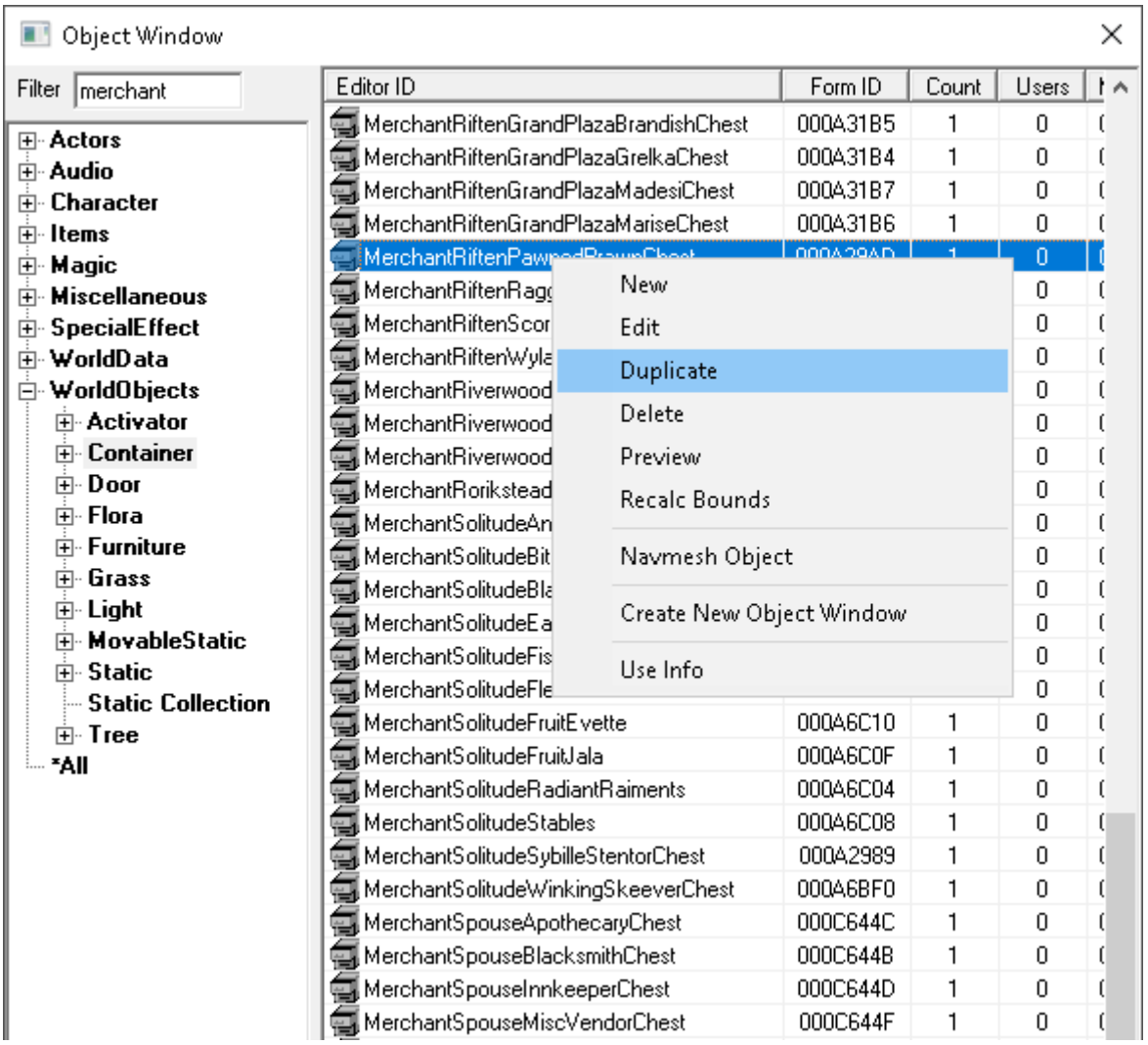


Figure 487 - Duplicating an existing merchant container.

Highlight the duplicate and press F2 to rename it.

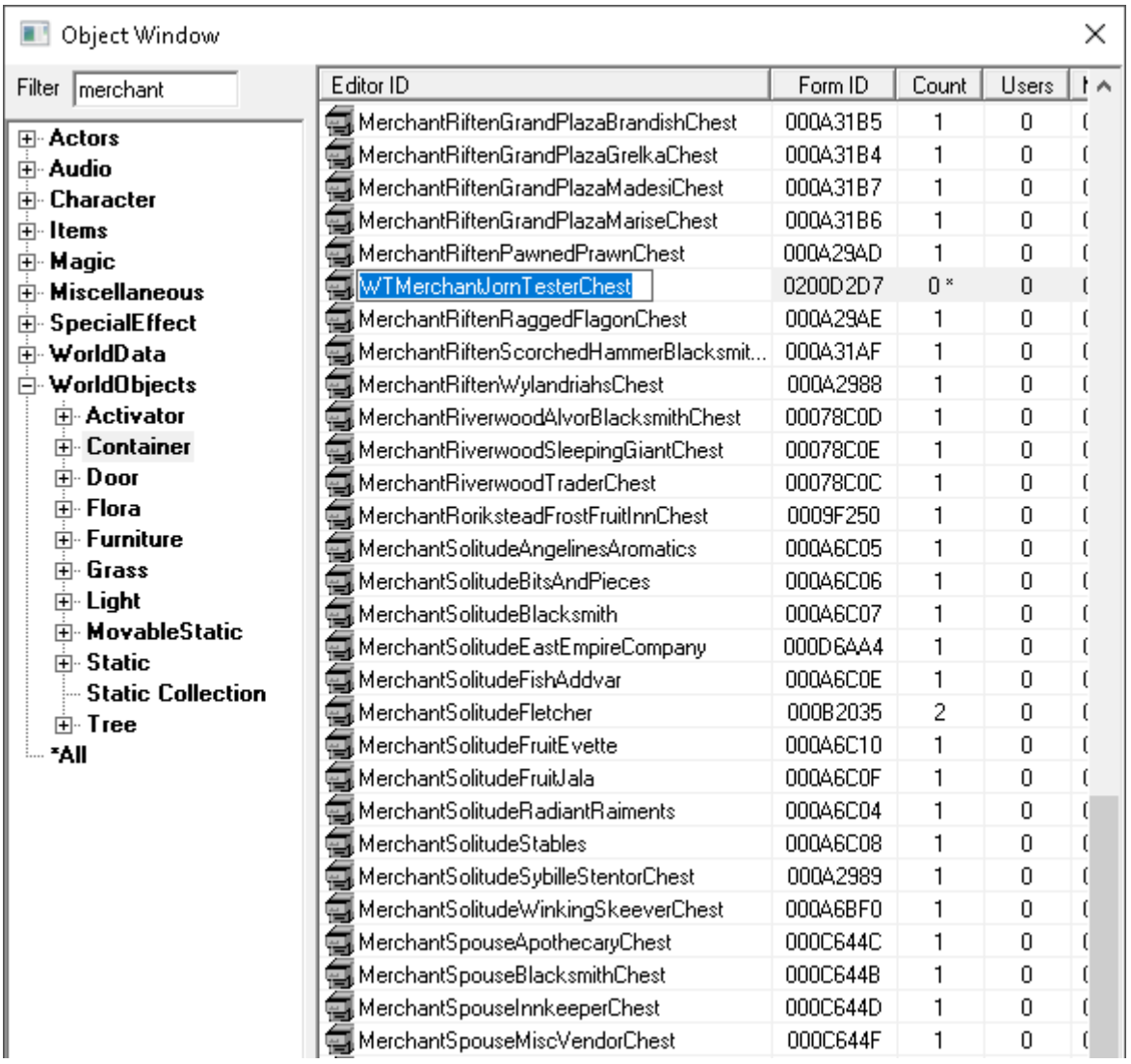


Figure 488 - Renaming the duplicate container.

When prompted to create a new object, select No.



Figure 489 - Create New Object prompt.

Objects that will form the merchant's inventory will be listed in the item list. Typically these will be levelled items so the merchant's daily stock will be randomized.

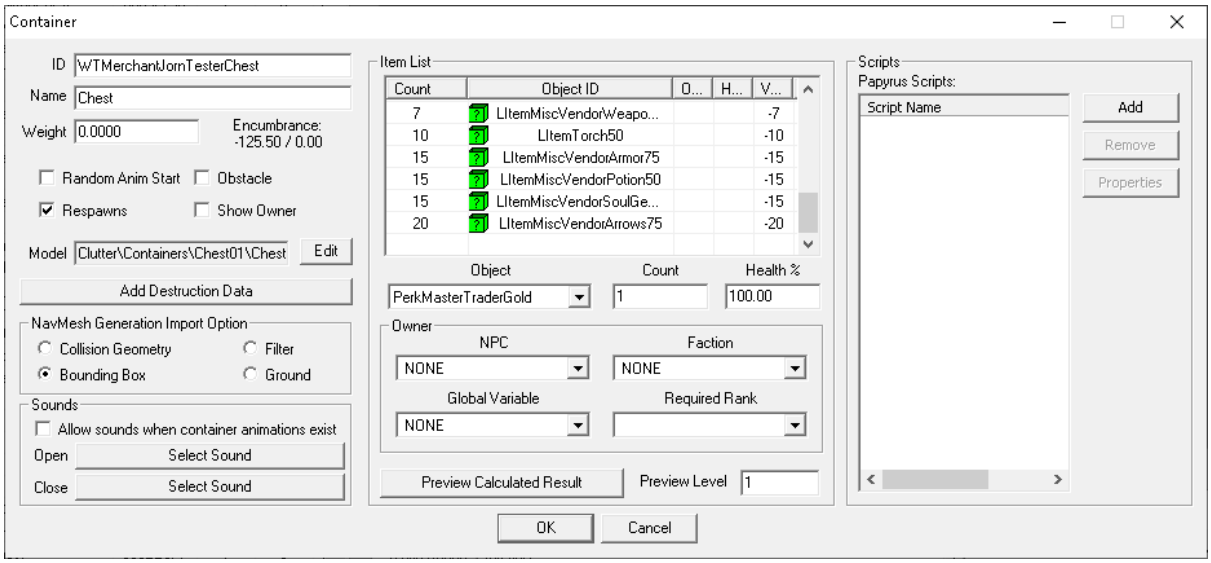


Figure 490 - Container inventory.

To remove an item from the item list, right-click on it and select Delete.

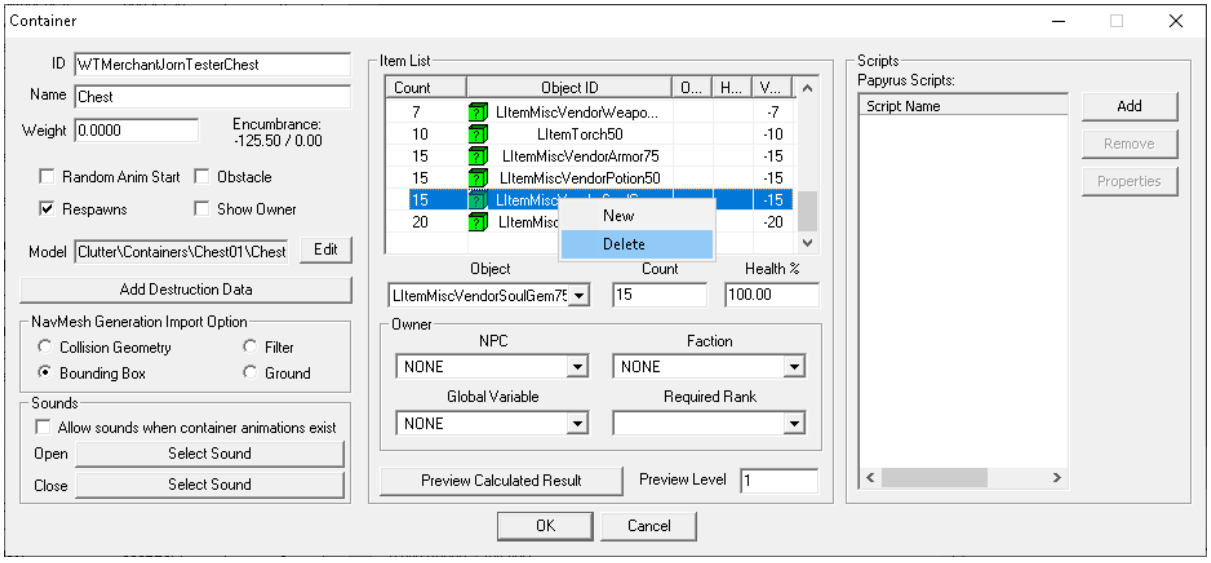


Figure 491 - Deleting an item from the item list.

To add an item to the item list, select New instead.

Once you're satisfied with the merchant's inventory, click OK to close out of the Container properties.

Drag and drop the container into the render window to add it to a cell. I added it to the RiverwoodSleepingGiantInn cell, the same cell Jorn, our NPC, was added to.

Make sure it's added below the floor so the player can't get to it.

It's not necessary to lock it or add ownership to it.

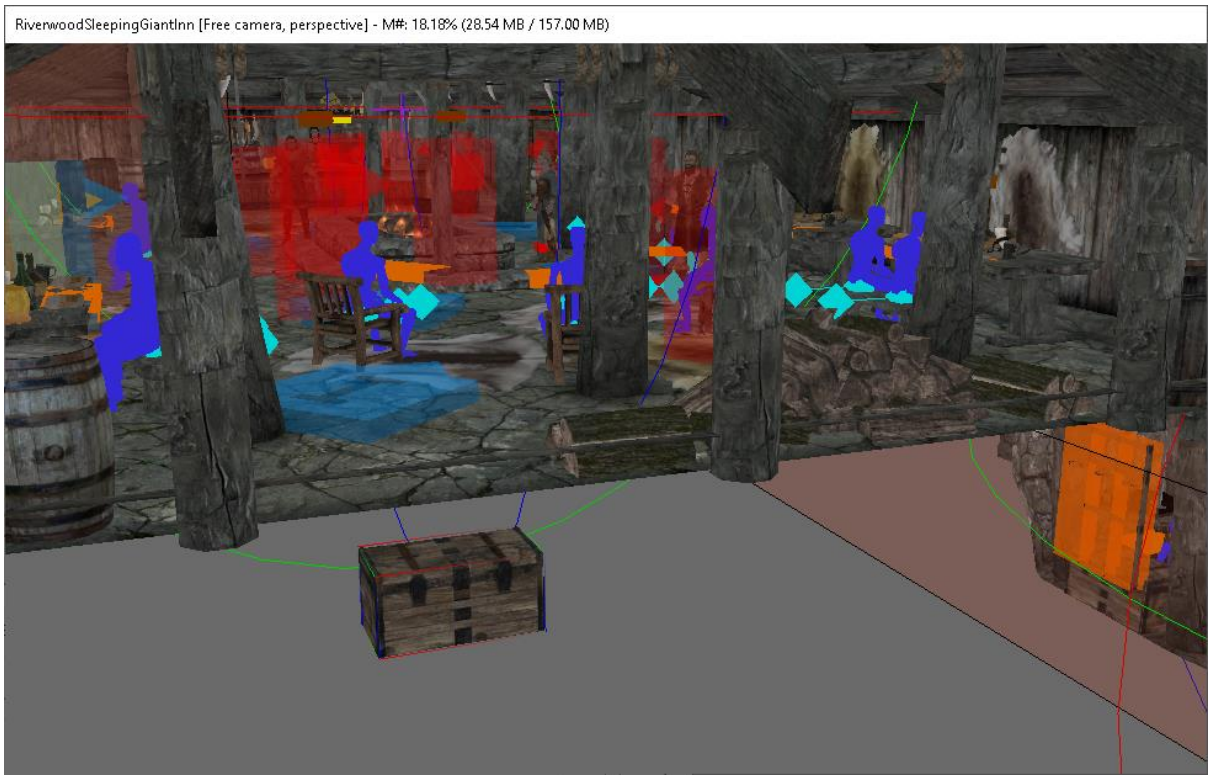


Figure 492 - Adding the container.

Next, we need to make a new faction linked to the merchant container. This will be used to determine business hours and vendor type.

In the Object Window, browse to Character > Faction and filter by 'services'.

For this example, I'm going to duplicate ServicesRiftenPawnedPrawn.

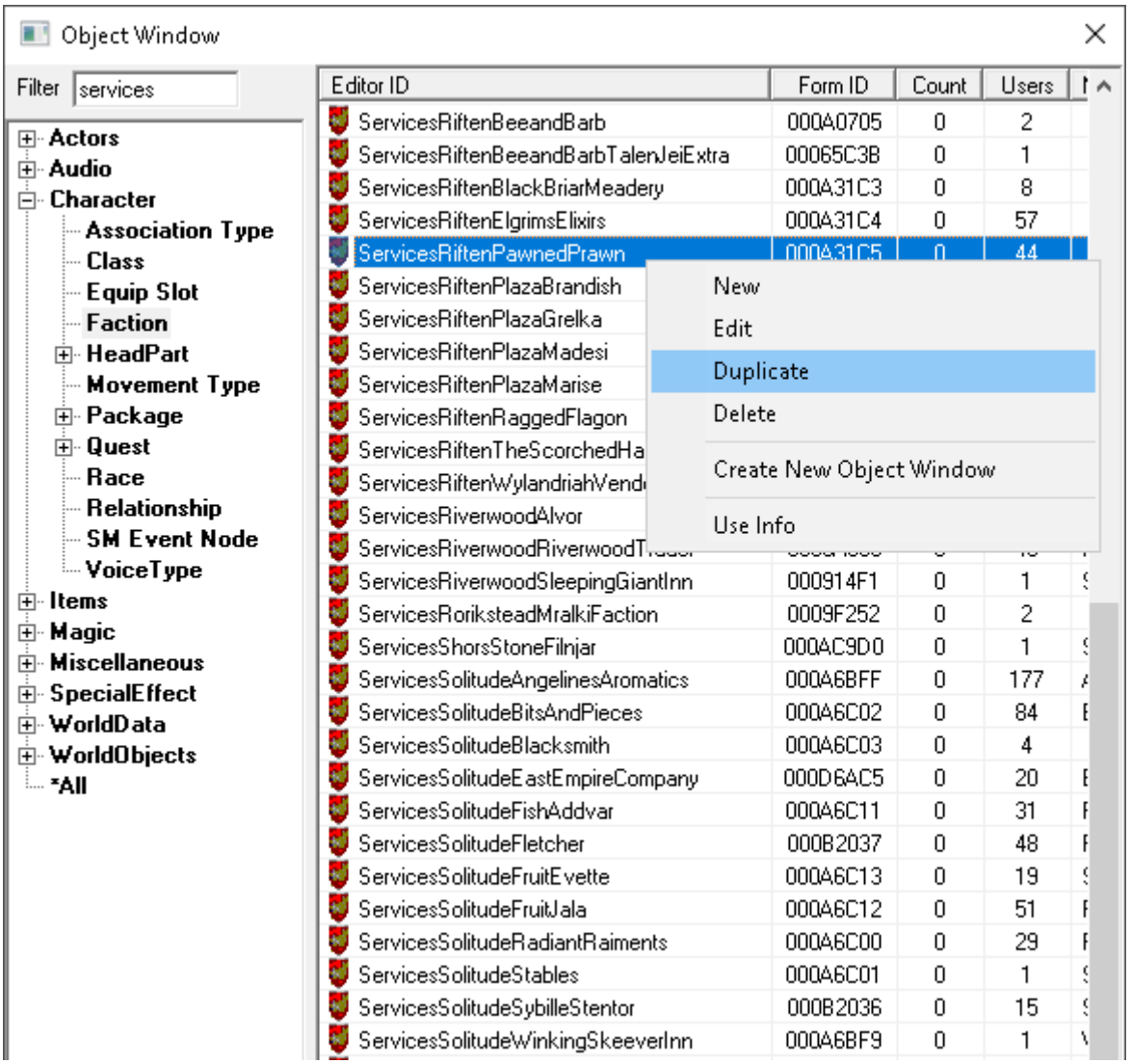


Figure 493 - Duplicating an existing faction.

Again, press F2 to rename it.

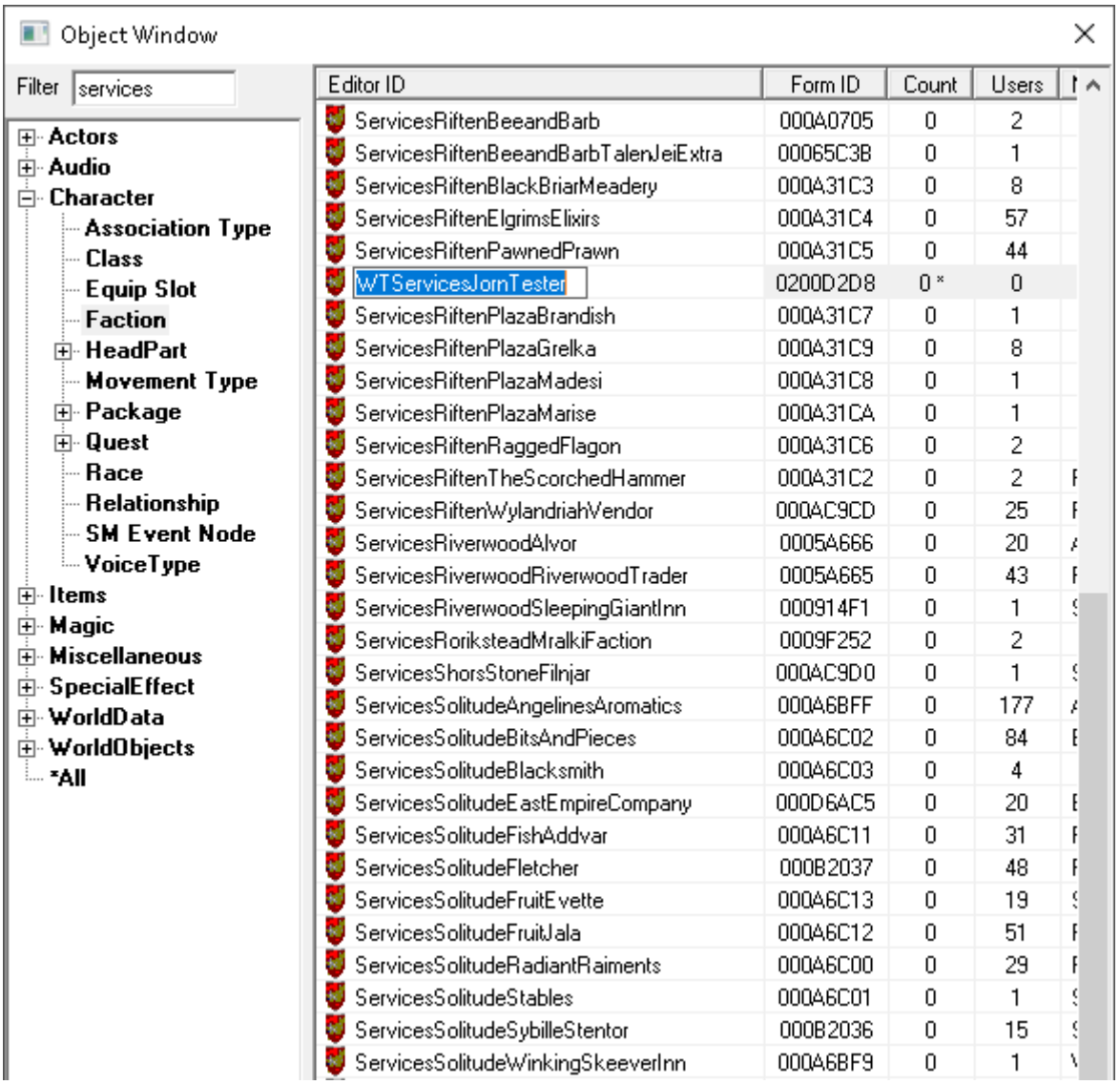


Figure 494 - Renaming the duplicate faction.

When prompted to create a new object, select No.



Figure 495 - Create New Object prompt.

Double-click or right-click on the faction and select Edit to open its properties.

Jorn, our sample NPC, returns to the Riverwood inn at 7:00 PM every evening, so let's have him run his little side business for 5 hours until he goes to sleep. I set the start hour for his shop to 19, ending at midnight.

Note: Normal business hours for most shops are 8 - 20.

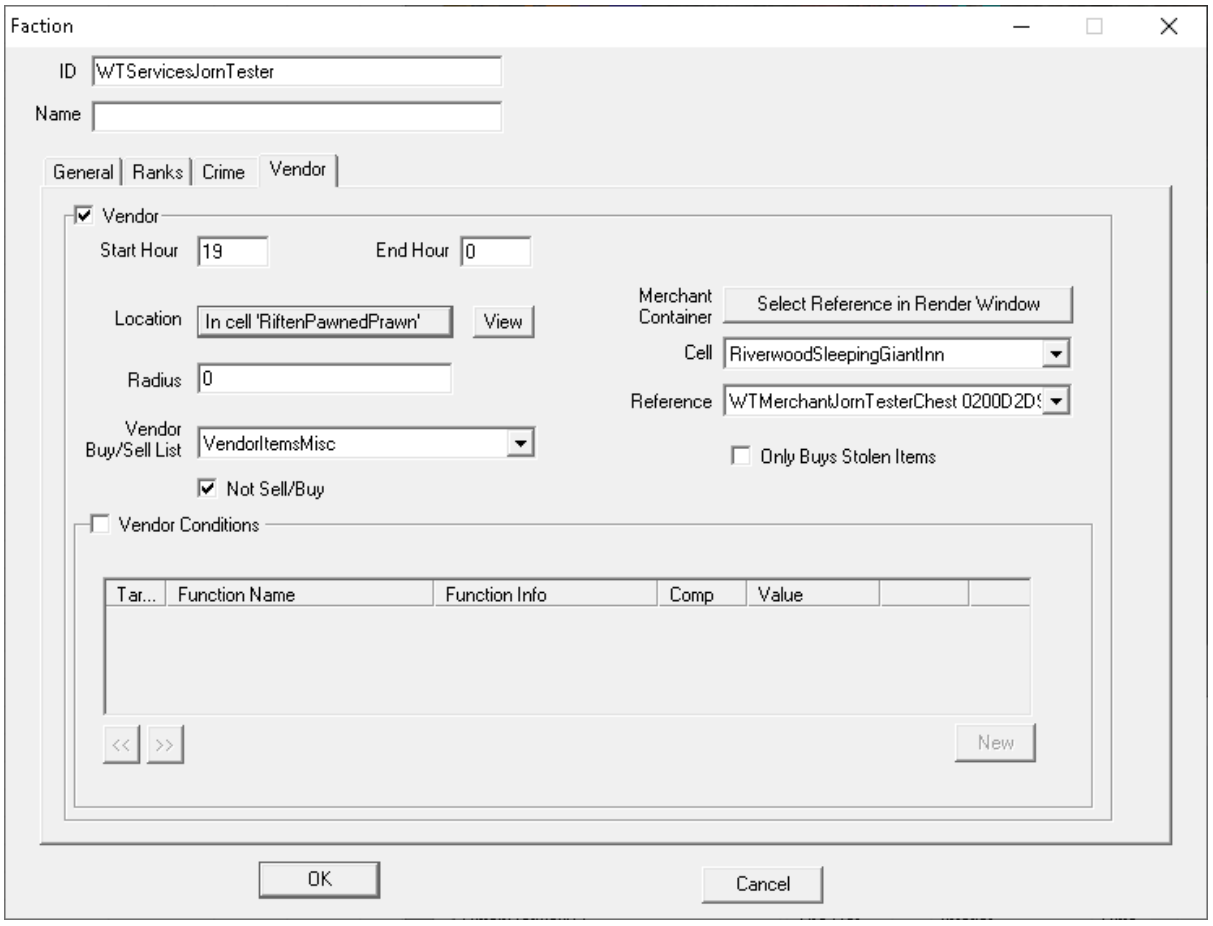


Figure 496 - Faction properties.

Click on the Merchant Container button and select the vendor chest you placed.

Lastly, click on the Location button.

Set it to RiverwoodSleepingGiantInn.

This means Jorn will only have things to buy and sell when he makes it back to the Riverwood inn.

Alternatively, if you set it to 'Near Self' instead, you'll be allowed to trade with Jorn as soon as the clock hits 7:00 PM regardless of where he is.



Figure 497 - Setting the vendor location.

Set the Radius field to 4096 so Jorn we'll be able to trade with Jorn no matter where he is in the the RiverwoodSleepingGiantInn cell.

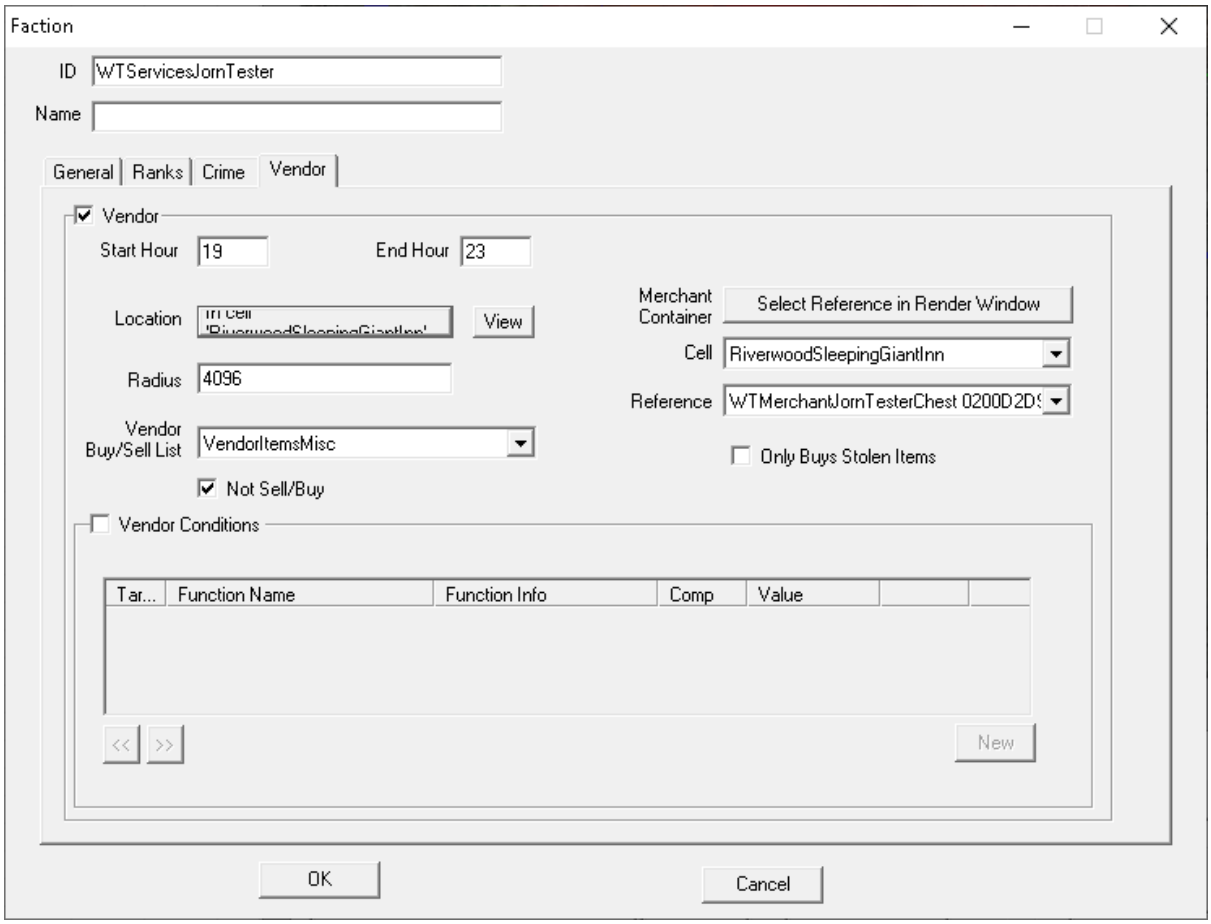


Figure 498 - Services faction set up.

Click OK to close out of Faction properties.

In the Object Window, go to Actors > Actor > Actor.

Double-click or right-click on our test NPC WTJornTester and select Edit.

Go to the Factions tab.

Right-click in the factions list and select New.

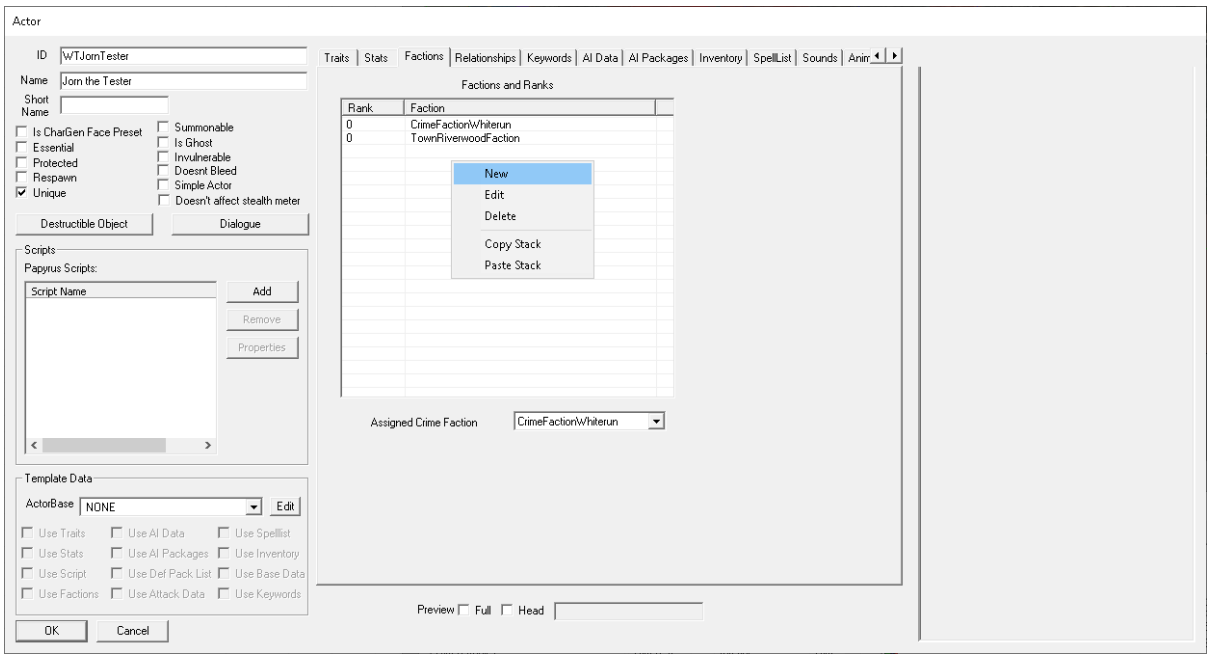


Figure 499 - Adding a new faction.

Select the services faction we just set up and click OK.

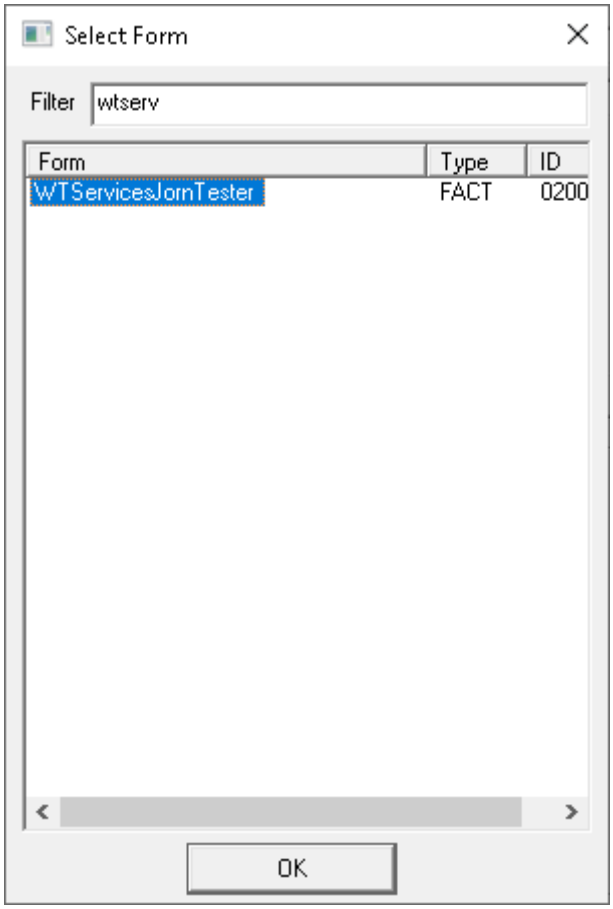


Figure 500 - Jorn's services faction highlighted.

We need to add Jorn to one more faction, so he has the dialogue that allows the player to trade with him.

Right-click in the factions list and select New.

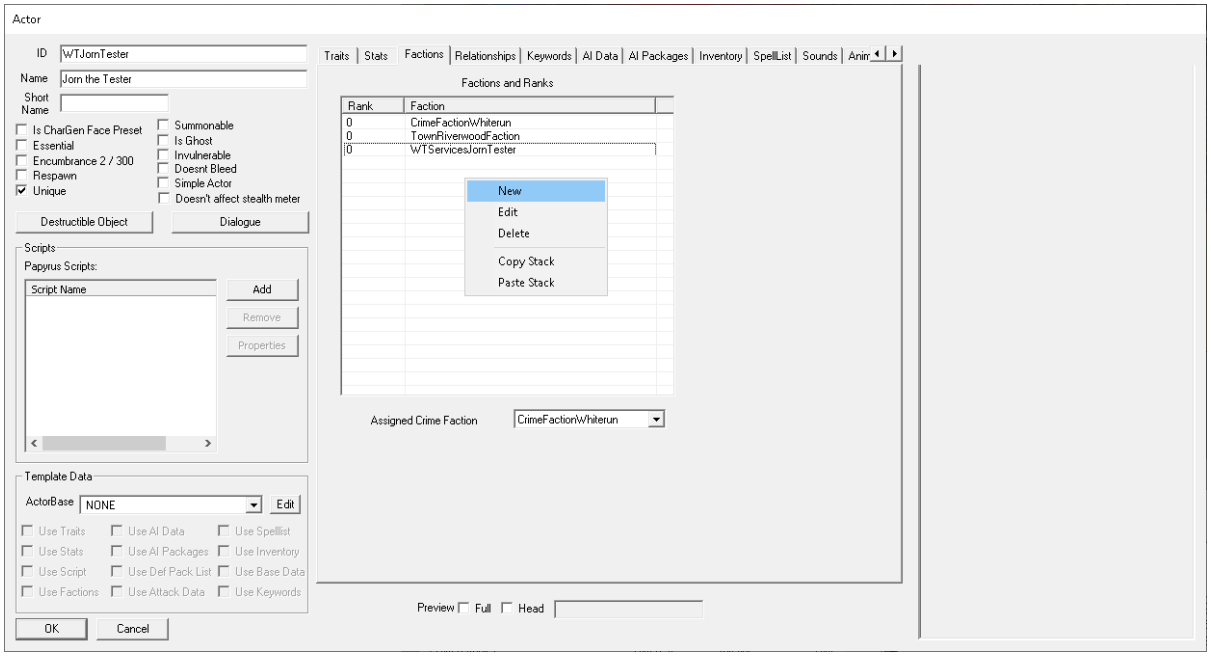


Figure 501 - Jorn added to his service faction.

Filter by 'jobmerchant', click on JobMerchantFaction to highlight it then click OK.

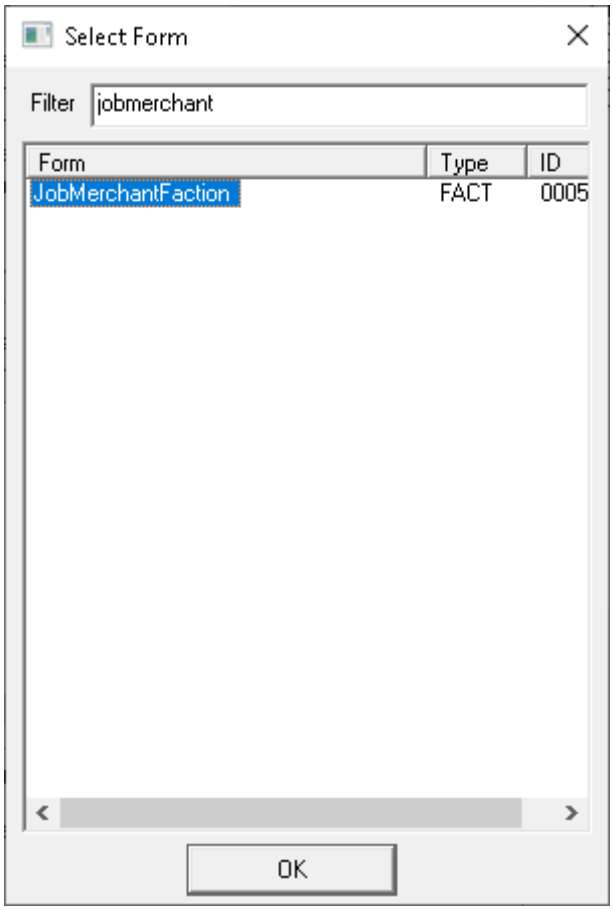


Figure 502 - JobMerchantFaction selected.

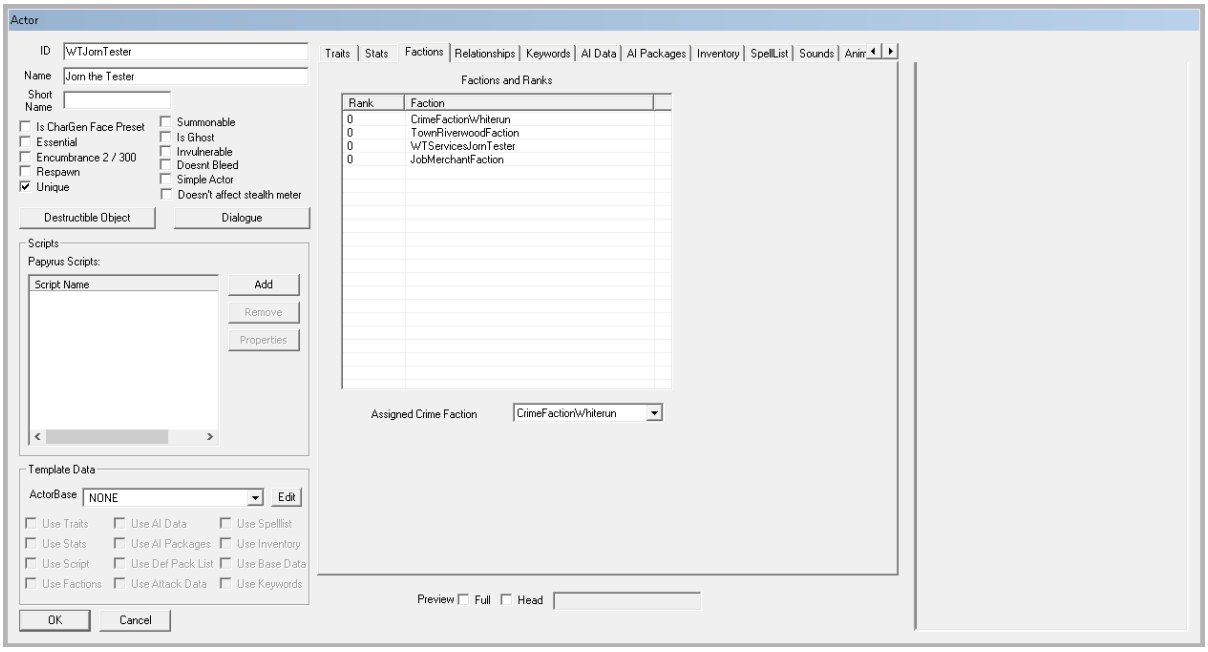


Figure 503 - Jorn added to the JobMerchantFaction.

If we go in-game and speak to Jorn, we should now be able to trade with him after 7:00 PM.



Figure 504 - Jorn with vendor dialogue during his business hours.

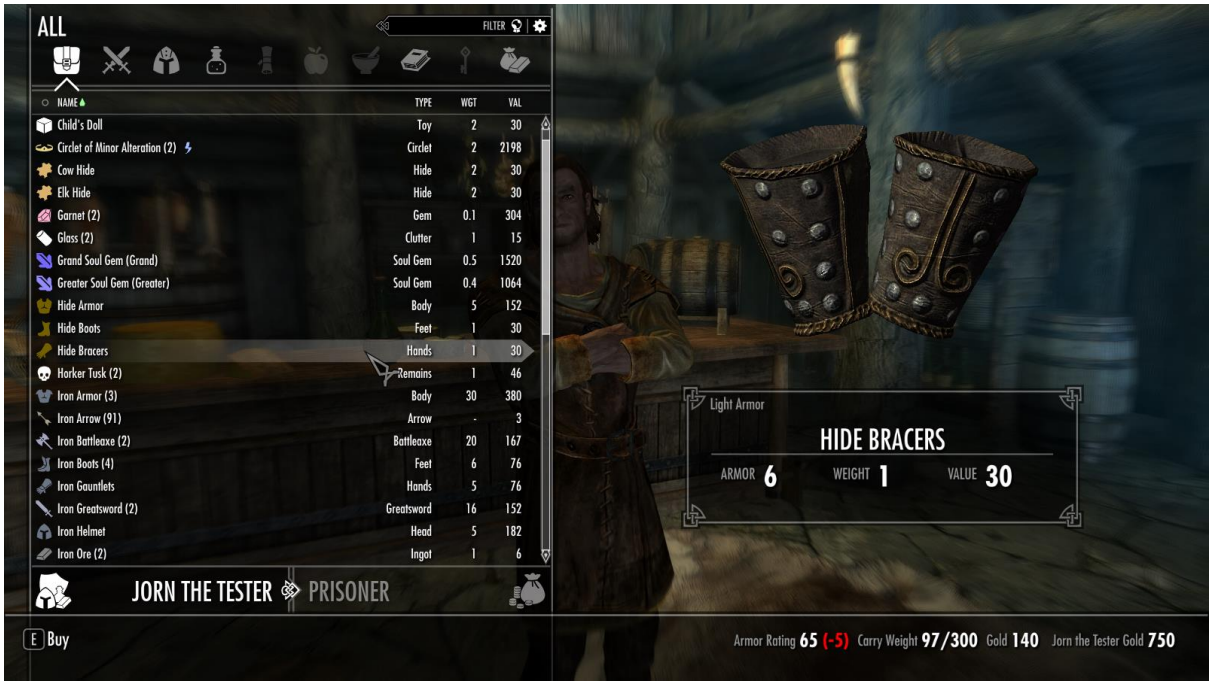


Figure 505 - Vendor inventory.

If your NPC uses custom voice acting, you can bring up the barter screen on a line of dialogue using the following papyrus:

```
akSpeaker.ShowBarterMenu()
```

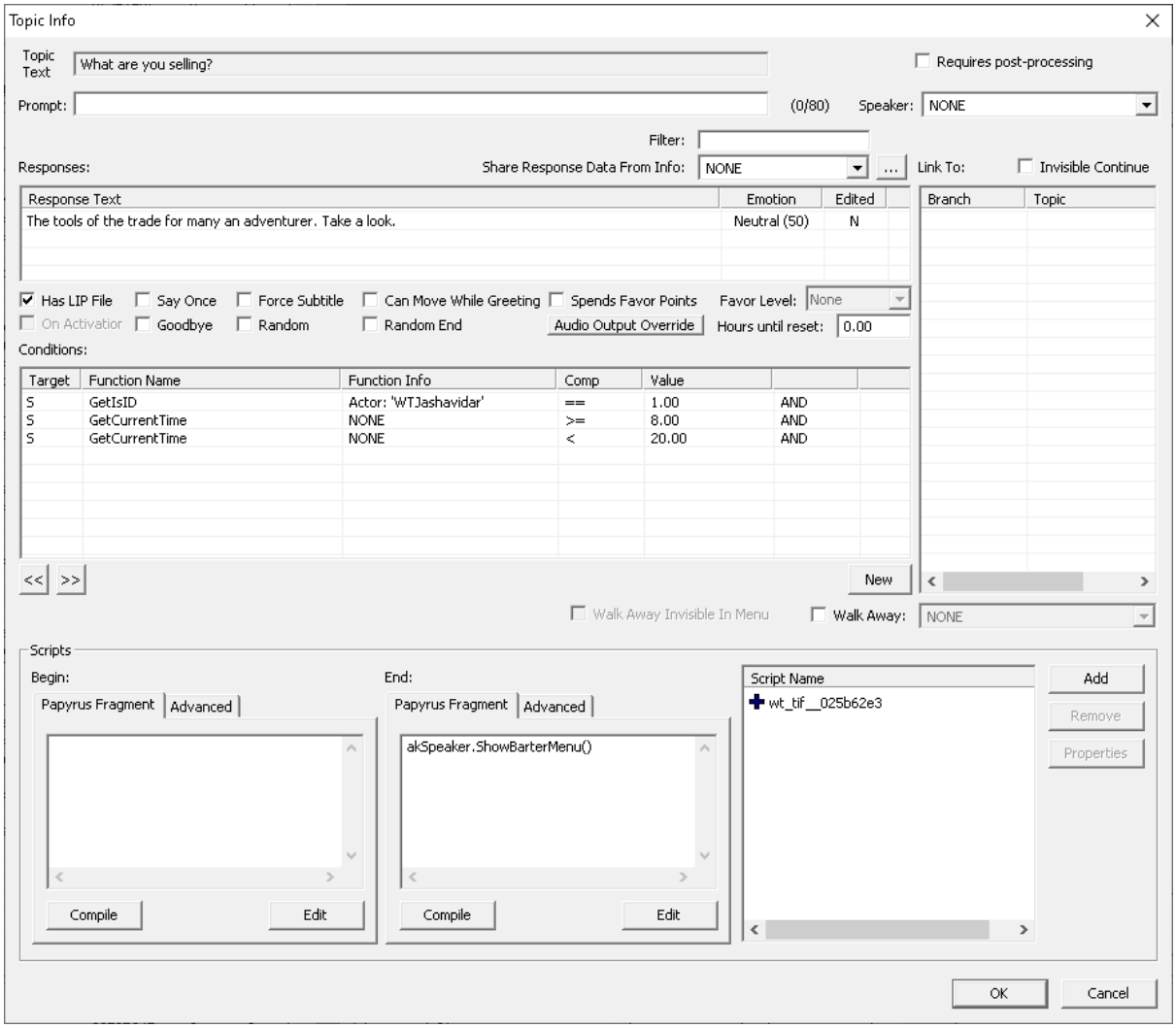


Figure 506 - Custom merchant dialogue.

Just make sure to add GetCurrentTime conditions to match the business hours set up in the service faction so this dialogue option doesn't appear outside of business hours.

For more information setting up quest dialogue, see the section Adding dialogue to a quest.

MAKING AN INNKEEPER

In this section I'll be covering the process of turning an NPC into an innkeeper that you can rent a bed from using the existing innkeeper system from the base game.

I'll be using the NPC we created in the previous section for this example.

The first thing we need to do is add our NPC to the JobRentRoomFaction faction.

In the Object Window, go to Actors > Actor > Actor and double-click or right-click on the NPC and select Edit.

Go to the Factions tab, right-click in the factions list and select New.

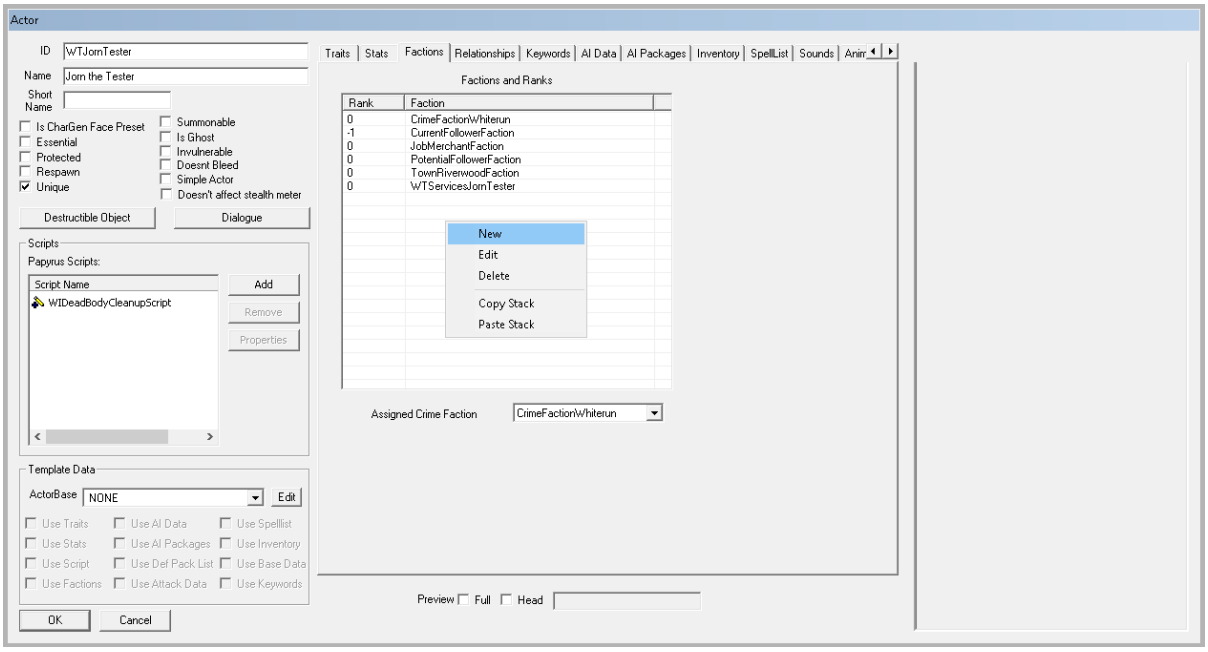


Figure 507 - Adding a new faction to Jorn.

Filter by 'rentroom', click on JobRentRoomFaction to highlight it then click OK.

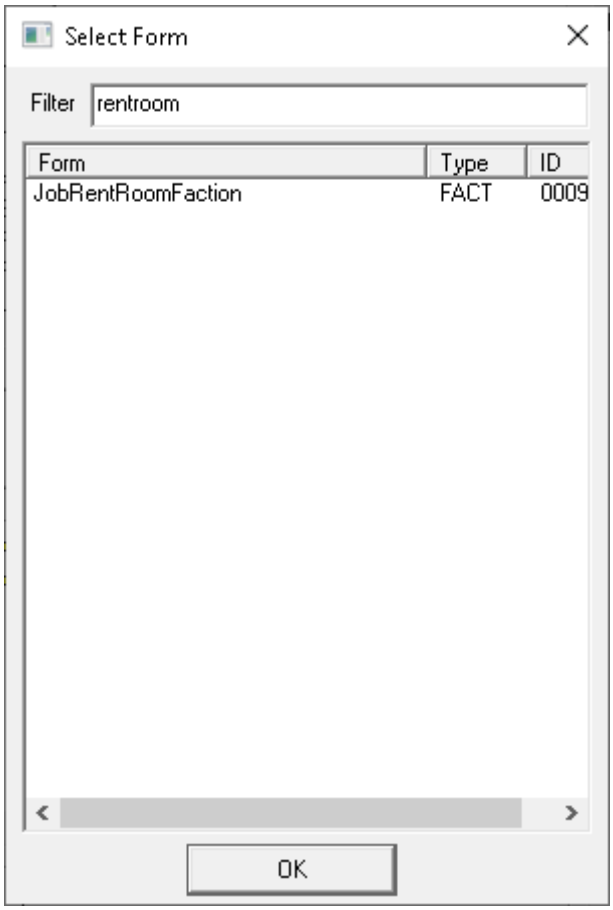


Figure 508 - Selecting the JobRentRoomFaction faction.

Next, we need to add the RentRoomScript script to the NPC. To do this, click on the Add button in the Papyrus Scripts section.

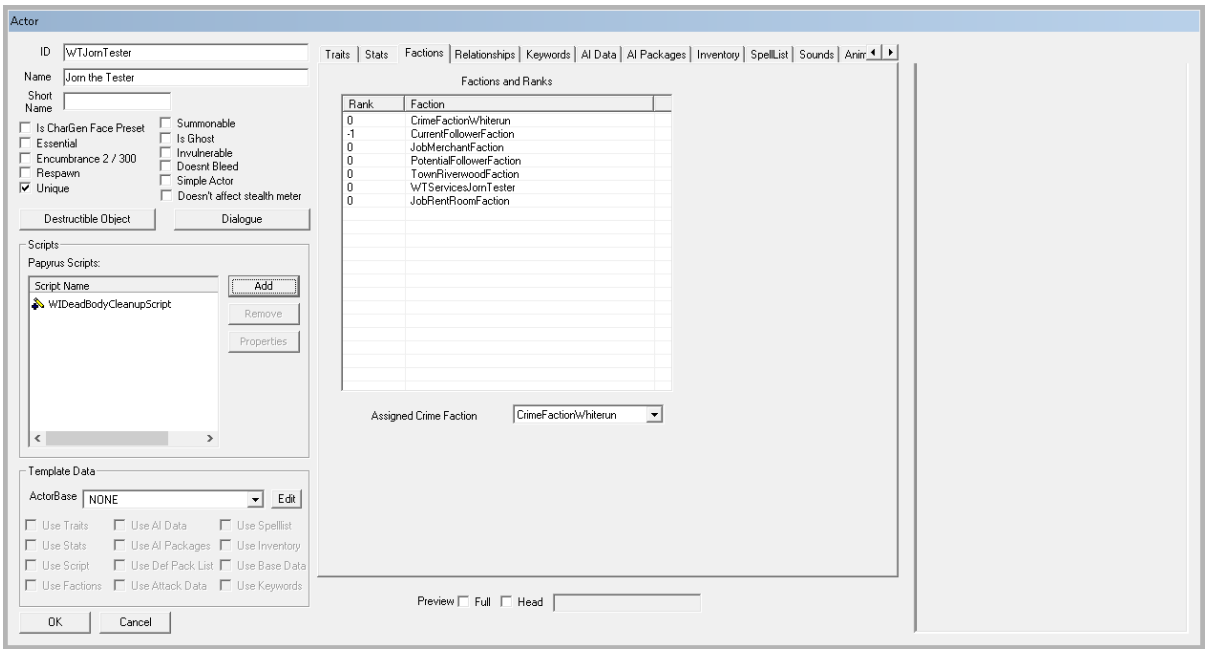


Figure 509 - Adding a new papyrus script.

Filter by 'rentroom', click on RentRoomScript to highlight it then click OK.

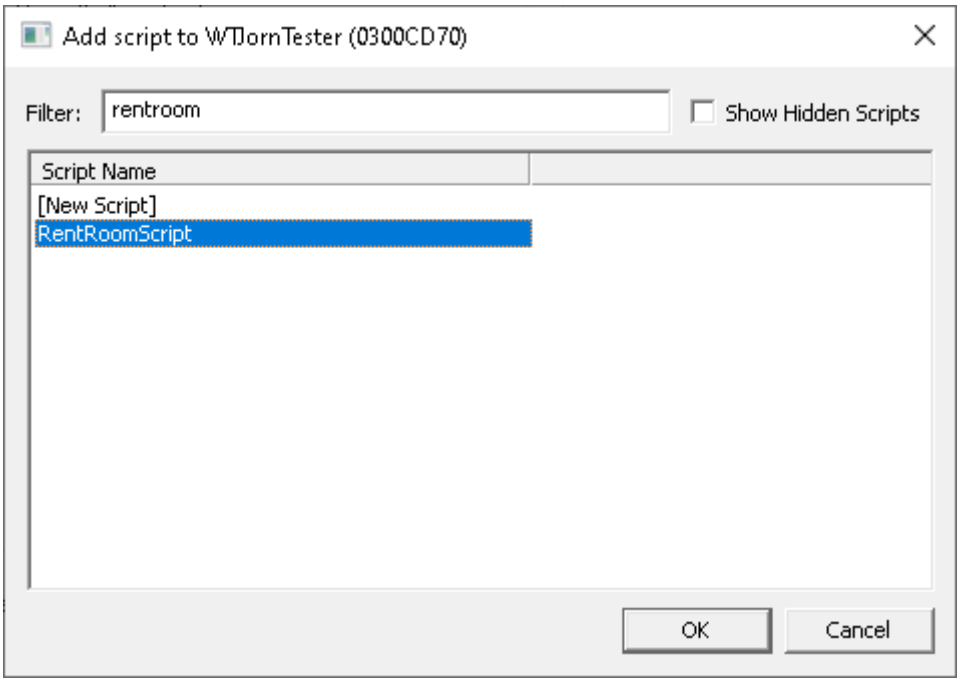


Figure 510 - Selecting the RentRoomScript script.

Click on the RentRoomScript script in the Papyrus Scripts section to highlight it then click on the Properties button.

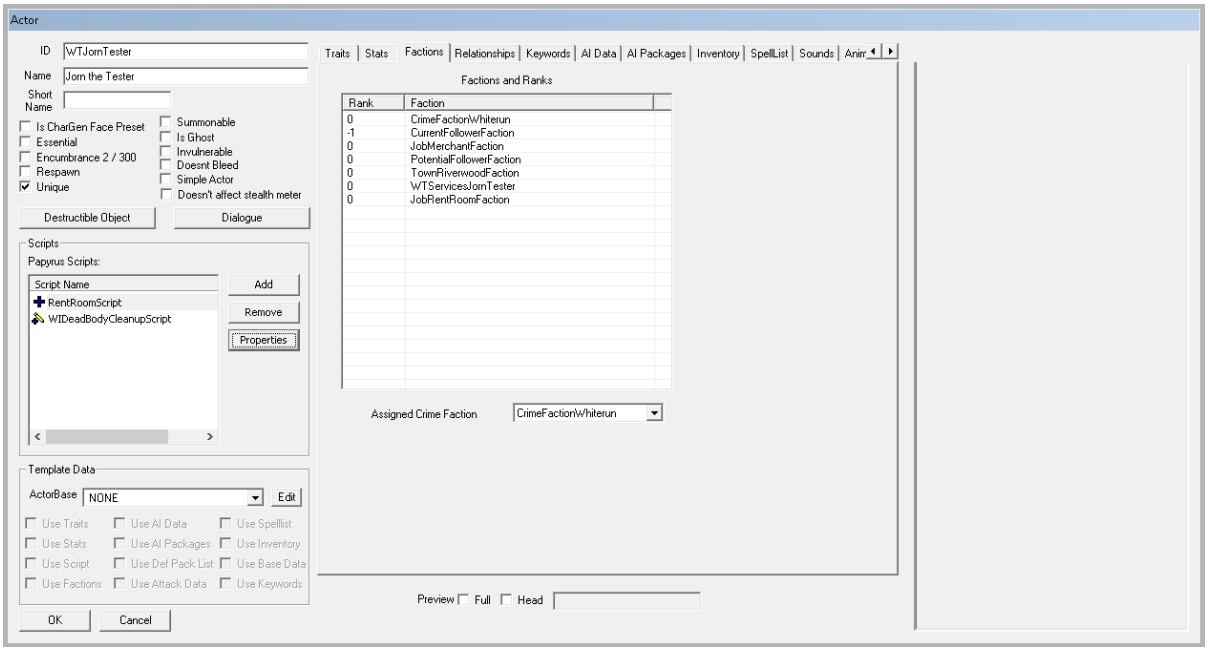


Figure 511 - Opening RentRoomScript script properties.

In the properties list, click on Bed to highlight it then click on the Edit Value button to the right.

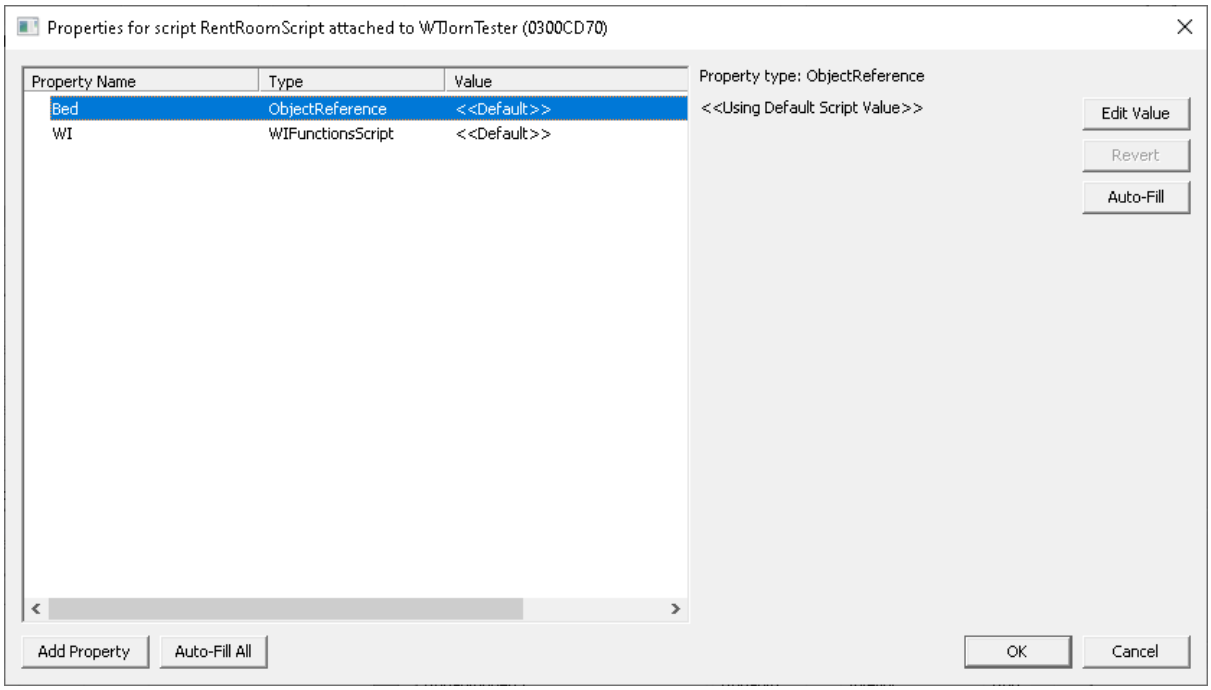


Figure 512 - Changing Bed properties.

Click on the 'Pick Reference in Render Window' button and select the bed in the render window that you want this NPC to rent out. This is the bed that the player will be allowed to sleep in after paying for a room.

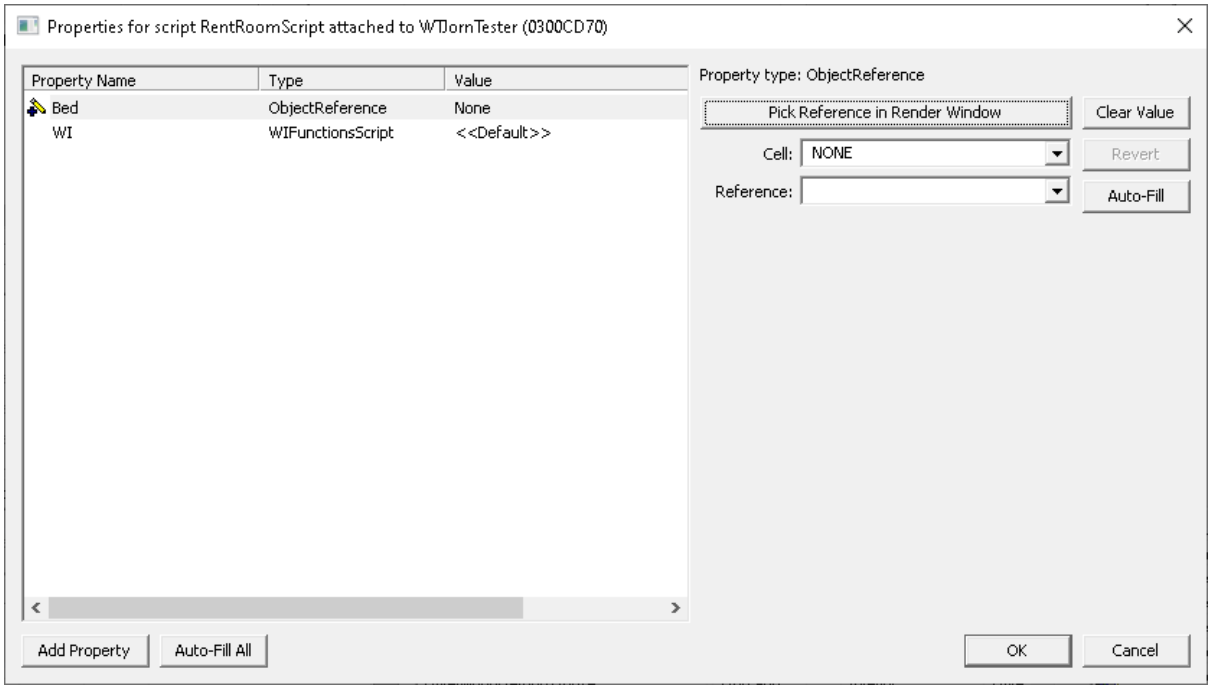


Figure 513 - Selecting a bed to rent out.

Click on the Auto-Fill All button to automatically add a value to the WI property.

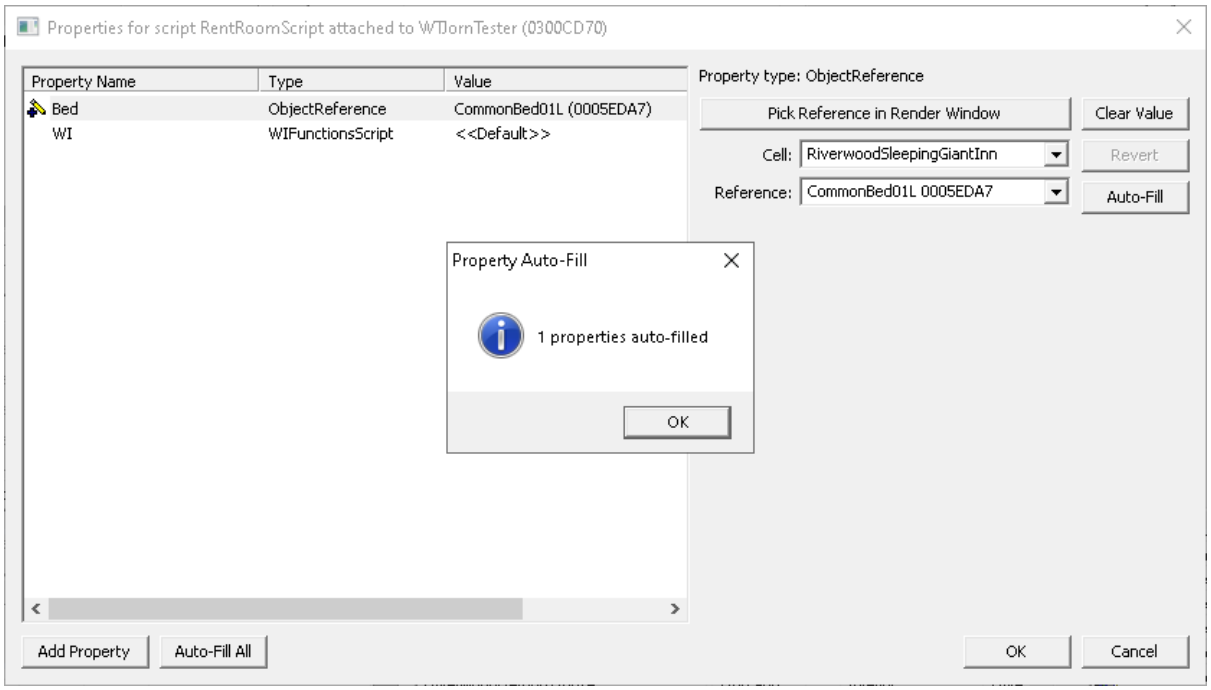


Figure 514 - Automatically adding a value to the WI property.

Click OK to close out of script properties.

Click OK to close out of actor properties.

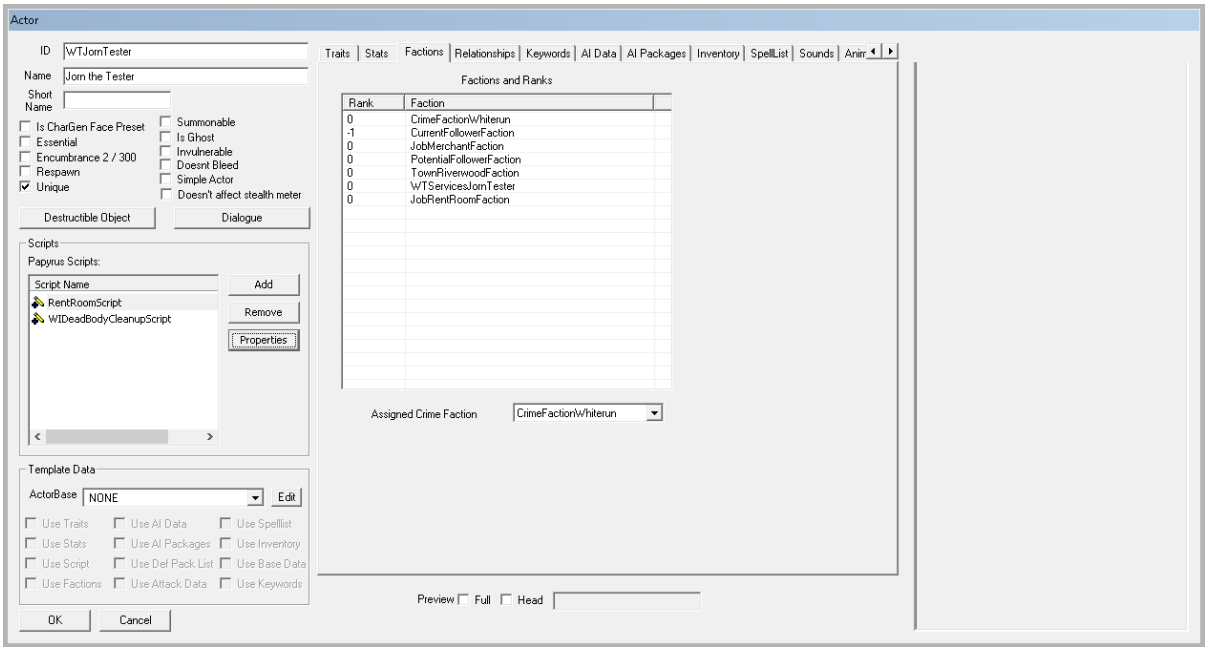


Figure 515 - Our NPC set up as an innkeeper.

If we speak to our NPC in-game, we should now be able to rent a room from them.



Figure 516 - Renting a room at the Sleeping Giant Inn from Jorn the Tester.

Jorn will also walk us over to the bed we rented.



Figure 517 - Jorn showing us to our bed.

If you're using a custom voice acted NPC, you'll need to manually remove the player's gold from their inventory and run a scene to make the NPC walk to the bed that the player has rented.

The example below is from Sigur, the innkeeper of The Hermit from Wyrmstooth.

```
game.GetPlayer().RemoveItem(GoldBase, 10)
WTInnkeeperWalkToScene.Start()
```

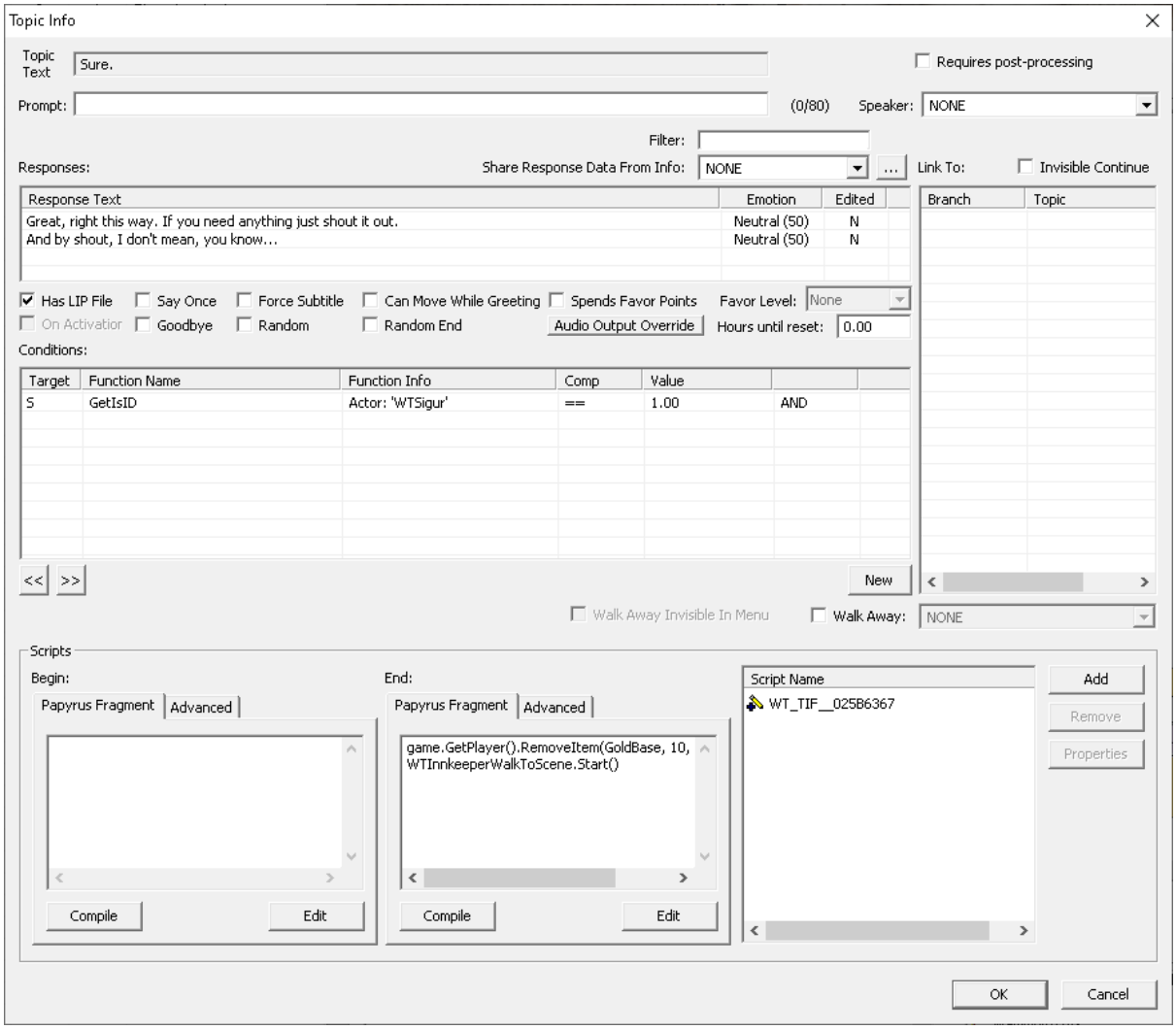


Figure 518 - Innkeeper with custom voice acted dialogue.

The script properties on the papyrus fragment:

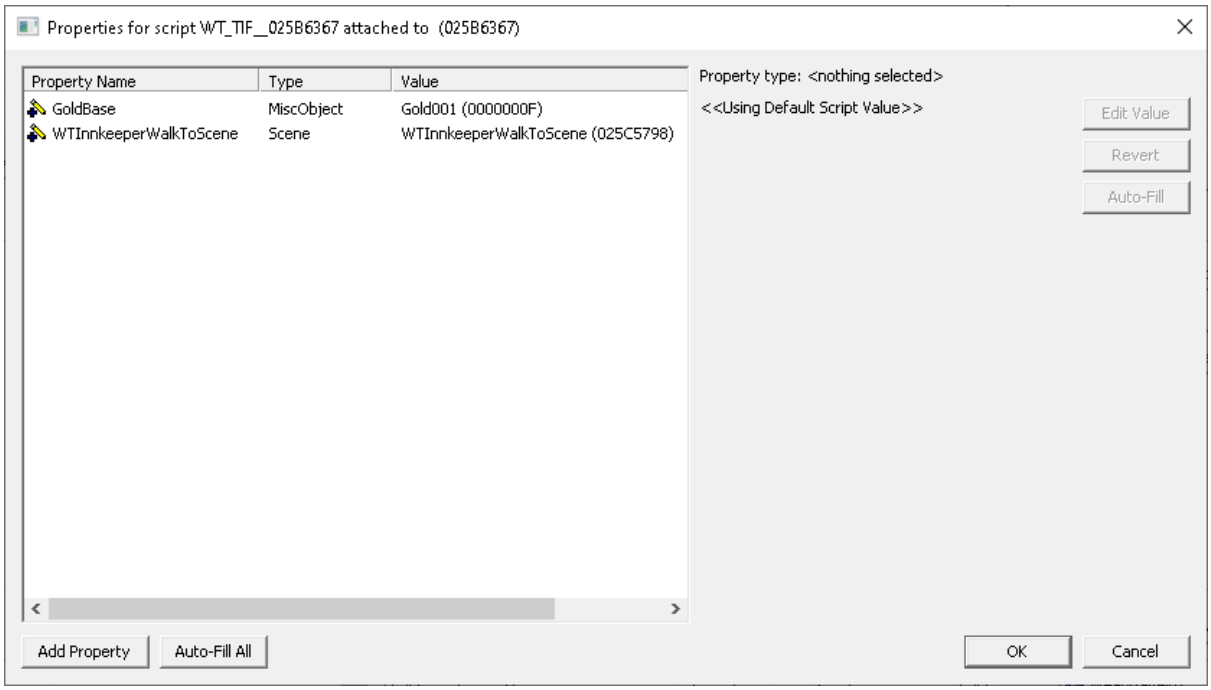


Figure 519 - Papyrus fragment script properties.

The WTInnkeeperWalkToScene scene basically just walks Sigur over to the bed in Phase 1, then back to the bar in Phase 2.

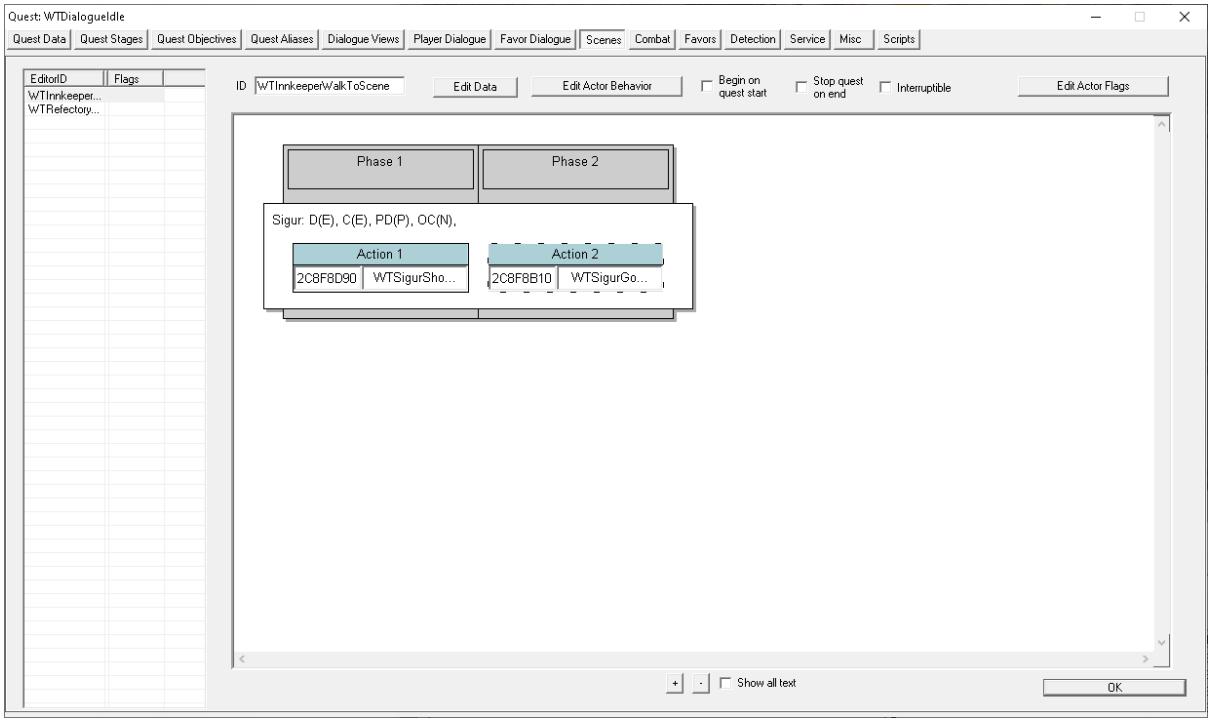


Figure 520 - WTInnkeeperWalkToScene scene set up.

For more information on how to set up scenes, see the section on Adding scenes.

For more information setting up quest dialogue, see the section Adding dialogue to a quest.

MAKING A BARD

This section will cover setting an NPC up as a bard. Like in the previous sections, I'll be using the Jorn the Tester NPC for this example. Maybe he's found a new passion and decided to pursue the bardic arts.

The first thing we need to do is add our NPC to the bard factions.

In the Object Window, go to Actors > Actor > Actor, double-click or right-click on the NPC and select Edit.

Go to the Factions tab, right-click in the factions list and select New.

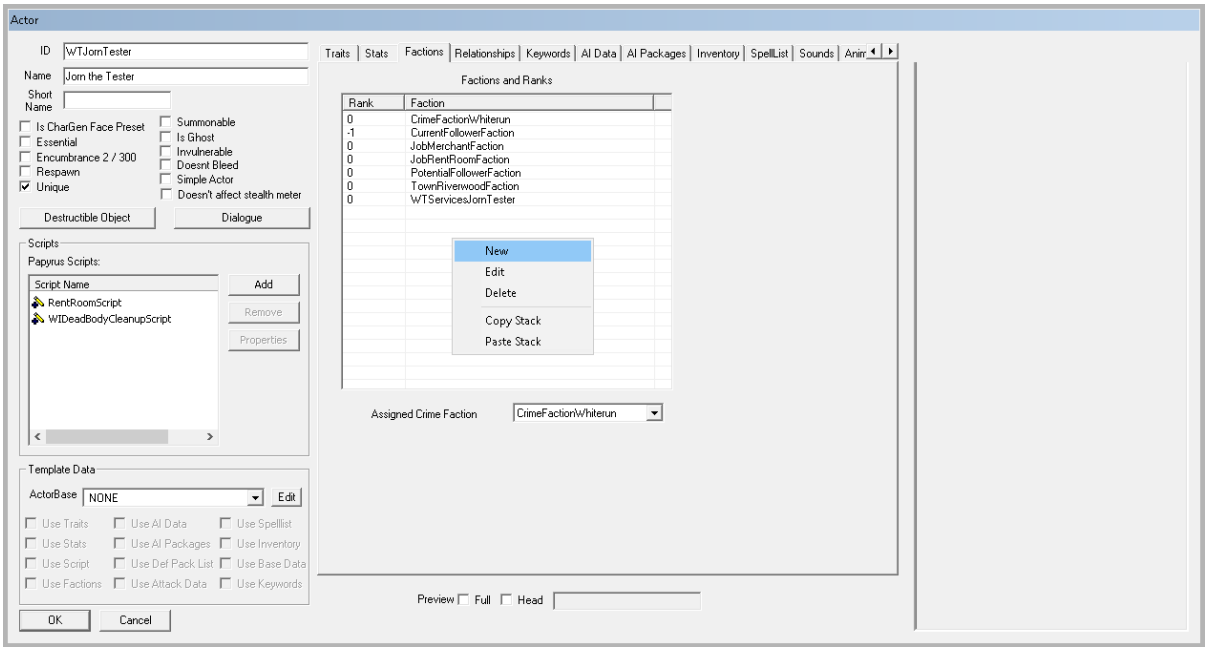


Figure 521 - Adding the first bard faction.

Filter by 'jobbard', click on JobBardFaction to highlight it then click OK.

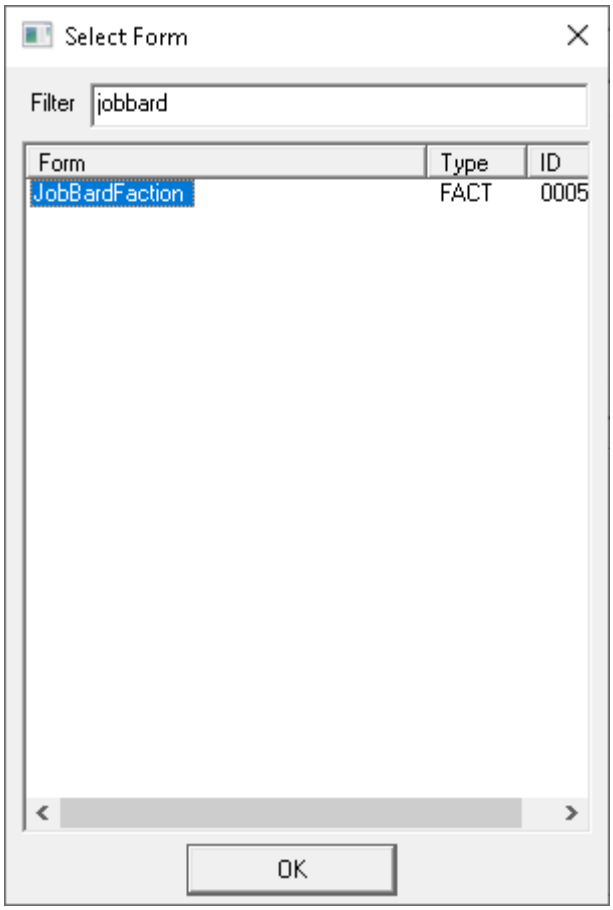


Figure 522 - Selecting the JobBardFaction faction.

Right-click in the faction list again and select New.

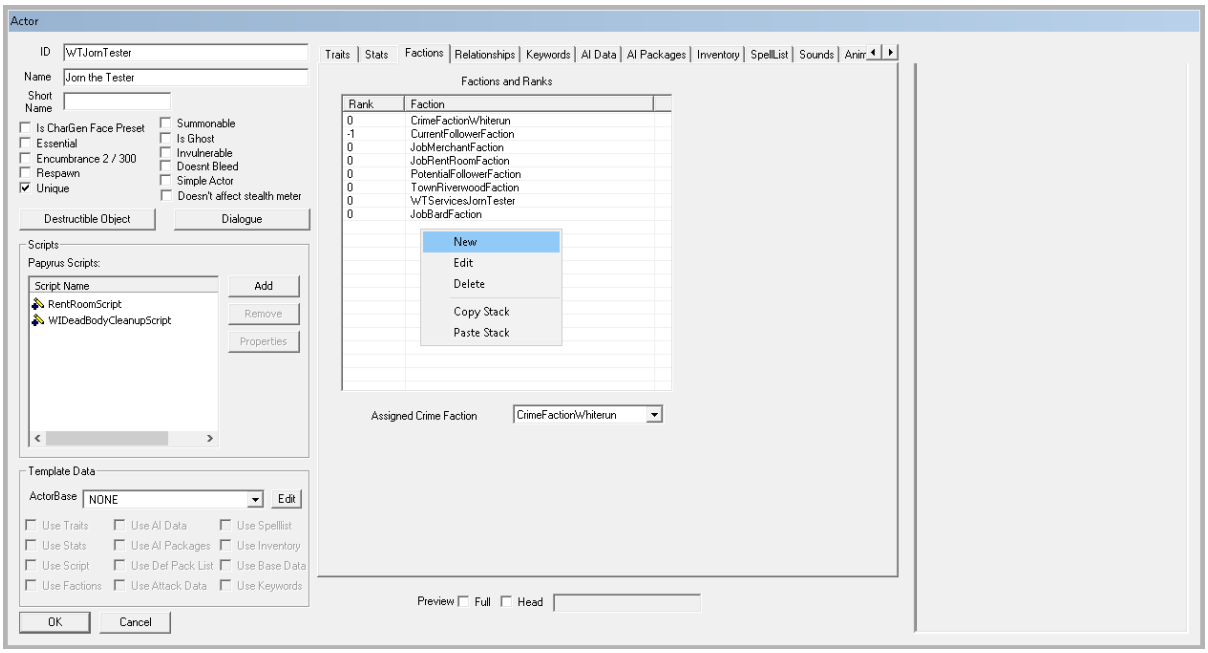


Figure 523 - Adding the second bard faction.

Filter by 'bardsinger', click on BardSingerFaction to highlight it then click OK. If you want your bard to only provide instrumental performances, select BardSingerInstrumentalOnlyFaction instead.

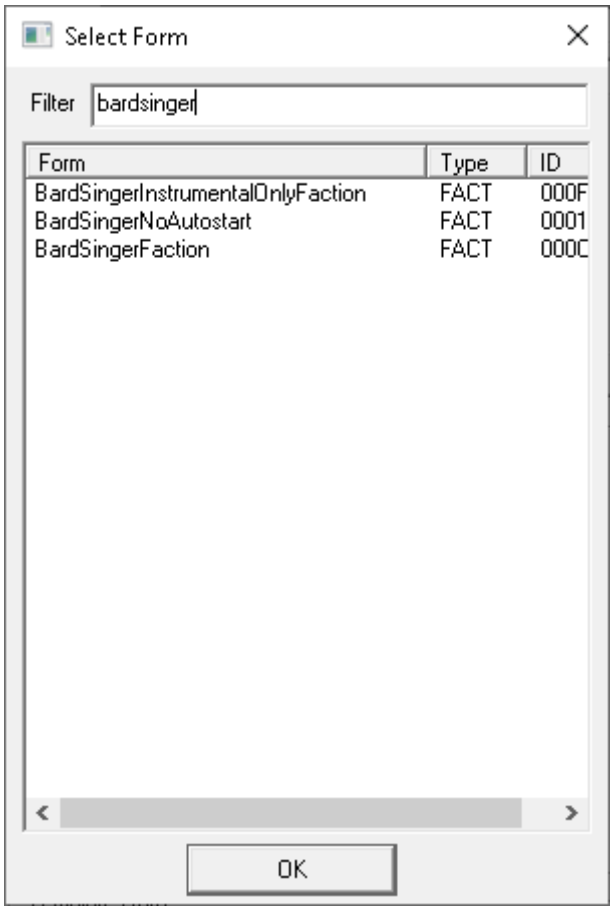


Figure 524 - Bard singer factions.

Our NPC should now be in the JobBardFaction and BardSingerFaction factions.

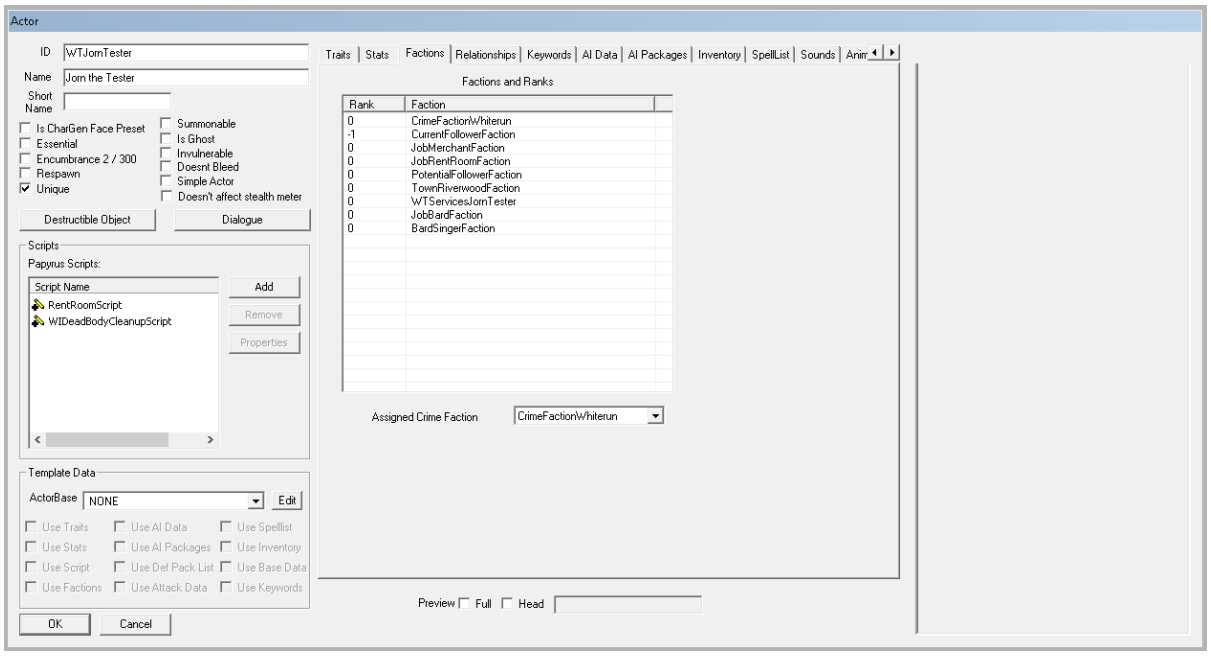


Figure 525 - JobBardFaction and BardSingerFaction factions added.

Click OK to close out of Actor properties.

Double-click or right-click on the NPC in the render window and select Edit.

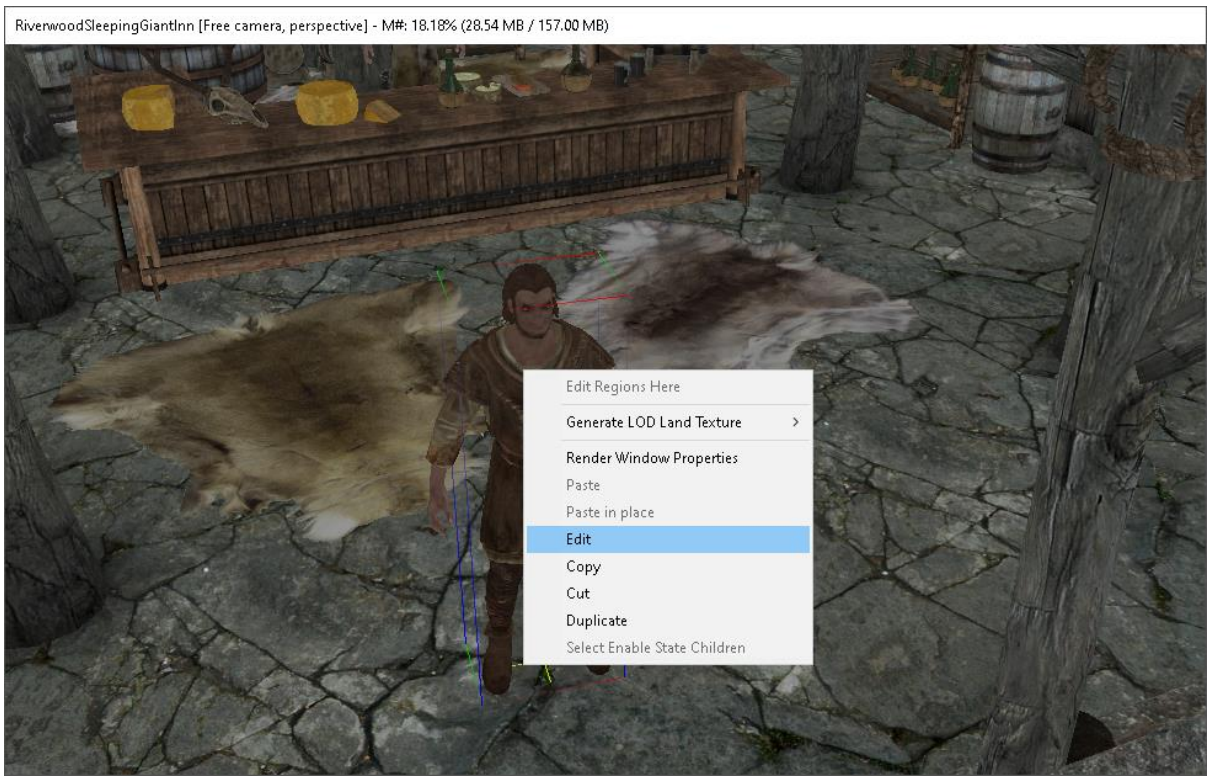


Figure 526 - Opening the properties of the NPC reference.

Go to the Linked Ref tab, right-click in the references list and select New.

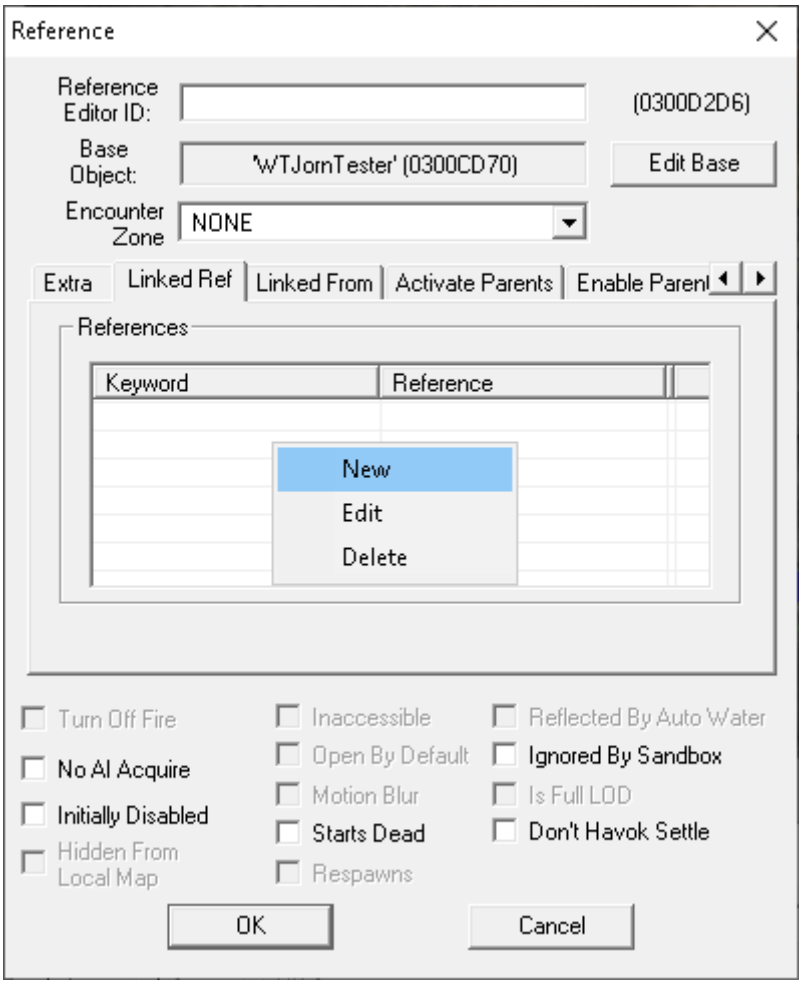


Figure 527 - Adding a new linked reference.

Click on the 'Select Reference in Render Window' button and select an XMarker to link the NPC to.

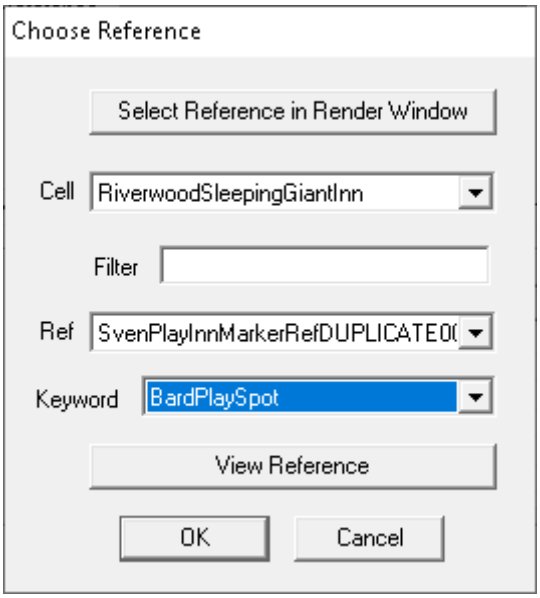


Figure 528 - Selecting the linked reference.

For this example I chose the same XMarkerHeading that Sven has been linked to.

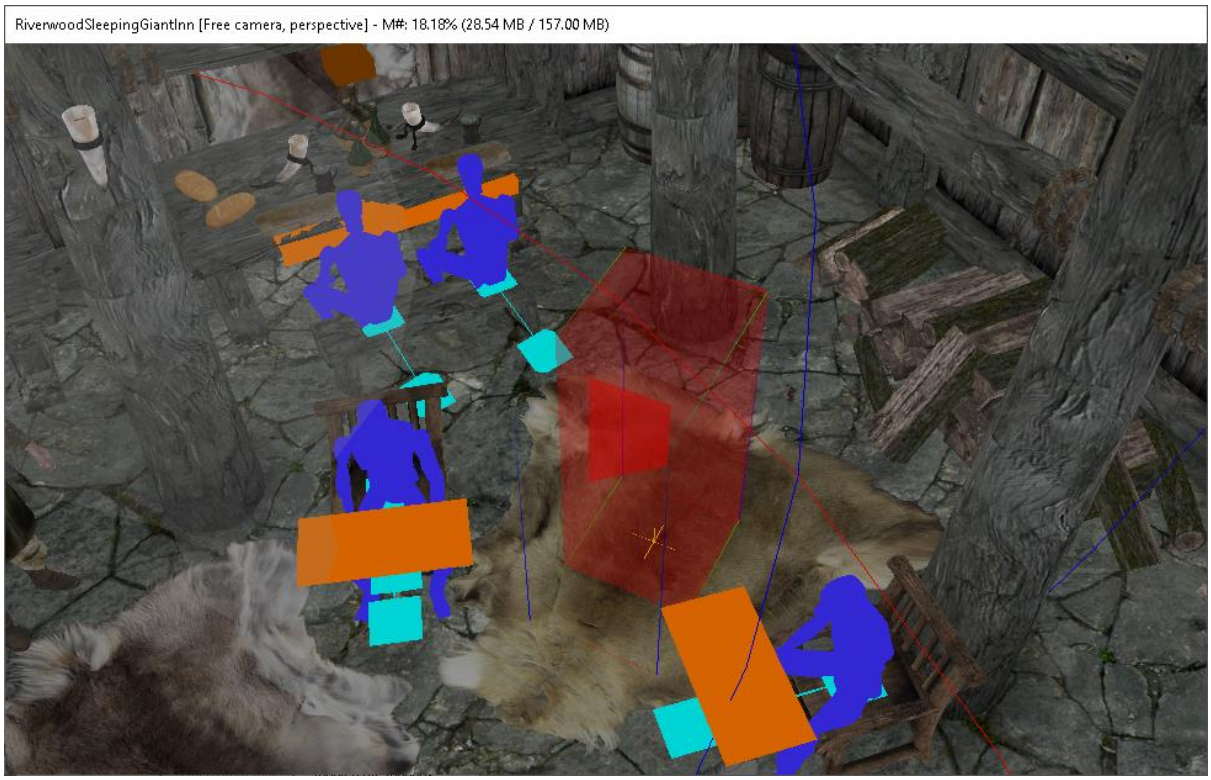


Figure 529 - The linked reference.

Make sure you set the Keyword drop-down to BardPlaySpot.

Click OK to close out of Reference properties.

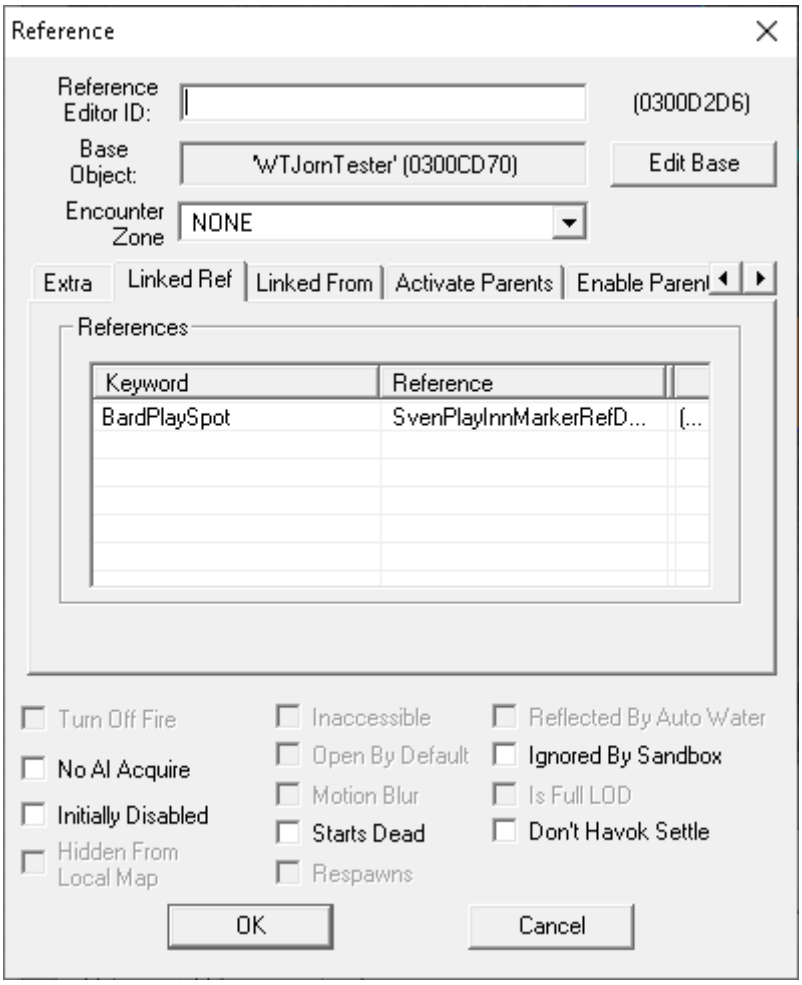


Figure 530 - NPC linked to an XMarkerHeading.

Lastly, we need to set up a package for our NPC.

In the Object Window, go back to Actors > Actor > Actor, double-click or right-click on the NPC and select Edit.

This time go to the AI Packages tab, right-click in the packages list and select New.

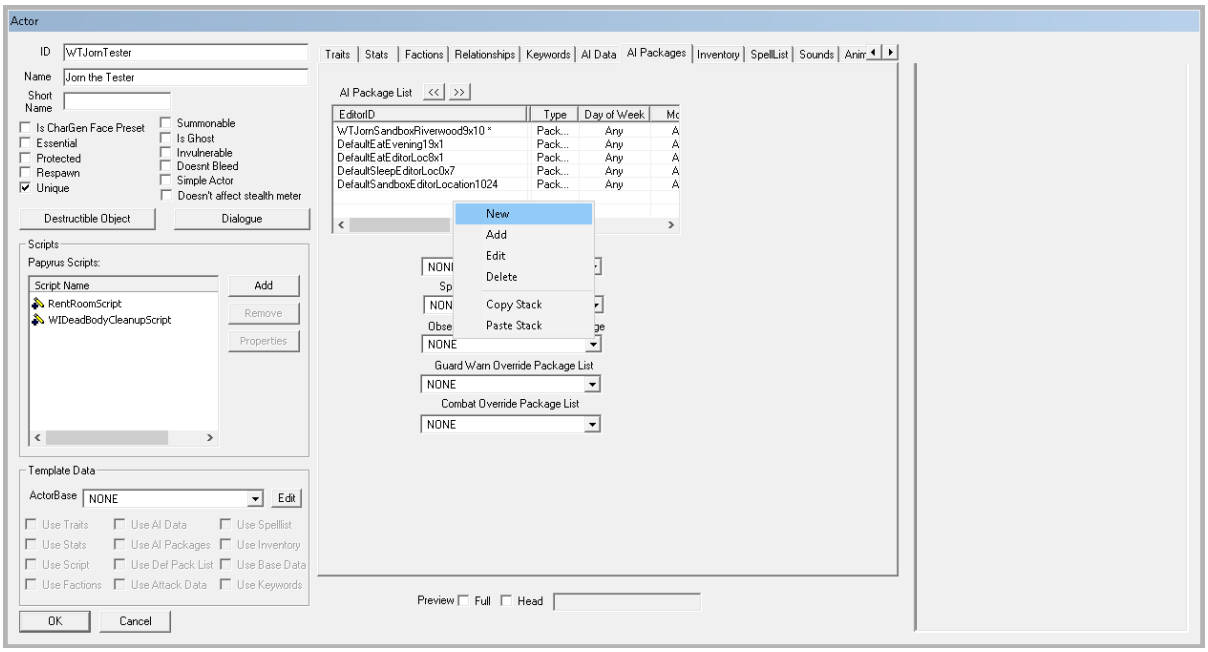


Figure 531 - Adding a new package.

Set the package ID. For this example I set it to WTJornBardPerformance9x10. Set the Package Template drop-down to Sandbox.

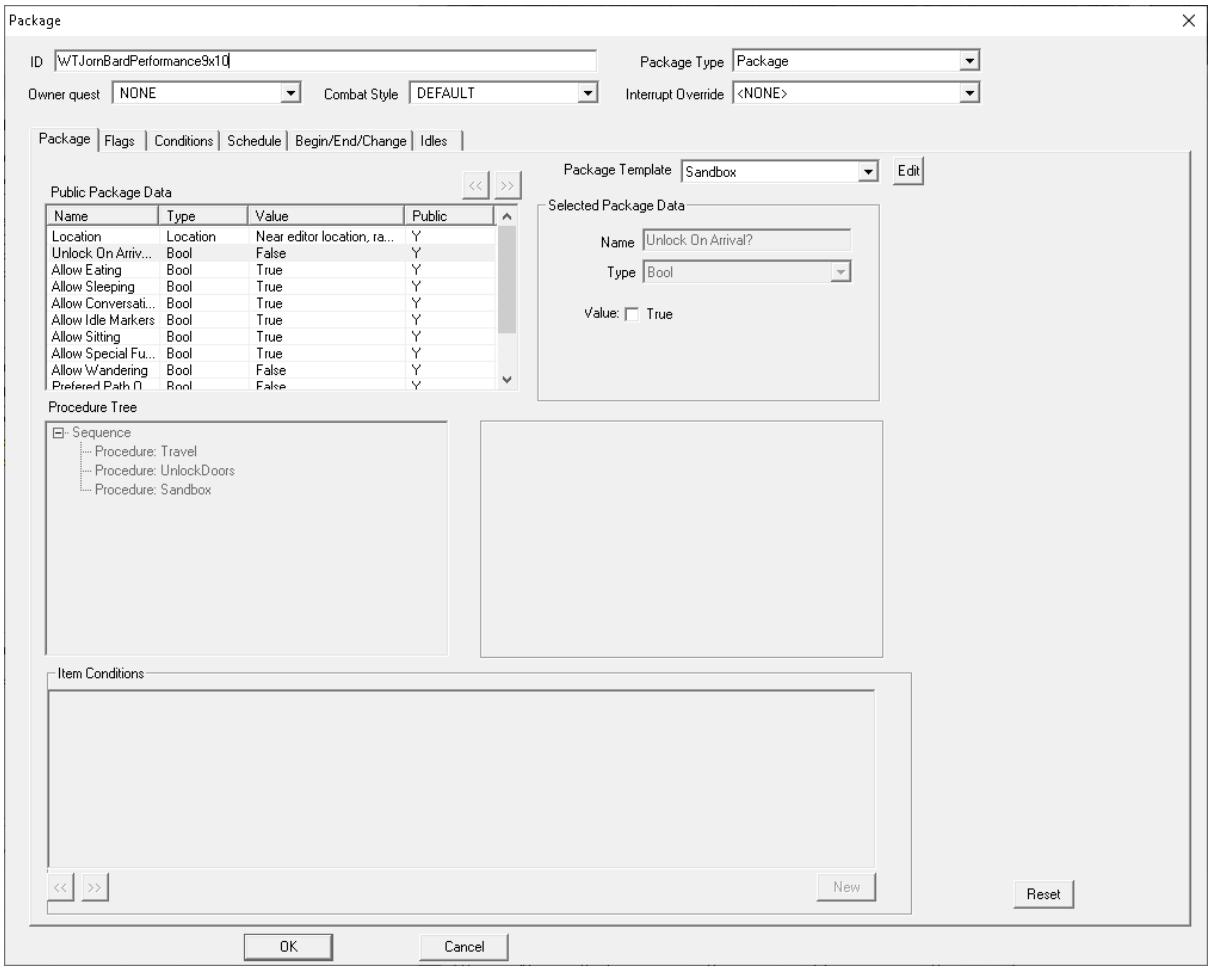


Figure 532 - Package tab.

In Public Package Data, click on Location. On the right hand side, set the Radius field to 0 then click on the button that currently reads 'Near editor location, radius 512'.

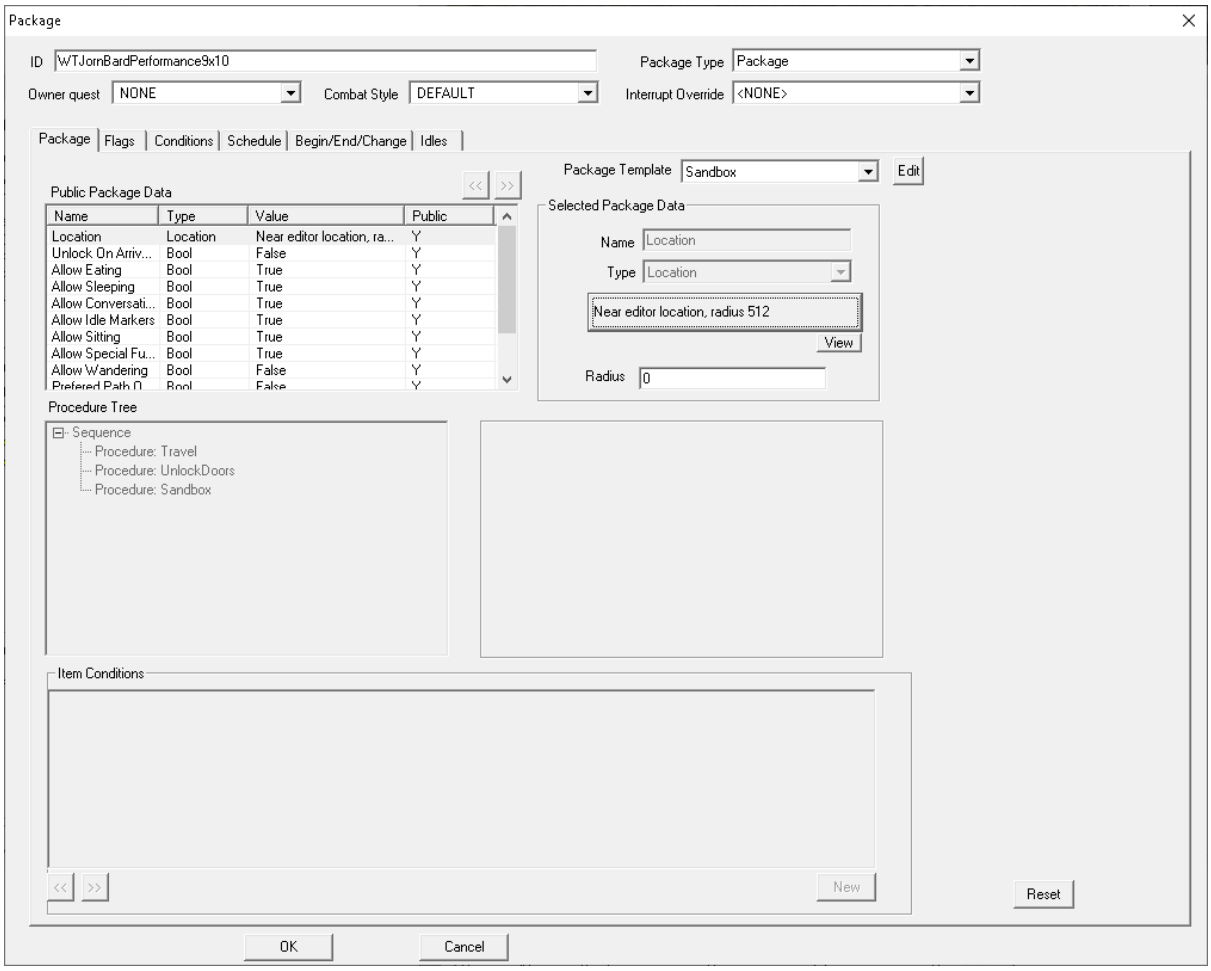


Figure 533 - Setting the sandbox location.

Select Near Reference, click on the Select Reference button, select the XMarker you linked the NPC to earlier and click OK.

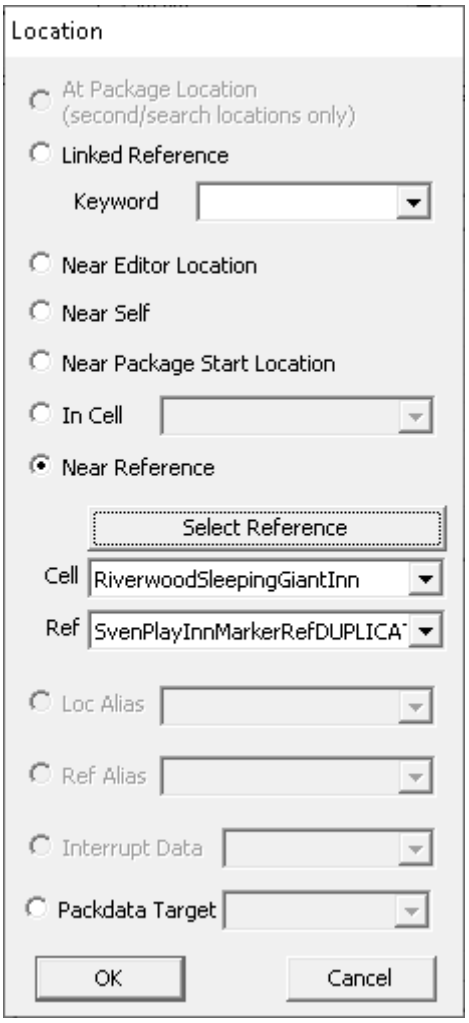


Figure 534 - Setting the sandbox package center location.

Go to the Schedule tab. I want Jorn to perform as a bard starting from 9:00 AM until 7:00 PM. Set the Hour drop-down to 9 and the Duration field to 10.0000.

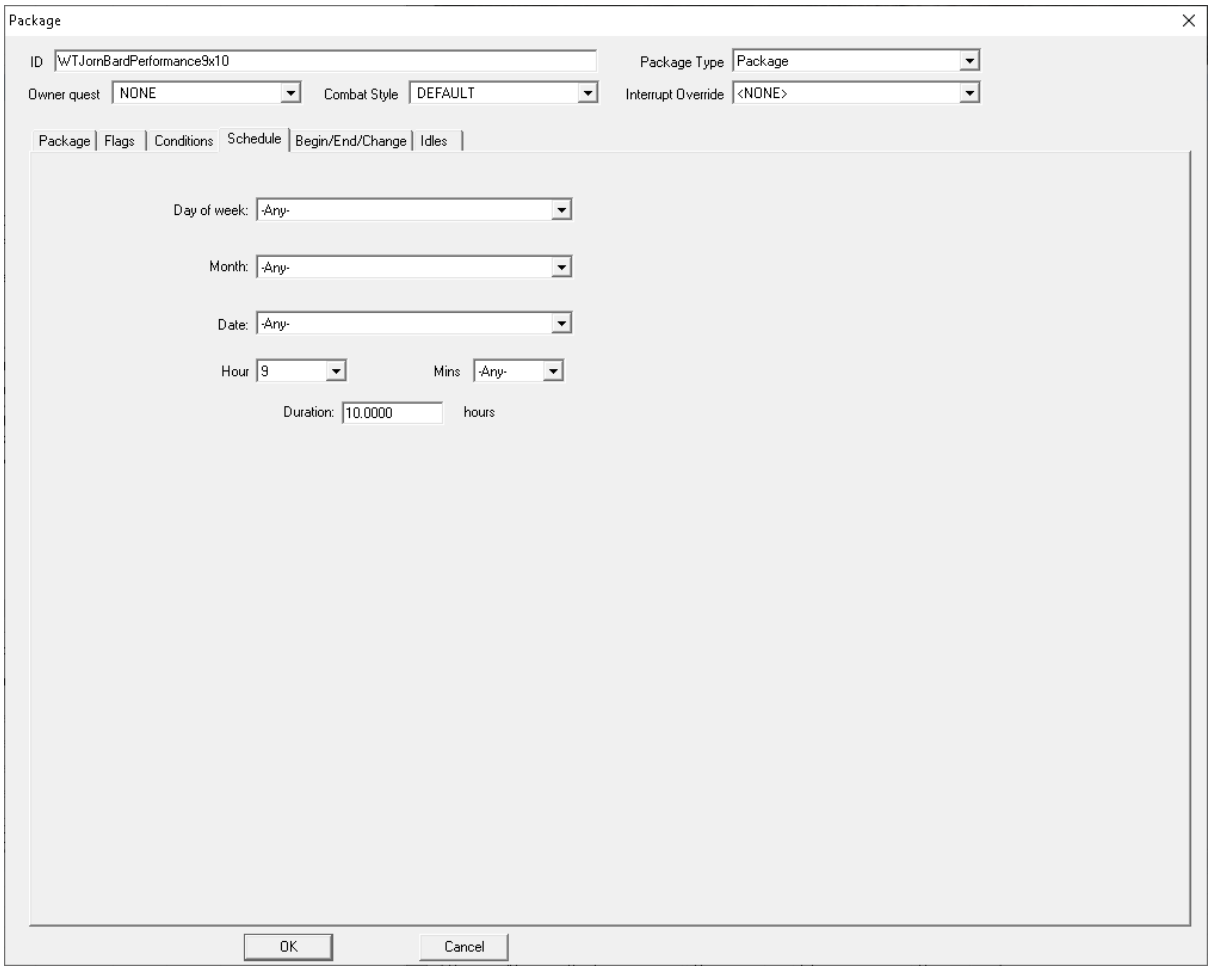


Figure 535 - Setting sandbox duration.

Click OK to close out of Package properties.

Right-click on the old sandbox package that we gave Jorn earlier (WTJornSandboxRiverwood9x10) and select Delete to remove it from the packages list.

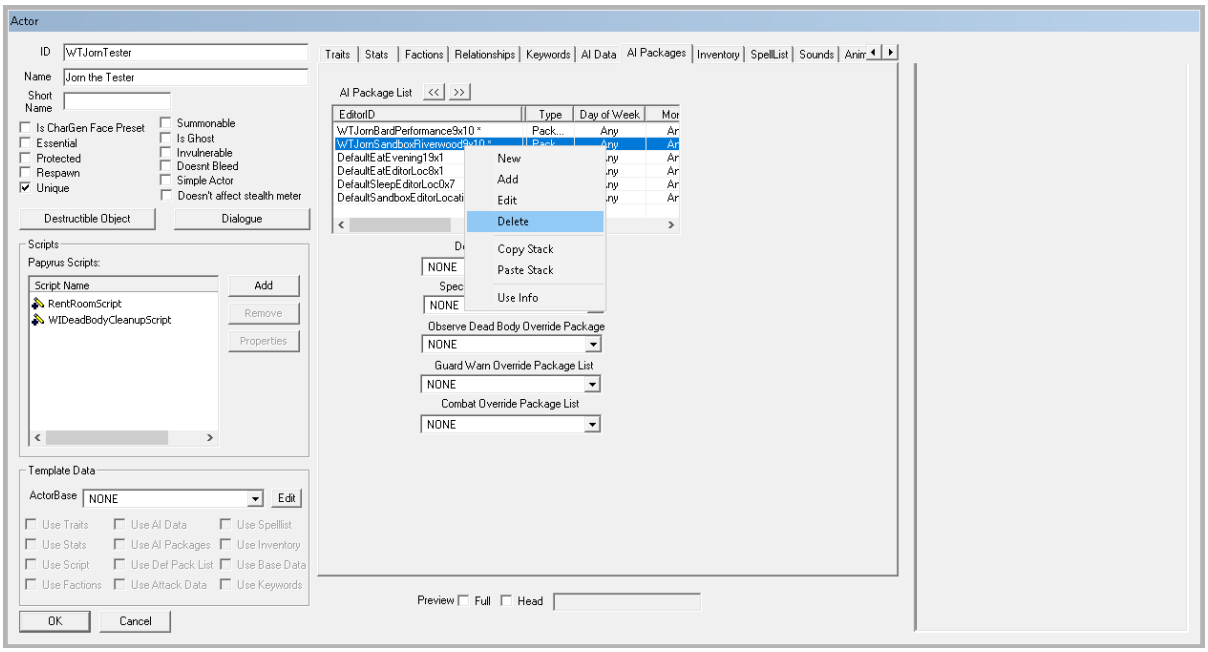


Figure 536 - Removing the old sandbox package.

Basically we want Jorn to spend his day performing at the inn instead of walking around Riverwood.

Click OK to close out of Actor properties.

Important: Bards will only perform if the performance location has the **LocTypeInn** keyword.

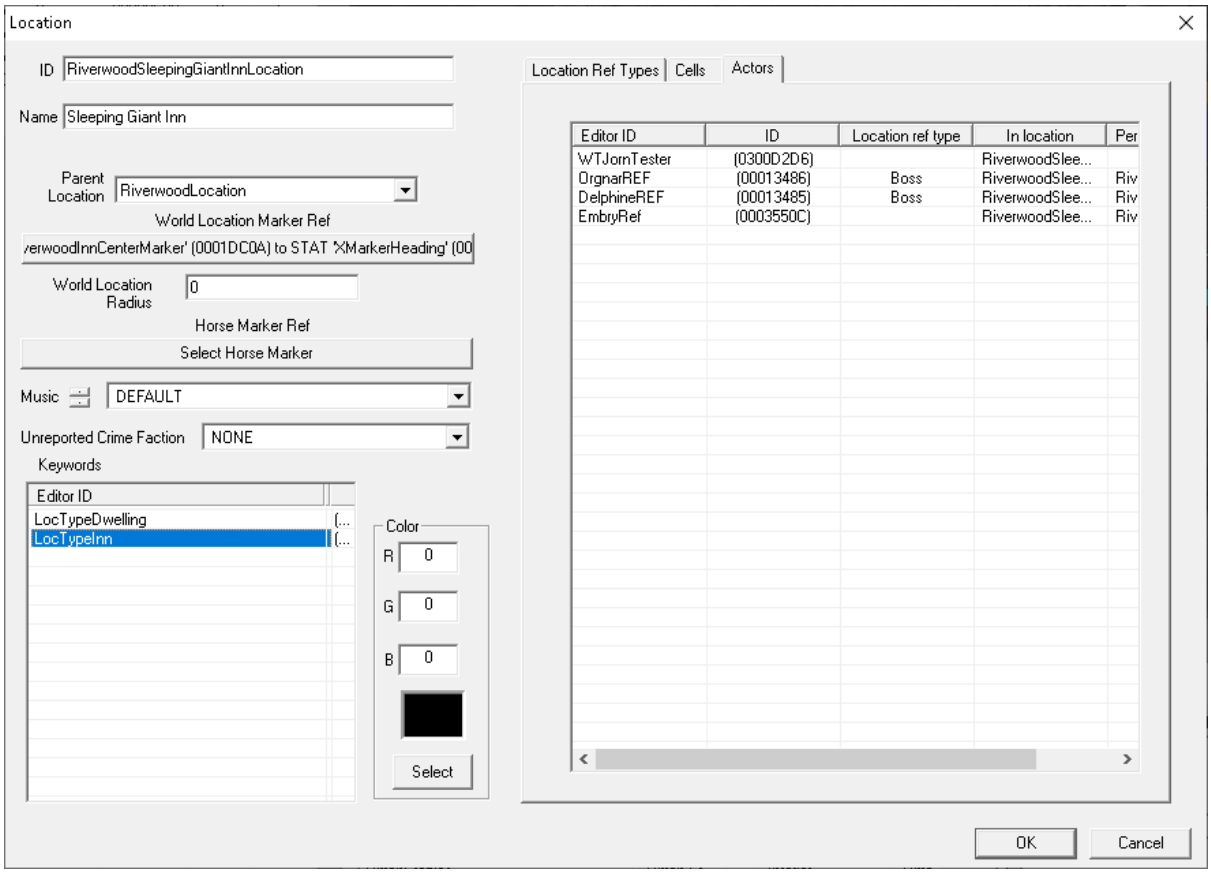


Figure 537 - Riverwood Sleeping Giant Inn location properties.

In-game we'll be able to see Jorn perform during 9:00 AM to 7:00 PM.



Figure 538 - Jorn playing the flute.

If we speak to him, we'll be able to request a performance.



Figure 539 - Requesting a performance.

Setting up a bard with custom performances is a lot more complicated and may be the subject of a separate tutorial.

MAKING A FOLLOWER

In this section I'll turn Jorn into a basic follower. Since Jorn is using a base-game voice type, we can accomplish this pretty easily by adding him to the PotentialFollowerFaction and CurrentFollowerFaction factions.

In the Object Window, go to Actors > Actor > Actor, double-click or right-click on WTJornTester and select Edit.

Go to the Factions tab, right-click in the factions list and select New.

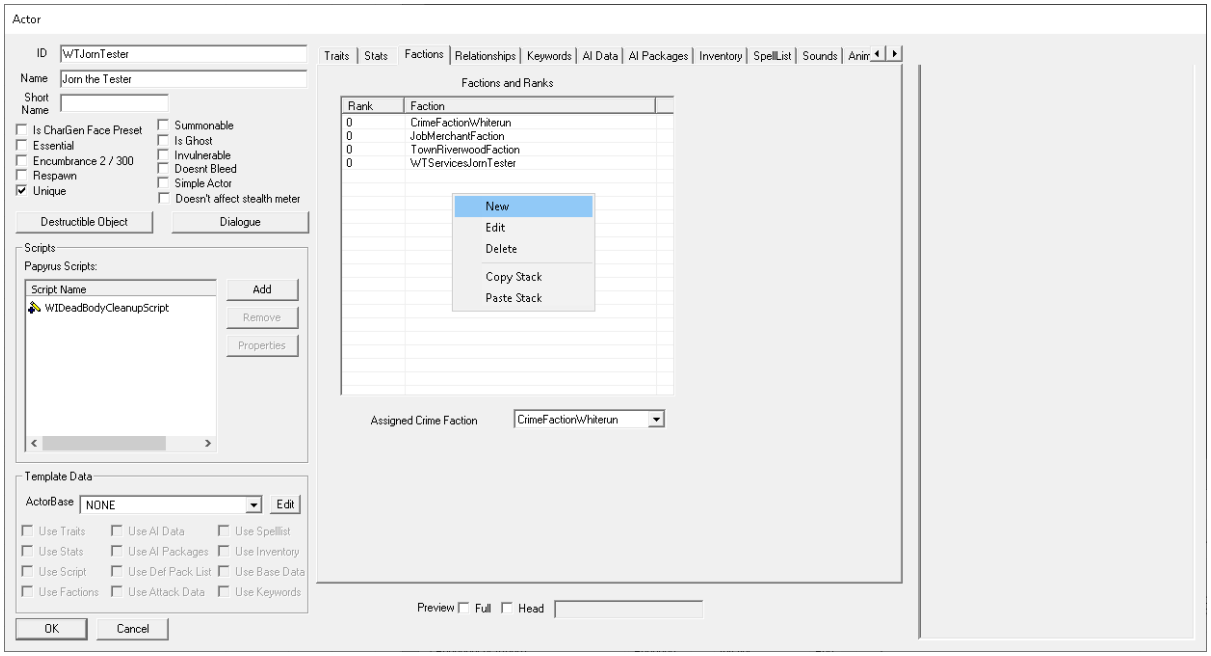


Figure 540 - Adding Jorn to PotentialFollowerFaction.

Filter by 'potentialfollower', click on PotentialFollowerFaction to highlight it then click OK.

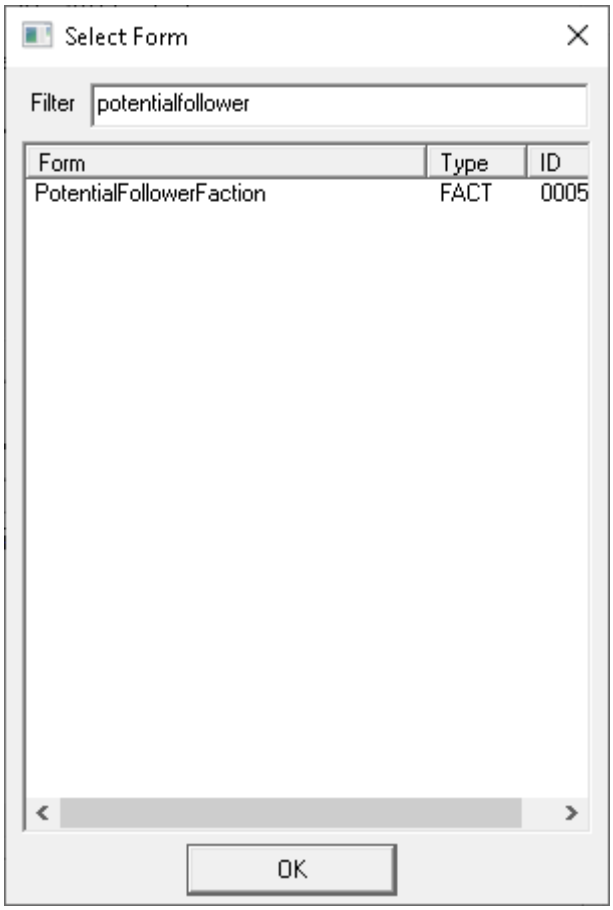


Figure 541 - Factions list.

Lastly, right-click again in the factions list and select New.

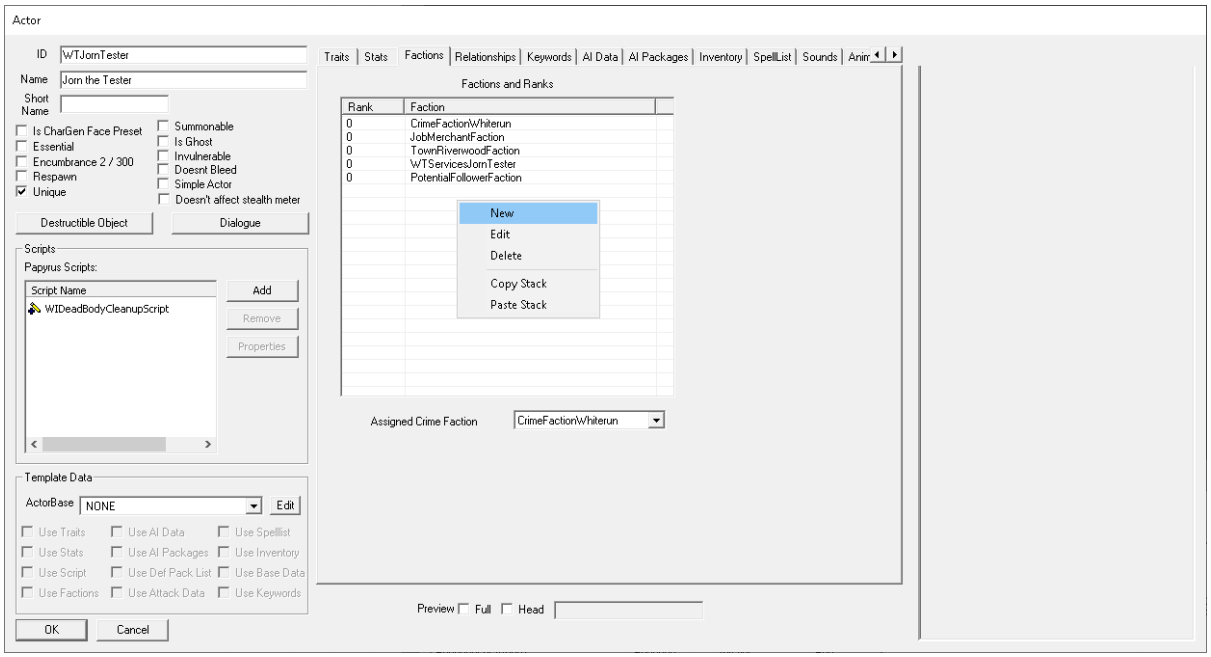


Figure 542 - Adding Jorn to CurrentFollowerFaction.

Filter by 'currentfollower', click on CurrentFollowerFaction to highlight it then click OK.

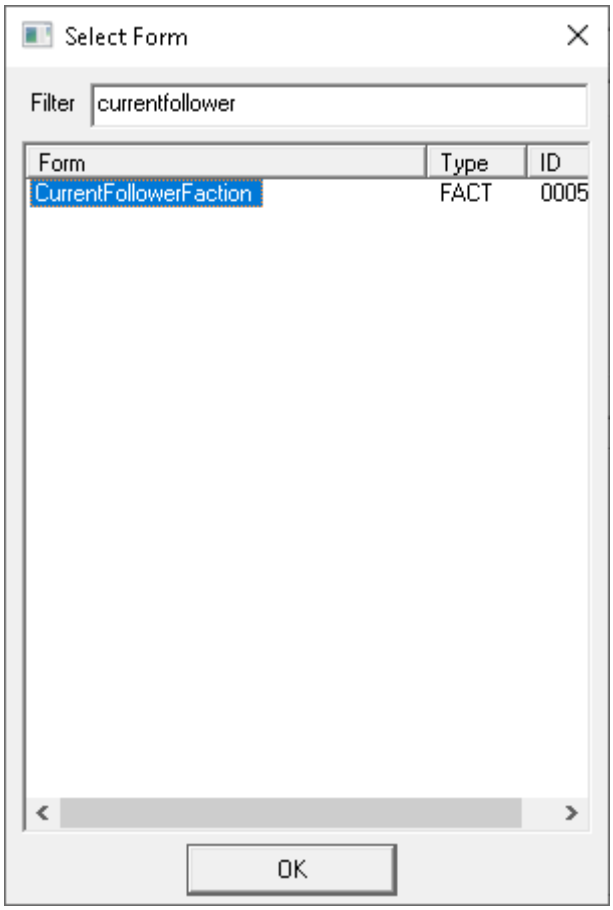


Figure 543 - Factions list.

Click in the Rank column next to CurrentFollowerFaction and set the value to -1.

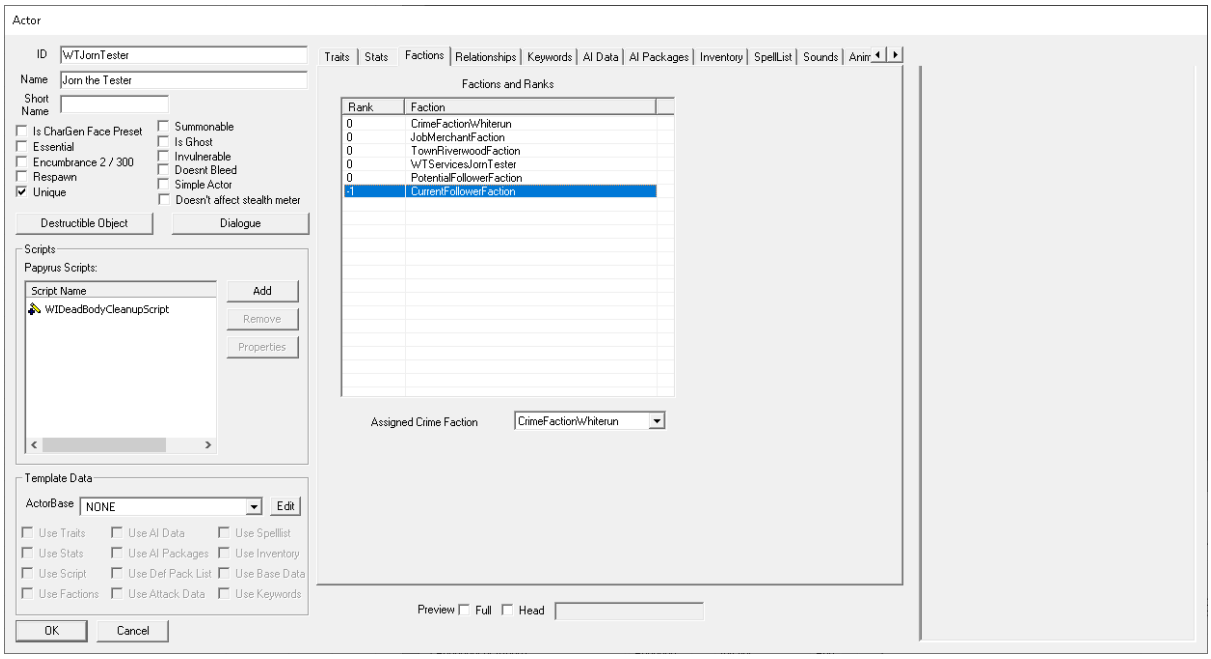


Figure 544 - Setting the faction rank of CurrentFollowerFaction.

Normally we need to do a favour for the NPC first before we can hire them, but we can increase our NPC's disposition towards the player with a relationship so that we can ask them to follow us immediately.

Go to the Relationship tab, right-click in the relationships list and select New.

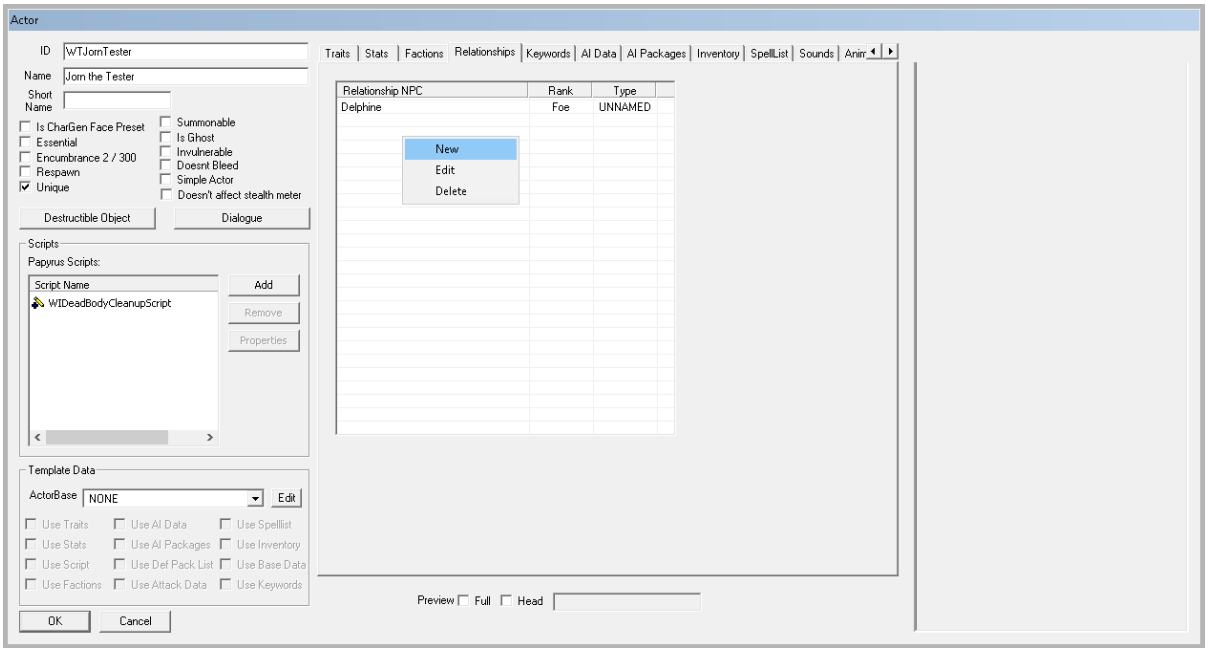


Figure 545 - Adding a new relationship to Jorn.

Set a unique ID. For this example I just went with WTJornPlayer. Make sure the Parent NPC drop-down is set to WTJornTester and set the Child NPC drop-down to Player. Finally, set the Relationship Level drop-down to Ally then click OK.

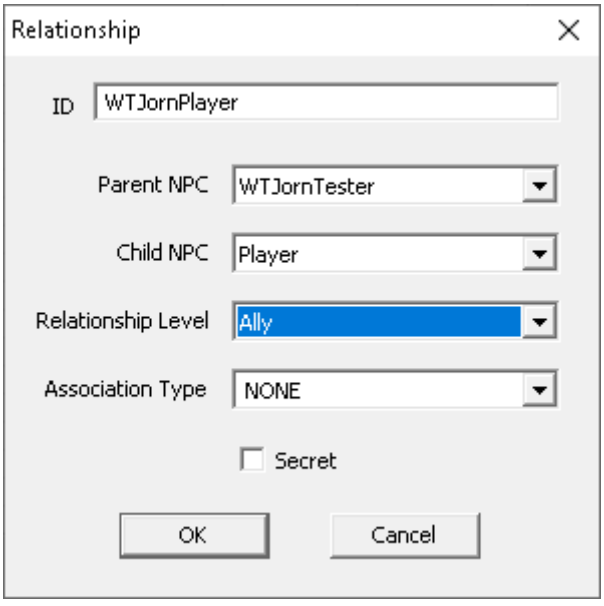


Figure 546 - Setting Jorn as the player's ally.

Click OK to close out of Actor properties.

If we go in-game, we should now be able to ask Jorn to follow us.



Figure 547 - Jorn set up as a follower.

If your NPC uses custom voice acting, we'll need to set up the various follower dialogue commands manually.

The following screenshots will show you examples from Athir's follower dialogue in Wyrmstooth.

Important: Add these topics to a quest that does not complete. If the quest containing these topics ever completes, these topics will become unavailable.

Firstly, here's how the 'I need your help, follow me' topic is set up. We can run the following papyrus to set this NPC as our current follower.

```
(self.GetOwningQuest() as
dialoguefollowerscript).SetFollower(akSpeaker)
```

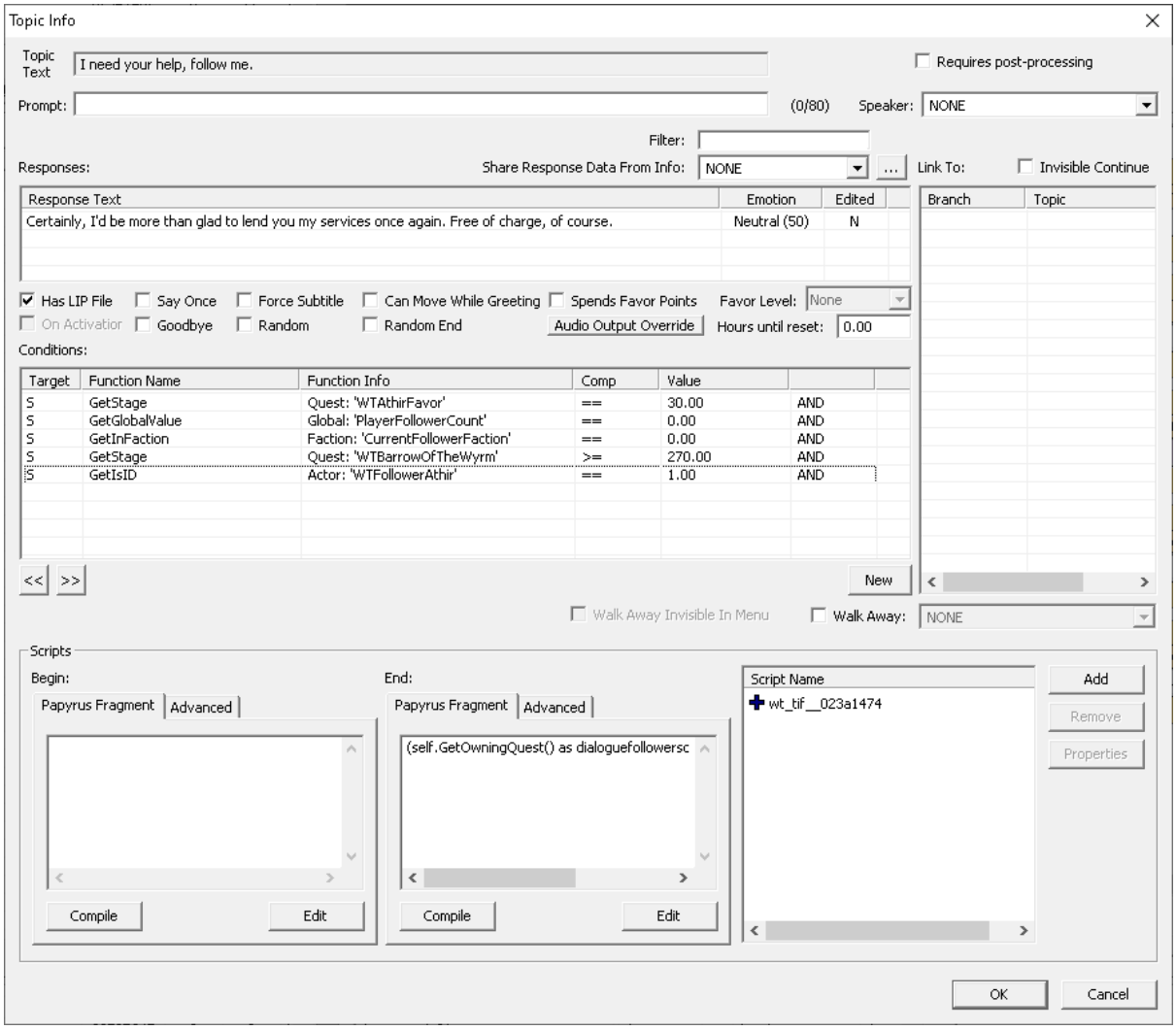


Figure 548 - 'Follow me' topic.

In the conditions, we're checking to make sure Athir isn't our current follower by checking the faction rank of CurrentFollowerFaction and ensuring it's set to 0. We also need to check the PlayerFollowerCount global value and make sure it's 0 so we won't see this line of dialogue if the player already has a follower.

Note: The default follower system permits only one NPC to follow the player at a time. The current follower needs to be dismissed before another NPC can be recruited.

Here's the set up for the 'Wait here' topic.

The papyrus we need to run for this topic is as follows:

```
(self.GetOwningQuest() as dialoguefollowerscript).FollowerWait()
```

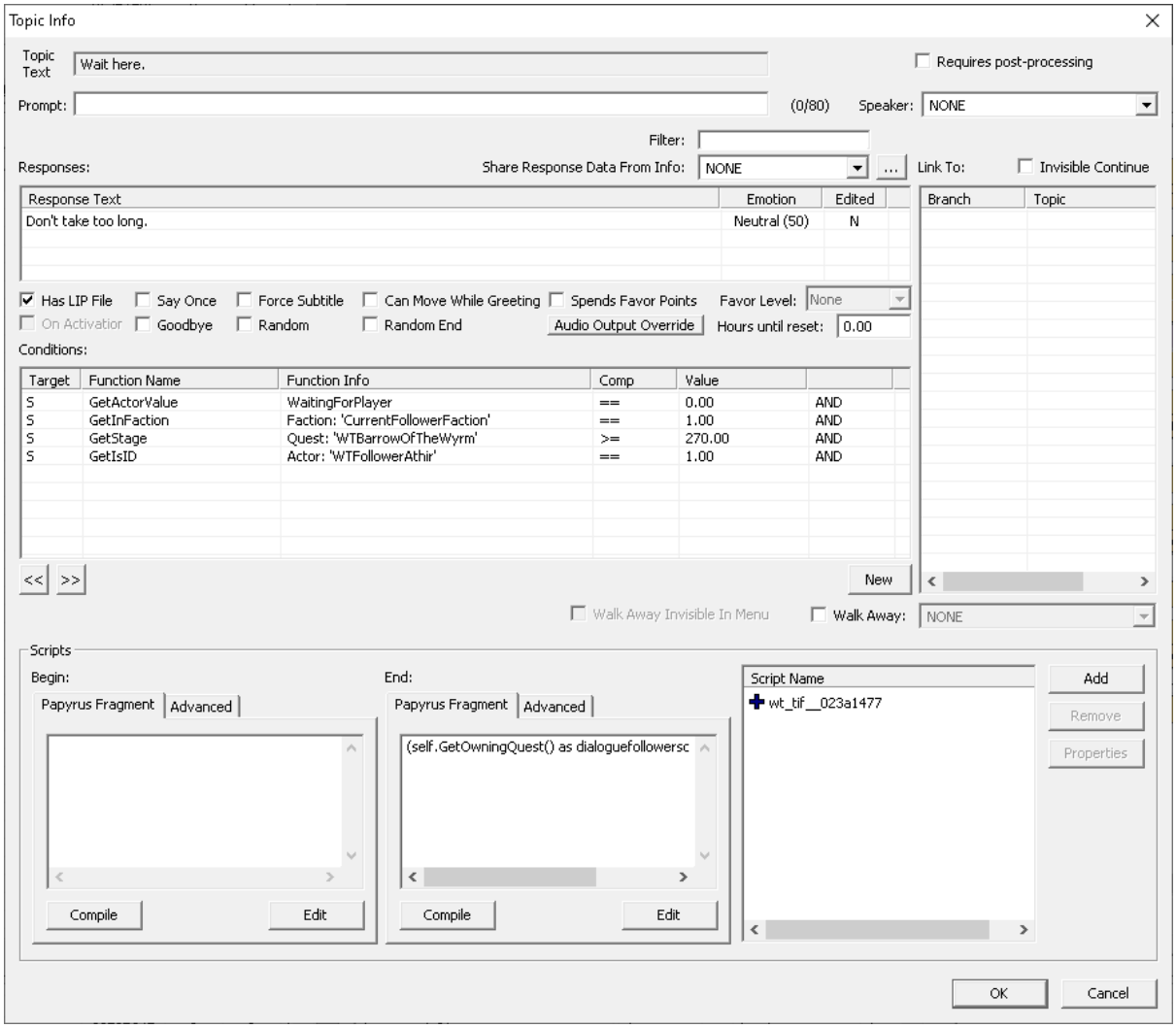


Figure 549 - 'Wait here' topic.

Now that Athir is our current follower, we need to make sure the faction rank of CurrentFollowerFaction is 1. Before we can tell him to wait, we need to make sure we haven't already told him to wait. We can do this by checking the WaitingForPlayer actor value on Athir and making sure it's set to 0.

Here's the set up for the 'Follow me' topic. This is used if we've asked our follower to wait and we want them to resume following us again.

The papyrus we need to run for this topic is as follows:

```
(self.GetOwningQuest() as dialoguefollowerscript).FollowerFollow()
```

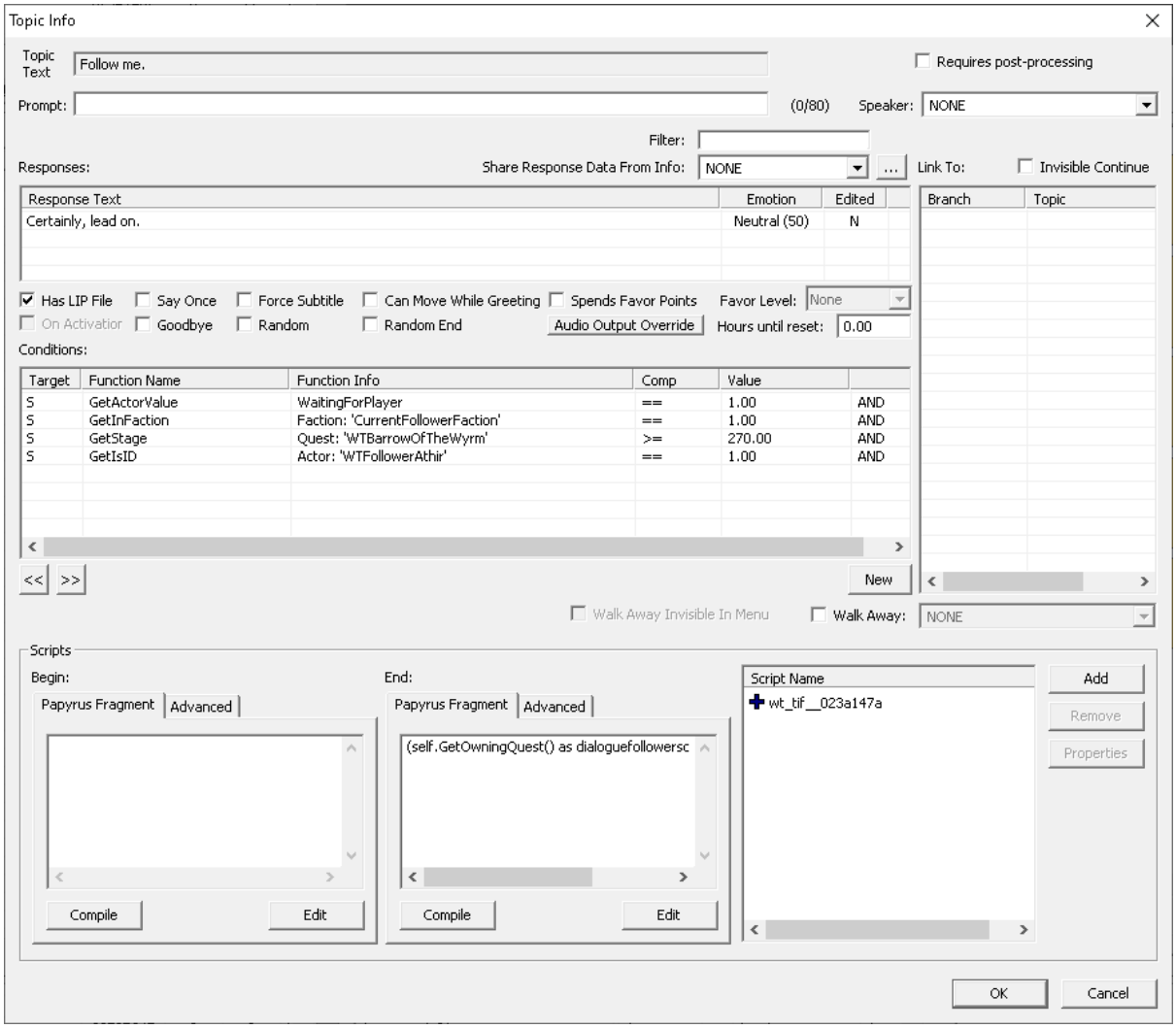


Figure 550 - 'Follow me' topic.

Here we're checking the WaitingForPlayer actor value to make sure it's set to 1 to indicate that this NPC is currently waiting for the player.

If we need to open the NPC's inventory to trade items, we can do so with the following command:

```
akSpeaker.OpenInventory(false)
```

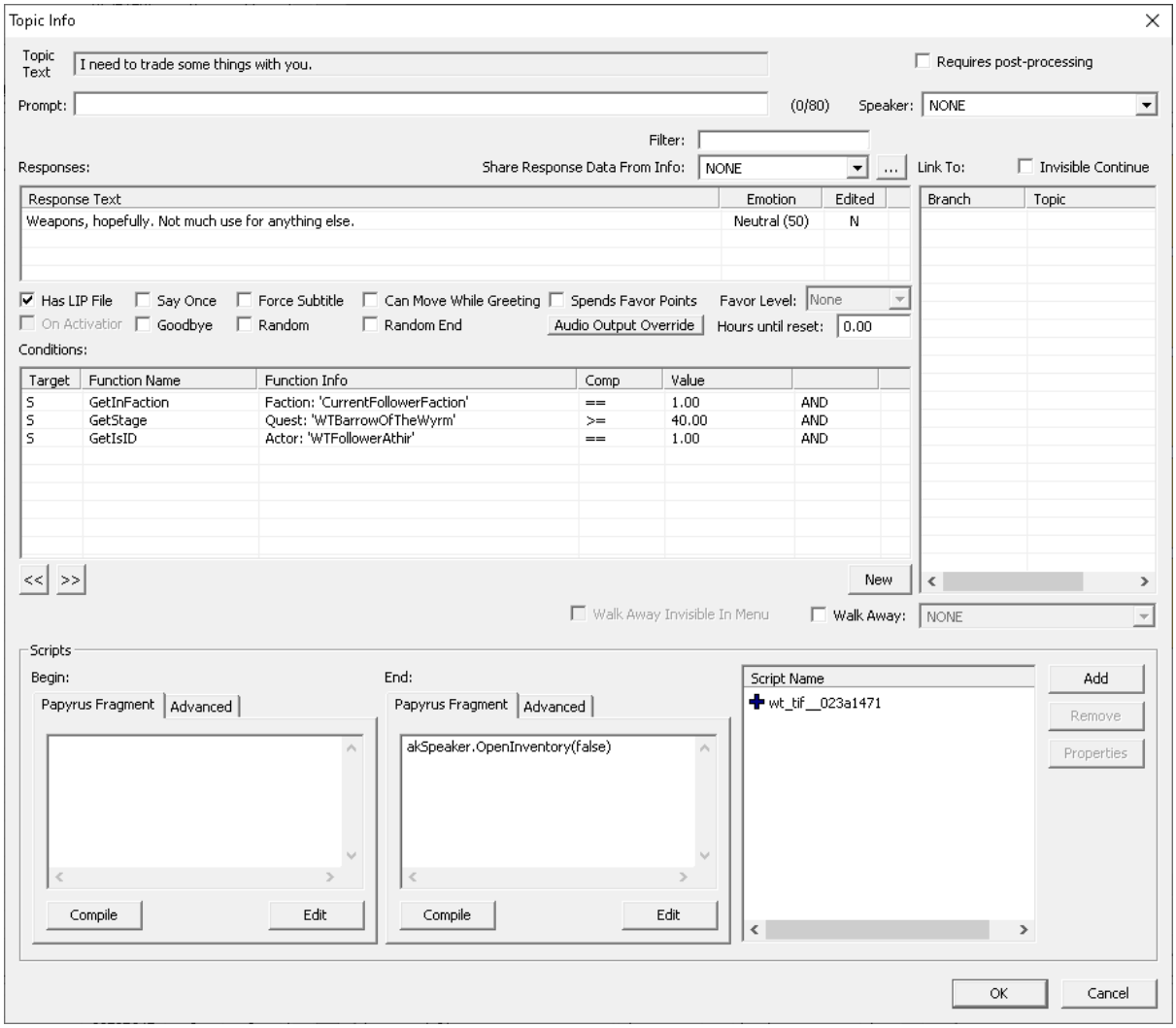


Figure 551 - 'I need to trade some things with you' topic.

We just need to make sure the NPC is in CurrentFollowerFaction.

To give the NPC a specific command, we can use the following papyrus code:

```
akSpeaker.SetDoingFavor(true)
```

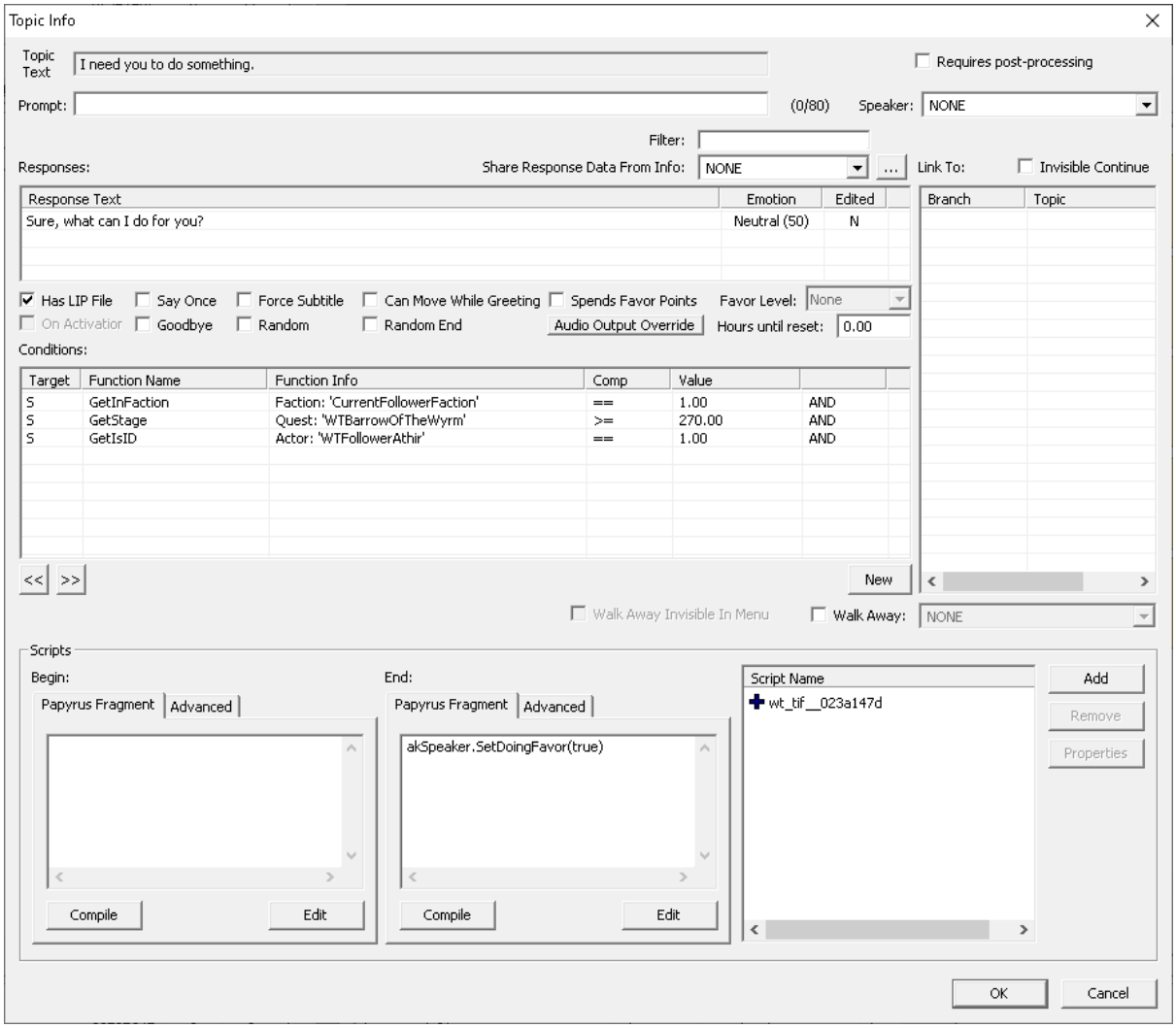


Figure 552 - 'I need you to do something' topic.

Again, we just need to make sure the NPC is in CurrentFollowerFaction.

Finally, to dismiss the NPC as our follower, we can use the following code:

```
(pDialogueFollower as dialoguefollowerscript).DismissFollower(0, 0)
```

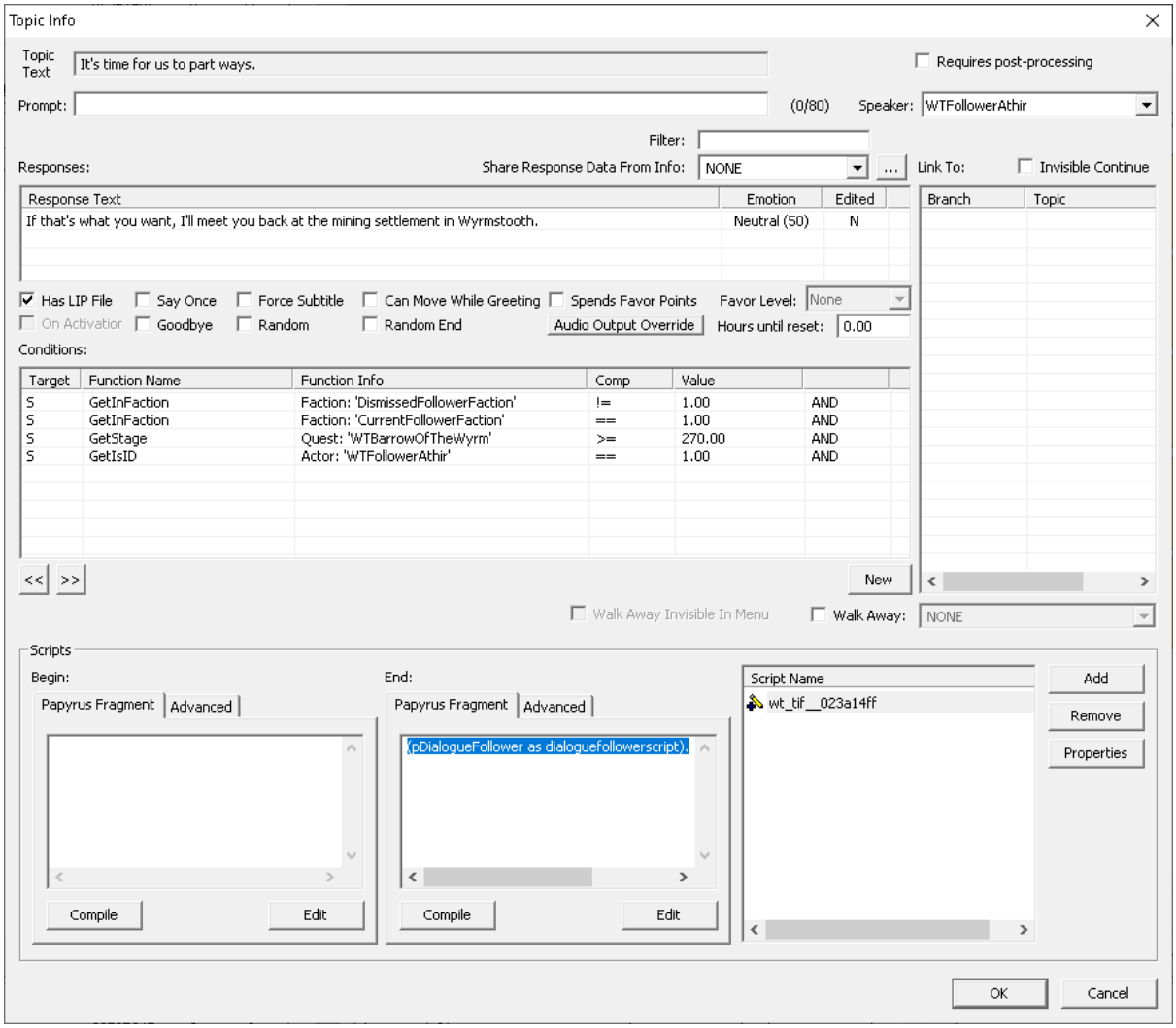


Figure 553 - 'It's time for us to part ways' topic.

Here we need to make sure the NPC is in CurrentFollowerFaction and is not a member of DismissedFollowerFaction.

Here are the properties on the papyrus fragment. It's just a Quest property named pDialogueFollower pointing to the DialogueFollower quest.

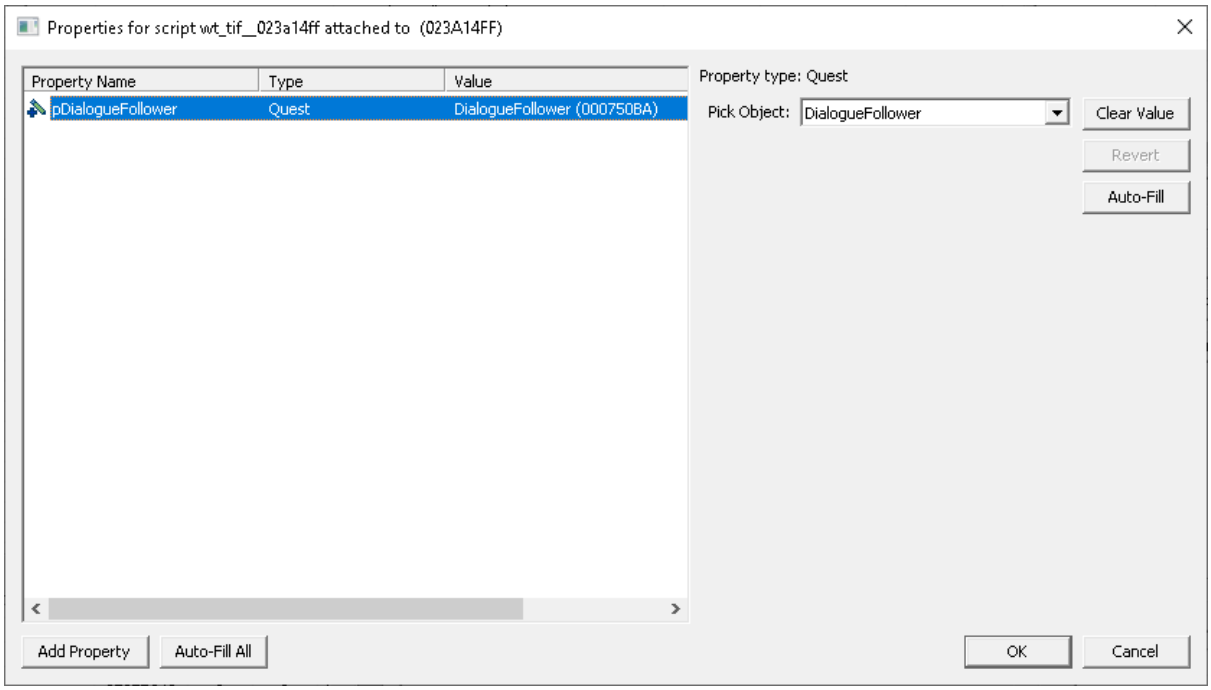


Figure 554 - Script properties for the dismiss topic papyrus fragment.

For more information setting up quest dialogue, see the section Adding dialogue to a quest.

SETTING UP A HOME OWNER

In this section I'll cover setting up an NPC to lock their doors at night and deliver trespass warnings to the player if they are caught breaking in and entering.

For this example, Jorn moved out of the Riverwood Inn and now he's going to take up residence in the interior cell LundsHut01. After removing the cobwebs, skeevers and Lund's corpse, it's time to make this place a home.



Figure 555 - Jorn in Lund's hut.

The first thing we need to do is set up a new faction for this house.

In the Object Window, go to Character > Faction, right-click on one of the existing factions and select New.

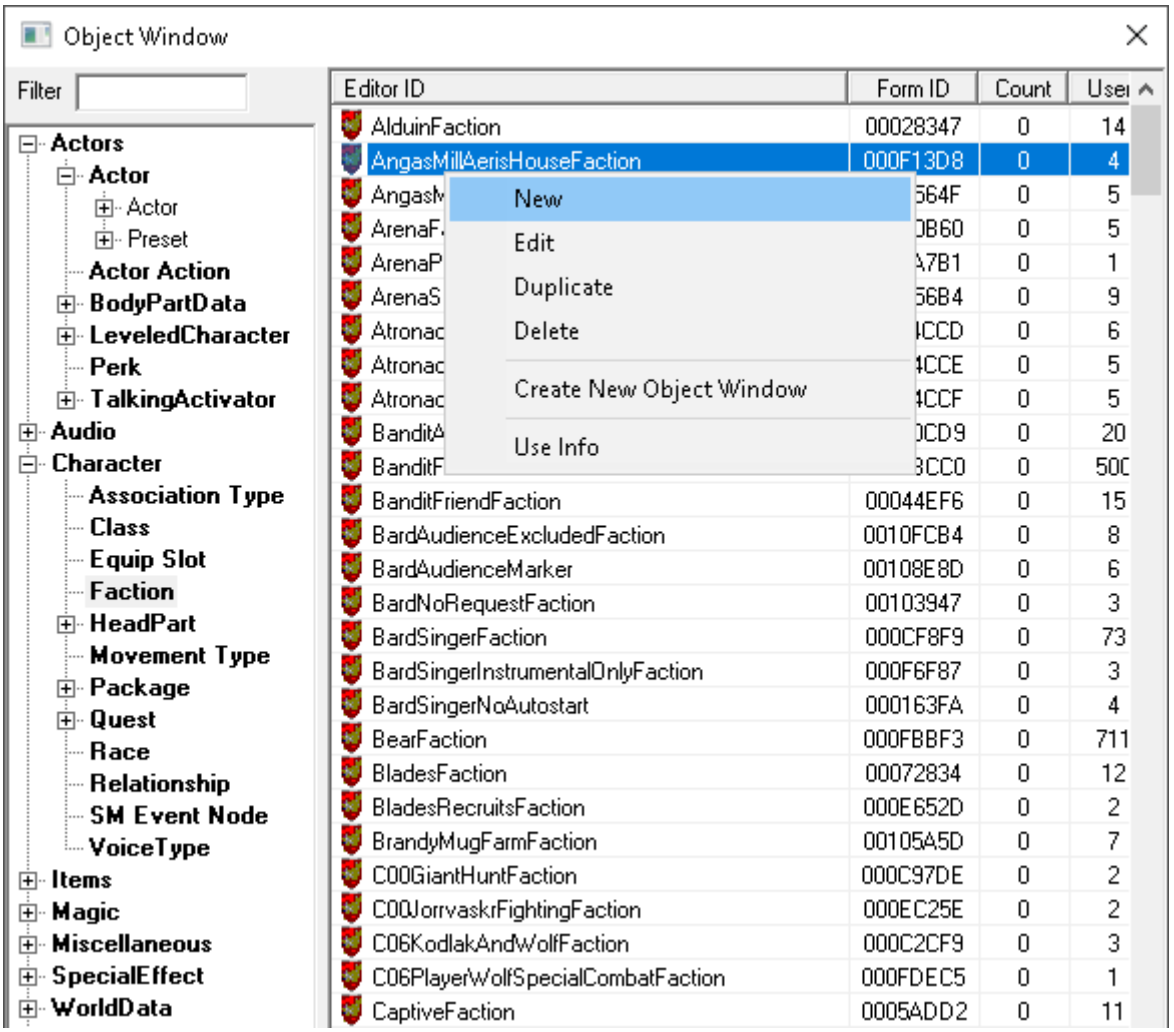


Figure 556 - Creating a new faction.

Set the faction ID. I'm just going to call it WTJornHouseFaction.

Make sure to tick 'Can Be Owner' then click OK.

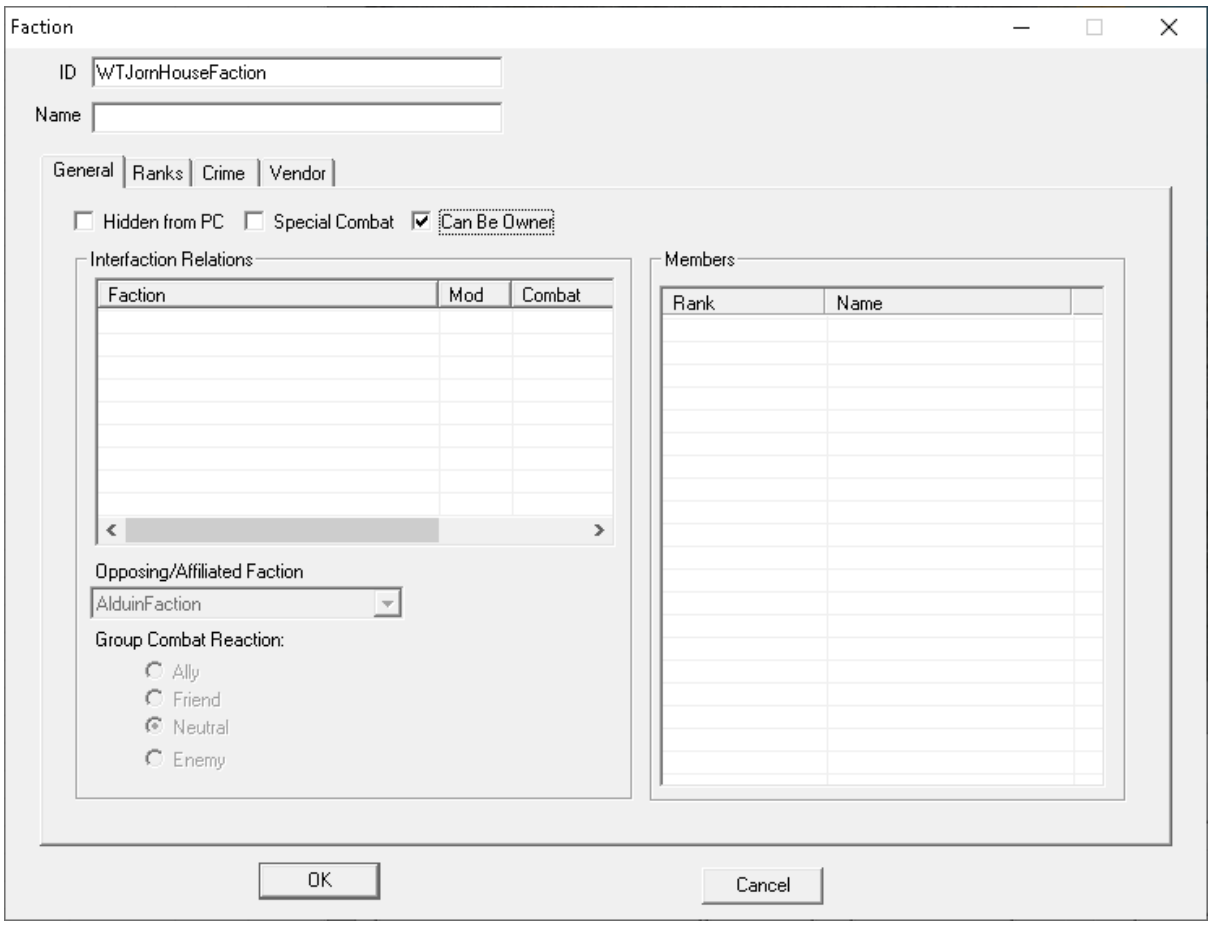


Figure 557 - Setting up the house faction.

Next, let's create a new key for the front door.

In the Object Window, go to Items > Key, right-click on one of the existing keys and select Duplicate.

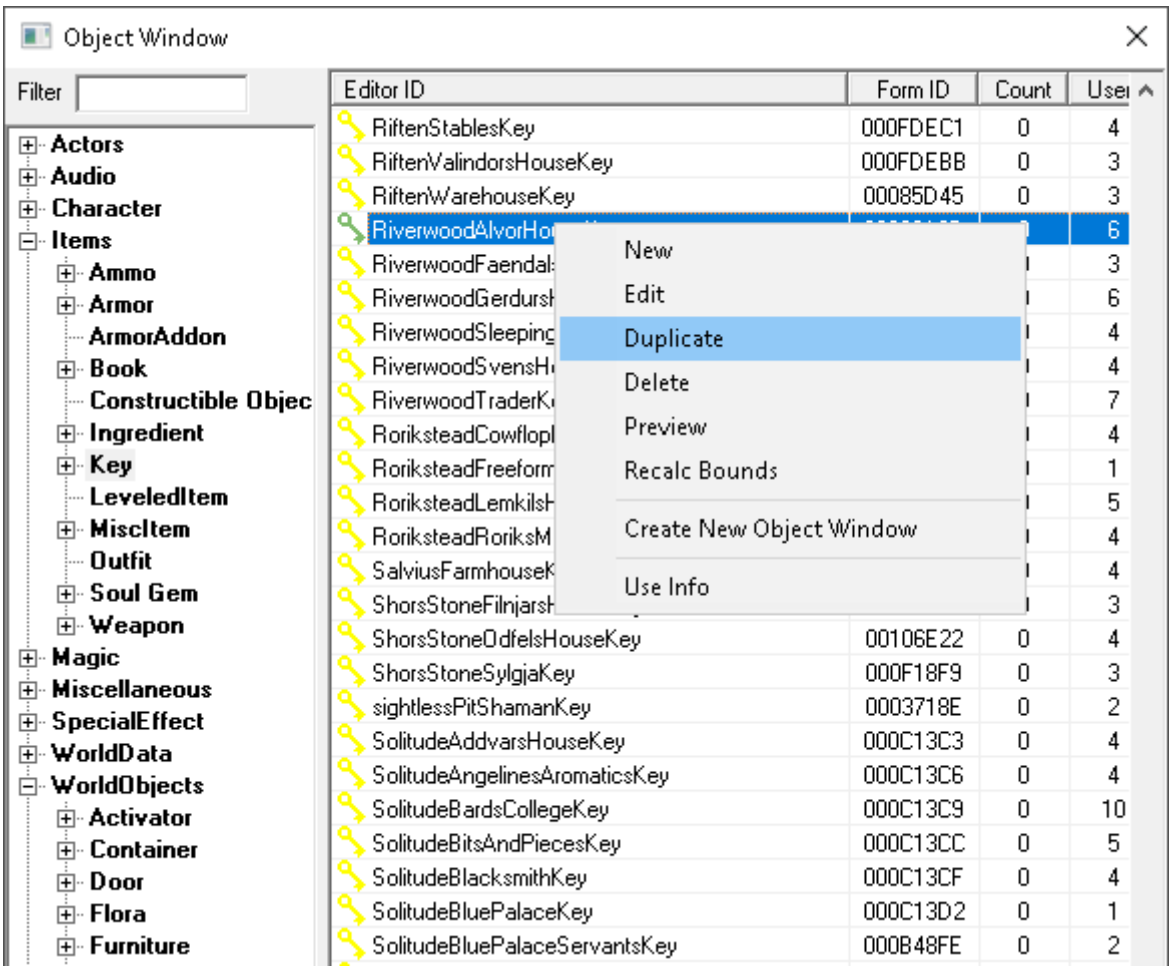


Figure 558 - Duplicating an existing key.

Click on the duplicate to select it then press 'F2' to rename it.

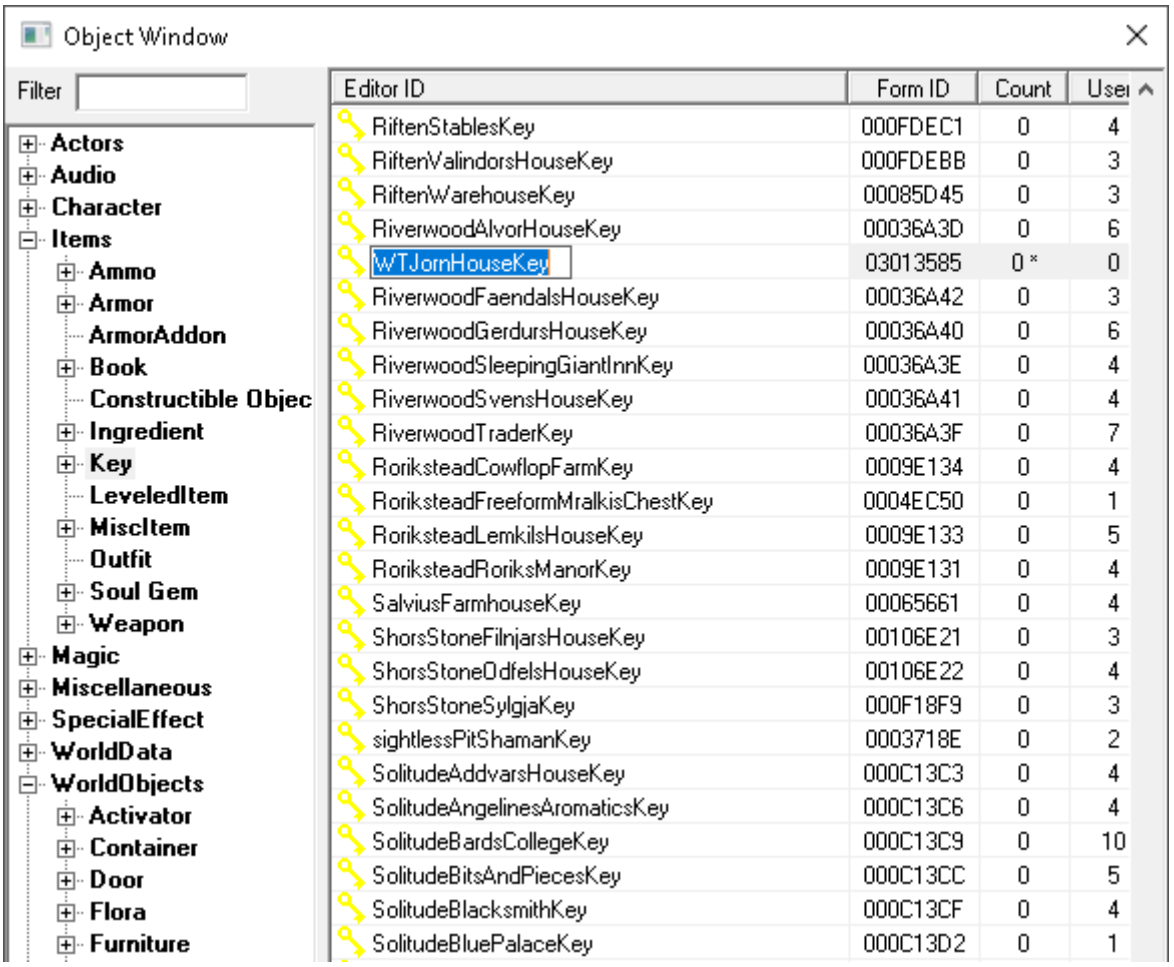


Figure 559 - Renaming the duplicate key.

When prompted to create a new object, click No.



Figure 560 - Create New Object pop-up.

Next we need to set this key as the key that unlocks the front door.

Double-click or right-click on the front door and select Edit.

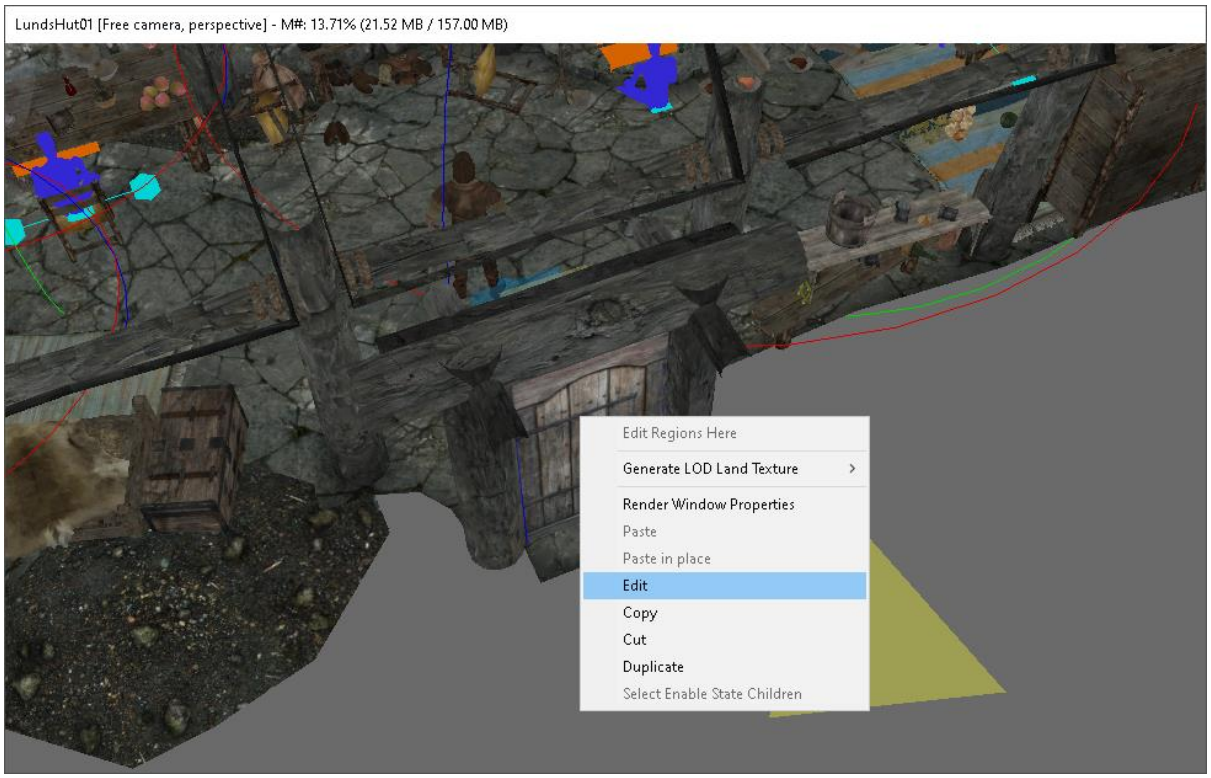


Figure 561 - Opening the reference properties for the front door.

Go to the Lock tab and tick the Locked tickbox.

Set the lock level. For this example I just set it to Adept.

Set the Key drop down to the key we just created.

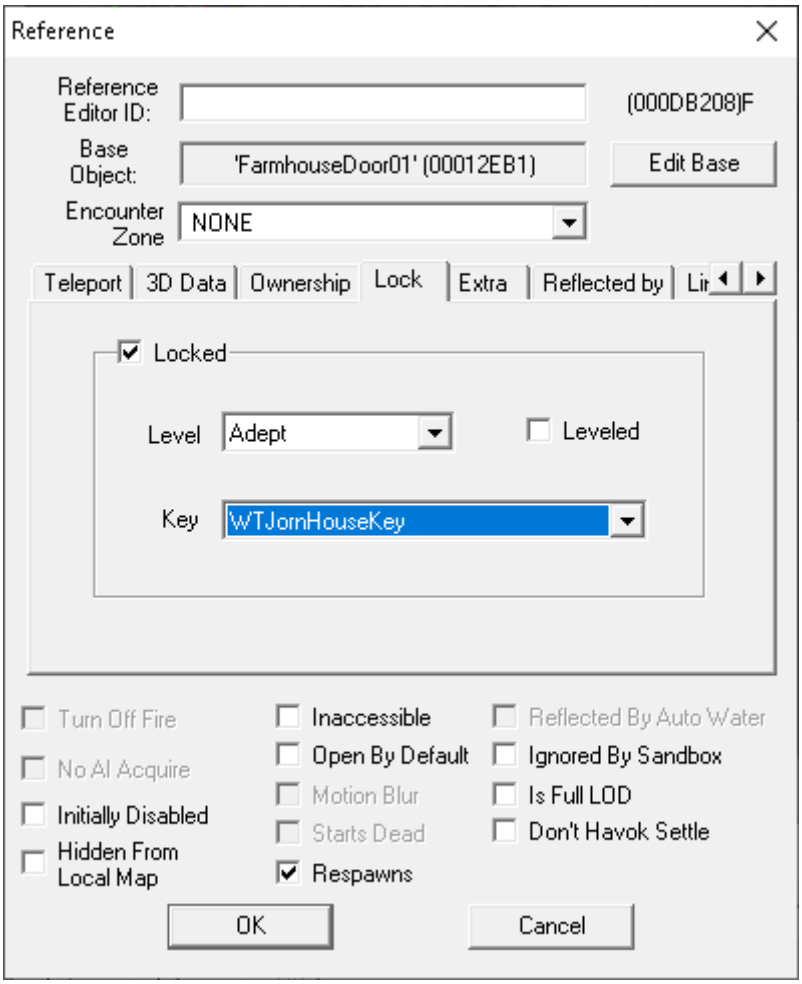


Figure 562 - Setting up a door lock.

Click OK to close out of Reference properties.

Next, we need to add Jorn to his new house faction.

In the Object Window, go to Actors > Actor > Actor and double-click or right-click on Jorn and select Edit.

Go to the Factions tab, right-click in the factions list and select New.

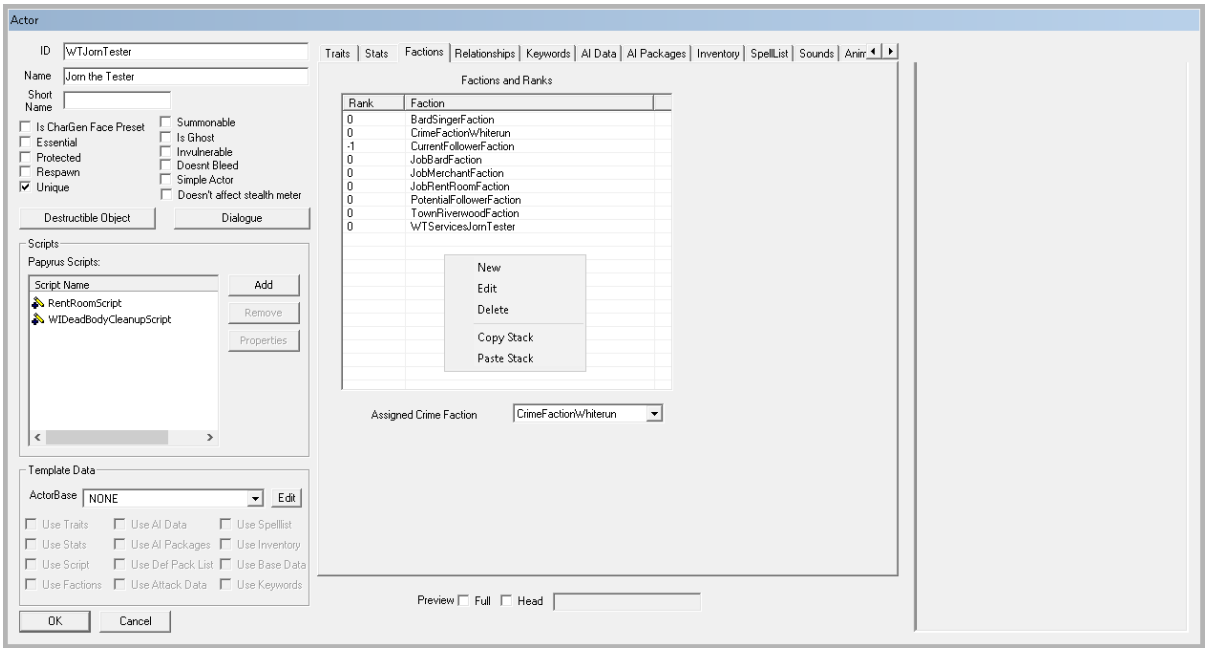


Figure 563 - Adding Jorn to the new house faction.

Select the house faction you just made then click OK.

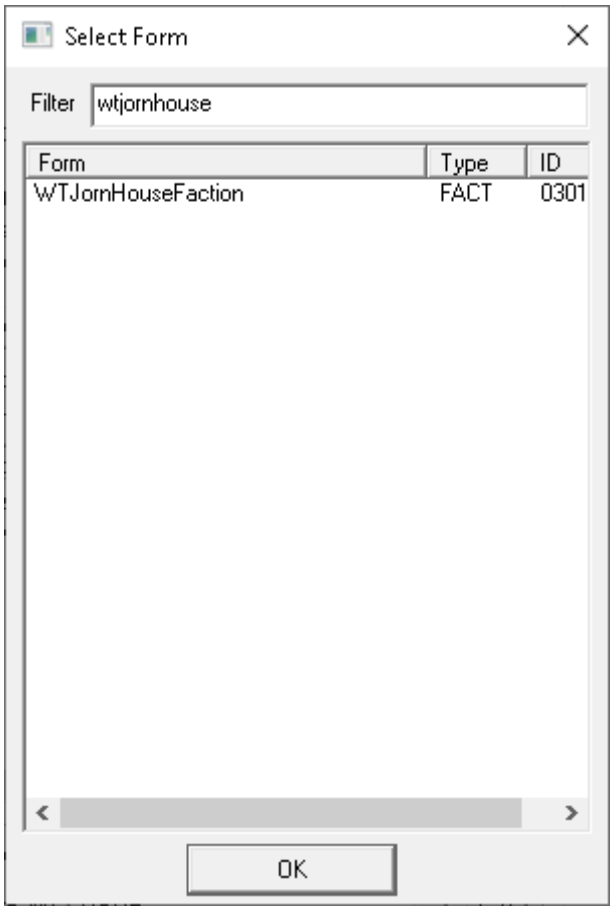


Figure 564 - Selecting the WTJornHouseFaction faction.

Since we've moving Jorn out of the Riverwood Inn, I'd also like him to retire as a bard and as a merchant so I removed him from those factions but left him as a potential follower.

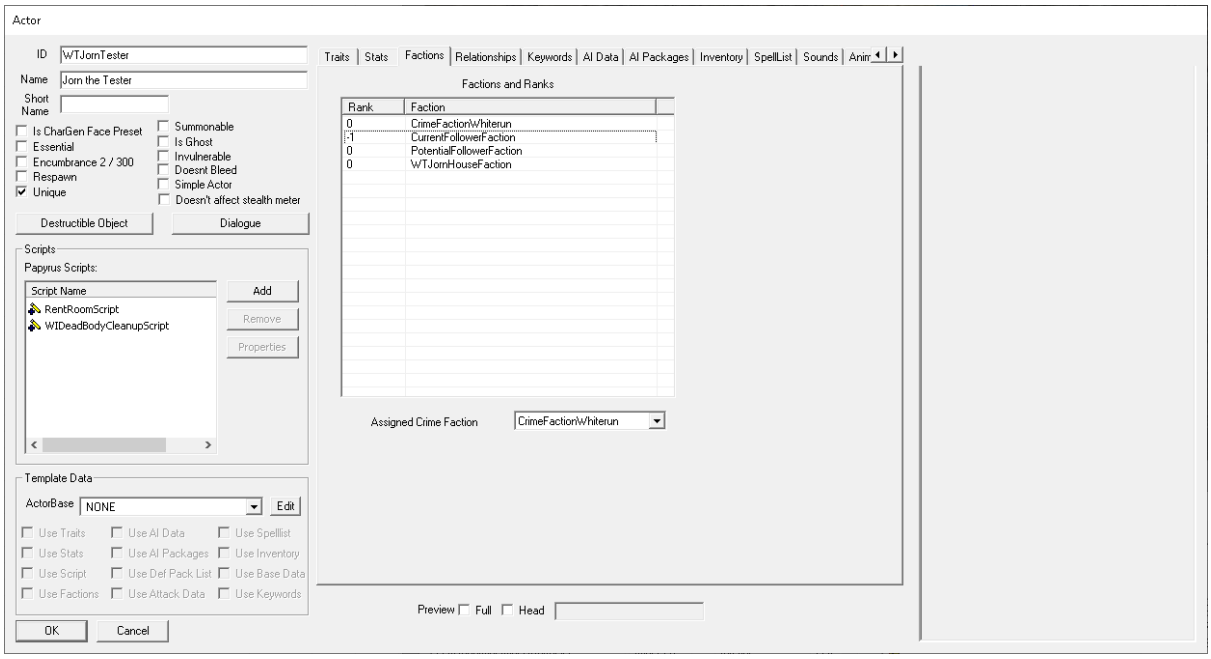


Figure 565 - Jorn removed from the bard and merchant factions.

Let's change his schedule a bit.

Go to the AI Packages tab and delete the WTJornBardPerformance9x10 package.

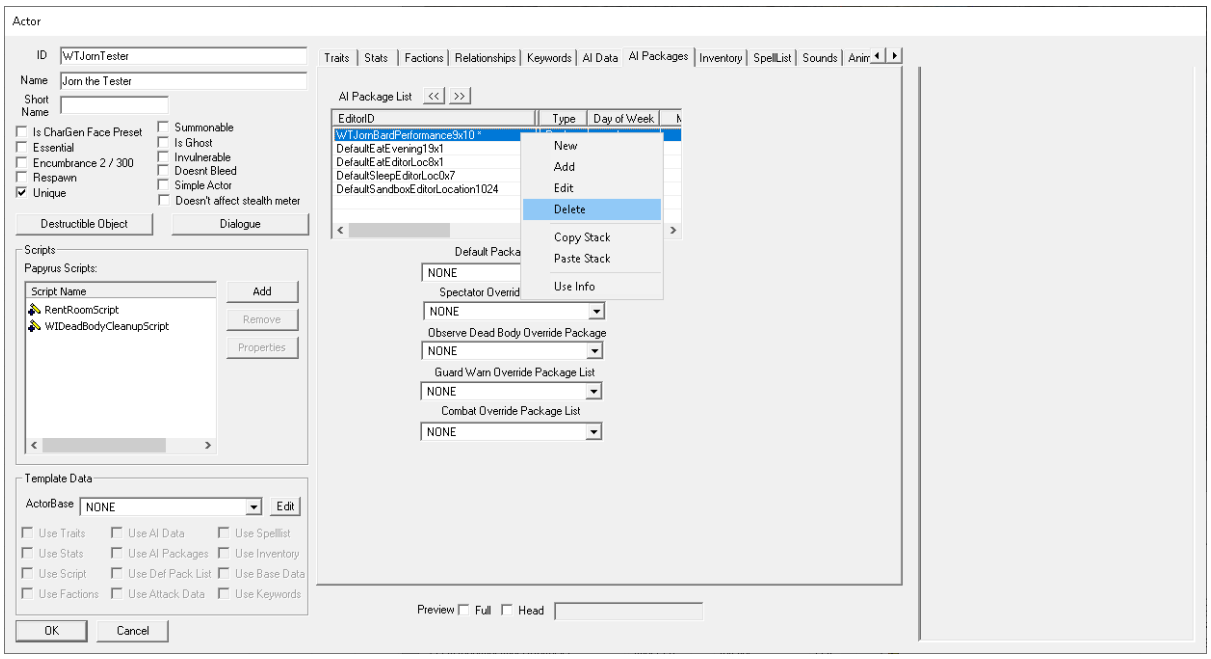


Figure 566 - Removing Jorn's bard performance package.

Let's have him spend some time outside during the day.

Right-click in the packages list and select New.

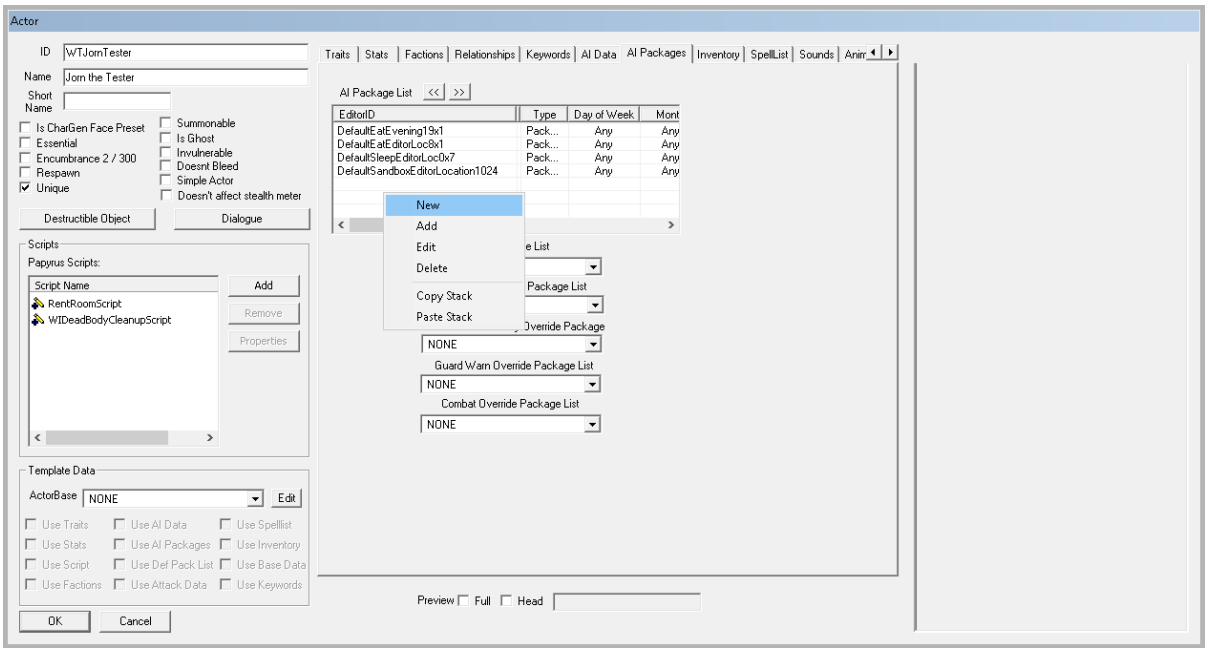


Figure 567 - Adding a new package.

Set the package ID. For this package I'm going to set it to WTJornSandboxOutside9x10.

Set the Package Template to Sandbox.

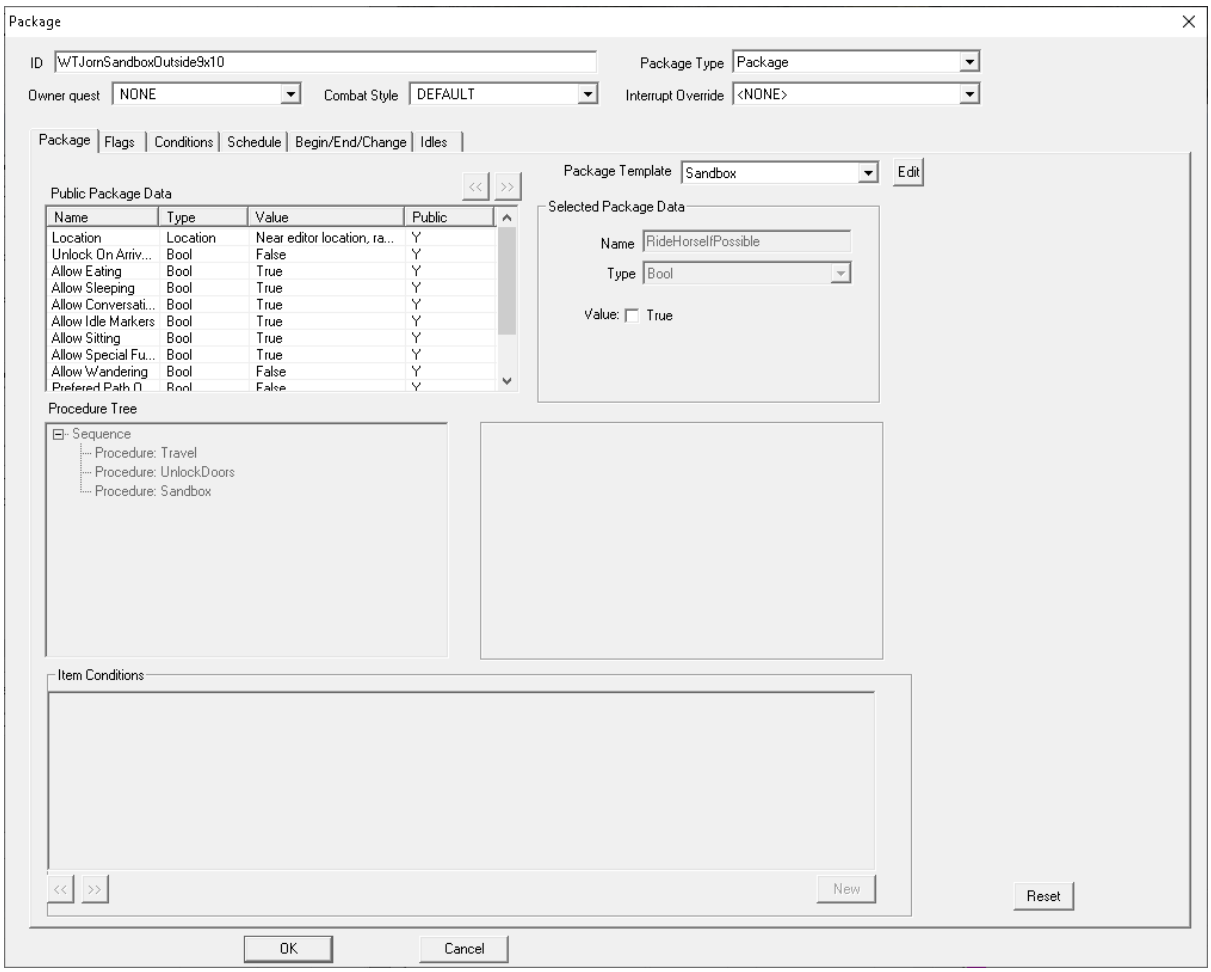


Figure 568 - Setting the ID and package template.

In the Public Package Data list, click on Location. Set the Radius field to 2048 then click on the button above it that currently reads 'Near editor location, radius 512'.

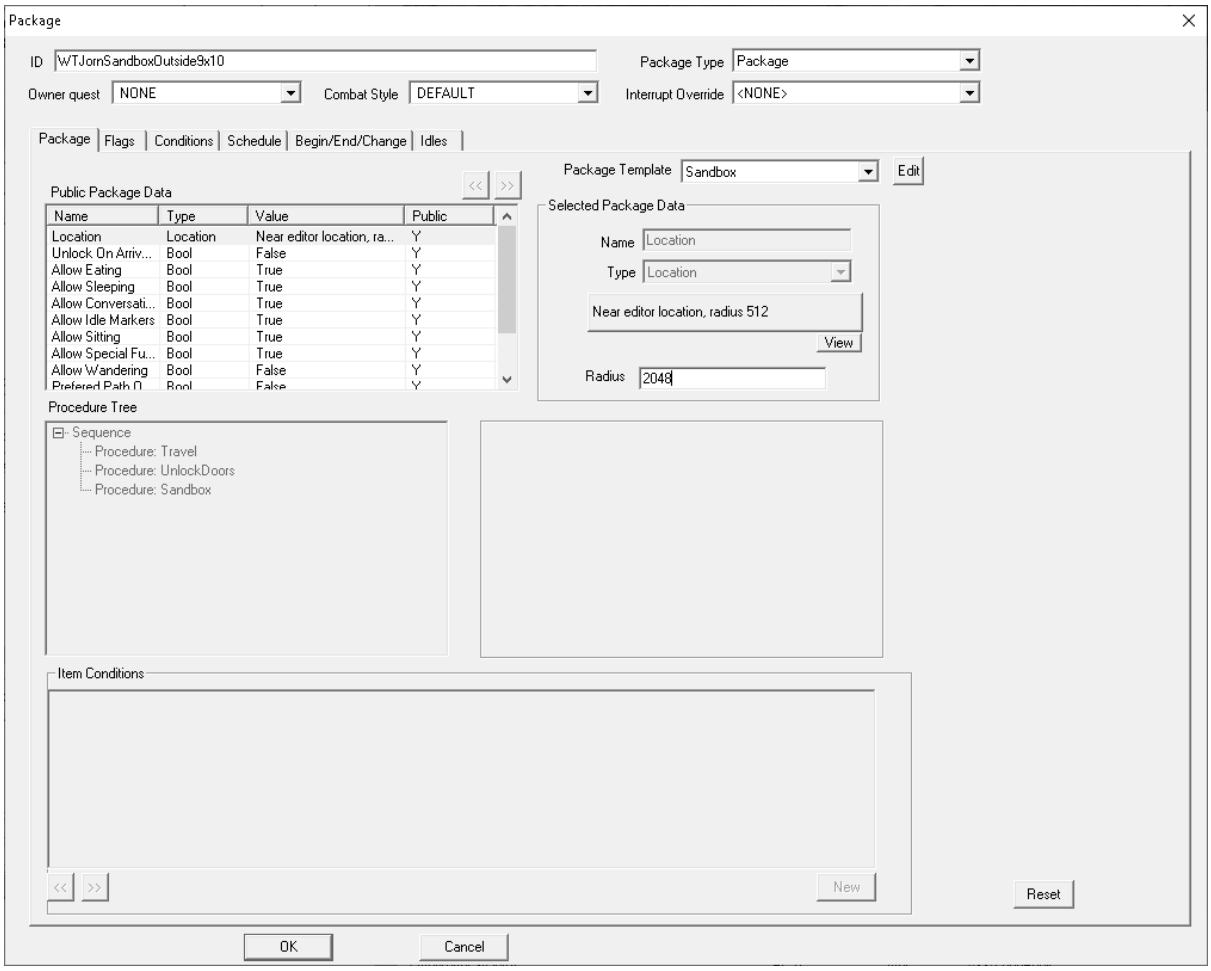


Figure 569 - Setting the package sandbox location.

Select Near Reference then click on the Select Reference button.

For this example, I'm just going to select the Map Marker outside Lund's Hut as the center point for Jorn's sandbox package.

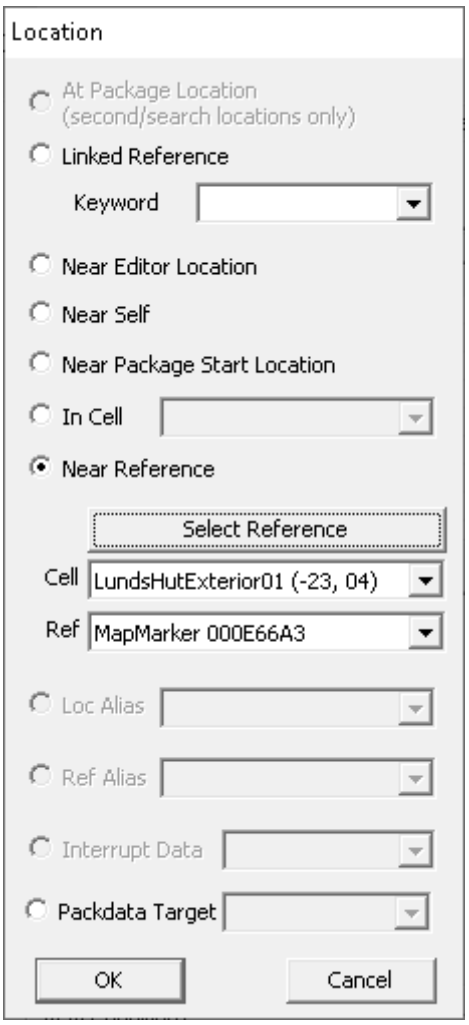


Figure 570 - Selecting the map marker as the center point of Jorn's sandbox package.

Click OK.



Figure 571 - The map marker outside Lund's Hut.

Normally you'd want to add in an XMarker or XMarkerHeading to use as the center point for a sandbox package.

There's not a lot to do outside Lund's Hut, so tick Allow Wandering.

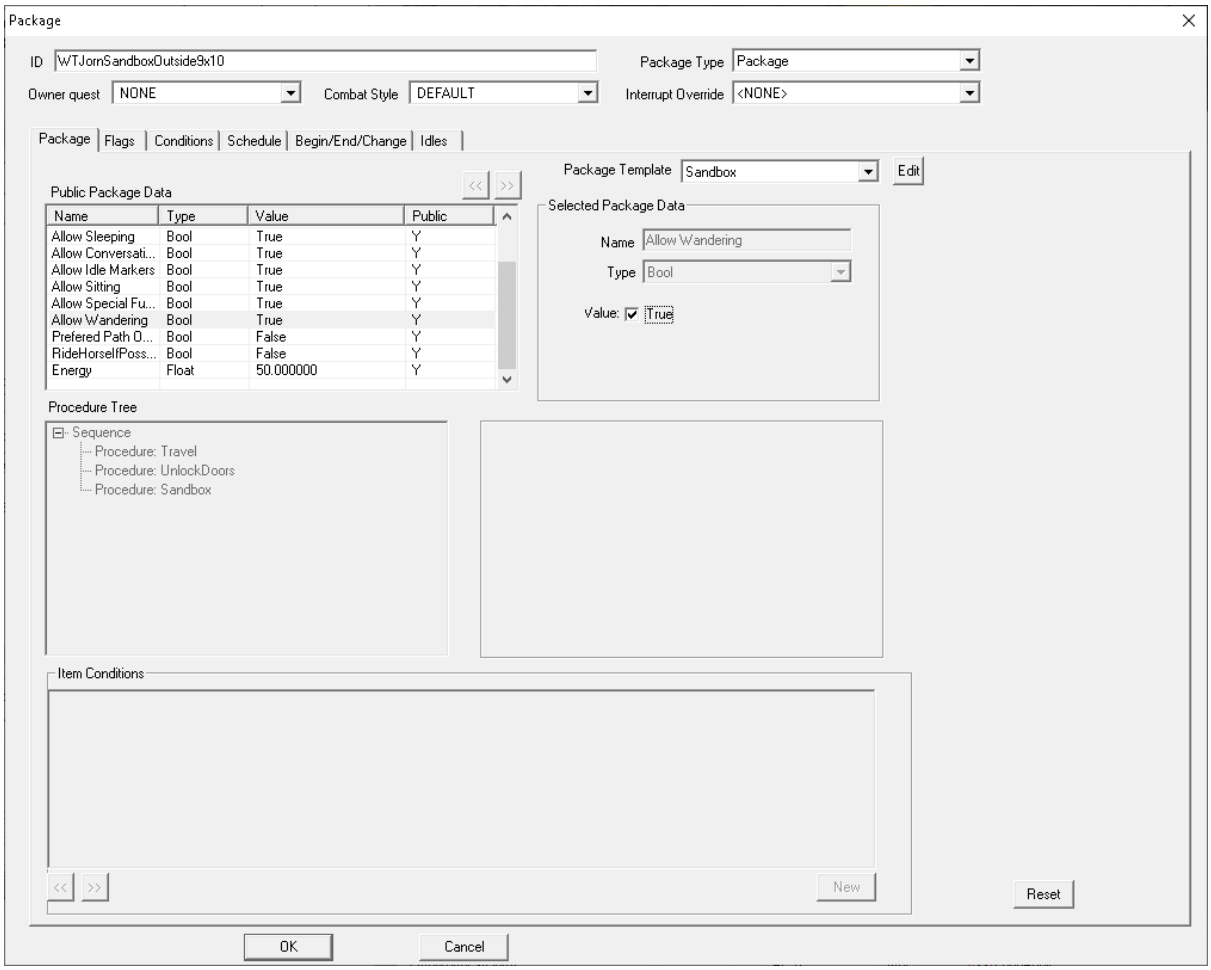


Figure 572 - Turning on Allow Wandering.

Lastly, go to the Schedule tab.

Set the Hour drop-down to 9 and the duration field to 10.0000.

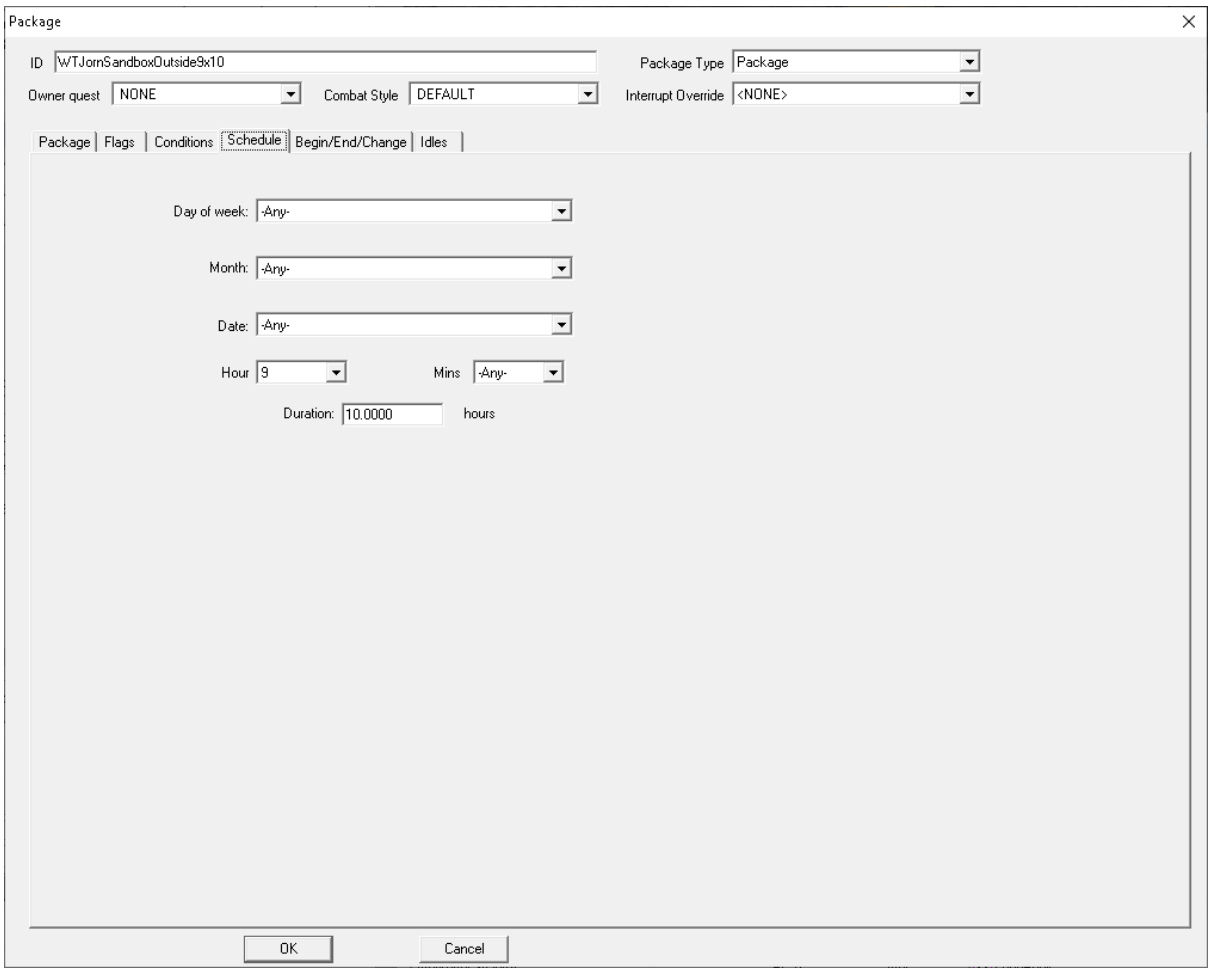


Figure 573 - Setting the schedule for Jorn's aimless wandering.

Click OK to close out of Package properties.

Ensure the new package is at the top of the package hierarchy by highlighting it and clicking on the '<<' button.

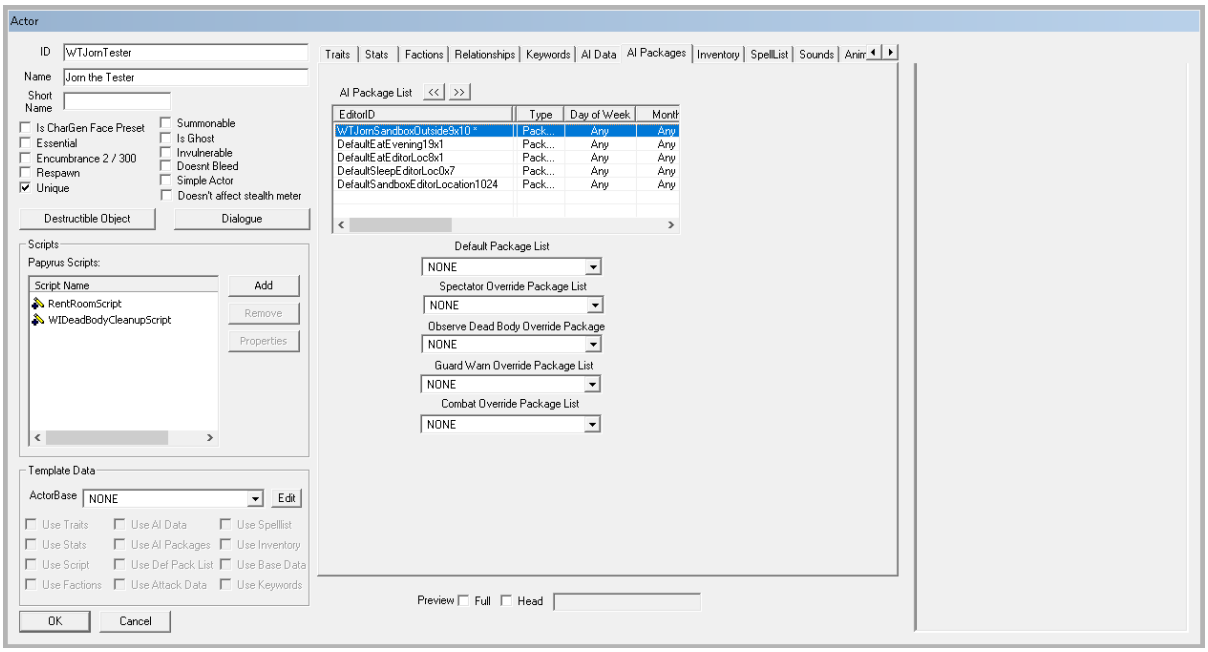


Figure 574 - Sorting the package hierarchy if necessary.

Set the Default Package List drop-down to DefaultHomeOwnerPackageList.

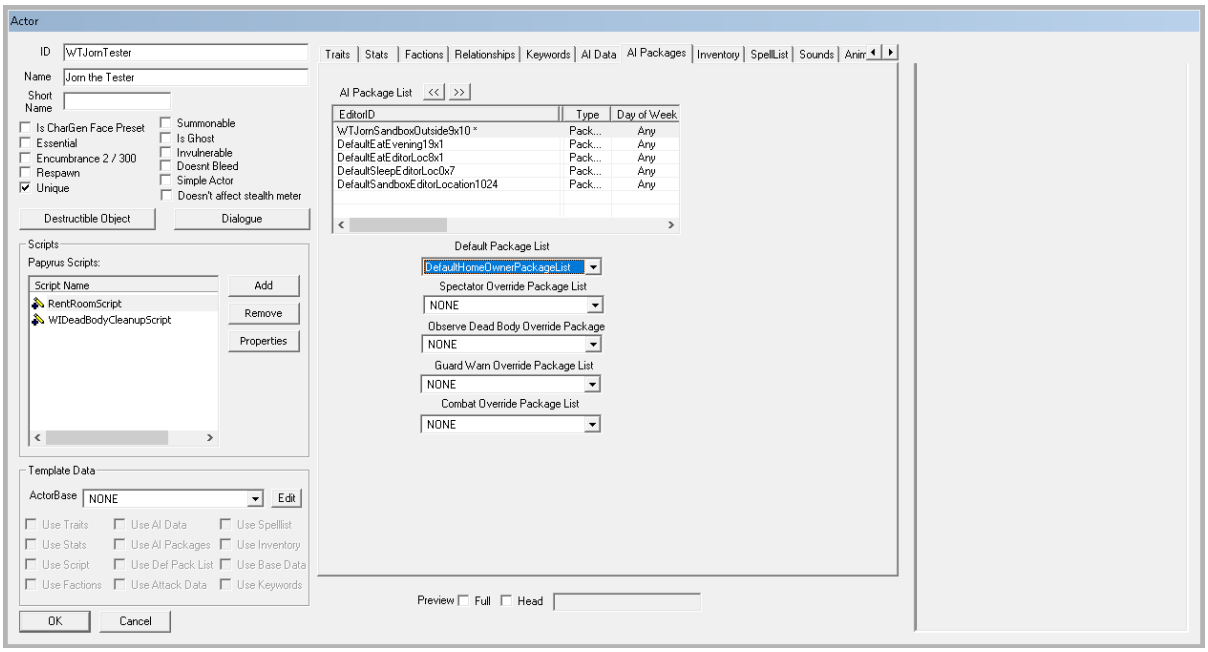


Figure 575 - Setting a default package list.

Click on RentRoomScript under Papyrus Scripts to highlight it then click on Remove to remove it from Jorn since we don't need it anymore.

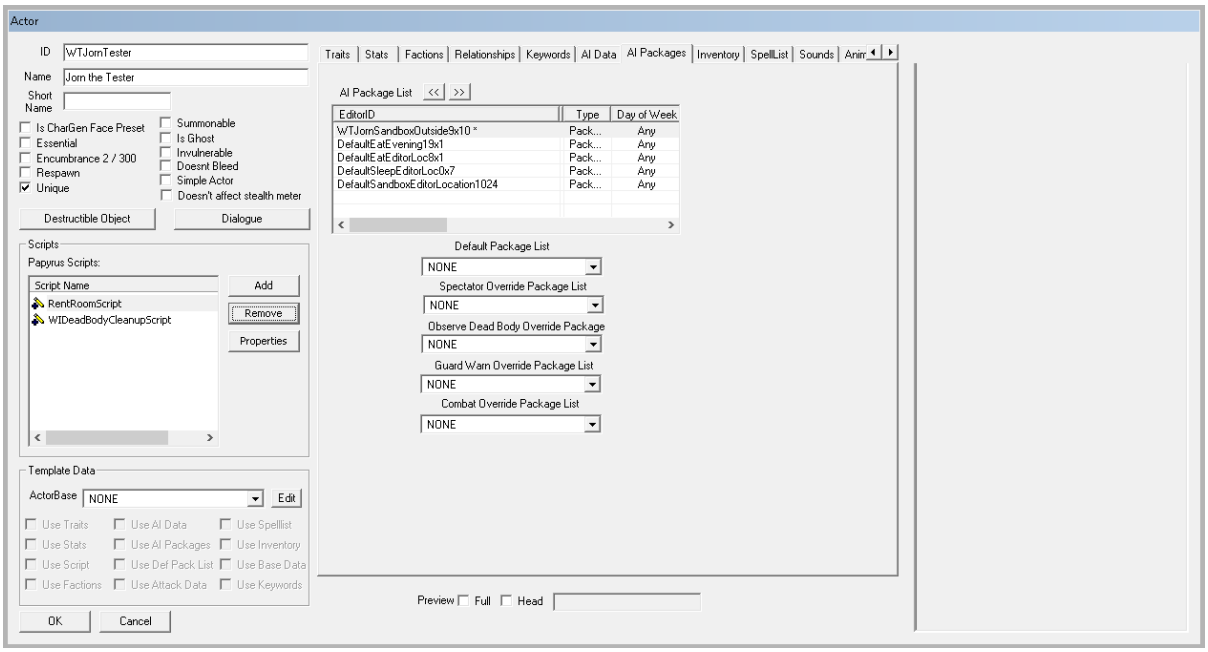


Figure 576 - Removing RentRoomScript.

Lastly, go to the Inventory tab, right-click in the inventory list and select New.

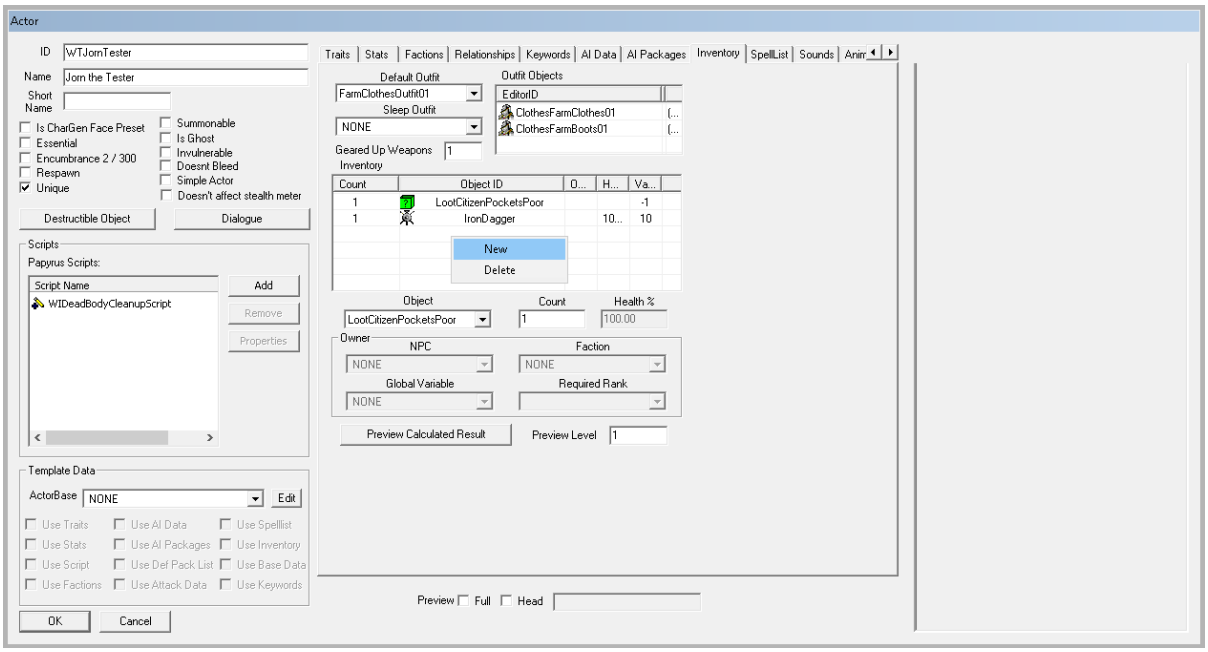


Figure 577 - Adding Jorn's key to his inventory.

Set the new inventory item to the key we created earlier.

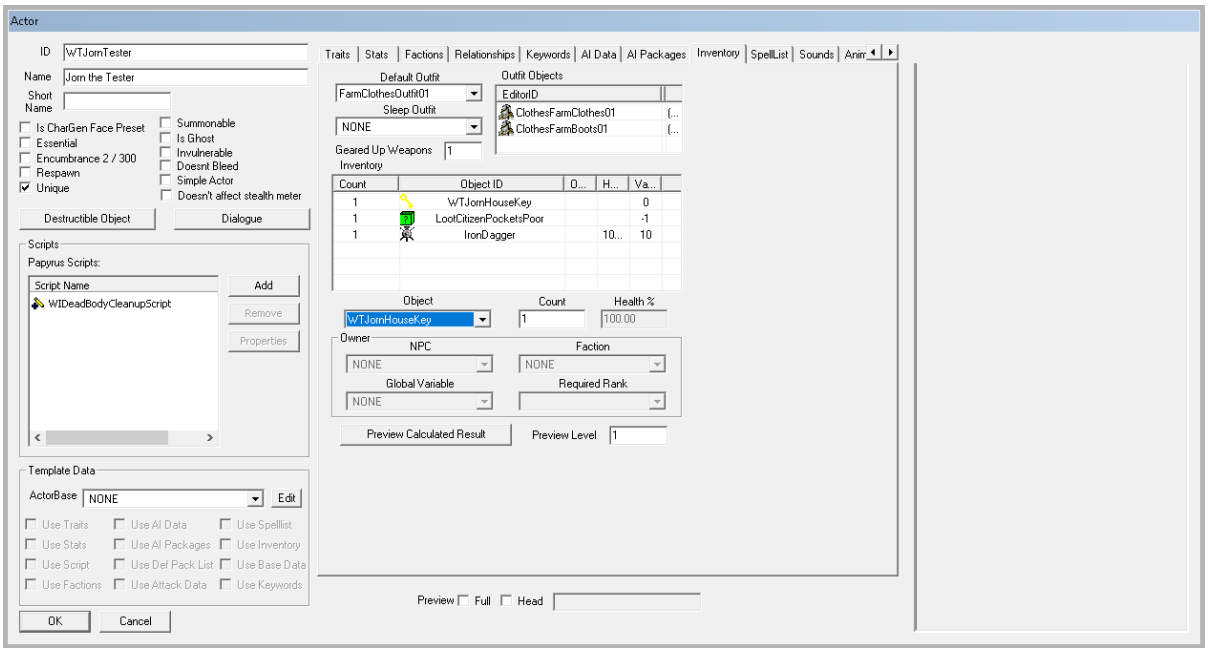


Figure 578 - Setting the new object as Jorn's key.

Click OK to close out of Actor properties.

Next, we need to set up a new locklist for the interior cell. This will contain a list of NPCs that are allowed to lock and unlock doors when entering/leaving this cell.

In the Object Window go to Miscellaneous > FormList, right-click on an existing form list and select New.

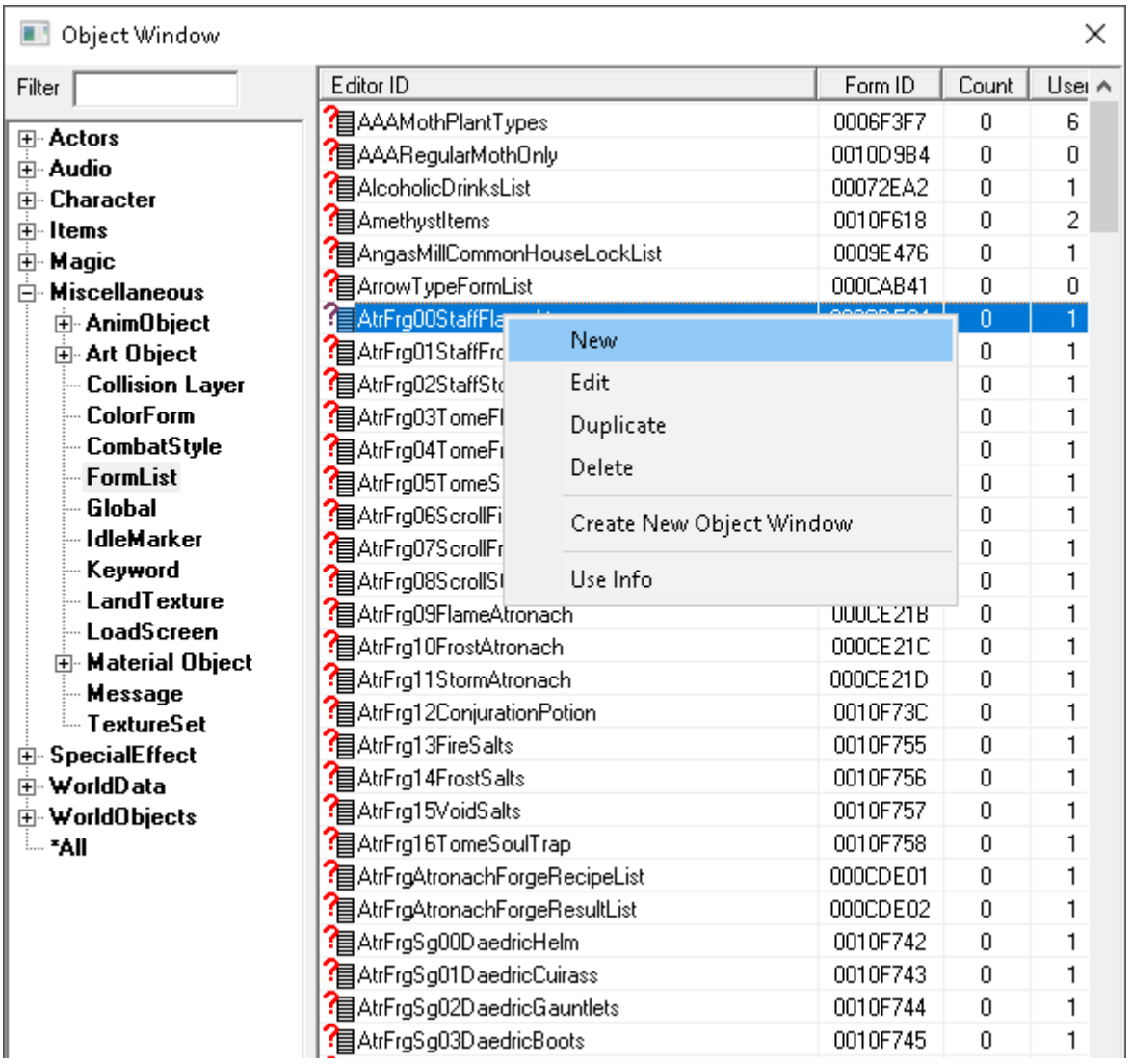


Figure 579 - Creating a new form list.

Set the form ID. For this example I'm going to call it WTJornHouseLockList.

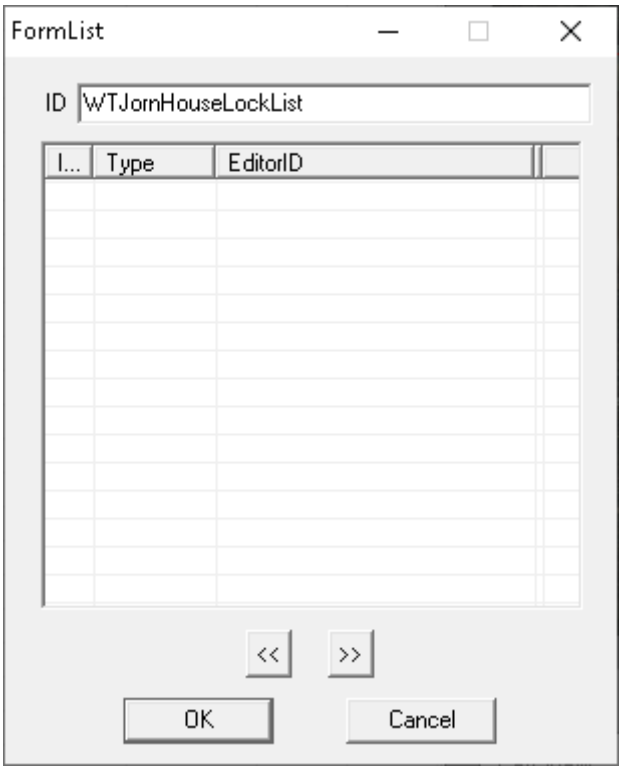


Figure 580 - New lock list.

In the Object Window, go to Actors > Actor > Actor and drag and drop WTJornTester into the form list.

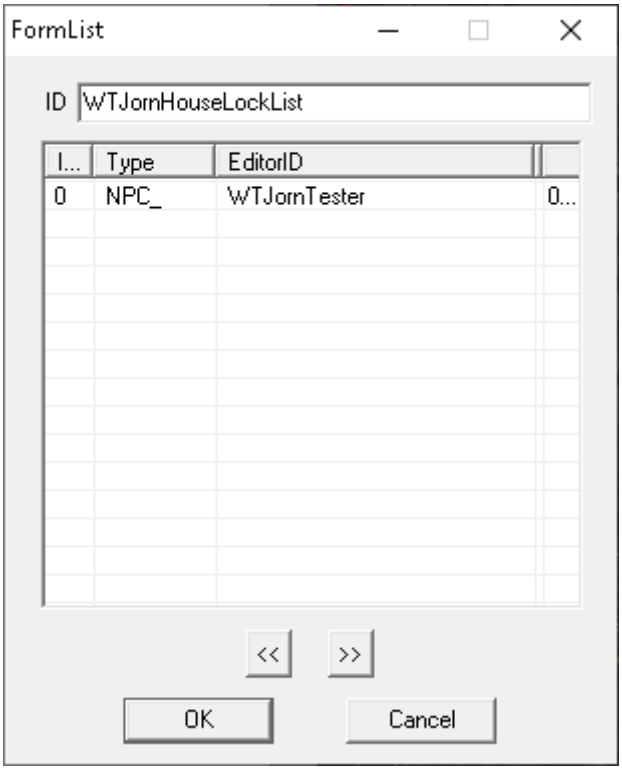


Figure 581 - Jorn added to thelock list.

Click OK to close the form list properties.

Now we need to configure the interior cell to set Jorn as the home owner.

In the Cell View window, right-click on LundsHut01 and select Edit.

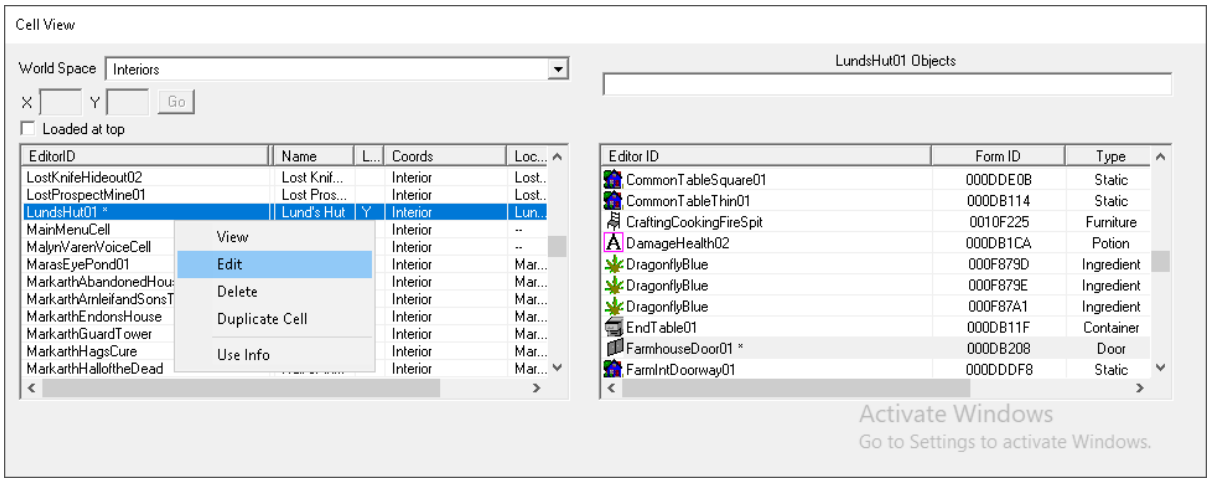


Figure 582 - Editing the Lund's Hut cell.

Go to the Interior Data tab.

Set the Owner faction drop-down to WTJornHouseFaction.

Set the Lock List drop down to WTJornHouseLockList.

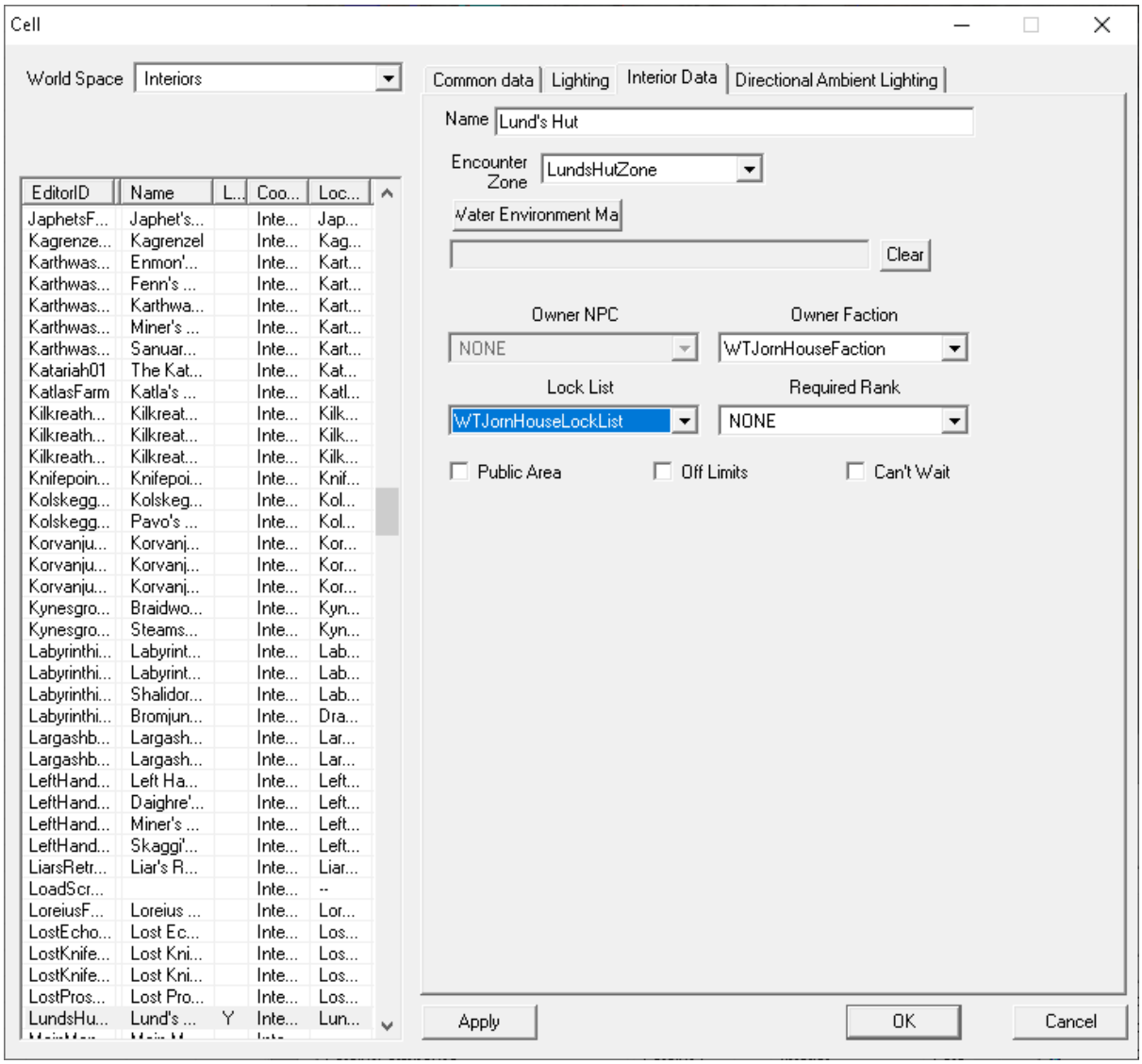


Figure 583 - Setting up cell ownership and assigning a lock list.

Click OK to close out of cell properties.

If we check in-game, we'll now see that the front door to Lund's Hut will be locked.



Figure 584 - Front door locked.

If we break in, Jorn will tell us to leave.



Figure 585 - Jorn giving us a trespass warning.

SETTING UP A BOSS AND MAKING A LOCATION CLEARABLE

In order to make a location 'clearable' we need to flag at least one NPC in that location as a 'boss'. Once all NPCs flagged as a boss in a specific location are dead, the cell will be marked as 'cleared' if it's associated location has the LocTypeClearable keyword.

Let's continue on with the previous example and set Jorn as the new boss of Lund's Hut.

Double-click or right-click on Jorn in the preview window and select Edit.

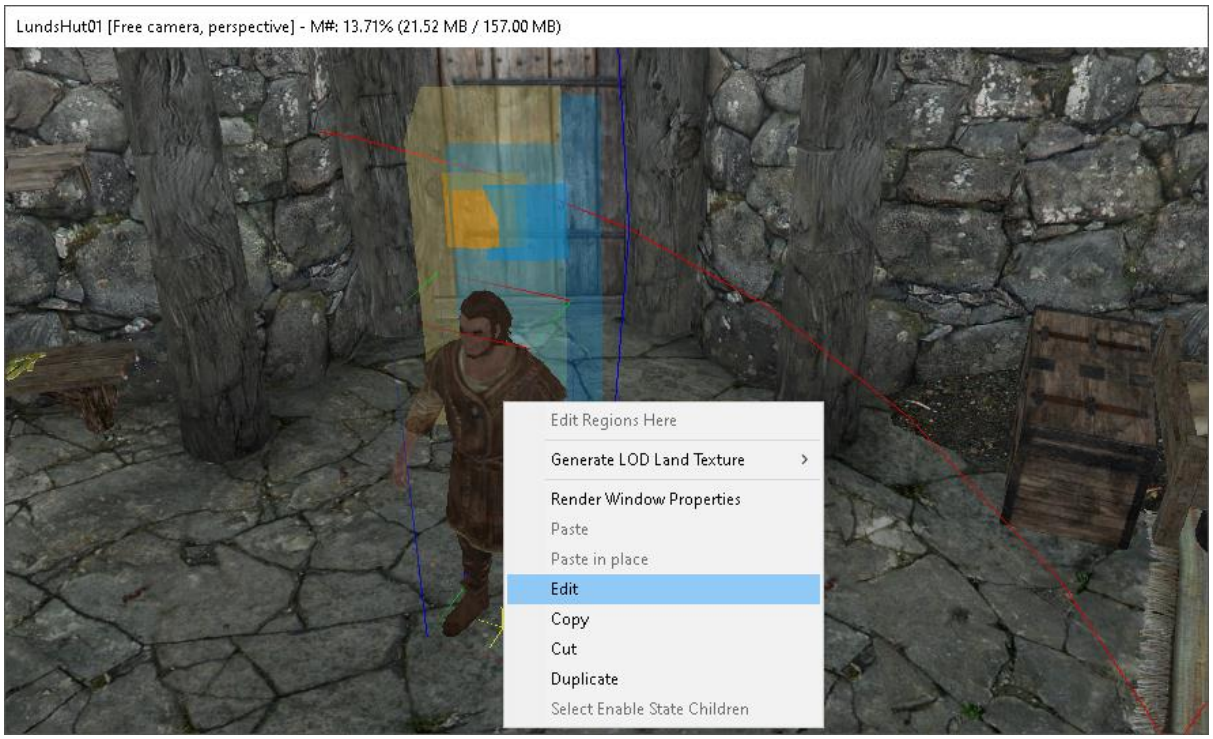


Figure 586 - Editing Jorn's reference.

Go to the Location Ref Type tab and set the Location Reference Type drop-down to 'Boss'.

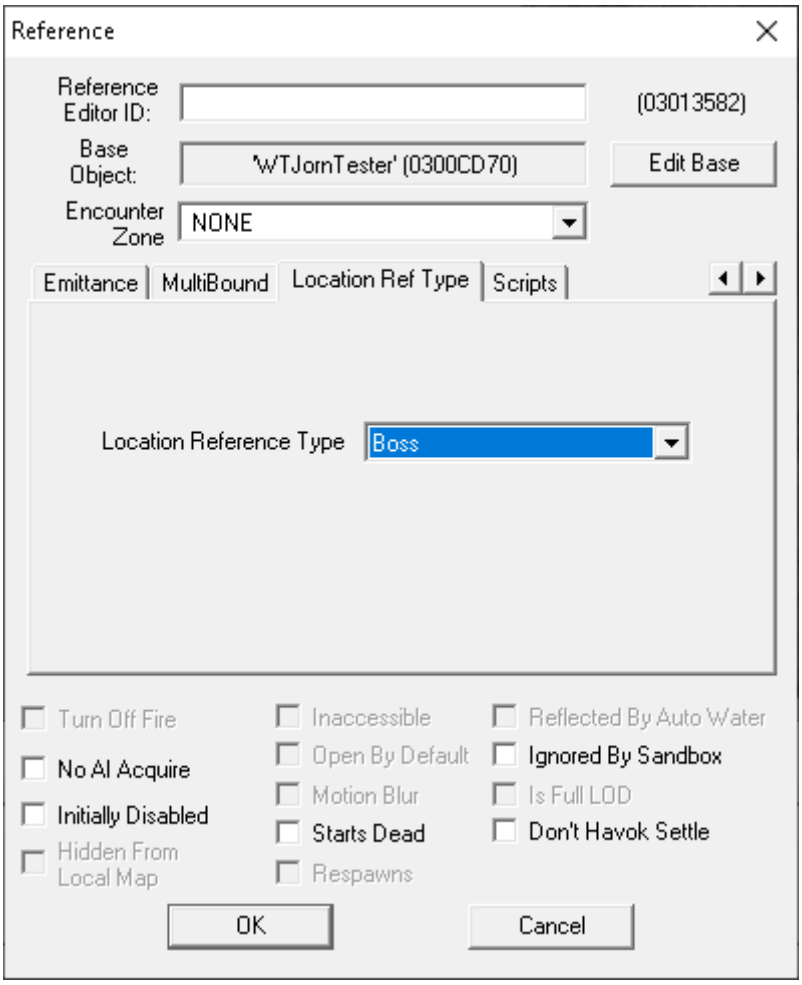


Figure 587 - Setting the location reference type.

Click OK to close.

Lund's Hut is a cell from the base game and already has a location set up for it. To take a look at it, in the Object Window go to WorldData > Location, right-click on LundsHutLocation and select Edit.

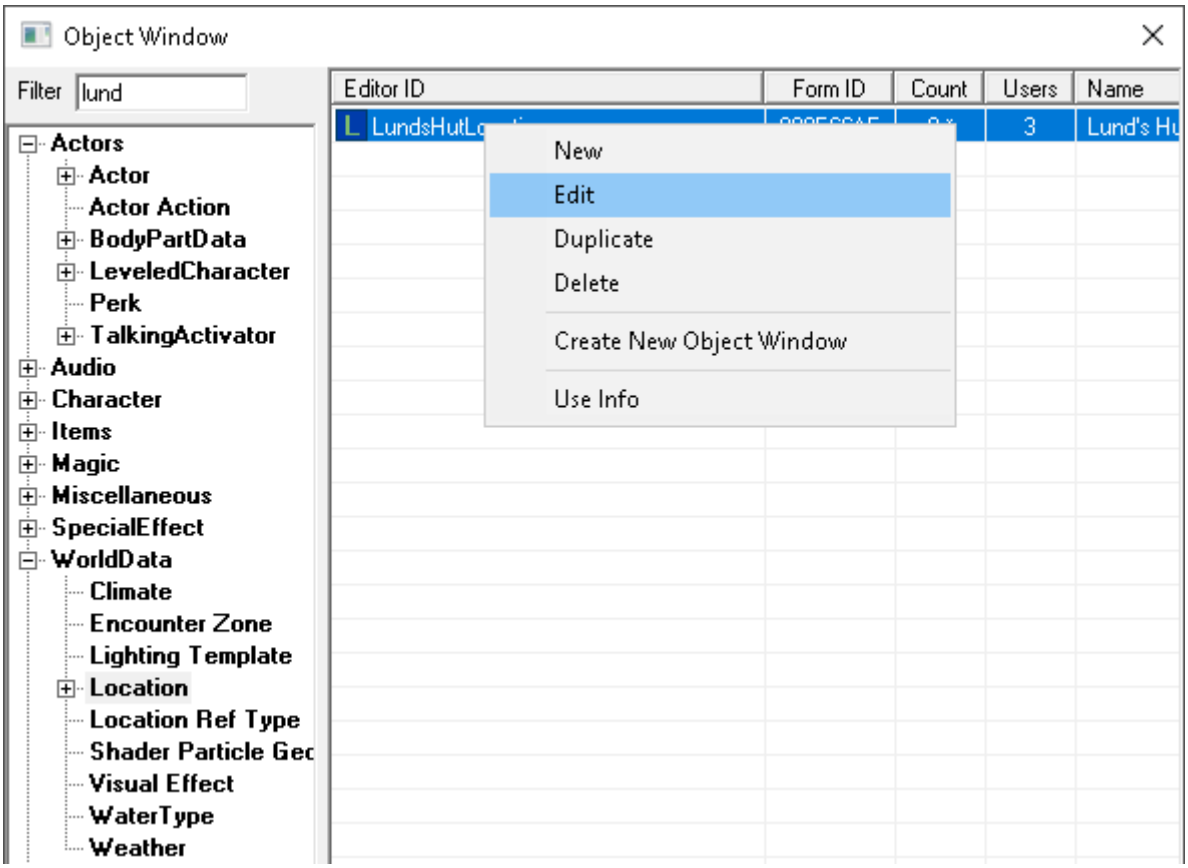


Figure 588 - Editing Location for Lund's Hut.

On the right hand side under the Location Ref Types tab, we can see WTJornTester flagged as the Boss for this location.

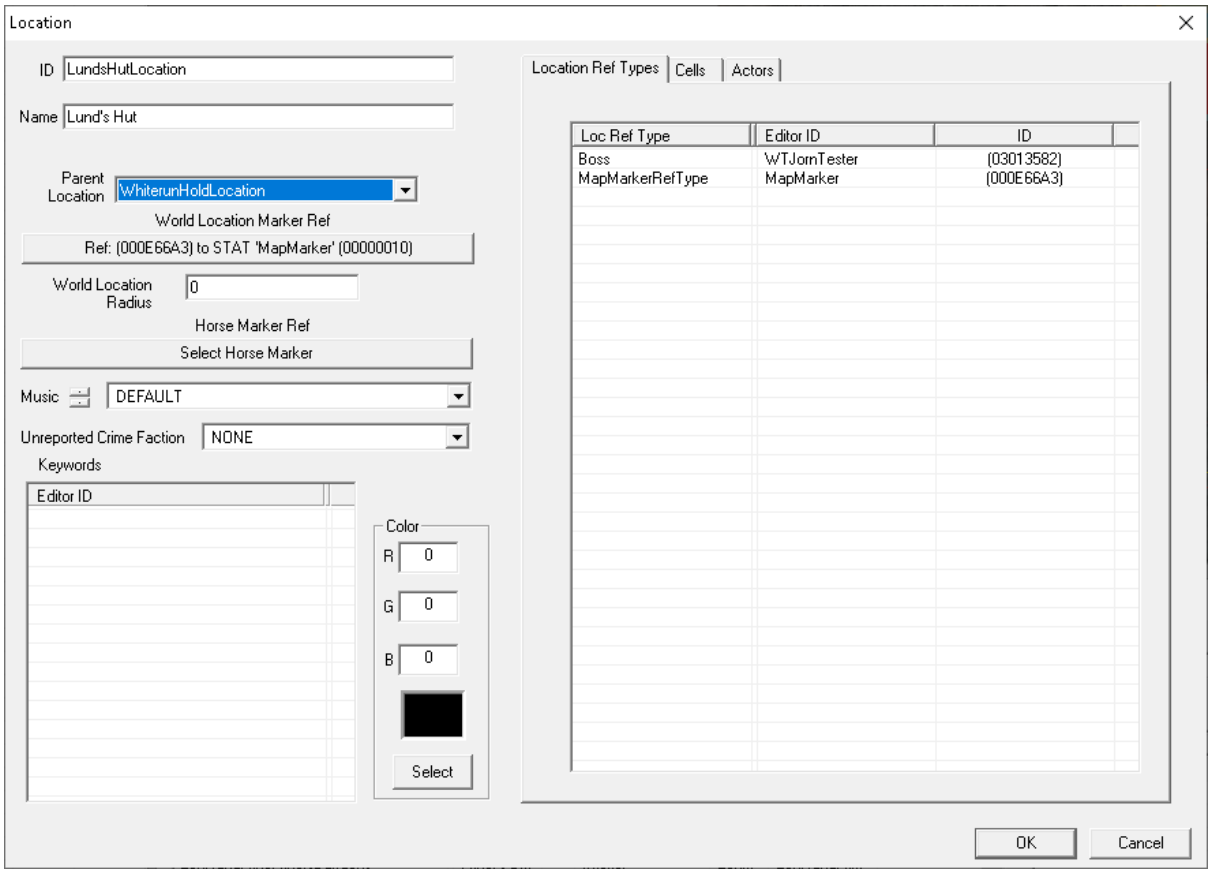


Figure 589 - Location reference types for Lund's Hut.

We'll need to add a keyword for this location to make it clearable.

Right-click in the Keywords list and select Add.

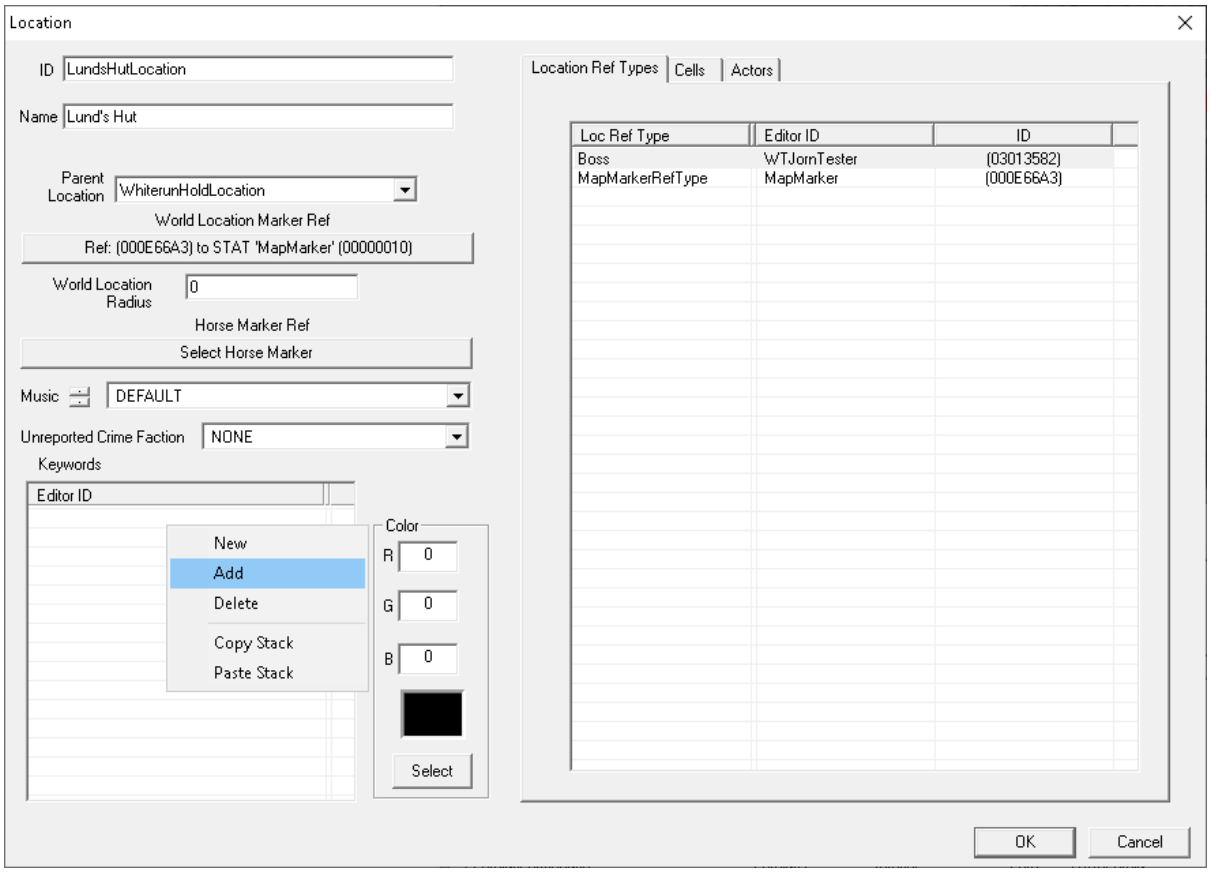


Figure 590 - Adding a new keyword for the Lunds Hut location.

Filter by 'clearable', select LocTypeClearable and click OK.

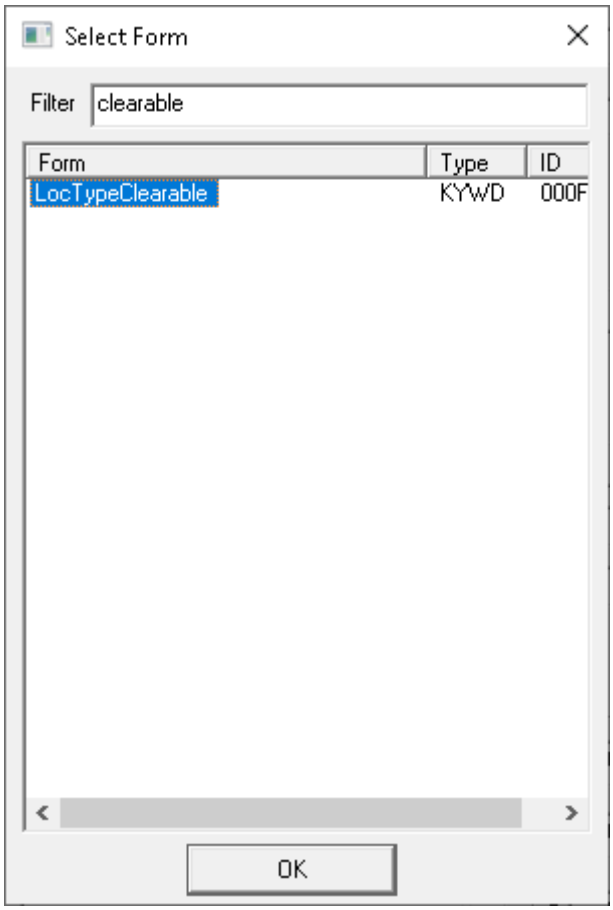


Figure 591 - Adding the LocTypeClearable keyword.

LocTypeClearable should now be listed in the Keywords list.

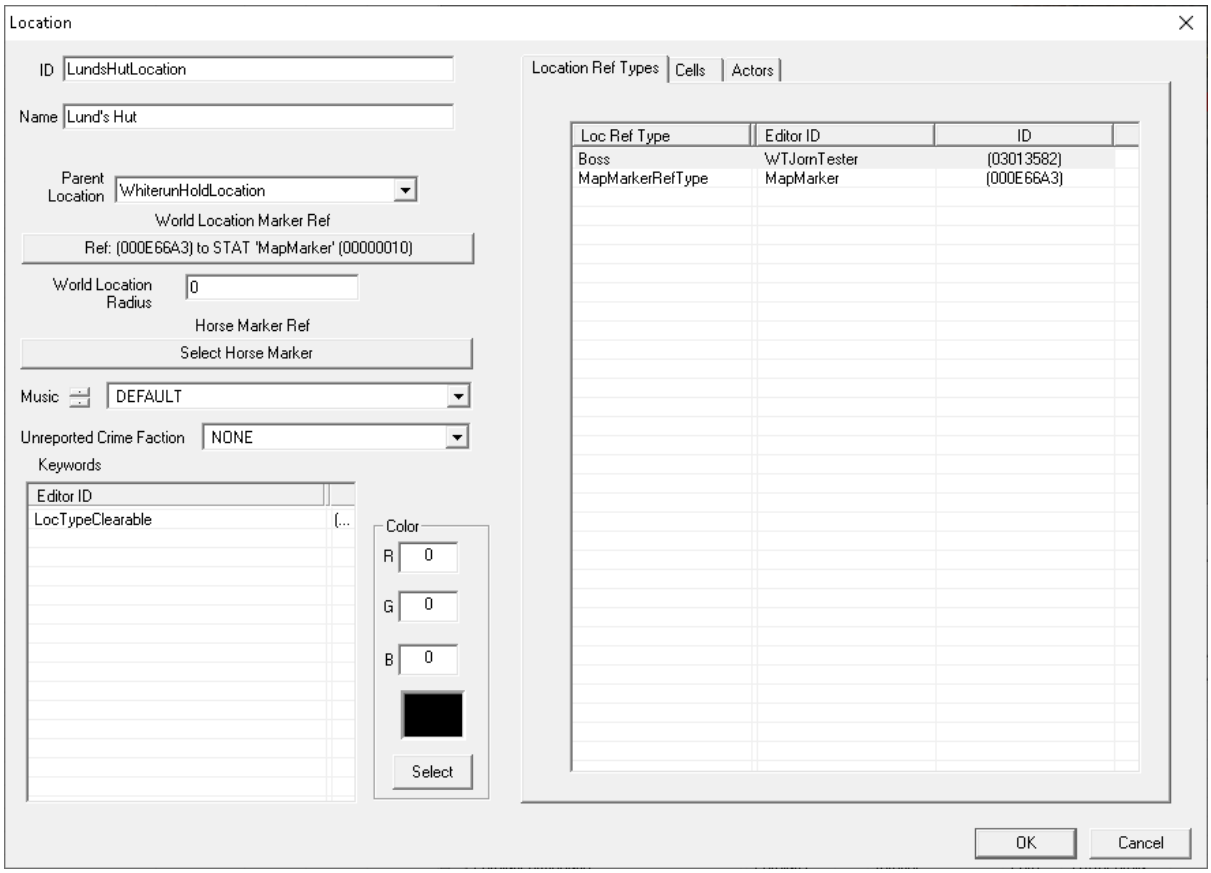


Figure 592 - LocTypeClearable added.

Click OK to close out of Location properties.

If we go in game and kill Jorn, we'll now see the location flagged as cleared on the map screen.



Figure 593 - Jorn is dead.

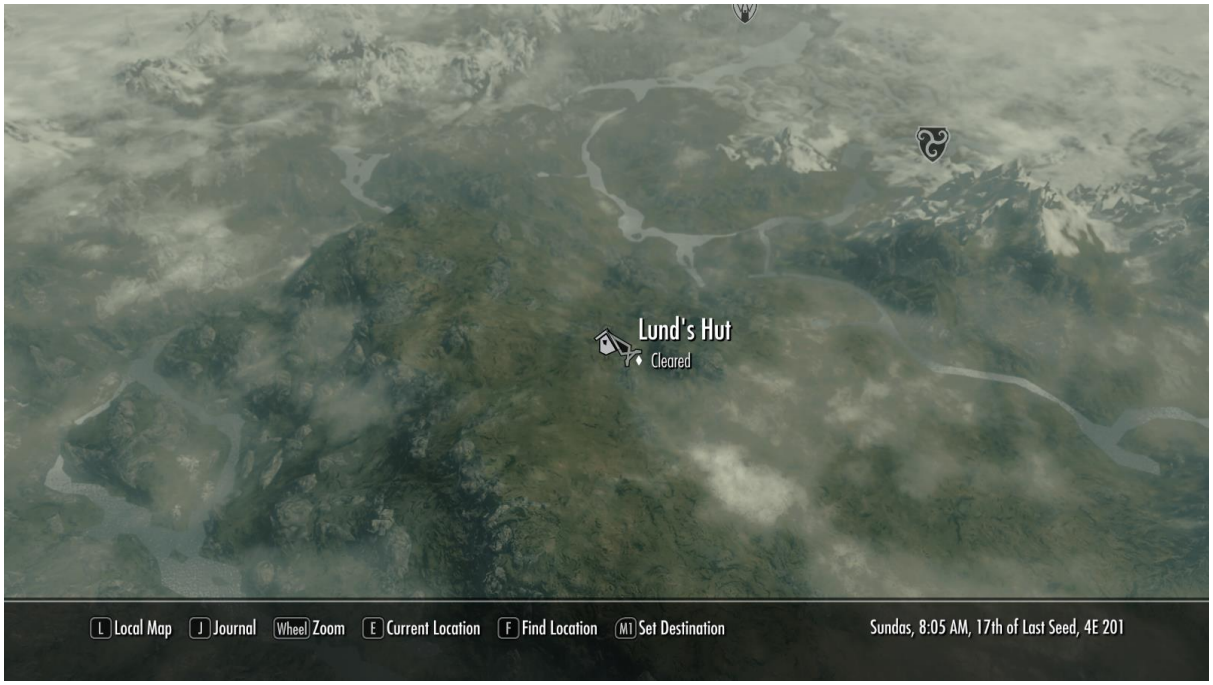


Figure 594 - Lund's Hut flagged as cleared.

And so Jorn the Tester's story comes to an end.

ADDING AMBIENT CREATURES

Let's start adding a bit of life to our world space.

In the Object Window under Actors, filter by 'lvl*prey'. This will give us five different ambient animal spawners. These will spawn in creatures like goats, foxes, rabbits, deer, etc. Drag and drop one of them into the current cell.

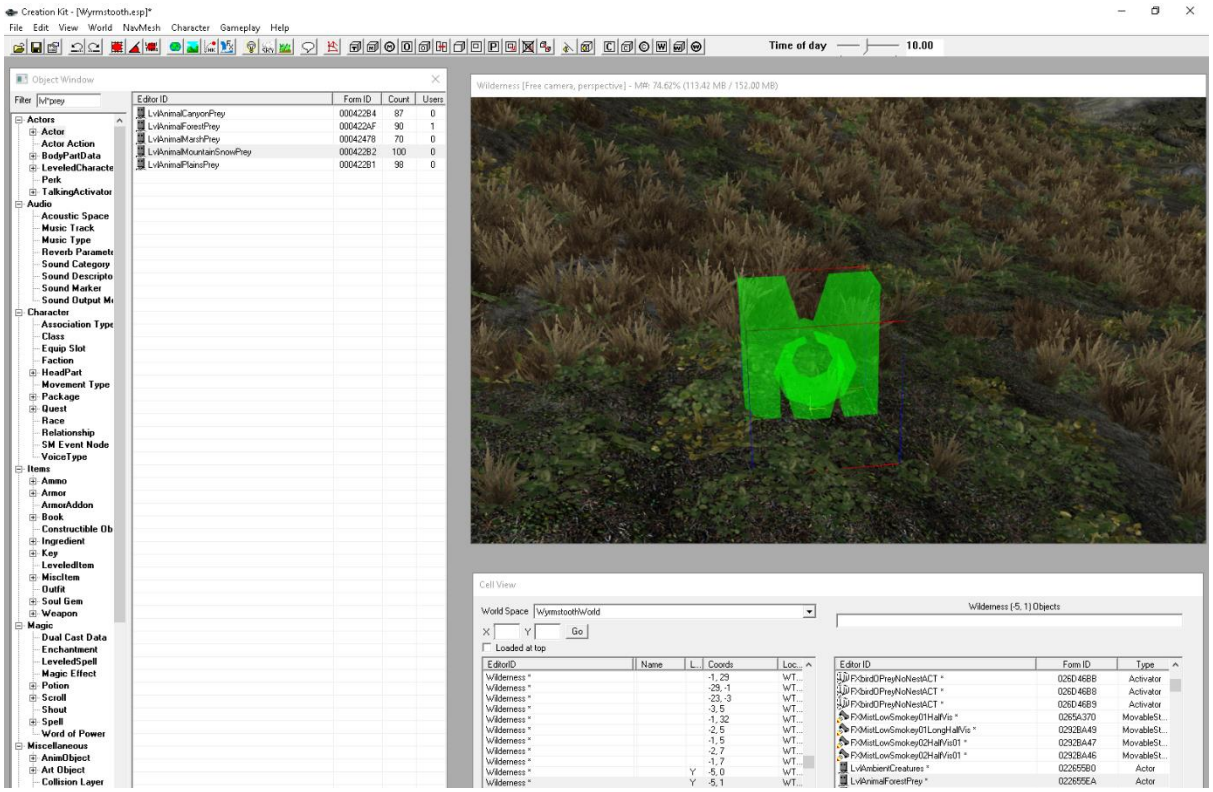


Figure 595 - Ambient creature spawner.

Levelled actor spawners are shown as an M with an arrow pointing in the direction the marker is facing. Their colour depends on the difficulty scaling.

White = No scaling.
Green = Easy.
Yellow = Medium.
Orange = Hard.
Red = Very Hard.

To add a predator spawner, filter by 'lvl*predator' then drag and drop one into the scene.

To increase the difficulty of the spawned creature, double-click on its marker in the render window, go to the Leveled Actor tab, and change its Level Modifier to Medium.

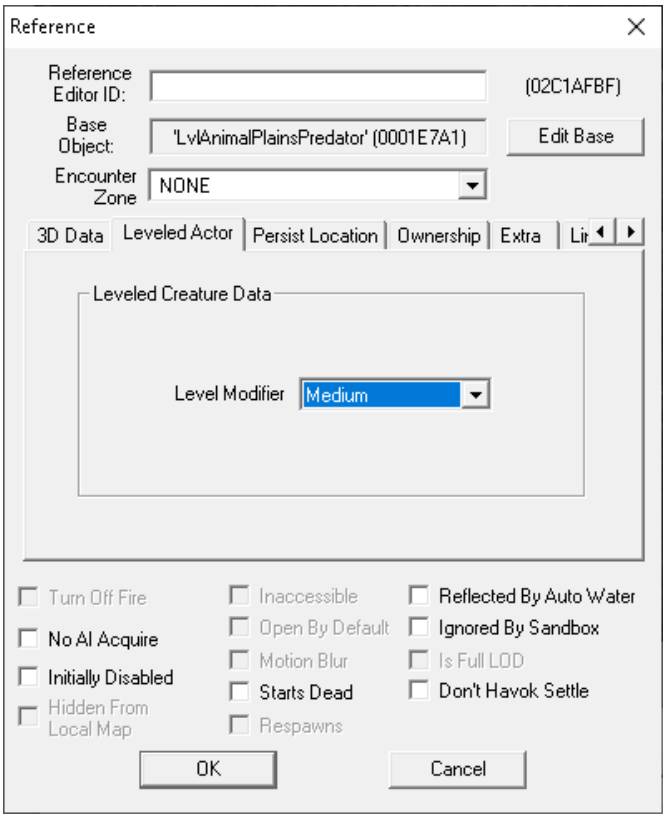


Figure 596 - Leveled Actor tab.

Click OK. The colour of its marker should change.

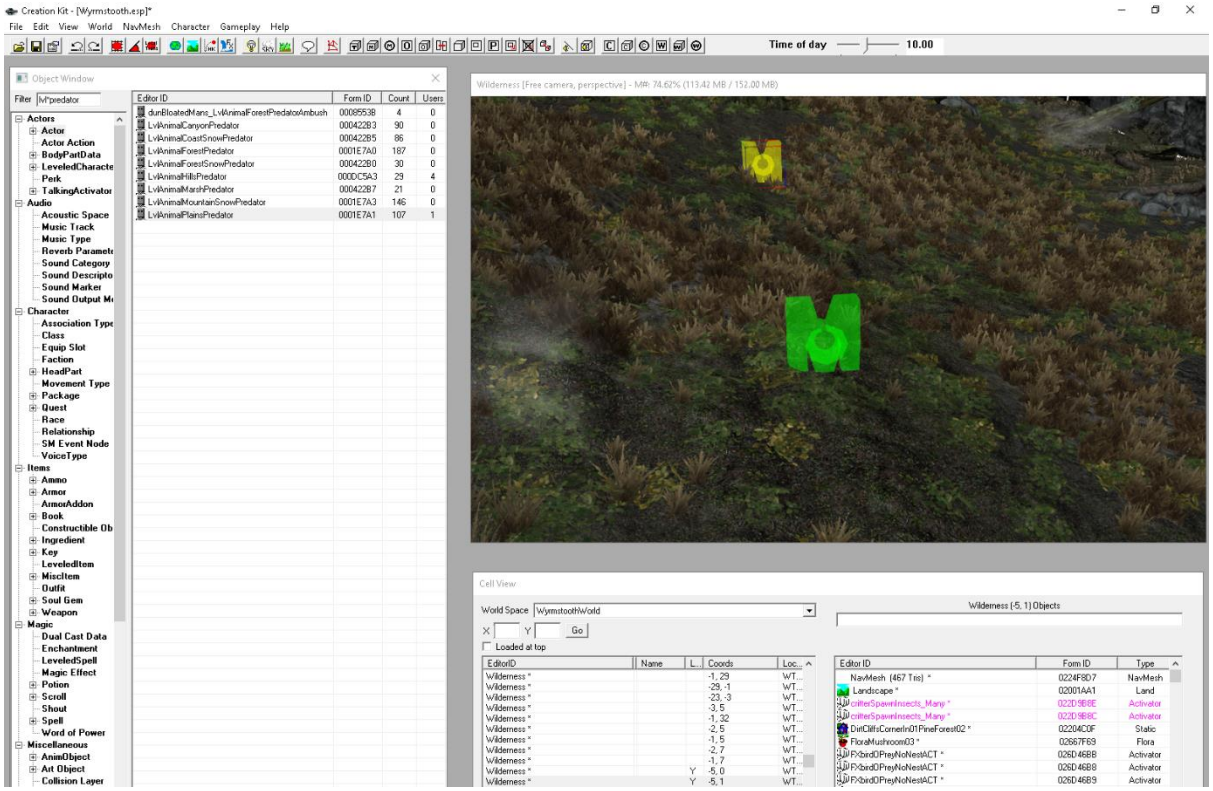


Figure 597 - Predator spawner.

Predators include aggressive creatures such as wolves, bears and sabre tooth tigers. They will hunt prey upon being spotted, including the player and other NPCs.

We can spawn in insects like butterflies, dragonflies and bees using a critter spawner. In the Object Window, browse to WorldObjects > Activator and filter by 'critterspawn'. Drag and drop a spawner such as critterSpawnInsects_Many into the render window. Critter spawners appear as a C above a green and blue sphere.

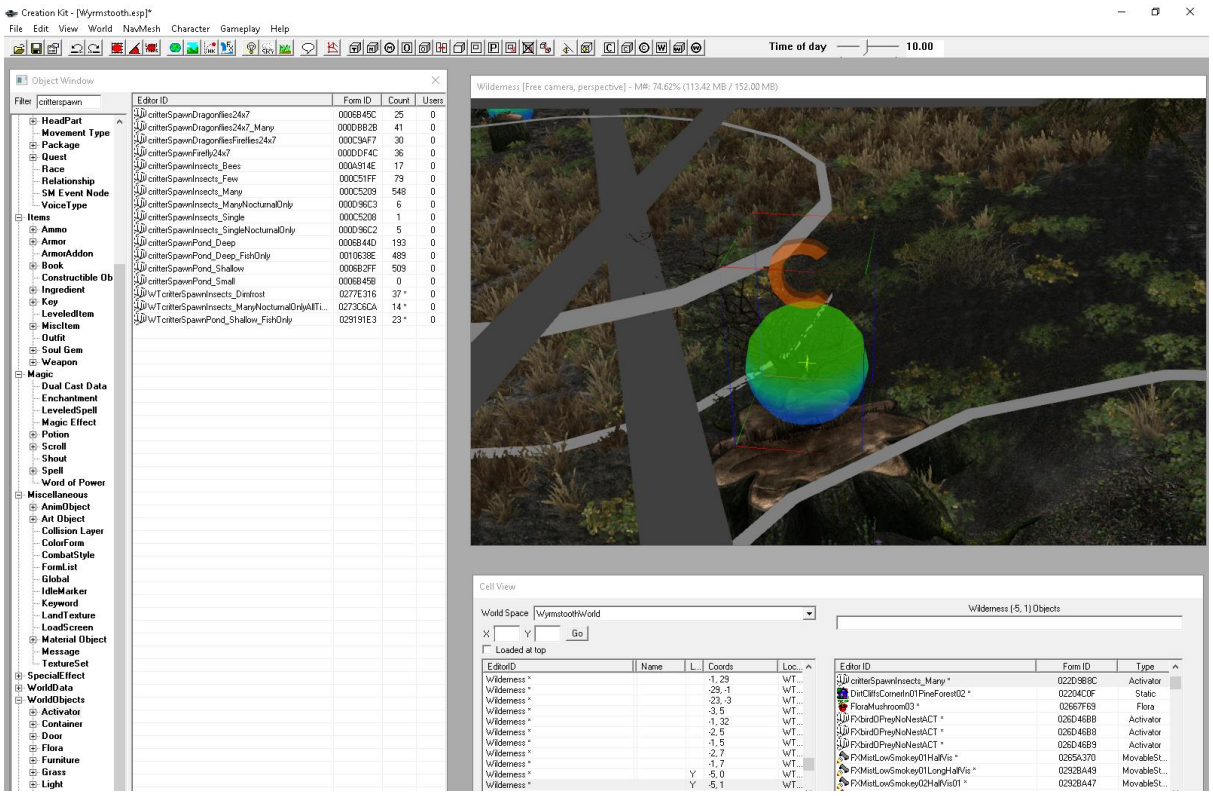


Figure 598 - Critter spawner.

Next, let's add some landing markers. They are used by butterflies to note spots where they are allowed to land.

In the Object Window, browse to WorldObjects > Static and filter by 'critterlanding'. Drag and drop CritterLandingMarker_Small into the render window.

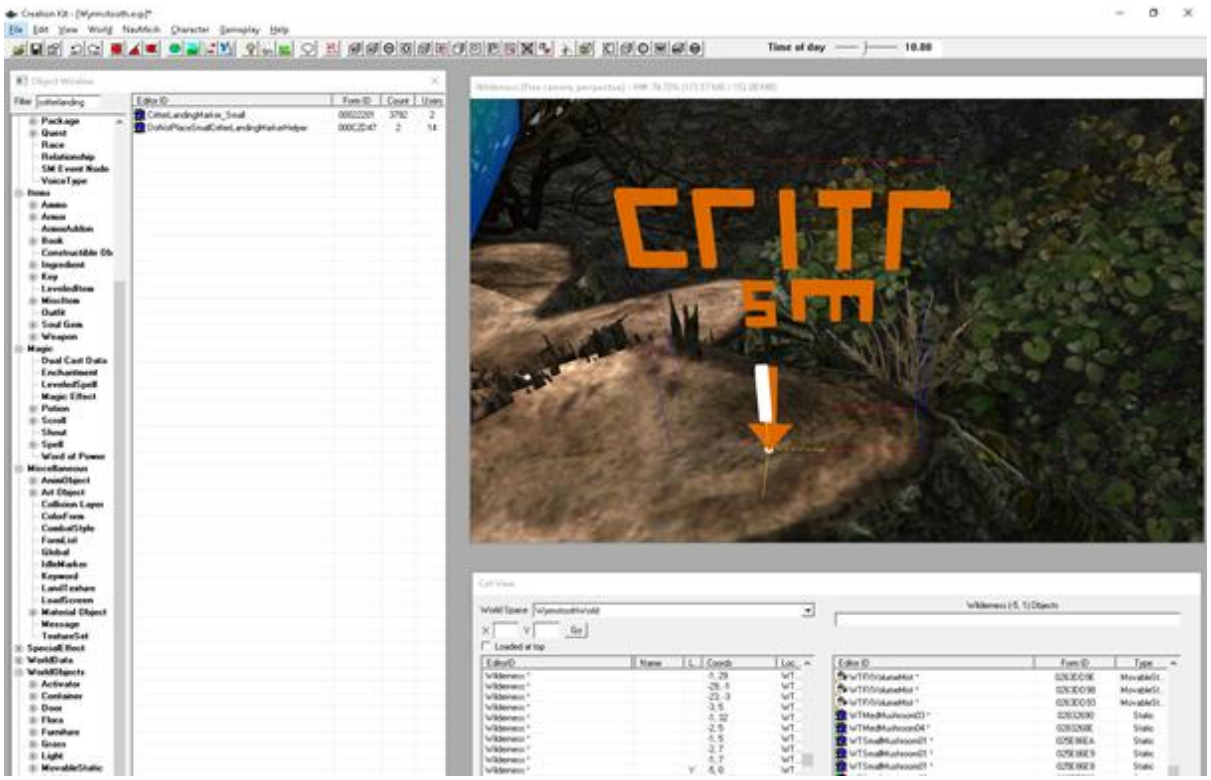


Figure 599 - Critter landing marker.

Critter landing markers are marked with an arrow pointing to a spot where a critter is allowed to land on. I usually place them on unobscured rock ledges, sign posts, building edges or on leaves of bushes where butterflies won't have to clip through solid objects to reach them.

Use the critterSpawnPond_* critter spawners to add fish to ponds and lakes.

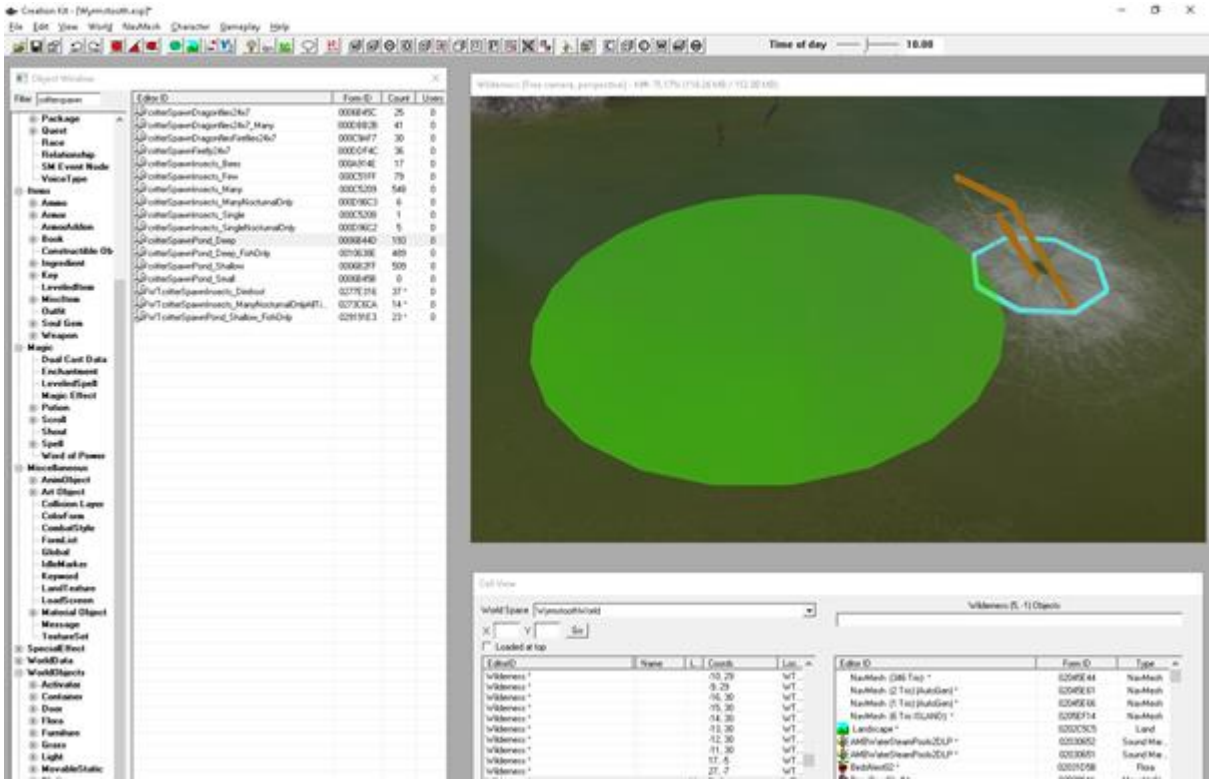


Figure 600 - Fish spawner.

They show up as an upside down green cone in the render window. Make sure the top of the cone is flush with the surface of the water, and the bottom of the cone doesn't clip through the ground.

To add birds, in the Object Window browse to WorldObjects > Activator and filter by 'fxbirdofprey'.

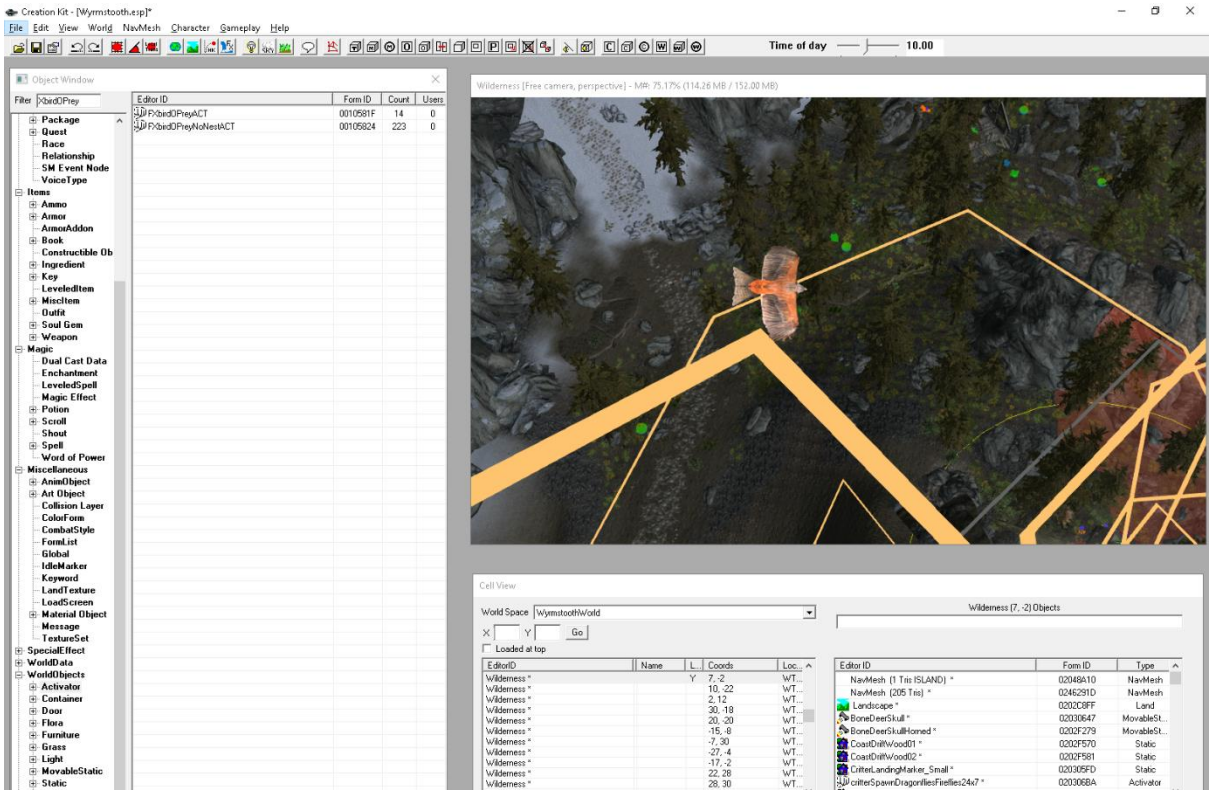


Figure 601 - Birds.

FXbirdOPreyNoNestACT adds free-flying birds, whereas FXbirdOPreyACT adds birds that will return to a nest that you can place in a tree or on top of a building.

CHAPTER 5: RANDOM ENCOUNTERS

ADDING RANDOM ENCOUNTER TRIGGERS

Random encounter triggers are used to spawn in NPCs from a roster of random encounters defined in the Story Manager.

For example, while traveling the road you may run into Talsgar the Wanderer, or a giant with a painted cow, or thalmor leading a prisoner, and so on.

We can add these random encounters to our own world space too, as well as set up new random encounters.

Let's start by adding a new random roadside encounter. The easiest way to do this is to copy an existing setup from the WarehousePrefabs interior cell.

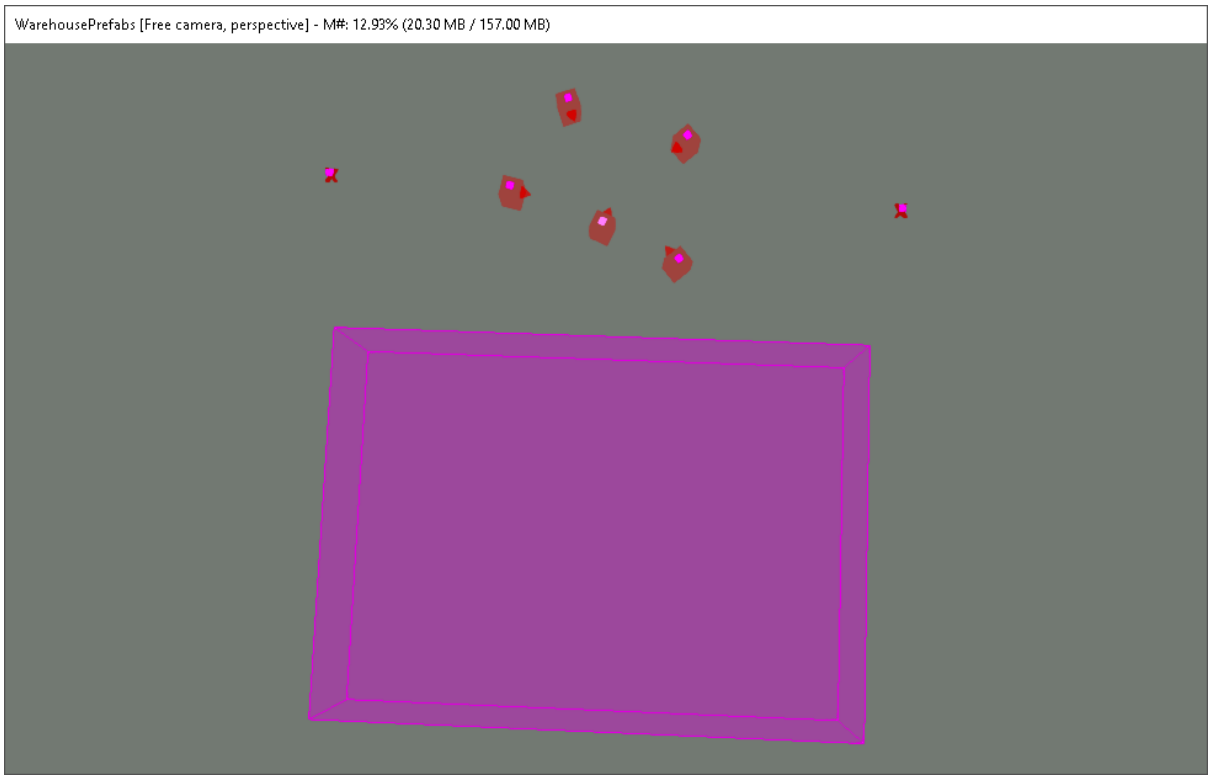


Figure 602 - Random roadside encounter setup in WarehousePrefabs.

The random roadside encounter will consist of five XMarkerHeadings, two XMarkers, and a WETrigger box. Select them and press CTRL + C to copy or go to Edit > Copy Render.

WETrigger is also linked to a secondary trigger box, but in most cases we shouldn't need it.

Basically when the cell containing the WETrigger box loads, a random encounter is generated, placing NPCs at the XMarkers or XMarkerHeadings.

NPCs that travel the road, such as the thalmor leading a prisoner or stormcloak/imperial patrols, will walk back and forth between the two XMarkers, so you'd typically want to place these far apart from each other.

NPCs that stand in place, such as scavengers with dead bodies or the orc wishing for a good death, will stand at the XMarkerHeadings.

Press CTRL + V to paste in the copied random encounter setup.

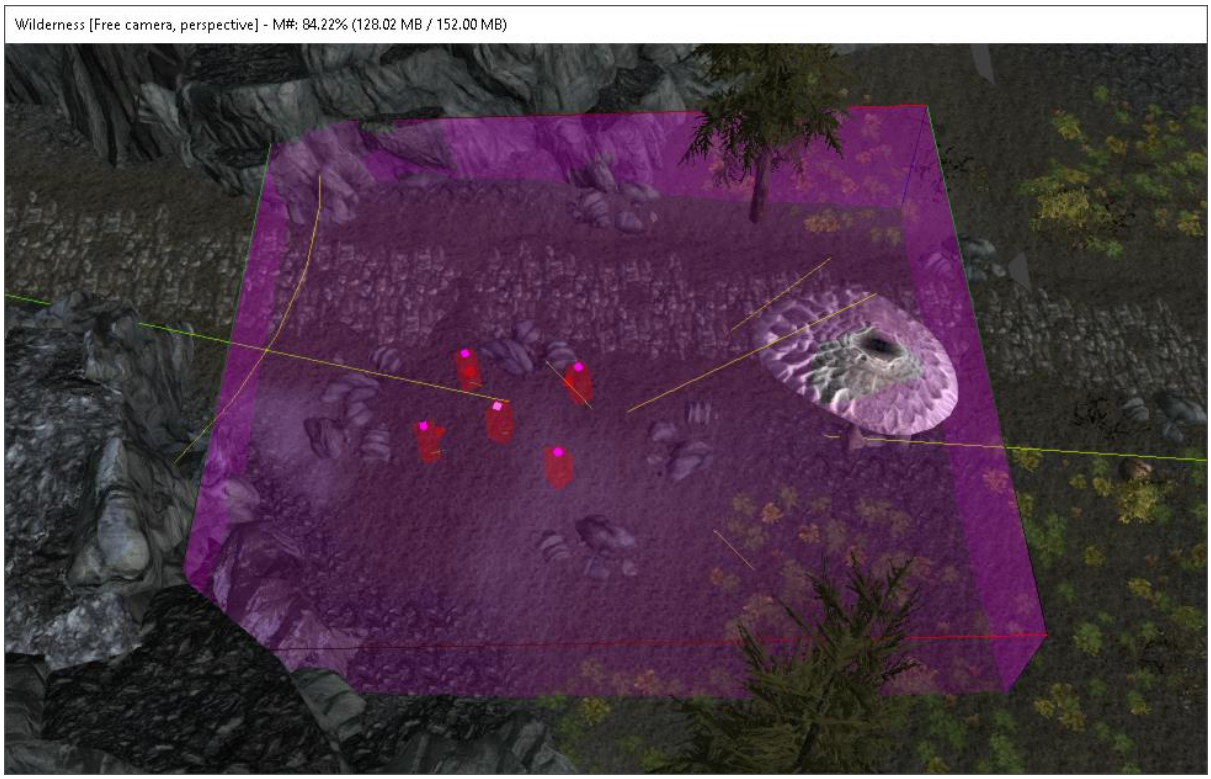


Figure 603 - A random roadside encounter placed in a world space.

The screenshot above shows an example of a random roadside encounter placed in a world space. I put the XMarkerHeadings just to the side of the road and moved the two XMarkers up and down the road.

Once you positioned the random roadside encounter in your world space, double-click on WETrigger to bring up its properties.

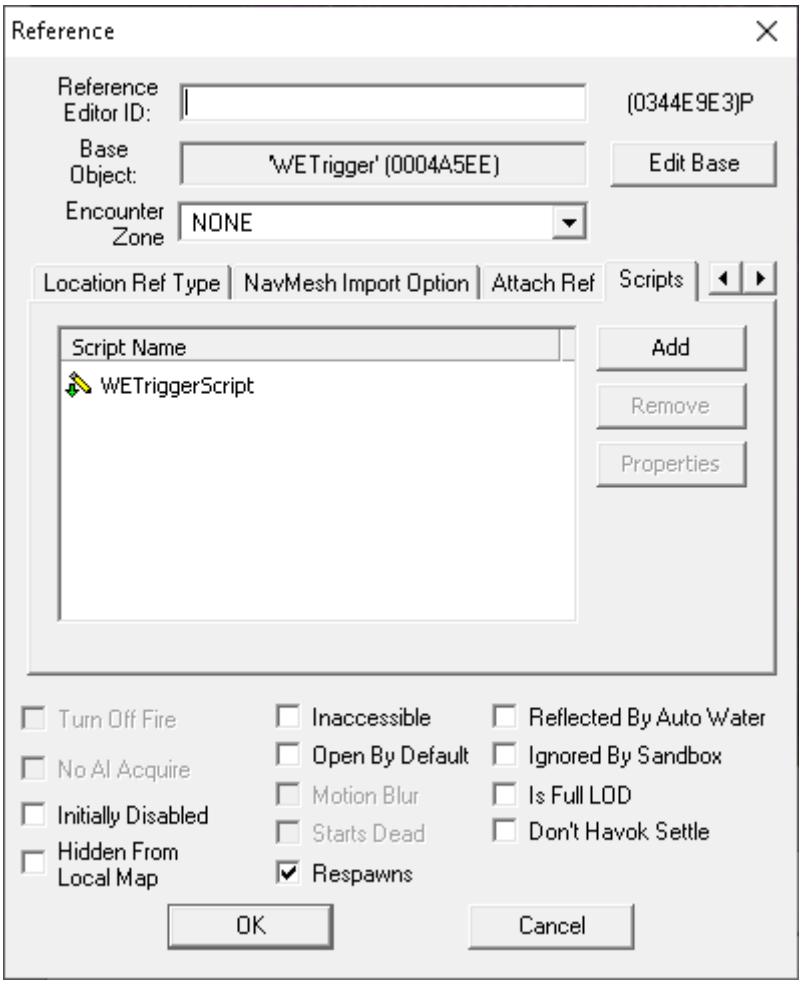


Figure 604 - WETrigger Reference.

Go to the Scripts tab, highlight the WETriggerScript and click on the Properties button.

Set RoadEncounter to TRUE, and TriggerOnLoad to TRUE.

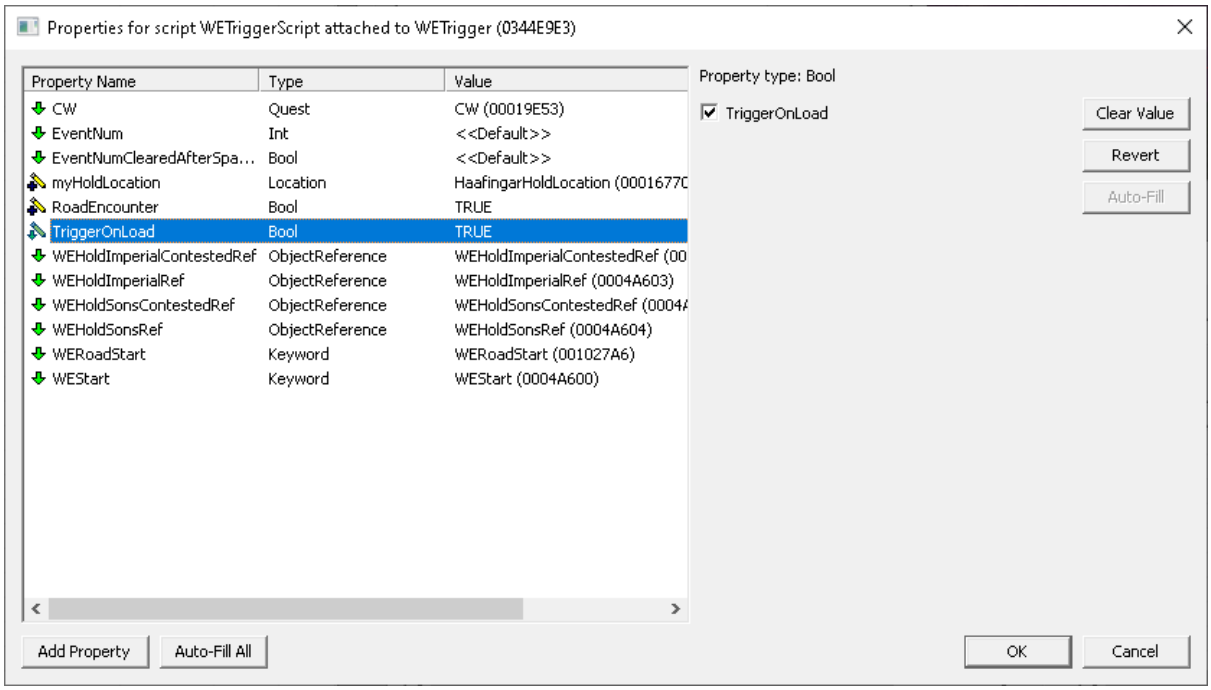


Figure 605 - WETriggerScript properties.

Later we'll need to change WERoadStart and WEStart to point to our own keywords, but for now let's just leave them set at the defaults.

myHoldLocation can also be left at HaafingarHoldLocation for now.

Click OK when you're finished.

Next, let's add a random wilderness encounter. These spawn encounters that don't travel a road, such as two wizards duelling, spriggan vs. hagraven, and so on.

Go back to the WarehousePrefabs cell.

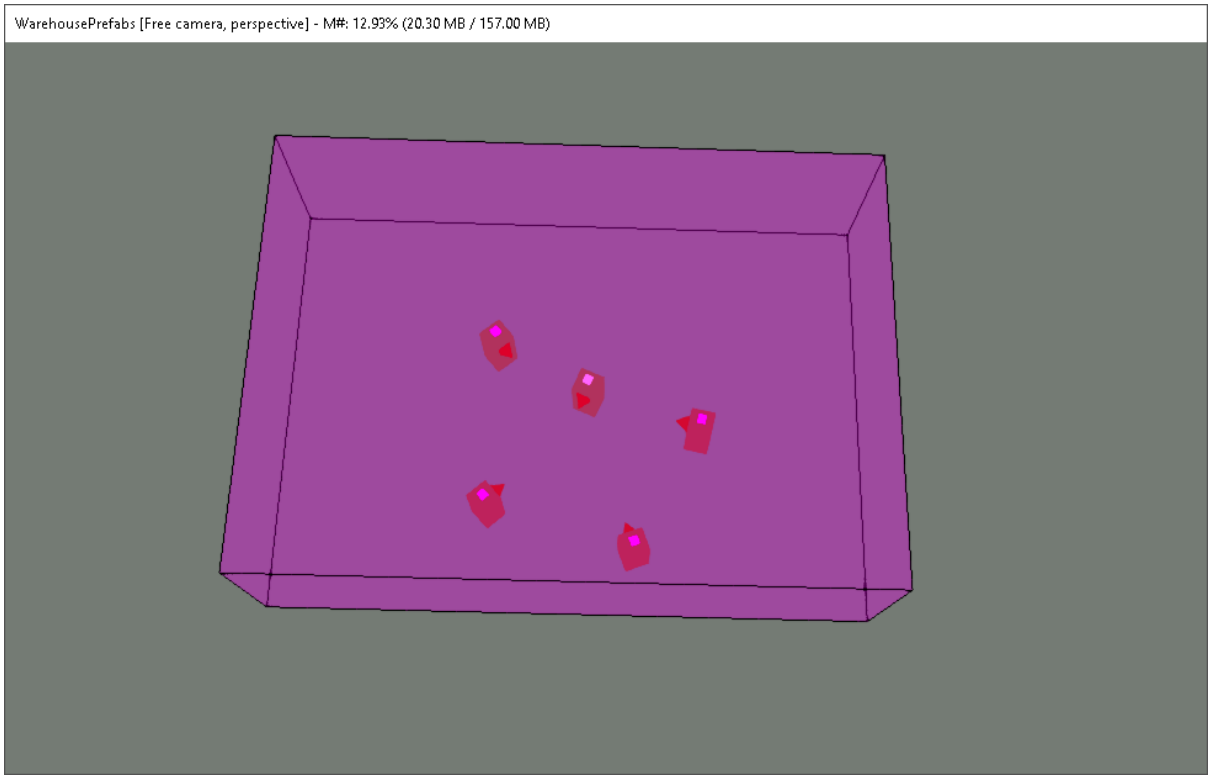


Figure 606 - Random wilderness encounter.

The random wilderness encounter will consist of five XMarkerHeadings and a WETrigger box.

Select them and press CTRL + C to copy or go to Edit > Copy Render.

Press CTRL + V to paste in the copied random encounter.

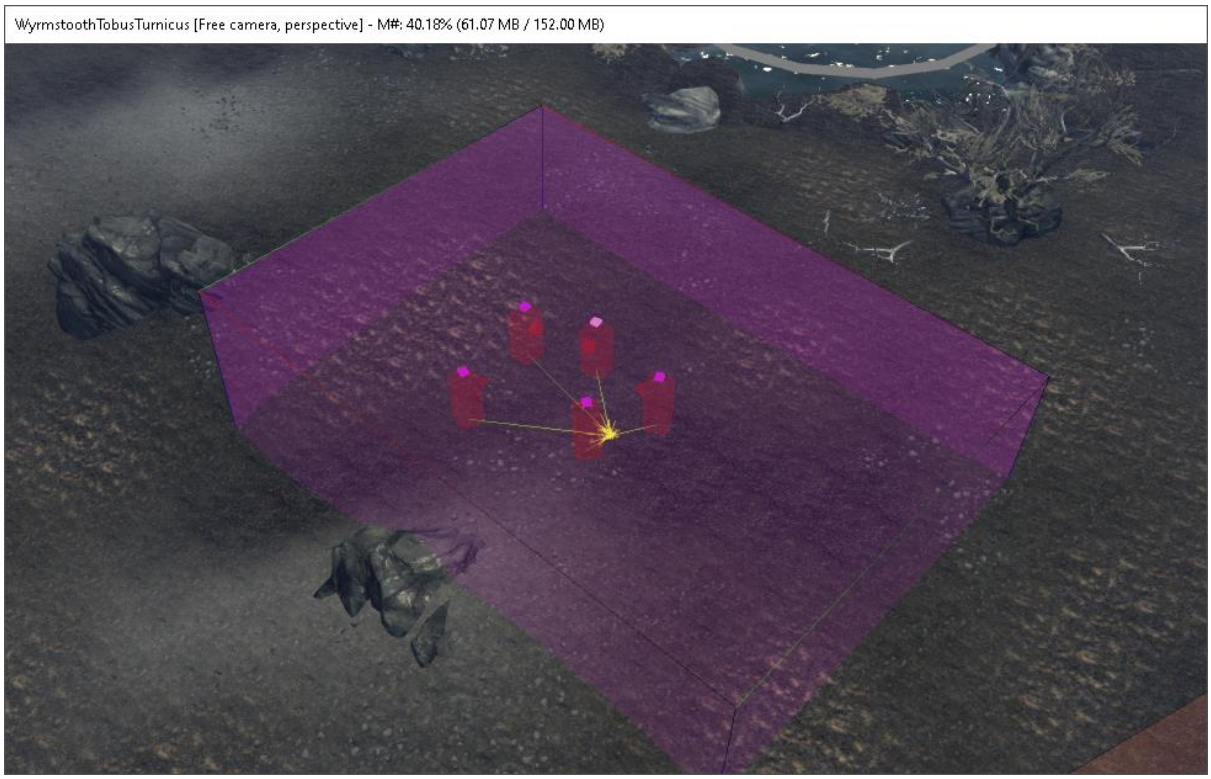


Figure 607 - A random wilderness encounter placed in a world space.

The screenshot above shows an example of a wilderness encounter placed in a world space.

Right-click on the WETrigger and select Edit or double-click on the WETrigger to bring up its properties.

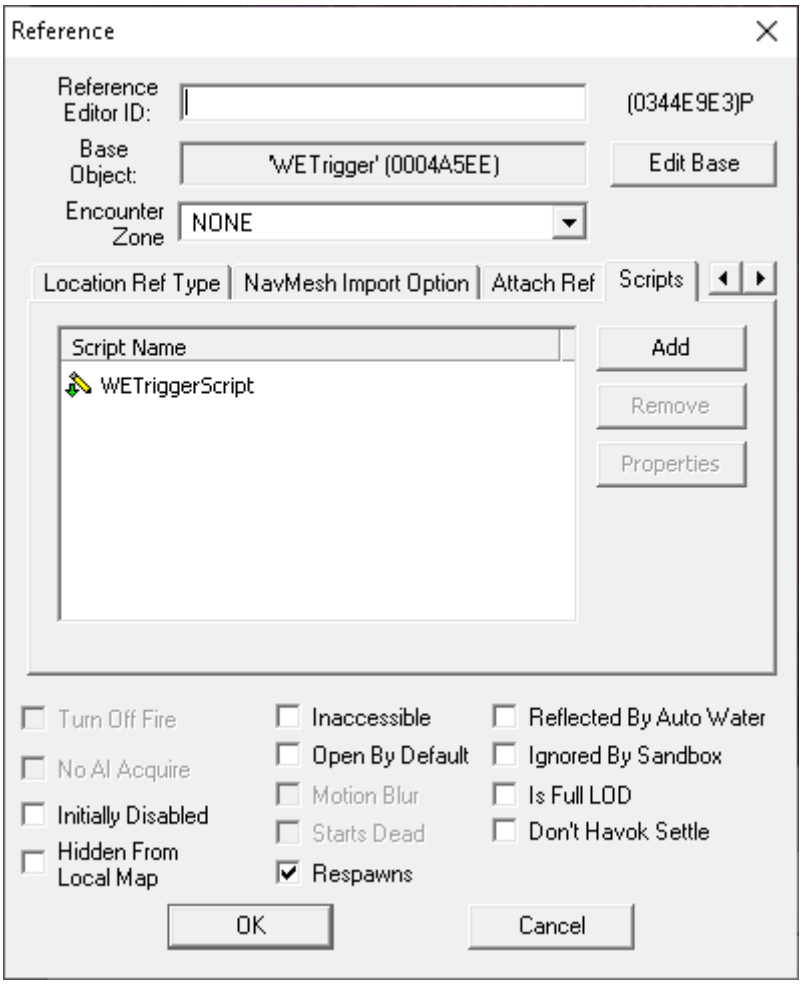


Figure 608 - WETrigger Reference.

Go to the Scripts tab, highlight the WETriggerScript and click on the Properties button.

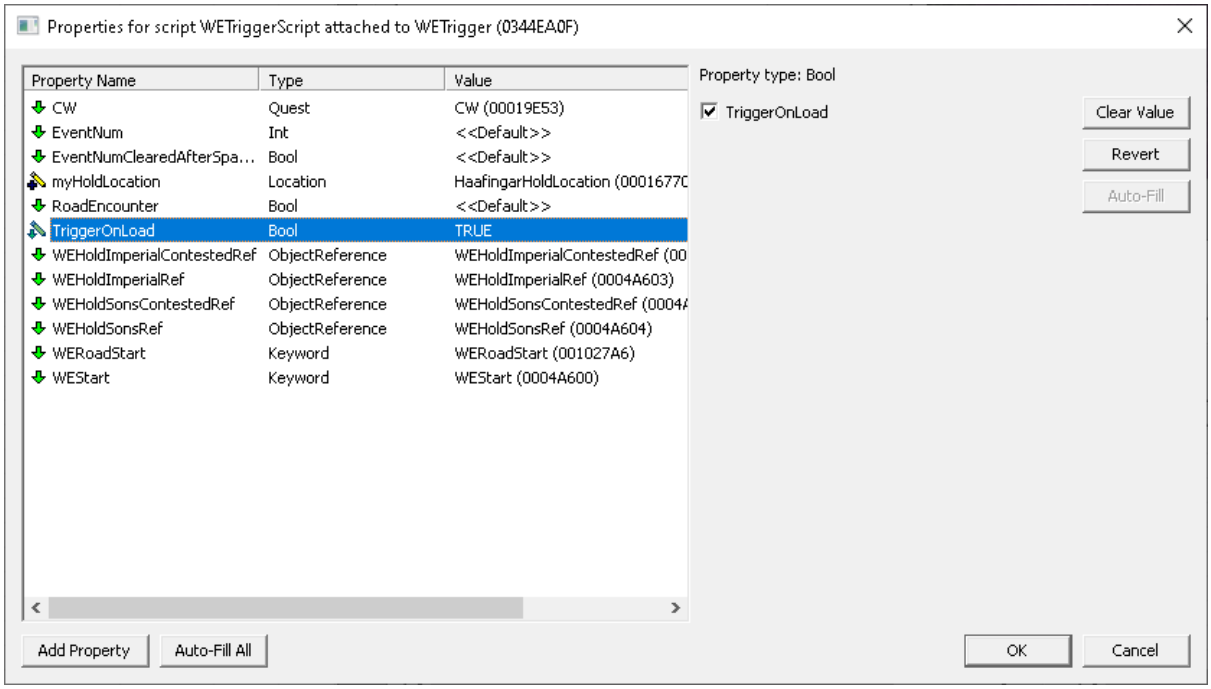


Figure 609 - Random wilderness encounter WETriggerScript properties.

For random wilderness encounters, we just need to set TriggerOnLoad to TRUE. Keep RoadEncounter set to <<Default>>.

Again, we'll need to come back later and set WERoadStart and WEStart to point to our own list of random encounters, but for now we'll leave these at the defaults.

Click OK when you're finished.

CREATING OUR OWN LIST OF RANDOM ENCOUNTERS IN THE STORY MANAGER

By default, the random encounter spots we added are generating encounters based on the WEQuests and WERoadQuests lists in the Story Manager.

In the Object Window, go to Character > SM Event Node and open Script Event.

Expand WEQuestNode to show WEQuests and WERoadQuests.

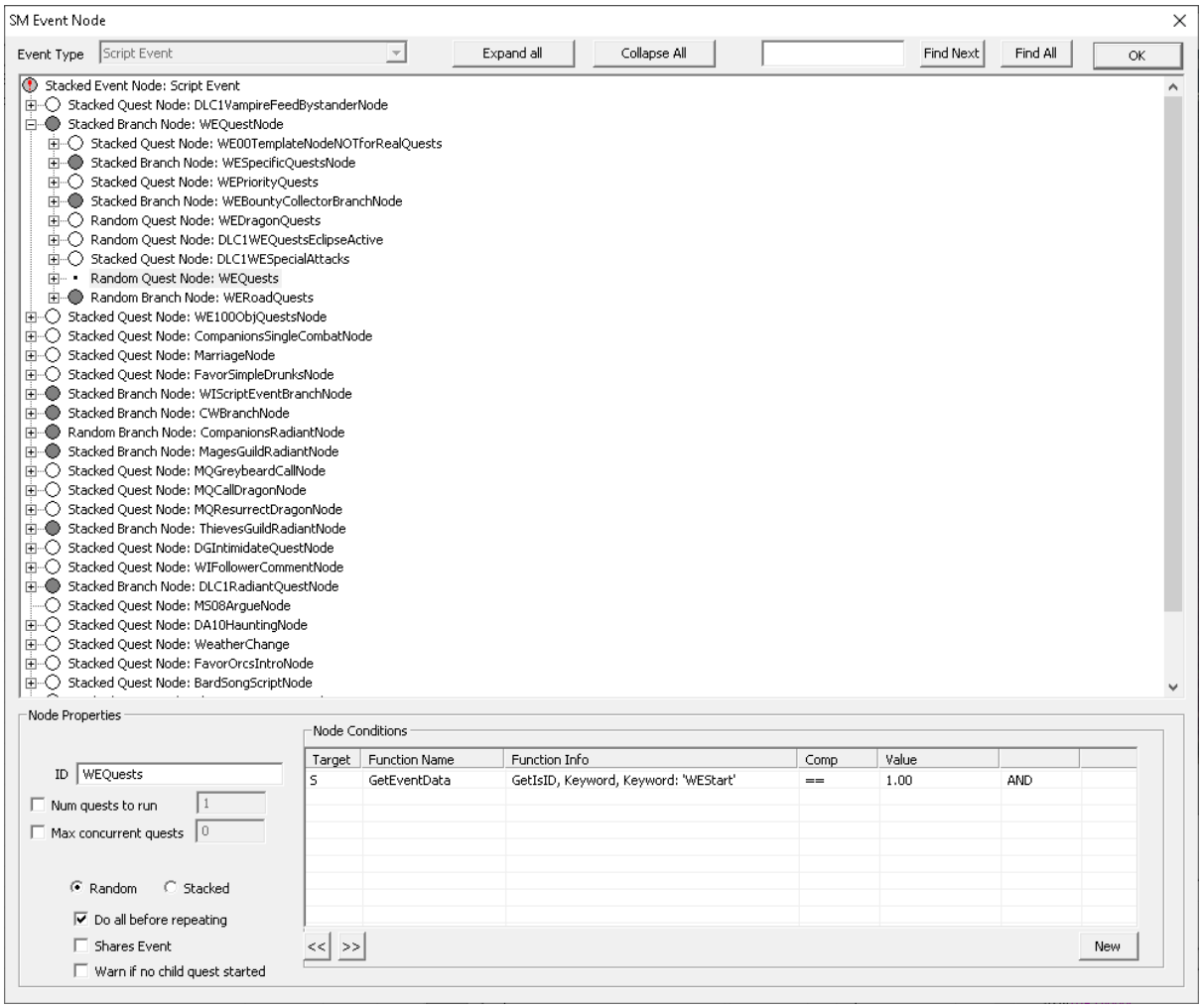


Figure 610 - WEQuests in the Story Manager.

If you click on WEQuests or WERoadQuests you'll see the keyword condition under Node Conditions. WEQuests has a keyword condition for 'WEStart' and WERoadQuests has a keyword condition for WERoadStart.

If you expand WERoadQuests and expand the WERoadQuestNode node beneath it, you'll see a list of all possible random roadside encounters.

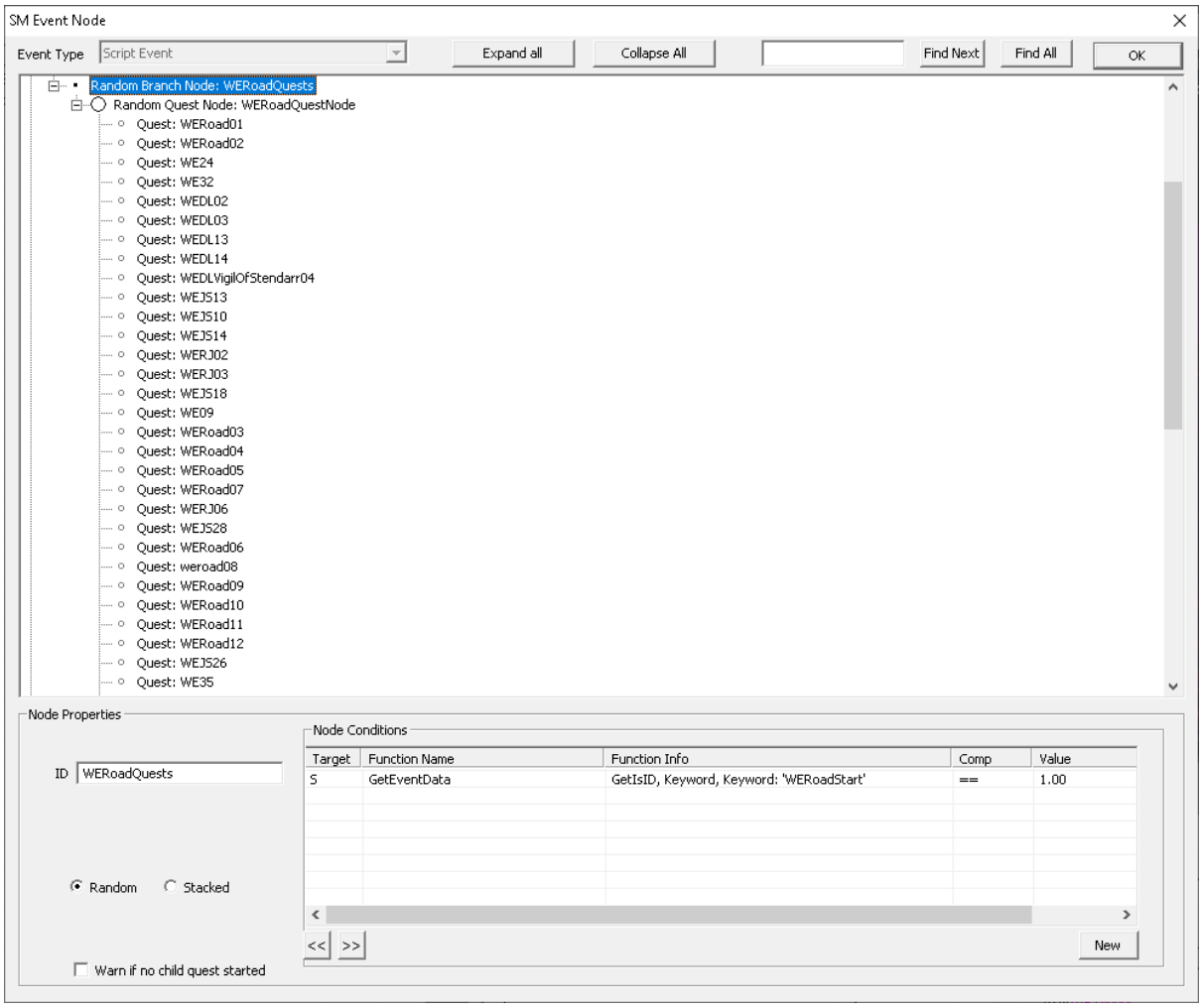


Figure 611 - The list of random roadside encounters.

Similarly, if you expand WEQuests you'll see a list of all possible random wilderness encounters.

Let's get started on adding our own random wilderness and random roadside encounter branches for our custom world space.

The first thing we need to do is create new keywords for our own wilderness and roadside encounters.

In the Object Window go to Miscellaneous > Keyword.

Right-click on one of the existing keywords and select New.

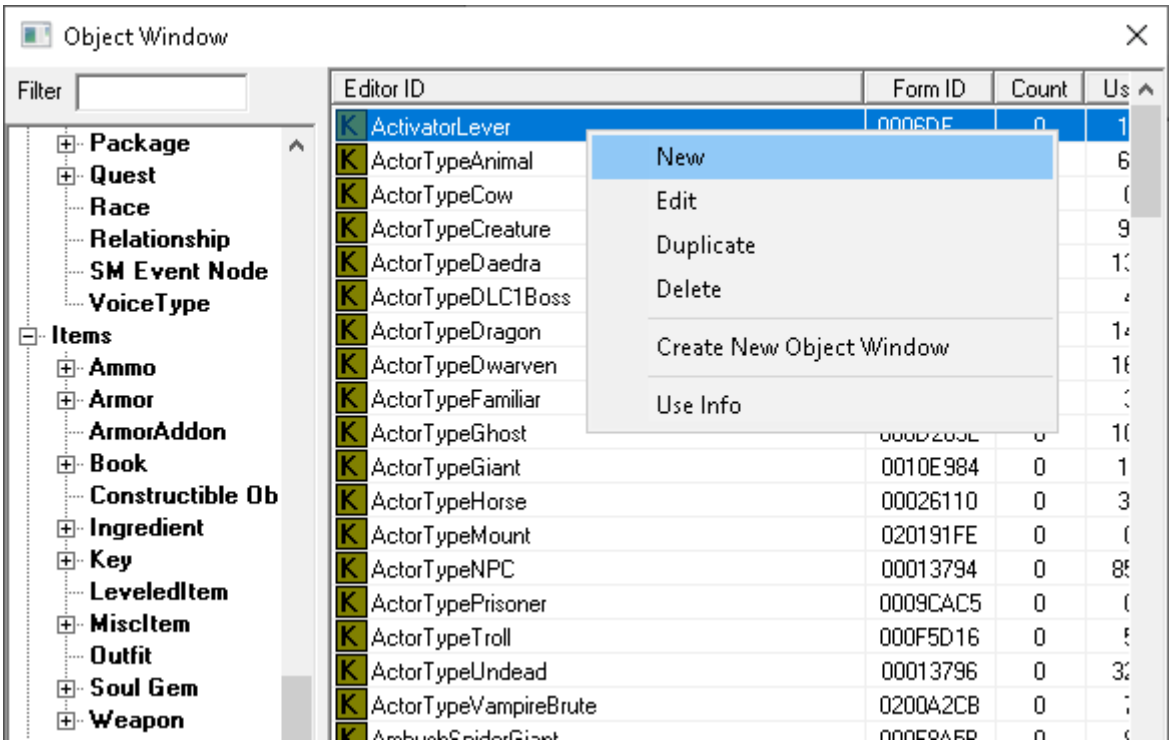


Figure 612 - Adding a new keyword.

Using Wyrmstooth as an example, I kept the keyword names identical to the default keywords, but with an added 'WT' prefix. E.g.: WTWERoadStart for my own random roadside encounters.

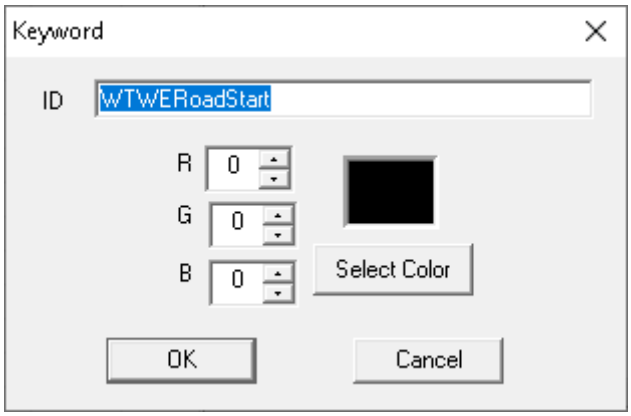


Figure 613 - Naming a new keyword.

Click OK to create a new keyword.

Repeat these steps to create a new keyword for wilderness encounters. E.g.: WTWEStart.

Once you've created two new keywords, we can continue.

Back in the Story Manager, right-click on 'Stacked Event Node: Script Event' and select New Branch Node.

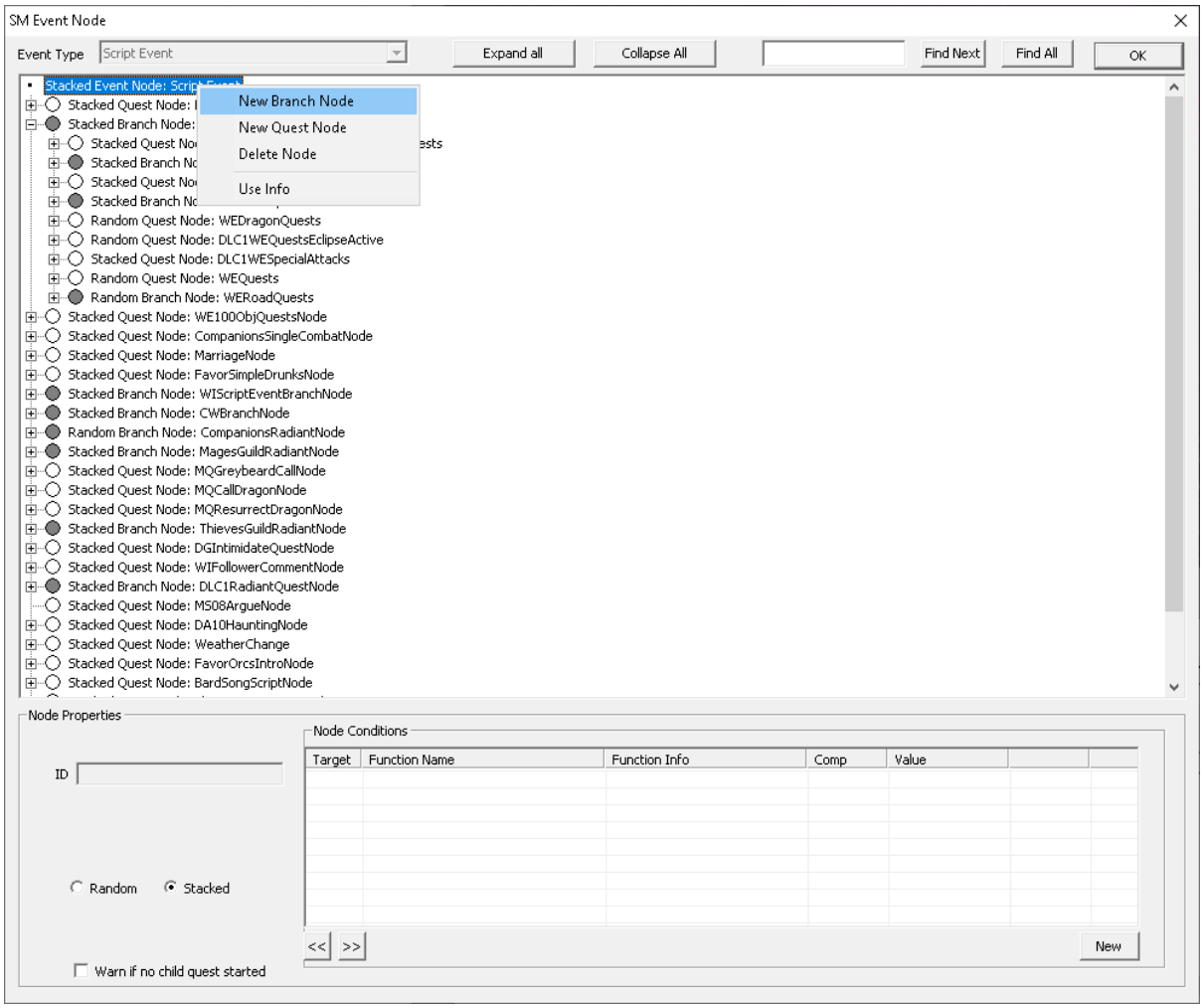


Figure 614 - Creating a new branch node in the Story Manager.

Enter an ID for this node in the ID field. In my example, I just added the 'WT' prefix to WEQuestNode.

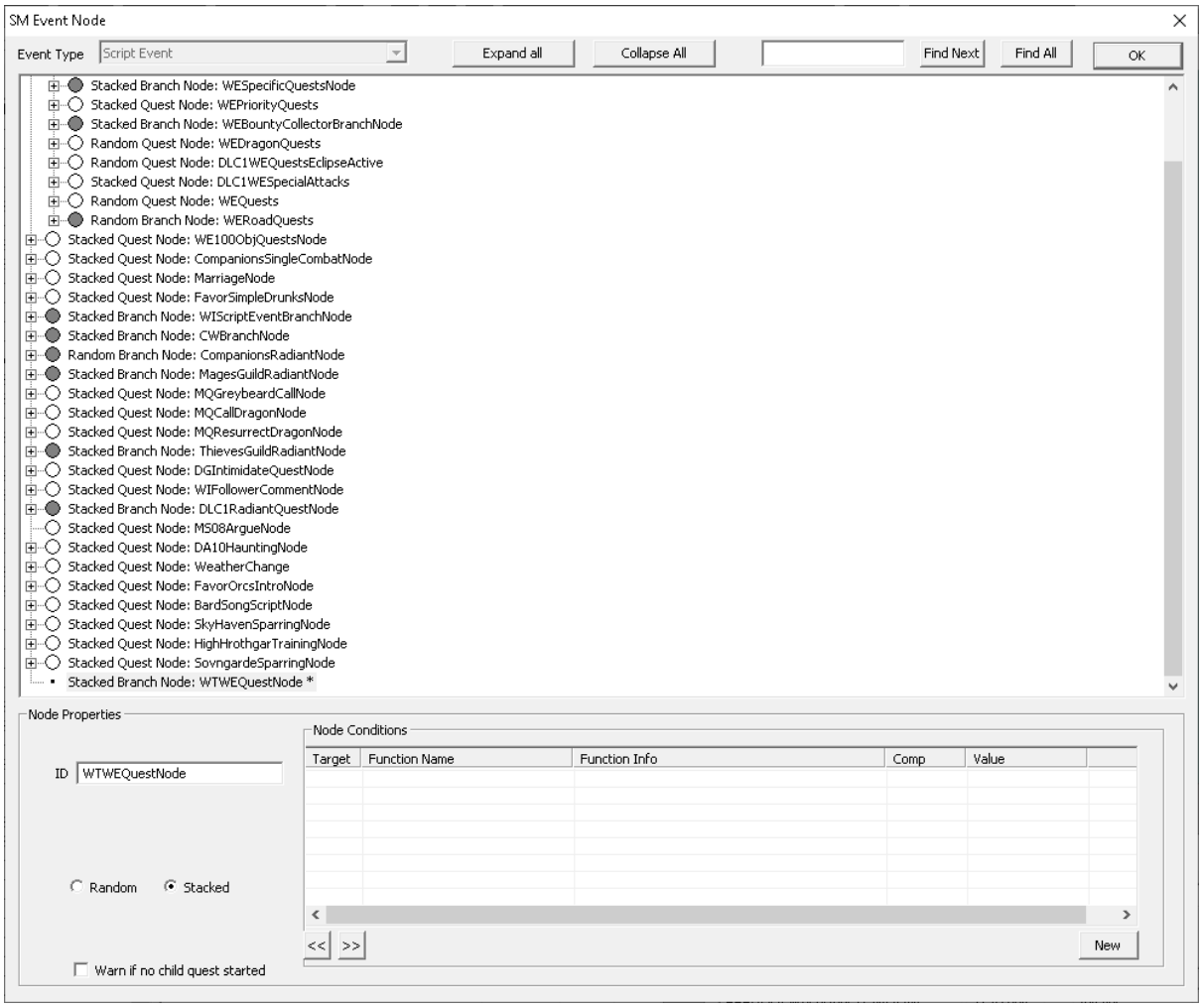


Figure 615 - Added a new root WE encounter branch.

Leave this node set to Stacked.

Click on WEQuestNode, right-click on a condition listed in Node Conditions, and select Copy All Conditions.

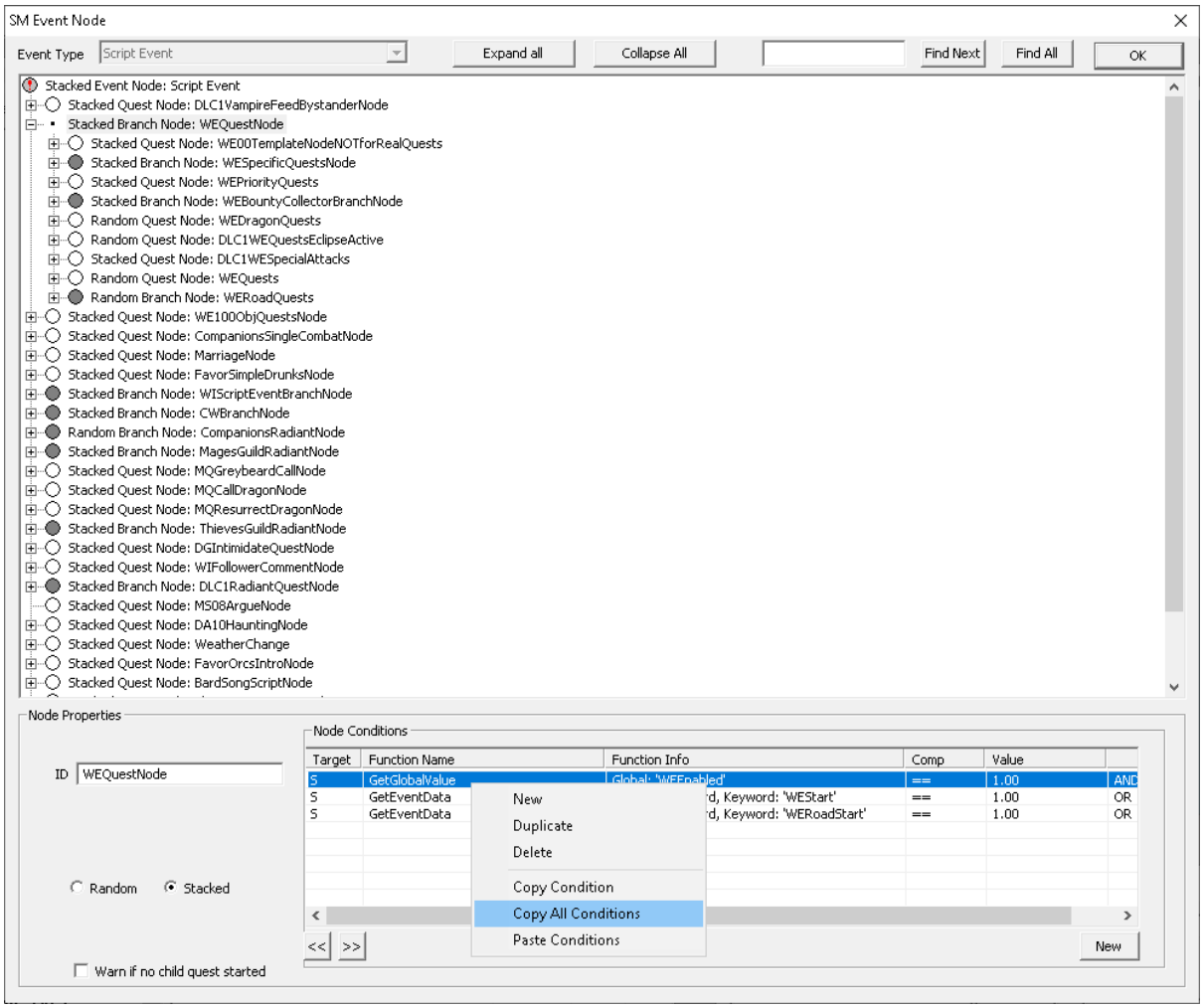


Figure 616 - Copying conditions on WEQuestNode.

Navigate back to your WEQuestNode, right-click on Node Conditions and select Paste Conditions.

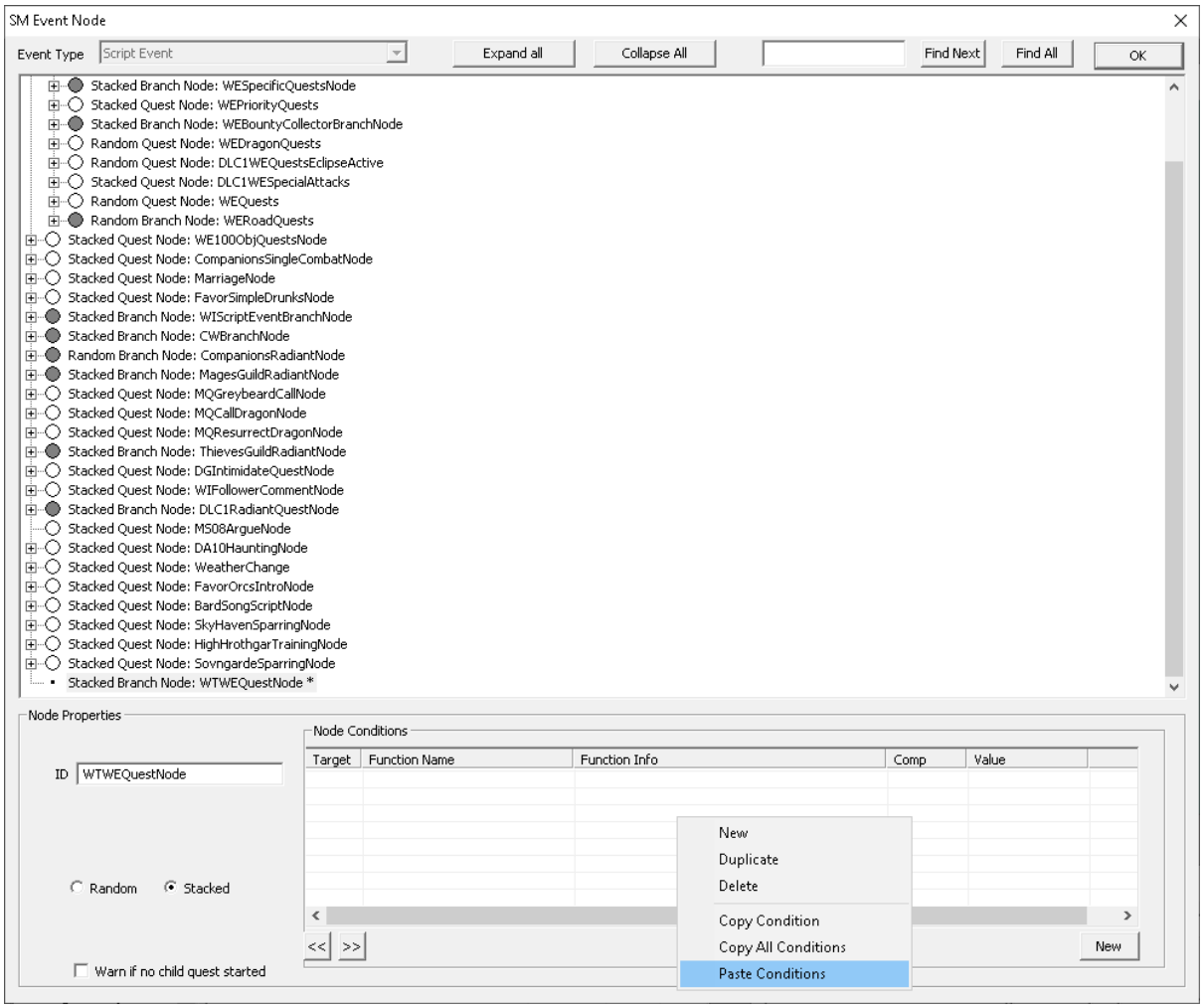


Figure 617 - Pasting the copied conditions to our own branch.

You should now have the same three conditions listed under Node Conditions for your WEQuestNode branch.

Our Node conditions list should appear as follows:

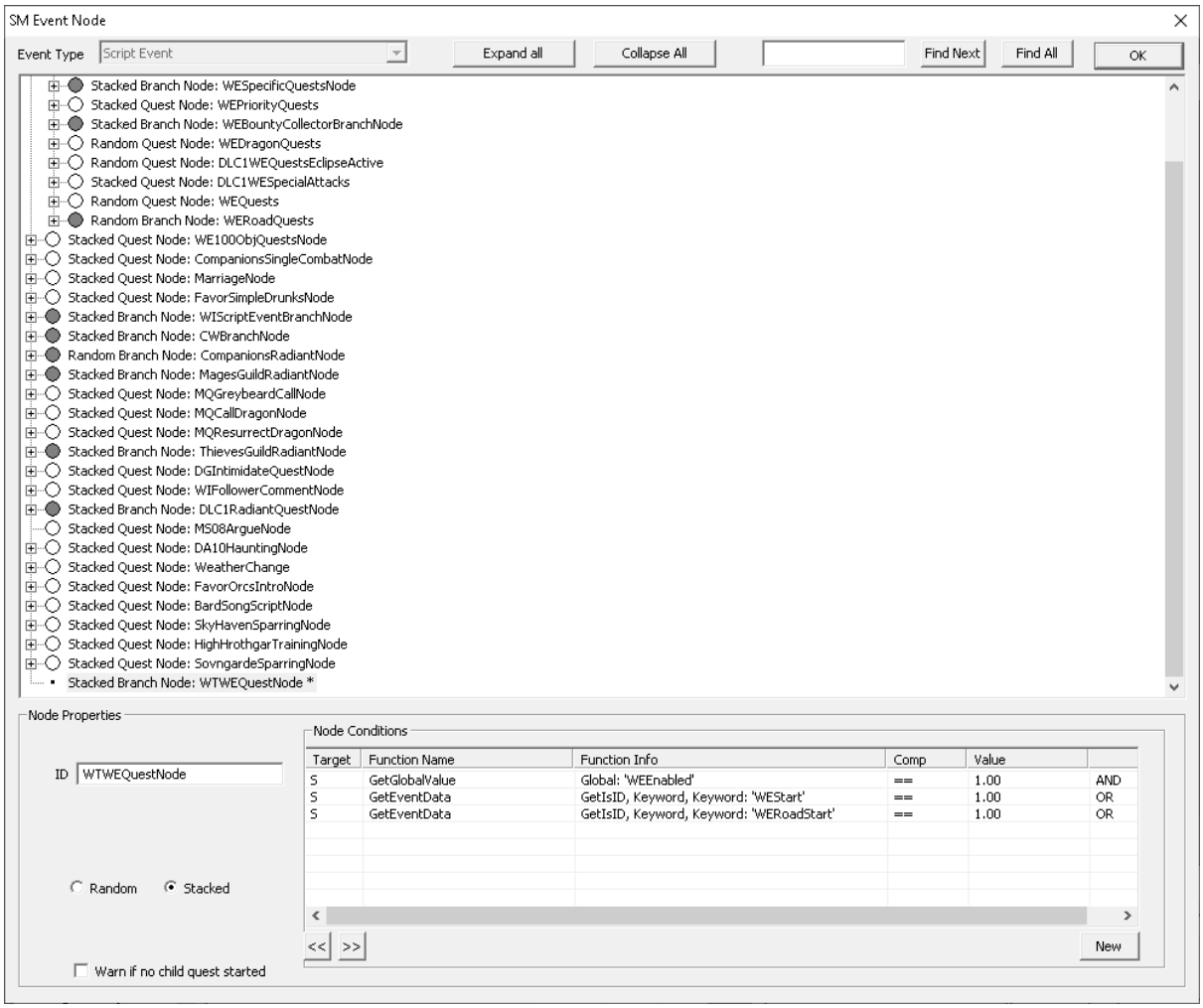


Figure 618 - Copying the WEStart keyword condition.

Double-click on the keyword condition for WEStart to bring up its properties.

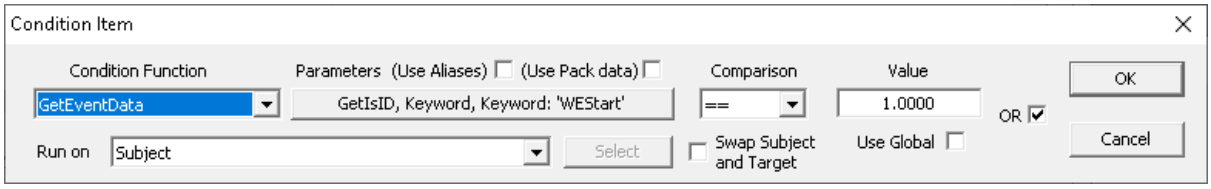


Figure 619 - Condition properties for the WEStart keyword condition.

Click on the button that reads: GetIsID, Keyword, Keyword: 'WEStart'

Under Parameter 3, set the drop-down menu to the WEStart keyword you created earlier. Following on with the previous example that would be WTWEStart.

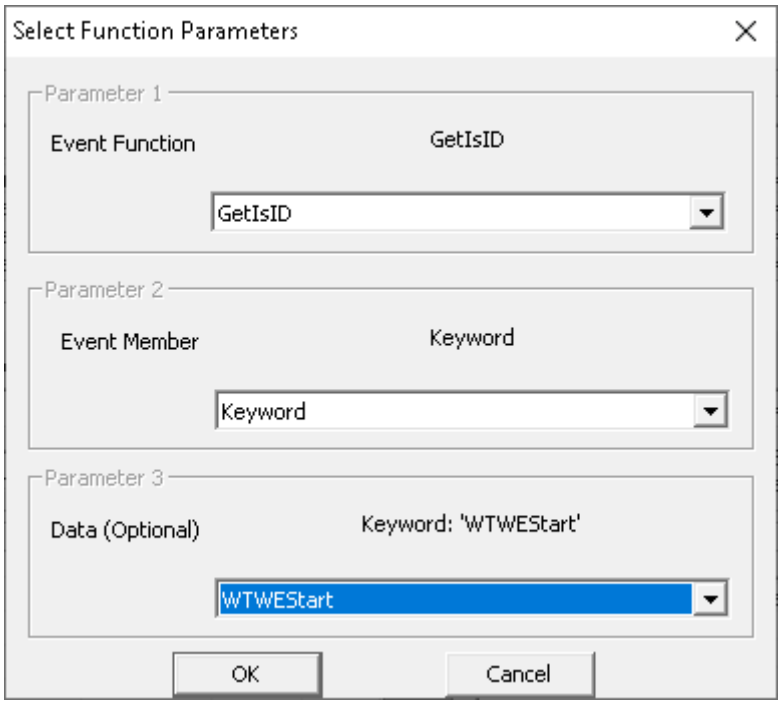


Figure 620 - Setting our own WEStart keyword.

Click OK to close out of Select Function Parameters.

Click OK to close out of Condition Item.

Repeat those steps to change the keyword condition for WERoadStart as well.

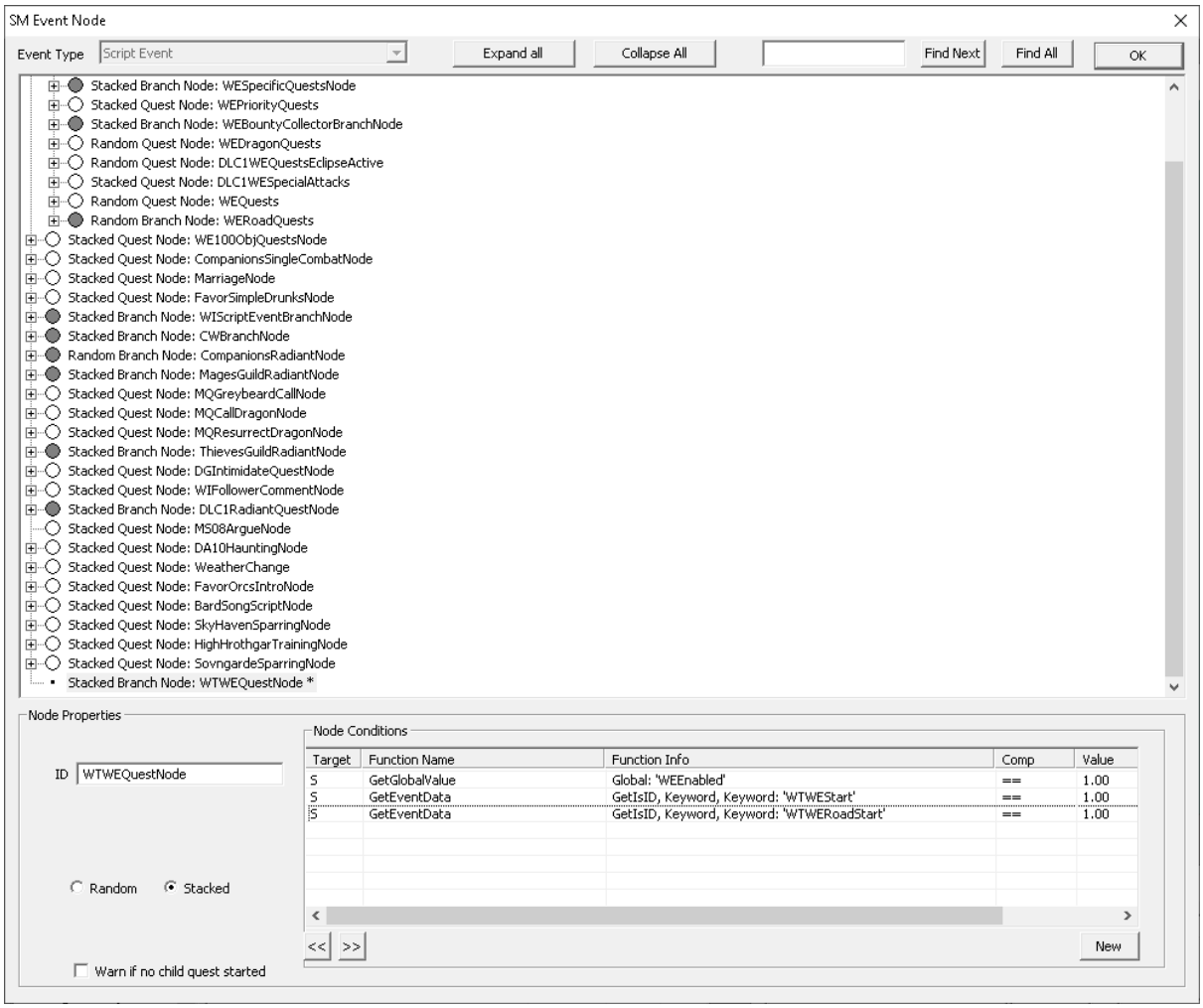


Figure 621 - Our WEQuestNode with new conditions added.

Our Node Conditions list should now appear as per the screenshot above.

Right-click on your WEQuestNode and create three new quest nodes.

Set the ID of the first branch node to RoadQuestNode, the second branch node to WildernessQuestNode, and the quest node to DragonQuests. In the example below I also added the 'WT' prefix.

The third node, DragonQuests, will be used to spawn random dragon encounters in our world space.

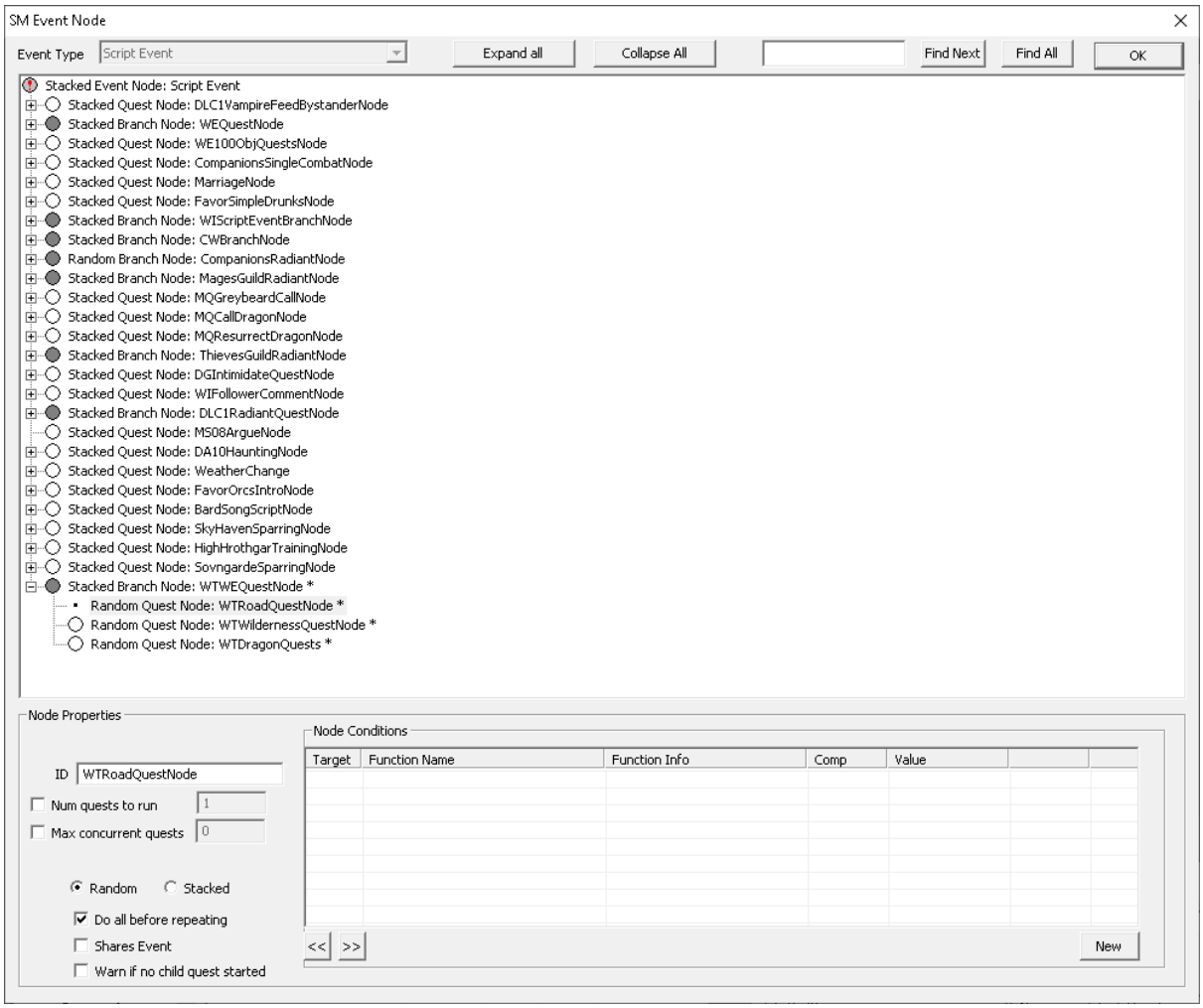


Figure 622 - Adding new branch nodes within our WEQuestNode branch.

Change all three from Stacked to Random. Quests added to stacked nodes will be processed sequentially, while quests added to random nodes will be processed in a randomized order.

Seeing as though we just want to game to choose the event randomly, we'll set these nodes to Random.

For our DragonQuests node, tick 'Max concurrent quests' and set the field next to it to '1'.

This helps ensure no more than one random dragon encounter can occur at a time.

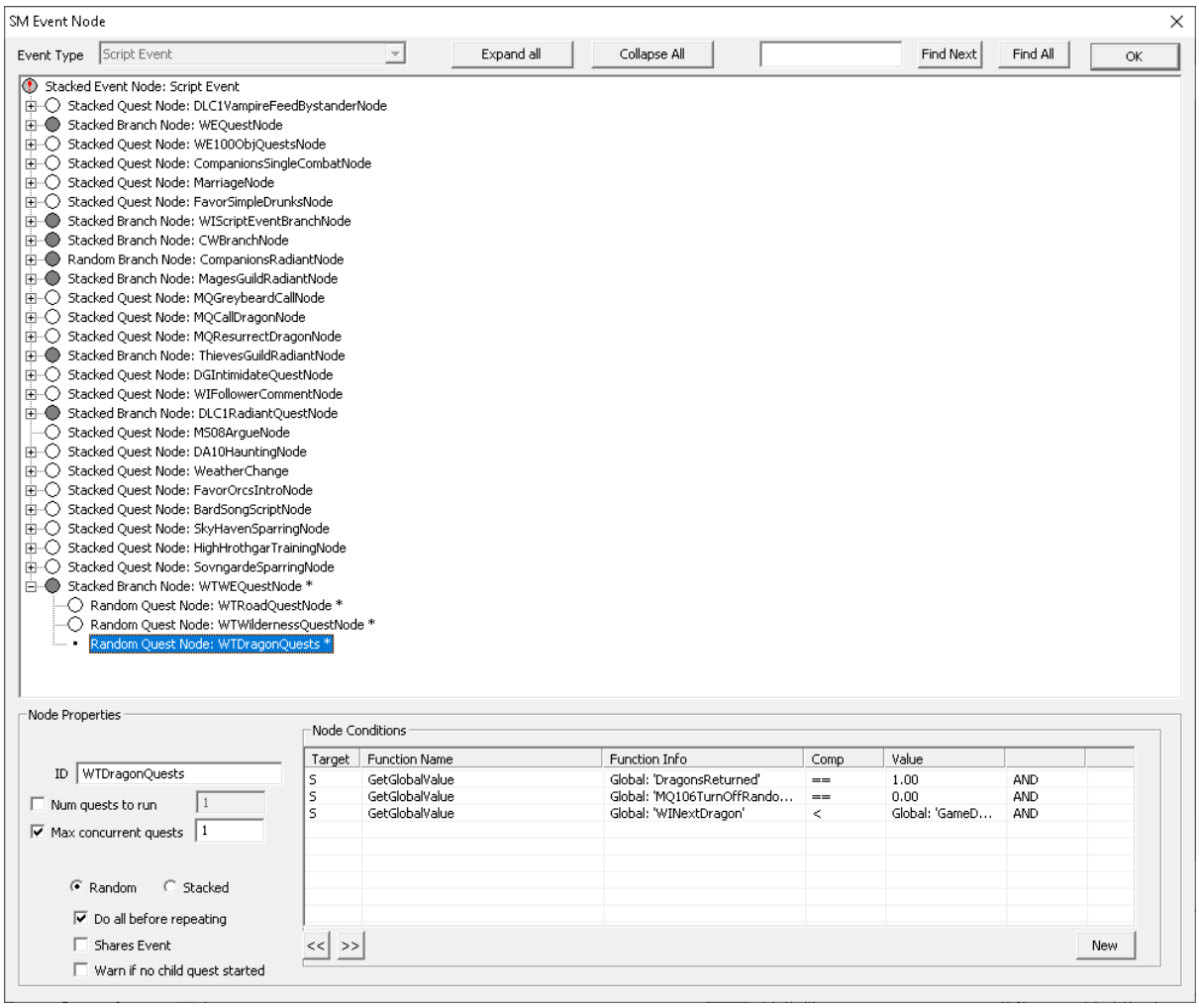


Figure 623 - Only ever run one event at a time from this node.

From our WEQuestNode parent, copy the keyword condition for our WERoadStart keyword and paste it to the Node Conditions on RoadQuests. Copy the keyword condition for our WEStart keyword and paste it to the Node Conditions on WildernessQuests.

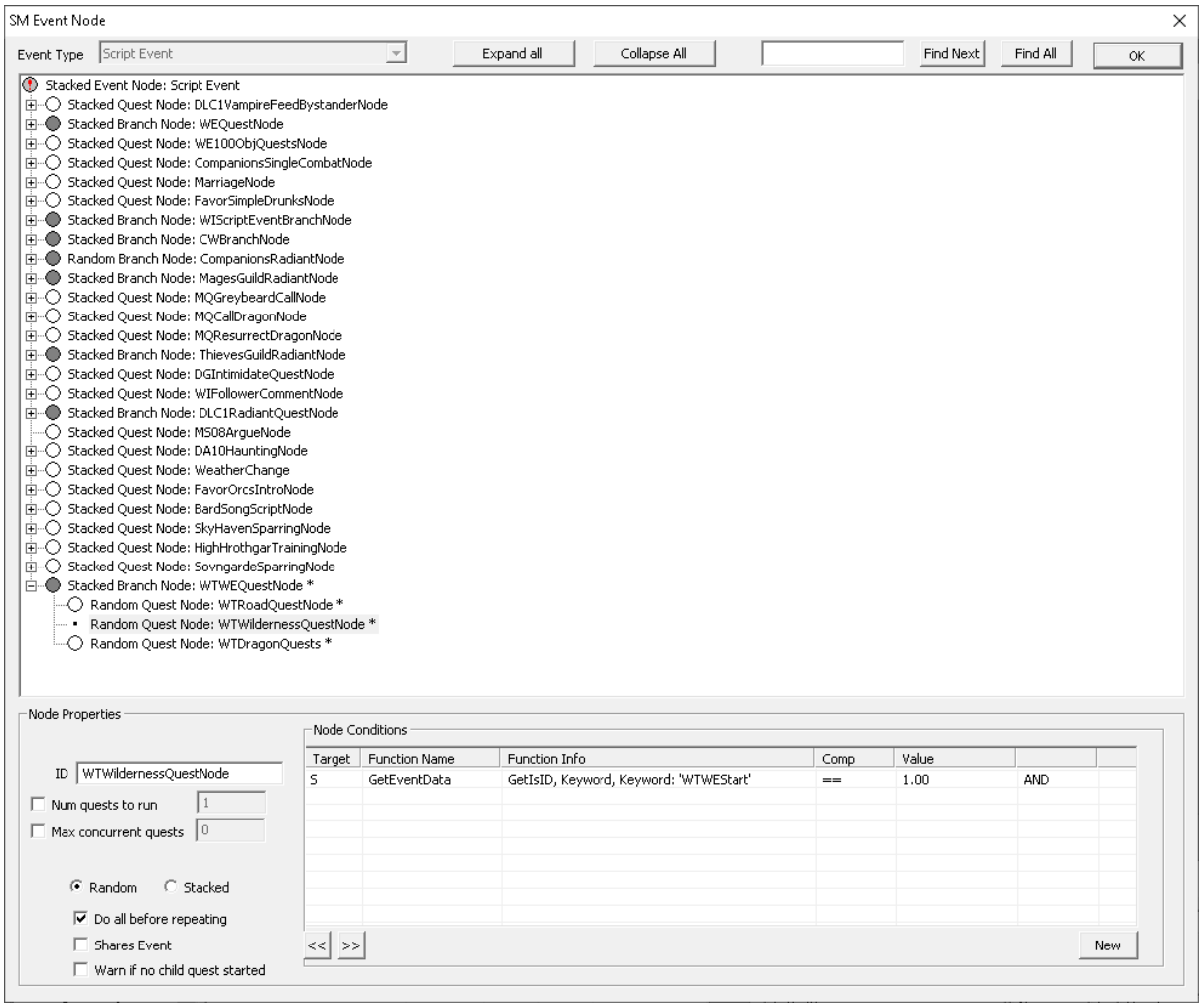


Figure 624 - Copied conditions to our RoadQuests and WildernessQuests branches.

Browse to the original WEQuestNode branch above, expand it and click on WEDragonQuests.

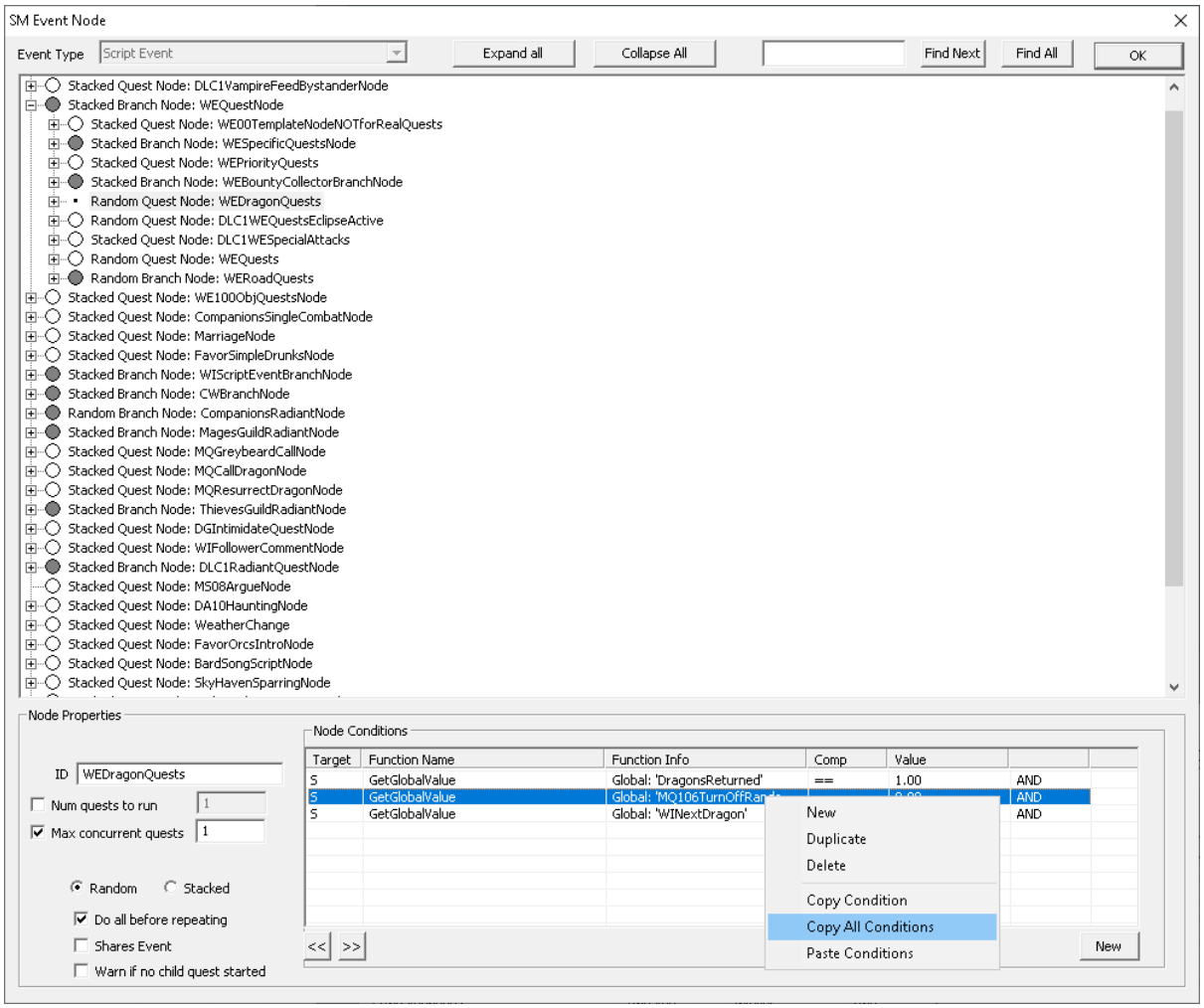


Figure 625 - Copying the conditions from WEDragonQuests.

Right-click on one of the conditions under Node Conditions and select Copy All Conditions.

Navigate to your DragonQuests branch, right-click in Node Properties and select Paste Conditions.

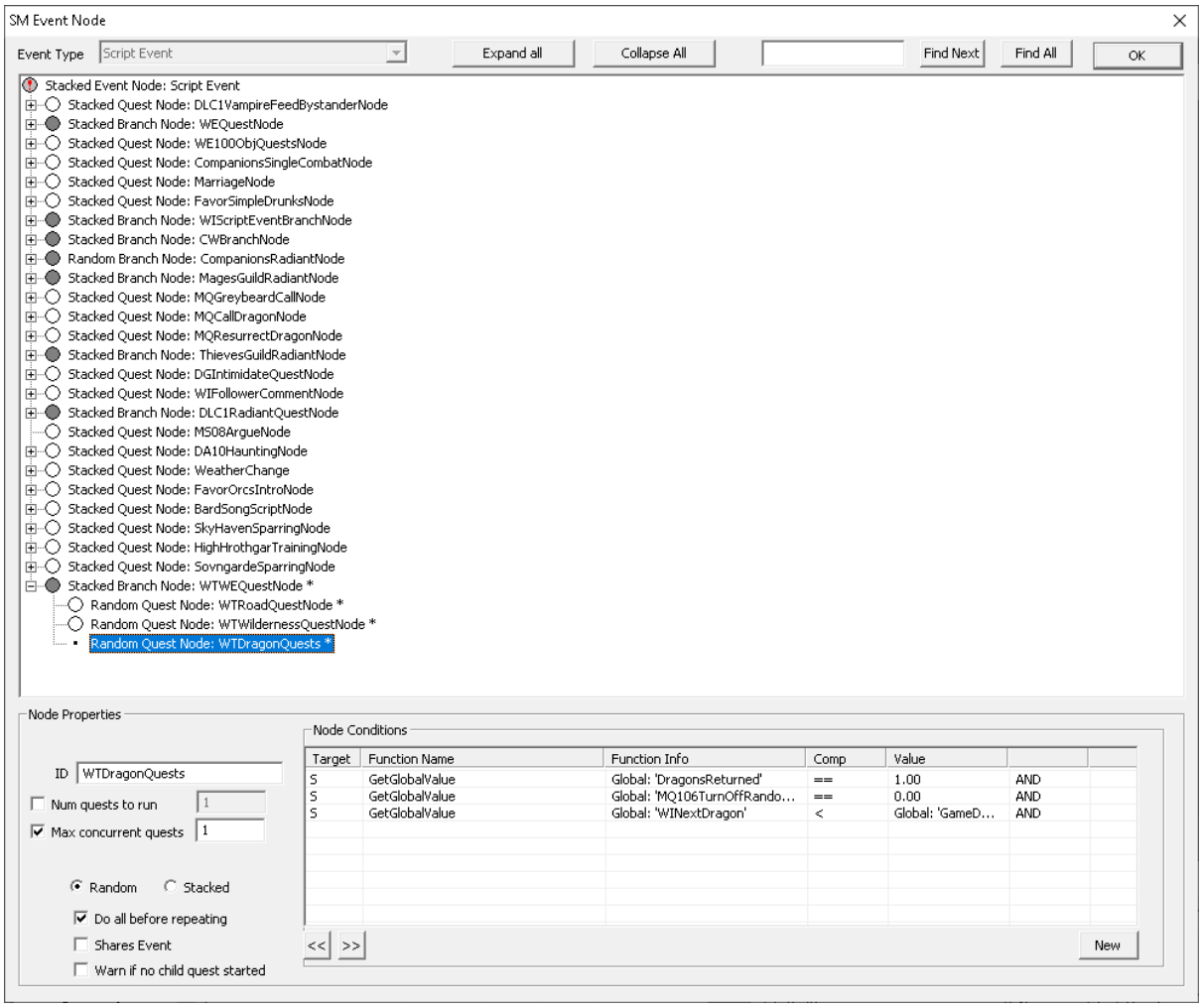


Figure 626 - Pasting conditions to our DragonQuests quest node.

Now we can start adding the random encounters we want to see in our world space.

To add random encounters to our RoadQuestNode, right click on your RoadQuestNode and select Add Quests.

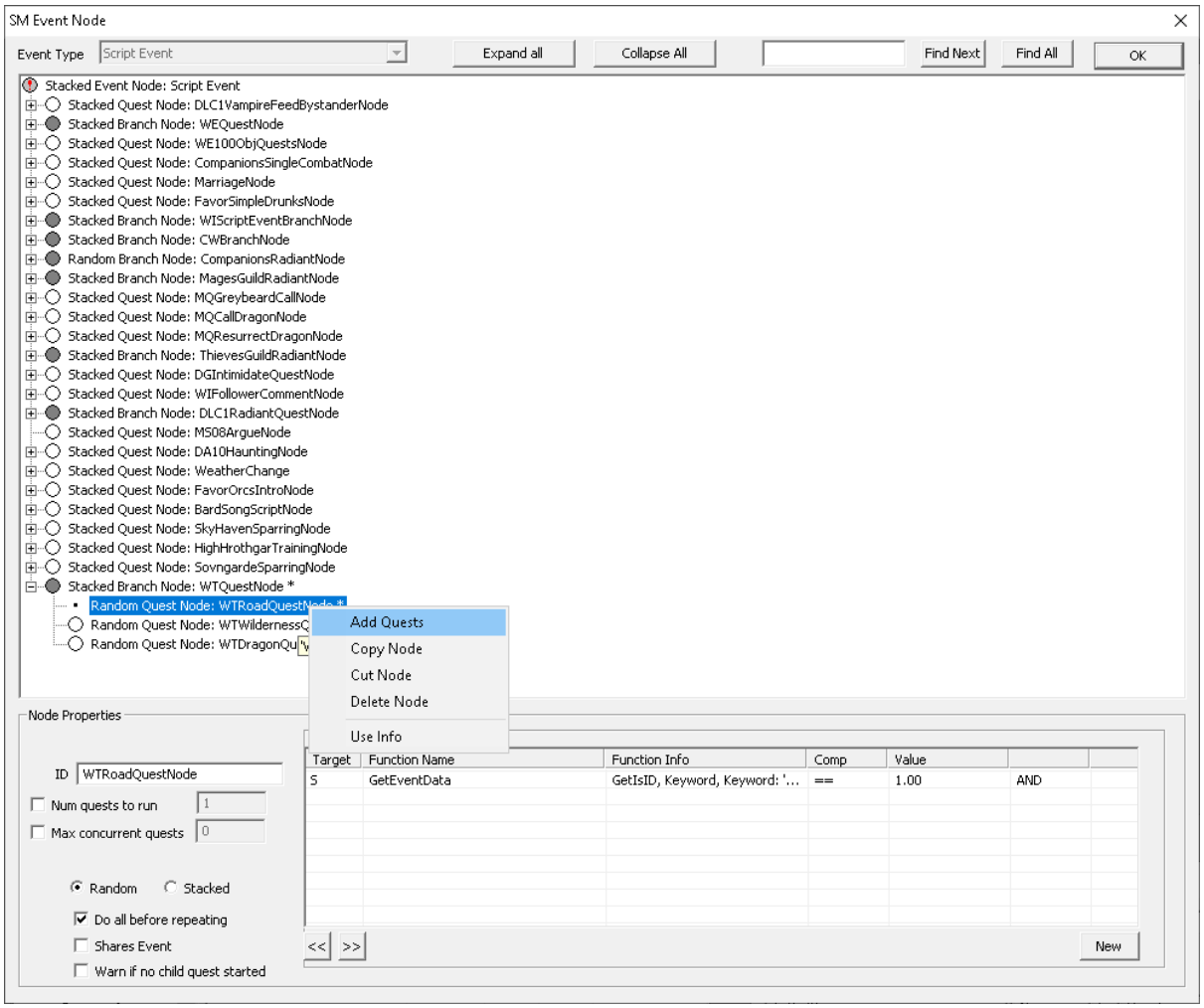


Figure 627 - Adding a new random encounter to the Story Manager.

Random encounters from the base game are prefixed with 'WE' so filter by 'WE'.

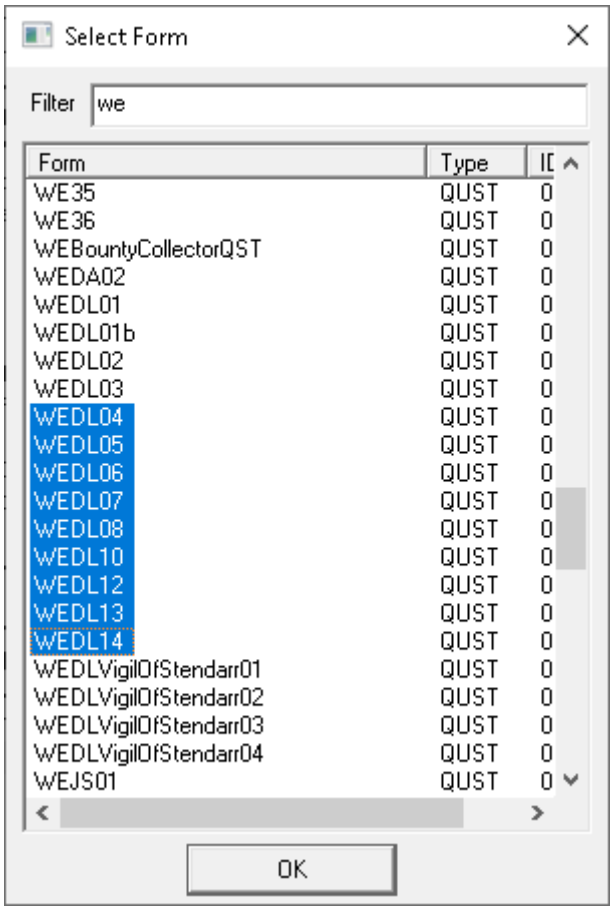


Figure 628 - Selecting the random encounters to add.

Highlight the random encounters you want to add and press OK.

To get a list of random encounters, browse to Character > Quests in the Object Window and filter by 'WE'. A description of the random encounter is generally found in the Name column to the right.

You can adjust the position of the random encounter quests added to the Story Manager by left-clicking on them and dragging them to the preferred location. You can even drag quests from one branch to another.

Steps for creating our own random encounters will be covered in Creating New Random Encounters.

For Wyrmstooth, I use a mix of my own custom random encounters plus a few random encounters from the base game as you can see in the screenshot below. Random encounters prefixed with 'WE' are from the base game. The ones with the 'WT' prefix are custom random encounters that I created.

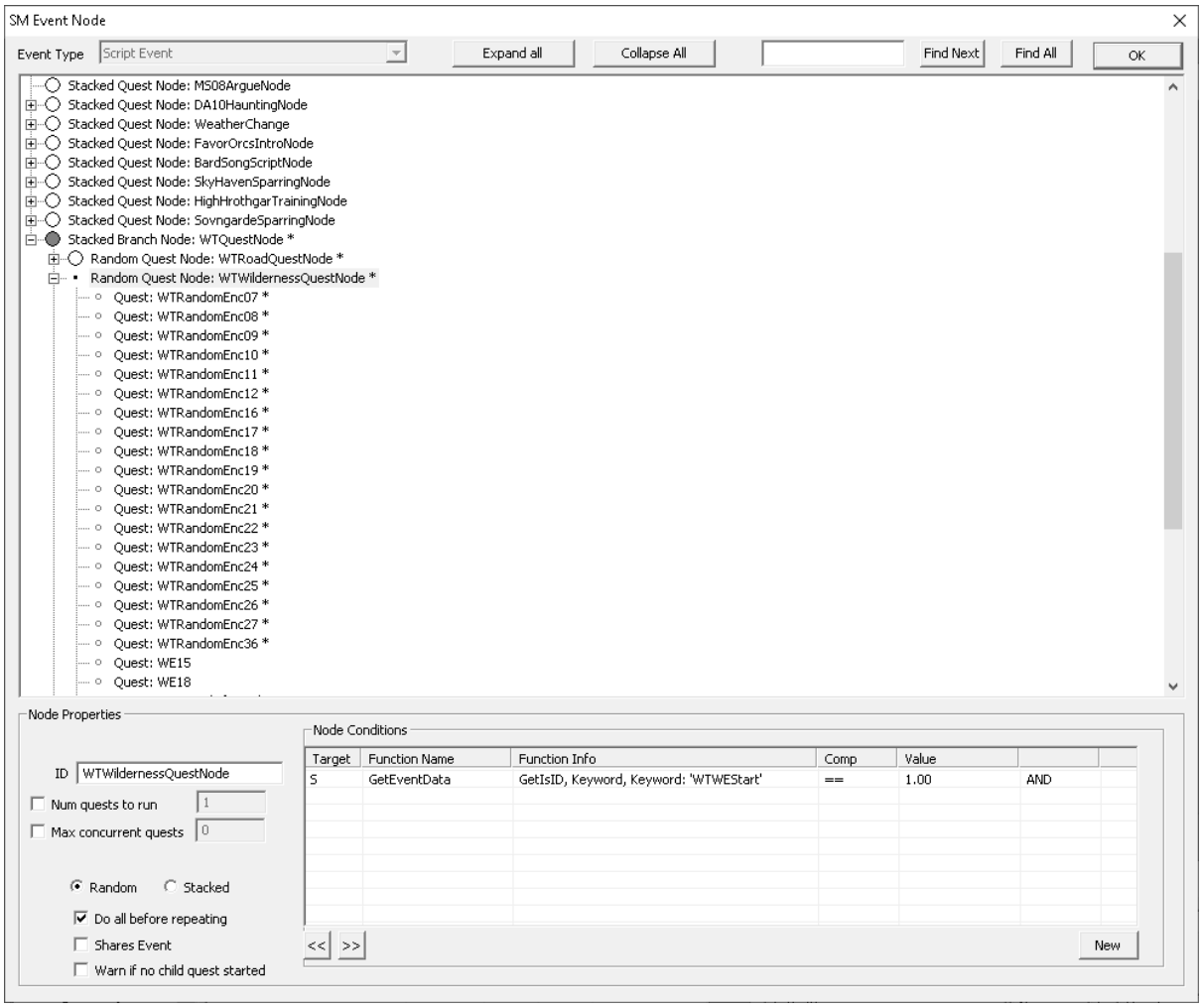


Figure 629 - Random encounter quests added.

For example, WE11 will be sabre cats hunting mammoths, WEDL02 will be Talsgar the Wanderer, WEJS25 will be two wizards duelling, and so on.

Add WE04 and WE08 to your DragonQuests quest node. WE04 will create a random encounter involving a dragon attacking the player, while WE08 will create a random encounter involving a dragon flying around the player before flying away.

Lastly, we'll need to change our WETriggers we placed in the world earlier to use the new keywords we created.

In the Object Window, go to World Objects > Activator and filter by 'wetrigger'.

Right-click on WETrigger and select 'Use Info'.

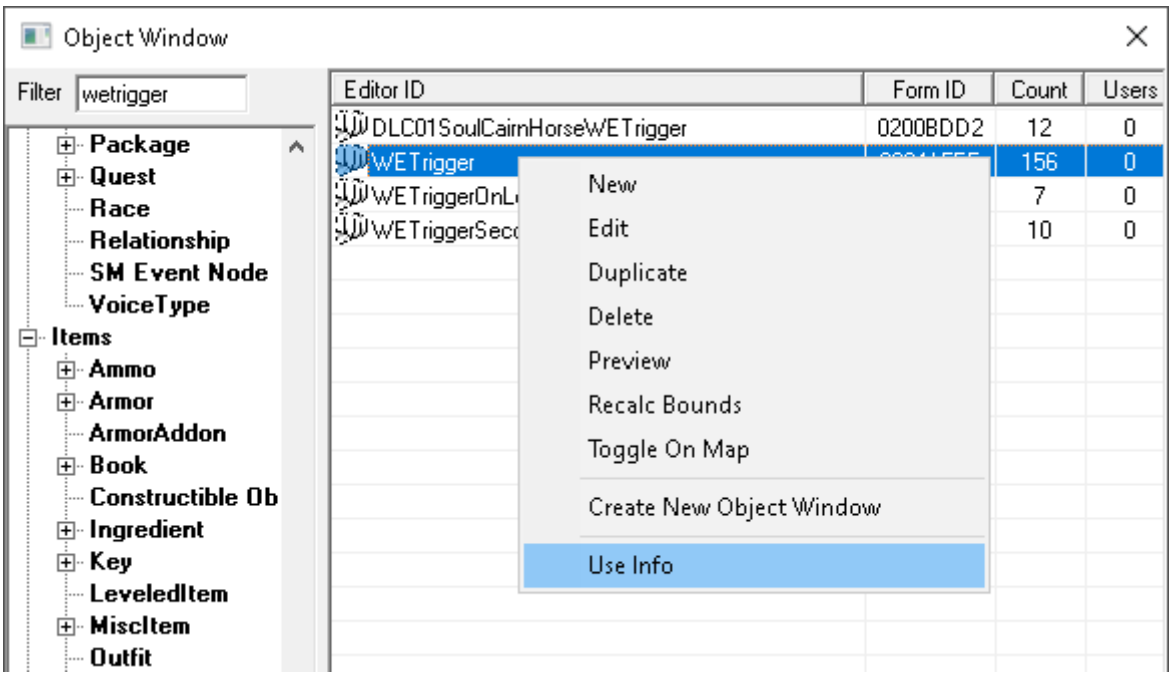


Figure 630 - Use Info.

This will bring up a list of every instance of WETrigger. Click on the World Editor ID column to sort by world space name.

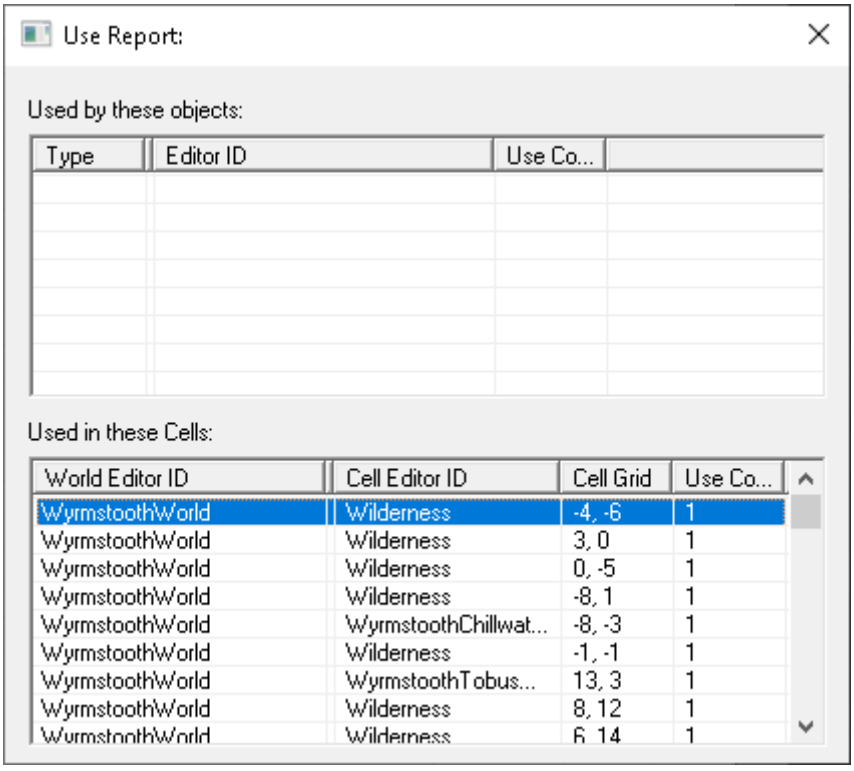


Figure 631 - Every instance of WETrigger.

Double-click on an instance in the 'Used in these cells' list to jump to that instance in the render window.

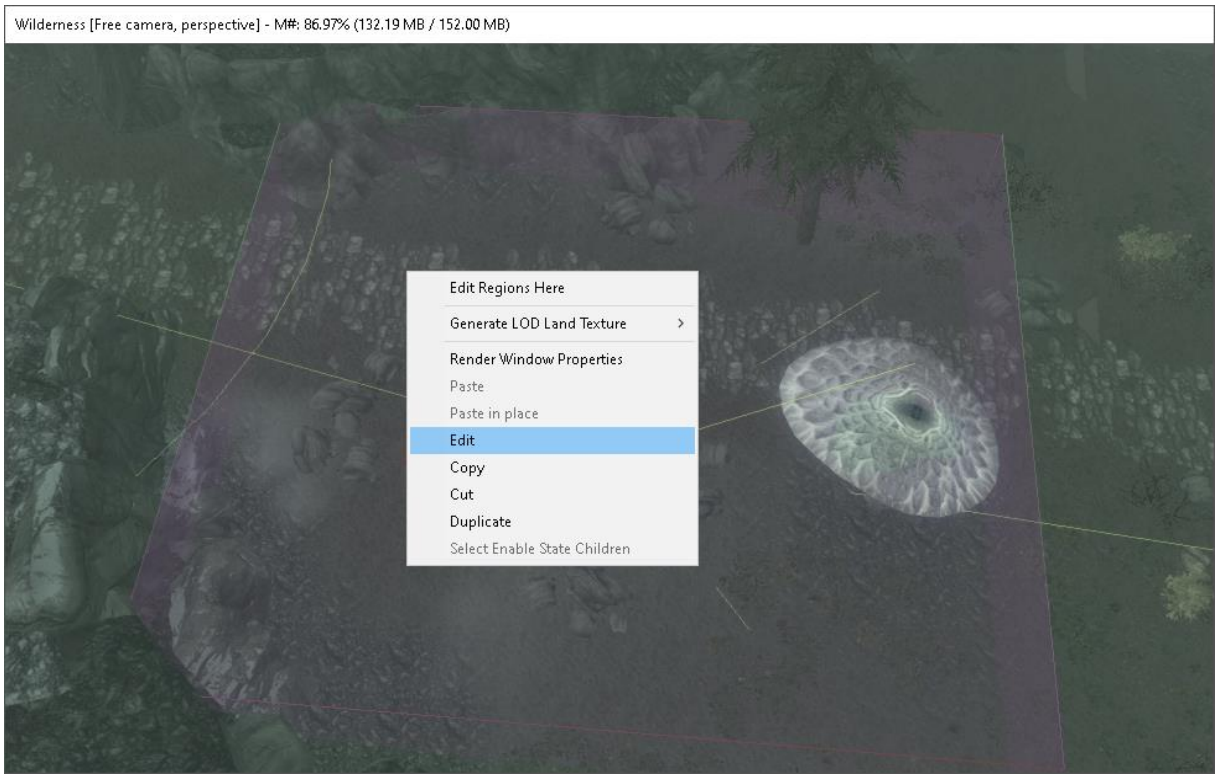


Figure 632 - Opening reference properties for the instance of WETrigger.

Right-click in the render window and select Edit or double-click on the WETrigger to bring up the properties for that instance.

Go to the Scripts tab, highlight the WETriggerScript and click on the Properties button.

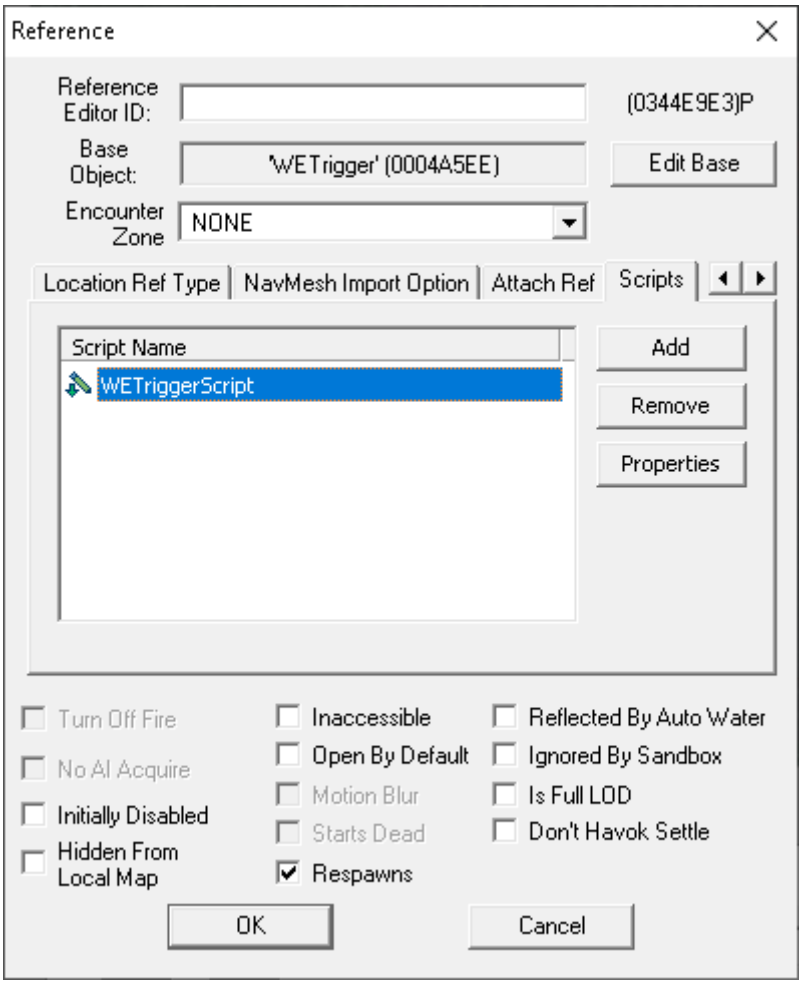


Figure 633 - Scripts tab in reference properties.

Change WERoadStart and WEStart to point to the new keywords you created earlier instead.

In the screenshot below I changed WERoadStart to WTWERoadStart and WEStart to WTWEStart.

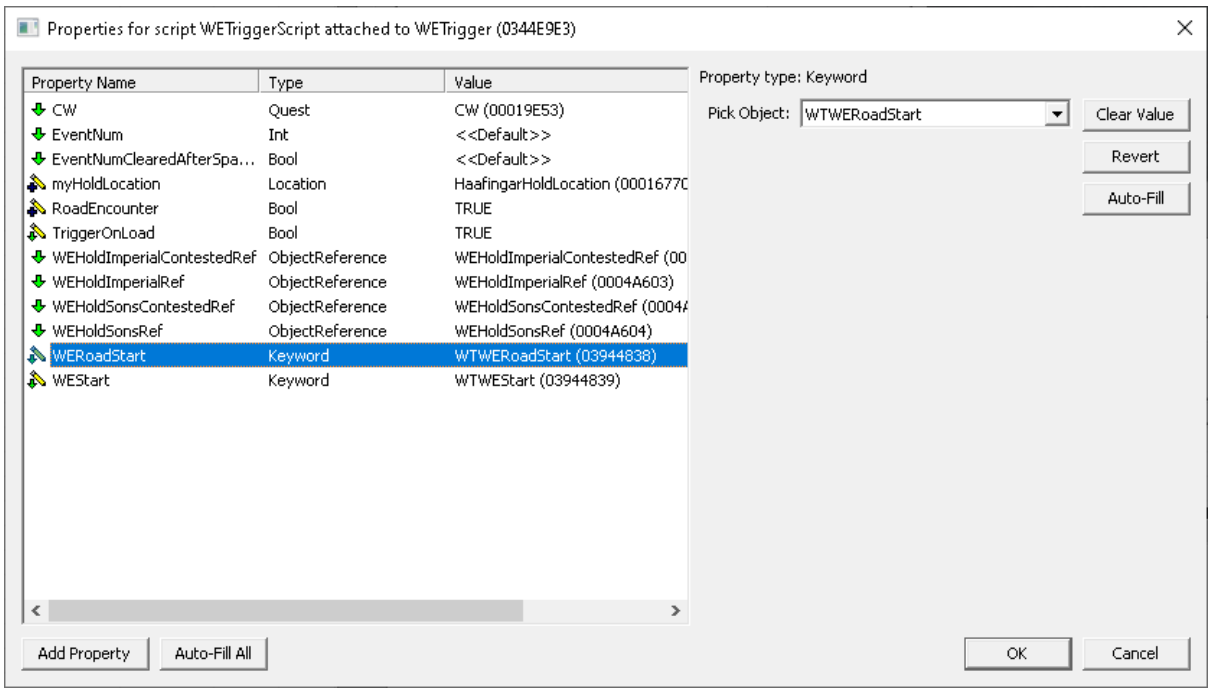


Figure 634 - Changing the keyword values in script properties.

Click OK to close out of the script properties.

Click OK to close out of the reference properties.

Go back to the list of WETrigger instances in Use Report, double-click on the next instance and repeat these steps.

Continue this until you've changed all instances of WETrigger in your world space.

CREATING NEW RANDOM ENCOUNTERS

This section will cover creating our own random encounters that we can then add to the Story Manager.

The easiest way to create a new random encounter is to duplicate an existing one that closely matches what we want to create.

Let's say we wanted to create a random encounter that spawns in an NPC that travels to the nearest settlement. Well, the random encounter WEDL02 already does that. It spawns in Talsgar the Wanderer, and when he spawns in he'll path to the nearest settlement and wander there until his event ends.

In the Object Window, filter by 'WEDL' and search for WEDL02.

Right-click on WEDL02 and select Duplicate.

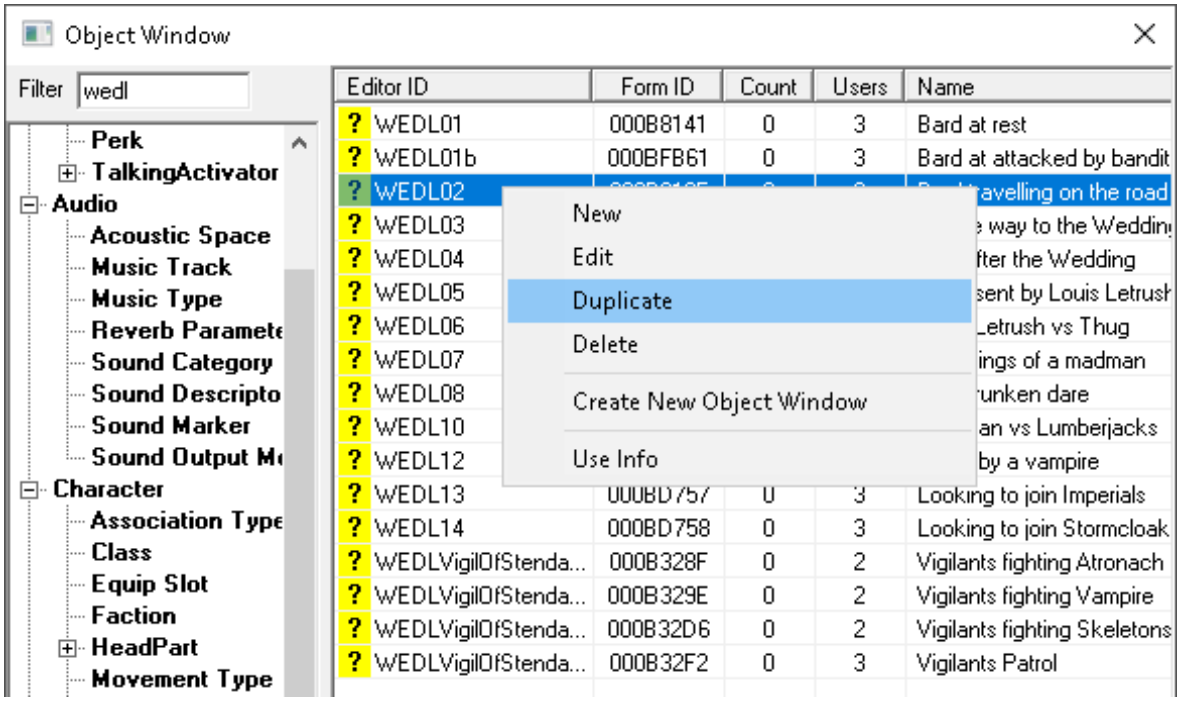


Figure 635 - Duplicating WEDL02.

This will create a duplicate quest that will be named WEDLDUPLICATE003.

Highlight WEDLDUPLICATE003, press 'F2' and rename it.

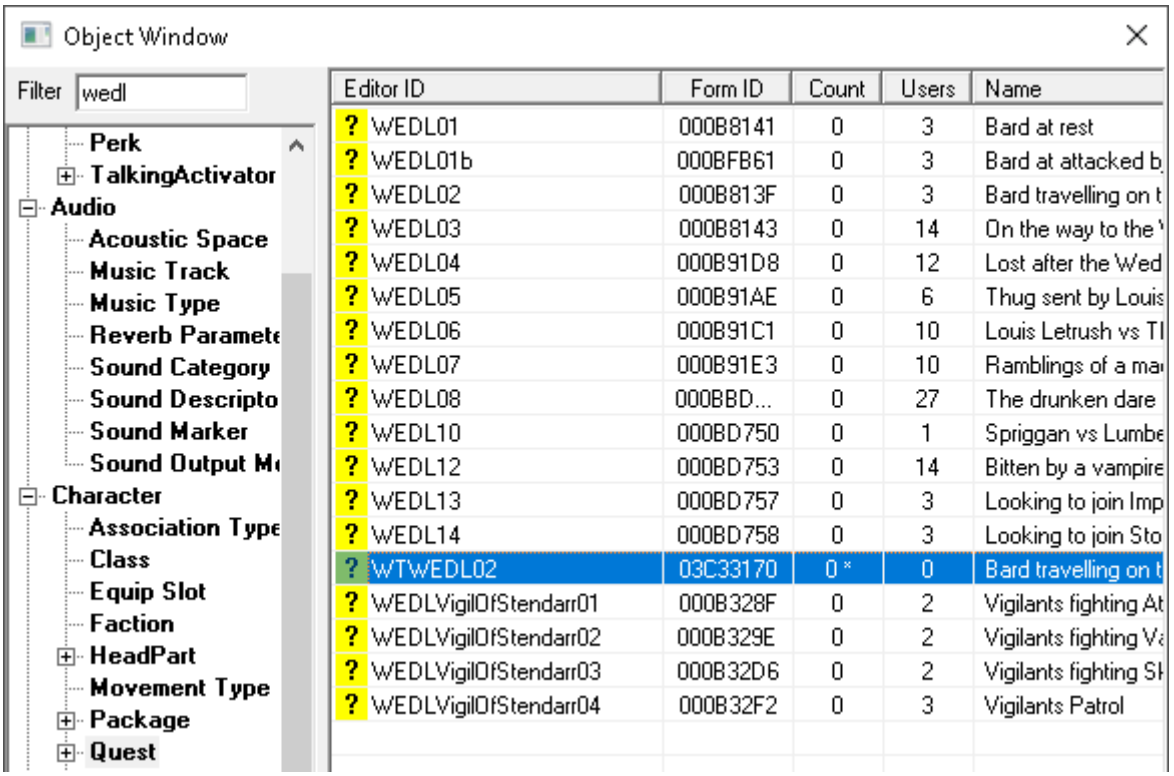


Figure 636 - Random encounter renamed.

For this example, I just reverted the name back to WEDL02 and added the 'WT' prefix.

Double-click on the new random encounter or right-click on it and select Edit.

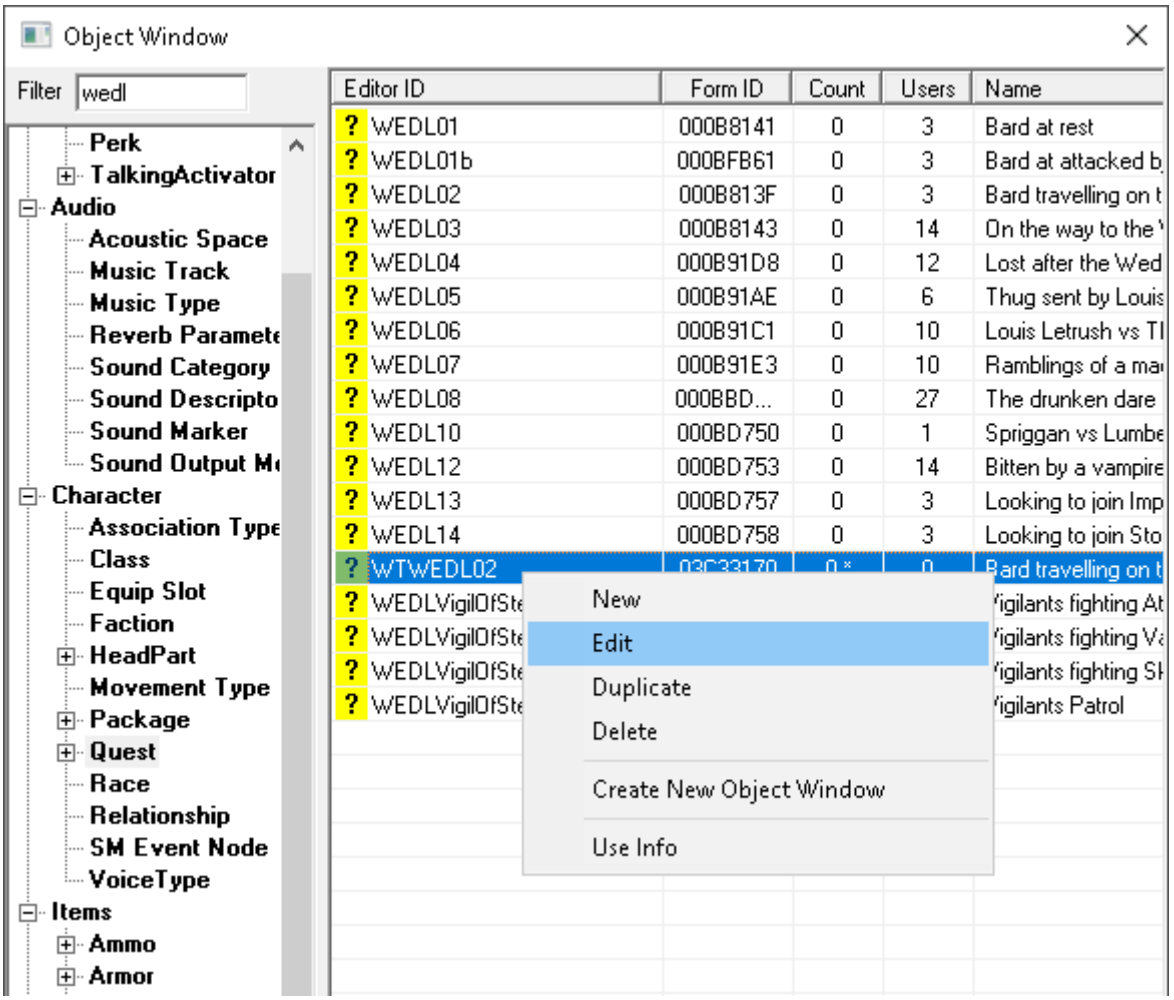


Figure 637 - Editing WTWEDL02.

Under the Quest Aliases tab, we'll see the Bard alias. This defines the NPC the encounter will be spawning in. Let's edit that first.

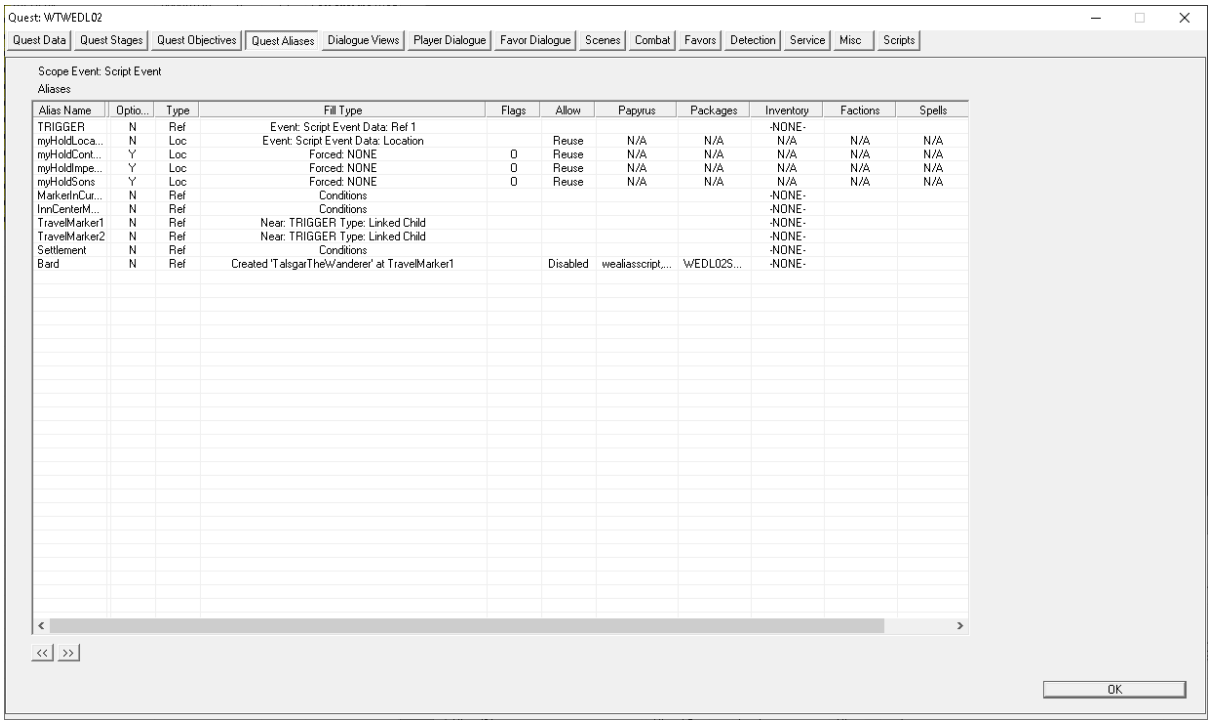


Figure 638 - WTWEDL02 Quest aliases.

Double-click on the Bard alias or right-click on it and select Edit.

In this example, we'll be spawning in a hunter instead, so change the drop-down menu next to 'Create Reference to Object' to 'LvlHunter' and change the Alias Name to 'Hunter'.

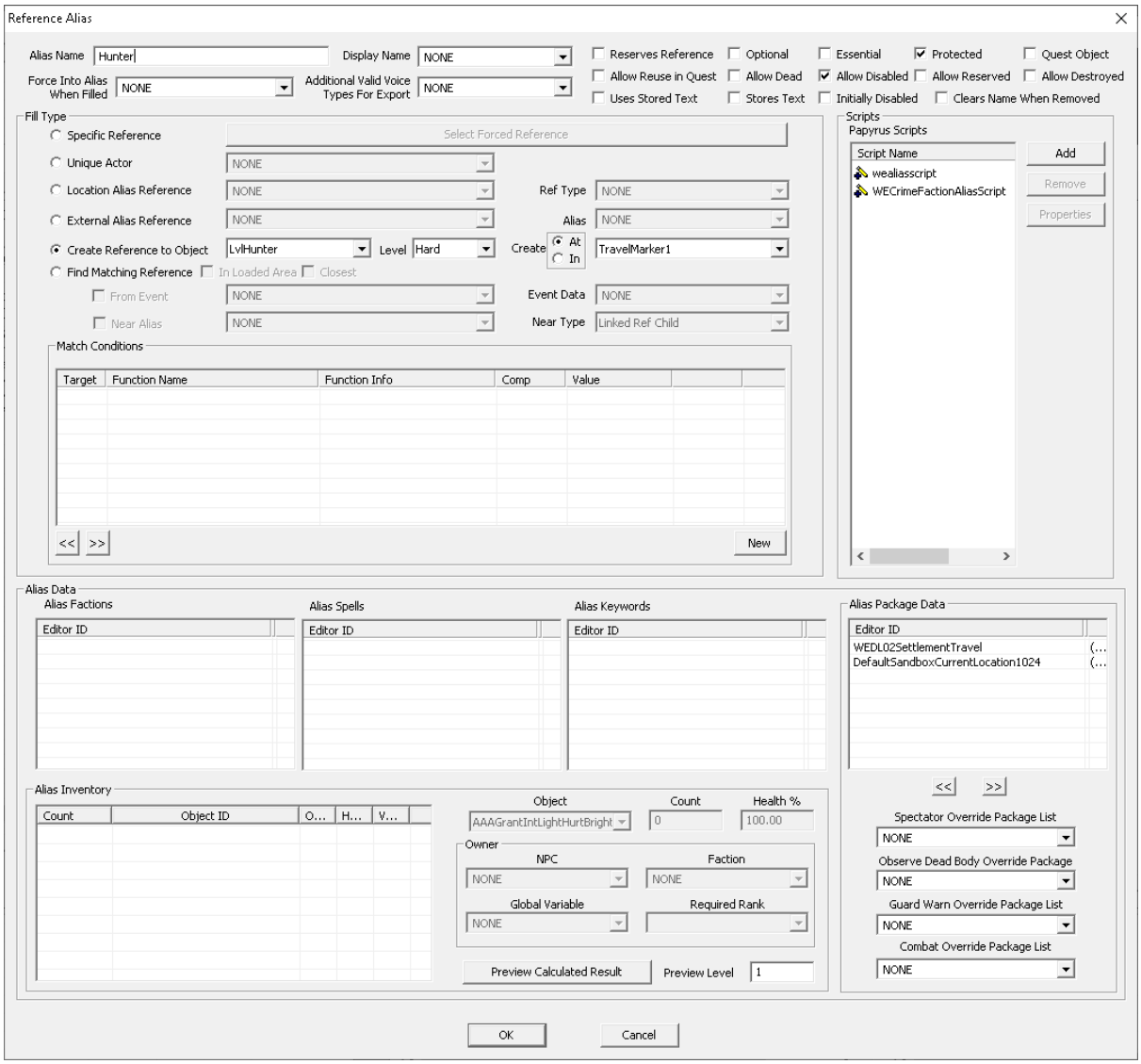


Figure 639 - Spawning in a hunter instead.

If you wanted to spawn in a unique NPC instead, such as Aela the Huntress, you would change the Fill Type to 'Unique Actor' and select 'AelaTheHuntress' from the drop-down menu.

When referencing a unique NPC, try using their unique ID where possible rather than a specific reference. If I set the Fill Type to Specific Reference instead and selected Aela's reference in the render window, this may create an unnecessary cell edit.

Near the bottom right you'll see the Alias Package Data list. This is a list of packages assigned to the NPC when the random encounter is running. These packages will override any packages already assigned to an NPC in their actor properties.

This alias is assigned two packages; WEDL02SettlementTravel and DefaultSandboxCurrentLocation1024.

So basically, when the NPC spawns in, they'll travel to the nearest inn. When they arrive, the travel package will end, and the sandbox package takes over to make them wander around that location.

Double-click on WEDL02SettlementTravel to open its properties.

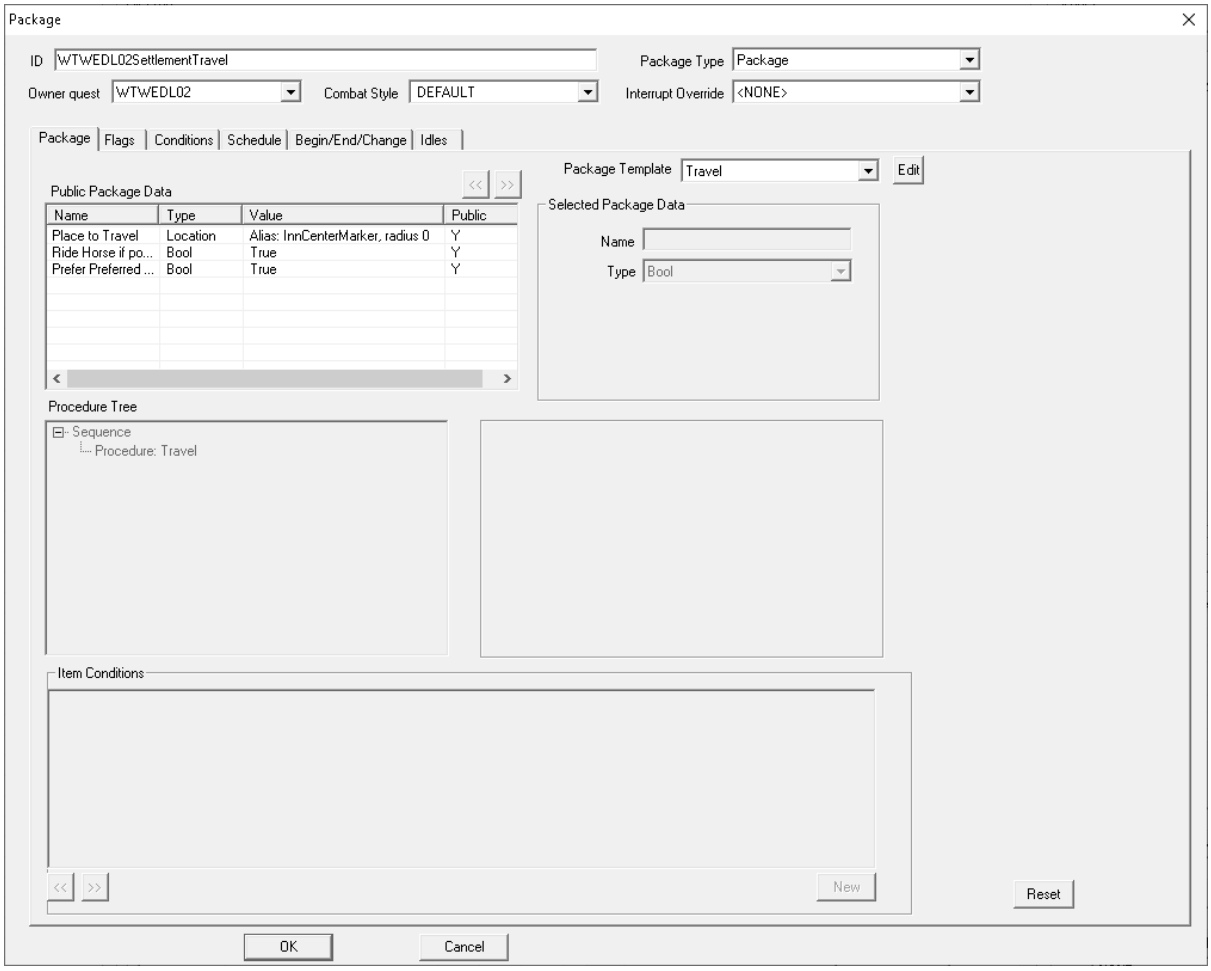


Figure 640 - Changing the package ID and Owner Quest.

In the screenshot above, I added the 'WT' prefix to the package ID and changed the Owner Quest to point to WTWEDL02 instead of WEDL02.

Click OK.

If prompted to create a new form, click Yes. We don't want to change the existing WEDL02SettlementTravel package, we want to create a new one for our new random encounter.

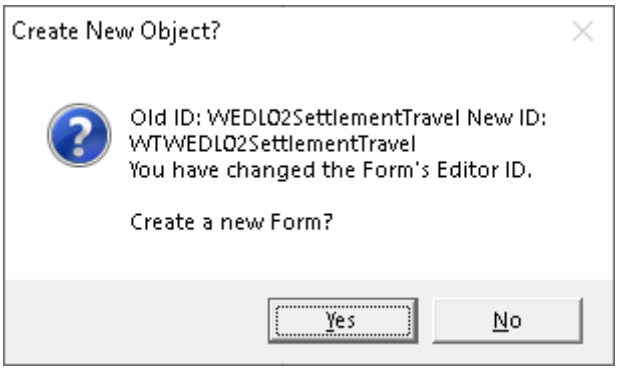


Figure 641 - Create a new form.

Highlight WEDL02SettlementTravel in the Alias Package Data list and press Delete, or right-click on it and select Delete.

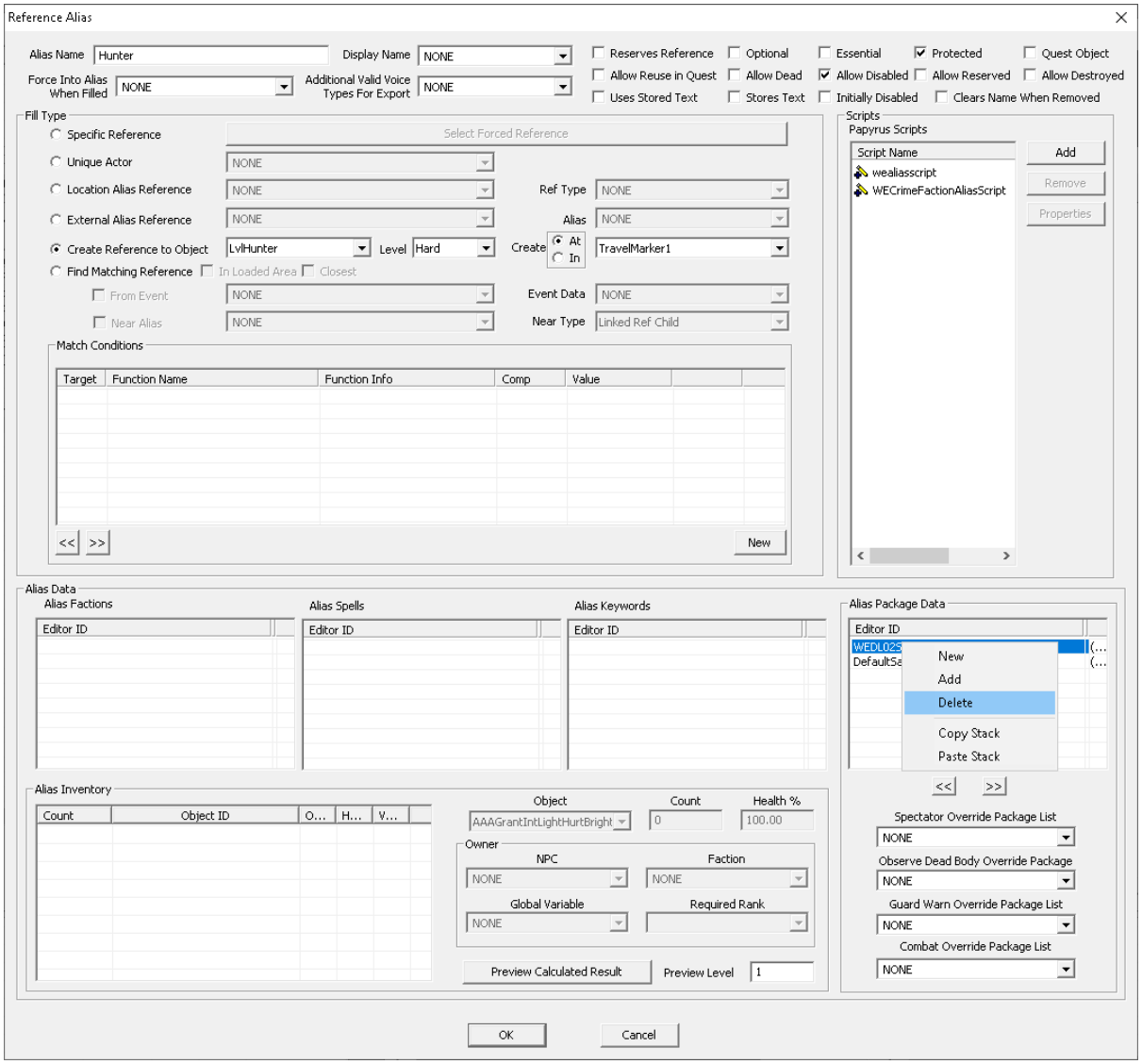


Figure 642 - Deleting the existing WEDL02SettlementTravel package.

Right-click in Alias Package Data and select Add.

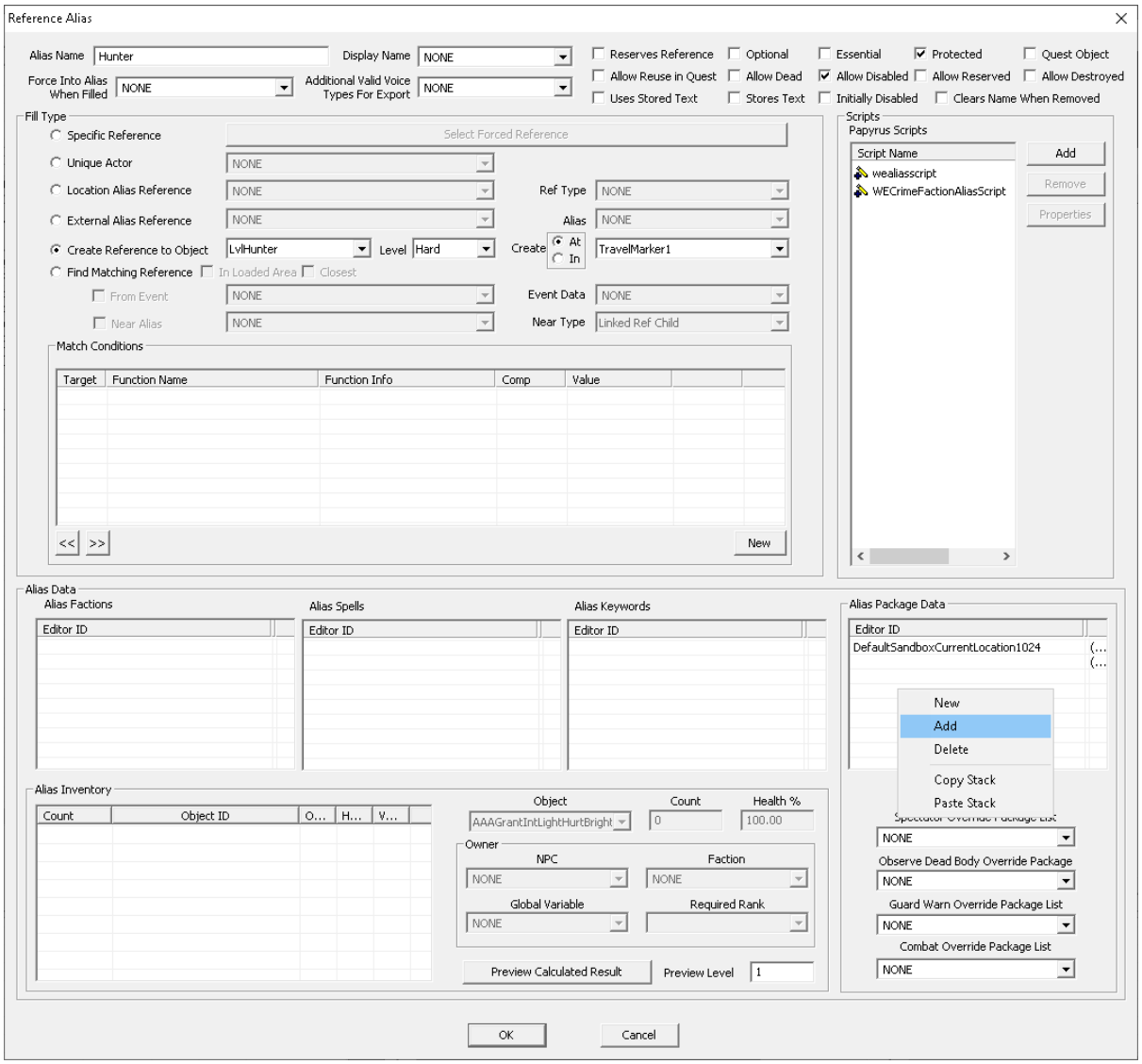


Figure 643 - Adding a new package.

Search for the package you just created, highlight it and click OK.

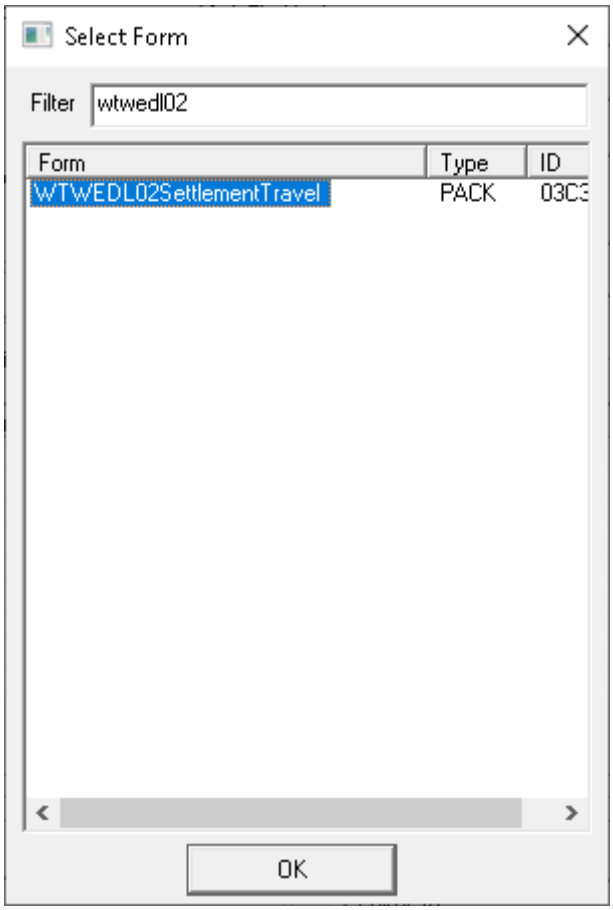


Figure 644 - Selecting the package we just created.

It'll appear at the bottom of the Alias Package Data list. We'll need to move it above DefaultSandboxCurrentLocation1024 so it goes first.

Highlight the new package and click on '<<' to move it above DefaultSandboxCurrentLocation1024.

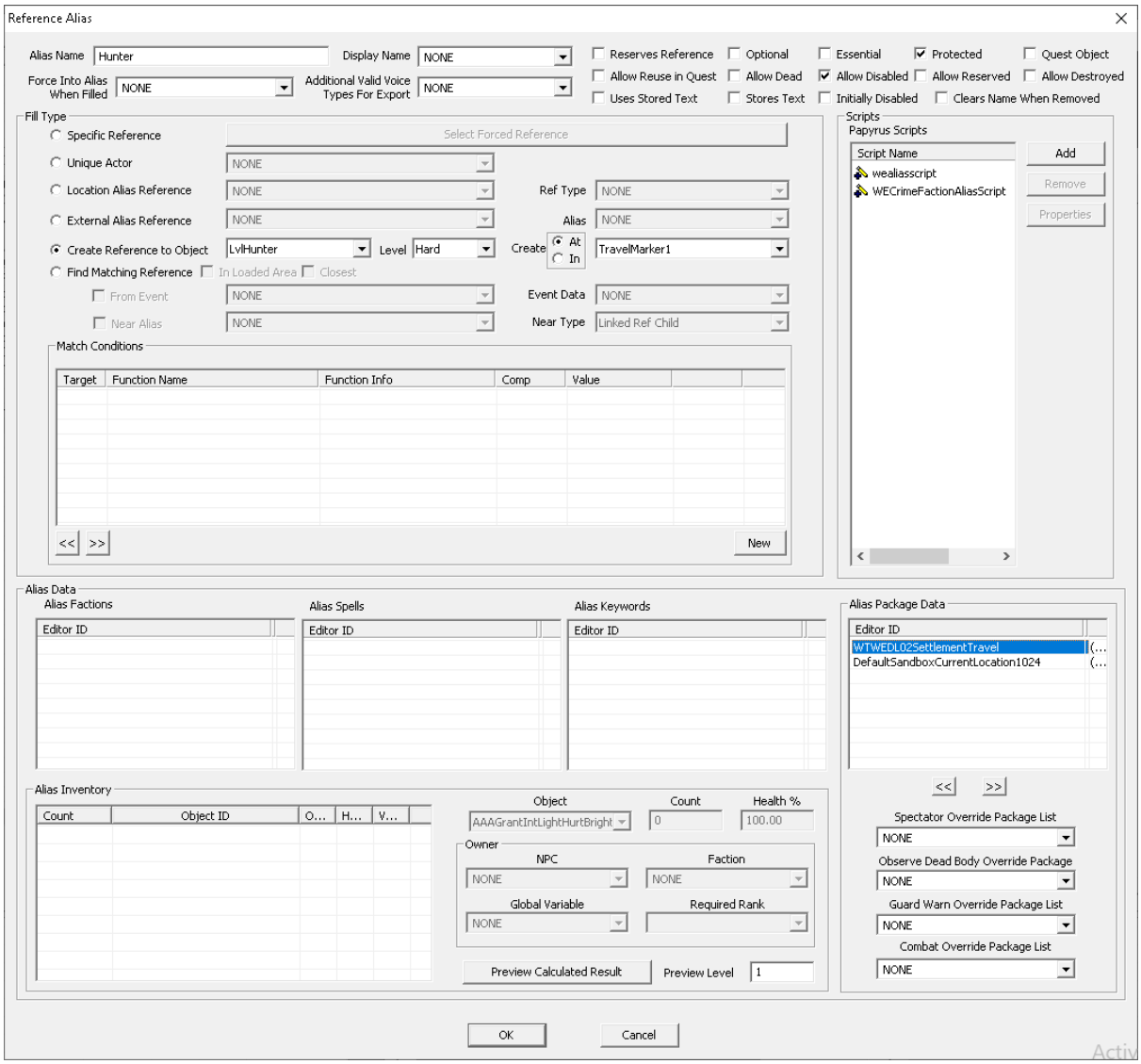


Figure 645 - Moving our package up in Alias Package Data.

Lastly, let's just change the Level drop-down menu to None, untick Allow Disabled and untick Protected.

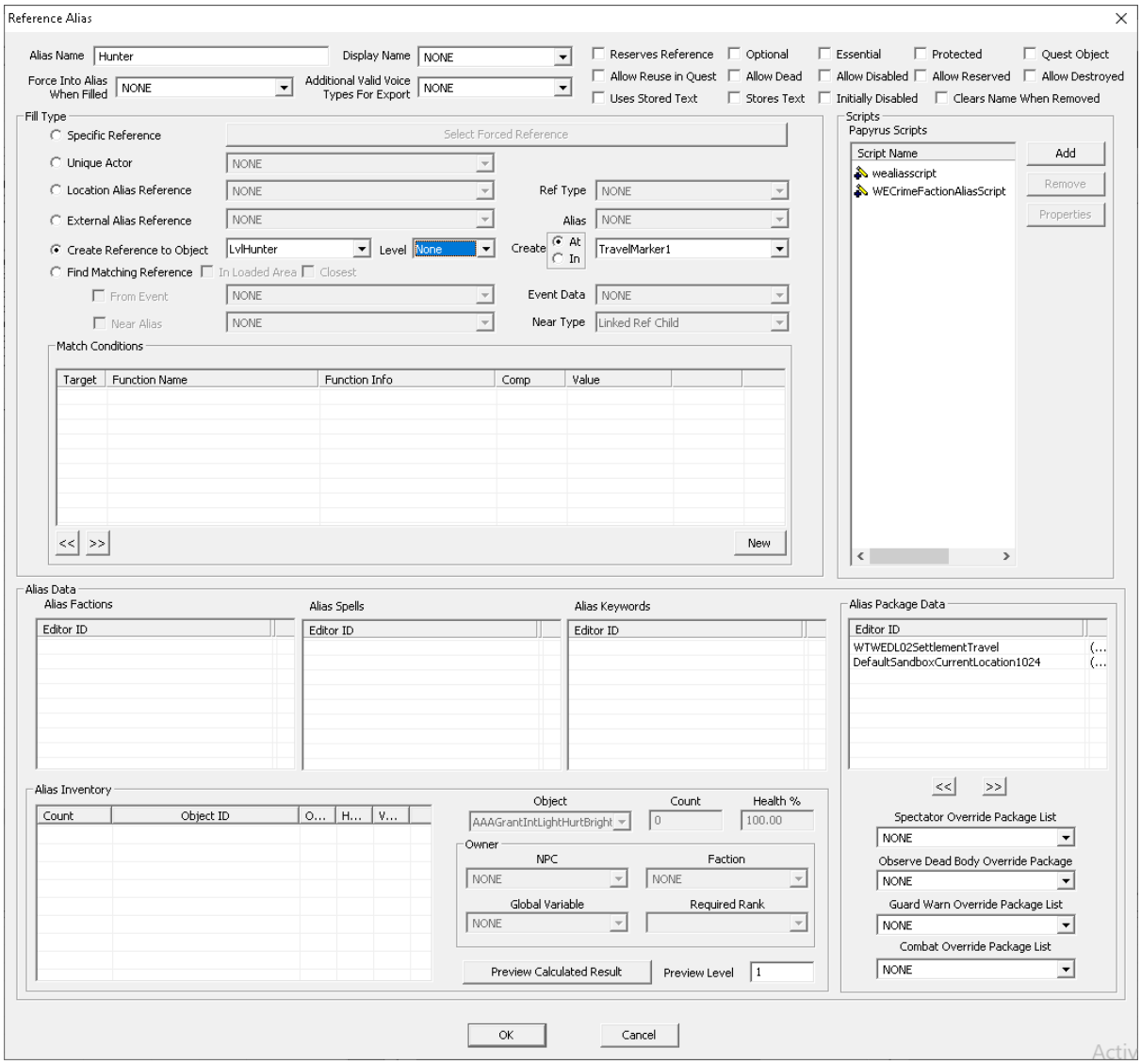


Figure 646 - Removed level scaling on the hunter.

Click OK to close out of the Reference Alias properties for the Hunter alias.

Now let's take a look at where exactly our hunter will be going.

Double-click on the Settlement alias or right-click on it and select Edit.

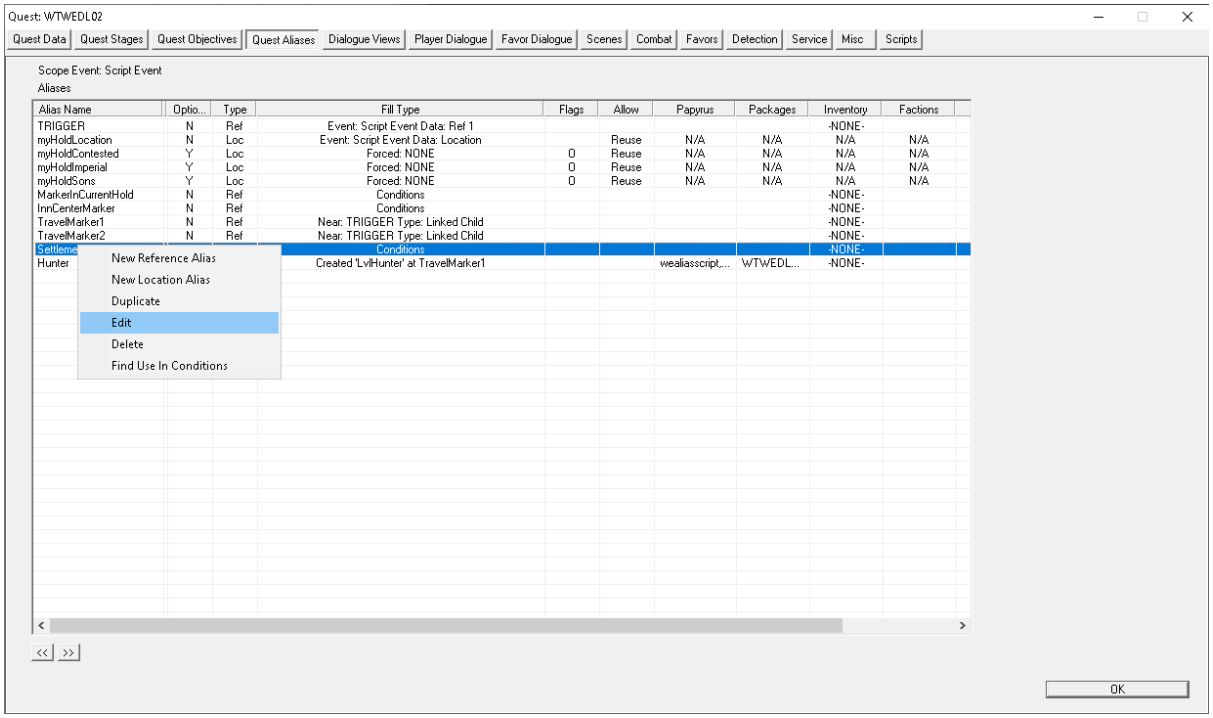


Figure 647 - Opening the properties for the Settlement alias.

So based on the conditions listed under Match Conditions, the location chosen must have a LocationCenterMarker and the CWCampImperial keyword.

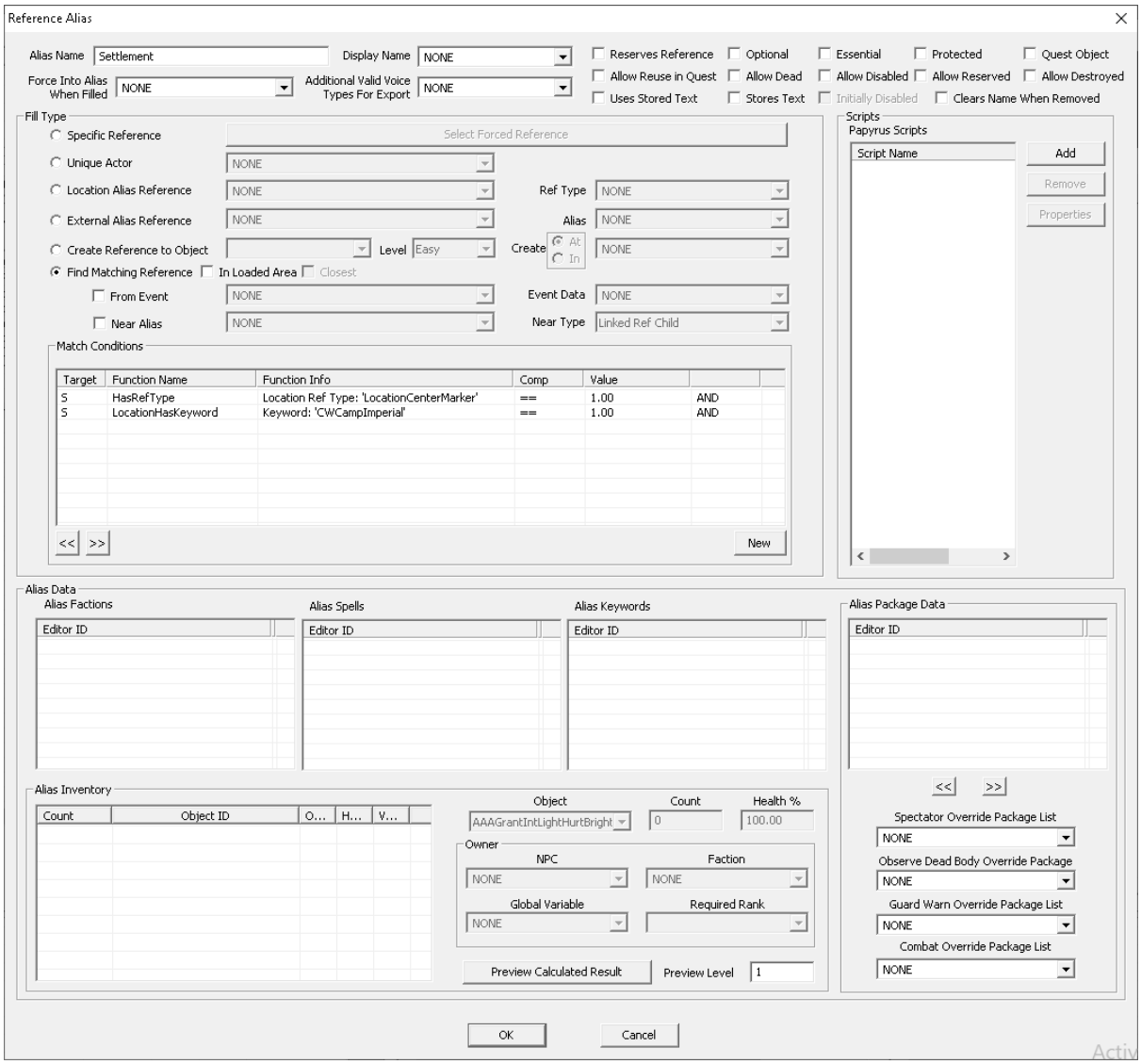


Figure 648 - Match Conditions for potential locations.

Let's change that keyword so our hunter actually goes to a settlement.

Double-click on the LocationHasKeyword condition.

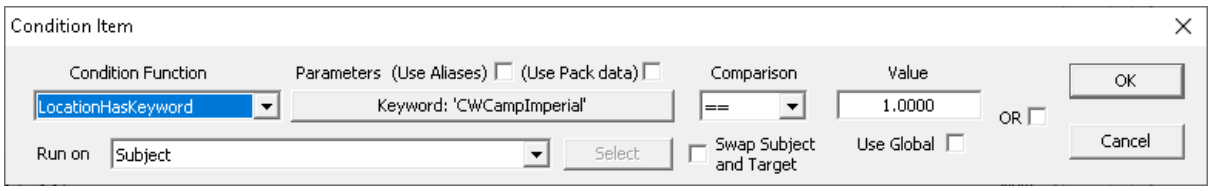


Figure 649 - Condition properties for the location keyword.

Click on the button with the CWCampImperial keyword.

Change CWCampImperial to LocTypeHabitationHasInn, then click OK.

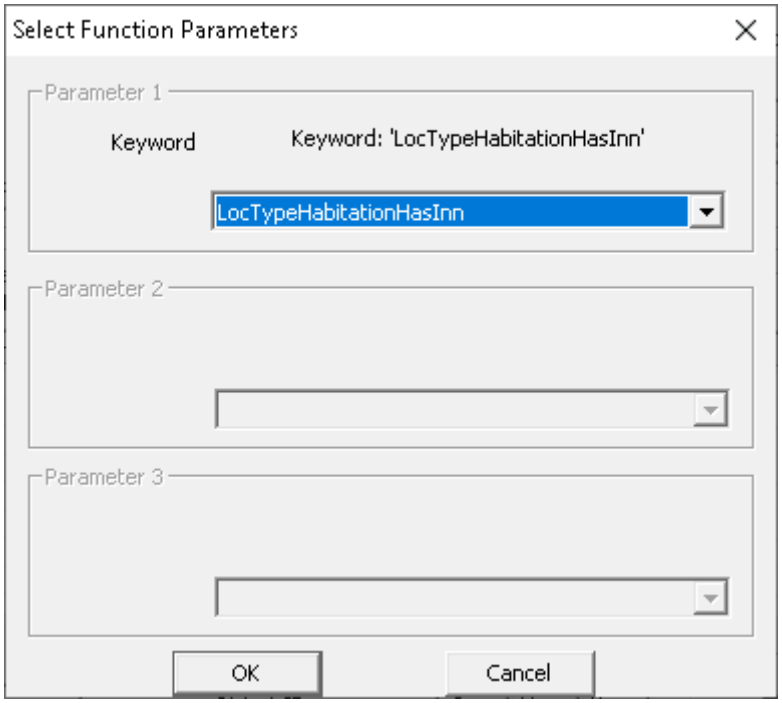


Figure 650 - Changing the keyword to LocTypeHabitationHasInn.

Our Condition Item properties should now look like this.

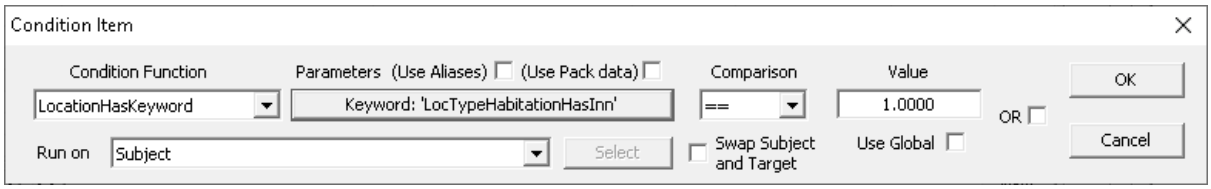


Figure 651 - Updated Condition Item with new keyword.

Click OK.

Our Reference Alias for the Settlement alias should now look like this:

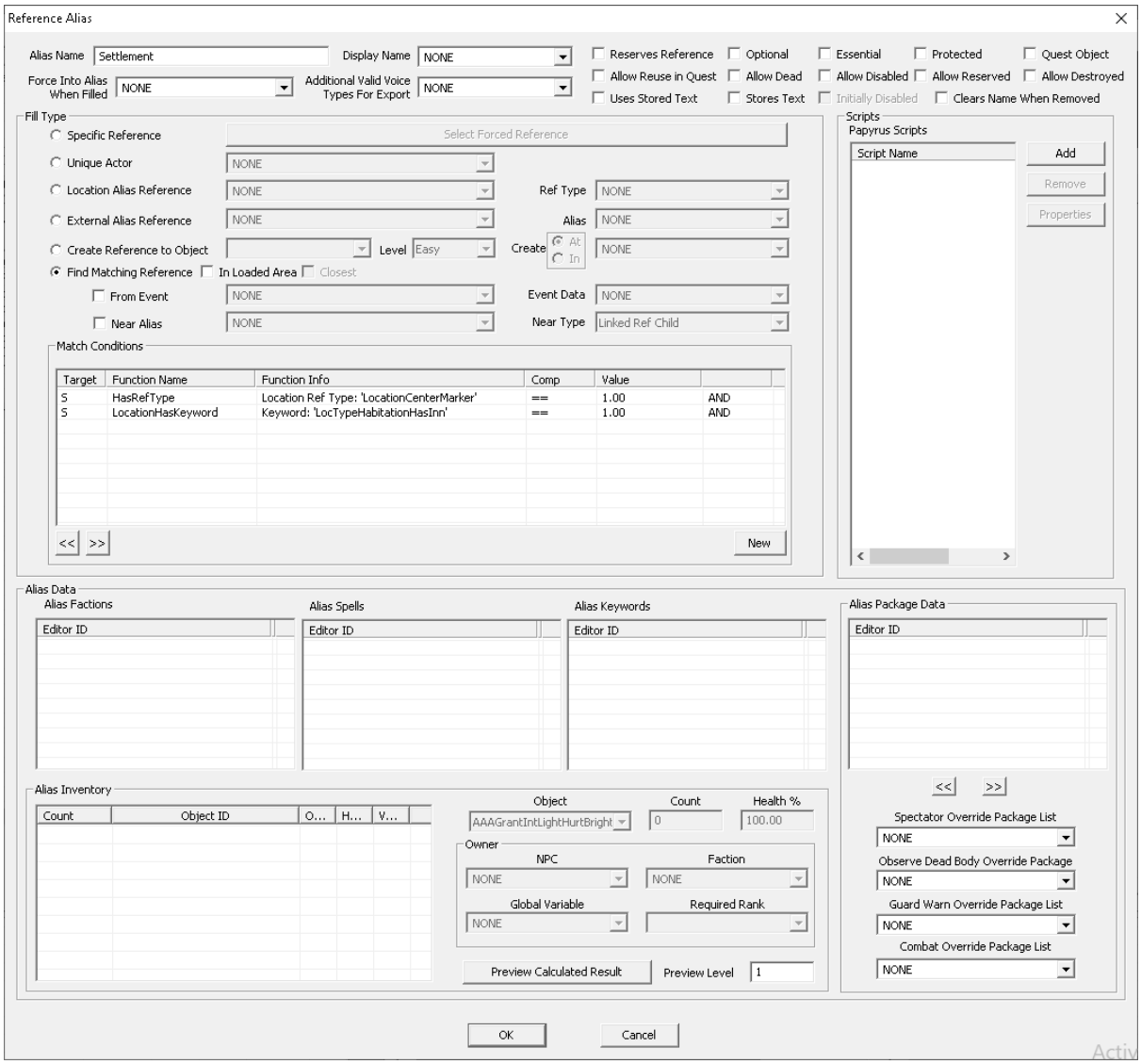


Figure 652 - Updated keyword condition in Reference Alias.

Click OK to close out of the Reference Alias properties.

Next, we need to update the scripting for the random encounter.

Go to the Quest Stages tab and click on stage 255.

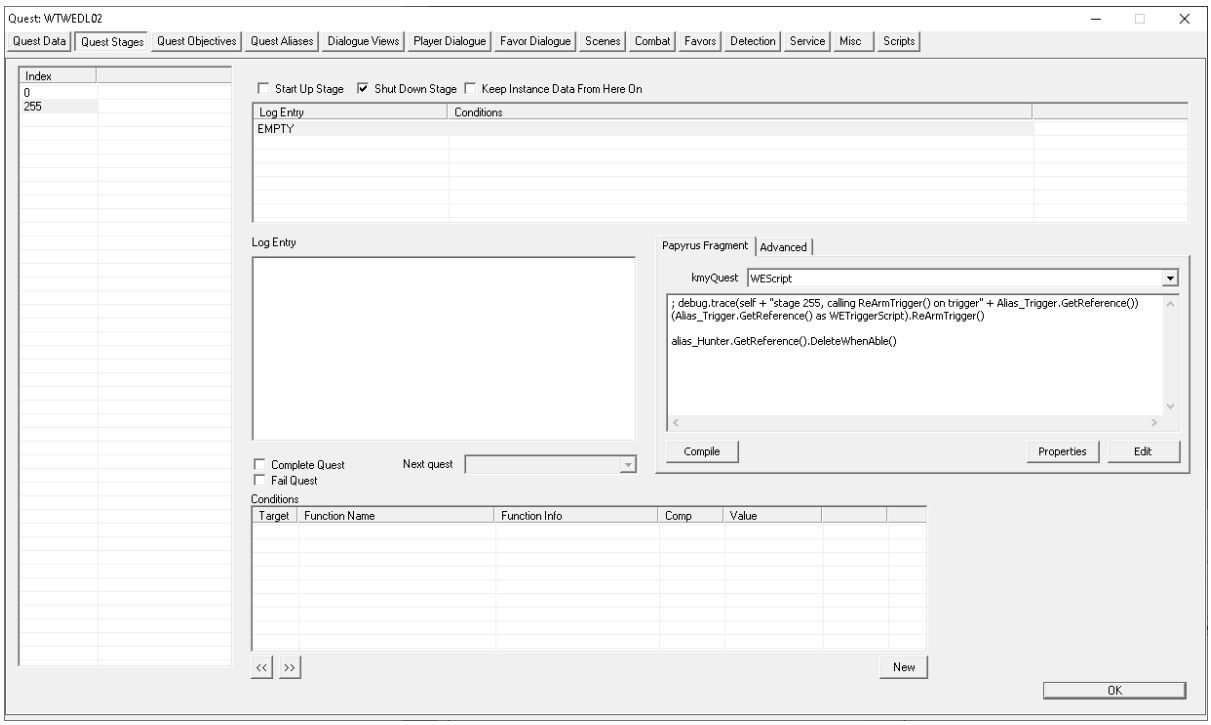


Figure 653 - Stage 255 of our random encounter.

To explain what's going on here, the scripting on stage 0 is run when our random encounter starts up. Seeing as though we're spawning in an NPC at TravelMarker1, we don't need to do anything here.

The scripting on stage 255 is run when our random encounter ends. We need to ensure any NPCs or objects generated by our random encounters are cleaned up to prevent save game bloat.

Change the line that reads:

```
alias_Bard.GetReference().DeleteWhenAble()
```

To:

```
alias_Hunter.GetReference().DeleteWhenAble()
```

Then click on the Compile button.

If you, for example, created an adversary for our Hunter to fight, like a bandit, you would also need to delete the bandit reference as well.

When dealing with unique NPCs, obviously we shouldn't be deleting them on stage 255.

For stage 0, we can move them into place with the following code:

```
if Alias_Aela.GetReference() != Alias_Follower.GetReference() &&
Alias_Aela.GetActorRef().IsDead() == false &&
Alias_Aela.GetActorRef().IsDisabled() == false

Alias_Aela.GetReference().MoveTo(Alias_TravelMarker1.GetReference())

endIf
```

So basically we're checking whether our unique NPC, Aela in this example, is our current follower, is dead, or is currently disabled. If all three conditions are False, then we'll move her to TravelMarker1. Otherwise, do nothing.

To check whether a unique NPC is our current follower, you would need to create a new Reference Alias named Follower and link it to the Follower alias in the DialogueFollower quest like so:

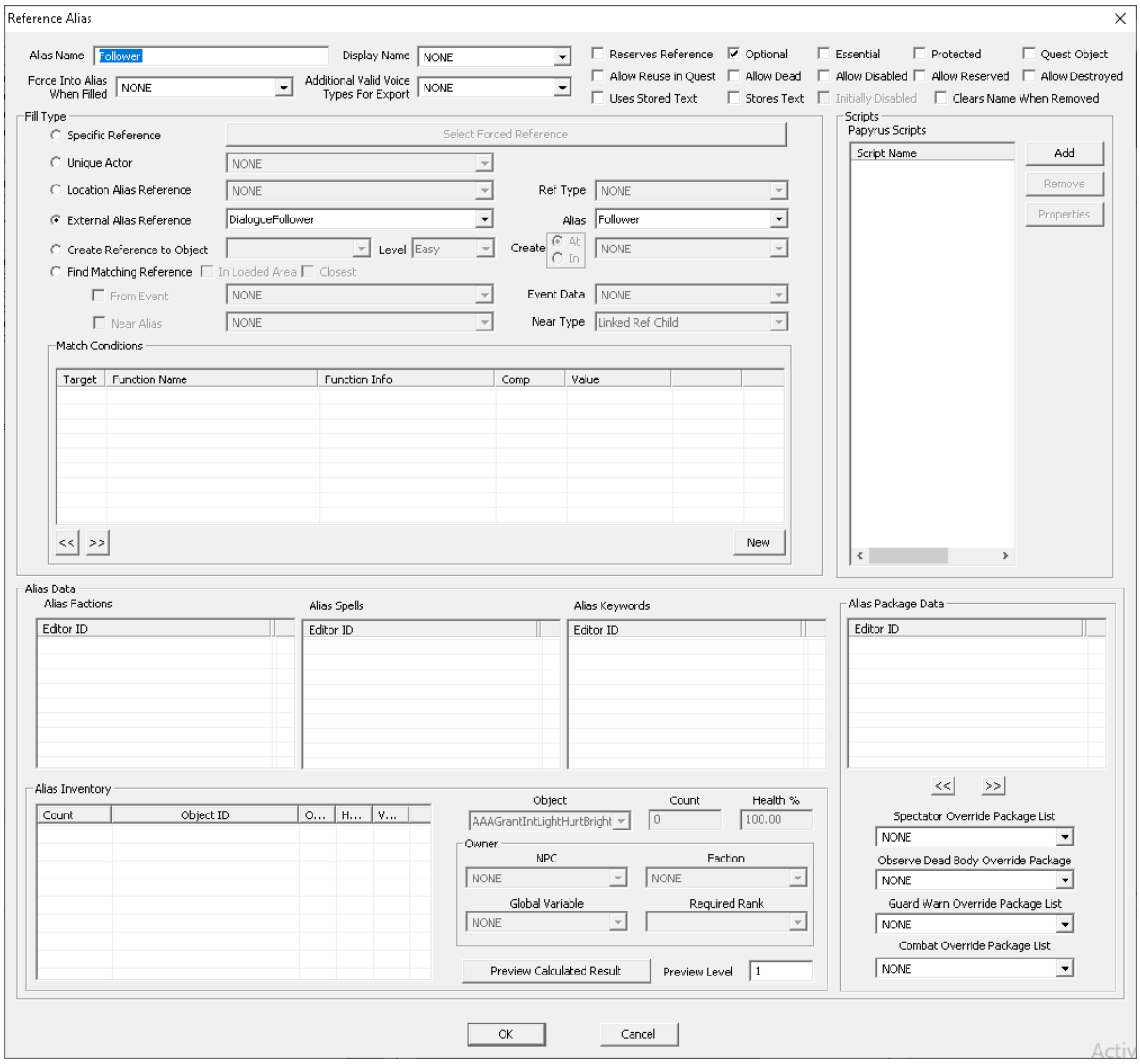


Figure 654 - Pulling alias data from another quest.

For stage 255 we can return a unique NPC to their original location with the following code:

```
if Alias_Aela.GetReference() != Alias_Follower.GetReference() &&
Alias_Aela.GetActorRef().IsDead() == false &&
Alias_Aela.GetActorRef().IsDisabled() == false

Alias_Aela.GetReference().MoveToMyEditorLocation()

endIf
```

Again, we're checking whether our unique NPC is currently our follower, is dead, or is currently disabled. If those conditions are all False, then we'll move the NPC back to their original location.

So let's get back to our Hunter random encounter.

Go to the Quest Data tab and change the Quest Name to provide a short description of what the random encounter is about. In my example, I set this to 'Hunter traveling to the nearest inn'.

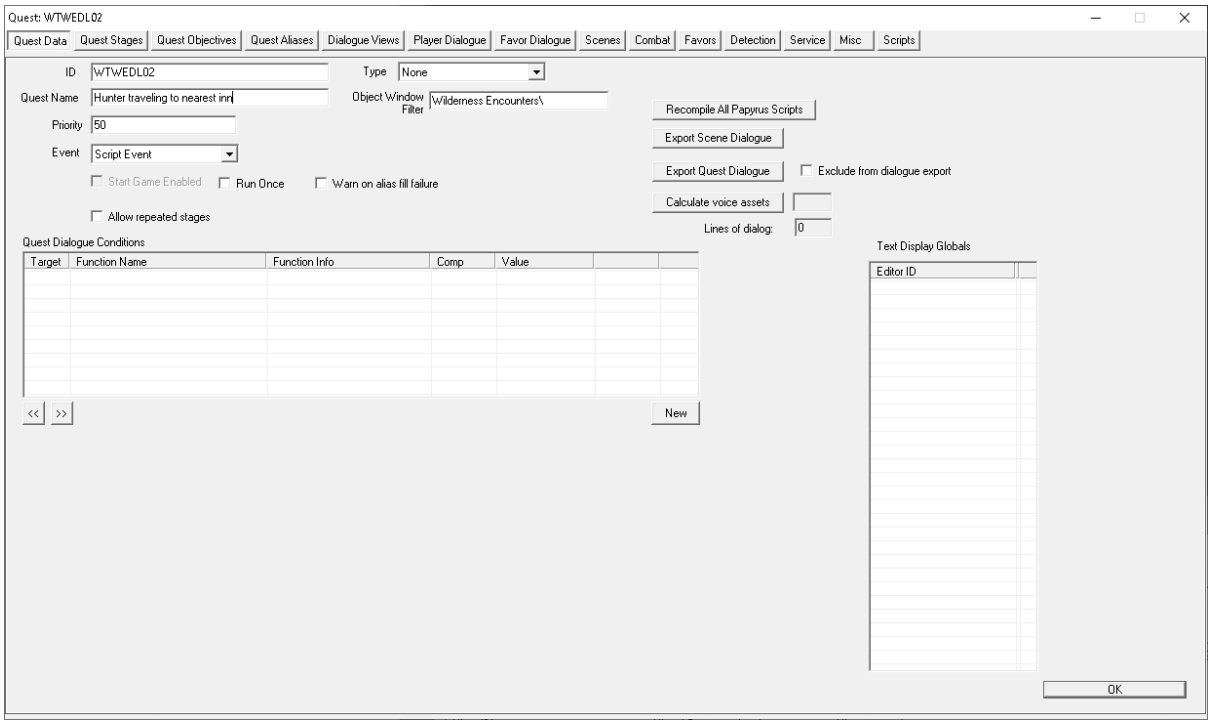


Figure 655 - Changing the description of the random encounter.

Click OK to close out of the Quest properties for our random encounter.

Now we need to add our random encounter to the Story Manager.

In the Object Window, go to Character > SM Event Node and double-click on Script Event or right-click on it and select Edit.

Expand the WildernessQuestNode you created in the previous section.

Right-click on WildernessQuestNode and select Add Quests.

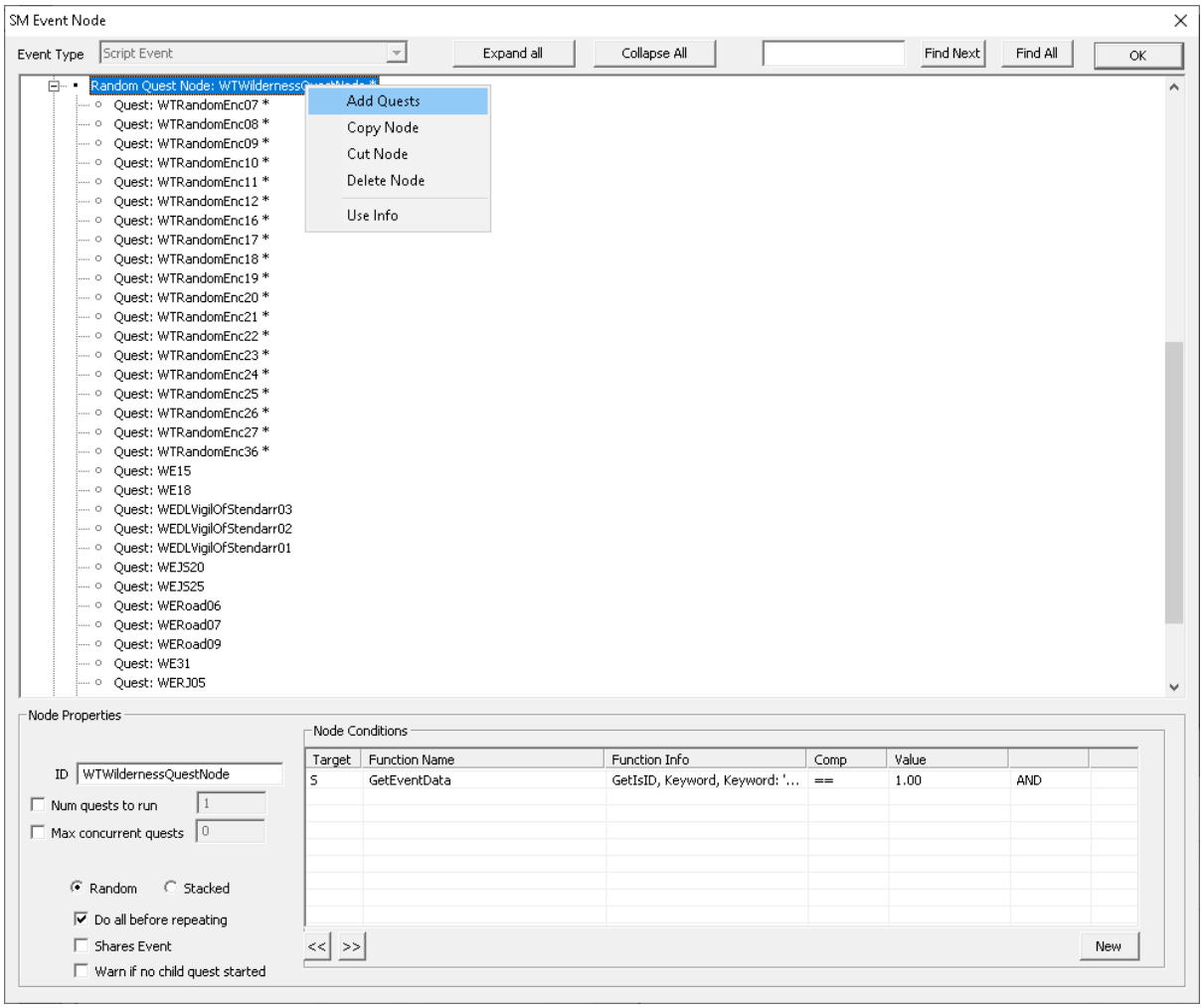


Figure 656 - Adding a new random encounter to WildernessQuestNode.

In my example I named the random encounter WTWEDL02 so that's what I'll be adding.

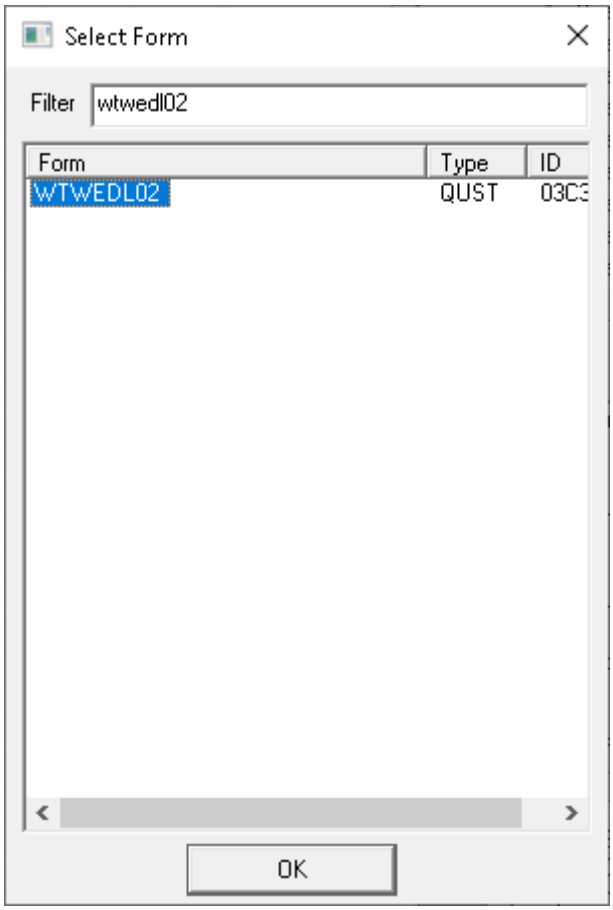


Figure 657 - Filtering by the quest ID of the random encounter we just made.

Click OK to add the random encounter to the Story Manager.

Confirm that the random encounter has been successfully added to the Story Manager.

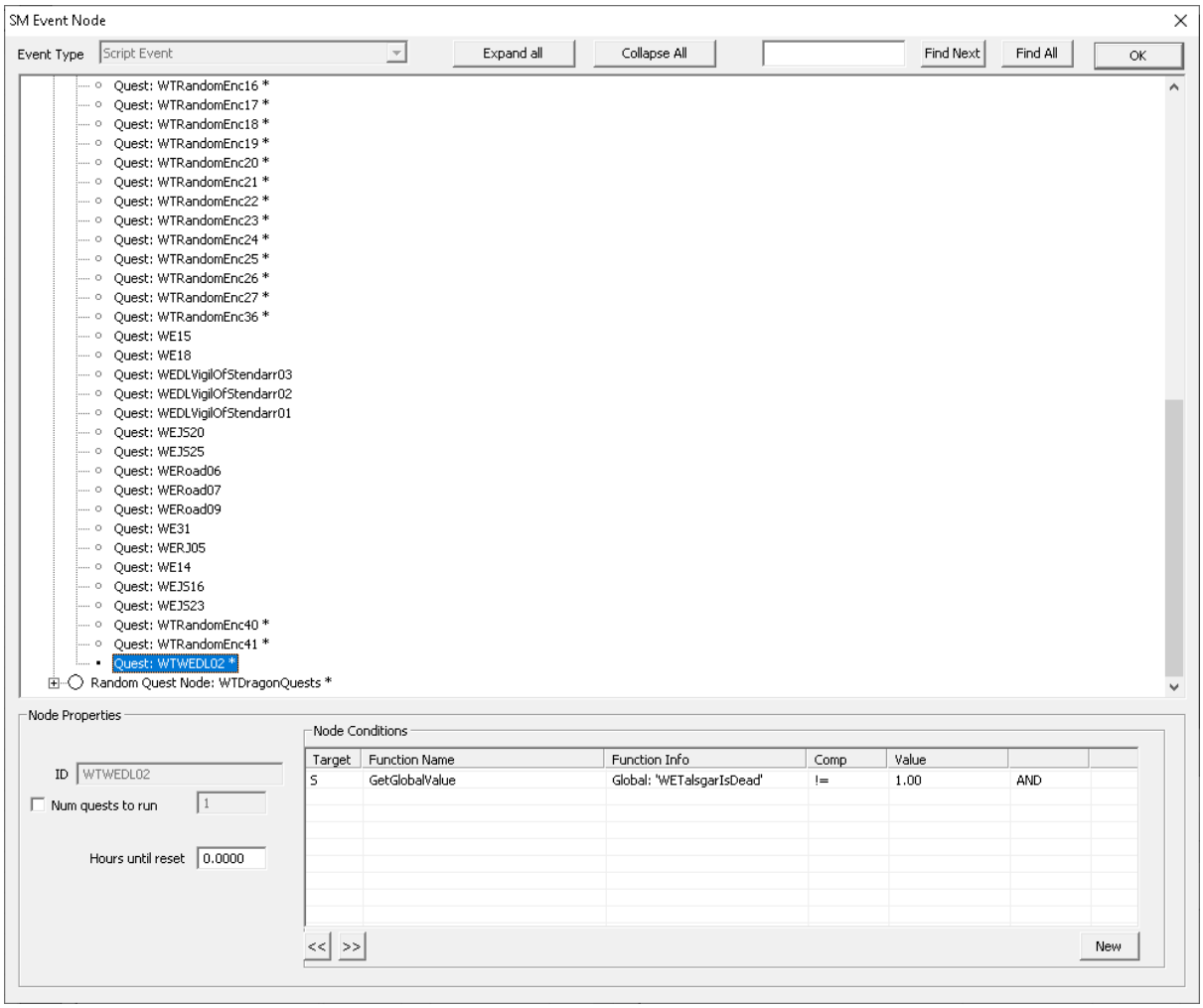


Figure 658 - Our new random encounter added to the Story Manager.

Highlight the GetGlobalValue condition checking the current value of 'WETalsgarIsDead'.

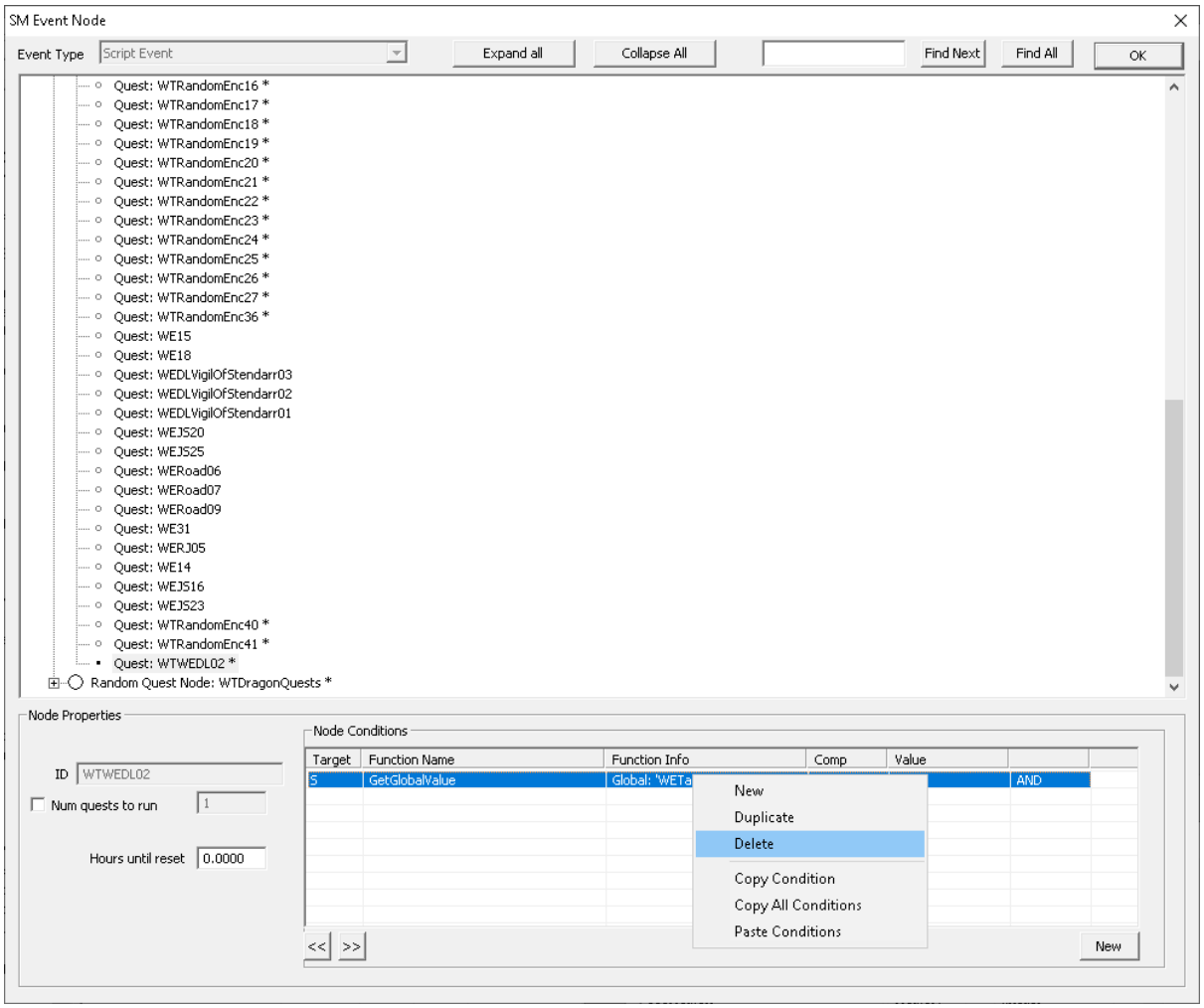


Figure 659 - Removing an existing condition.

Press Delete or right-click on it and select Delete to remove the condition.

Lastly, say for example we only want our random encounter to occur during the day. Well, we can add our own conditions to the random encounter in the Story Manager.

Right-click on Node Conditions and select New.

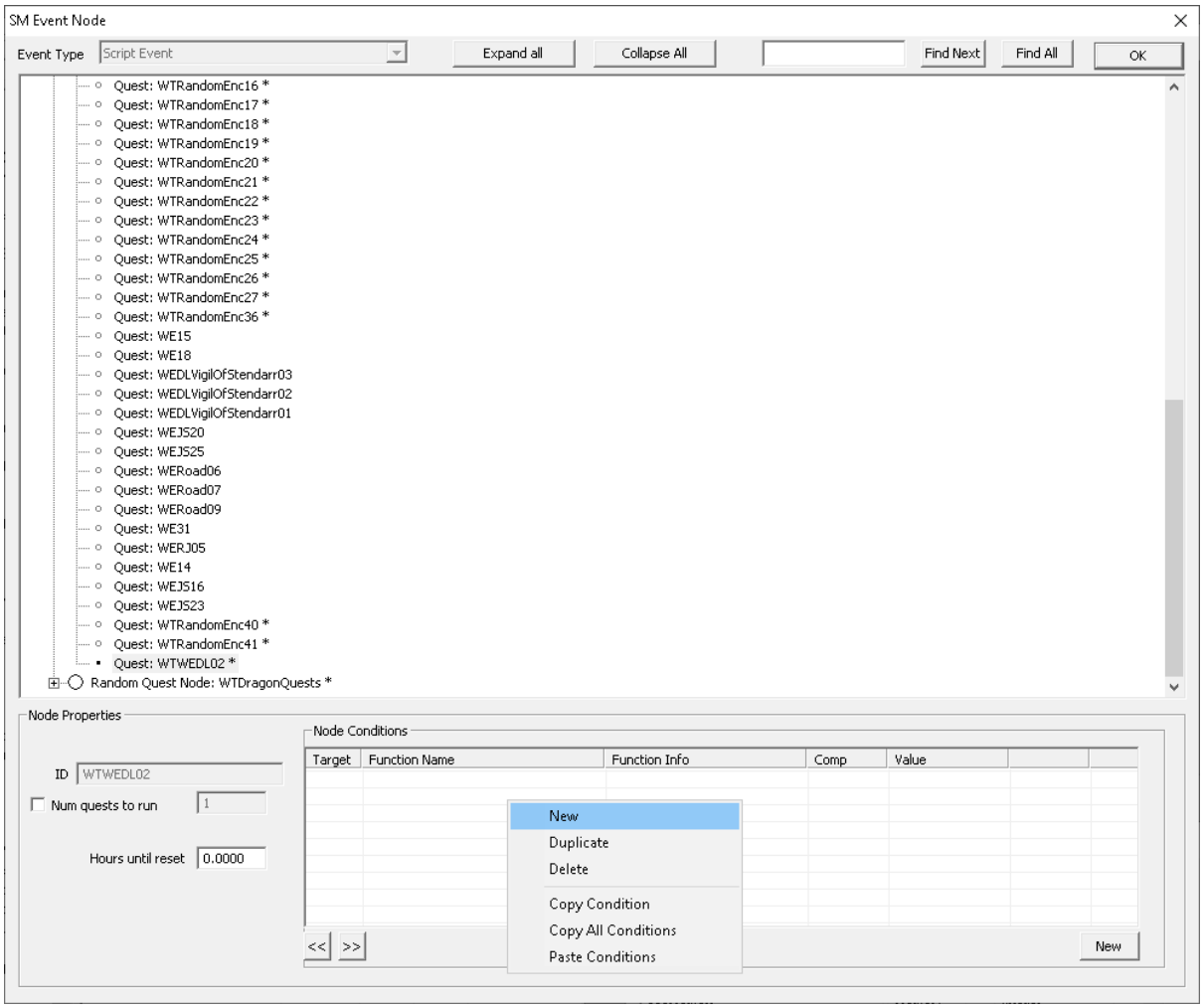


Figure 660 - Adding our own condition.

Set the Condition Function to GetCurrentTime. Set the Comparison operator to '>' and set the value to '7'. This means that it needs to be at least 7:00 AM for this random encounter to occur.

Click OK to close the Condition Item properties.

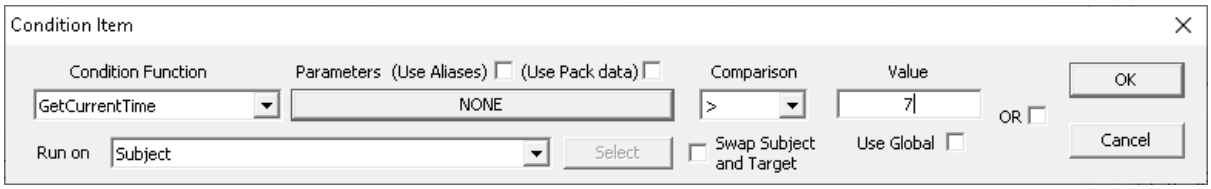


Figure 661 - Setting the first time condition.

Right-click and create another new condition.

For this one, set the Condition Function to GetCurrentTime, set the Comparison operator to '<' and set the Value to '20'. This means the random encounter can't occur later than 8:00 PM.

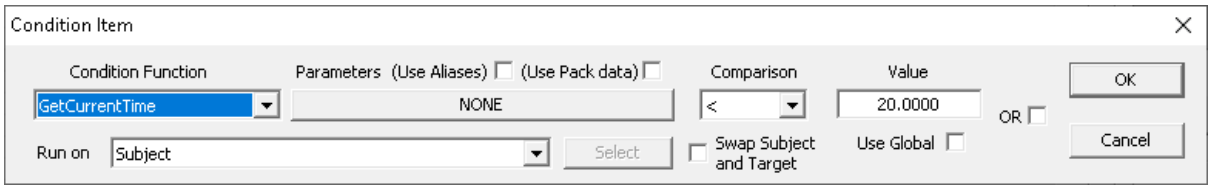


Figure 662 - Setting the second time condition.

Click OK to close out of the Condition Item Properties.

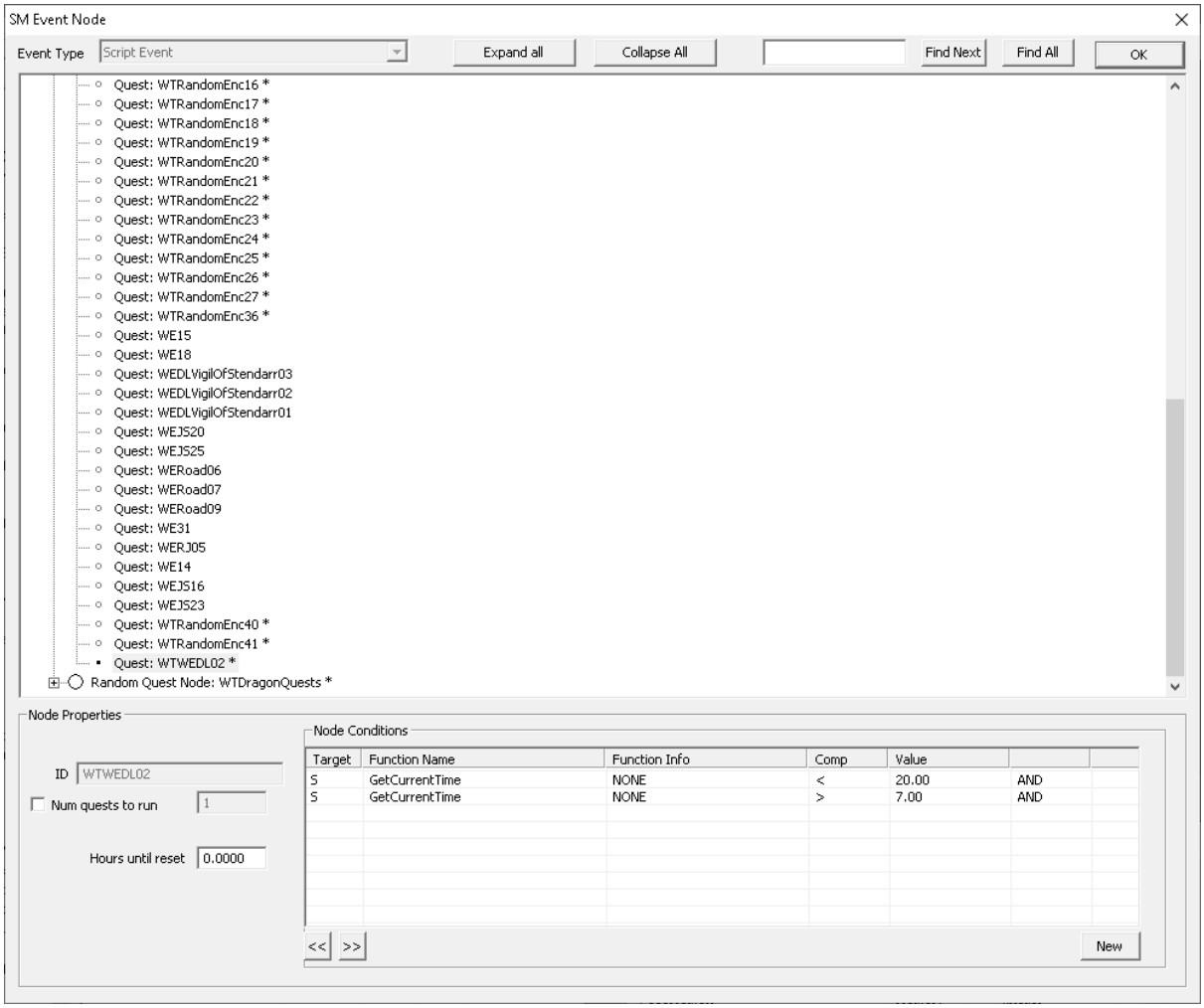


Figure 663 - New Node Properties on our random encounter.

We should now have two GetCurrentTime conditions on our WEDL02 random encounter as per the screenshot above.

For a full list of Condition Functions, see the article on the Creation Kit wiki.

Click OK to close out of the Story Manager.

CHAPTER 6: QUESTS, DIALOGUE AND VOICE ACTING

THE BASICS OF SETTING UP QUESTS

This section will cover the basics of setting up quests in the Creation Kit.

Before opening the Creation Kit, I would strongly recommend that you plan out your quest first. Start with a draft outline of your story and use that to help define the 'beats' of the quest and the dialogue between the player and the NPCs involved.

A simple fetch quest, for example, is generally comprised of the following four stages:

Stage 0 - The NPC is ready to give the player a quest.

Stage 10 - The NPC has told the player what to do or what to find and what kind of reward they should expect to receive.

Stage 20 - The player completes the task and needs to return to the quest-giver NPC.

Stage 30 - The player tells the quest-giver NPC that the task is done and has received a reward.

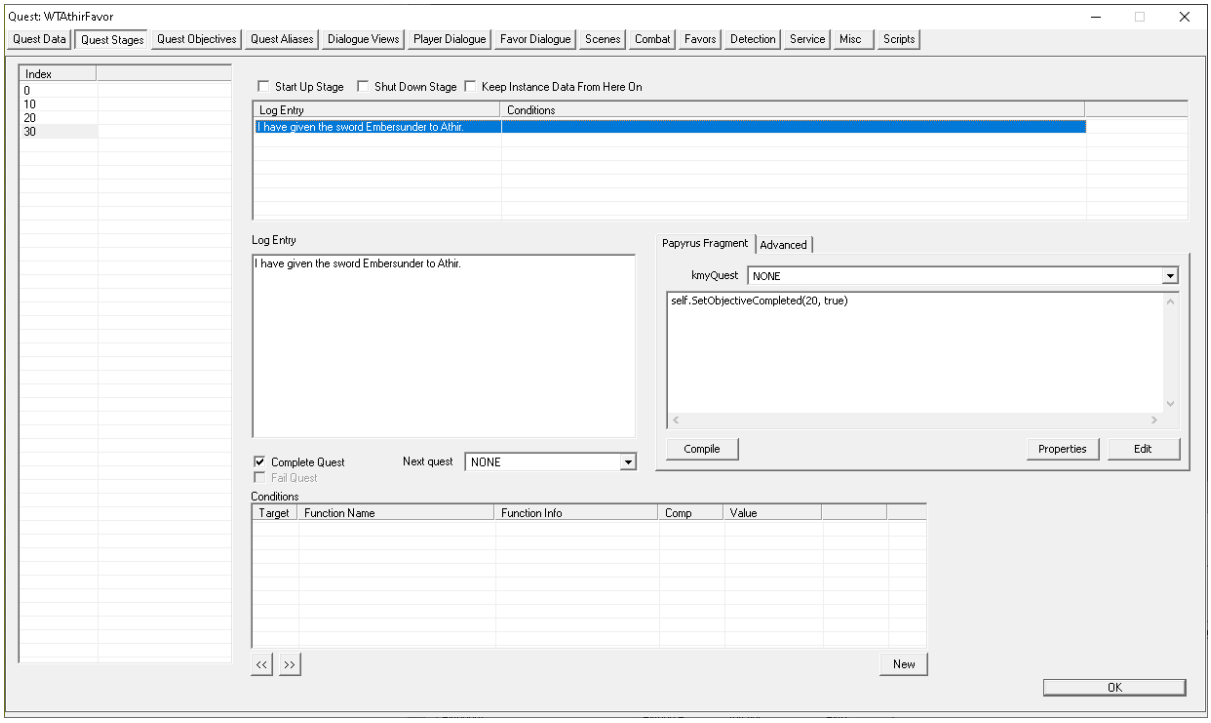


Figure 664 - A simple fetch quest.

You should also consider how your quest immerses the player in the game world, and what kind of reward the player will receive for completing it.

Teaching you how to write a questline is well beyond the scope of this tutorial, but I would highly recommend watching Josh Sawyer's GDC talk on Choice Architecture, Player Expression, and Narrative Design in Fallout: New Vegas.

Drawing the player's route on a screenshot of the world map can be a useful way to help plan out your quest.

The first quest in Wyrmstooth can be diagrammed as follows:

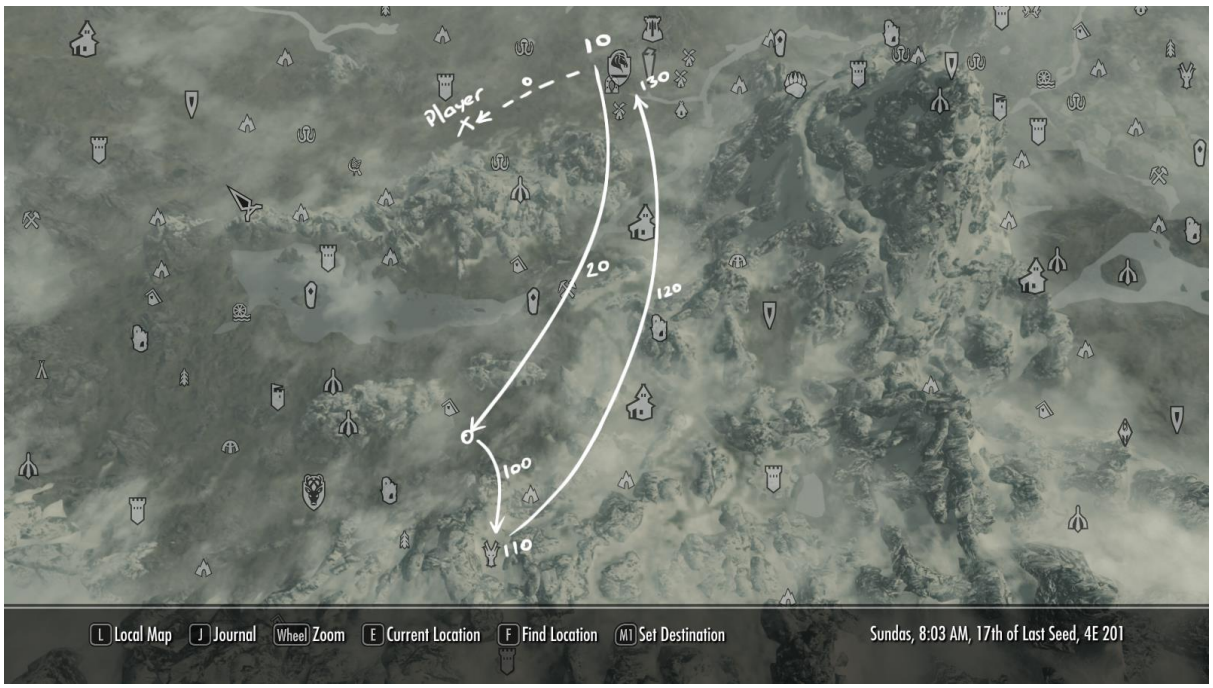


Figure 665 - Quest beat diagram of the first quest in Wyrmstooth.

Once the player meets quest starting requirements, an NPC named Theodyn Bienne spawns in at the Bannered Mare in Whiterun and travels to the player's current position to deliver a message (stage 0).

Once the player has talked to Theodyn, the current objective is to speak to his employer, Lurius Liore, currently staying at the Bannered Mare (stage 10).

Lurius will divulge more information about the threat and will instruct the player to meet up with some mercenaries along the road to Falkreath so that together you can team up and put the dragon to the sword (stage 20).

The player travels to Ancient's Ascent where the dragon is roosting (stage 100) and confronts the dragon. But the dragon knows the player is dragonborn and catches the player off-guard with a shout (stage 110).

While the player is disabled by the shout, the dragon delivers his exposition - an invitation of sorts to the island of Wyrmstooth, then flies away. One of the mercenaries instructs the player to return to the Bannered Mare to let Lurius know what transpired (stage 120).

The player returns to the Bannered Mare and speaks to Lurius (stage 130).

Note: There's a gap in the quest stages between stage 20 and 110 because originally I had the player hunt down each mercenary individually.

So let's now look at setting up a quest in the Creation Kit.

For this section, I'll be recreating a simple side quest from Wyrmstooth called 'Retrieving Embersunder'. It's a simple fetch quest that should teach you the basics of setting up quests in the Creation Kit.

In the Object Window, go to Character > Quest.

Right-click on one of the existing quests and select New.

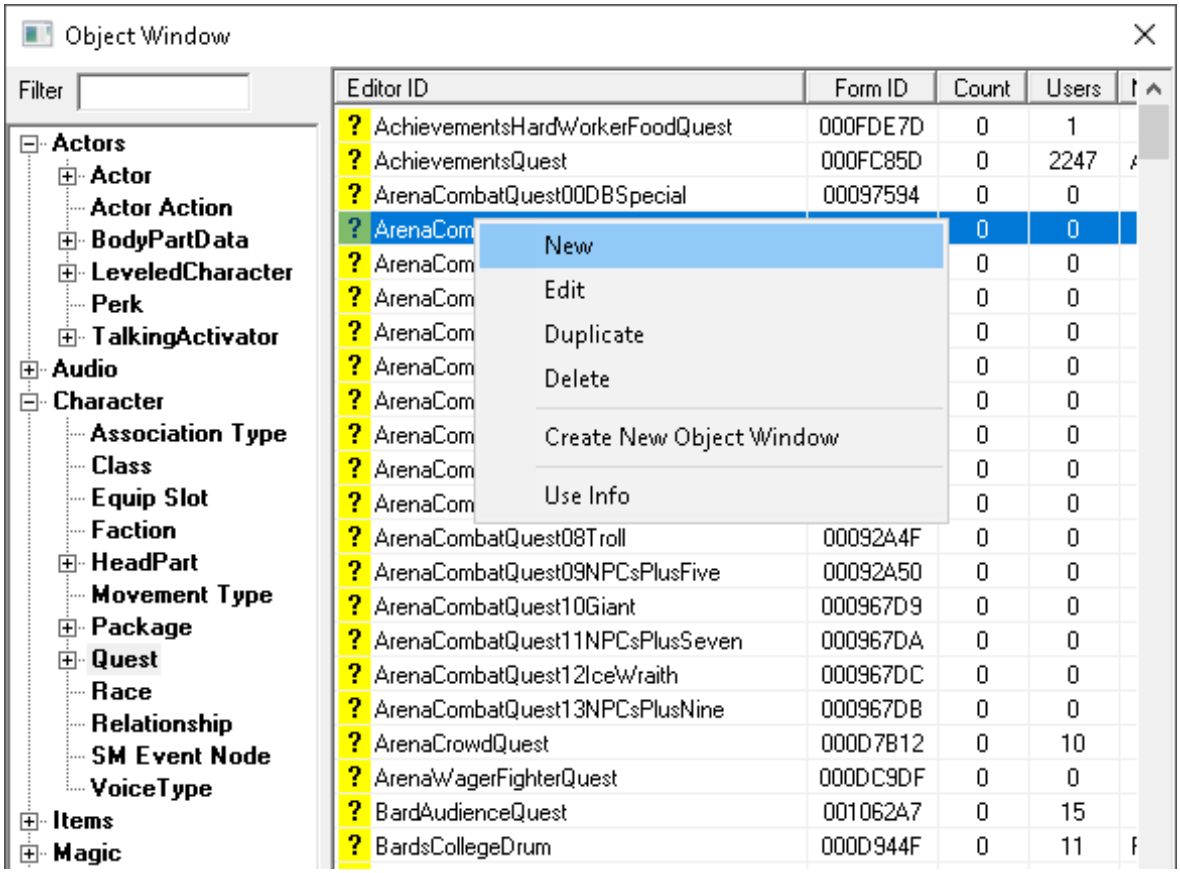


Figure 666 - Creating a new quest.

In the Quest Data tab, set the ID, Quest Name and Type. For my example, I just set the ID to WTTestQuest, the Quest Name to Test Quest and the Type to Side-Quests.

I also set the Priority to 60. This determines dialogue precedence. Dialogue in the DialogueViews tab from quests with a higher priority will have precedence over dialogue from quests with a lower priority.

Leave 'Start Game Enabled' and 'Run Once' ticked.

Start Game Enabled means this quest will start up as soon as the game loads. This means any quest dialogue defined in that quest will become available as soon as the game loads.

Run Once means this quest can only be completed once. For repeatable radiant quests, you're going to want to untick that.

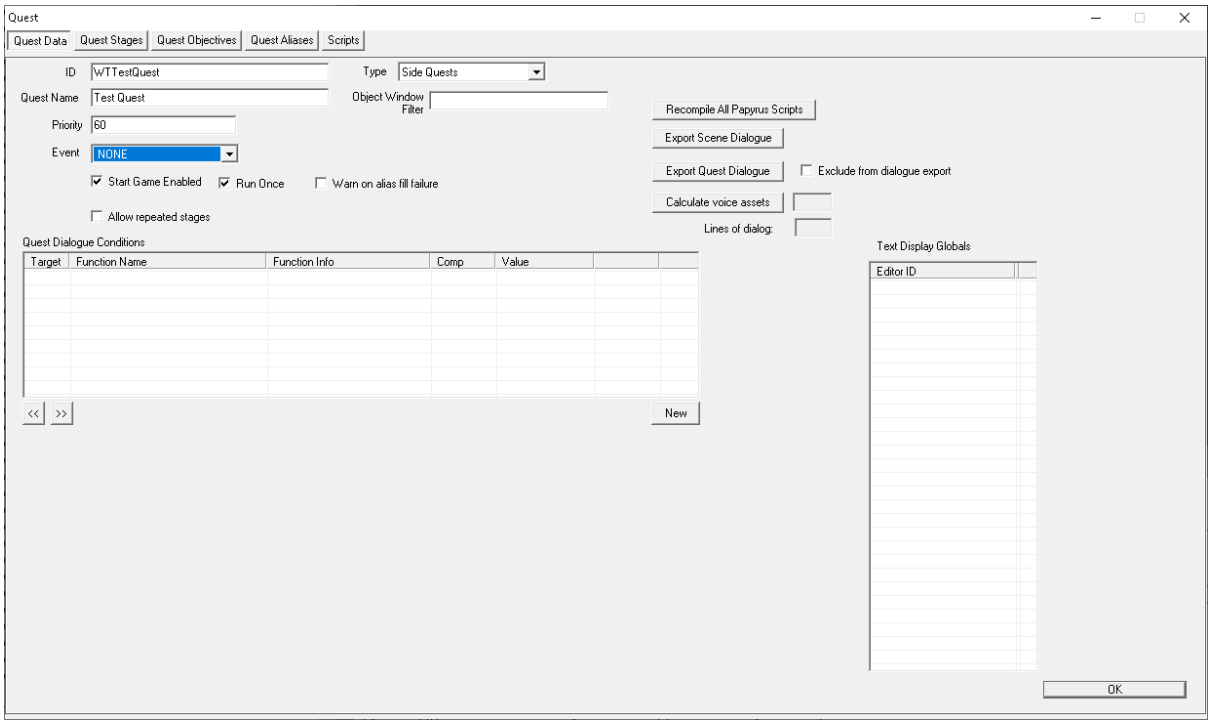


Figure 667 - Quest Data tab.

Important: Before doing anything else, click OK to close out of the Quest properties, then re-open the quest.

You should now see more tabs relating to quest dialogue.

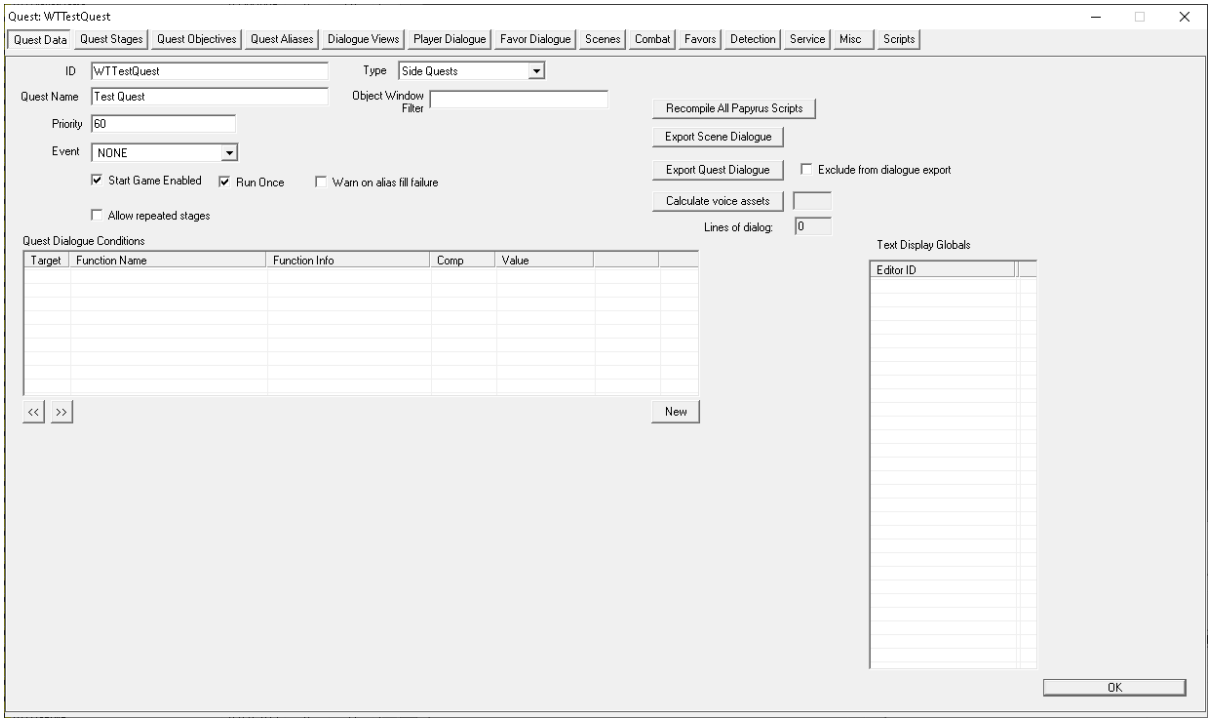


Figure 668 - More tabs in Quest properties.

Go to the Quest Stages tab. Right click on the quest stage list on the left hand side and select New to add a new stage.

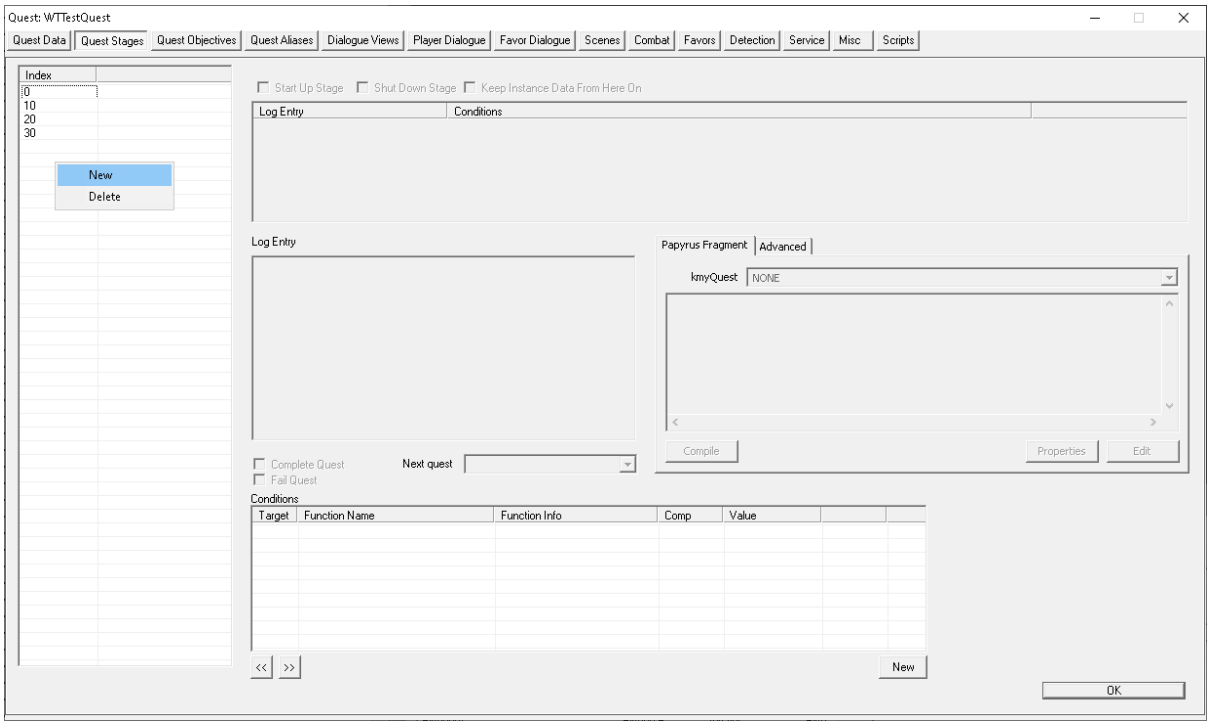


Figure 669 - Setting up quest stages.

I created four new stages: 0, 10, 20, and 30.

Right-click in the area under the 'Start Up Stage' tickbox and select New.

This will create a new log entry that will appear in your journal at that stage in the quest.

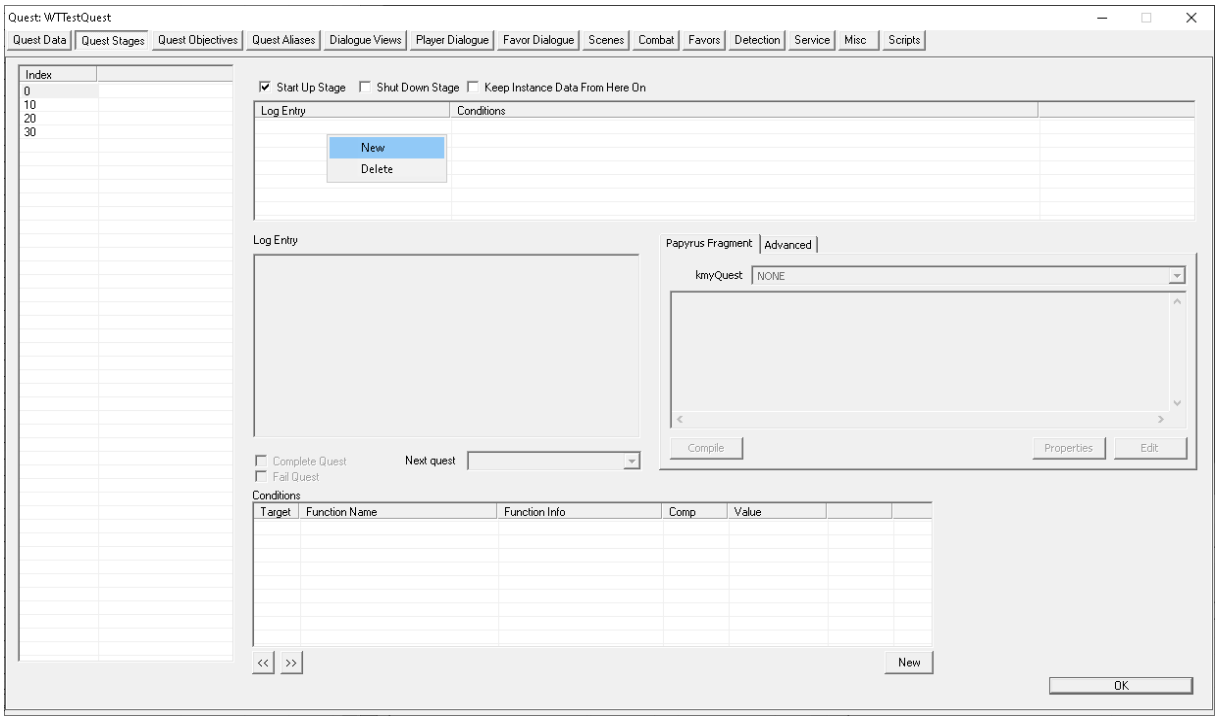


Figure 670 - Setting up the start-up stage.

It will also allow you to run papyrus on that stage.

Create a new log entry for stages 10, 20 and 30 as well.

We can leave the Log Entry fields blank for now.

Go to Stage 30.

Tick 'Complete Quest'. Advancing to this stage will flag the quest as completed in the player's journal.

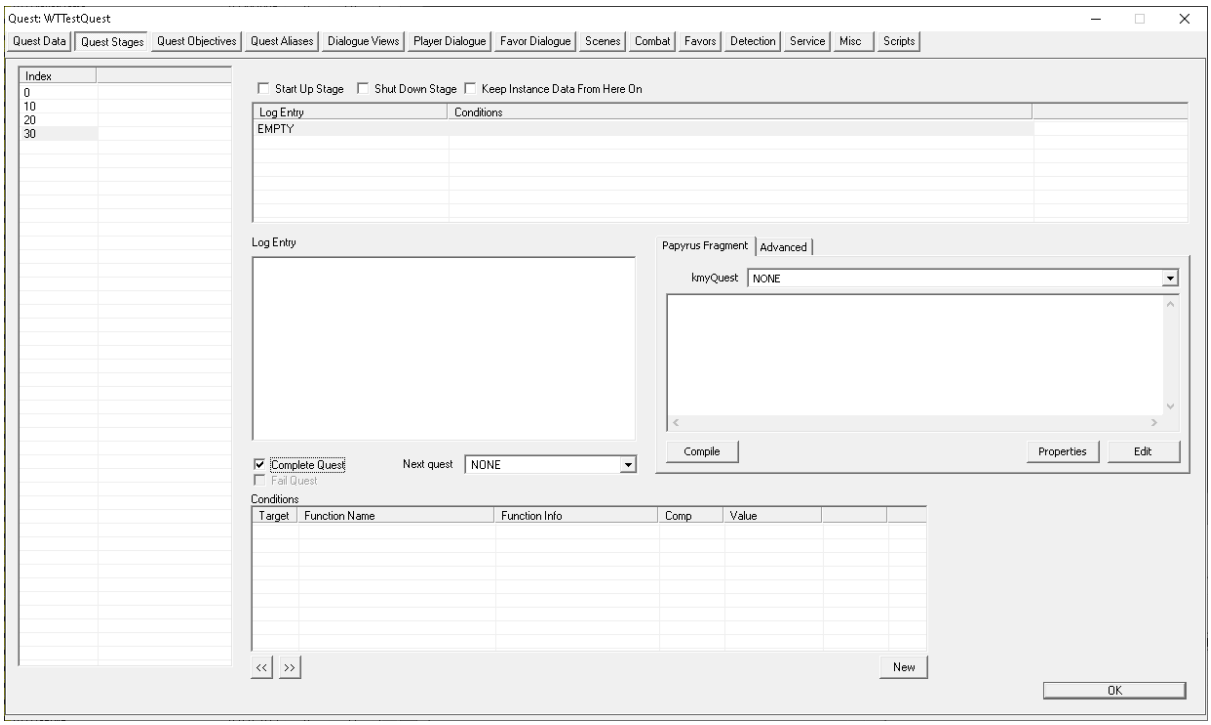


Figure 671 - Setting up the quest complete stage.

Go to the Quest Aliases tab.

Quest Aliases can be items, objects, NPCs or even locations that are involved in the quest.

For example, for a fetch quest, the item the player has to retrieve and the NPC giving the quest will both need to be added to the quest as reference aliases.

To add a reference alias, right-click in the aliases list and select New Reference Alias.

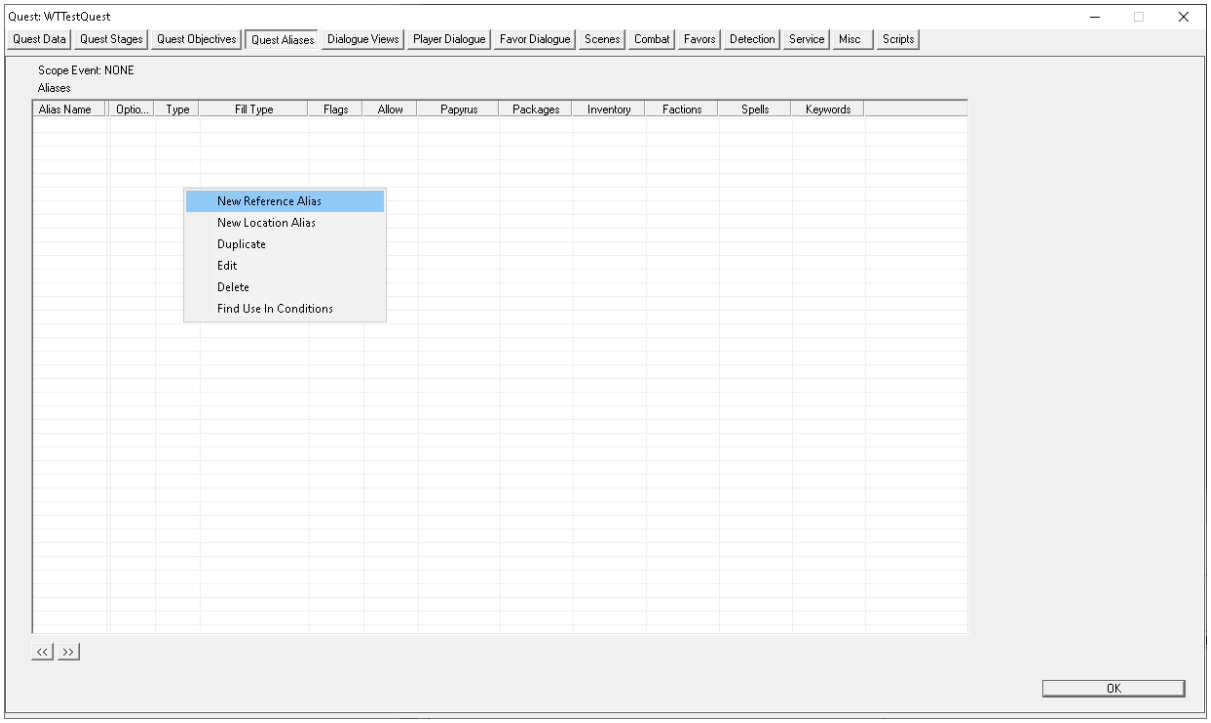


Figure 672 - Adding a new Reference Alias.

Enter in the name of the alias in the Alias Name field. For this example, I'm going to add in Athir, a unique NPC from Wyrmstooth.

Set the Fill Type to 'Unique Actor' then select the NPC from the drop-down to the right.

I also ticked the 'Essential' tickbox as I don't want this NPC to die while this quest is active.

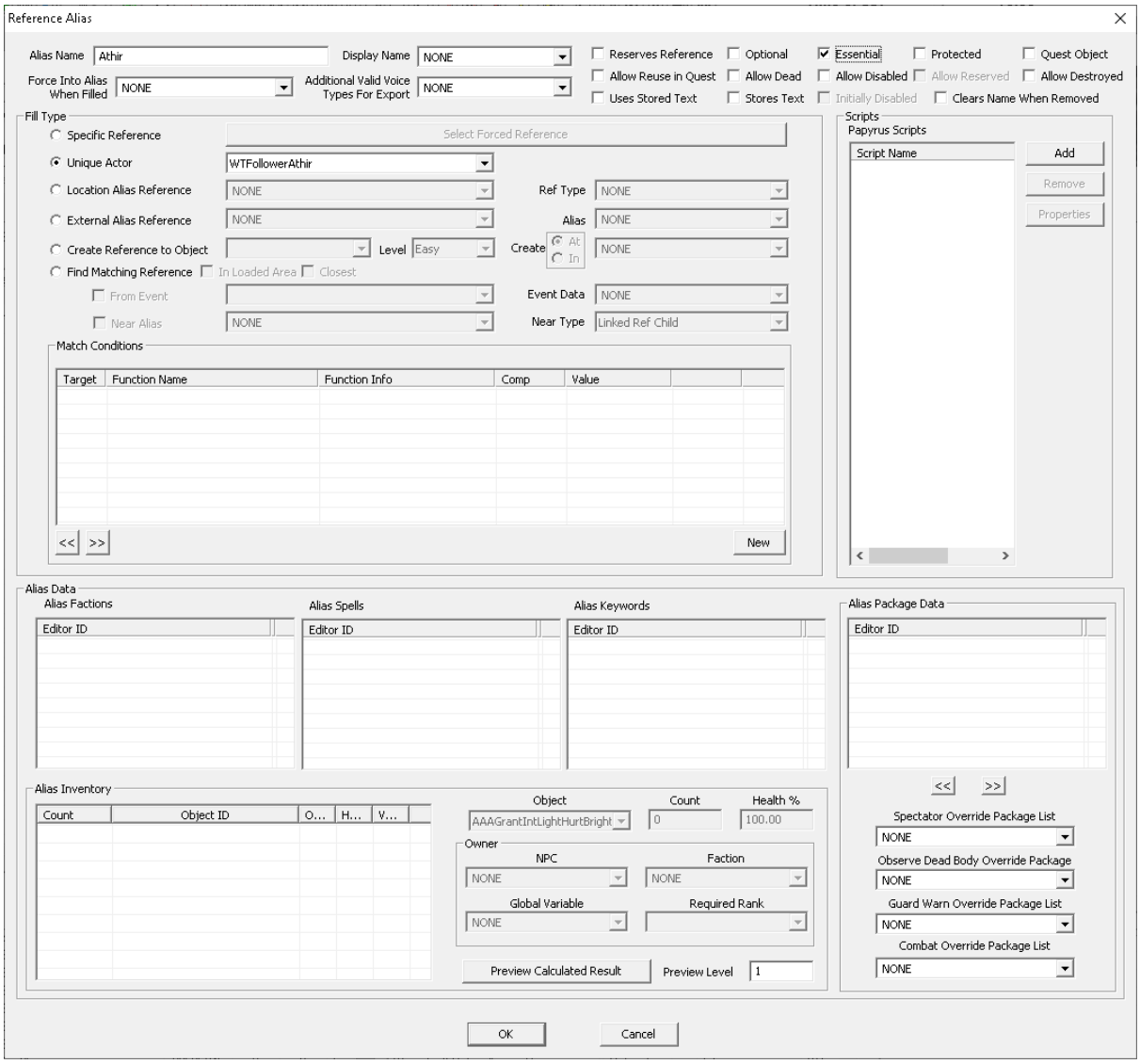


Figure 673 - Reference alias for Athir.

A 'unique' NPC is generally an NPC that has its own name, not something generic like 'bandit' or 'fox'.

Note: If you're selecting an NPC, use 'Unique Actor' where possible for the sake of compatibility. Selecting an NPC using Specific Reference will create an NPC edit, however selecting the same NPC through Unique Actor doesn't.

Next, let's select an item for the player to retrieve.

Right-click on the alias list again and select New Reference Alias.

Tick 'Quest Object' then set the Fill Type to 'Specific Reference'.

Quest Object flags this item as un-droppable.

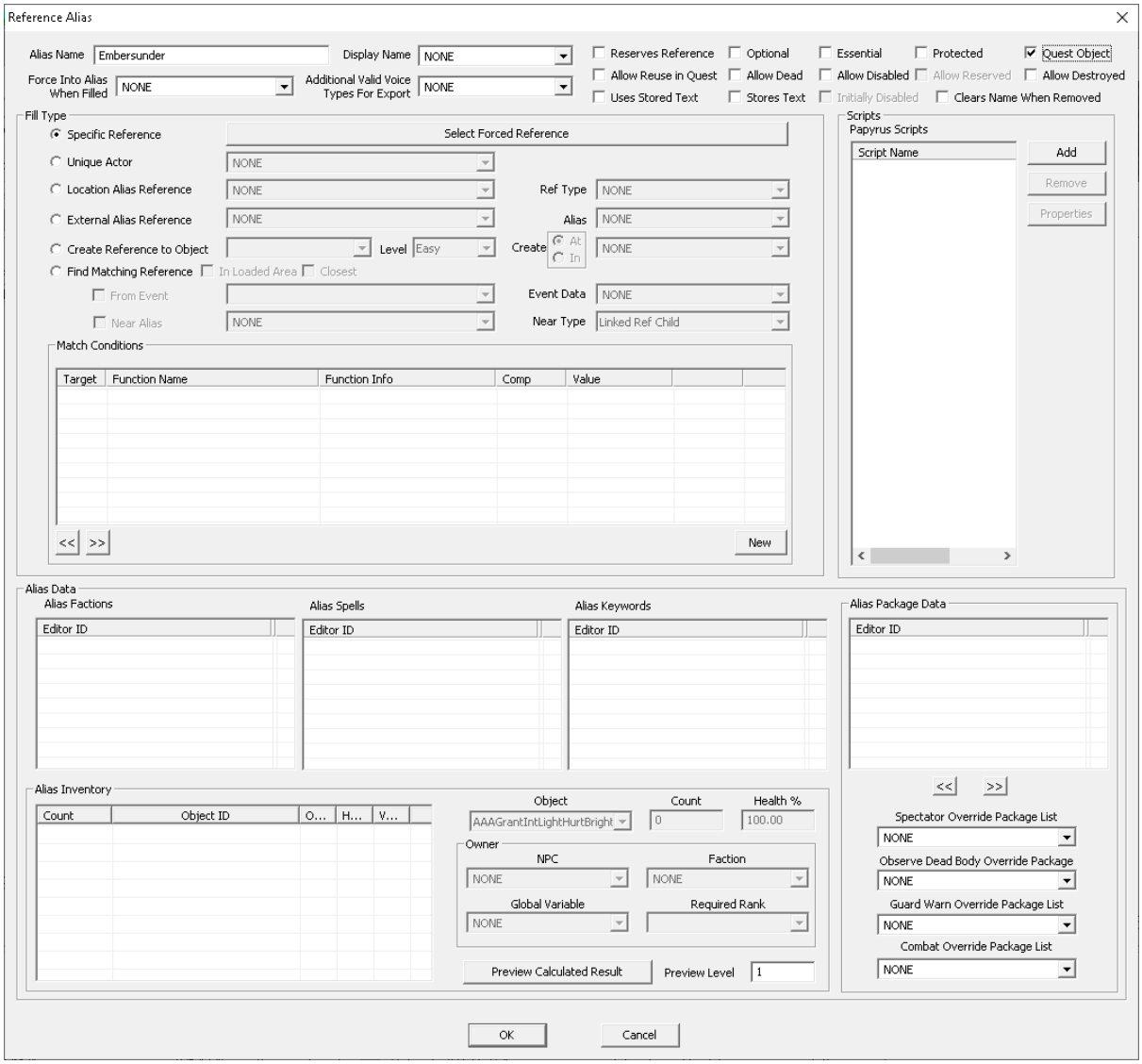


Figure 674 - Reference Alias for Embersunder.

To select the reference, click on the button next to it that currently reads 'Select Forced Reference'.

In the Choose Reference pop-up, click on 'Select Reference in Render Window'

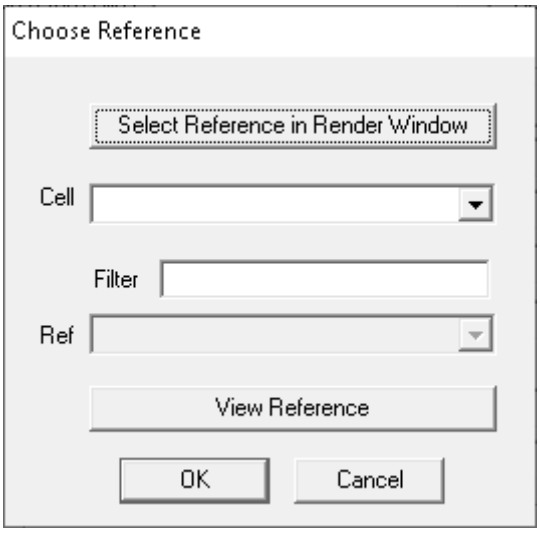


Figure 675 - Choose Reference pop-up.

This will change your mouse cursor to a crosshair in the render window.

The sword I want the player to retrieve currently exists in a special holding cell, which contains quest-related objects and NPCs related to Wyrmstooth that I want to have 'off-screen' until they're ready.

When the quest starts up, I'll be scripting the sword into place.

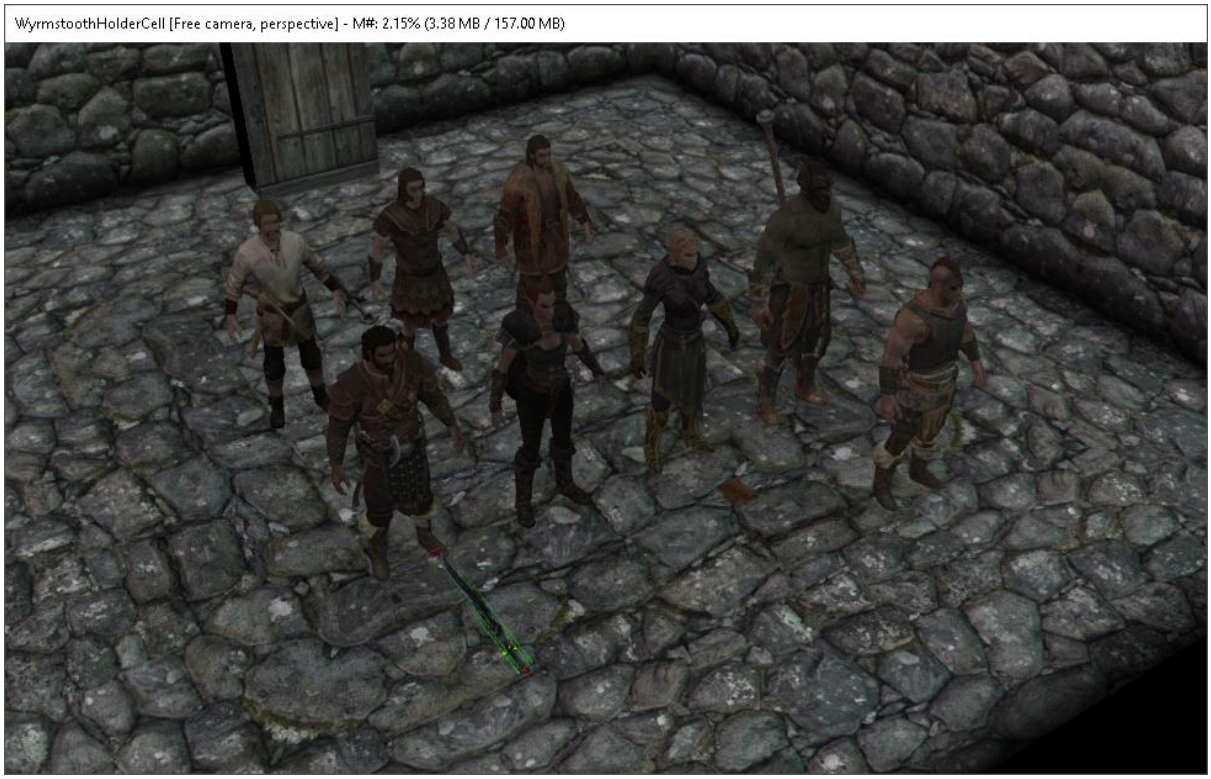


Figure 676 - A holding cell with quest-related NPCs and objects.

Double-click on the sword or the item you want the player to retrieve in the render window to select it.

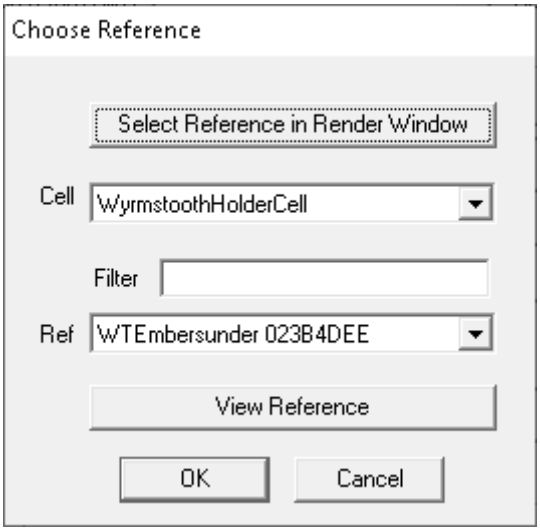


Figure 677 - Forced Reference.

Click OK to the Choose Reference pop-up.

Click OK to the Reference Alias properties.

Lastly, we need to define the location to move the sword to once the quest starts up.

Right-click in the alias list again and select New Reference Alias.

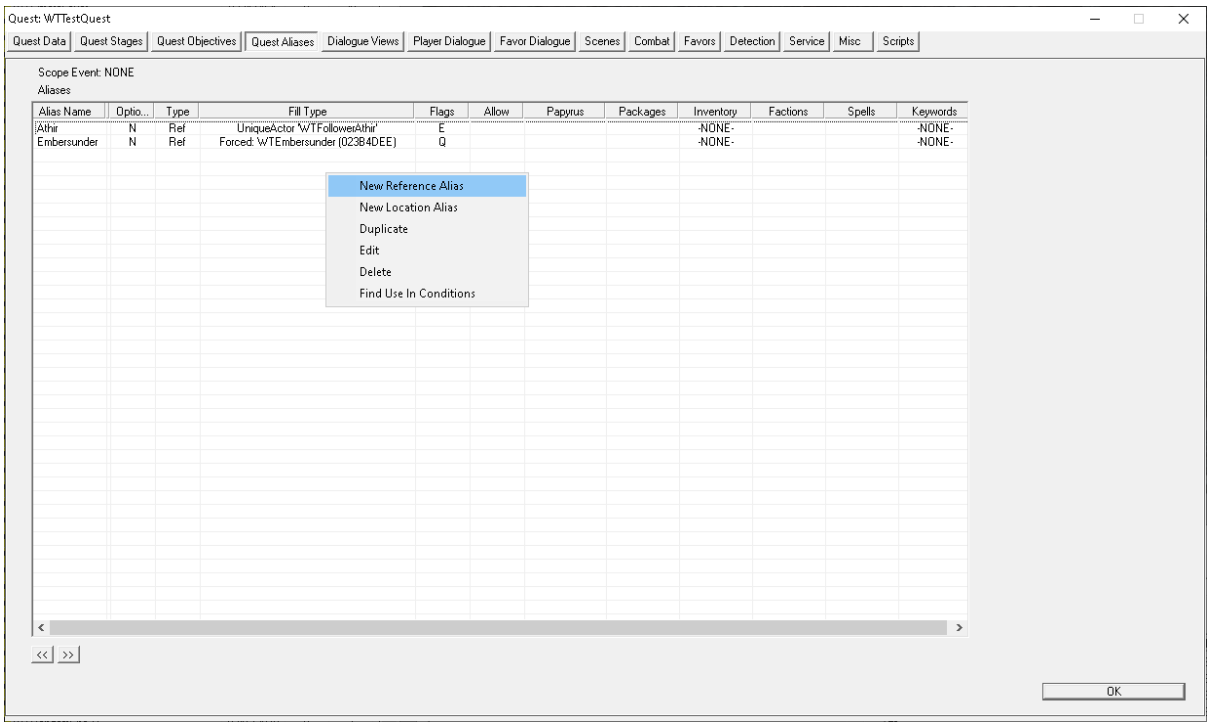


Figure 678 - Adding a third reference alias.

Set the Alias Name to BossLocation.

Set the Fill Type to Specific Reference then click on 'Select Forced Reference'.

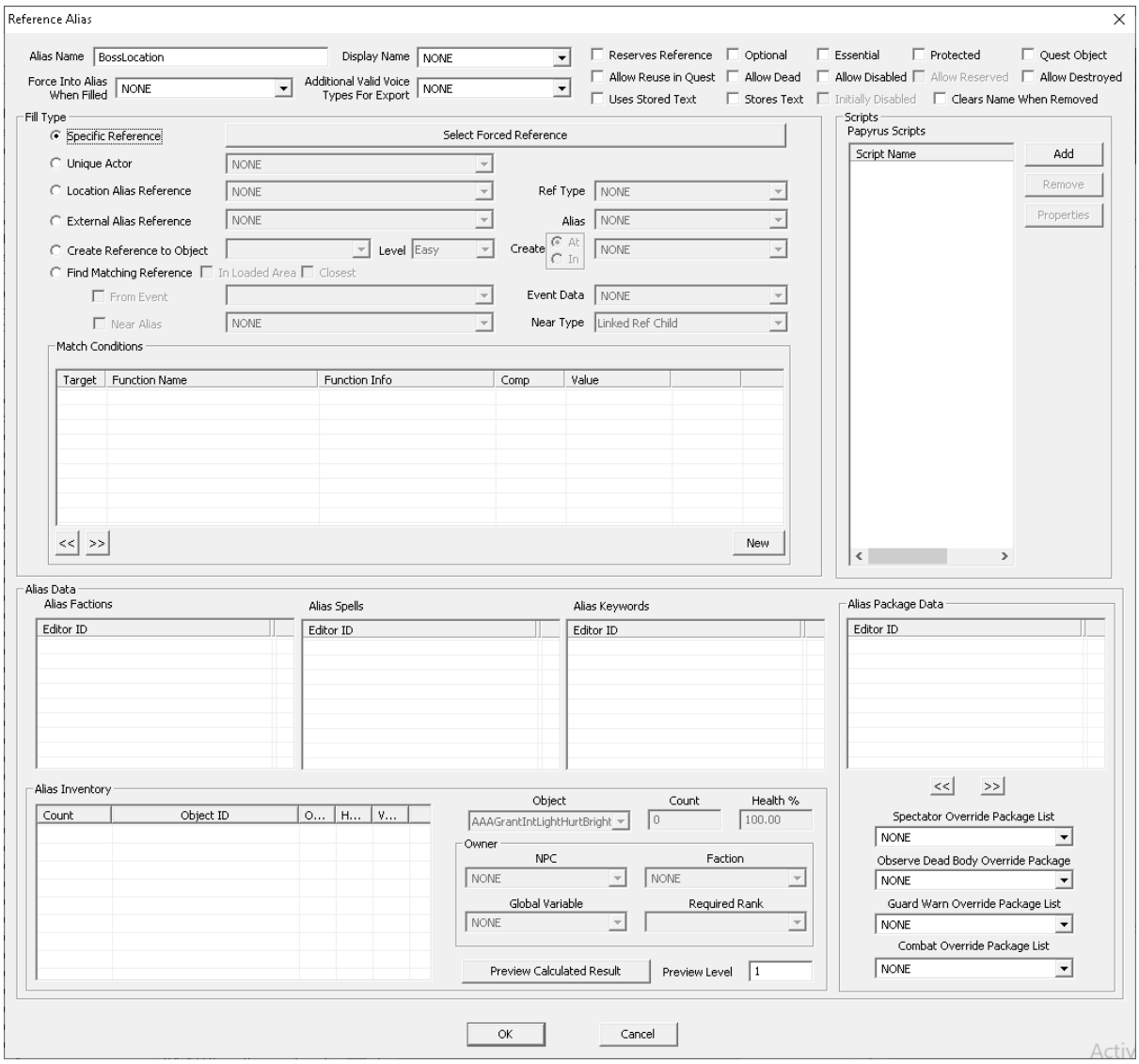


Figure 679 - Setting the BossLocation reference alias.

On the Choose Reference pop-up, click on 'Select Reference in Render Window'.

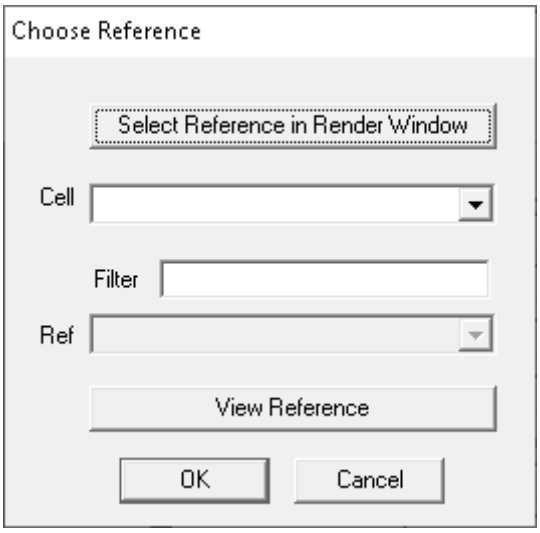


Figure 680 - Selecting the reference alias from the render window.

For this example, I'm going to select the BossTreasureMarker next to the boss chest at Hag's End. This is in the HagsEndOrigin cell of the DeepwoodRedoubtWorld world space.



Figure 681 - Selecting the location to move the sword to.

Click OK to the Choose Reference pop-up.

Click OK to the Reference Alias properties.

We should now have three quest aliases as per the screenshot below.

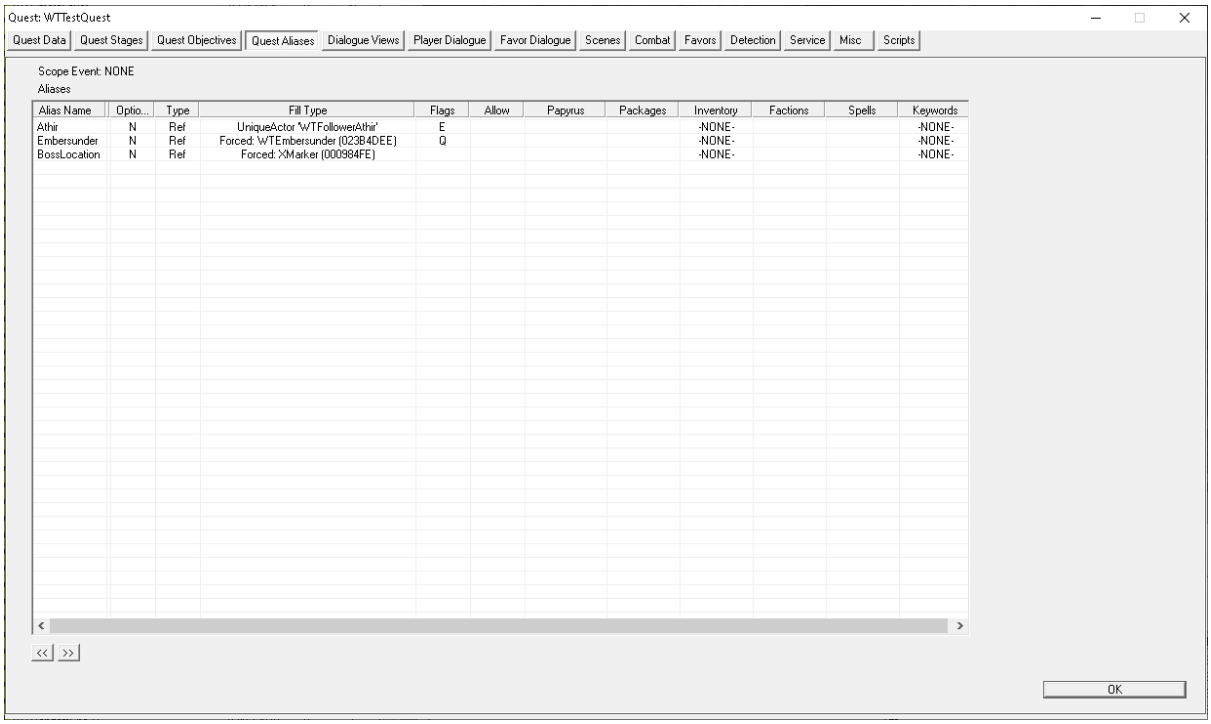


Figure 682 - Our quest aliases added.

Go to the Quest Objectives tab.

Right-click in the objectives list and select New.

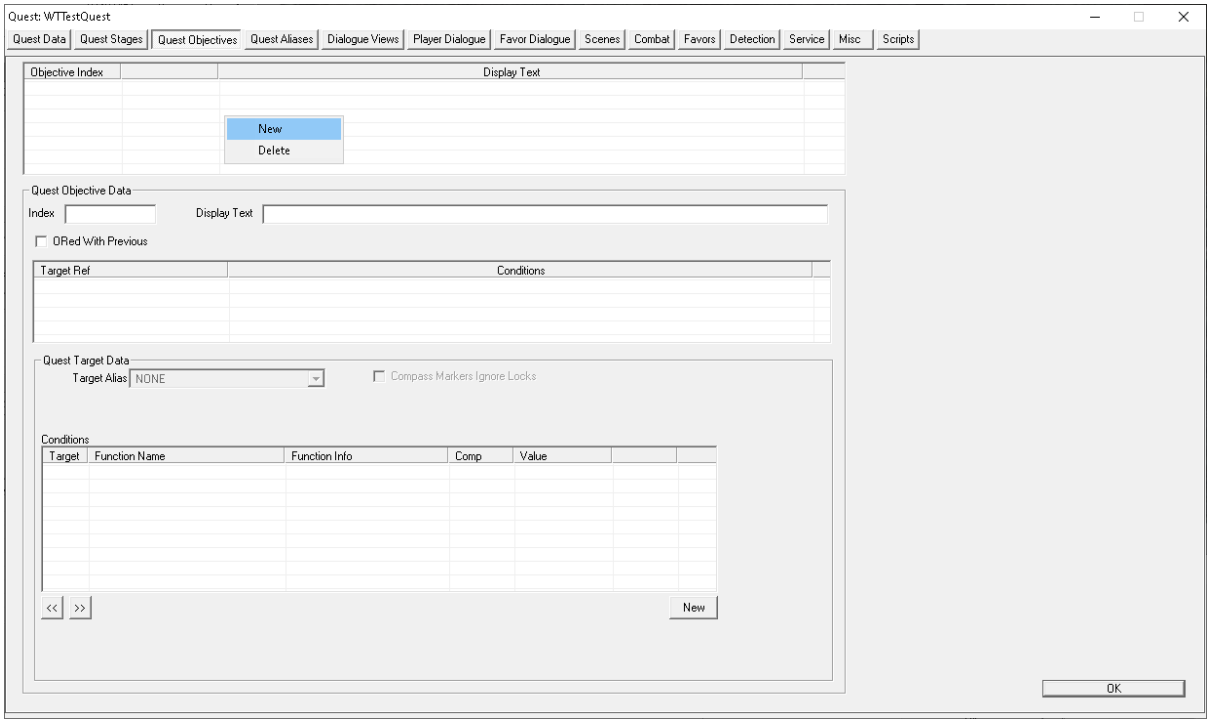


Figure 683 - Adding a new quest objective.

Set the Index to '10' and the Display Text to 'Retrieve Embersunder'.

This will be the objective for stage 10 of this quest. The display text is the objective that appears in the player's journal.

Right-click on the target list below and select New.

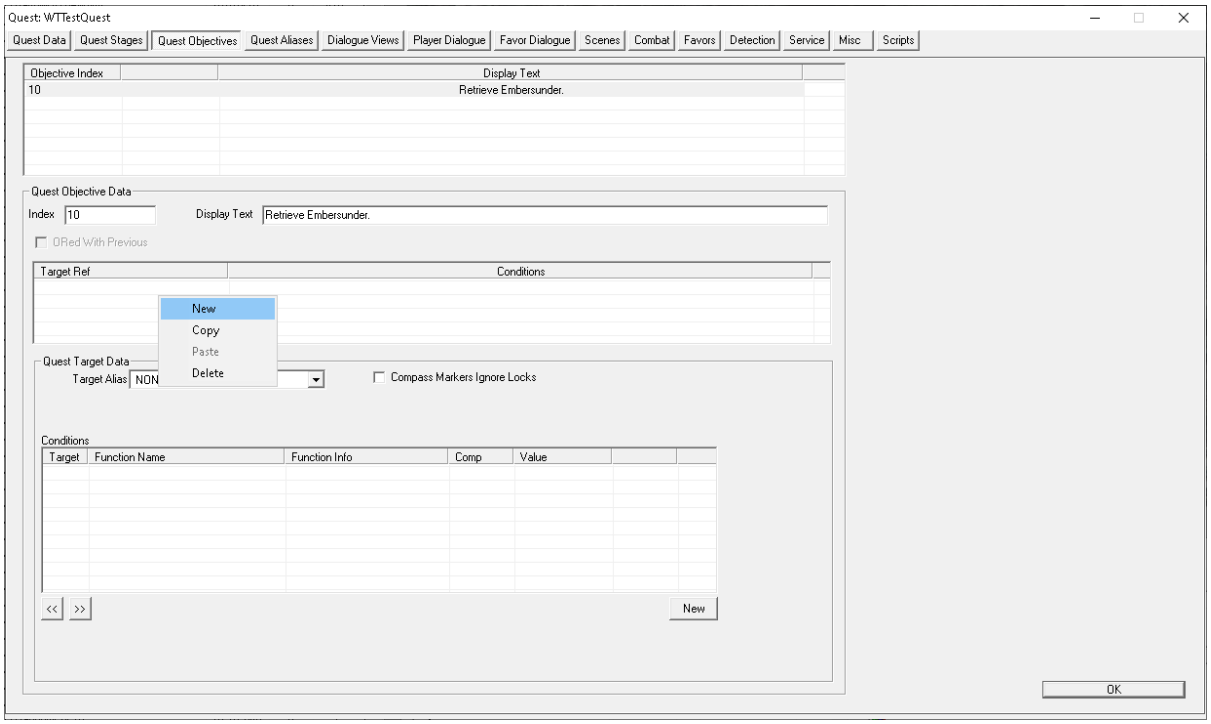


Figure 684 - Adding a quest objective.

Under Quest Target Data, set the Target Alias to Embersunder.

The reference selected here will have an objective pointer above it in-game at that stage of the quest.

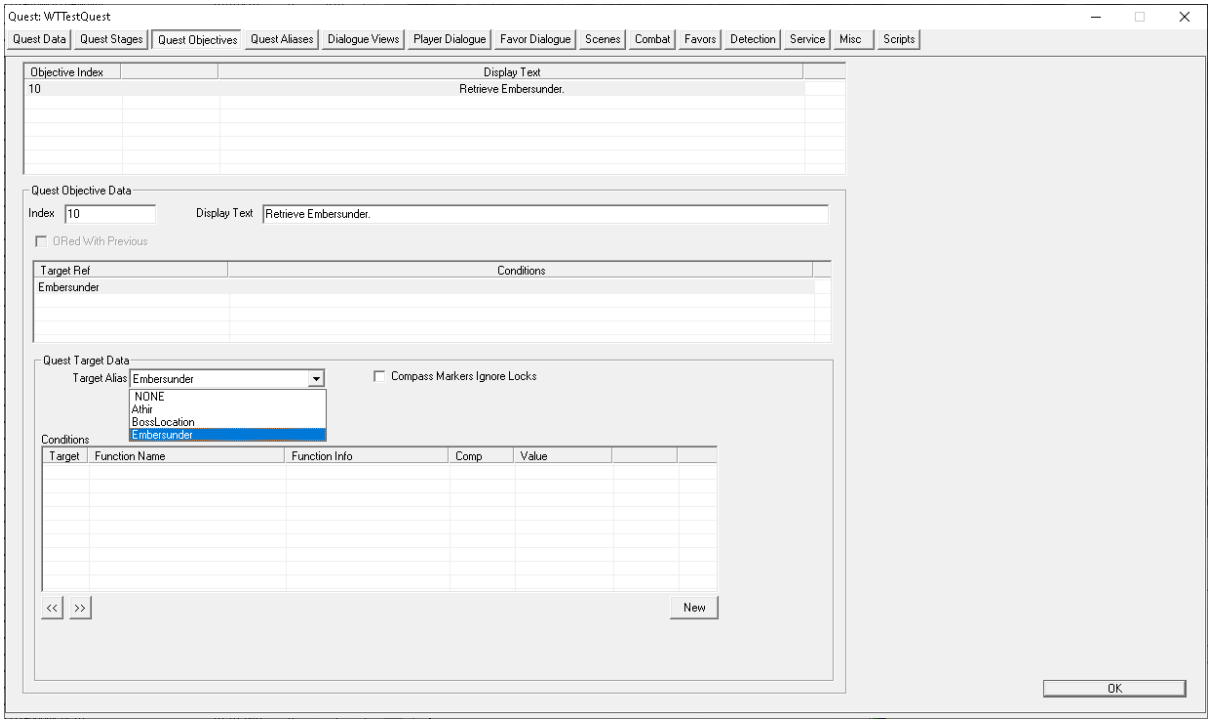


Figure 685 - Setting the quest target.

Create a second objective for stage 20. Set the Display Text to 'Return to Athir.'

Set the Target Ref to Athir.

This will place an objective over Athir once the player has obtained the sword.

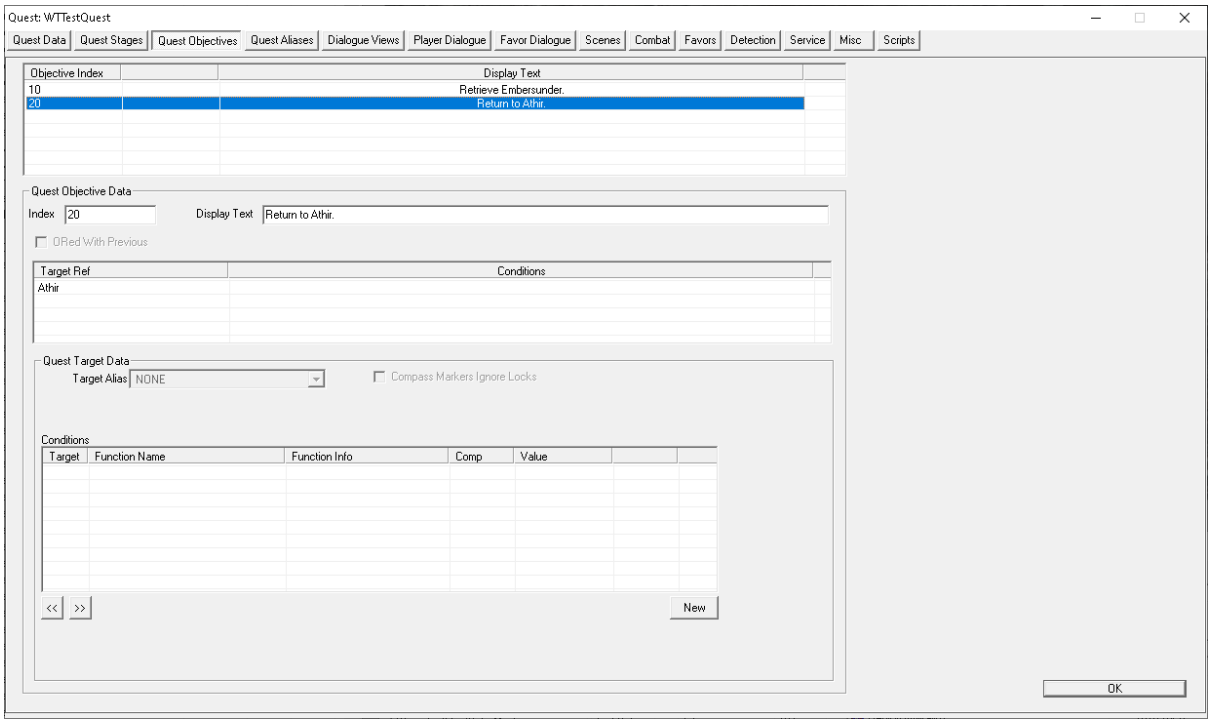


Figure 686 - Setting the second objective.

Go back to the Quest Stages tab.

Now we can enter a Log Entry for stage 10. I just went with:

Athir has asked me to find a sword called Embersunder. I should search the forsworn camp at Hag's End as Athir suggested.

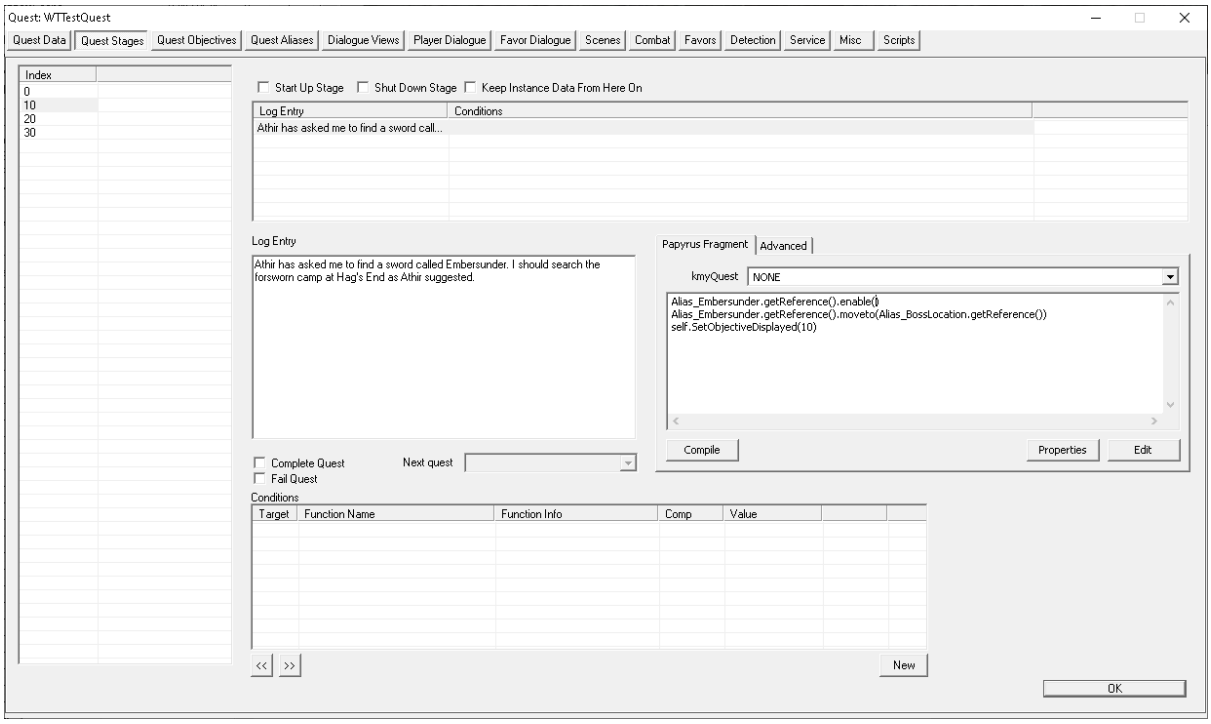


Figure 687 - Stage 10 of the sample fetch quest.

Under the Papyrus Fragment tab, enter in the following code:

```
Alias_Embersunder.getReference().enable()
Alias_Embersunder.getReference().moveto(Alias_BossLocation.getRefere
nce())
self.SetObjectiveDisplayed(10)
```

The sword is flagged as disabled by default so we need to enable it first.

It's then moved to the BossMarker reference alias and the stage 10 quest objective is displayed in the journal with a quest marker pointing to the sword.

For stage 20, I set the Log Entry to:

I have found the sword Embersunder for Athir. I should bring it to him when I get the chance.

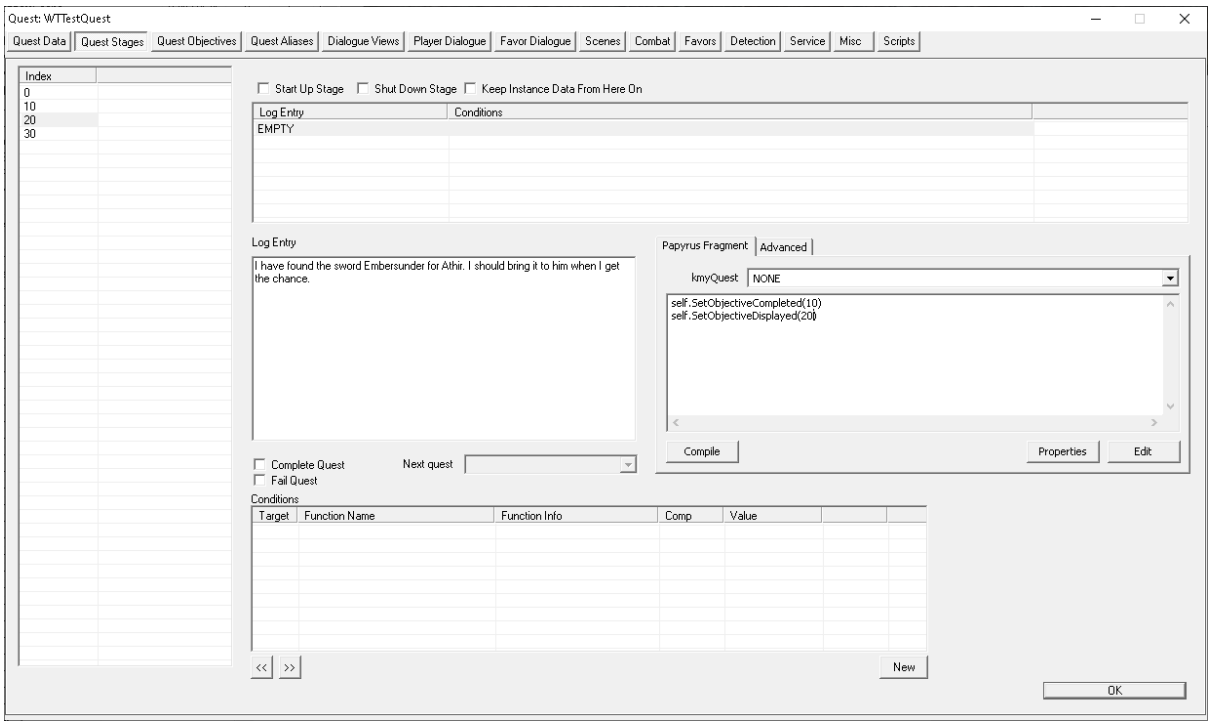


Figure 688 - Stage 20 of the sample fetch quest.

Under the Papyrus Fragment tab, enter in the following code:

```
self.SetObjectiveCompleted(10)
self.SetObjectiveDisplayed(20)
```

Once the player successfully retrieves the sword, we need to clear the objective to obtain the sword and replace it with the objective to return to our quest-giver Athir.

Finally, for stage 30, I set the Log Entry to:

I have given the sword Embersunder to Athir.

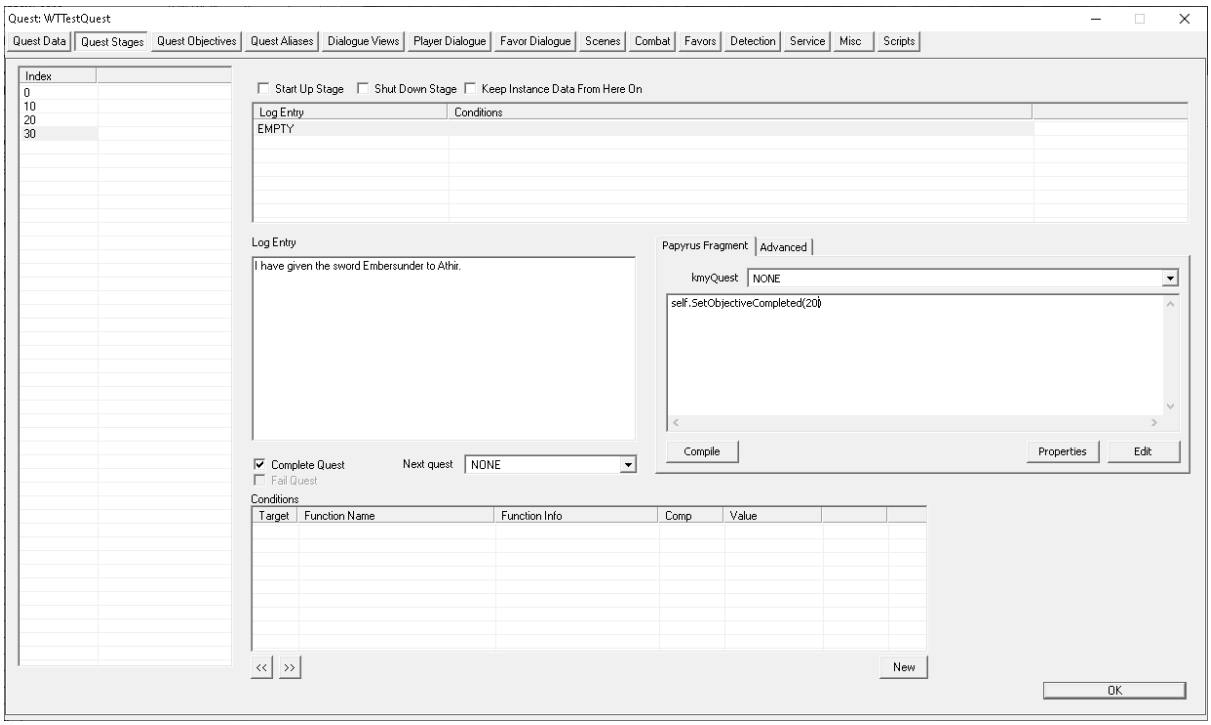


Figure 689 - Stage 30 of the sample fetch quest.

Under the Papyrus Fragment tab, enter in the following code:

```
self.SetObjectiveCompleted(20)
```

Now that we've given the sword to Athir we can complete the last objective and shut down the quest.

You can use the `stop()` command to manually shut down the quest or tick 'Shut Down Stage' on stage 30.

Important: Quest dialogue will become unavailable once the quest shuts down.

Lastly, we need some way of detecting when the player has acquired the sword. To do this, double-click or right-click on the sword in the render view and select Edit.

Go to the Scripts tab and select Add.

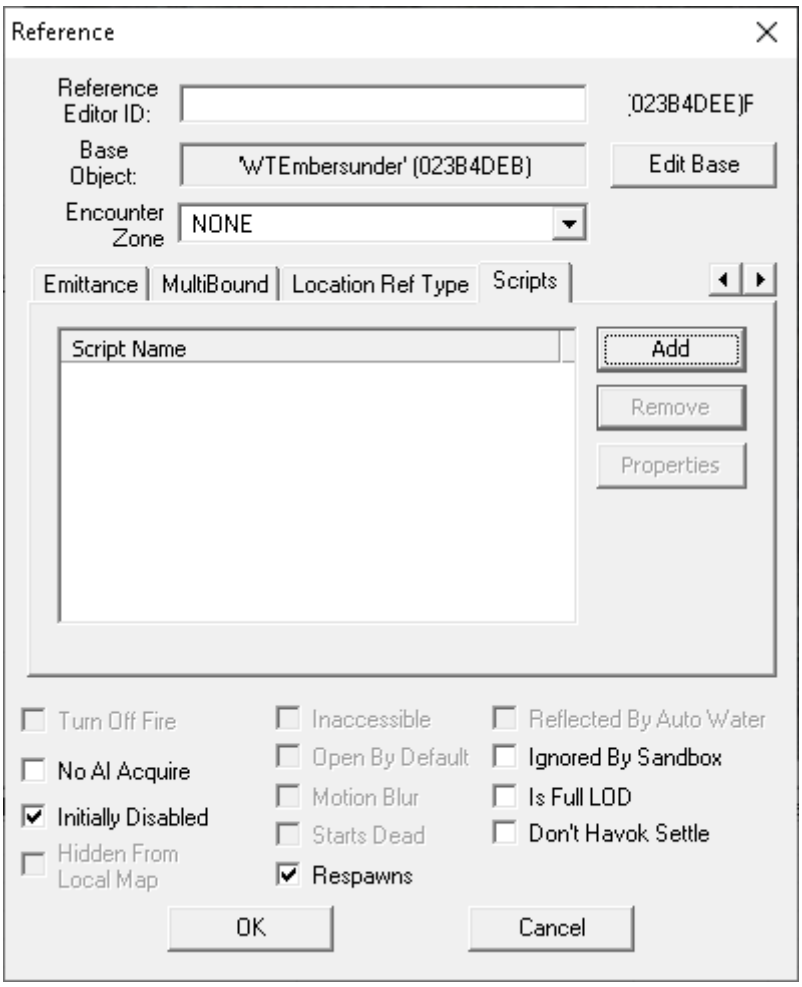


Figure 690 - Scripts tab.

Filter by ‘defaultsetstageonplayeracquireitem’.

Highlight that script in the script list and click OK.

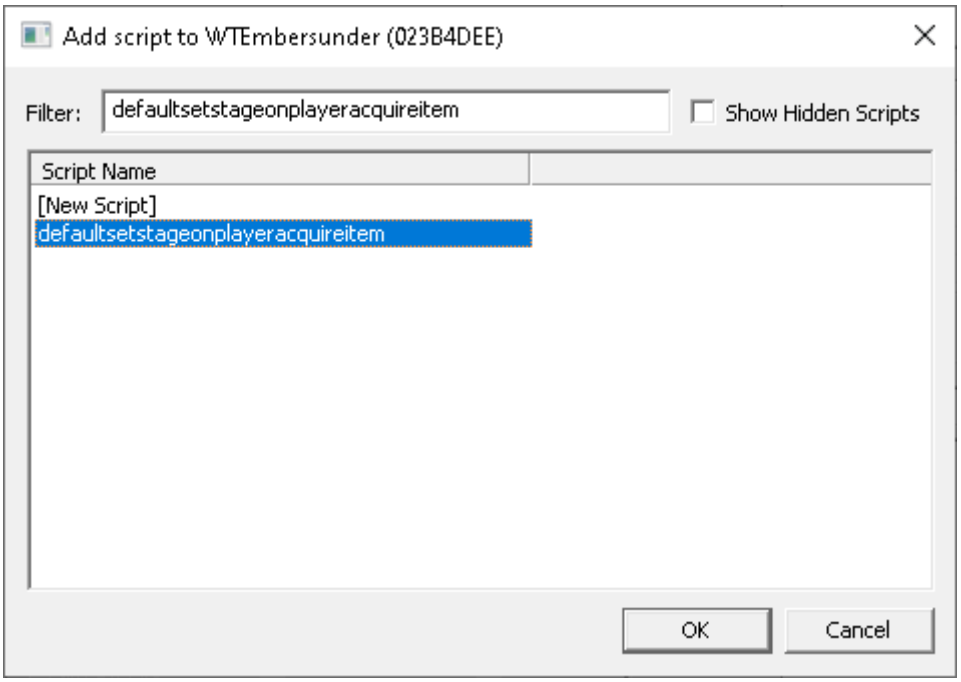


Figure 691 - Add script pop-up.

Highlight ‘defaultsetstageonplayeracquireitem’ in the Scripts tab then click Properties.

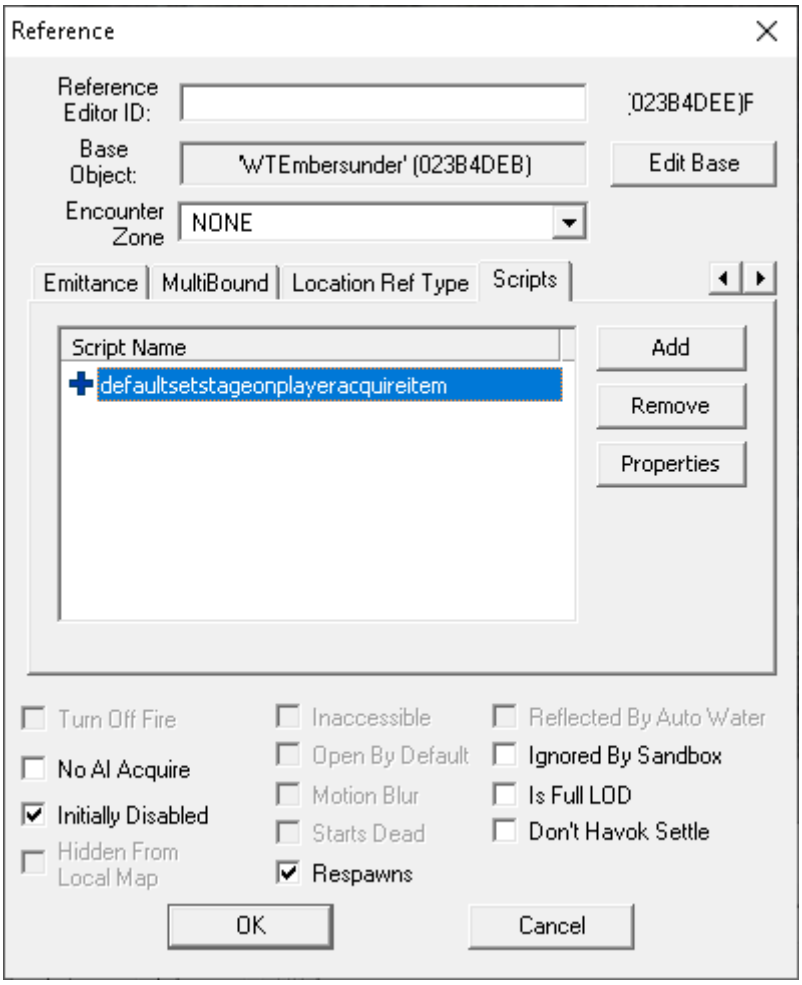


Figure 692 - Script added to the object reference.

Highlight myQST and set the quest value from the Pick Object drop-down. For this example, that's going to be WTTestQuest.

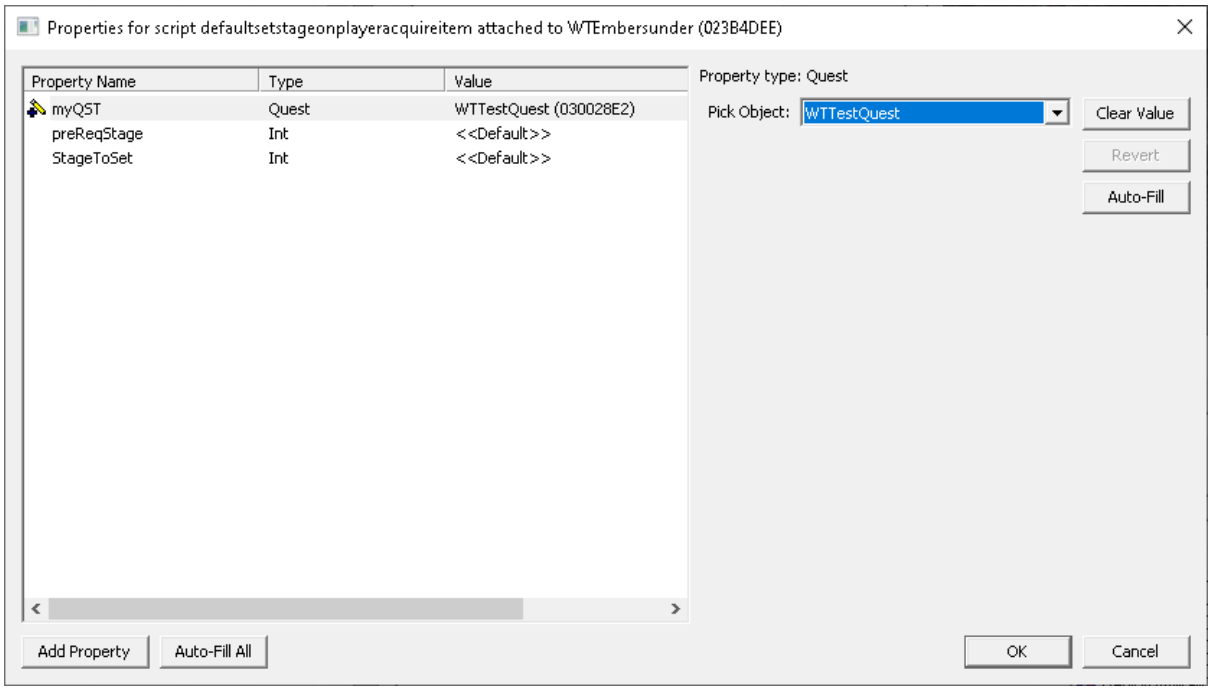


Figure 693 - Setting the myQST variable value.

Set preReqStage and click on Edit Value. Set preReqStage to 10.

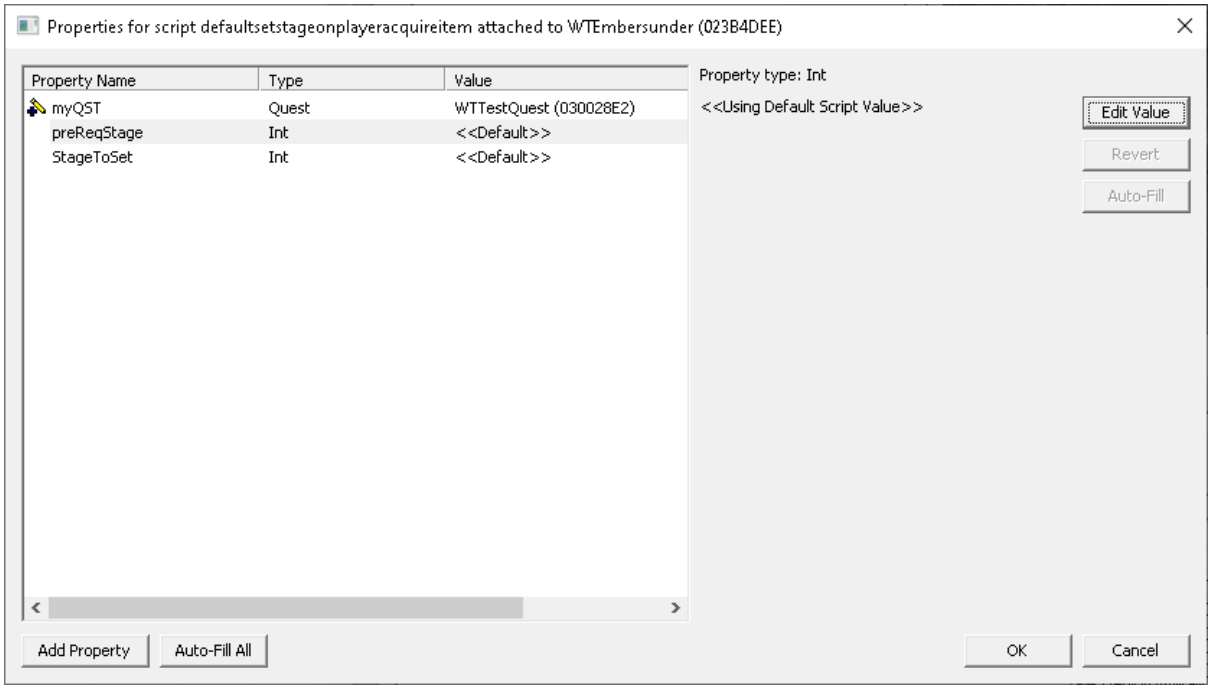


Figure 694 - Setting the prerequisite quest stage.

Now go to StageToSet and set it to 20.

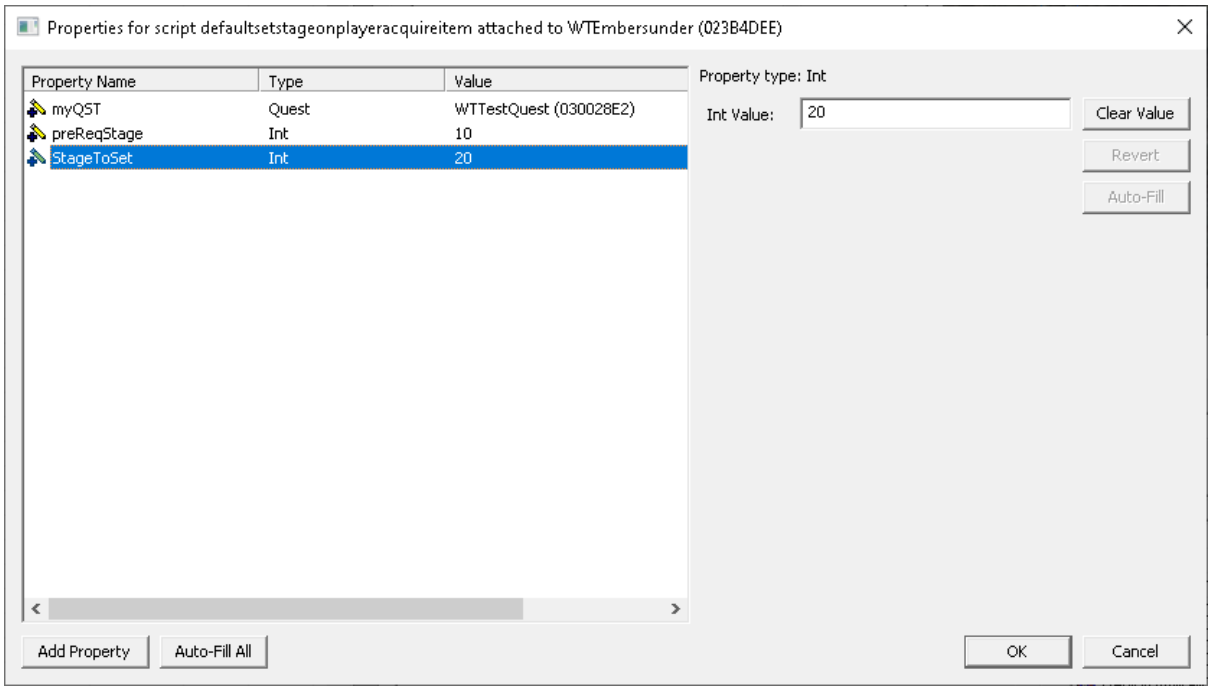


Figure 695 - Setting StageToSet to 20.

Click OK to close the script properties.

Click OK to close the Reference properties.

When the player picks up this item it'll set the quest stage to 20. The papyrus on stage 20 changes the objective to 'Return to Athir' and the quest marker will be set to Athir's current position.

The papyrus for setting stage 10 and stage 30 of this sample quest will be added to the dialogue itself which I'll be covering in the next section.

ADDING DIALOGUE TO A QUEST

All dialogue in the game is attached to quests. Some quests, such as DialogueGeneric, serve the singular purpose of providing NPCs with their generic hello, goodbye, combat and trespass dialogue.

Note: For the sake of compatibility, I'd recommend adding new dialogue to your own quests where possible rather than modify existing quests.

In this section, I'll be continuing where I left off in the previous section and will show you how to add in dialogue for our sample fetch quest.

Back in our quest properties, go to the Dialogue Views tab. Right-click in the Dialog Views list and select New.

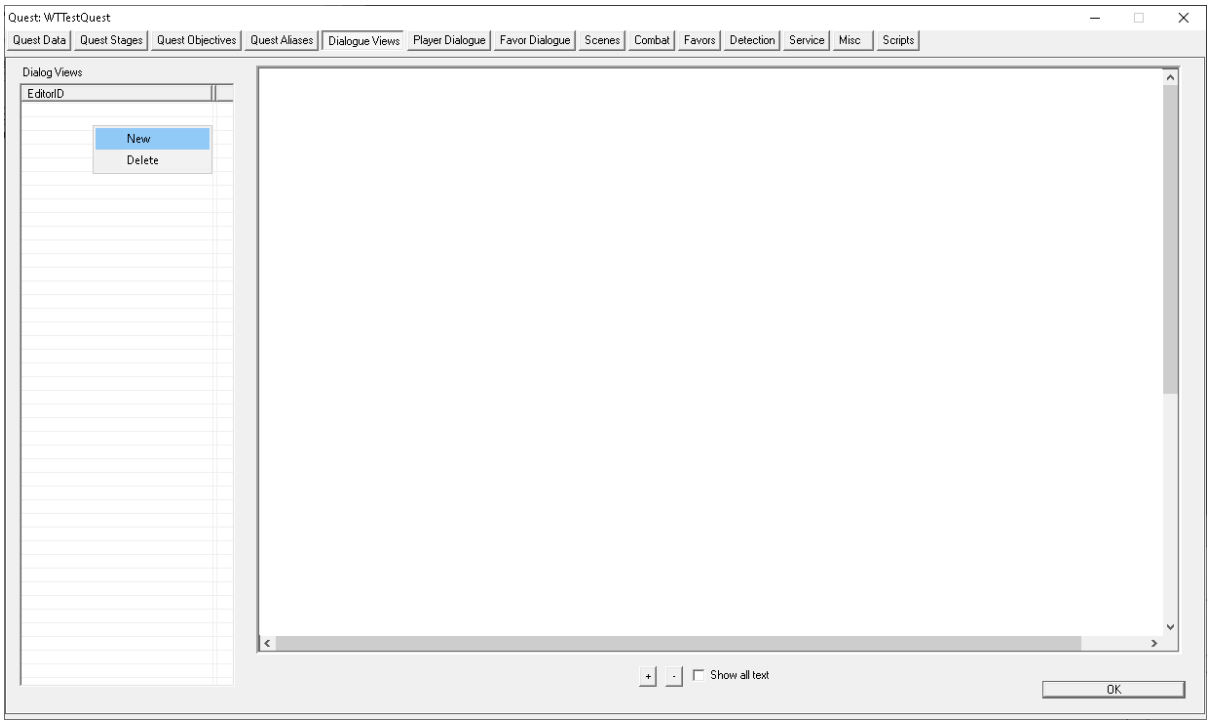


Figure 696 - Adding a new dialogue view.

Give the dialogue view a unique ID. For this example, I'm going to call it WTTestQuestAthirDialogView.



Figure 697 - Giving our dialogue view a unique ID.

Click OK.

Highlight the new dialogue view. Right-click in the area to the right and select 'Create branch'.

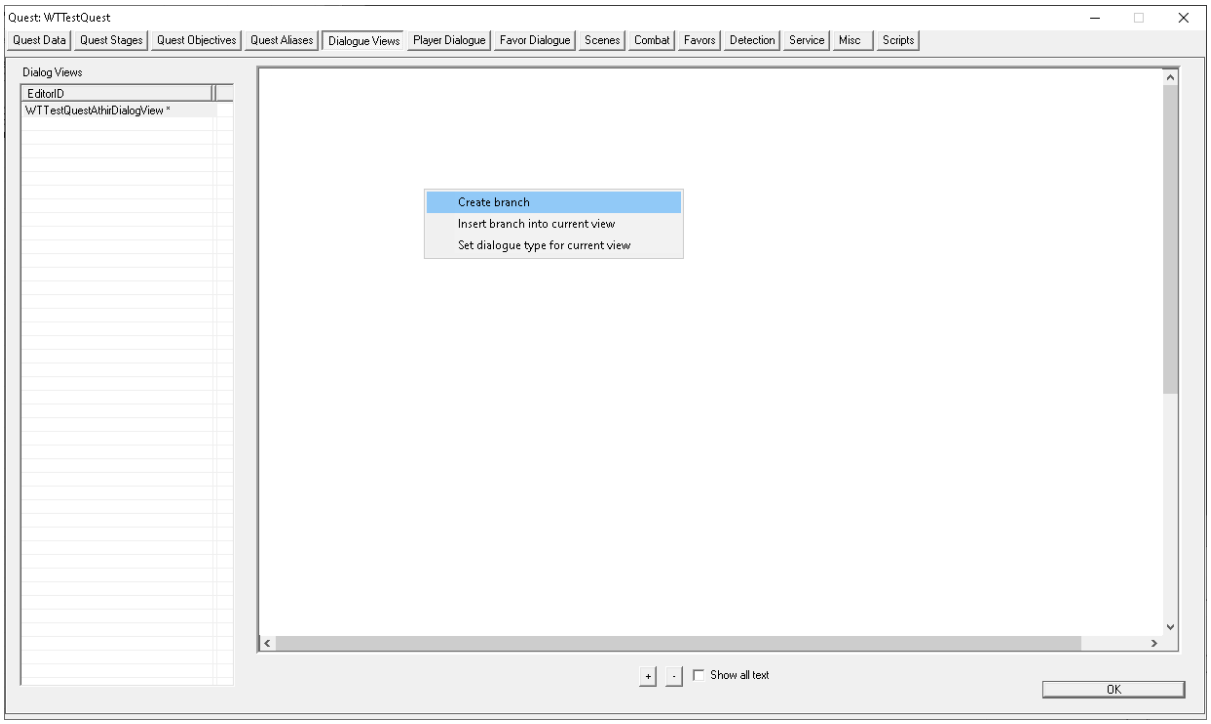


Figure 698 - Creating a new branch.

Enter in a branch ID and click OK. I'm going to call it WTTestQuestAthirBranch1.

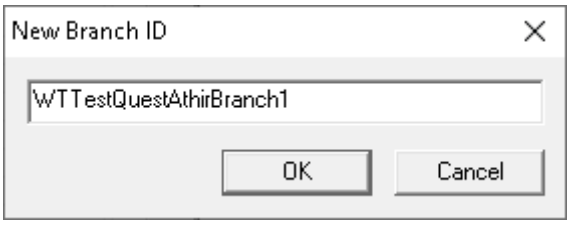


Figure 699 - Adding a branch ID.

You'll also be prompted to enter in the ID for the first topic. Generally this should include the branch ID. In the following example I went with WTTestQuestAthirBranch1Topic1.

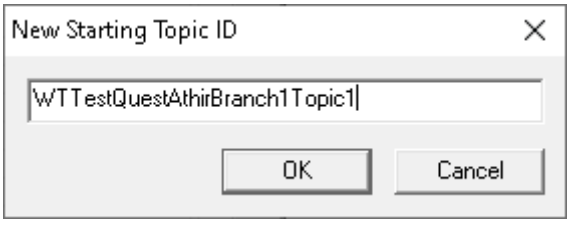


Figure 700 - Adding the first topic ID.

The new branch and topic should appear as per the screenshot below.

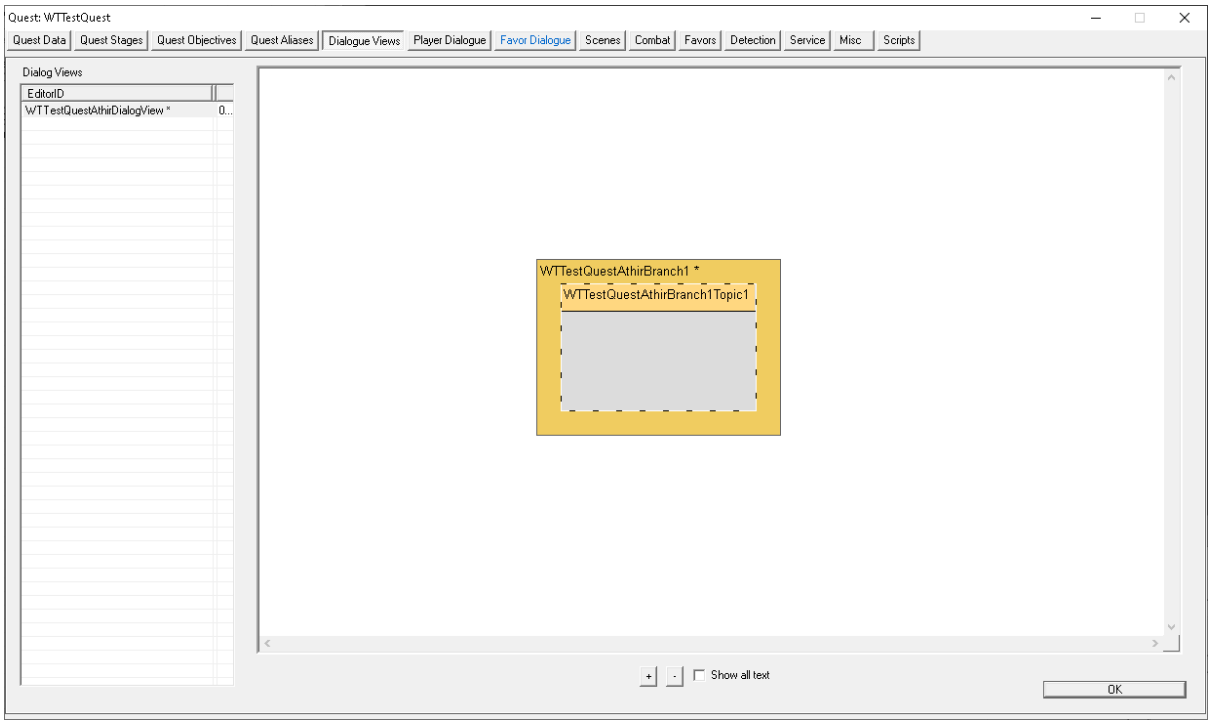


Figure 701 - New branch and topic.

Double-click on the topic within the branch to open it.

Enter in the Topic Text. The topic text is a player's dialogue option that is available when speaking to an NPC.

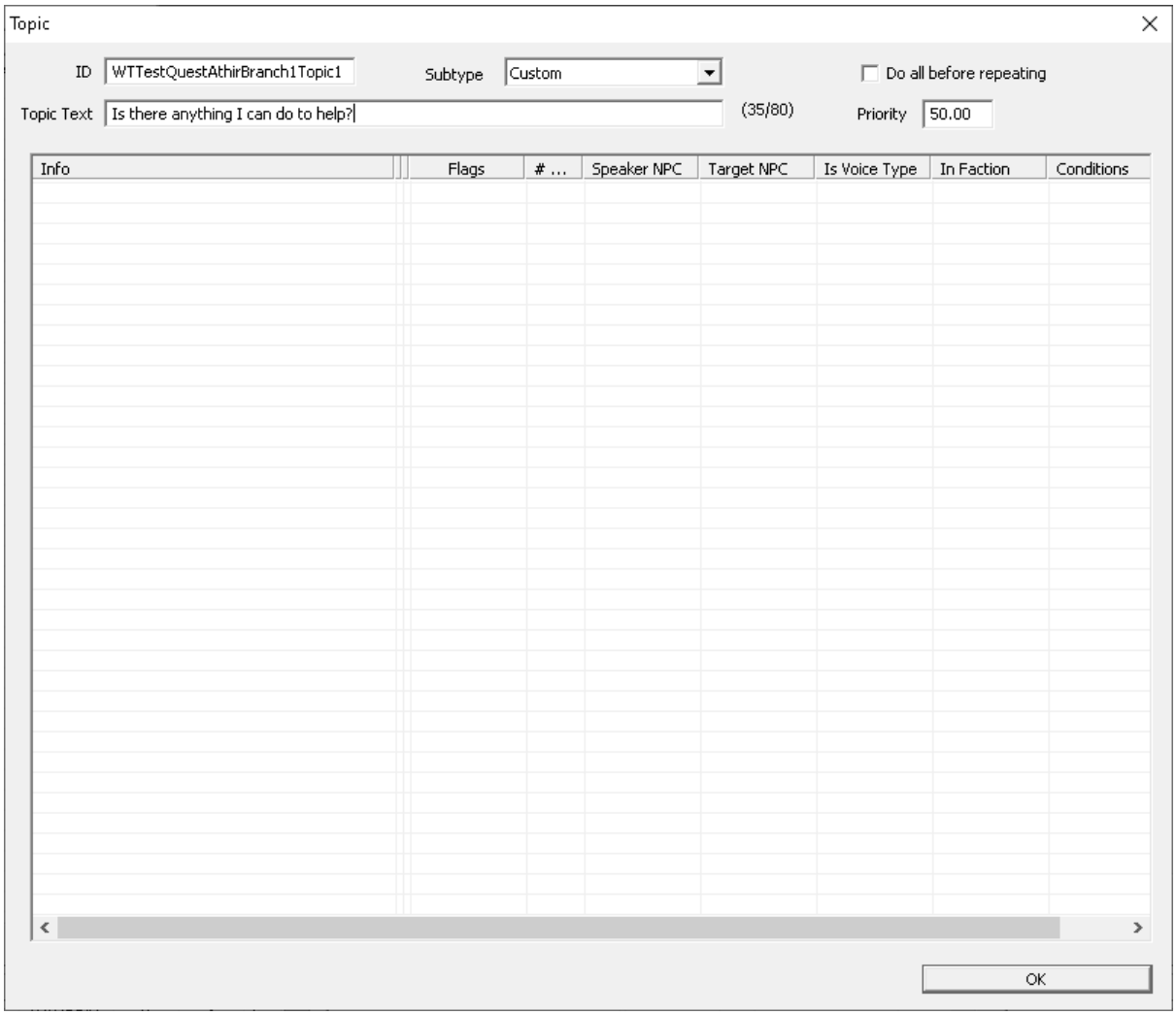


Figure 702 - Adding topic text.

Right-click in the space beneath the topic text and select New to add a new NPC response.

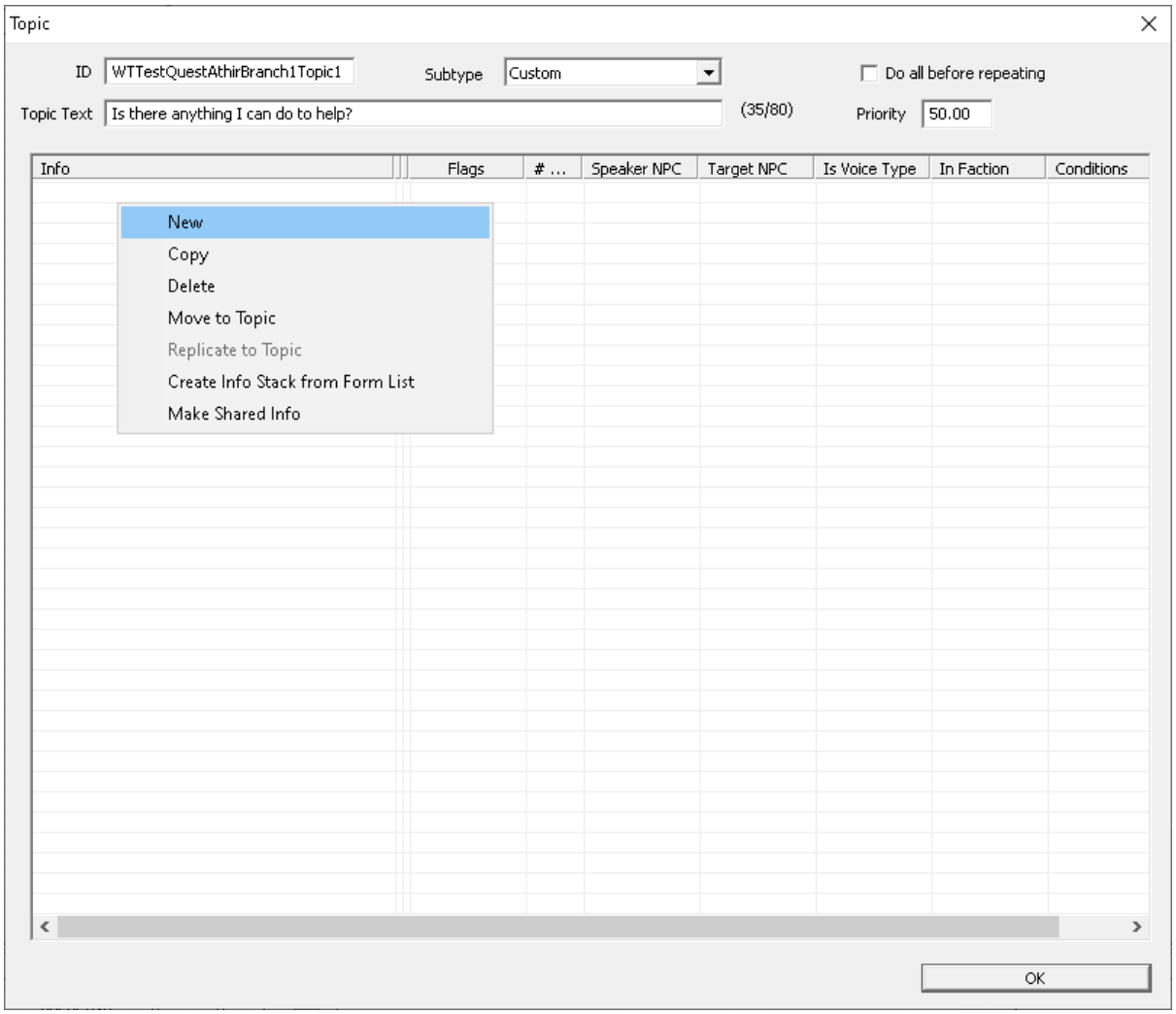


Figure 703 - Adding a new NPC response.

Enter in the NPC's response in the Response Text field. This will be used as the subtitle for the line of the dialogue that the NPC will say.

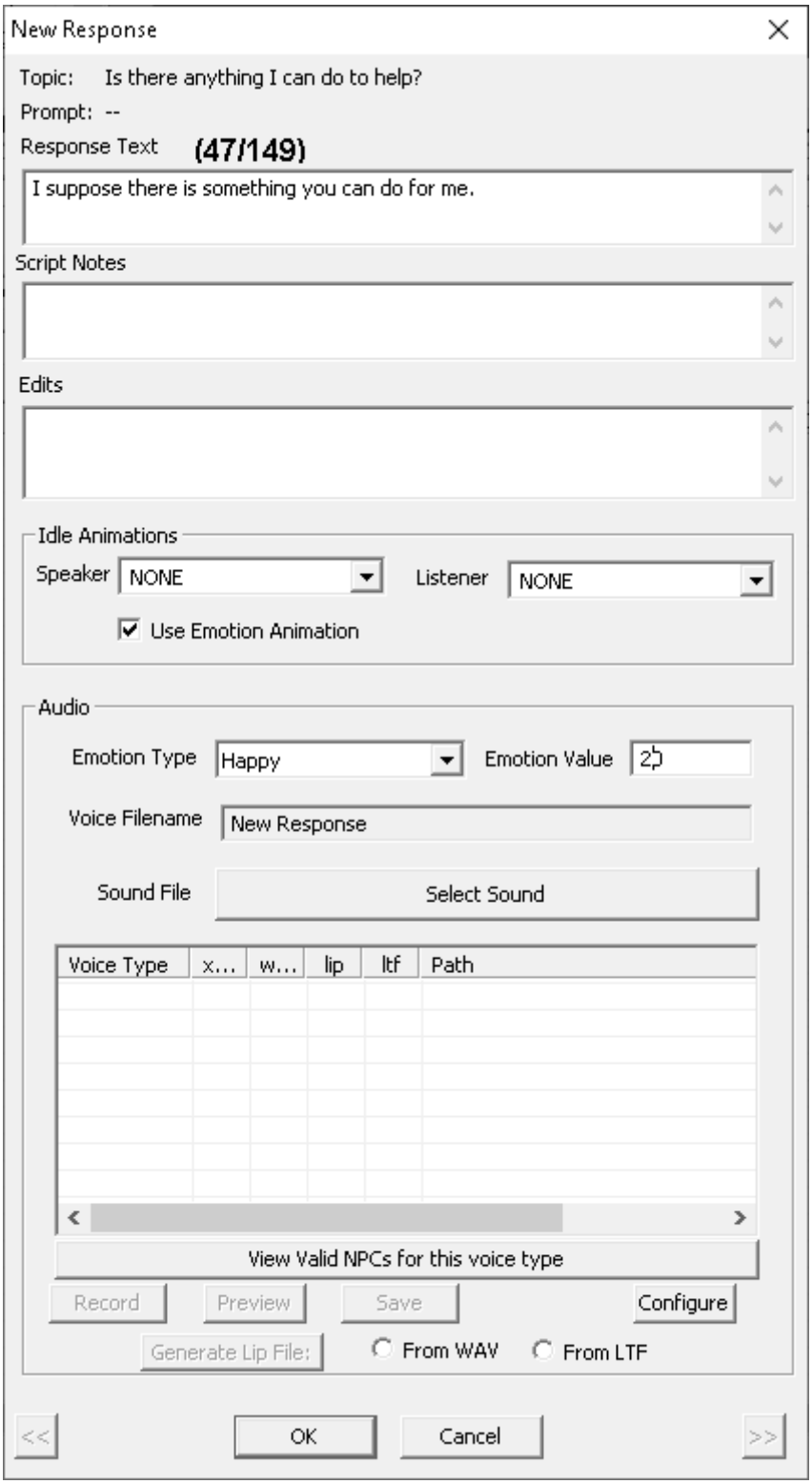


Figure 704 - Configuring the NPC's response.

If you want the NPC to play an animation while speaking, select the animation in the Speaker drop-down under Idle Animations.

The Emotion Type drop-down can be used to influence an NPC's facial animations while they're speaking a line of dialogue. In the previous example, I set the Emotion Type to Happy. You can control the amount of emotion they show in the Emotion Value field. The lower the number, the closer to neutral their expression will be.

Click OK to close the New Response screen.

Our Topic Info screen should now appear as per the screenshot below:

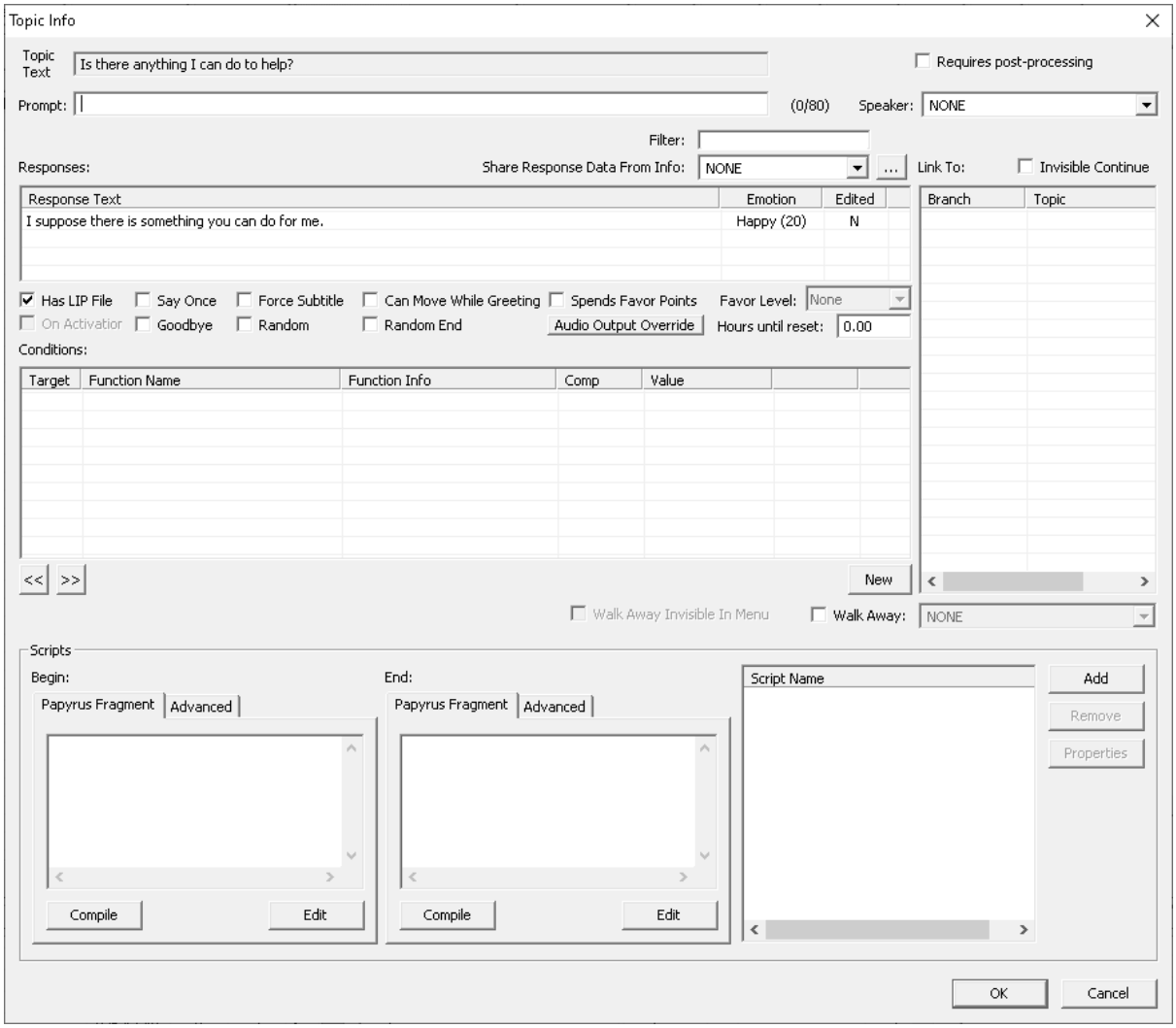


Figure 705 - Topic Info.

To add successive lines of dialogue, right click in the Responses list and select New.

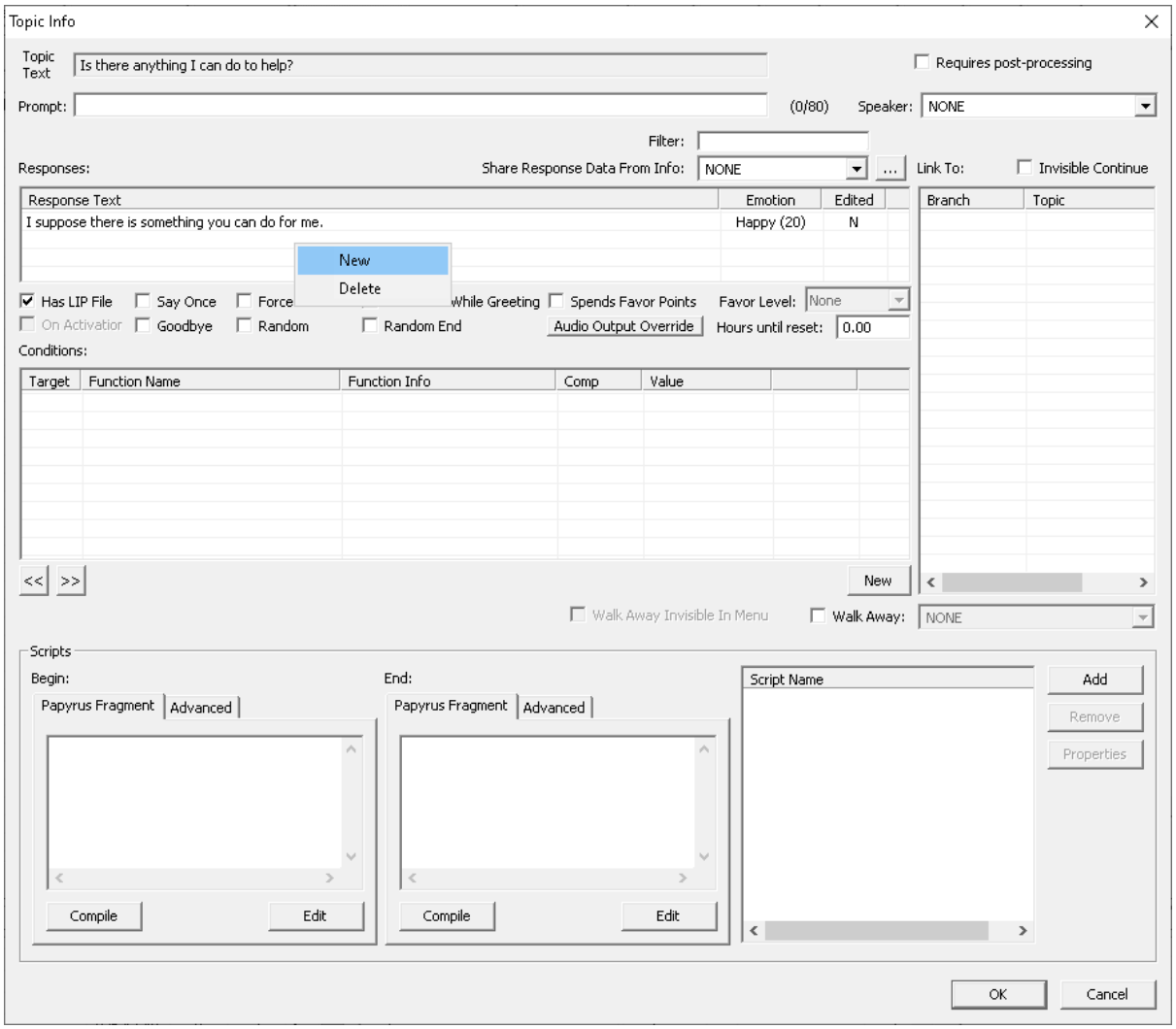


Figure 706 - Adding more lines of dialogue to this topic.

I went ahead and added a few more lines for this example.

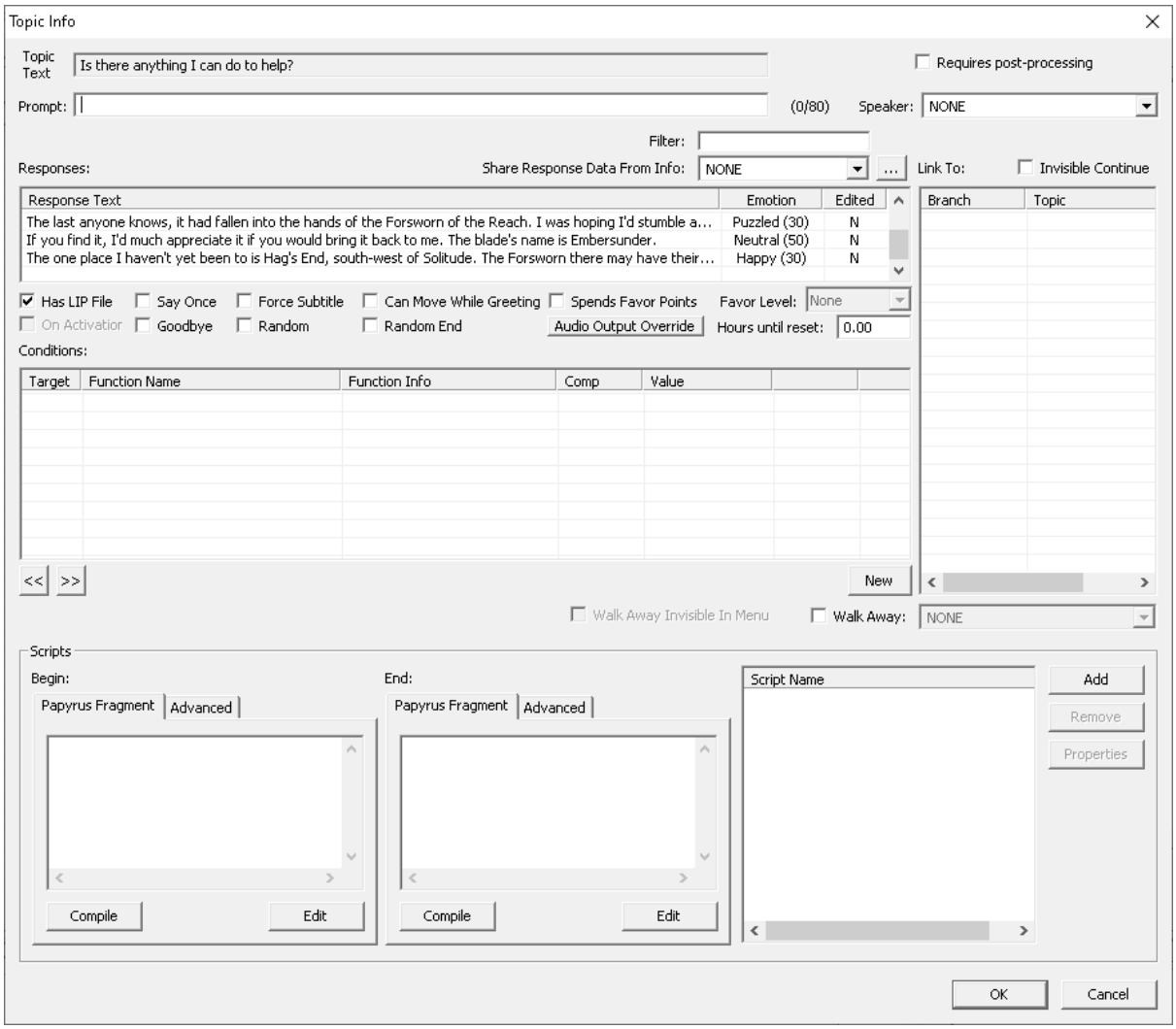


Figure 707 - Added multiple responses.

Next, we need to select the NPC that these lines will apply to.

Right-click in the Conditions list and select New.

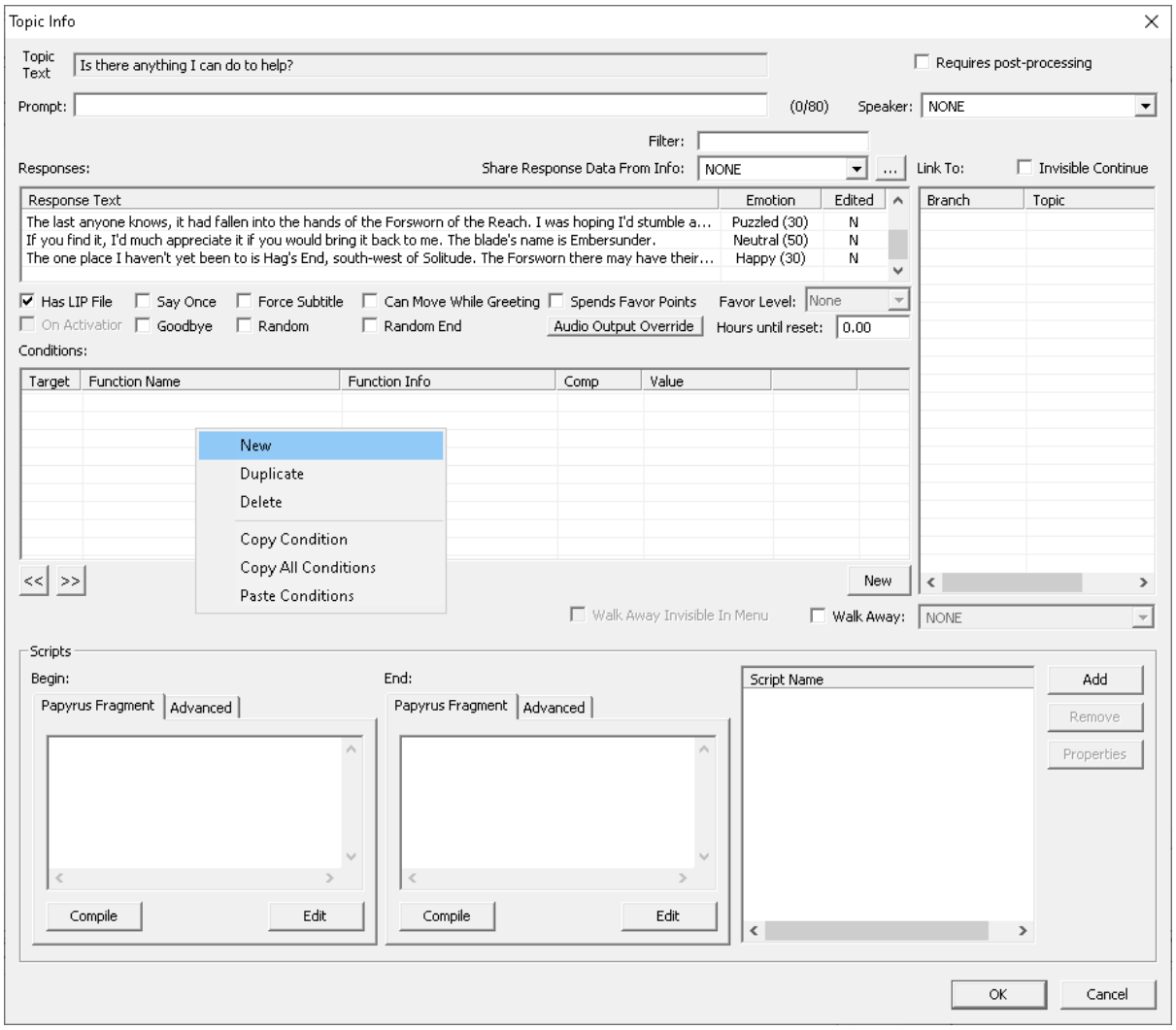


Figure 708 - Adding the first dialogue condition.

Ensure the Condition Function is set to GetIsID then click on the button next to it that currently says 'INVALID'.

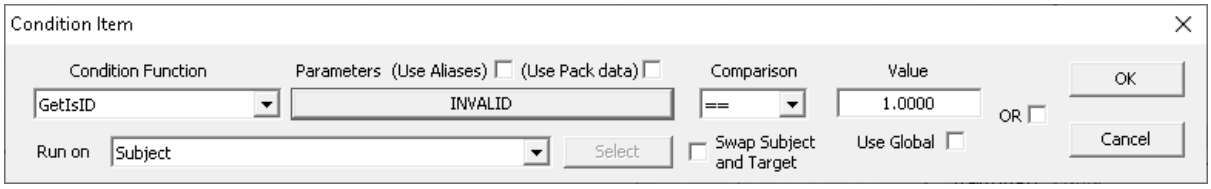


Figure 709 - Setting an NPC ID condition.

Select the form ID of the NPC. In this example, that's going to be our quest-giver Athir.

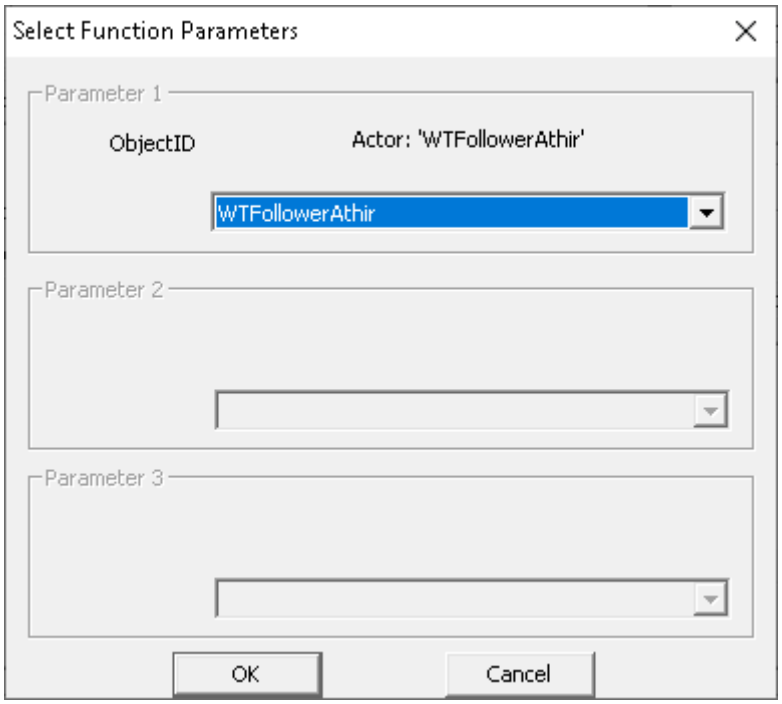


Figure 710 - Selecting the NPC ID.

Click OK to close Select Function Parameters.

Click OK to close Condition Item properties.

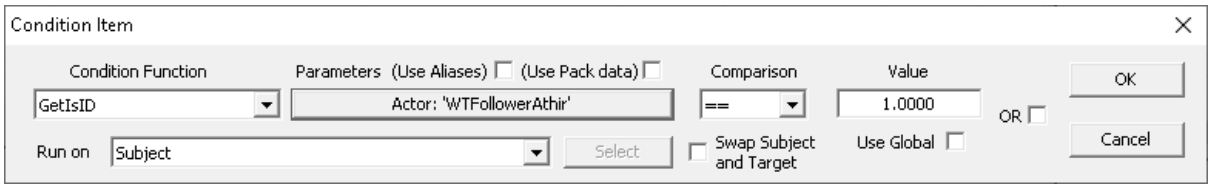


Figure 711 - NPC ID has been set.

Next, we need to add another condition to only show this topic if we haven't yet received this quest from our quest-giver.

Right-click again in the Conditions list and select New.

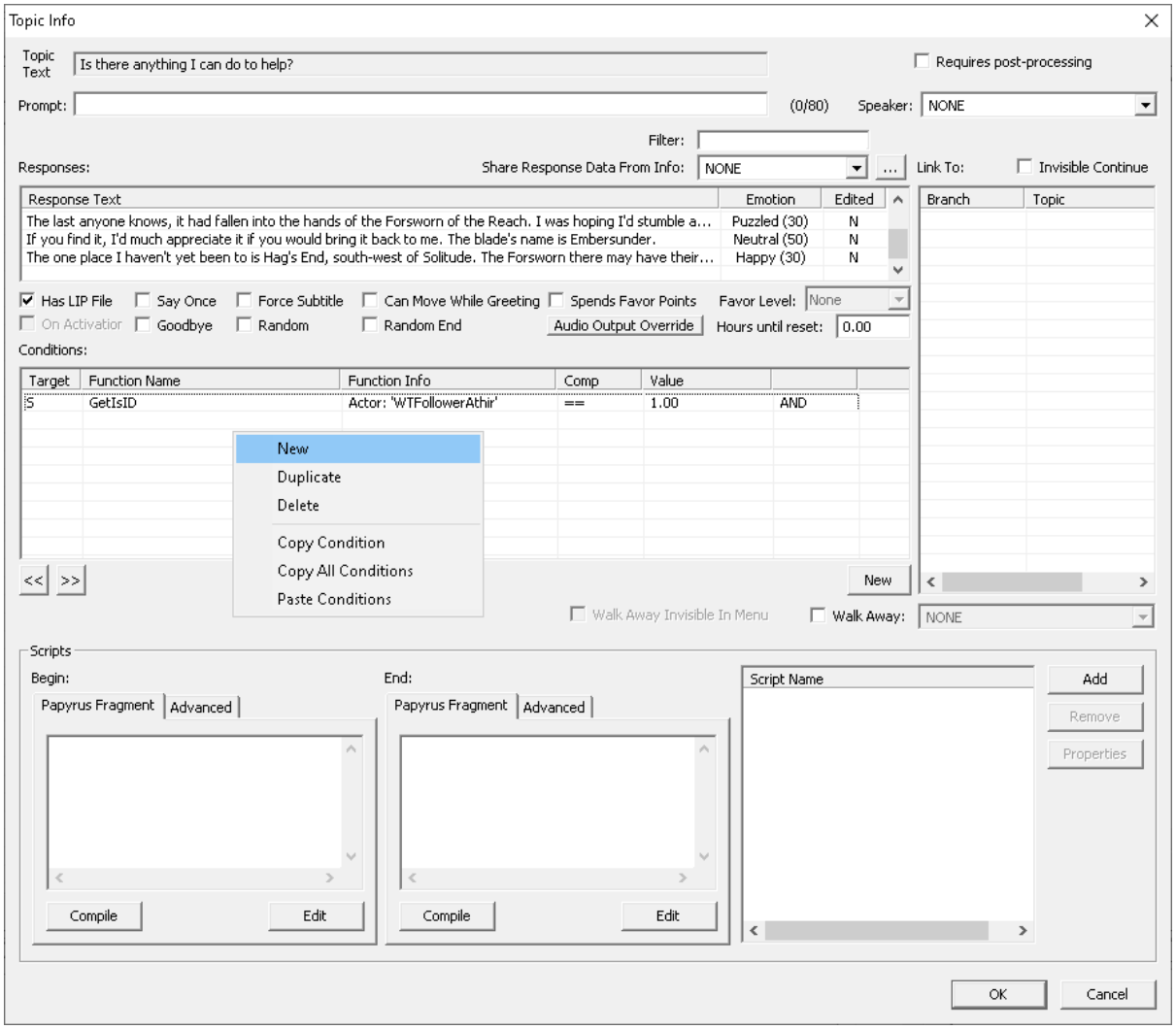


Figure 712 - Adding a GetStage condition.

Set the Condition Function to GetStage then click on the button next to it that currently says 'INVALID'.

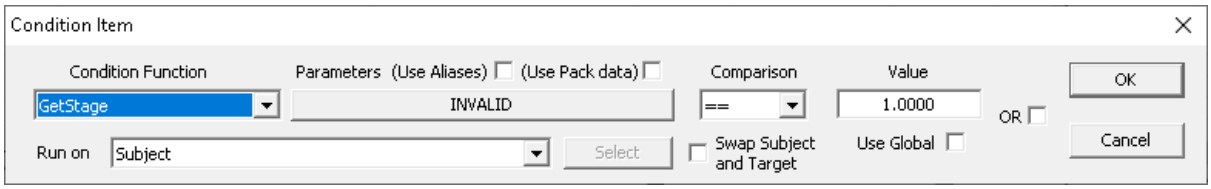


Figure 713 - Set the Condition Function to GetStage.

Select the Quest ID from the Quest drop-down, then click OK.

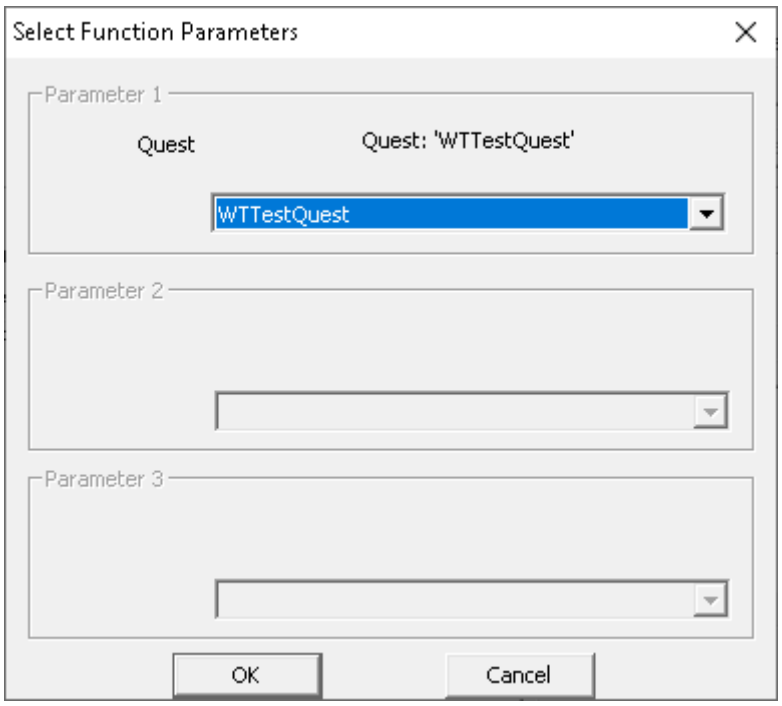


Figure 714 - Setting the Quest ID.

Set the Comparison operator to '==' and the value to 0. We only want this topic to appear if the current stage is 0.

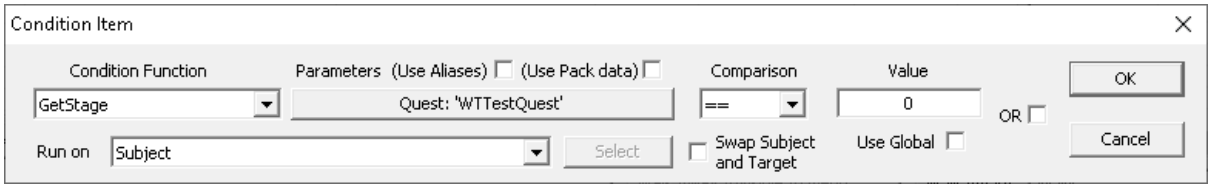


Figure 715 - Quest ID has been set.

Click OK to close out of Condition Item properties

In the Scripts section, enter in the following code into the End field and click Compile.

```
self.GetOwningQuest().SetStage(10)
```

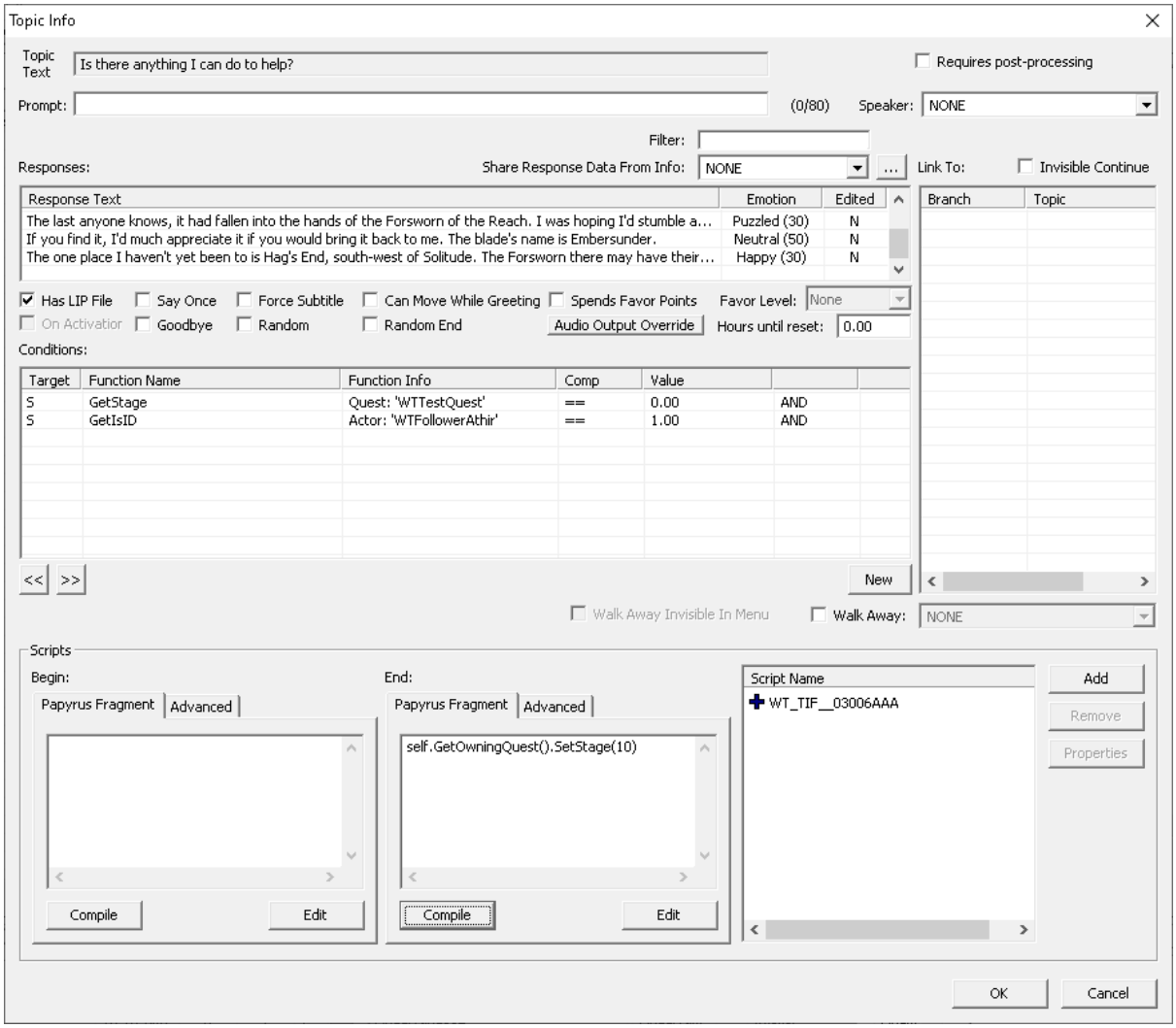


Figure 716 - Adding papyrus to execute when the NPC finishes speaking.

This code will be run when the NPC finishes speaking and will advance our sample fetch quest to stage 10.

When you press Compile, a new papyrus script fragment will be generated automatically. The papyrus fragment generated for this topic is WT_TIF_03006AAA.

Click OK to exit out of the Topic Info properties.

Our first branch and topic should now appear as per the following screenshot:

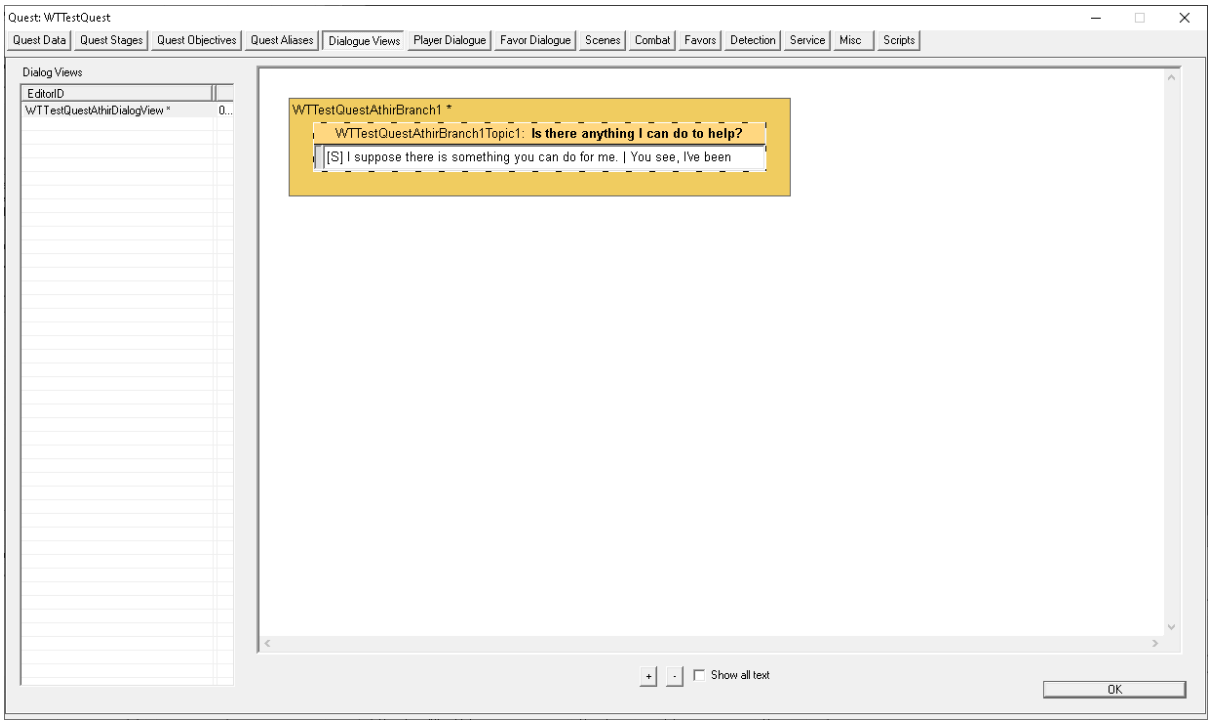


Figure 717 - First topic added.

So just to recap, our quest starts up behind the scenes at stage 0 when the mod is loaded.

After asking Athir 'Is there anything I can do to help', the quest will advance to stage 10, the sword will be moved into position, and the quest 'starts up' in the player's journal when the first objective becomes active.

When the player picks up the sword, the quest will be advanced to stage 20.

Now we need to add in a second dialogue branch to add the dialogue for returning the sword to our quest-giver.

Repeat the previous steps above to add a second branch and topic.

Note: When adding additional topics to an existing branch, the Creation Kit won't prompt us for a topic ID in a pop-up, so we need to manually change the topic ID in the ID field. For this example, I just called it WTTestQuestDialogueChainTopic2.

In my example, I set the second branch ID to WTTestQuestAthirBranch2 and set its topic ID to WTTestQuestAthirBranch2Topic1.

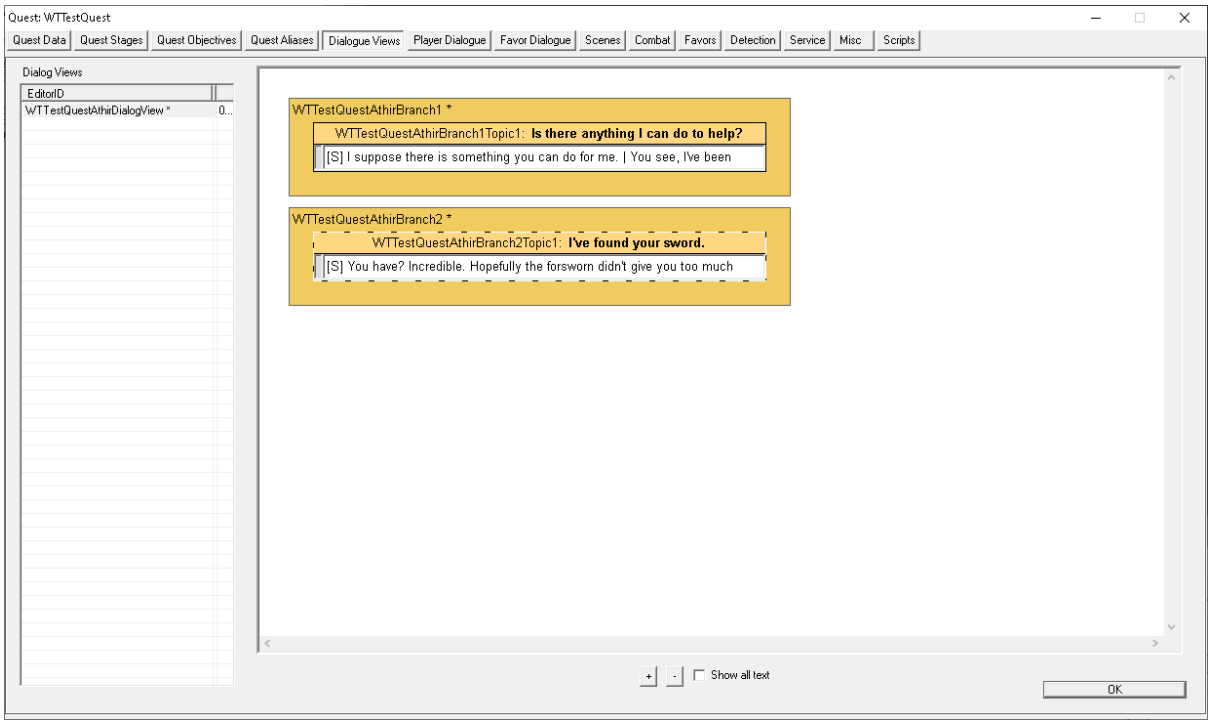


Figure 718 - Added the second branch and topic.

I set the Topic Text field for the second branch's topic to 'I've found your sword'.

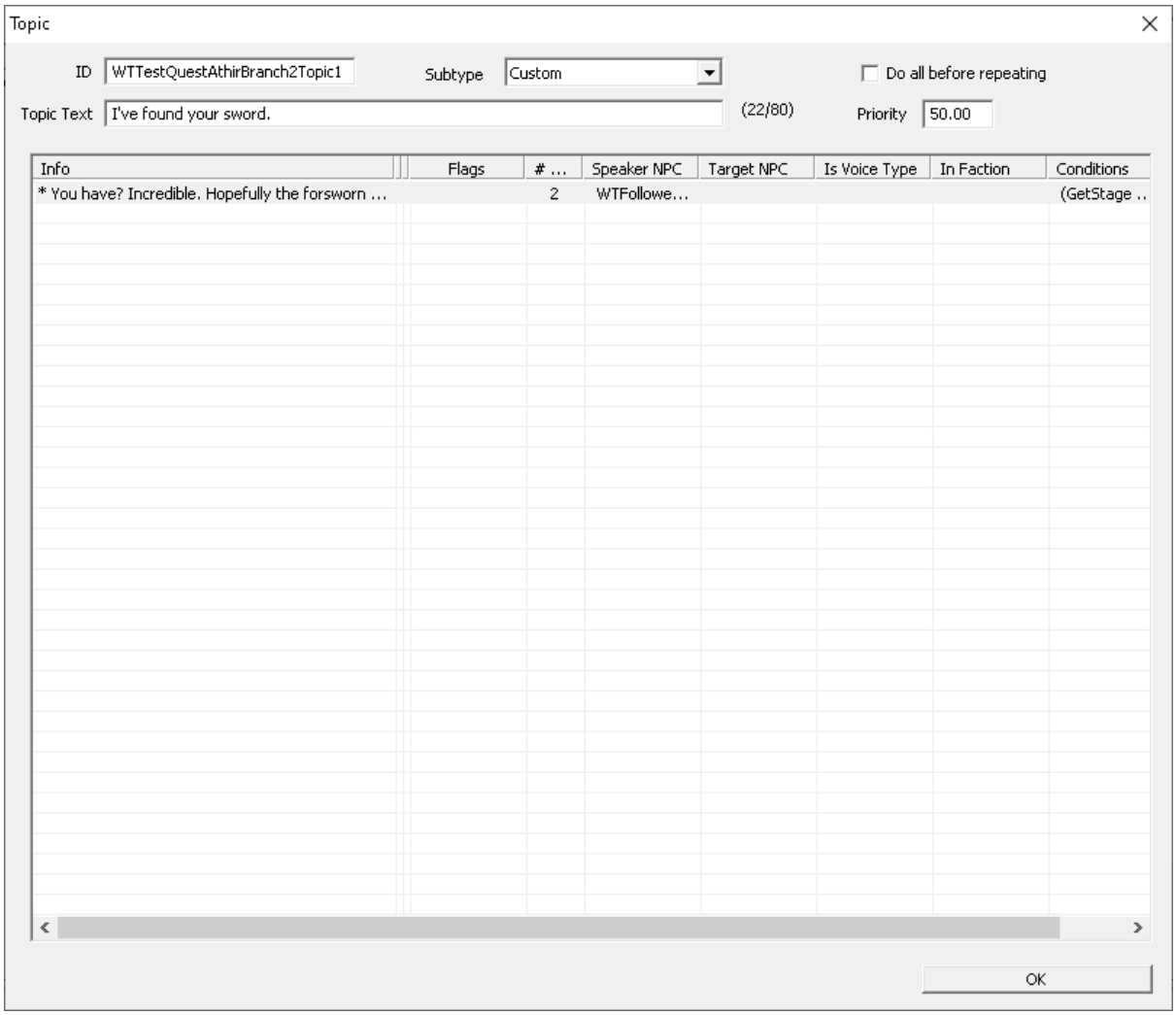


Figure 719 - The player's dialogue for the second branch.

The response for this topic was set up as per the following screenshot:

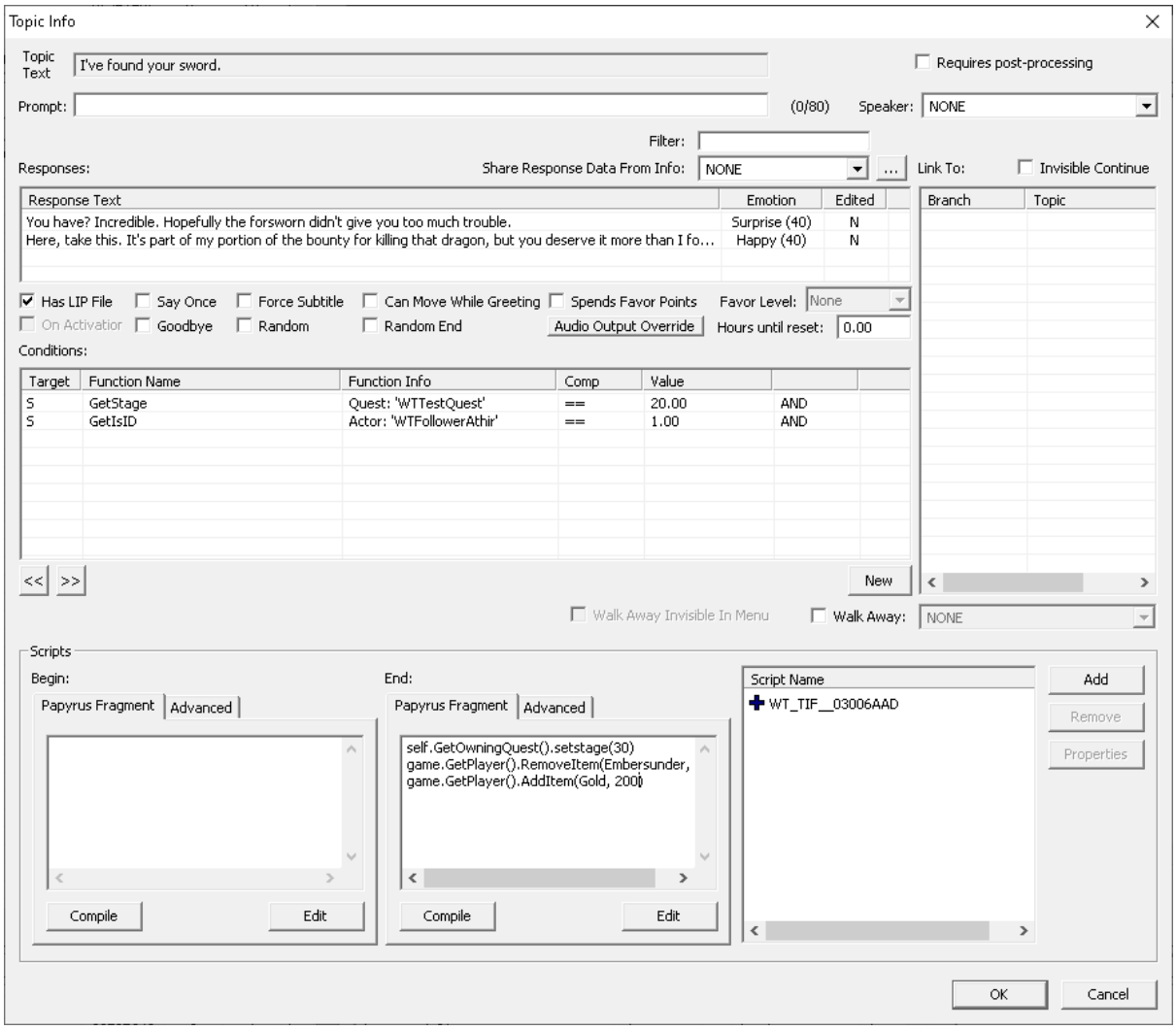


Figure 720 - NPC response for returning the sword.

In the Conditions list, this time I set the GetStage condition to '20' so this topic only appears once the player has the sword.

The papyrus added to the End field is as follows:

```
self.GetOwningQuest().setstage(30)
game.GetPlayer().RemoveItem(Embersunder, 1)
game.GetPlayer().AddItem(Gold, 200)
```

To break down this code, first we set our quest stage to 30 so we can end it. Then we remove the sword from the player's inventory. Lastly, we give the player a 200 gold reward for completing the quest.

Currently, if we try to Compile the End code, we'll get the following warning:

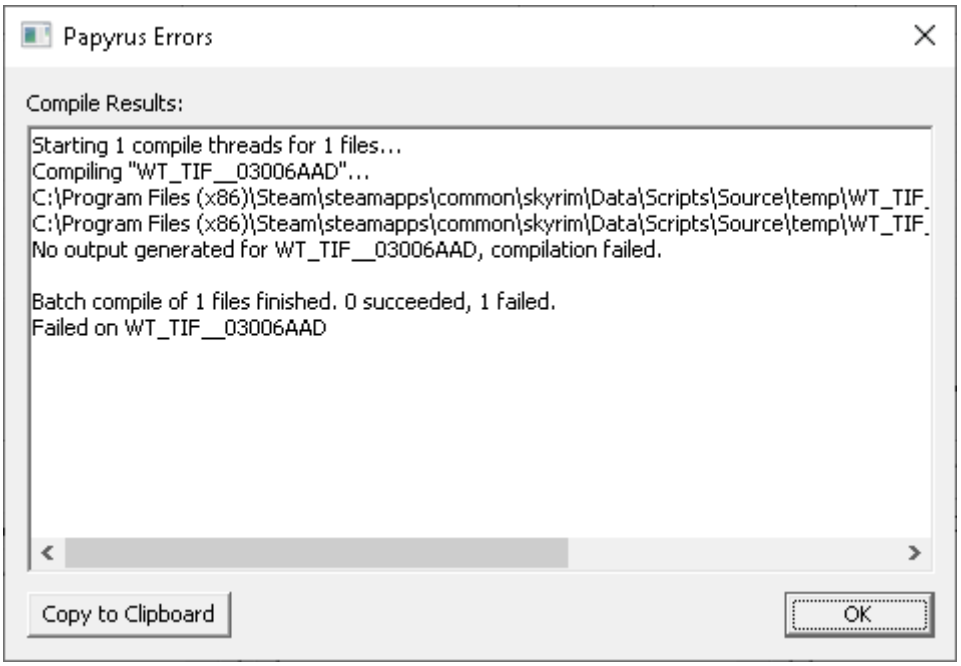


Figure 721 - Papyrus warning.

This is occurring because the 'Embersunder' and 'Gold' variables haven't yet been defined in the script fragment.

To rectify this situation, highlight the papyrus fragment and click on the Properties button.

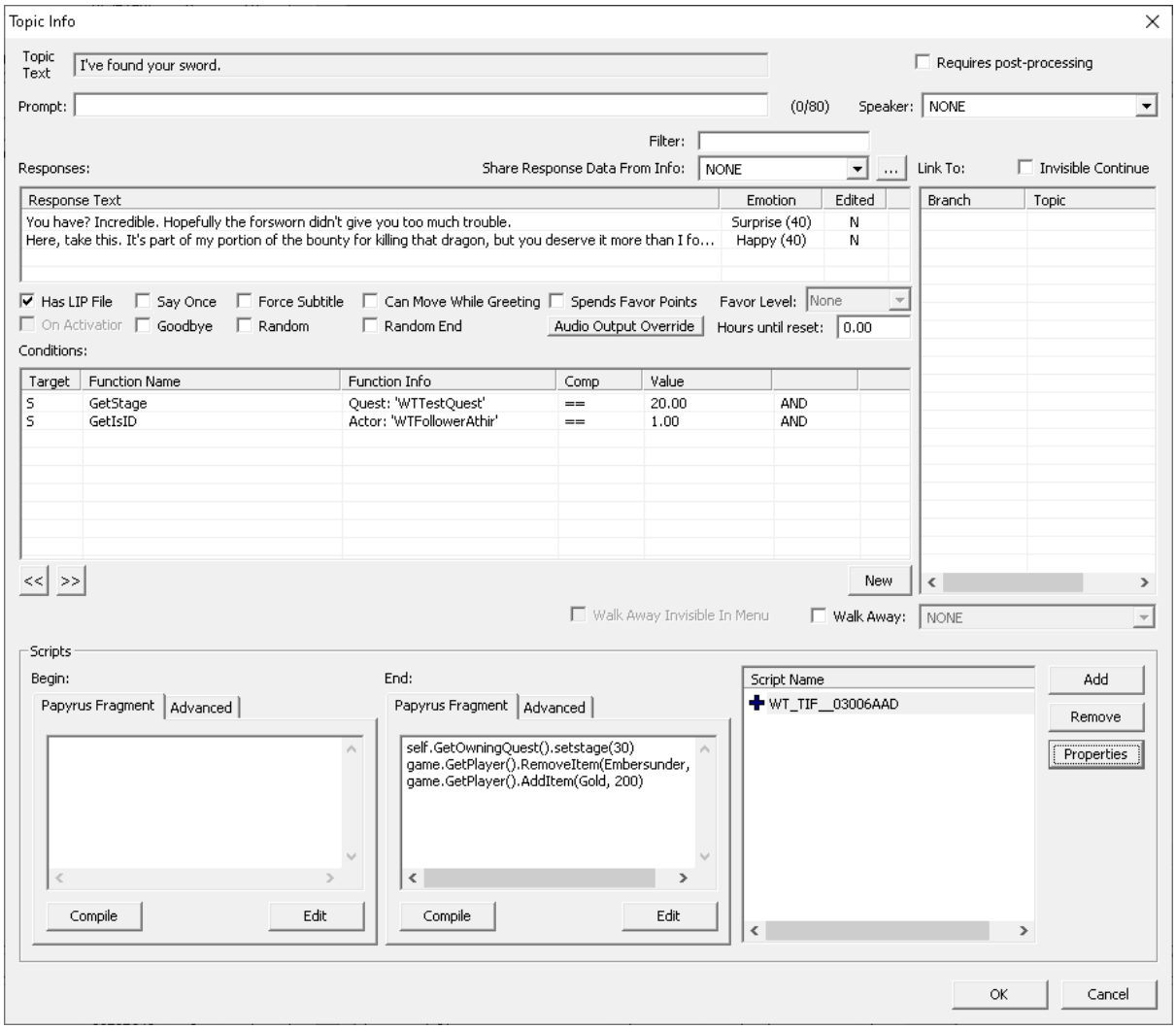


Figure 722 - Opening properties of the papyrus fragment.

If you see the following error, just click OK and ignore it.

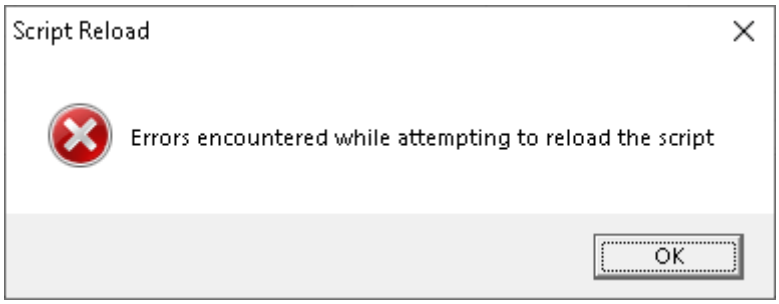


Figure 723 - Script reload error.

Click on the Add Property button.

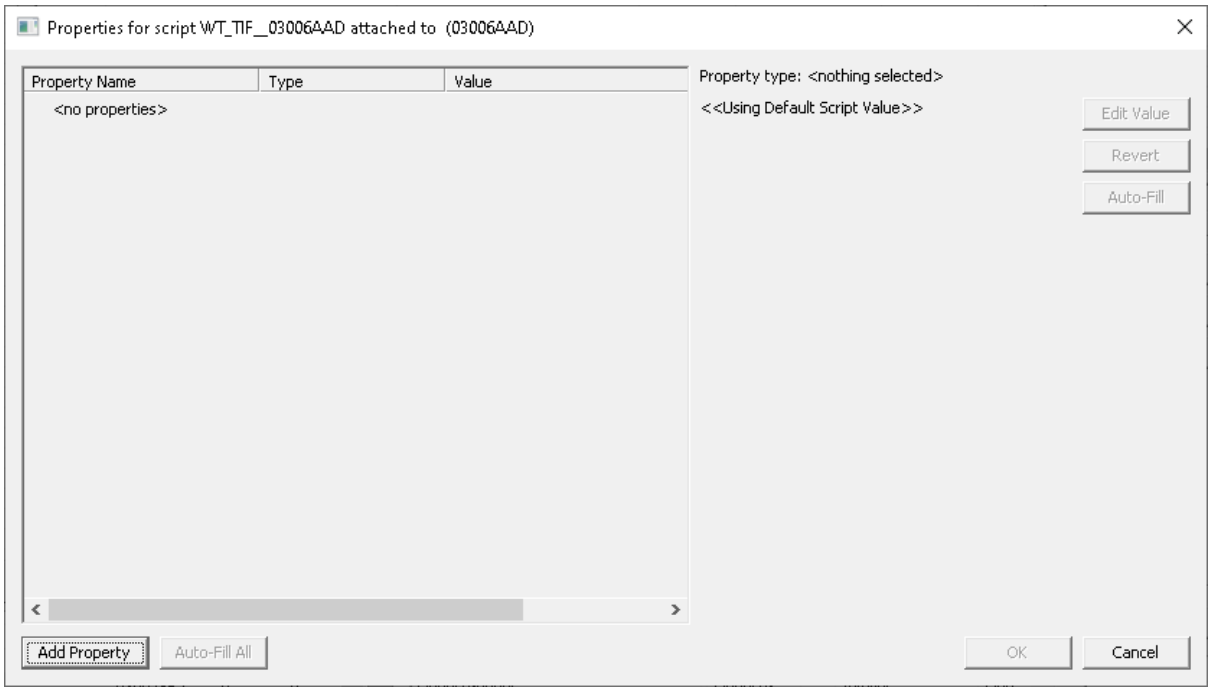


Figure 724 - Adding a new script property.

Set the Type drop-down to MiscObject and set the Name field to 'Gold' then click OK.

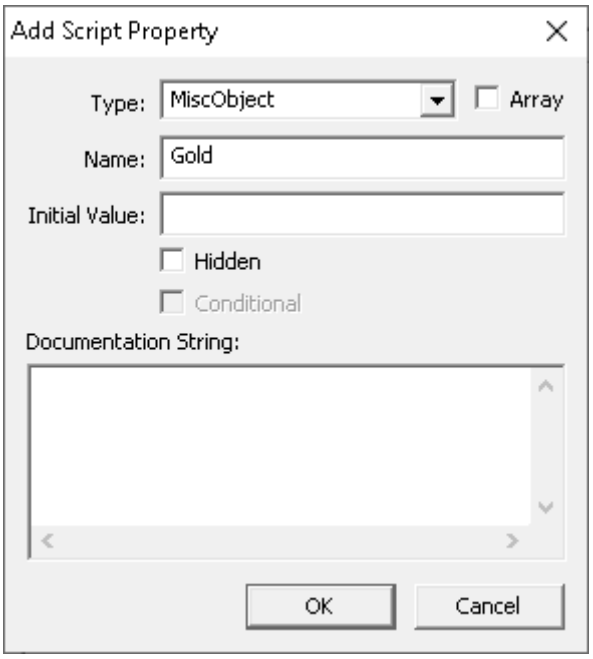


Figure 725 - Adding the Gold variable.

Click on Add Property again. This time set the Type drop-down to WEAPON and the Name field to Embersunder then click OK.

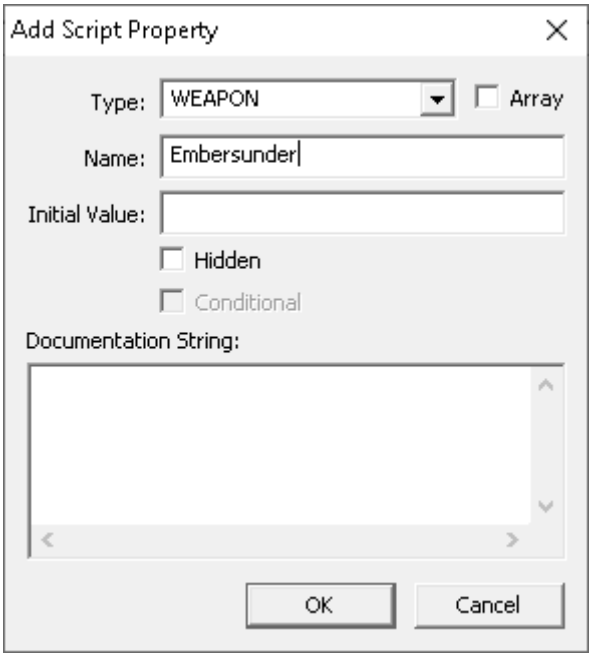


Figure 726 - Adding the Embersunder variable.

The two variables should now appear in the Properties list. Next, we need to set their values.

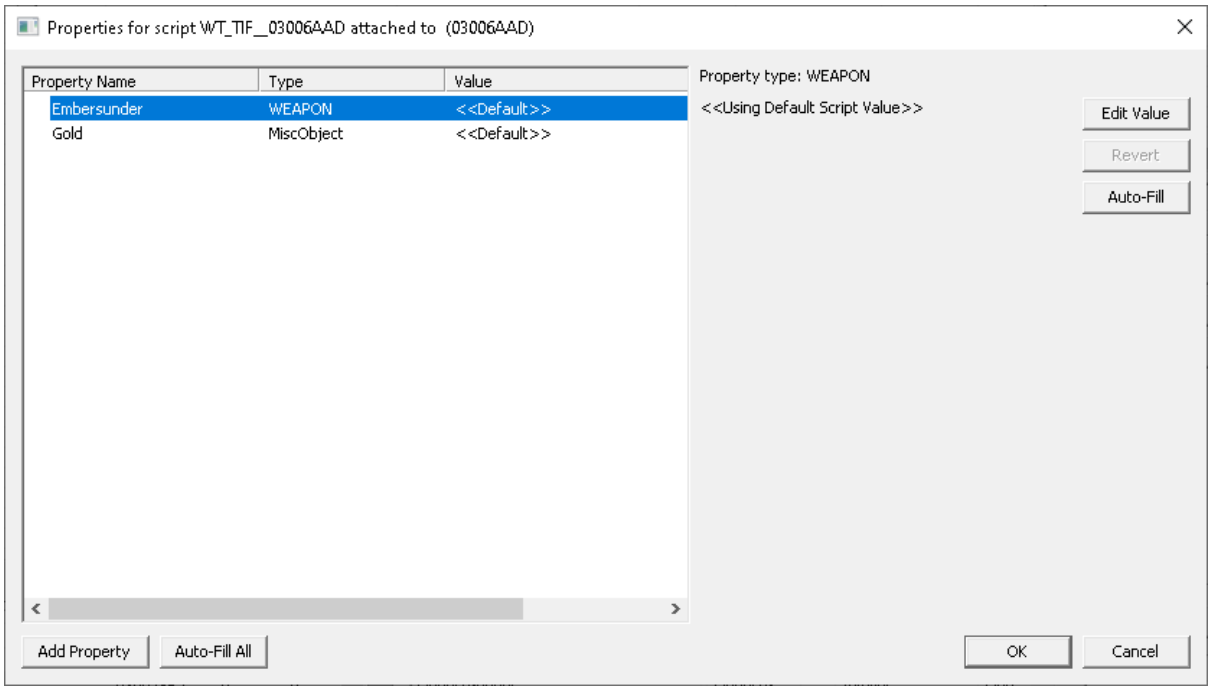


Figure 727 - The two variables added.

Highlight the Embersunder variable and click on the Edit Value button.

Set the Pick Object drop-down to point to the sword the player obtained.

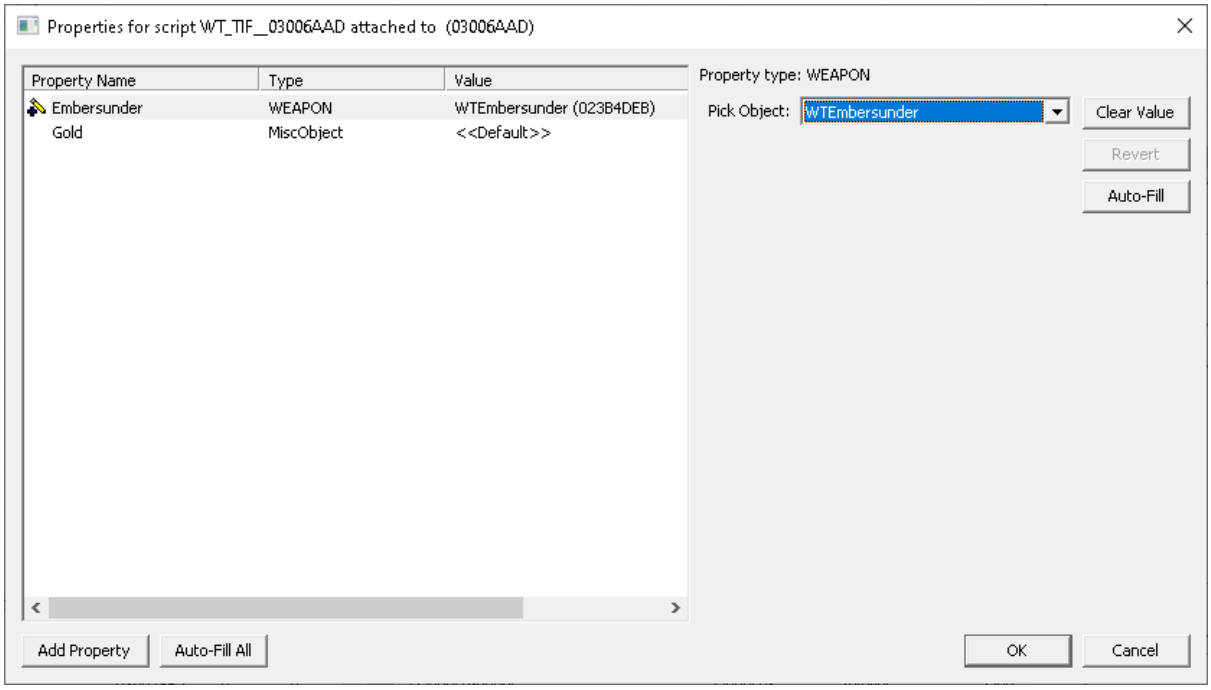


Figure 728 - Setting the Embersunder variable's value.

Next, highlight the Gold variable and click on the Edit Value button.

Set the Pick Object drop-down to point to Gold001.

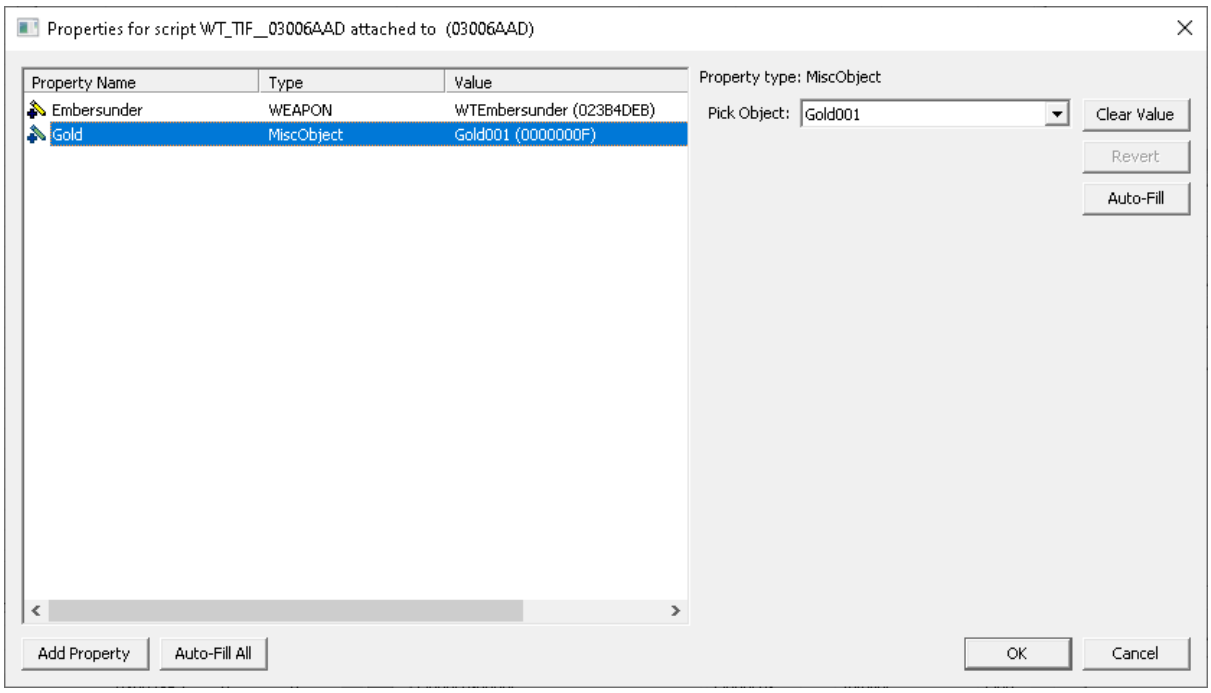


Figure 729 - Setting the Gold variable's value.

Click OK to close out of script properties.

Now when we compile the papyrus code in End, we should no longer receive an error.

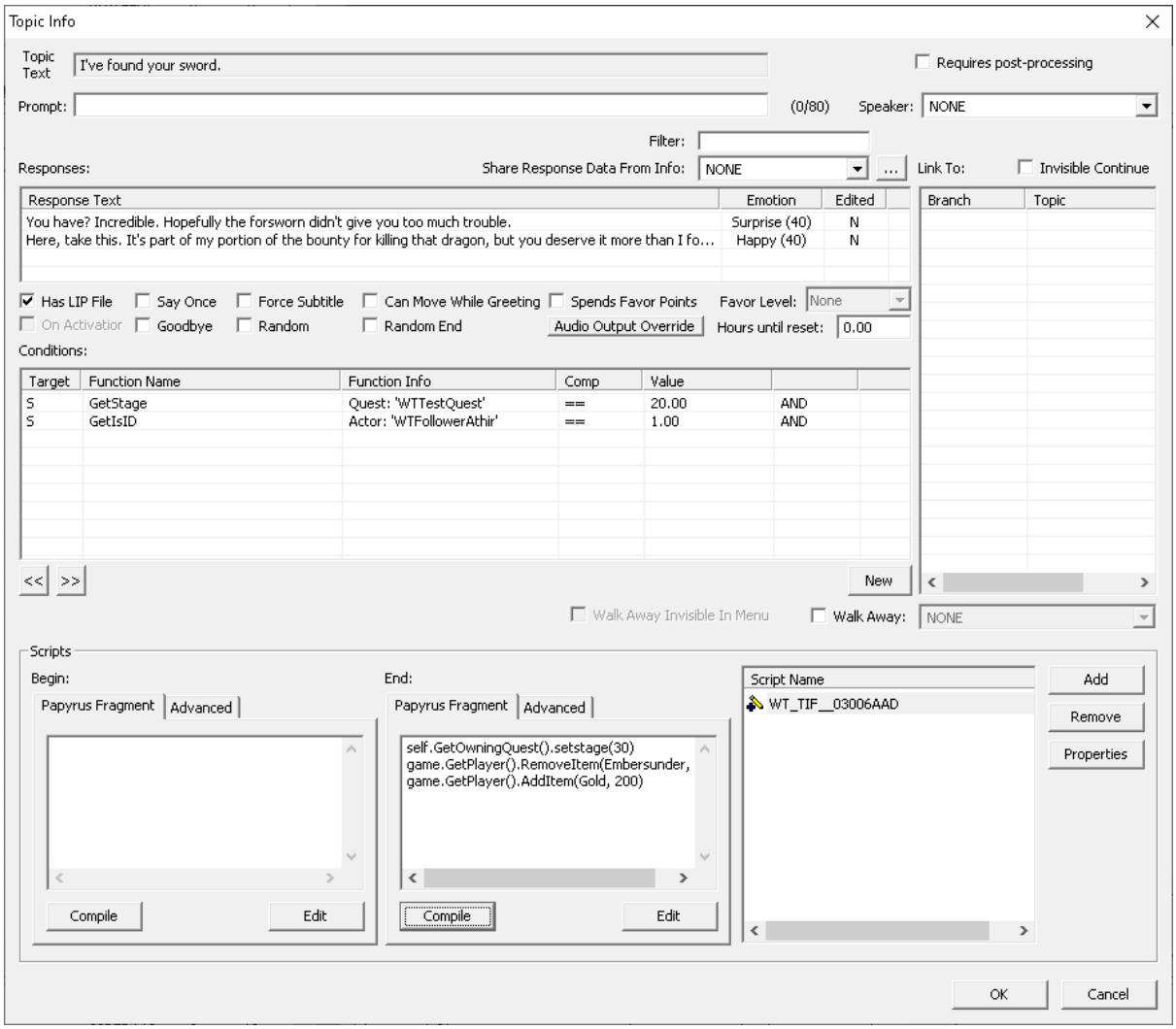


Figure 730 - Papyrus compiled successfully.

Click OK to close out of Topic Info, click OK to close out of Topic, then click OK to close out of Quest properties.

CHAINING DIALOGUE TOPICS TOGETHER

Say for example we wanted an NPC to say something and give the player multiple choices to choose from. We can accomplish this by chaining topics together in Dialogue Views.

Let's start by creating a new branch.

In the Dialogue Views tab under WTTestQuestDialogueView, right-click in the empty area and select 'Create branch'.

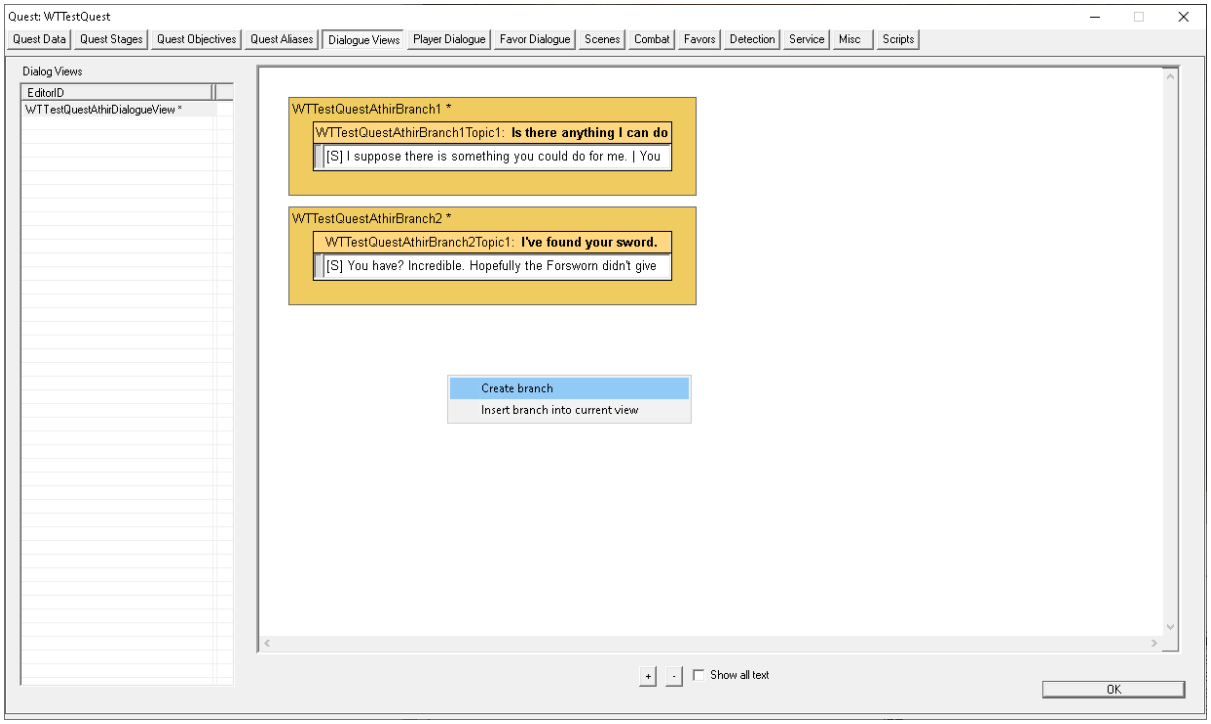


Figure 731 - Creating a new branch.

Set the branch ID then click OK. For this example I set it to WTTestQuestDialogueChain.



Figure 732 - Setting the branch ID.

Set the topic ID of the first topic then click OK. Again, for this example I just named it WTTestQuestDialogueChainTopic1.

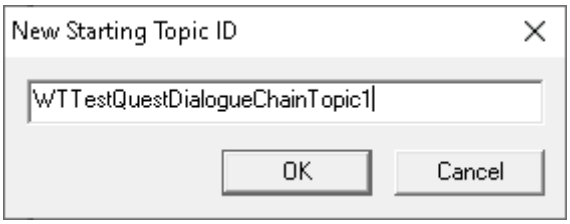


Figure 733 - Setting the topic ID of the first topic.

Double-click or right-click on the new topic and select 'Edit topic'.

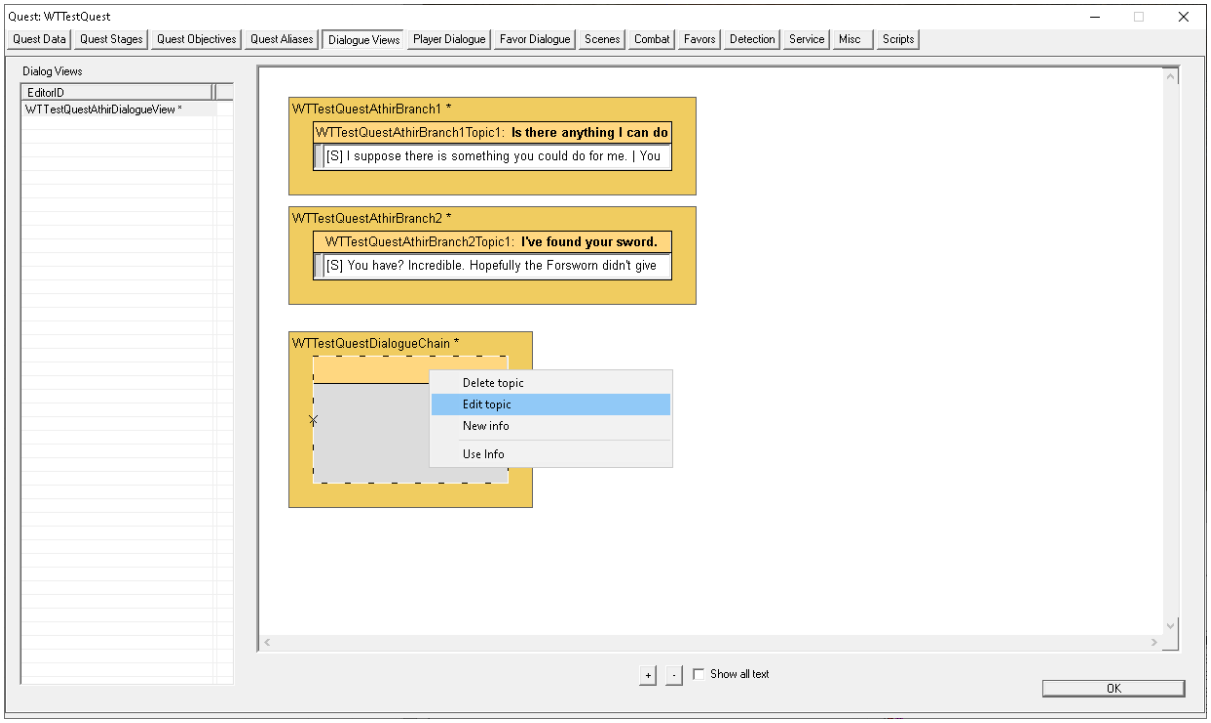


Figure 734 - Editing the first topic in the dialogue chain.

Enter topic text into the Topic Text field then right-click in the Info list below and select New to create a new NPC response.

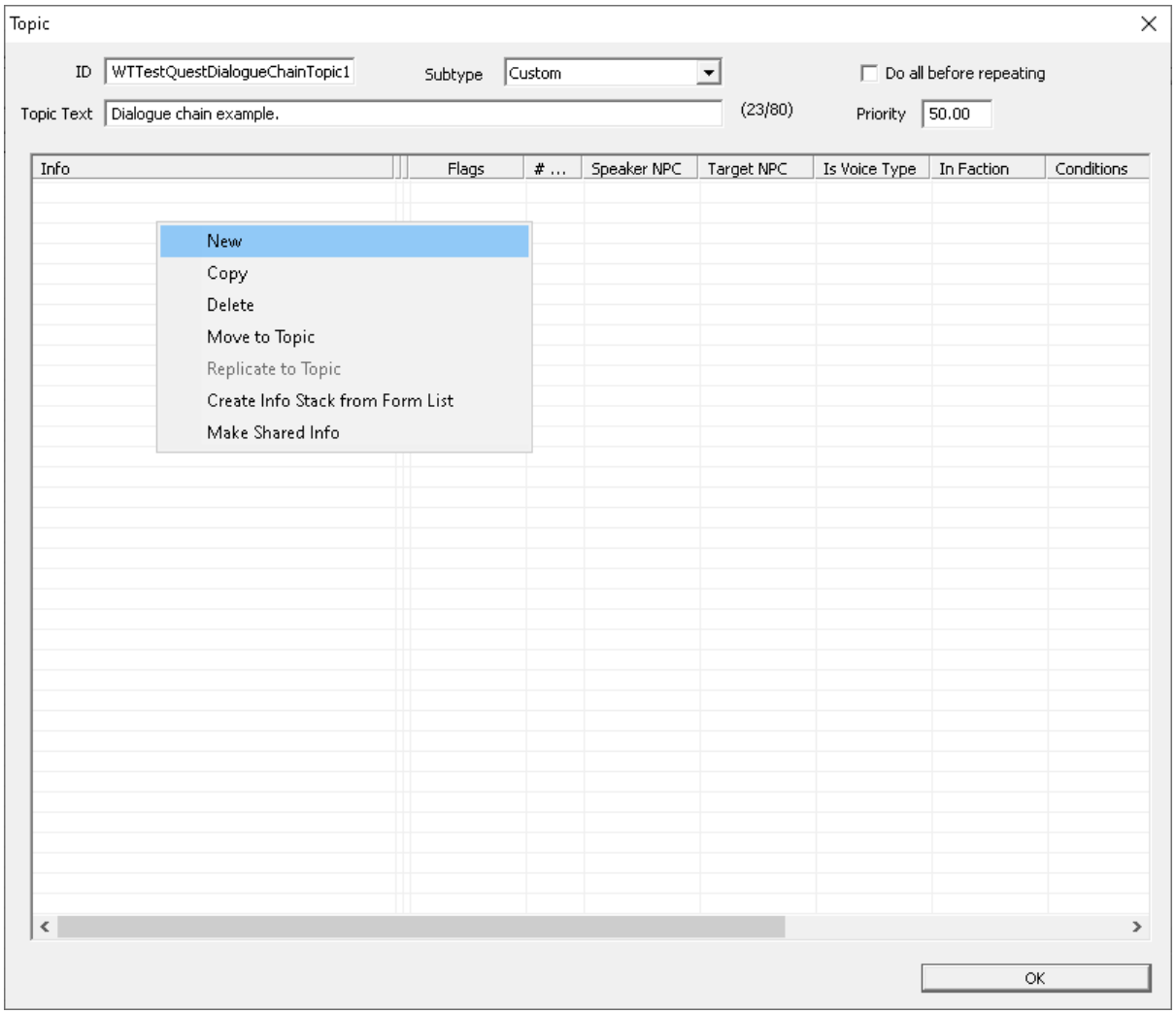


Figure 735 - Setting the topic text and adding a response.

Enter in the dialogue that the NPC will respond with in the Response Text field then click OK.

Obviously not the most riveting line of dialogue. I'm just keeping things simple here for the sake of this example.

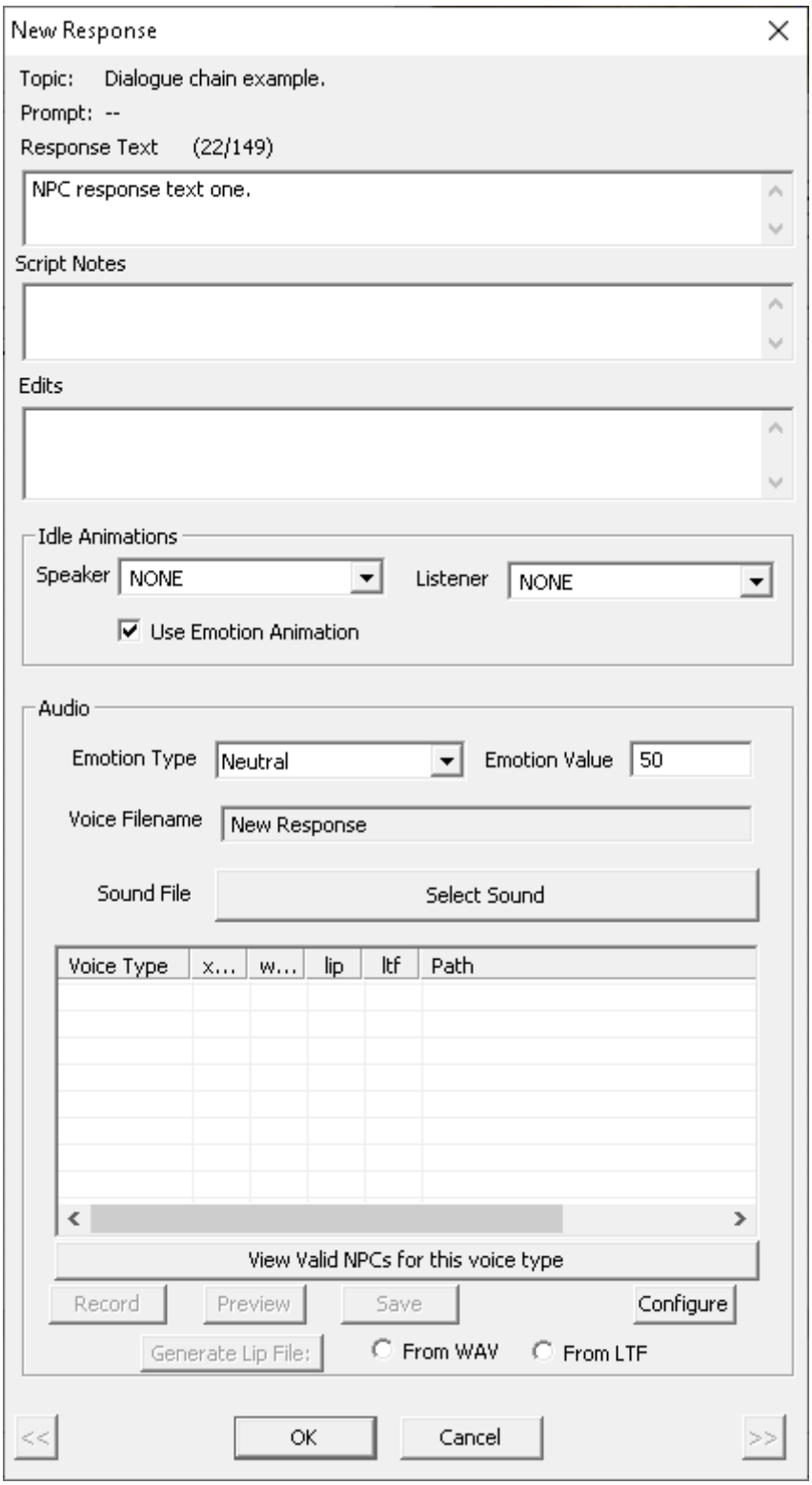


Figure 736 - The NPC's first response.

Right-click in the Conditions list and select New.

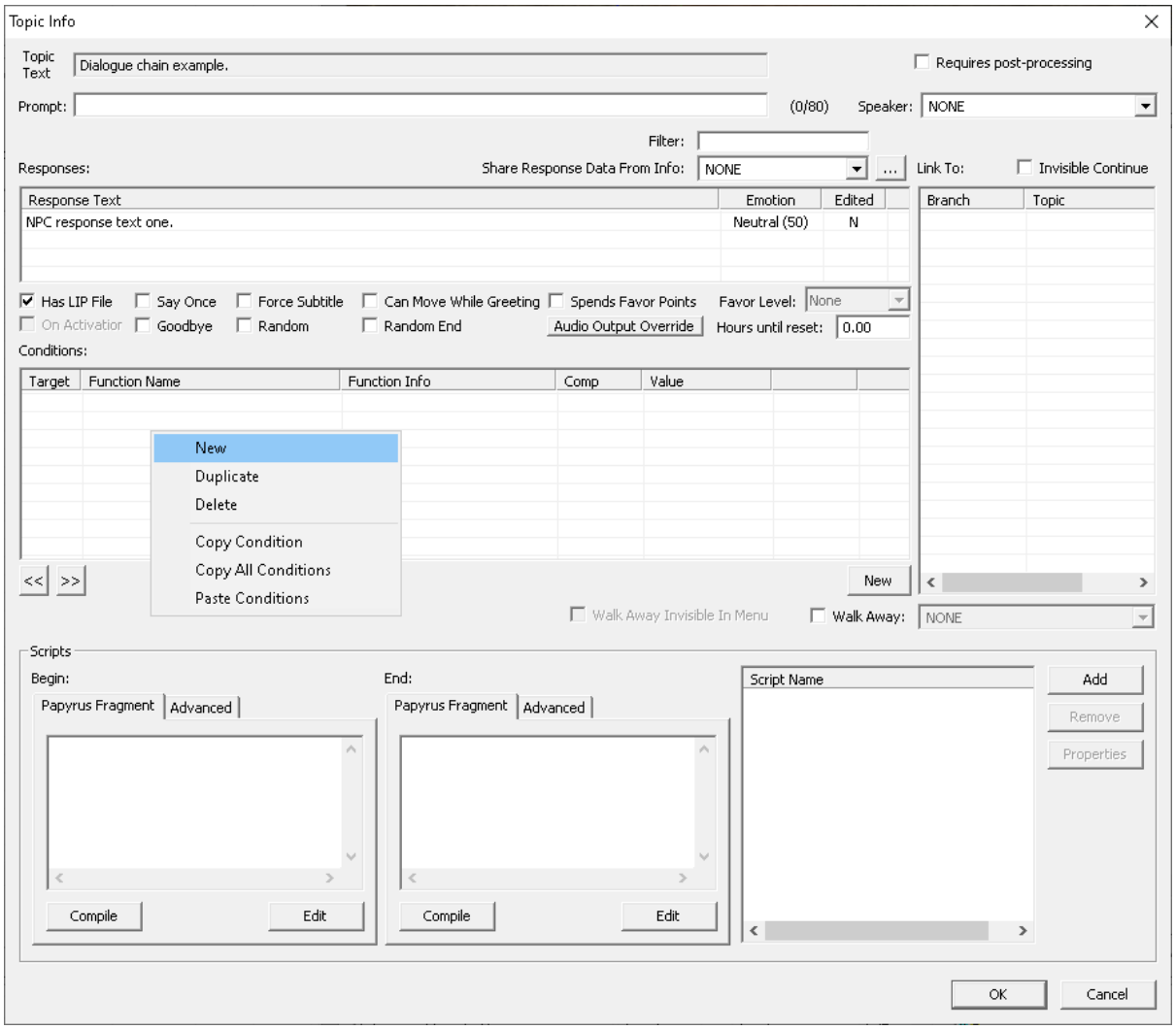


Figure 737 - Adding a new condition.

Ensure the Condition Function drop-down is set to GetIsID and set the actor. Since I'm setting this example up in the WTTestQuest example, I'm just going to reuse Athir.

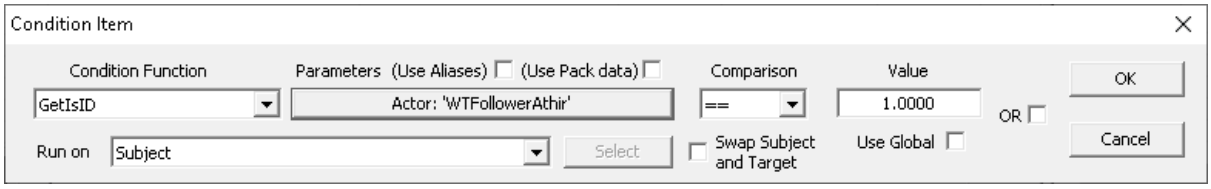


Figure 738 - Setting Athir as the speaker of this line of dialogue.

Click OK to close out of Condition Item.

Click OK to close out of Topic Info.

Click OK to close out of Topic.

Right-click in the WTTestQuestDialogueChain branch and select ‘Add topic’.

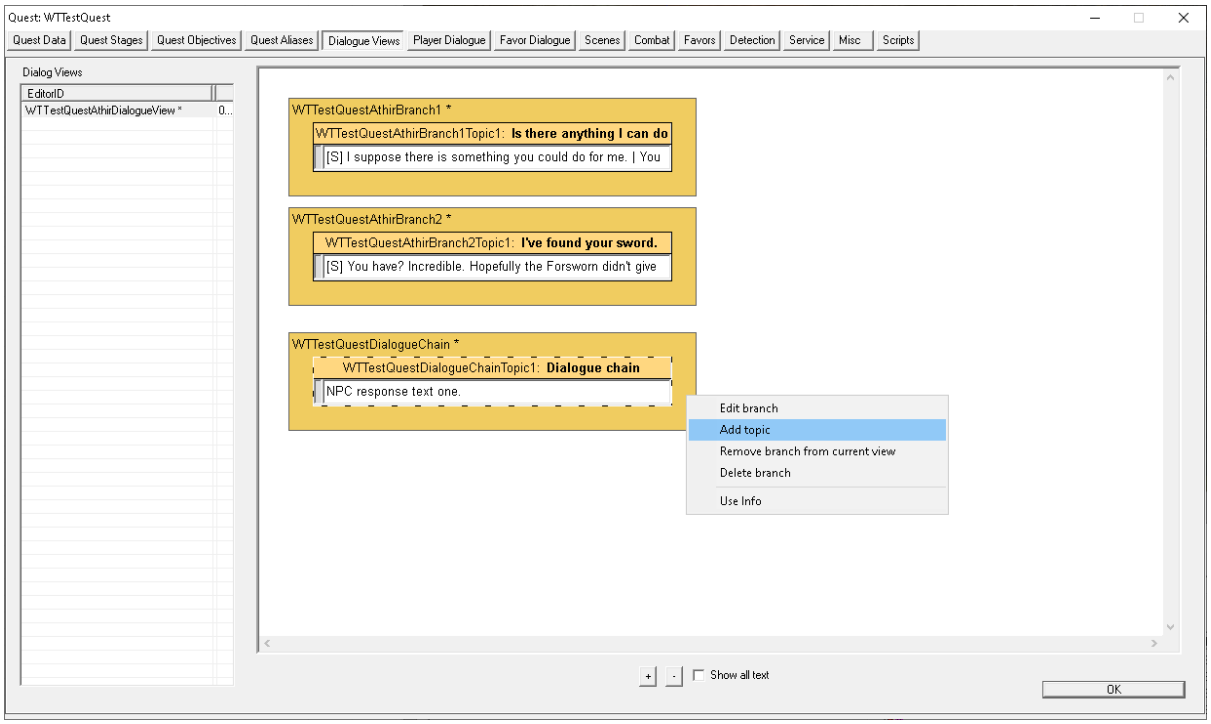


Figure 739 - Adding a second topic to the WTTestQuestDialogueChain branch.

Set the player response in the Topic Text field then right-click in the Info list and select New.

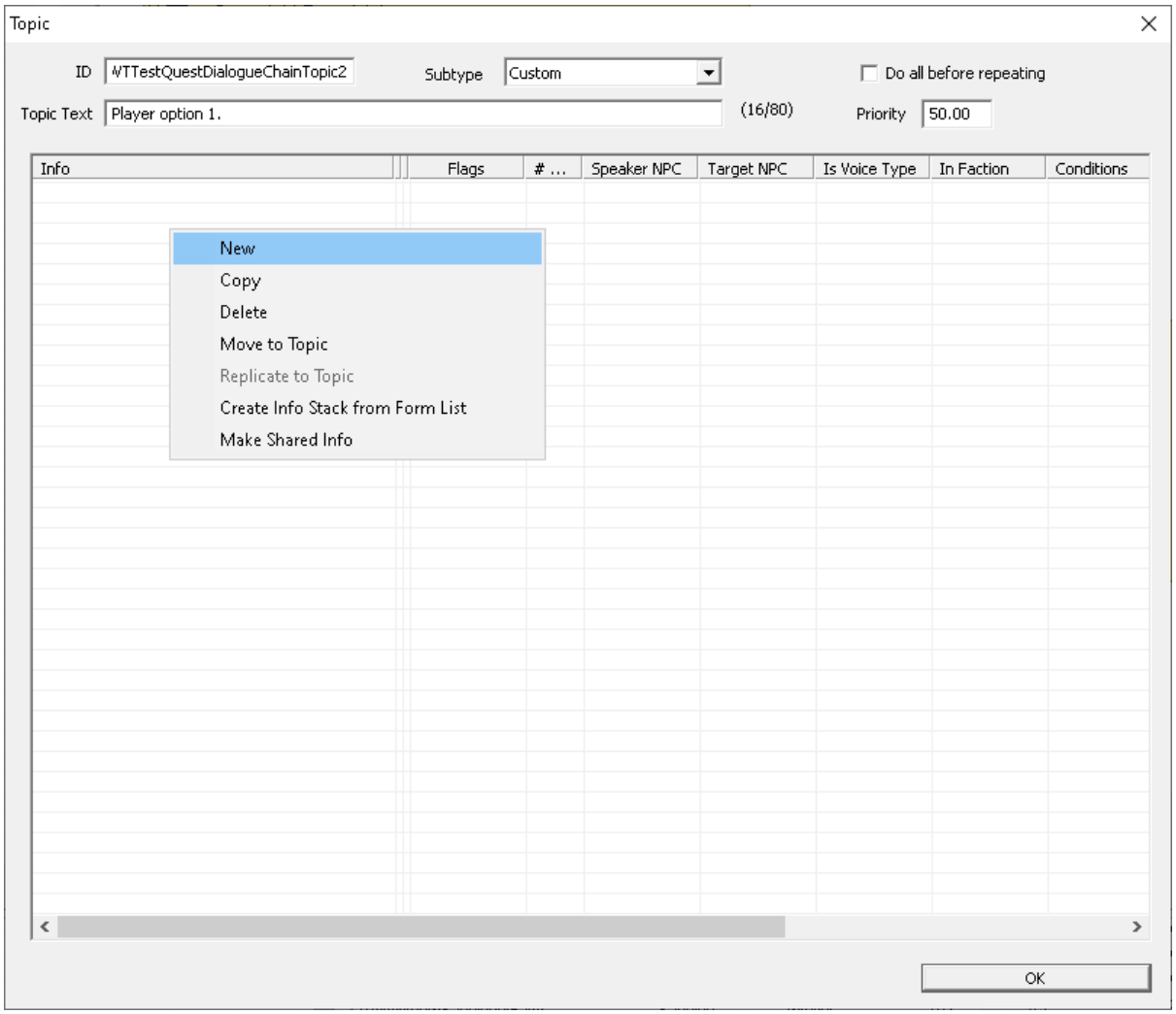


Figure 740 - Setting the ID, topic text and adding a response.

Repeat the steps mentioned earlier to add in the NPC's response.

Then add a third topic to the same branch by once again right-clicking in the WTTestQuestDialogueChain branch and selecting 'Add topic'.

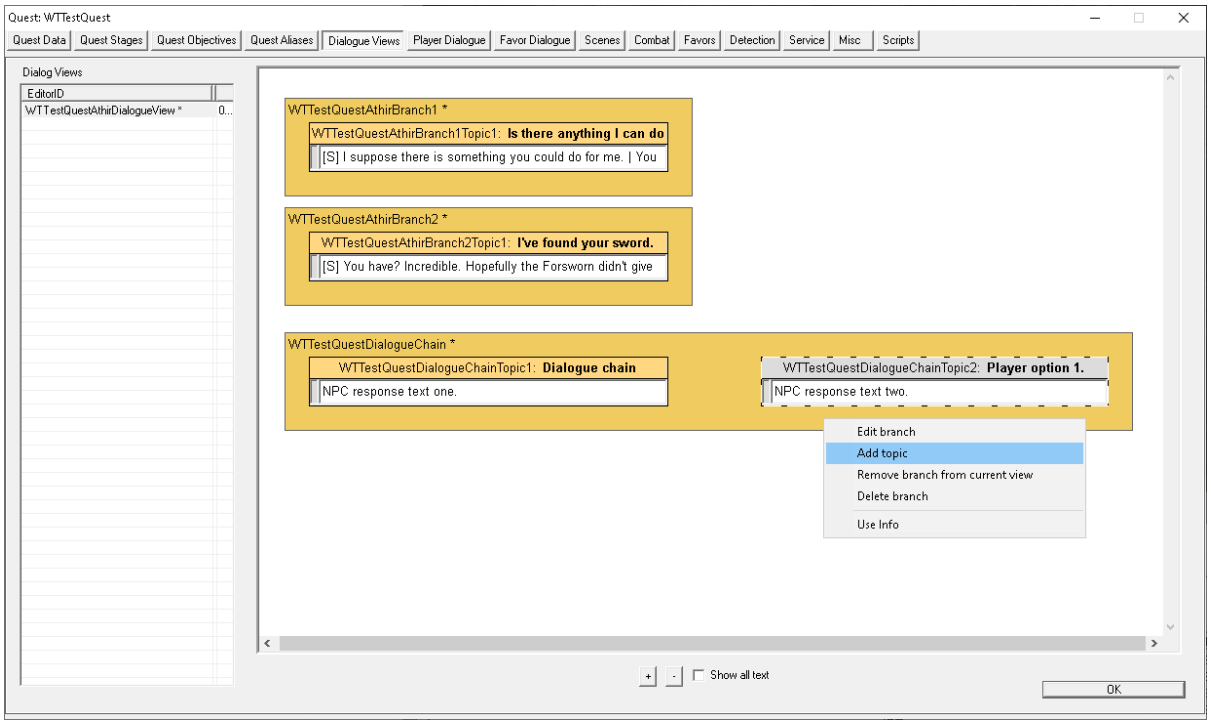


Figure 741 - Adding a third topic to WTTestQuestDialogueChain.

We should now have three topics in the same branch. Just align them into position like they are in the screenshot below:

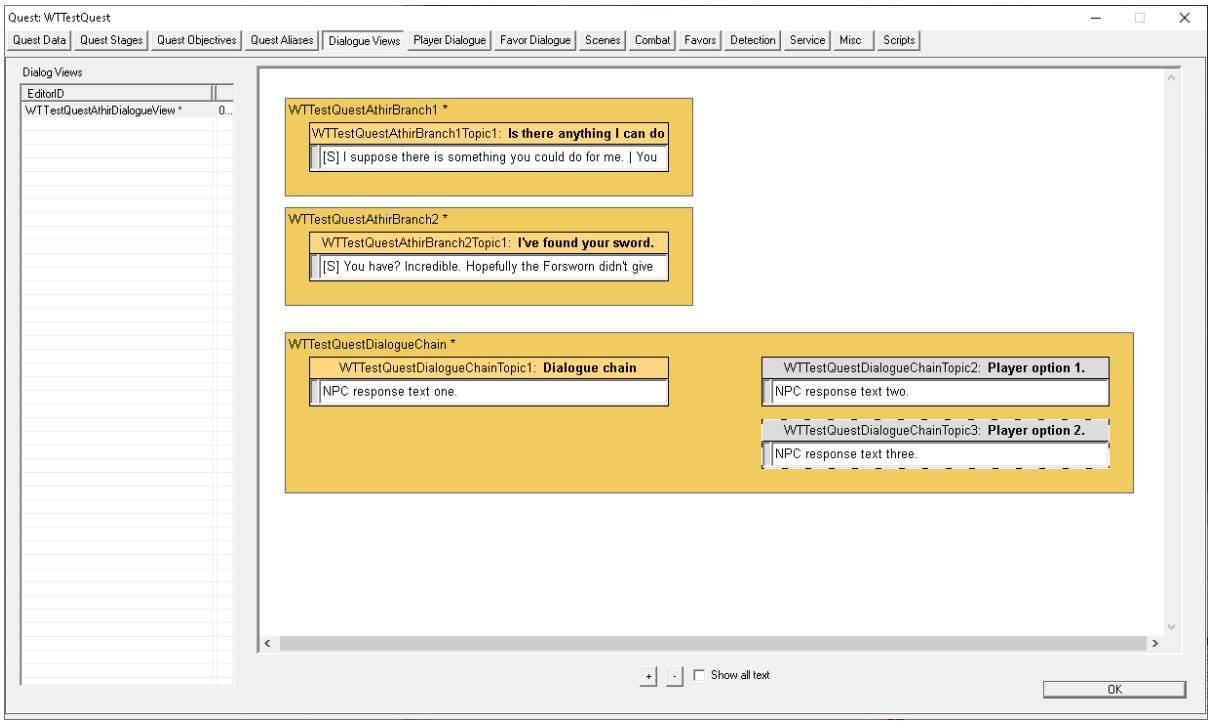


Figure 742 - Three topics in the same branch.

Now to chain them together.

Click on WTTestQuestDialogueChainTopic1 to select it. When you hover over the NPC response text, your mouse cursor should change into a hand icon. Left-click and drag from the response text in WTTestQuestDialogueChainTopic1 to WTTestQuestDialogueChainTopic2 to link these two topics together.

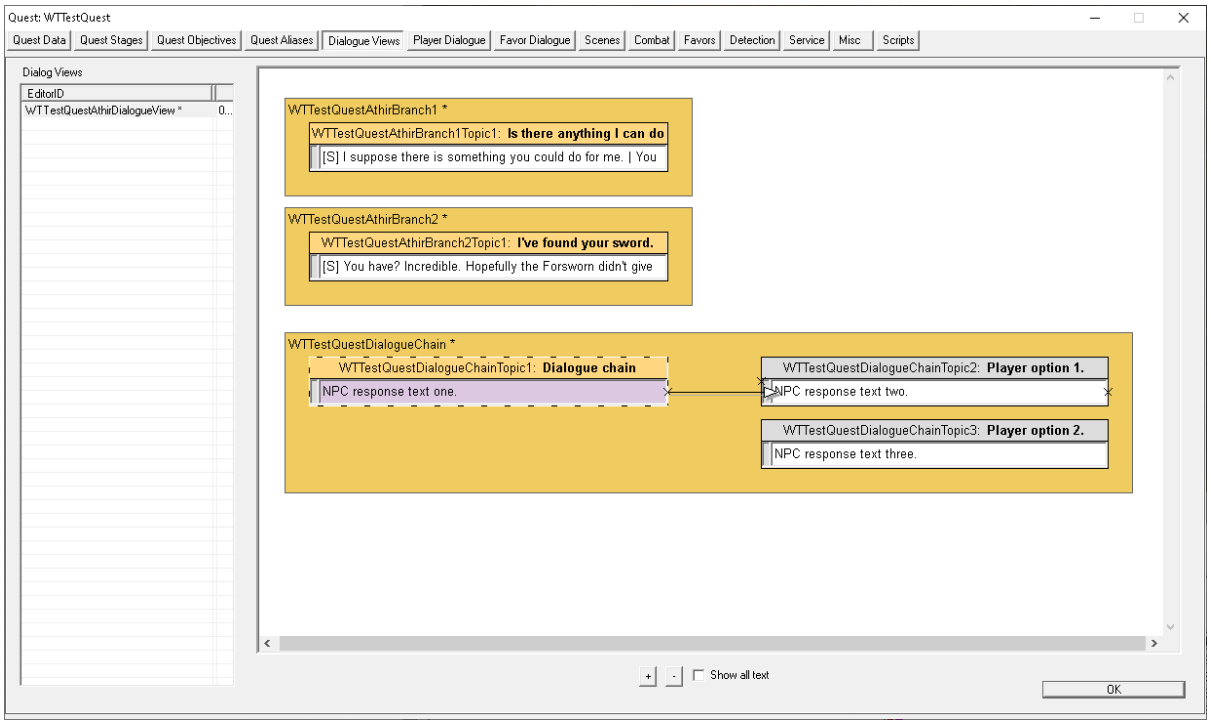


Figure 743 - Linking topic 1 to topic 2.

Repeat these steps to link WTTestQuestDialogueChainTopic1 to WTTestQuestDialogueChainTopic3 as well.

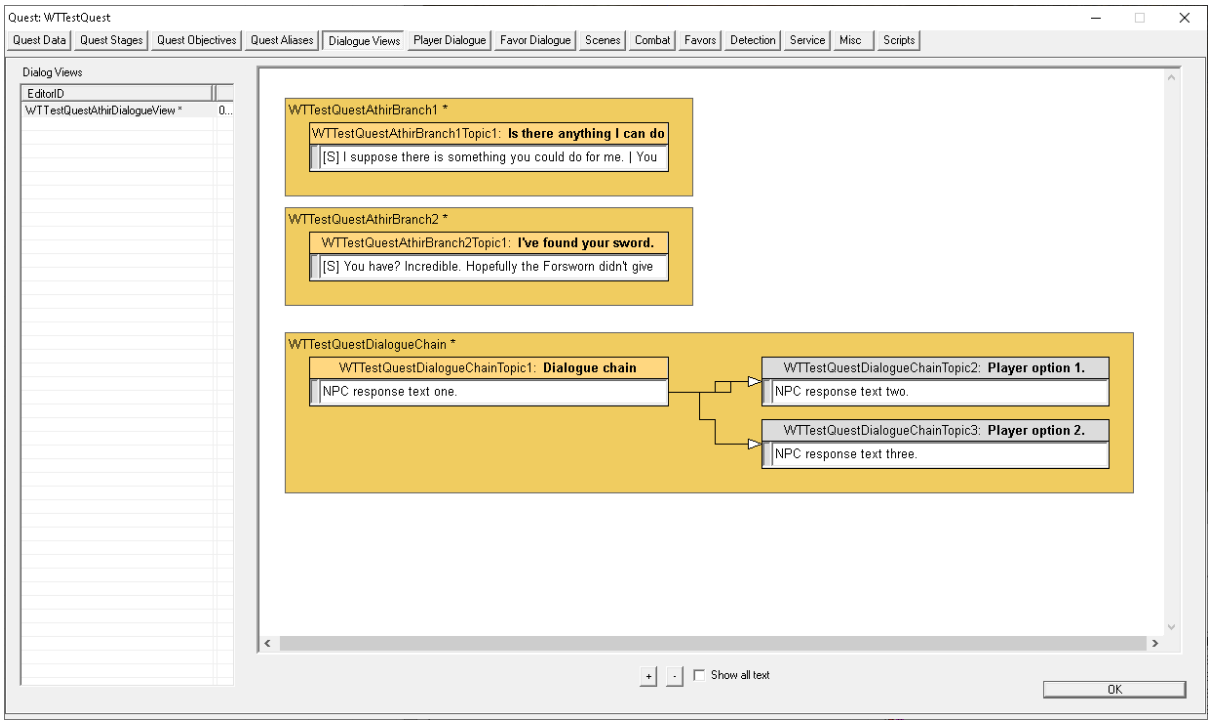


Figure 744 - Linking topic 1 to topic 3.

In-game we should now see those two choices presenting themselves when we speak to Athir after selecting 'Dialogue chain example'. But what if the player tabs out of dialogue before choosing the next response?

By default, if the player tabs out before selecting a response, dialogue with the NPC will end. But we can set up a default choice instead to continue dialogue with the NPC.

In order to do this, open the Topic Info properties of the preceding line of dialogue, tick Walk Away, then select the topic to automatically progress to should the player attempt to tab out of dialogue.

In the following example, I set the Walk Away topic to WTTestQuestDialogueChainTopic2.

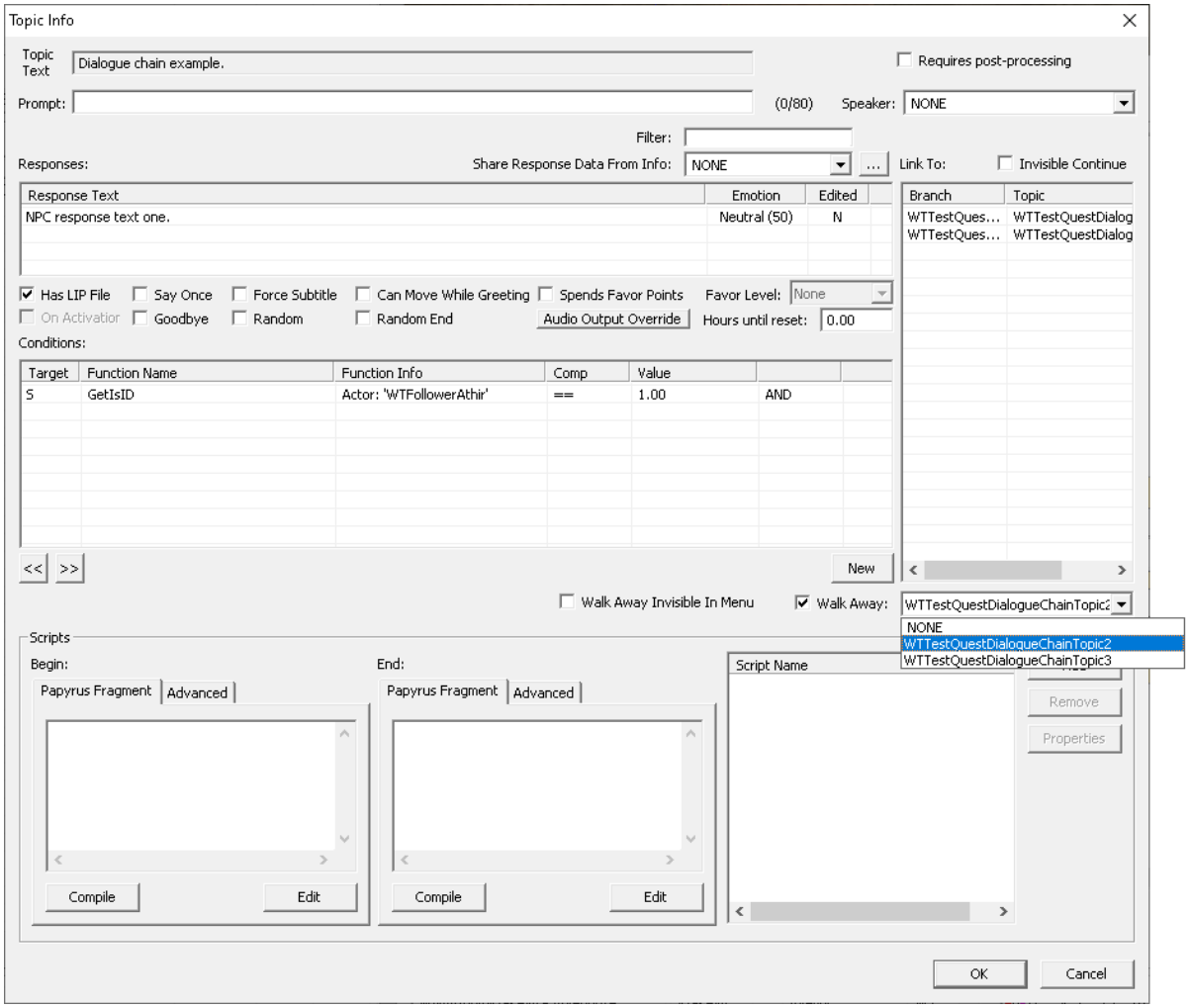


Figure 745 - Setting the walk away topic for 'Dialogue chain example'.

In the Dialogue Views tab, we should now see a red arrow between WTTestQuestDialogueChainTopic1 and WTTestQuestDialogueChainTopic2.

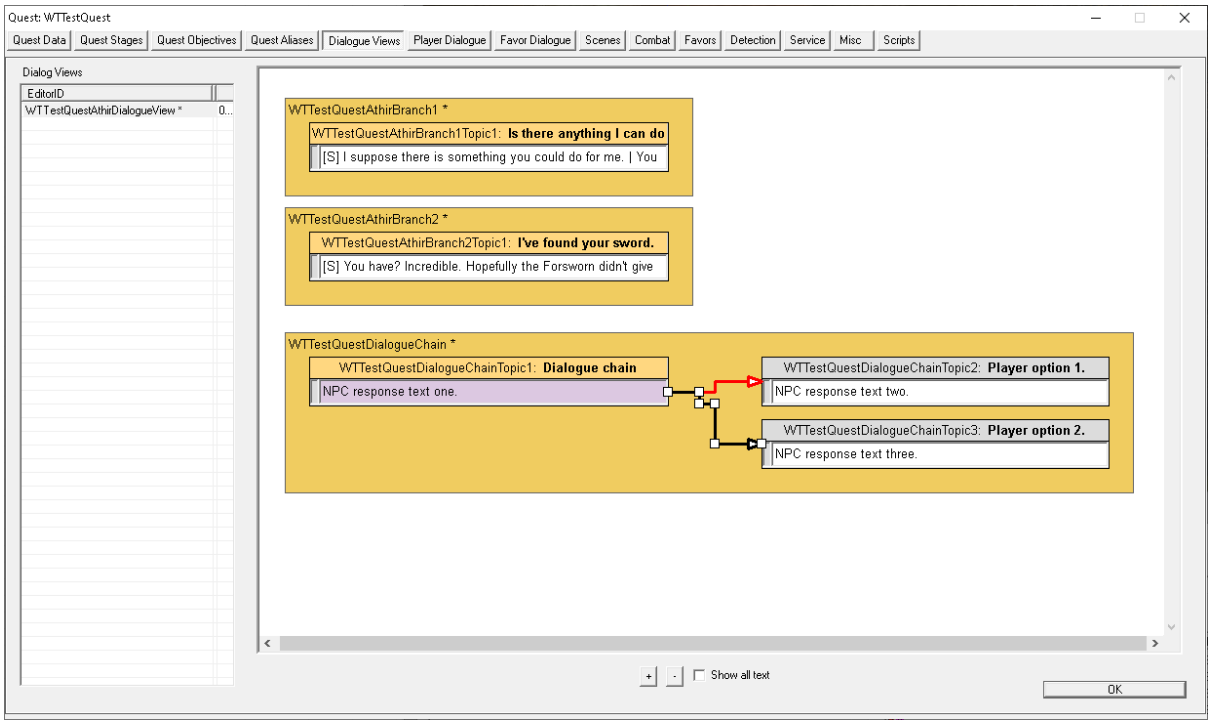


Figure 746 - Walk Away shown in Dialogue Views.

Another thing we can do is set up an invisible continue. This allows us to link multiple topics together.

An invisible continue is represented in the dialogue view as a green line between topics.

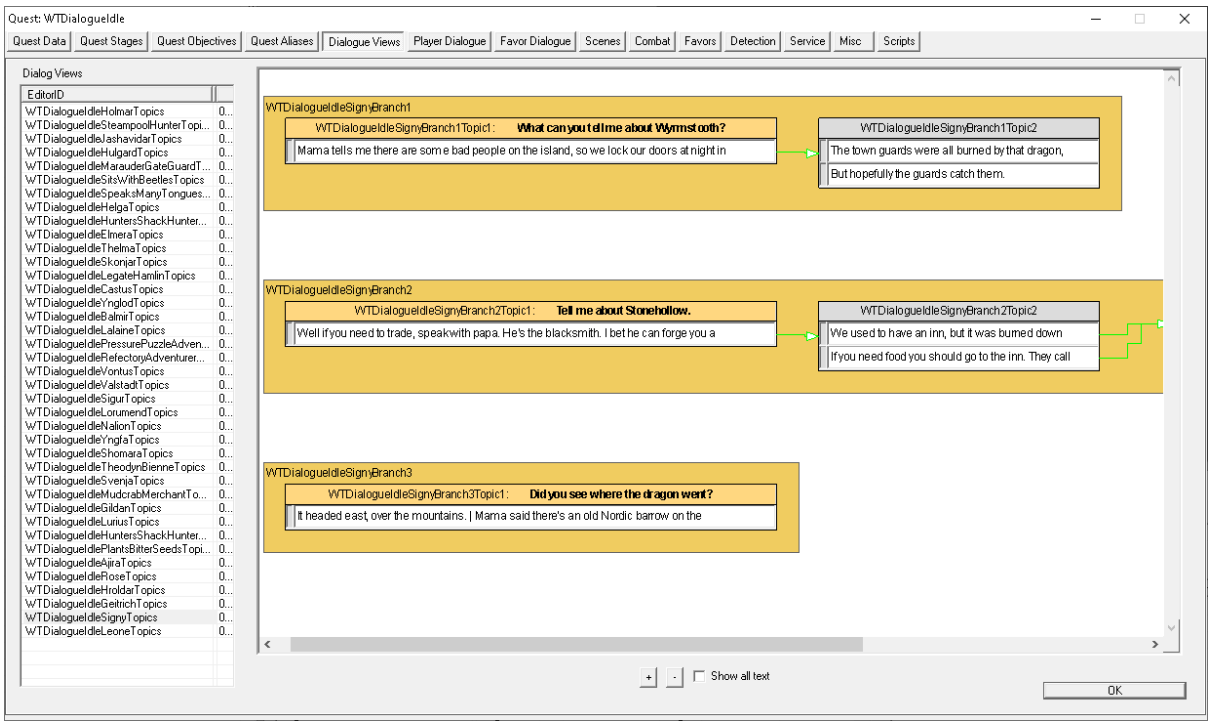


Figure 747 - Invisible Continue.

We can use invisible continues to break up a single big topic without needing to prompt the player to select a dialogue response in between. In the following example I used an invisible continue to change what an NPC says based on different conditions.

Open the first topic.

Tick 'Invisible Continue'.

If you haven't already linked the topics together, right-click in the dialogue list below it and select Add Link.

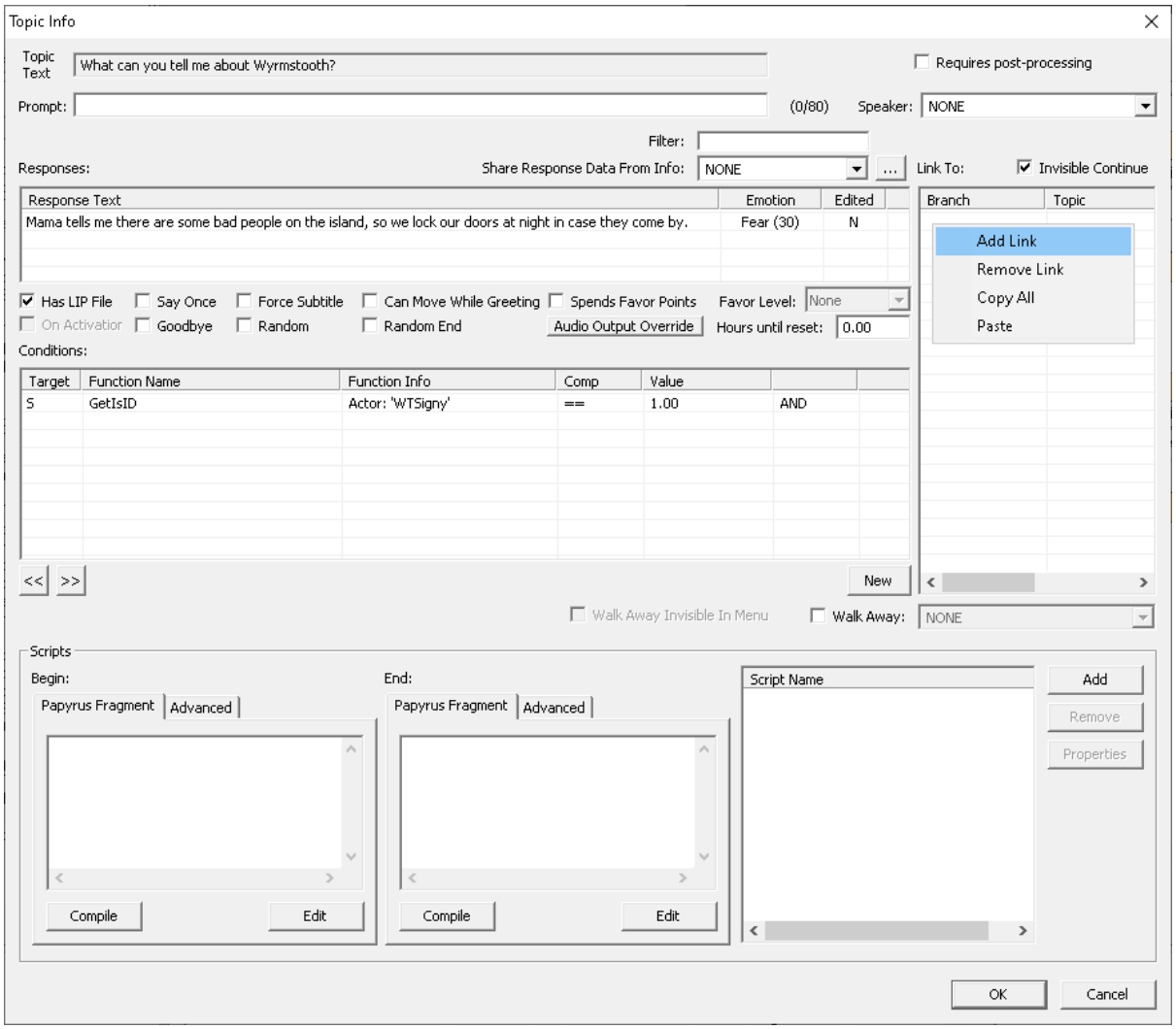


Figure 748 - Adding an invisible continue.

Select the topic you want to continue to then click OK.

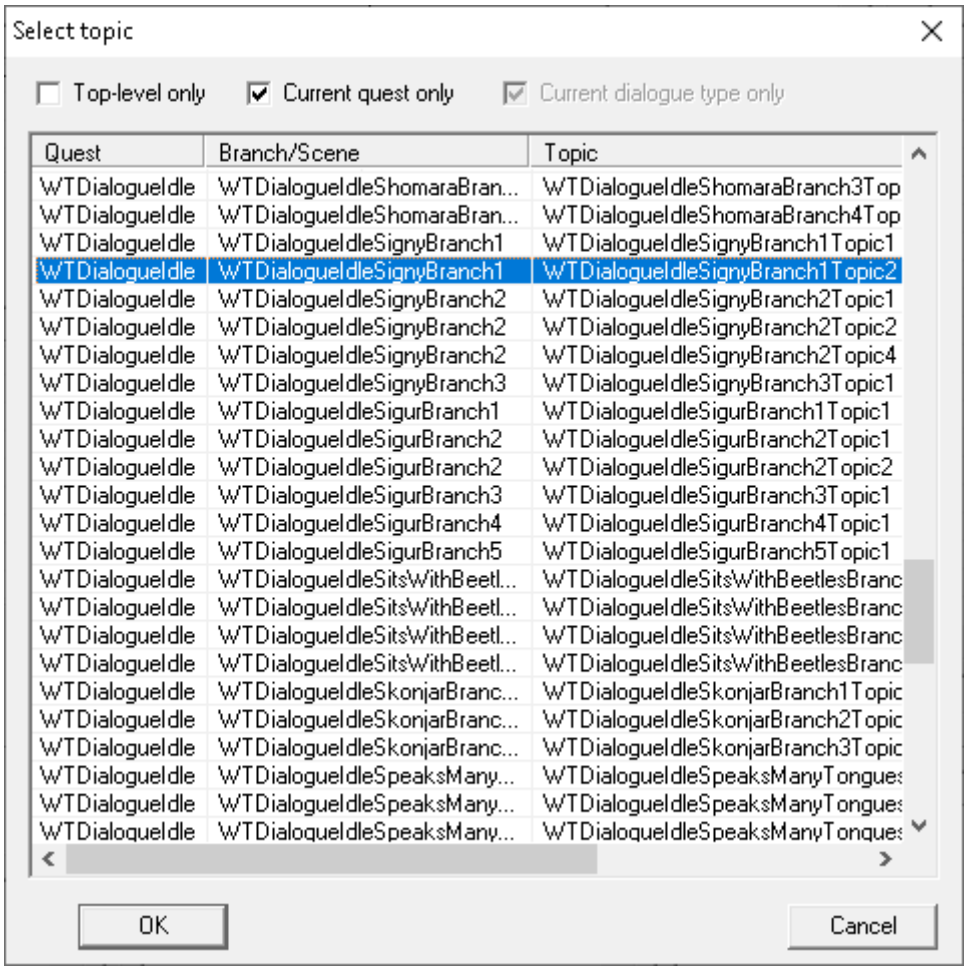


Figure 749 - Selecting the topic to continue to.

Confirm that the topic was added successfully then click OK to close out of Topic Info.

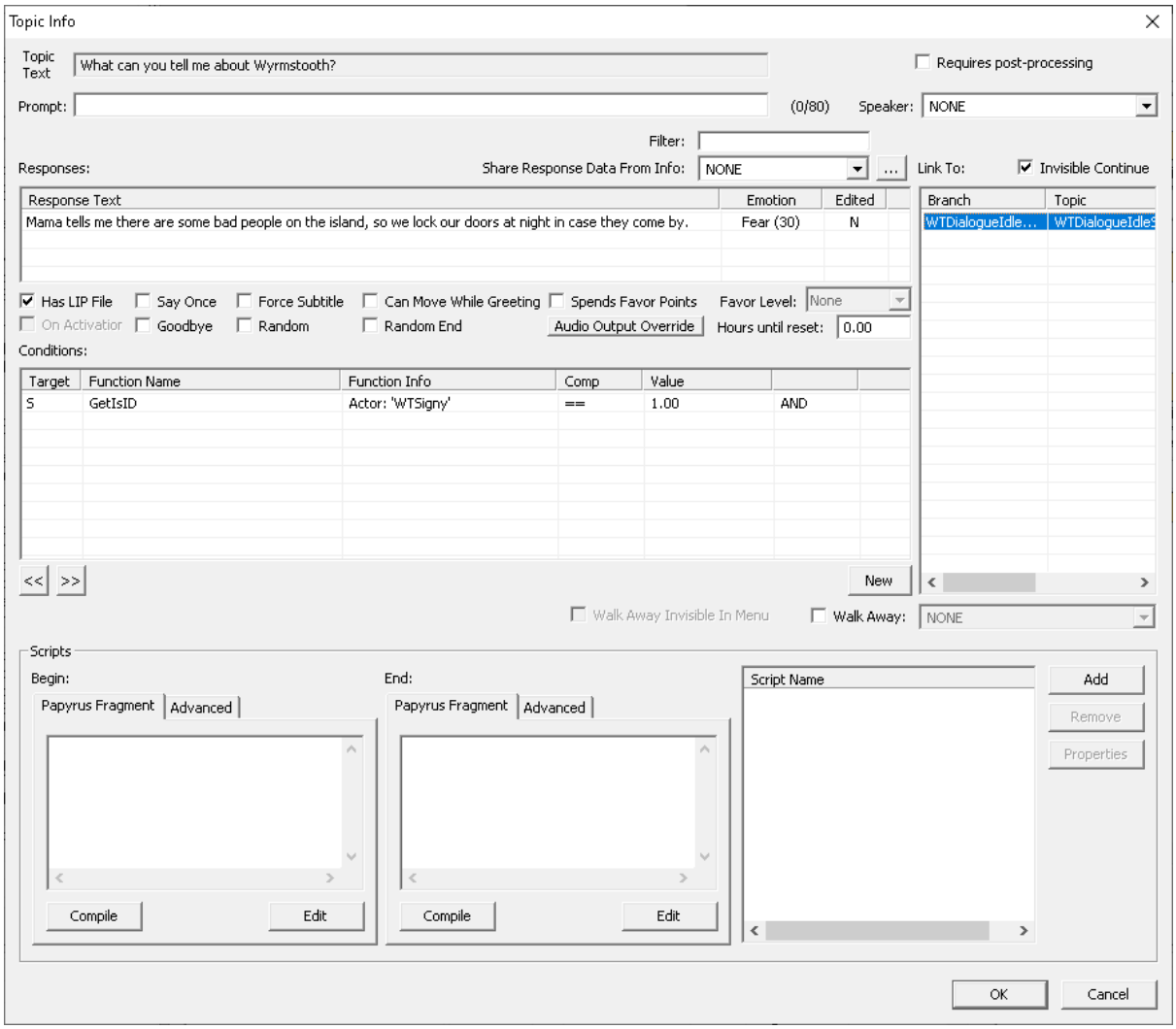


Figure 750 - Next topic linked to.

Ensure the Topic Text field is blank in the topic we've linked to.

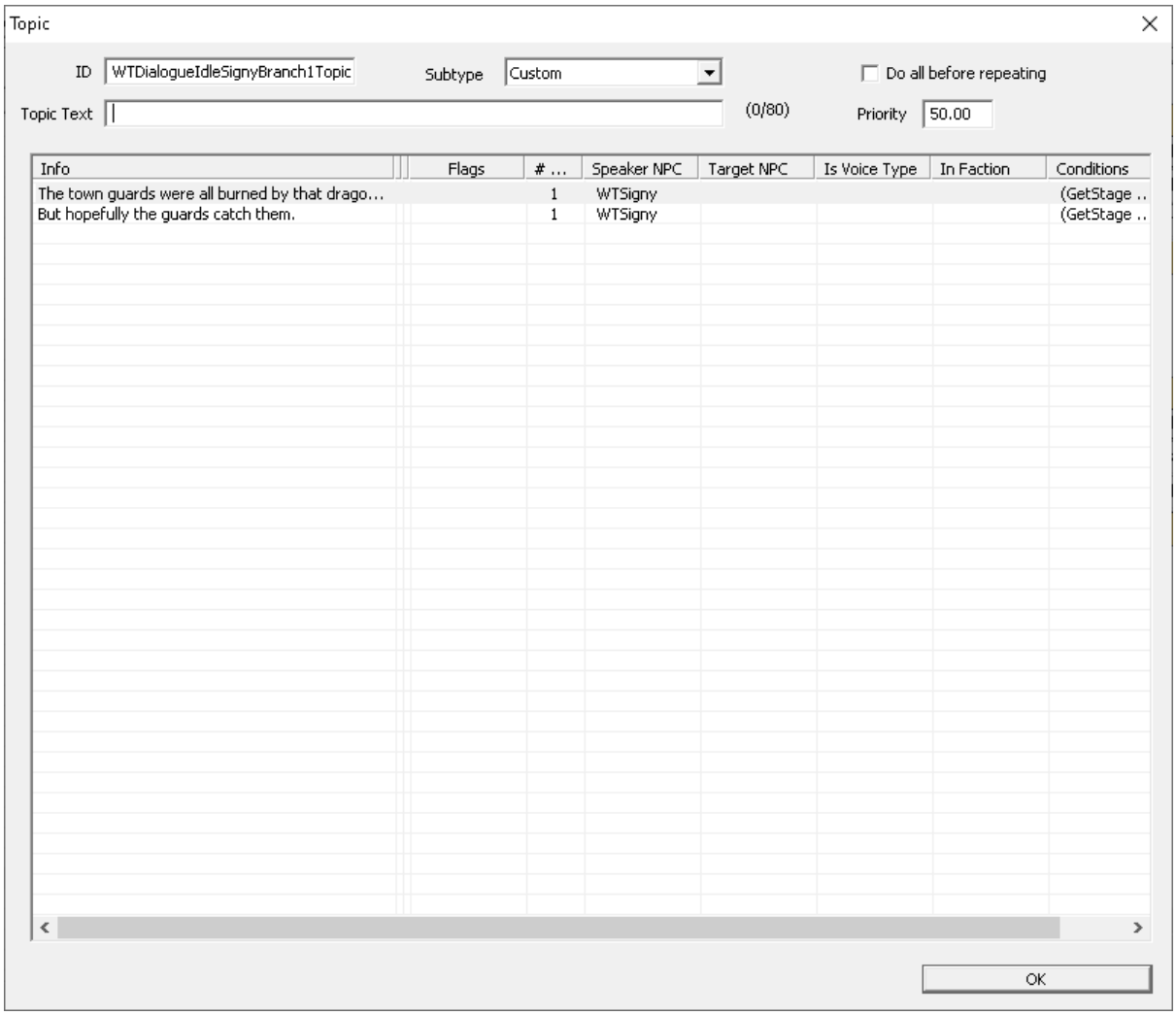


Figure 751 - Empty topic text.

So this topic contains two Topic Infos, allowing the NPC to say two different things based on our condition check.

If WTBarrowOfTheWyrm hasn't been completed yet, Signy will follow on and end the topic chain with this line:

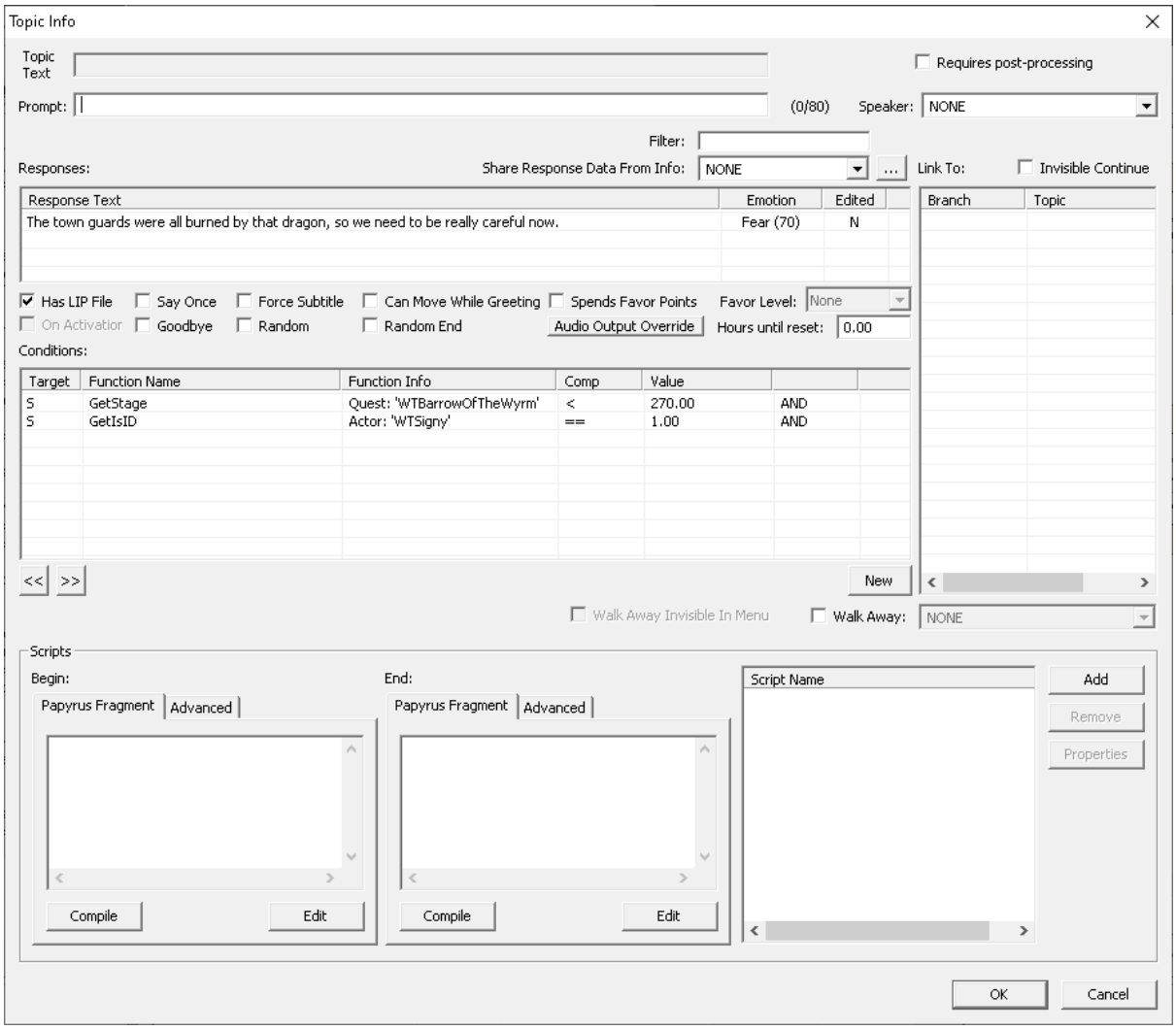


Figure 752 - First Topic Info in the linked topic.

Otherwise if WTBarrowOfTheWyrm has been completed, Signy will end the topic chain with this instead:

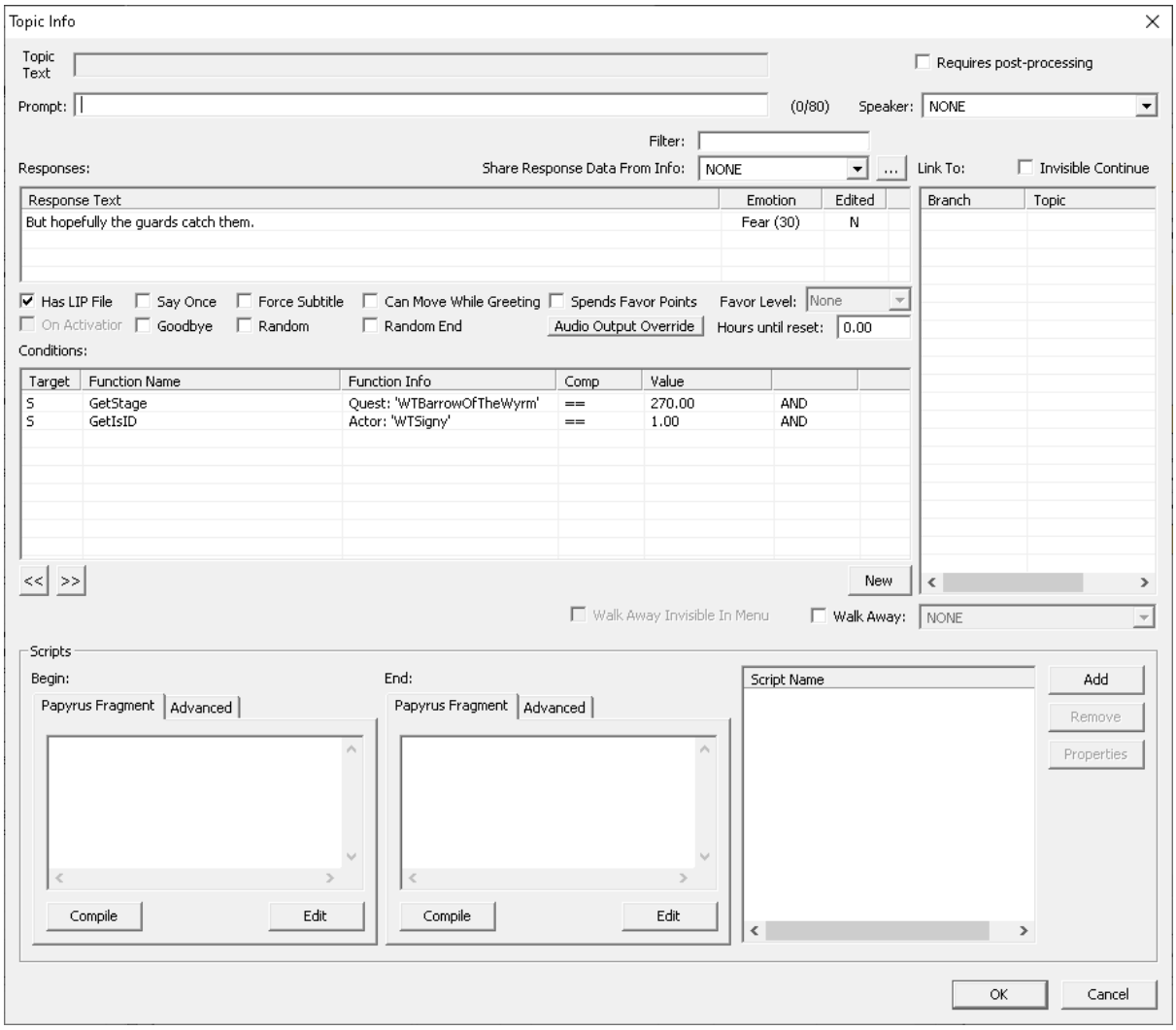


Figure 753 - Second Topic Info in the linked topic.

BLOCKING BRANCHES

If we wanted dialogue in a specific branch to be said by the NPC with priority over even the hello lines, we can set the branch up as a blocking branch.

Let's start by creating a new branch with a topic that we want the NPC to say first.

In the Dialogue Views tab, right-click in the blank area and select 'Create branch'.

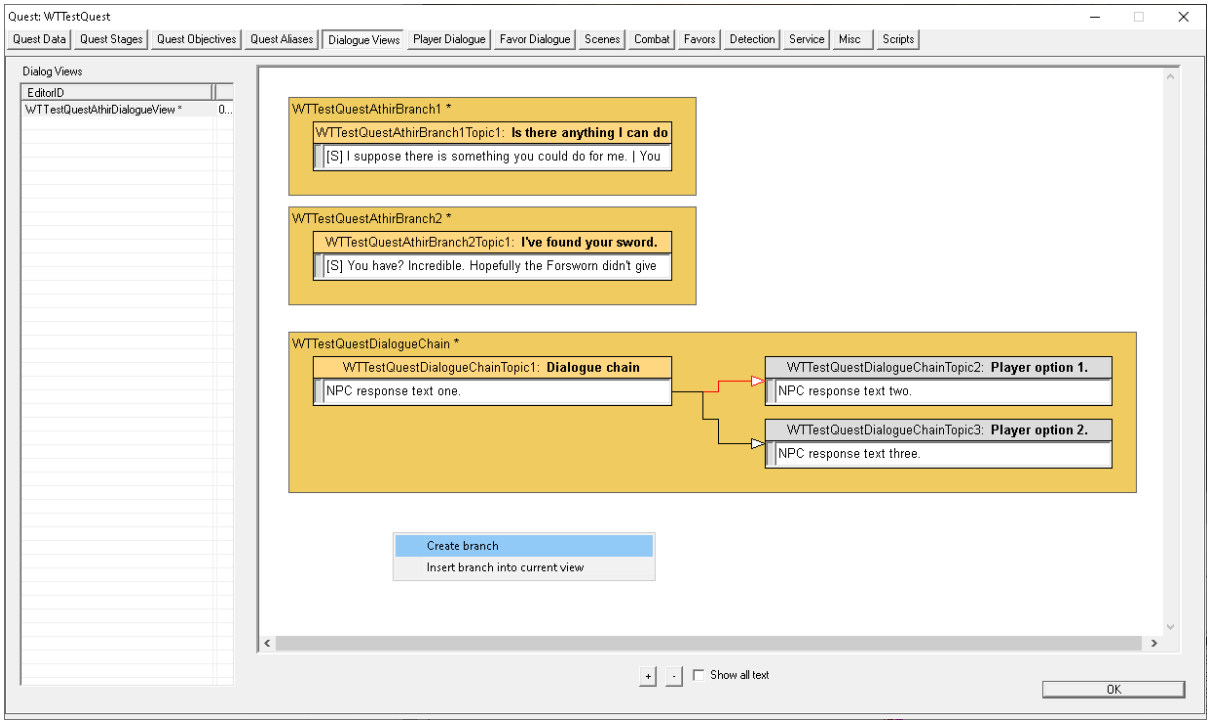


Figure 754 - Creating a new branch.

Enter in the ID of the new branch then click OK. For this example, I set it to WTTestQuestBlockingBranch.

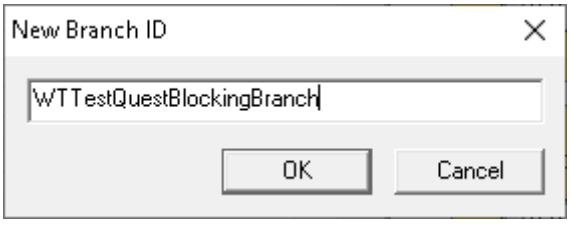


Figure 755 - Setting the ID of the new blocking branch.

Enter in the ID of the first topic in the new branch then click OK.

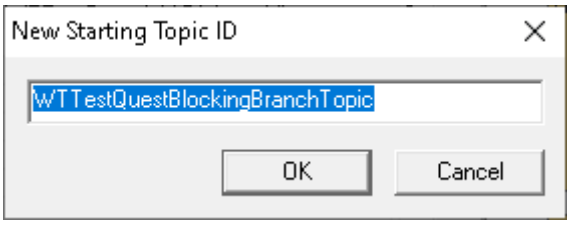


Figure 756 - Setting the ID of the new blocking branch's topic.

Double-click or right-click on the topic in the new branch and select 'Edit topic'.

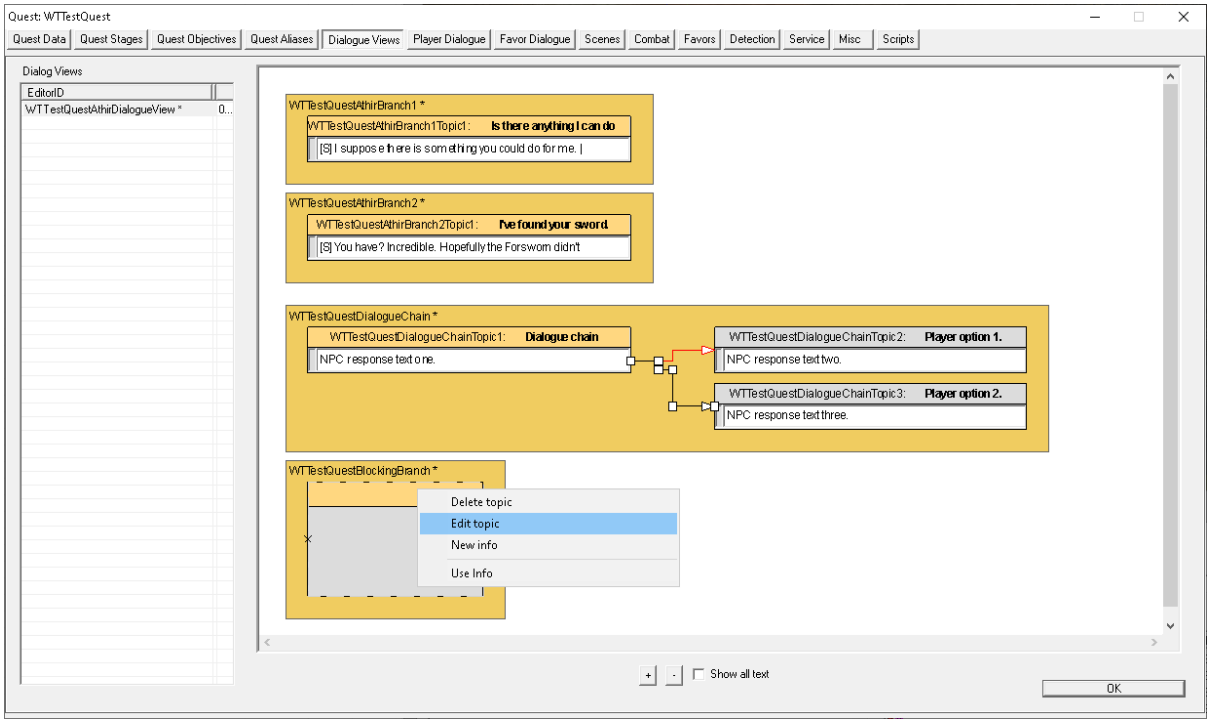


Figure 757 - Editing the topic in the new branch.

Don't add anything into the Topic Text field. Instead, just right-click in the Info list and select New.

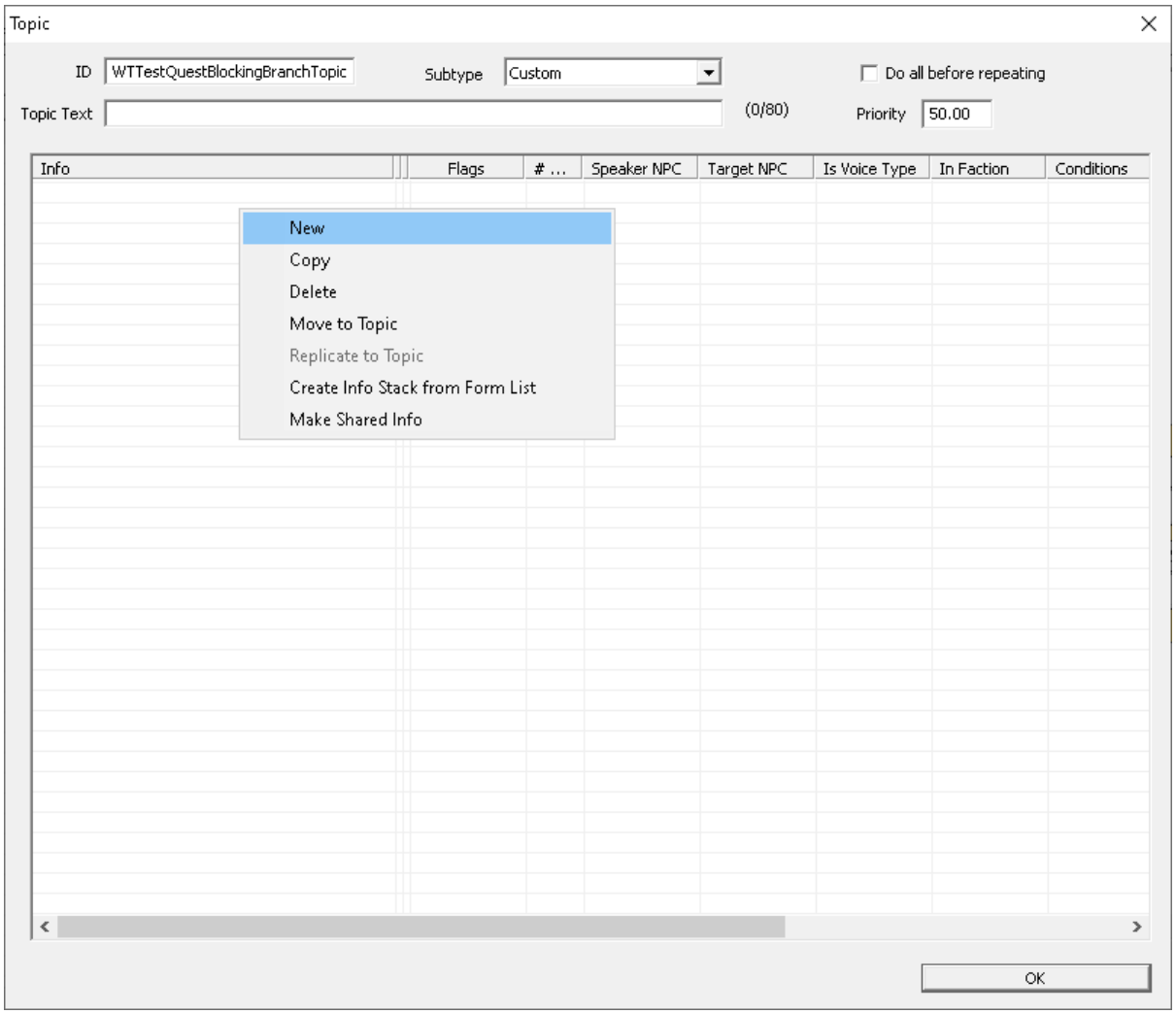


Figure 758 - Adding NPC response text.

Enter in the NPC's dialogue into the Response Text field then click OK.

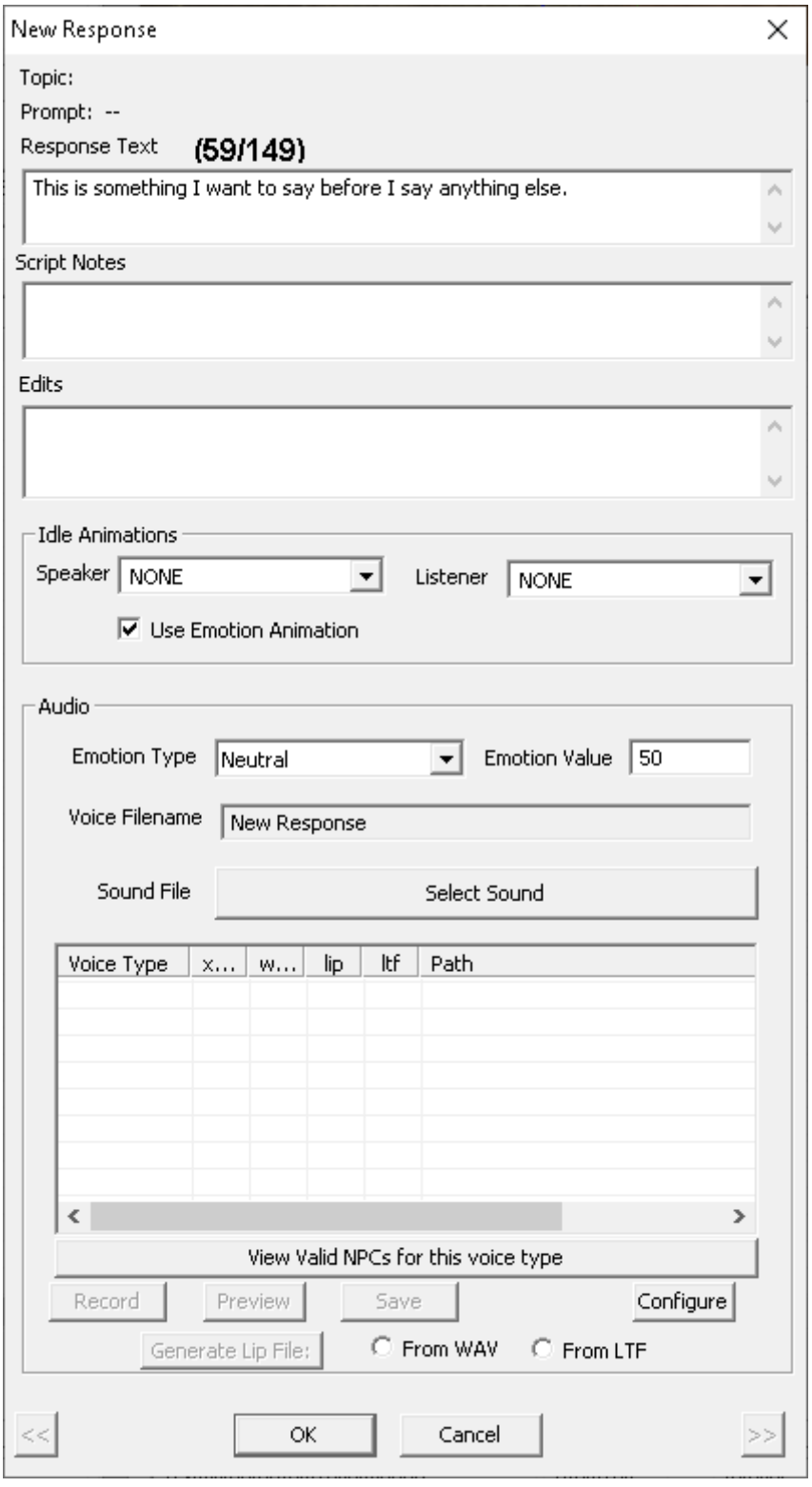


Figure 759 - Entering in the dialogue that we want the NPC.

Right-click in the Conditions list and select New.

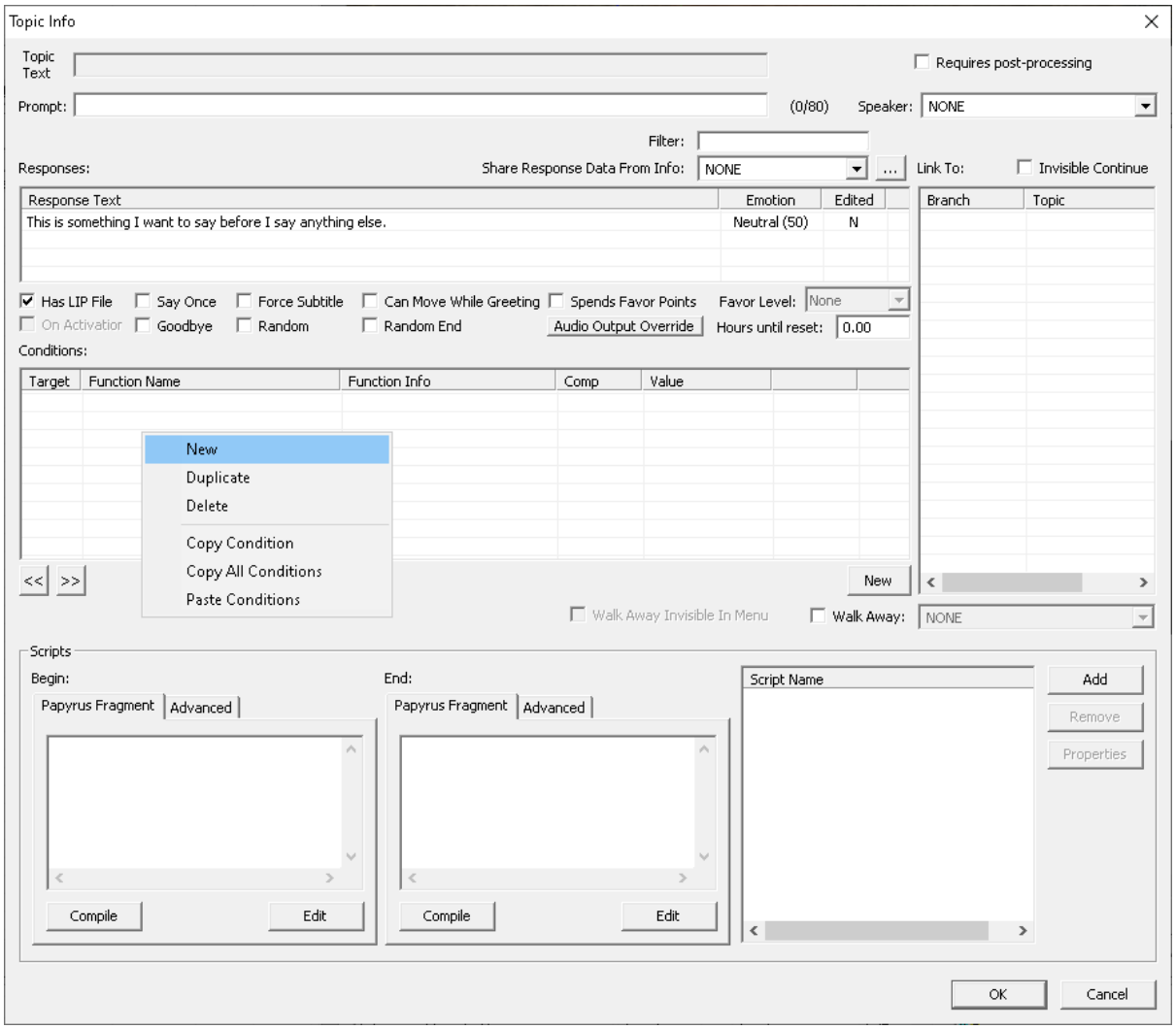


Figure 760 - Adding a new condition.

Ensure the Condition Function is set to GetIsID and set the actor. Again, for this example, that's going to be Athir.

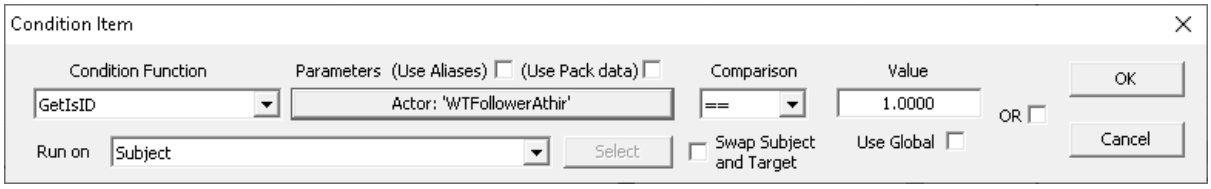


Figure 761 - Setting the GetIsID condition.

Click OK to close out of Condition Item.

Click OK to close out of Topic Info.

Click OK to close out of Topic.

Right-click in the branch and select 'Edit branch'.

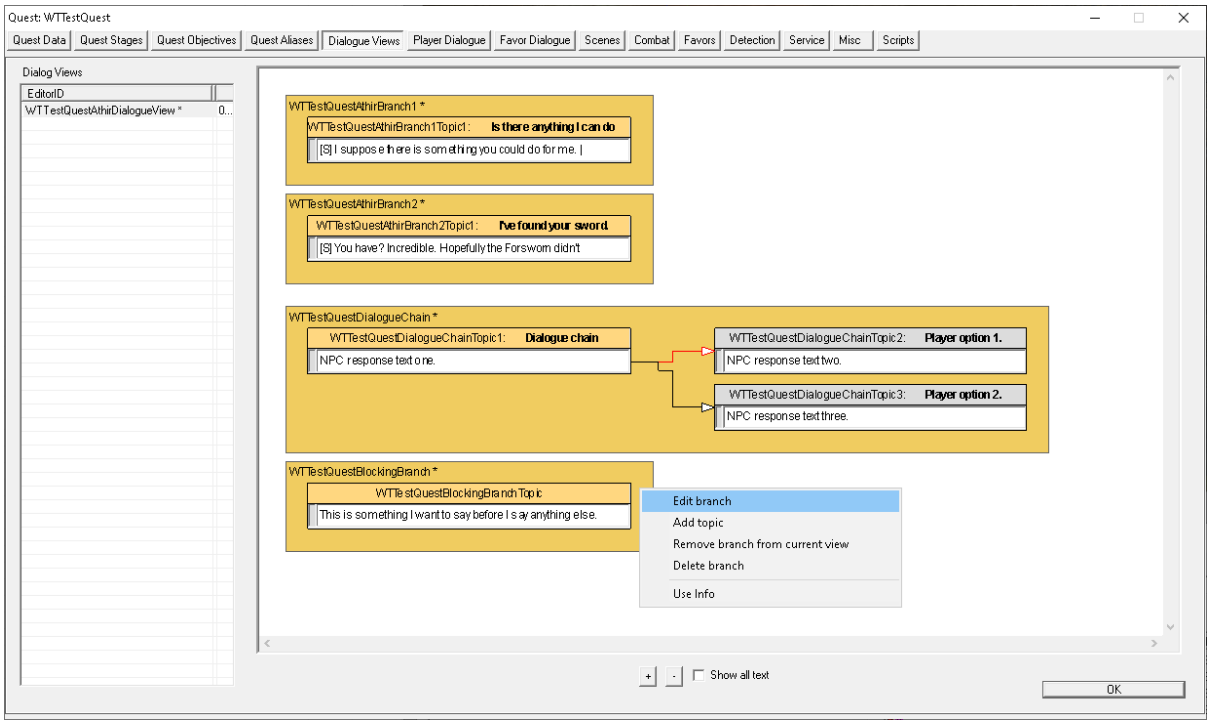


Figure 762 - Editing the branch properties.

Click on the Blocking radio button to set the branch up as a blocking branch, then click OK.

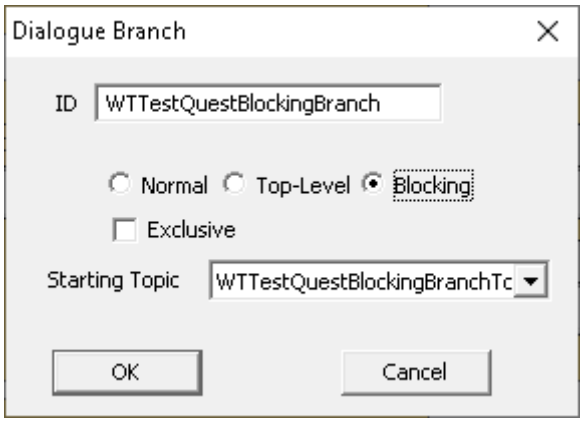


Figure 763 - Setting the branch from Top-Level to Blocking.

The colour of the branch should now change to green.

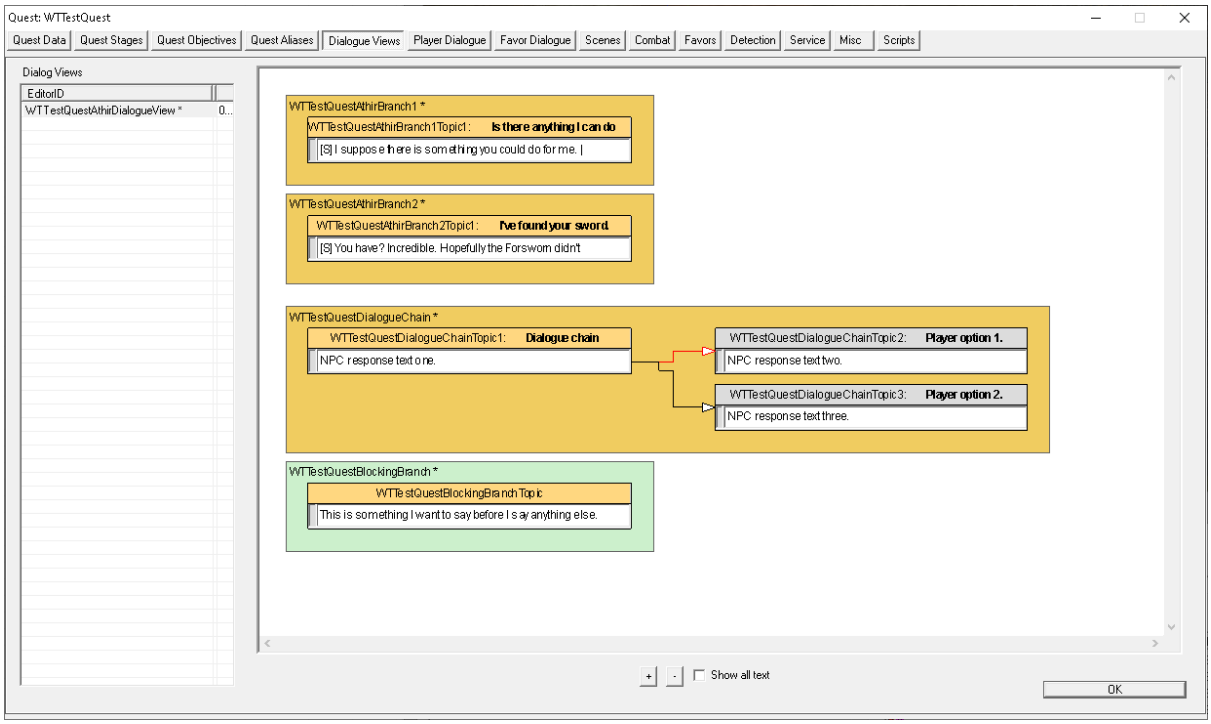


Figure 764 - The new branch set up as a blocking branch.

One caveat of using blocking branches is that they won't automatically flow on to the other branches we've defined, meaning we need to link them manually.

To make this easier, firstly move the blocking branch to the left of the other branches and topics.

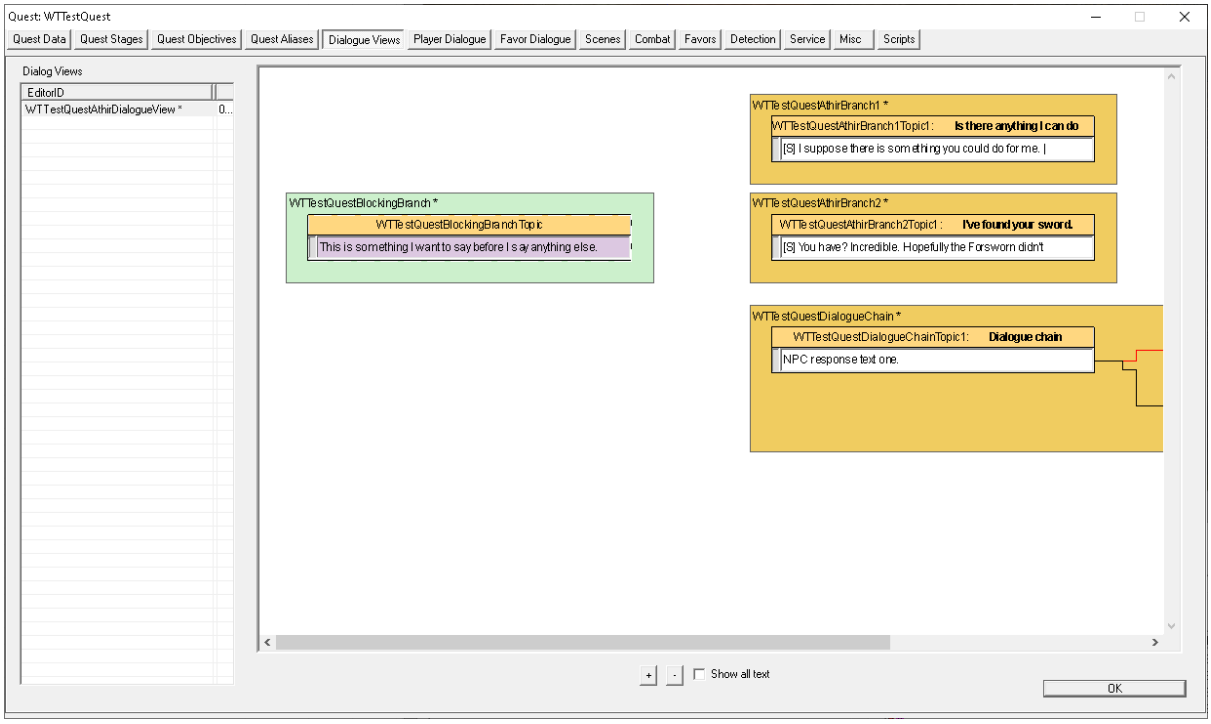


Figure 765 - Moving the blocking branch into position.

Left click on the topic response in the blocking branch's topic and drag and drop over to the other starting topics one at a time to link them all to the blocking branch's topic.

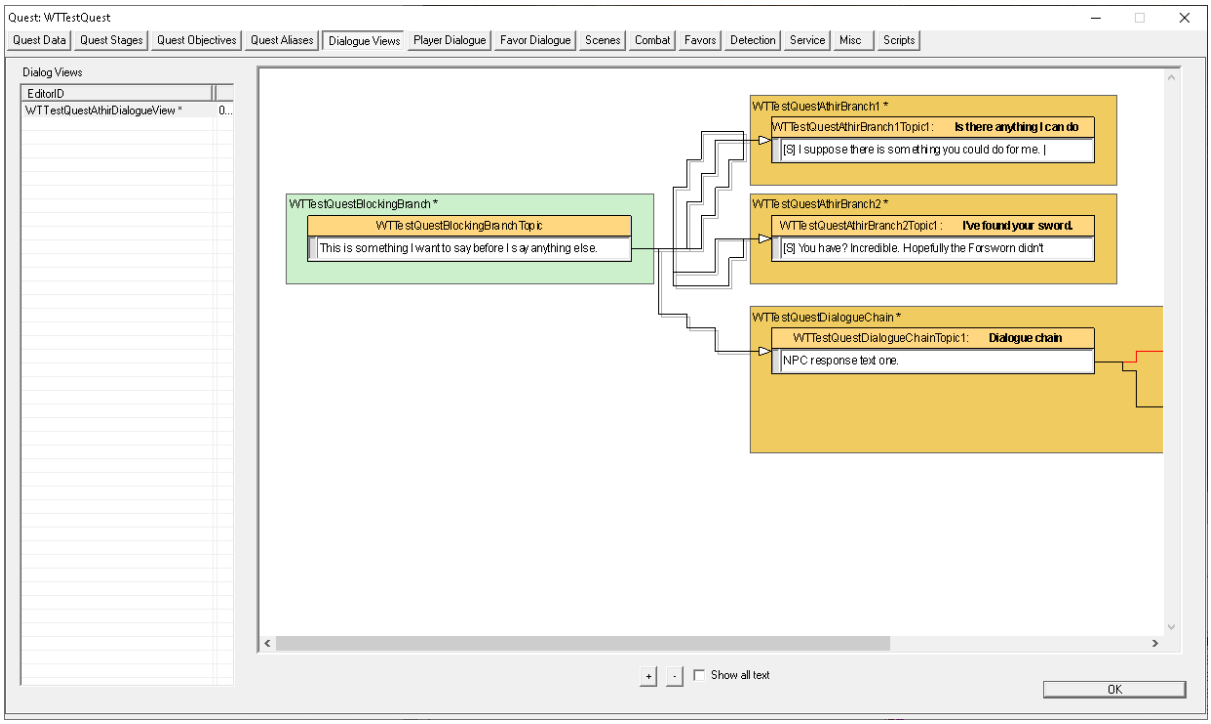


Figure 766 - The topic in the blocking branch linked to the other topics.

Note: The topic in the blocking branch does have priority over hello's, so you might want to add an extra condition to only use it in specific situations such as on a specific quest stage.

In the example below, I added a GetStage condition to ensure the topic in the blocking branch is only used when the current stage of WTTestQuest is 0.

Alternatively, if it's something you only want an NPC to only ever say once, tick the Say Once checkbox.

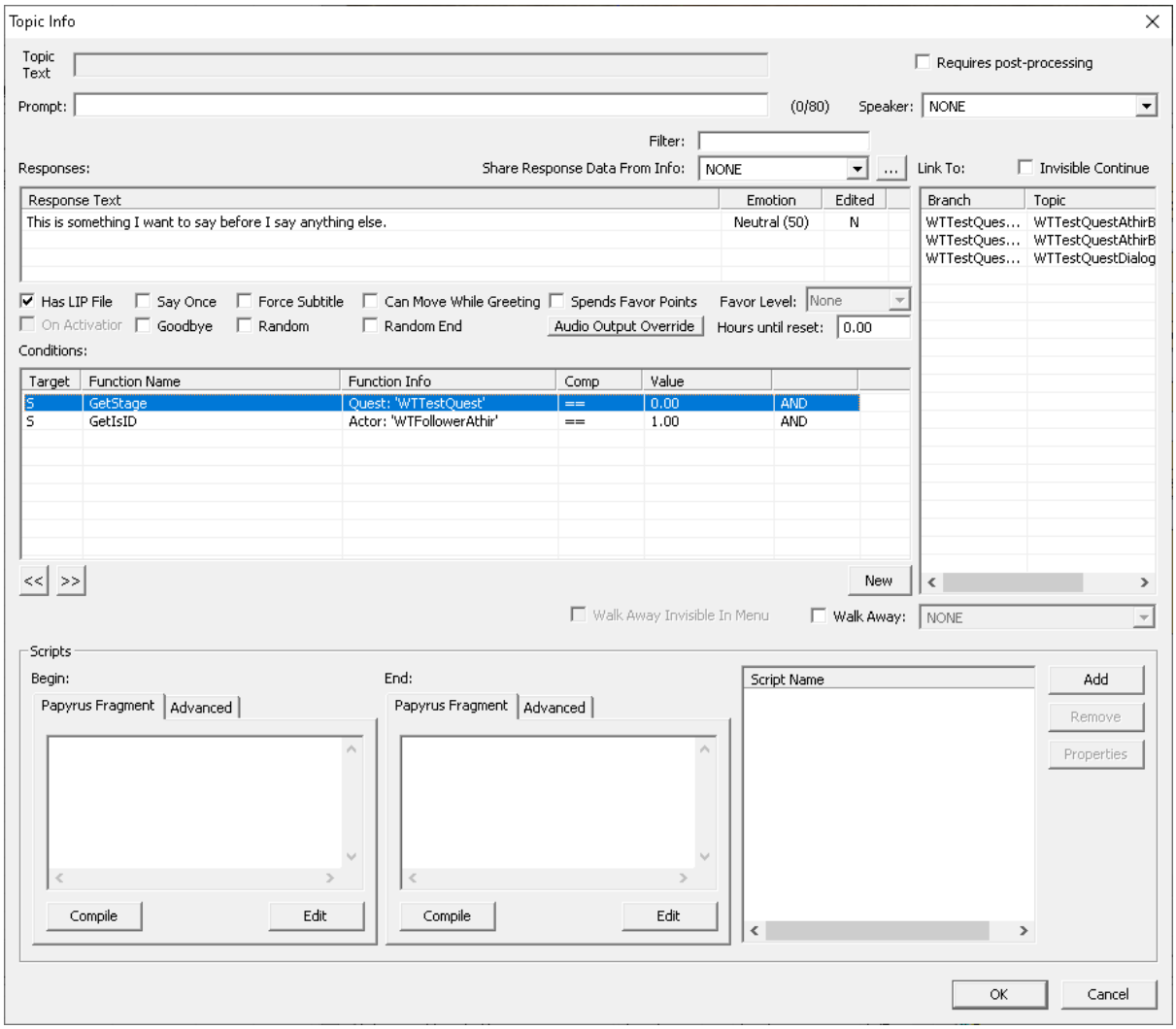


Figure 767 - Adding a getstage condition check to the blocking branch's topic.

GENERATING AN .SEQ FILE

Before we can test out our dialogue in-game, we need to generate or update an SEQ file.

Important: Every time you add a new quest containing new dialogue, you will need to generate a new SEQ file.

Click on the Save button in the Creation Kit to save these changes before continuing.

Open xEdit.

Ensure your mod is selected then click OK to begin loading.

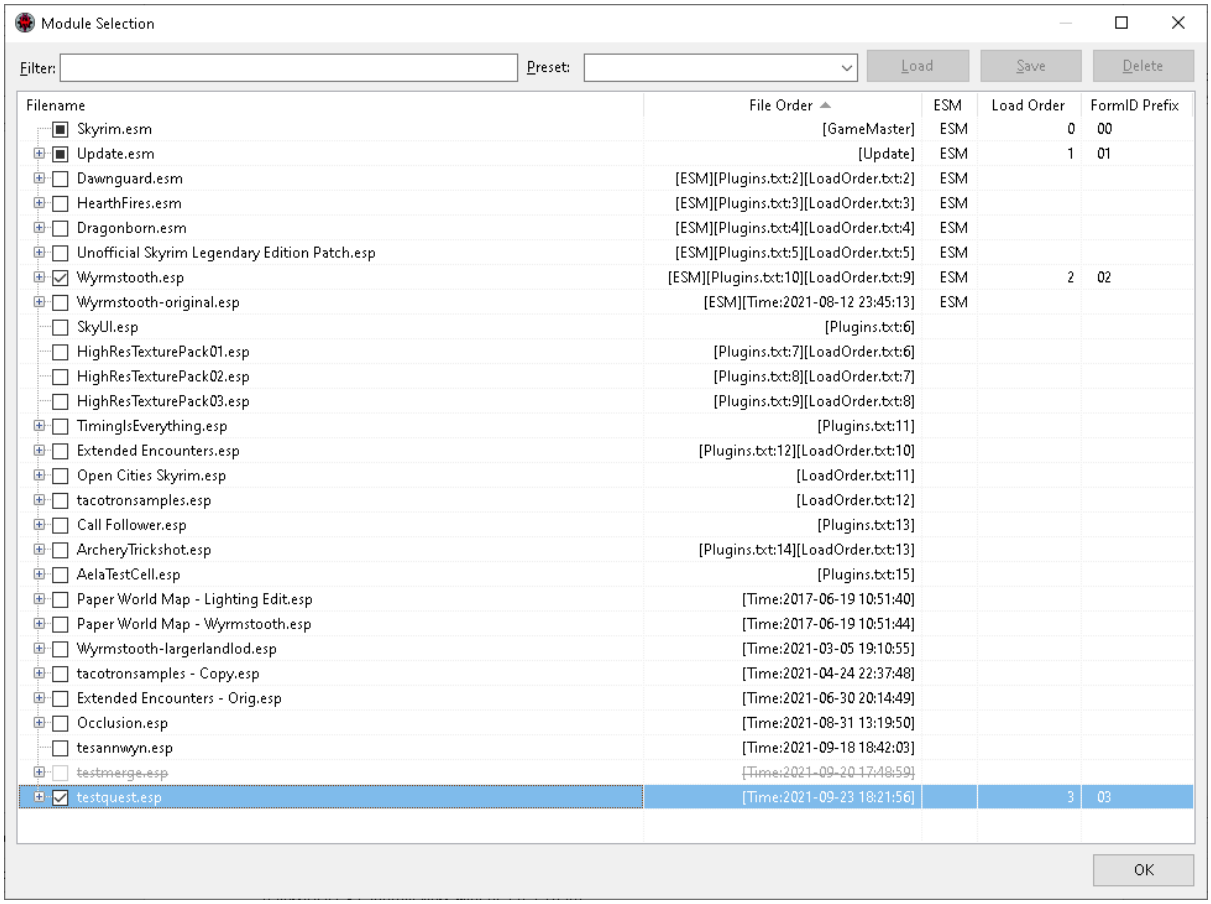


Figure 768 - Loading the mod in TES5Edit.

Right-click on your mod and select Other > Create SEQ File.

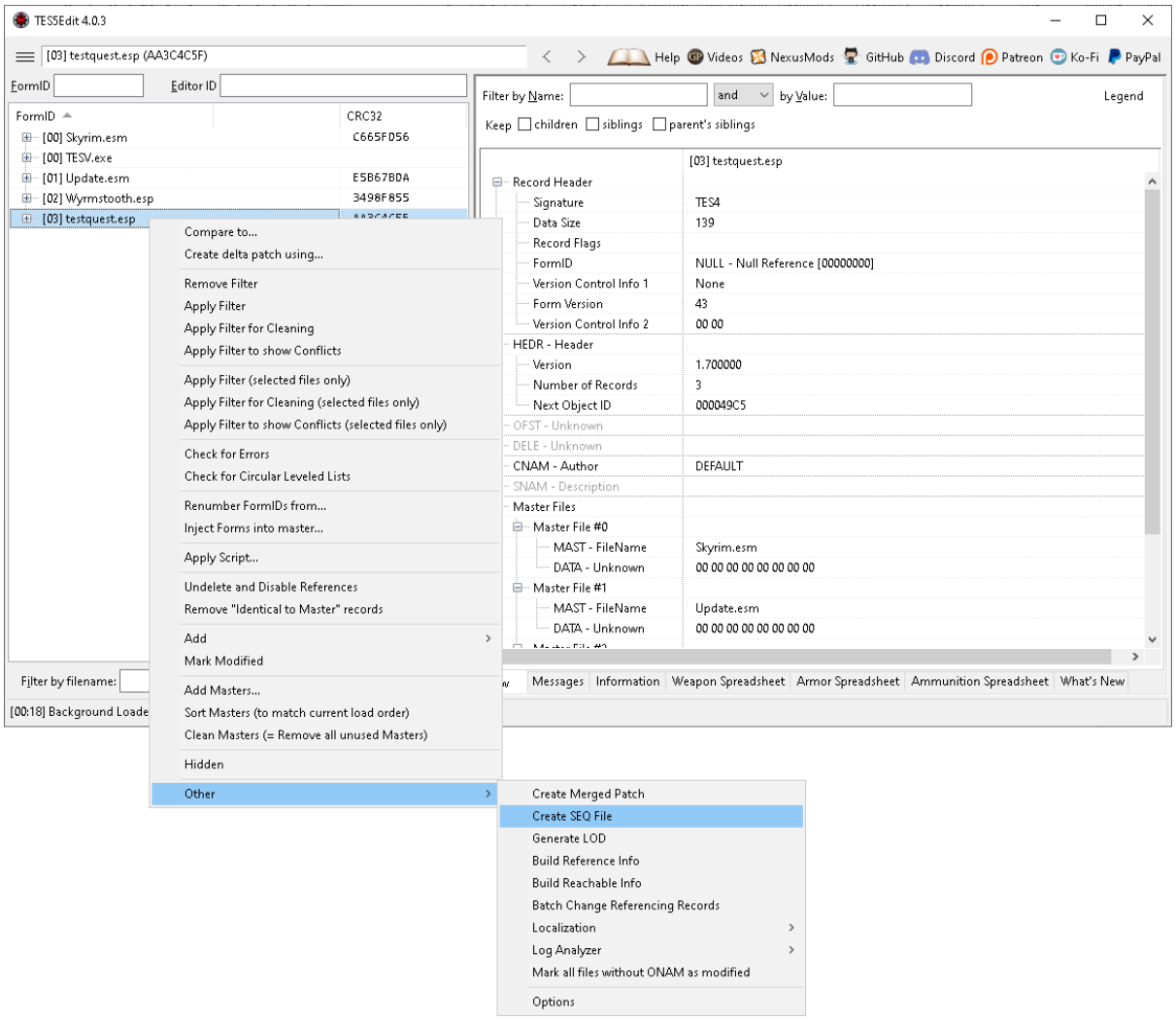


Figure 769 - Creating a new SEQ file.

This process should complete pretty quickly. Once it's done, you should see '[Create SEQ file done]' appear in the bottom left.

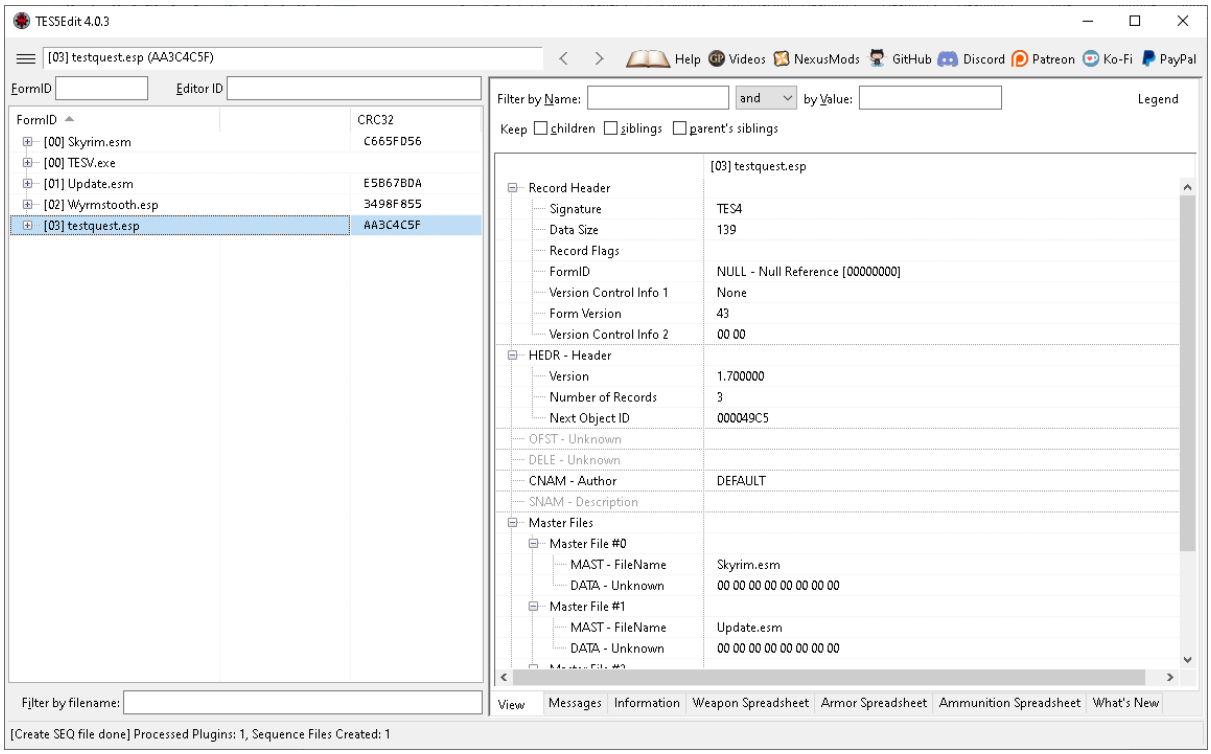


Figure 770 - SEQ file created.

To confirm the SEQ file has indeed been created, navigate to the Skyrim\Data\SEQ or Skyrim Special Edition Data\SEQ folder.

You should see an .seq file here with the same name as your .esp file.

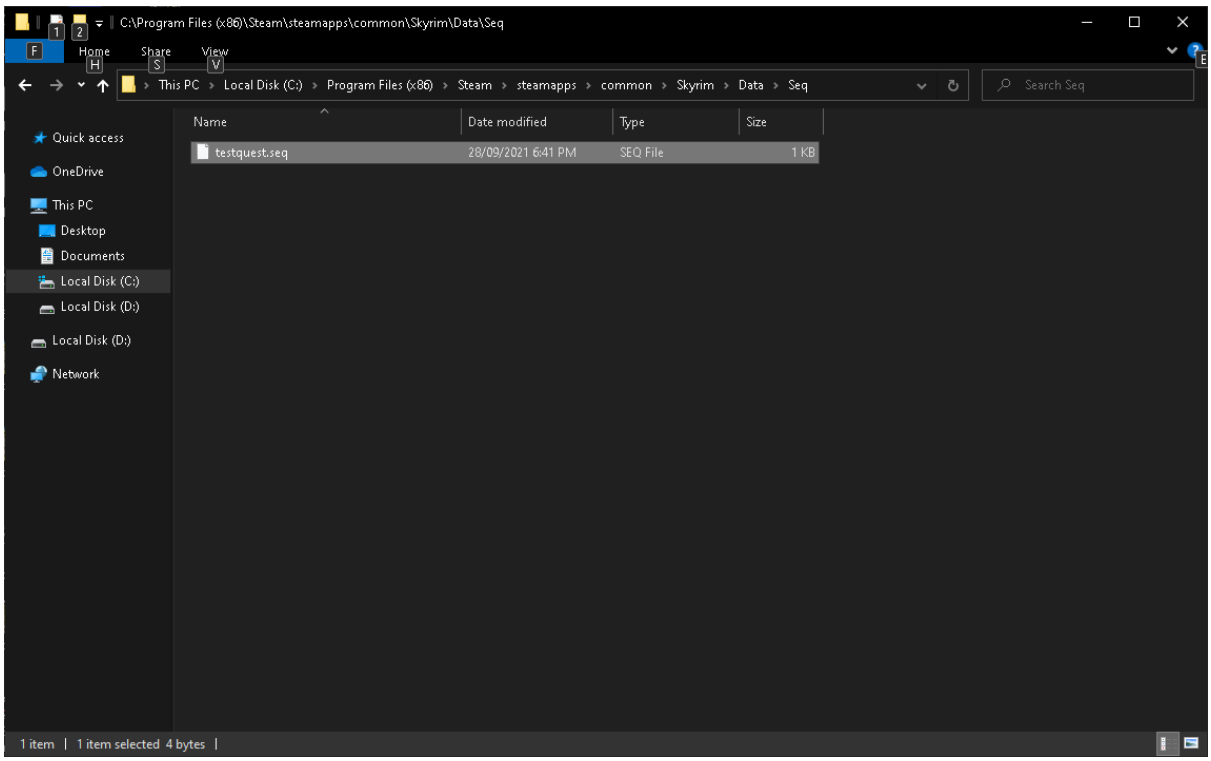


Figure 771 - SEQ file exists.

In the next section, I'll be covering the process of recording voice acting for your dialogue and generating .lip files for NPC lip-synching.

ADDING VOICE ACTING AND LIP SYNCHING

Now that we have some dialogue in our test quest, we'll need to add voice acting and lip synching.

Important: Voice acting needs to be recorded at 44.1 KHz and saved in the .wav format so we can process them later with Unfuzer.

Before passing your dialogue over to a voice actor, you will need to set up a spreadsheet containing a list of all lines spoken by an NPC. This spreadsheet should include notes for the voice actor to follow to help guide their performance.

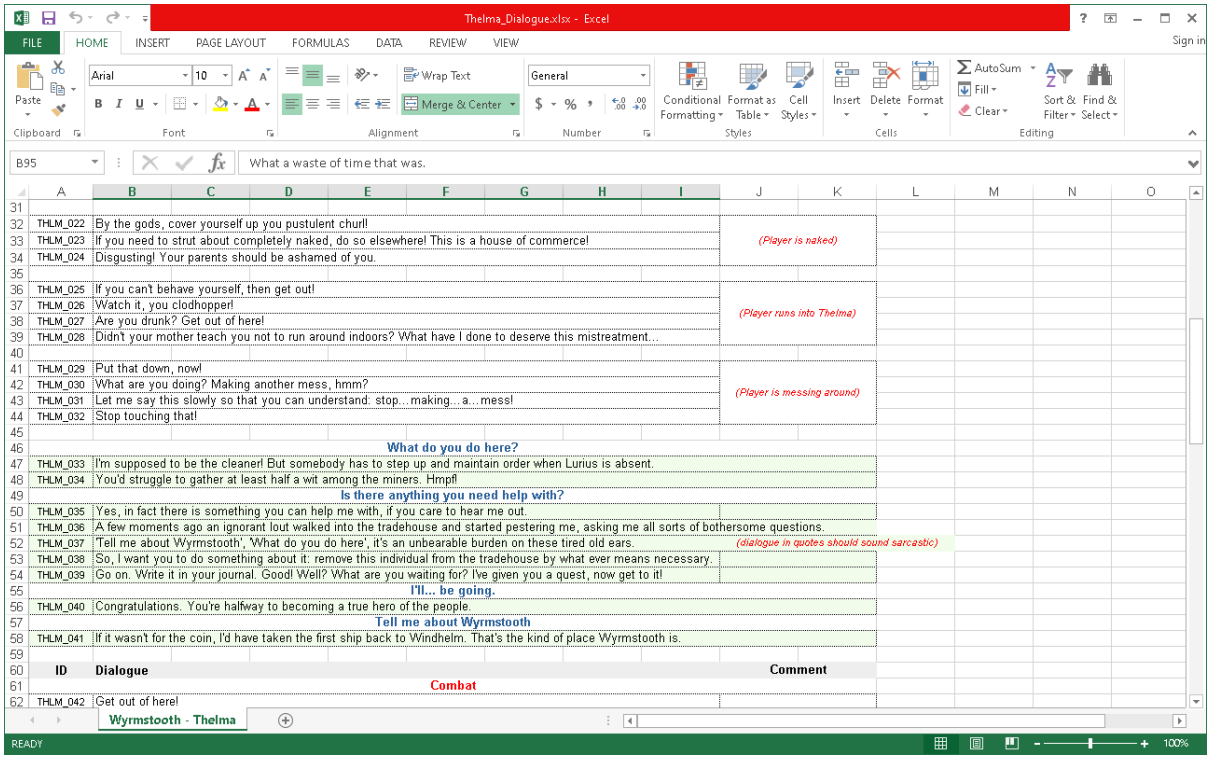


Figure 772 - Sample dialogue spreadsheet.

Once you've received the voice acting from your voice actor, browse to the Skyrim\Data\Sound\voice or Skyrim Special Edition\Data\Sound\voice folder.

Create a new folder named after your mod. For this test quest example, that'll be 'testquest.esp'.

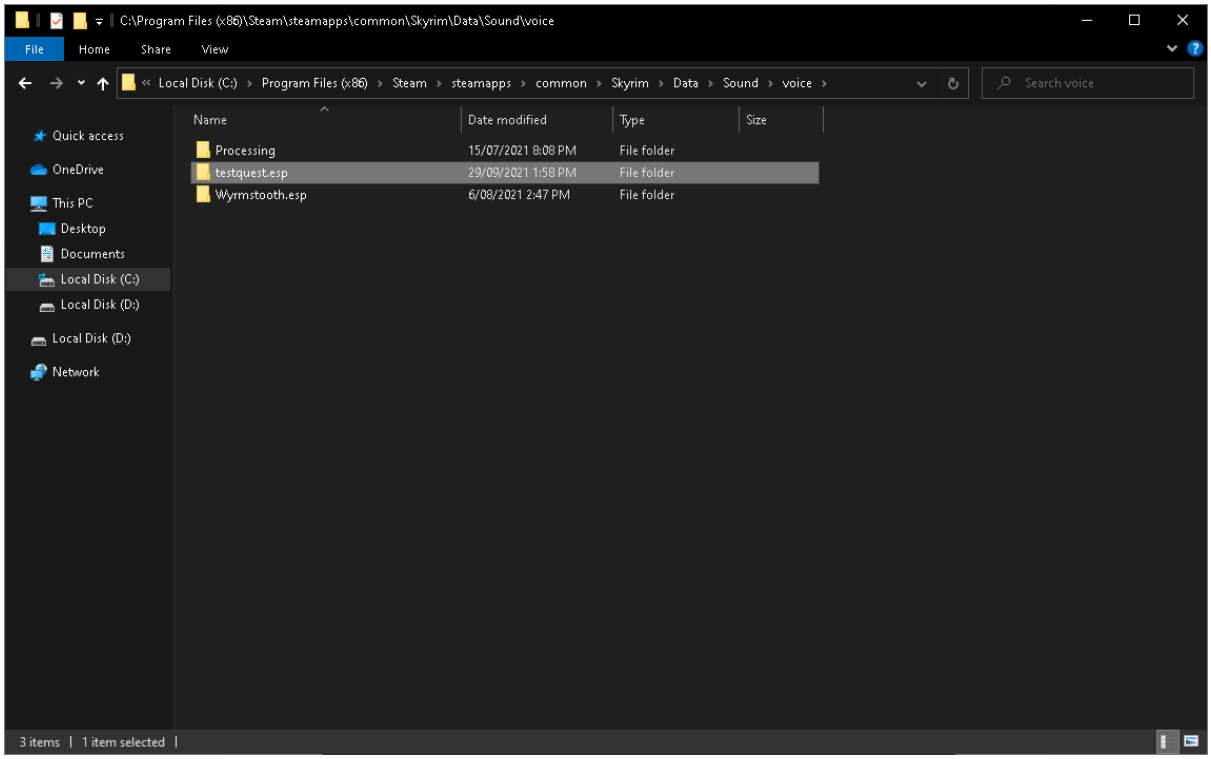


Figure 773 - Voice Type folders.

Go into that folder and create a new folder named after the voice type assigned to the NPC.

For example, if the NPC uses the MaleNord voice type, name that folder MaleNord.

In my example, Athir is from the Wyrmstooth mod and uses the WTVoiceType voice type, so I created a folder called 'WTVoiceType'.

Note: Ideally, you should be using a custom voice type for NPCs that have their own voice acting. Players may find it a bit jarring if an NPC's voice actor keeps changing.

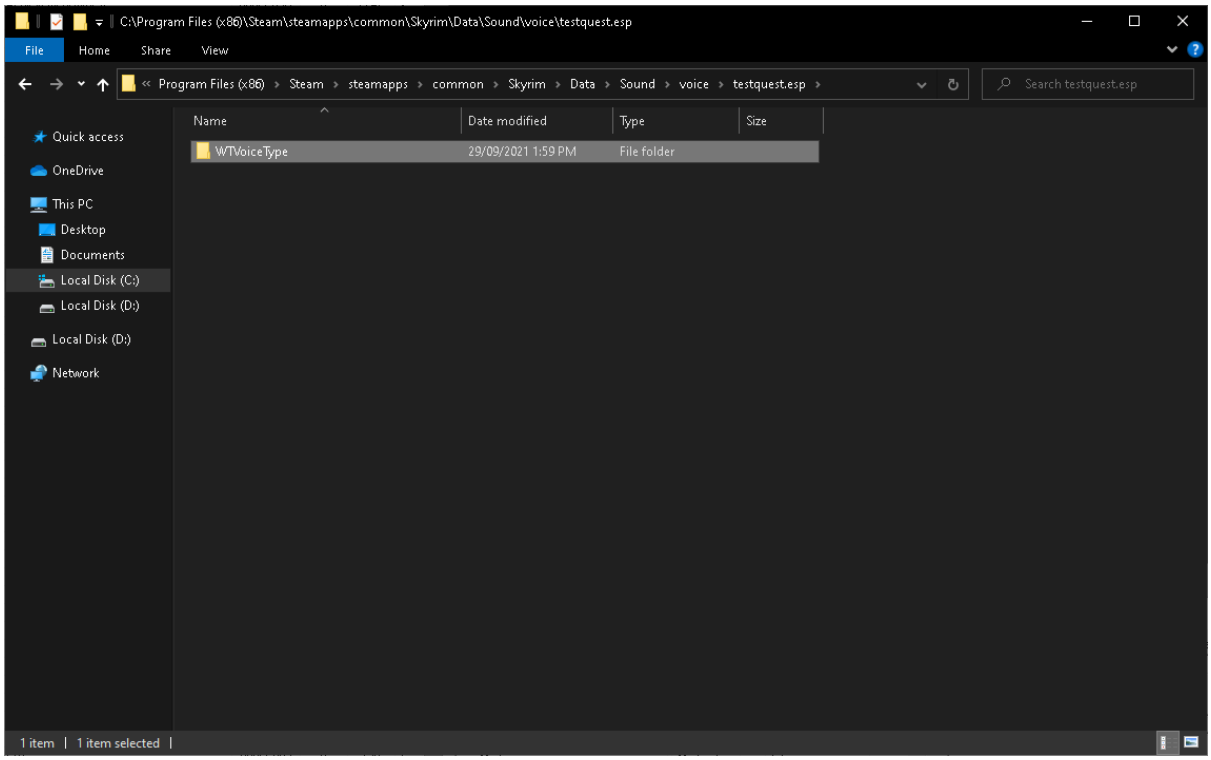


Figure 774 - Creating a voice type folder.

Copy your voice acting files into that folder.

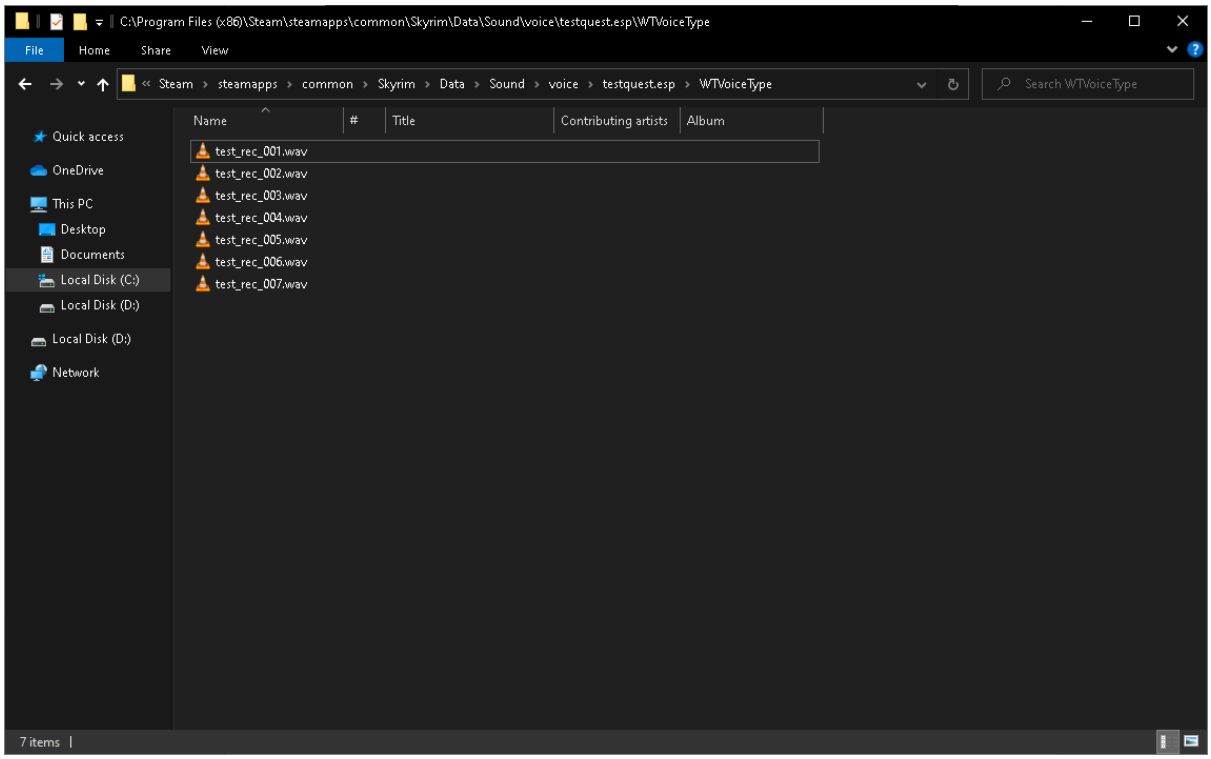


Figure 775 - Folder containing voice acting files.

Next, we need to rename these .wav files to match the internal naming scheme.

In the Quest properties, go to the Dialogue Views tab and double-click on the response text below a topic to open it.

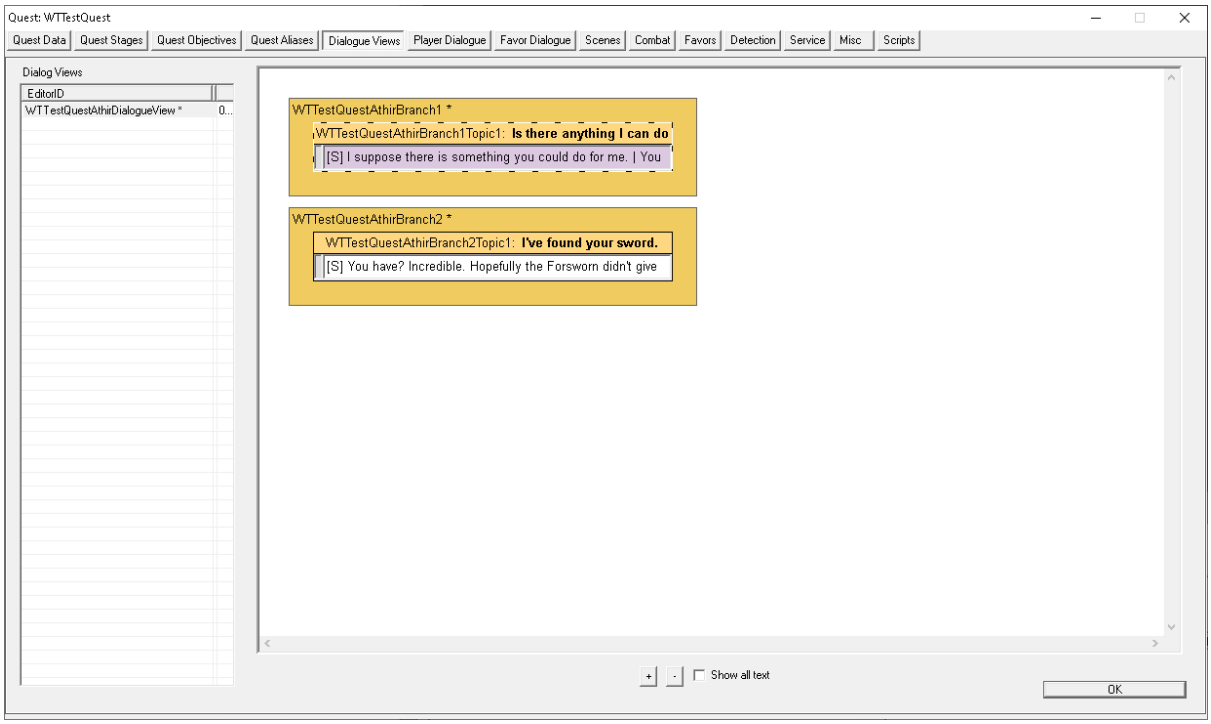


Figure 776 - Dialogue Views tab.

Double-click on the first response to open its properties.

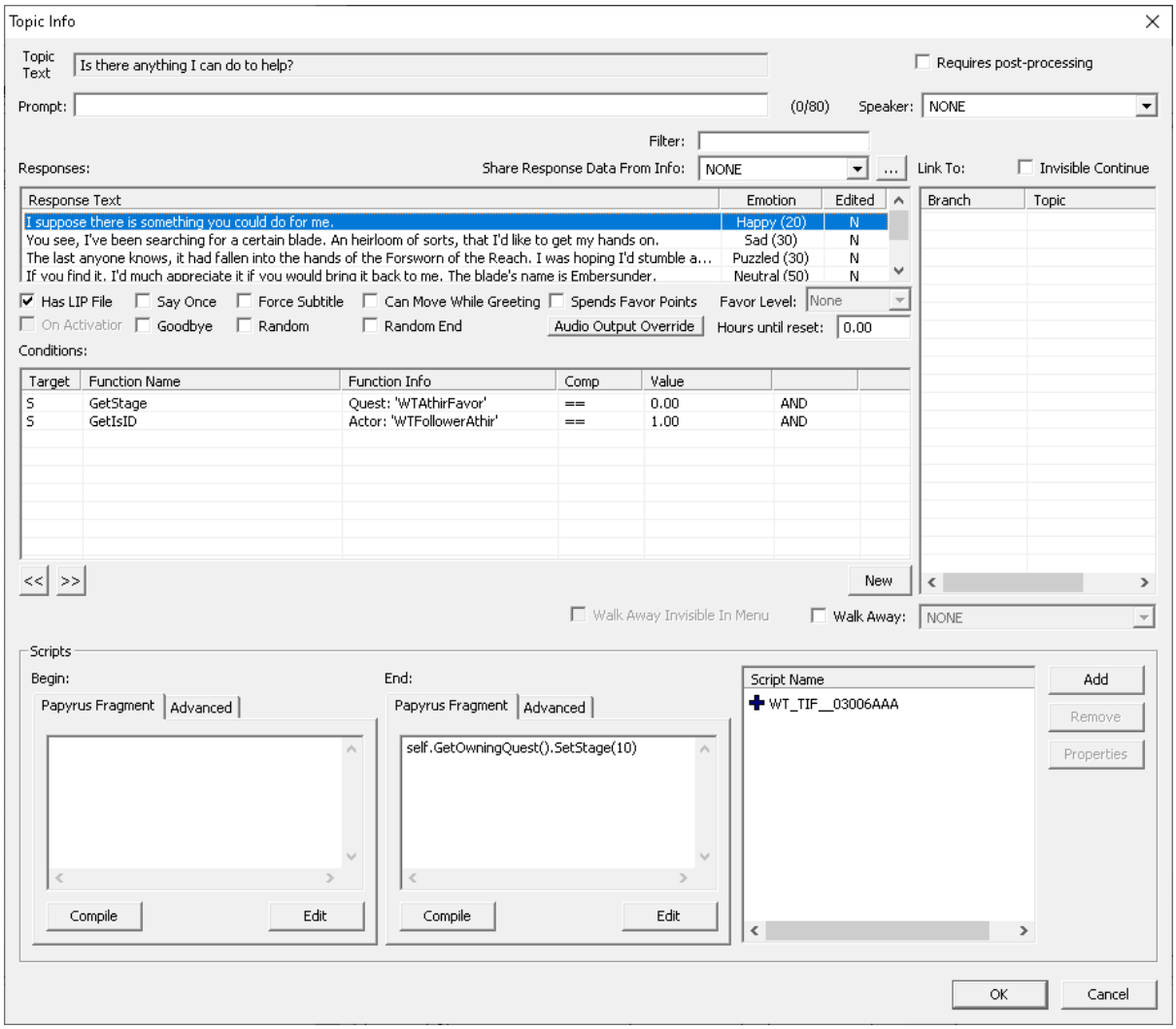


Figure 777 - Topic Info.

The file name for this line appears in the Voice Filename field. Select this string and press CTRL + C to copy it.

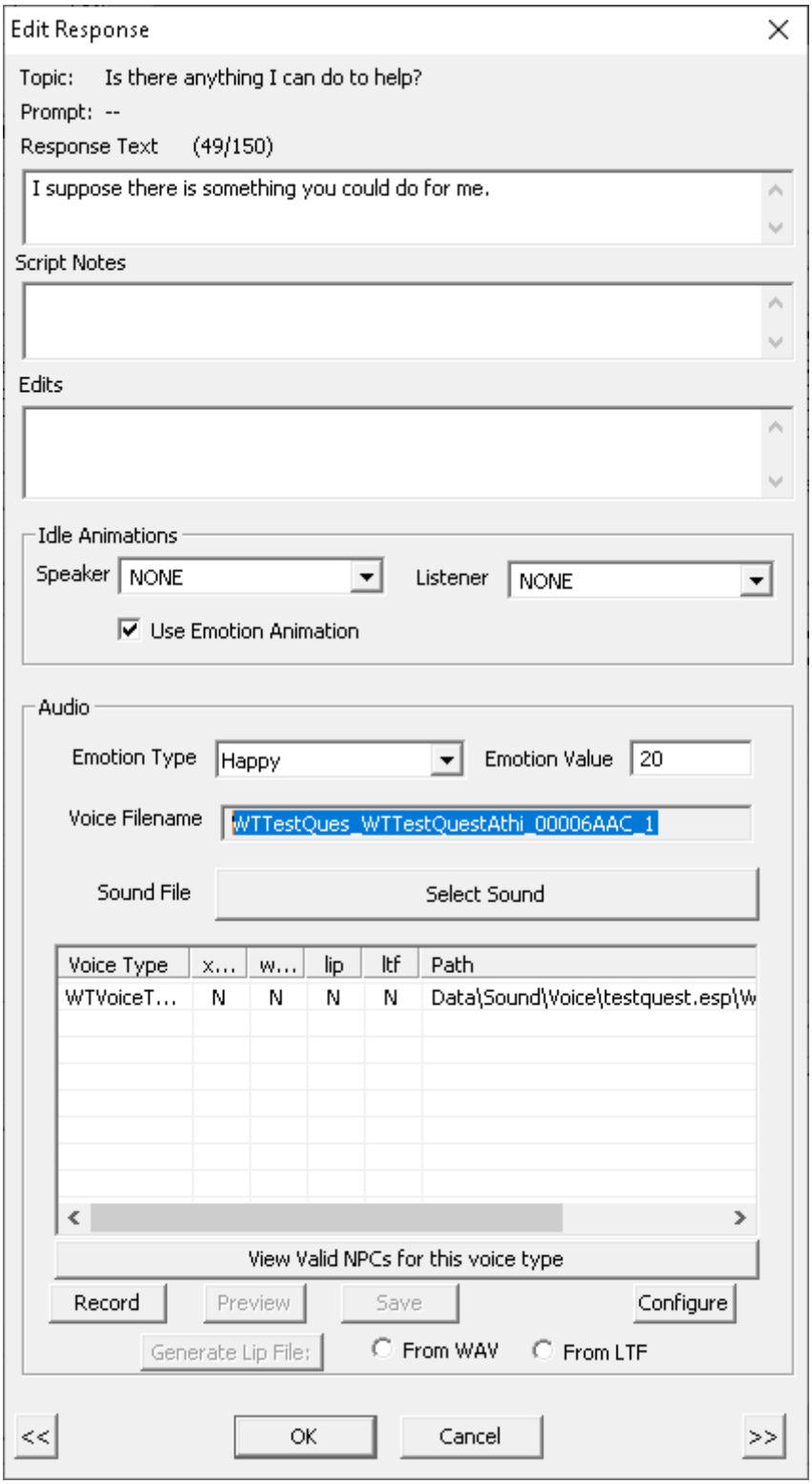


Figure 778 - Voice Filename.

Go to the folder containing the voice files and rename the corresponding .wav file for this line.

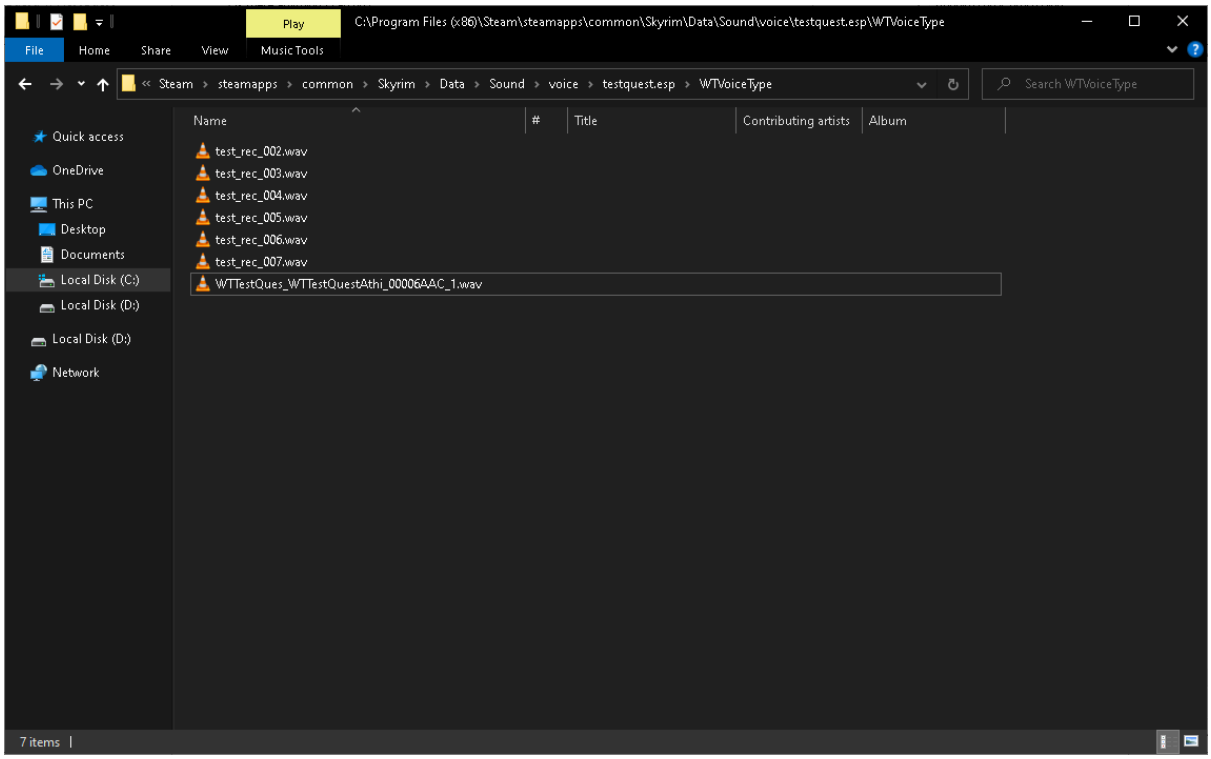


Figure 779 - Renamed the .wav file.

Repeat these steps to rename the rest of the .wav files for this topic.

The filenames are sequential, so if the Voice Filename for the first line is WTTestQues_WTTestQuestAthi_00006AAC_1, the second line will be WTTestQues_WTTestQuestAthi_00006AAC_2 then WTTestQues_WTTestQuestAthi_00006AAC_3 and so on.

Repeat these steps to rename the .wav files for the responses from the other topic too.

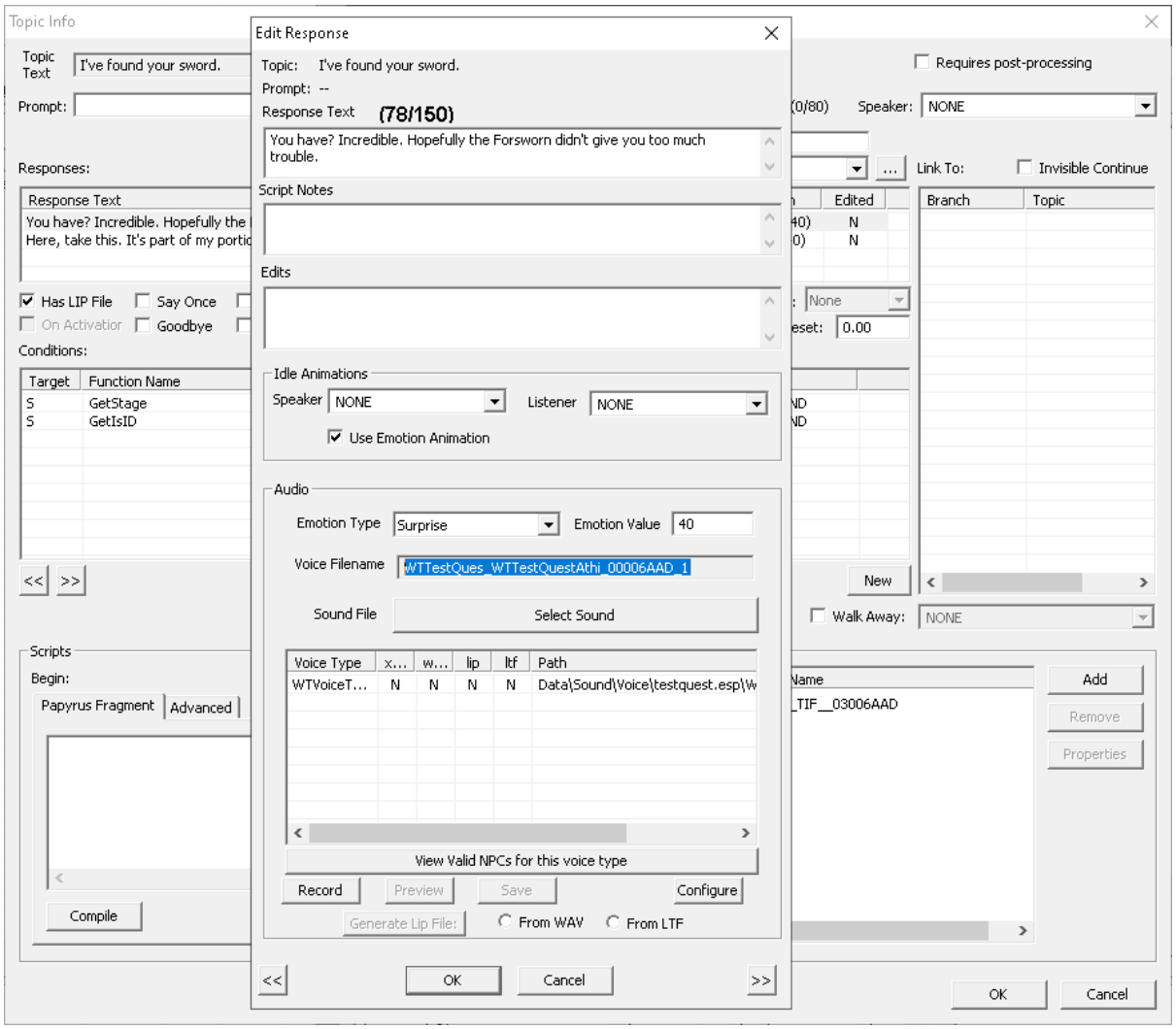


Figure 780 - Getting the Voice Filename for the other topic.

All the .wav files should now have been renamed like they have in the following screenshot:

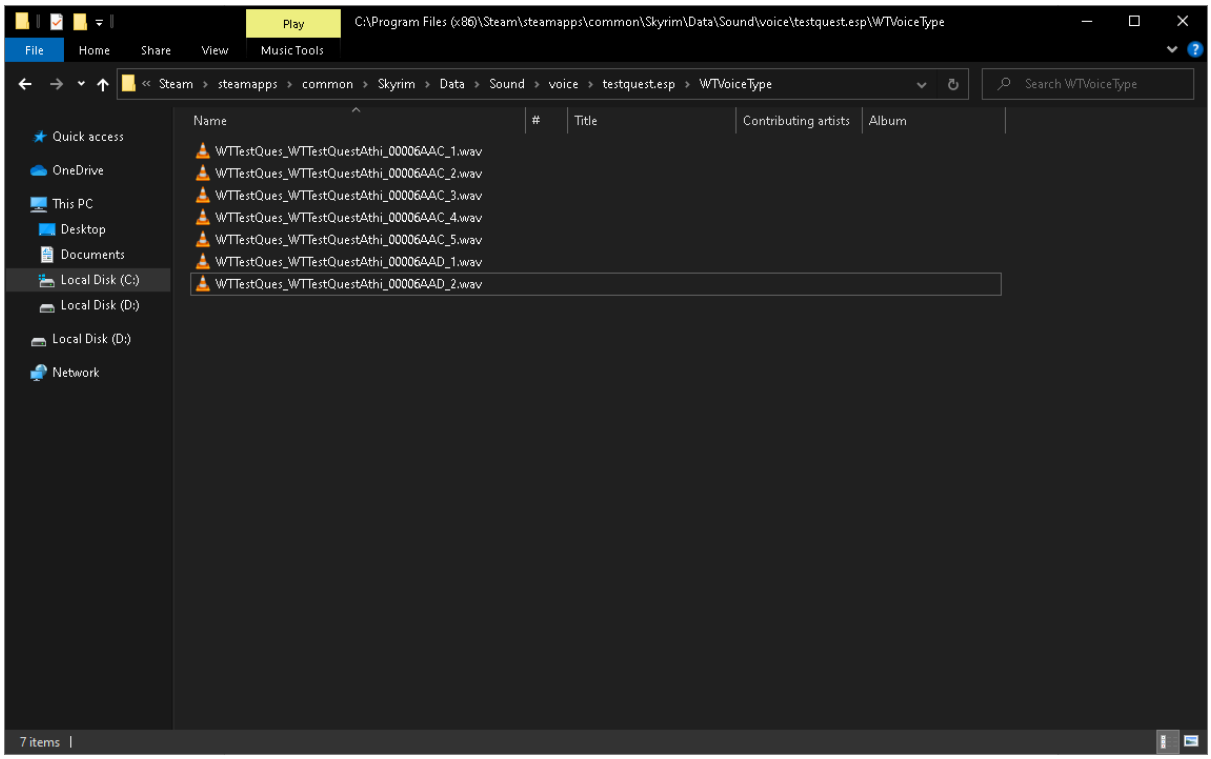


Figure 781 - .wav files renamed.

Next we need to generate the lip sync data. The lip sync data we generate will be stored in .lip files.

When the process is complete, we should have a .lip file for each .wav file in the voice type folder.

Go back to the first line of the first topic.

The 'Generate Lip File:' button will be greyed out until you select the .wav file in the list above. Once selected, the 'Generate Lip File:' button should un-grey itself if the corresponding .wav file can be found in the voice type folder.

Click on the 'Generate Lip File:' button to generate the .lip file.

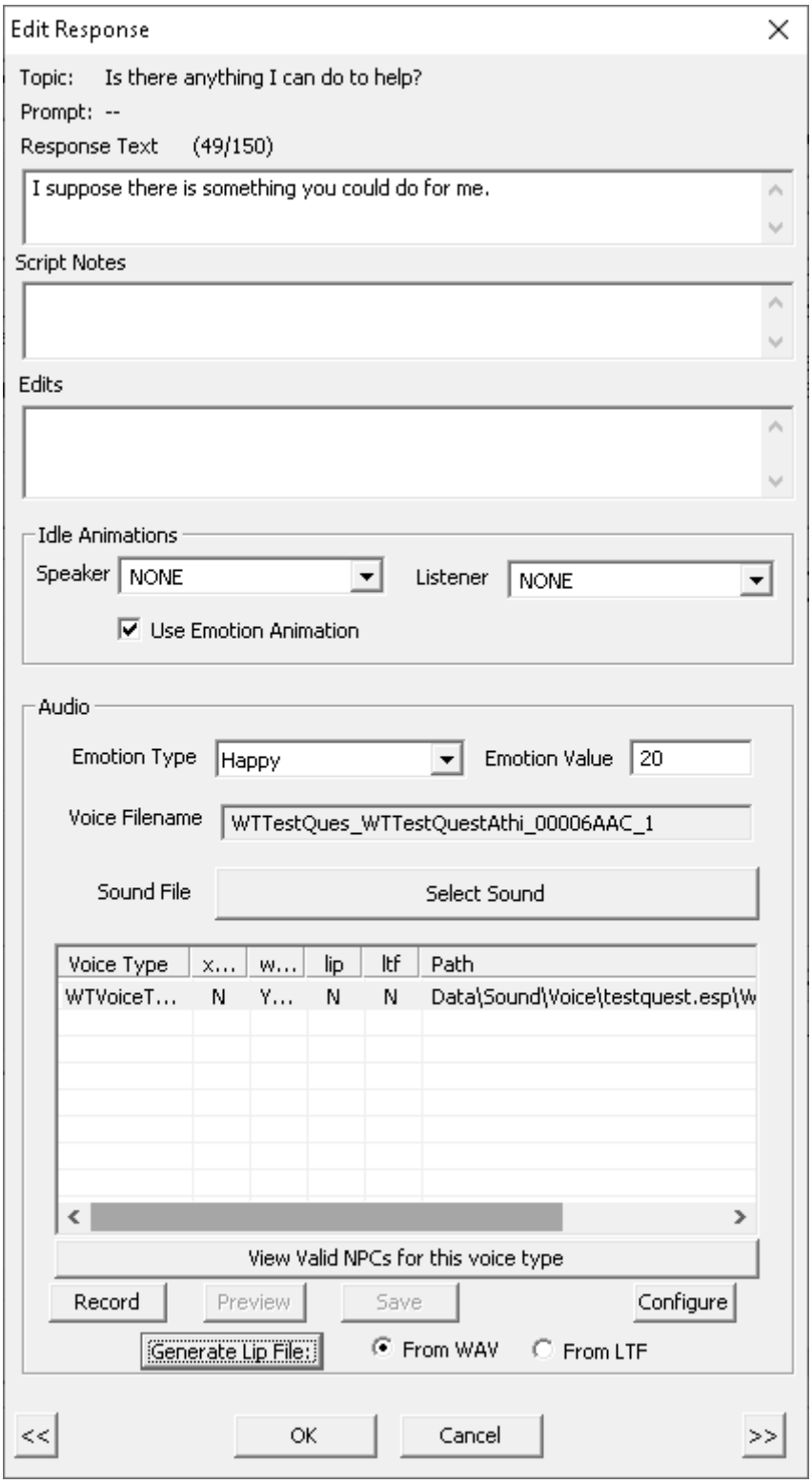


Figure 782 - Generating a .lip file.

A .lip file should appear in the voice type folder.

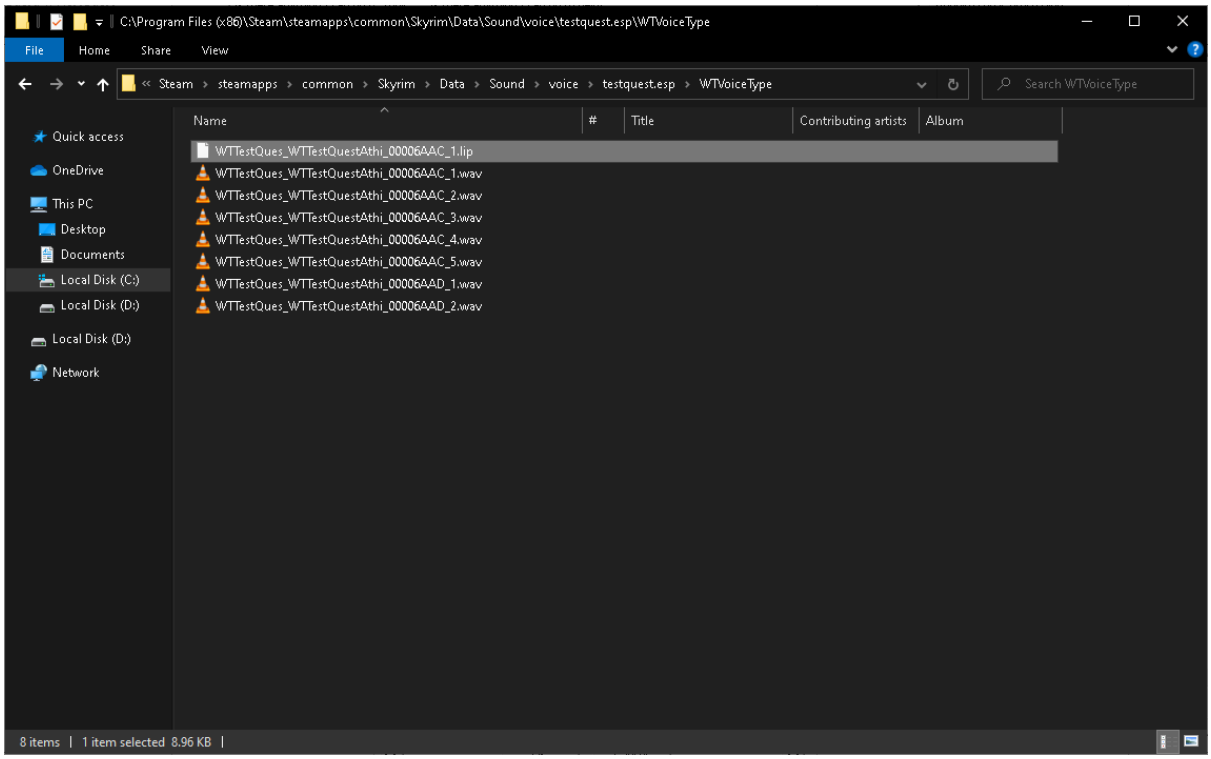


Figure 783 - .lip file created successfully.

Repeat these steps to generate a .lip file for the other responses in this topic.

Repeat these steps to generate .lip files for the responses in the other topic too.

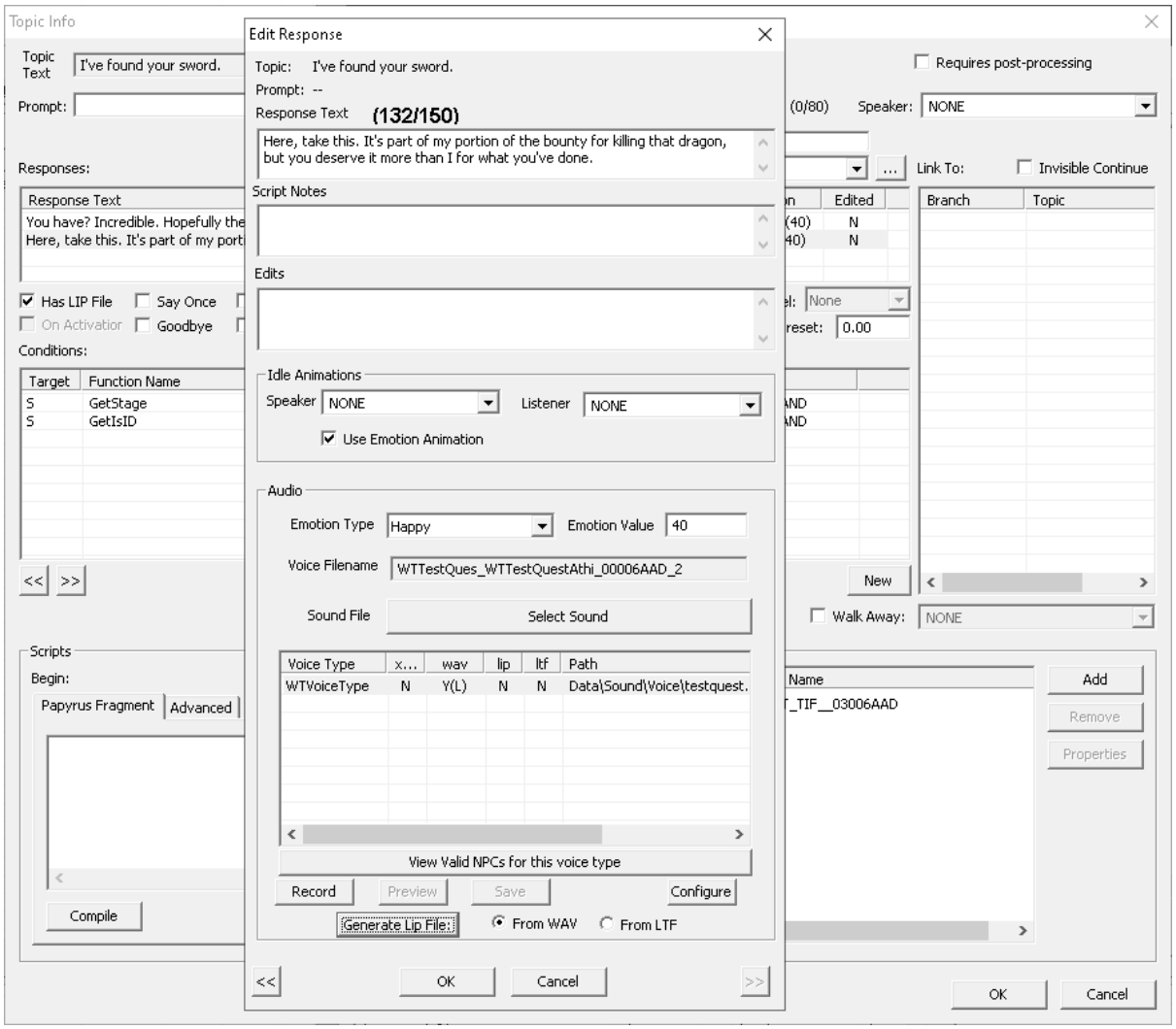


Figure 784 - Generating .lip files for the responses in the second topic.

We should now have a .lip file for each .wav file in the voice type folder.

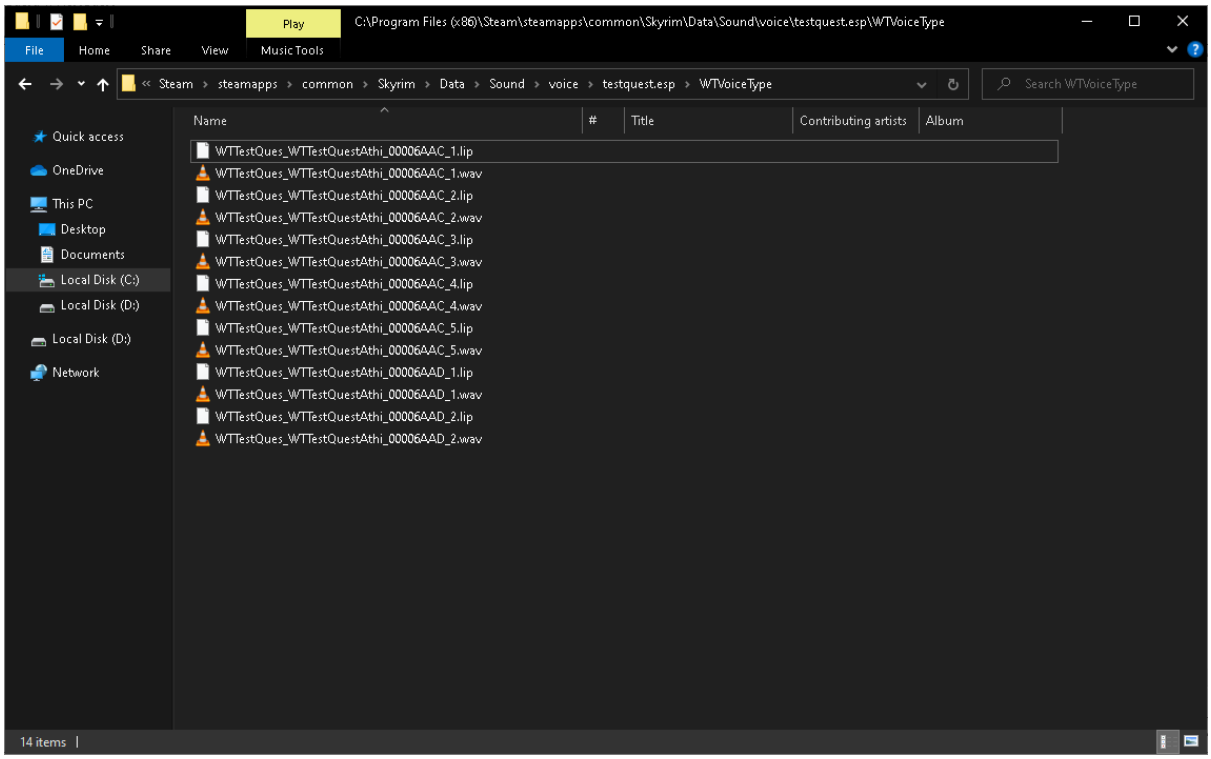


Figure 785 - A .lip file generated for each .wav file.

Lastly, we need to pack the pairs of .lip and .wav files together in a .fuz file. We'll need the Unfuzer utility which can be downloaded from Nexusmods.

After extracting Unfuzer to a new folder, launch it by running Unfuzer.exe.

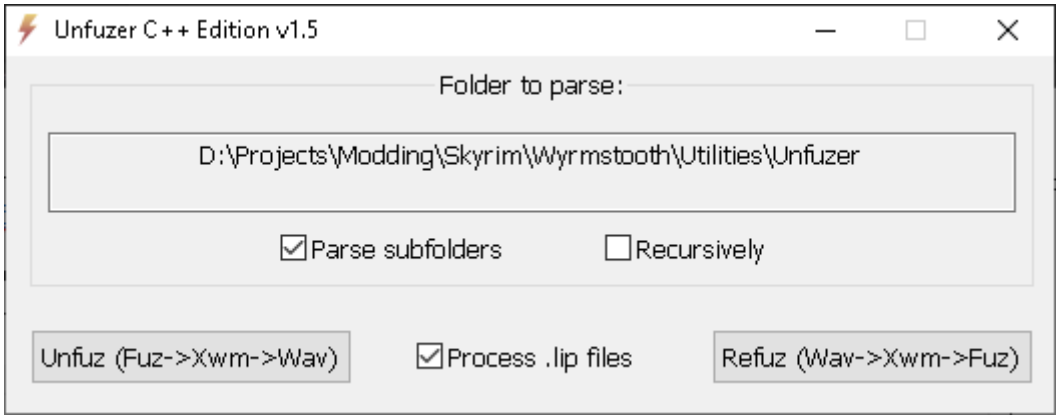


Figure 786 - Unfuzer.

The first thing we need to do is change the path to point to our voice type directory.

Click on the path beneath 'Folder to parse:'.

Select the folder containing the voice type folder. In our example, that's 'C:\Program Files (x86)\Steam\steamapps\common\Skyrim\Data\Sound\voice\testquest.esp\WTVoiceType'.

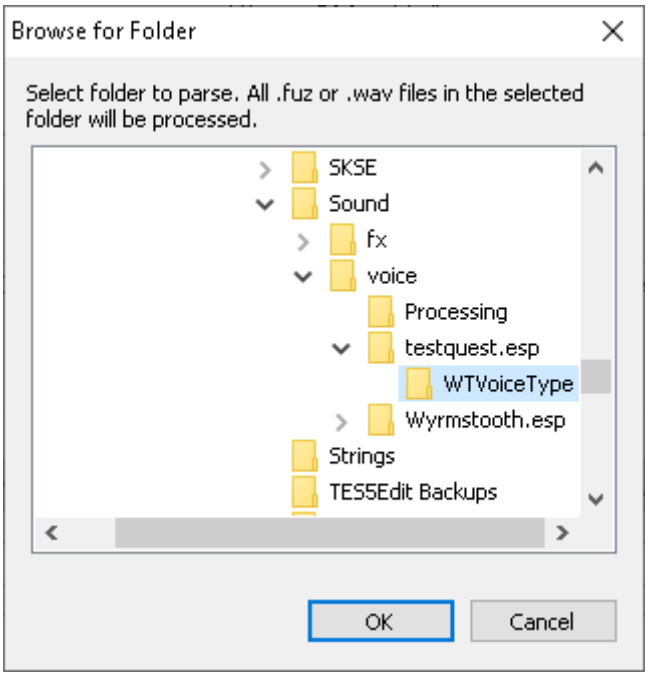


Figure 787 - Selecting the voice type folder.

Click OK.

There are no subfolders so we can untick 'Parse subfolders'.

Ensure 'Process .lip files' is ticked.

Click 'Refuz (Wav->Xwm->Fuz)' to begin the process.

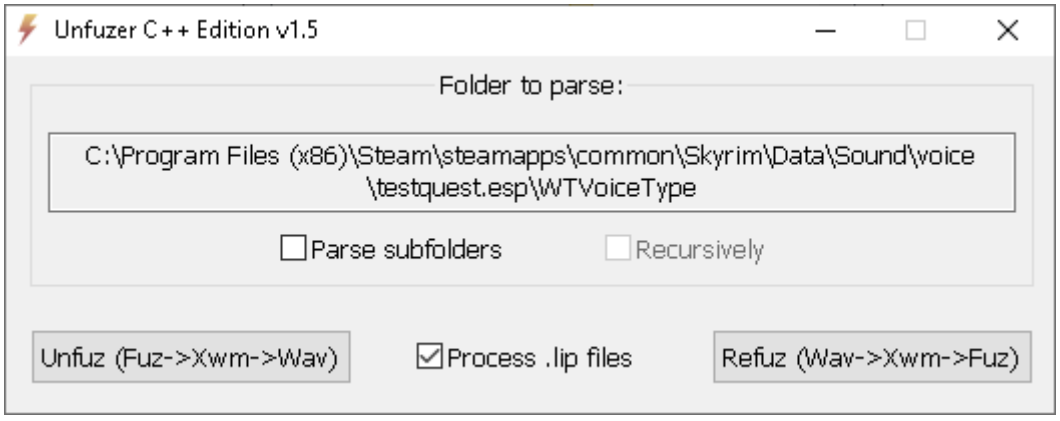


Figure 788 - Ready to fuz.

Go back to the voice type folder. The .wav and .lip files should now be gone and in their place you should see a .fuz file for each pair of .wav and .lip files.

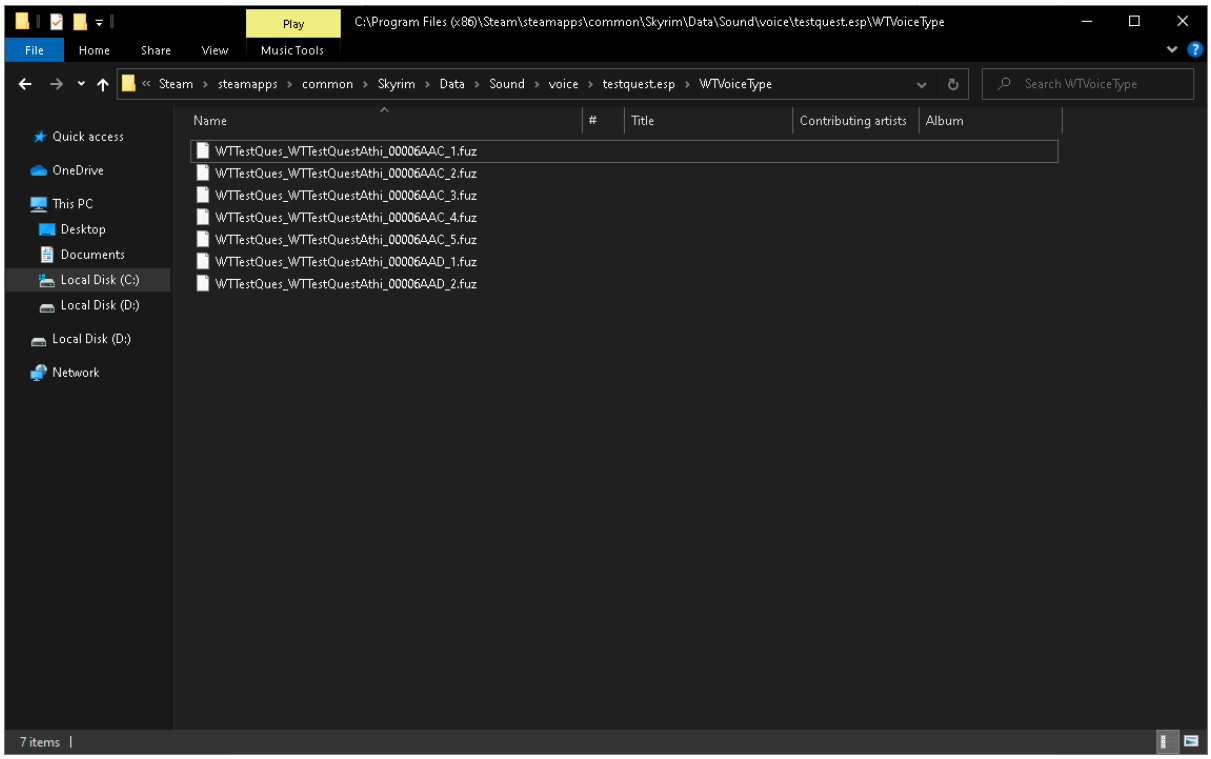


Figure 789 - The .wav and .fuz files fuzed together.

The last thing left to do is to check out our dialogue in-game to confirm it works.



Figure 790 - The NPC saying their response to the topic selected by the player.

ADDING HELLO, COMBAT, CRIME AND OTHER INCIDENTAL DIALOGUE

This section will cover adding incidental dialogue such as hellos, combat barks, trespass, theft, and pickpocket alerts, and so on. Most of these topics are set up in basically the same manner, so I'll cover a few examples which should give you enough of an idea.

Important: Incidental dialogue should go into a quest with 'Start Game Enabled' ticked.

To start things off, let's create a new quest. See the section The basics of setting up quests for instructions on how to do this. For this example, I named the quest DialogueGeneric with a 'WT' prefix.

Note: To prevent reanimated NPCs from using these lines, you can add a 'IsCommandedActor == 0' condition to the quest as per the screenshot below:

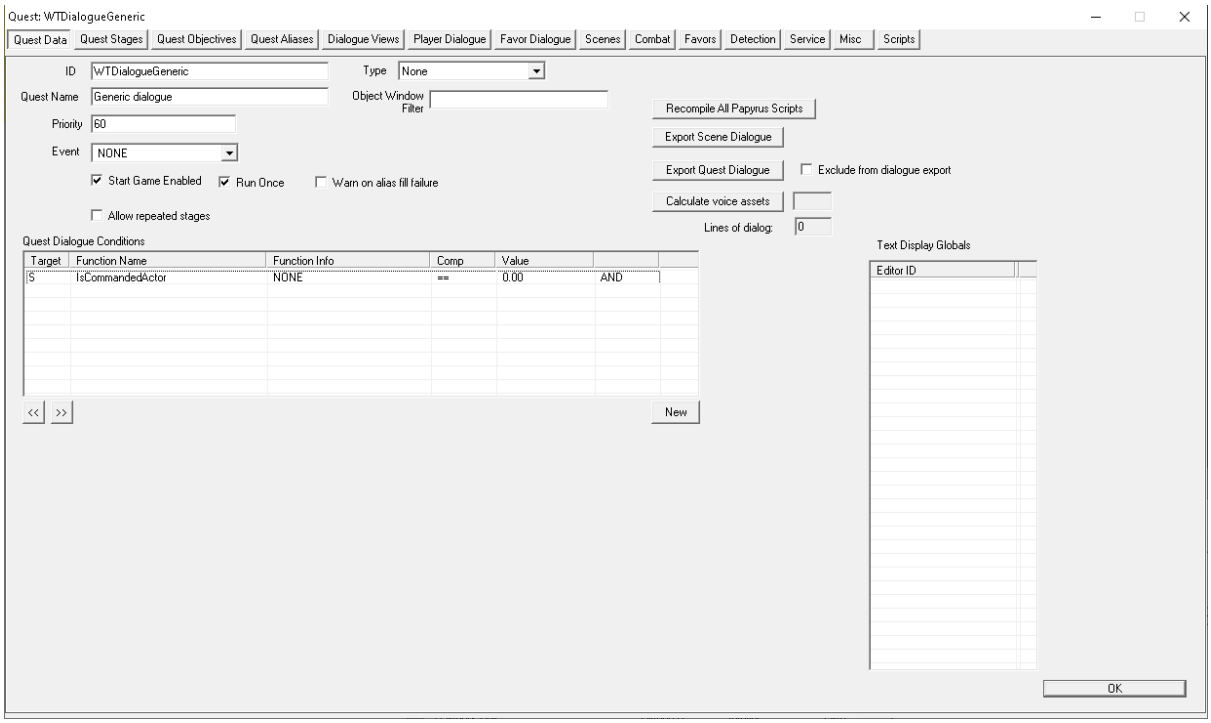


Figure 791 - Quest Data tab.

To add that condition, right-click in the Quest Dialogue Conditions list and select New.

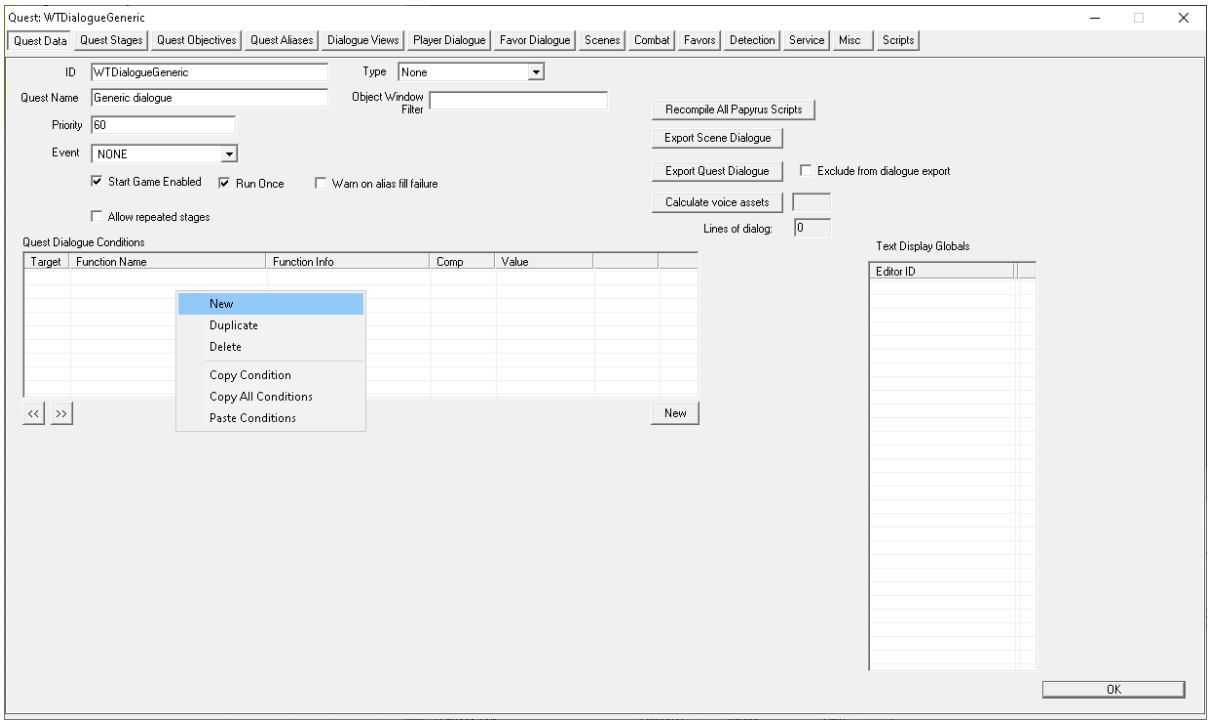


Figure 792 - Adding a Quest Dialogue Condition.

Set the Condition Function drop-down to 'IsCommandedActor' and set the Value field to 0.

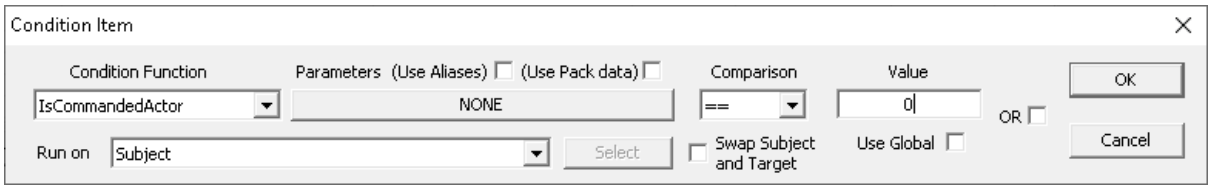


Figure 793 - IsCommandedActor condition function.

Click OK to close out of the Condition Item properties.

Let's start by adding some hellos. These lines are spoken by an NPC automatically when close to the player.

Go to the Misc tab, right-click in the Topics list and select New Topic.

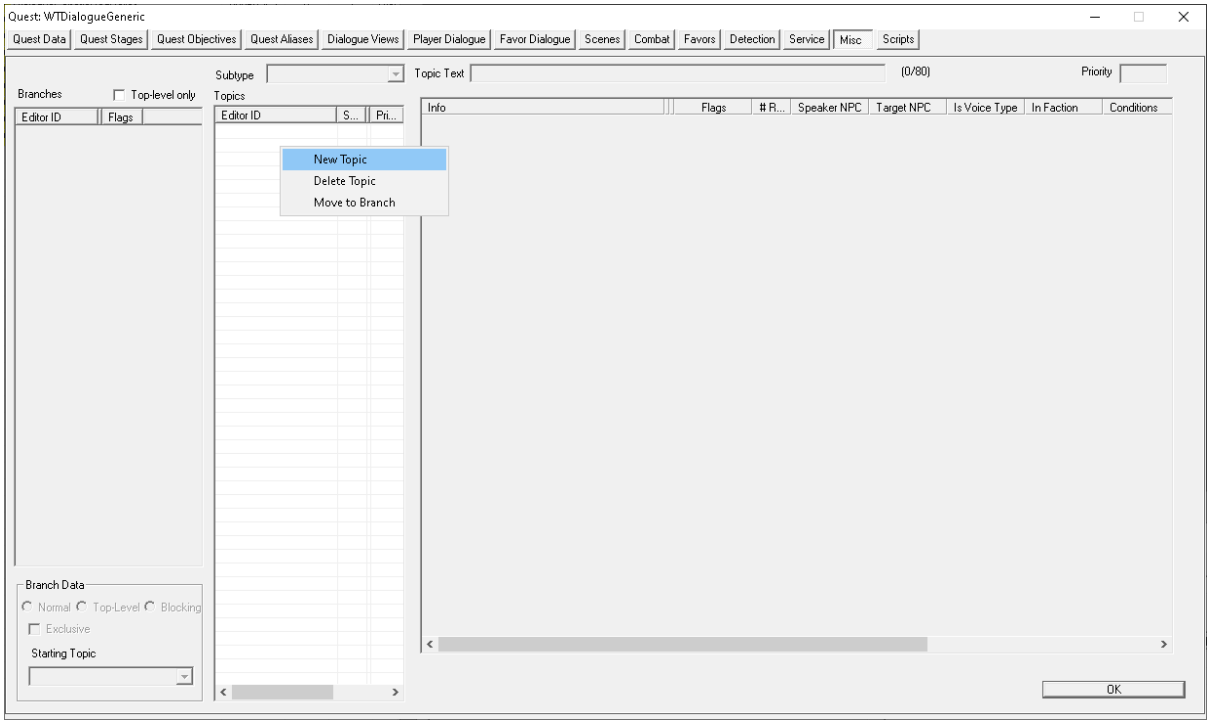


Figure 794 - Adding a new topic.

Select 'Hello' then click OK.

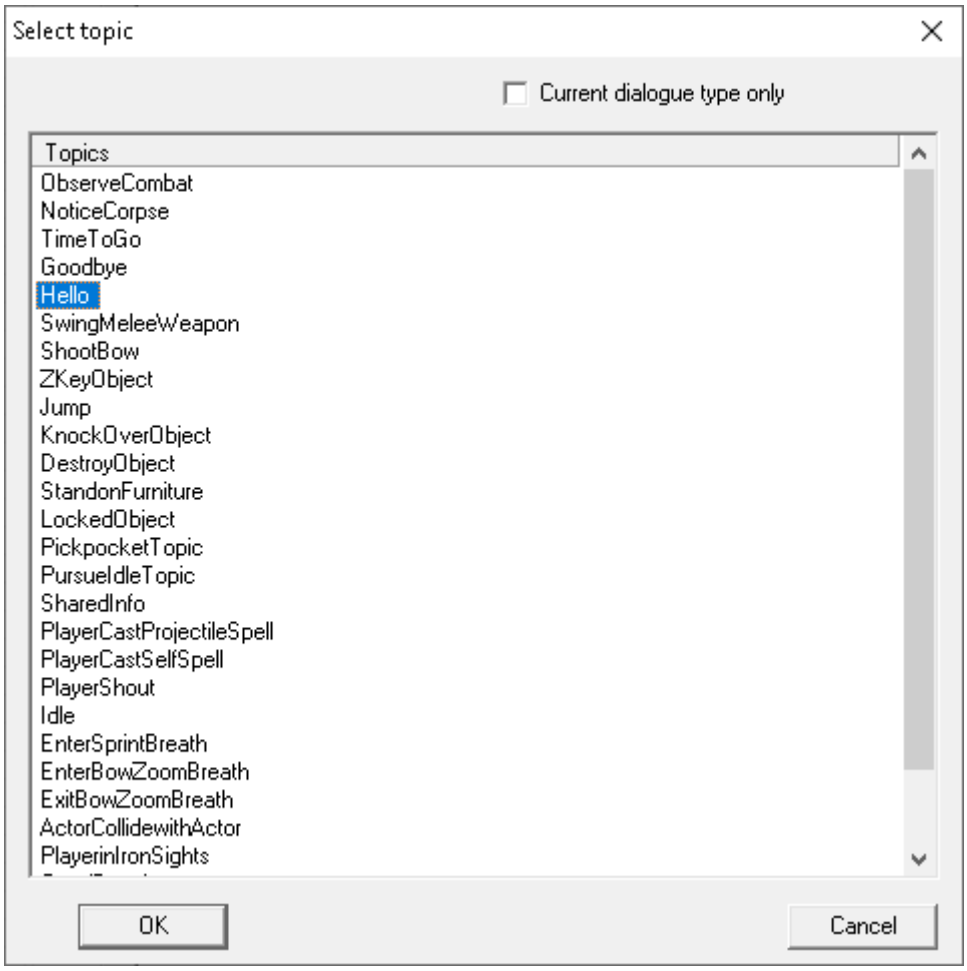


Figure 795 - List of topic types.

Next, you'll be prompted to enter in an ID for the topic.

For this example, I just called it WTDialogueGenericHello.



Figure 796 - Topic ID.

Click on the Hello topic in the Topics list to highlight it.

Now right-click in the section to the right of the Topics list and select New to add a new hello.

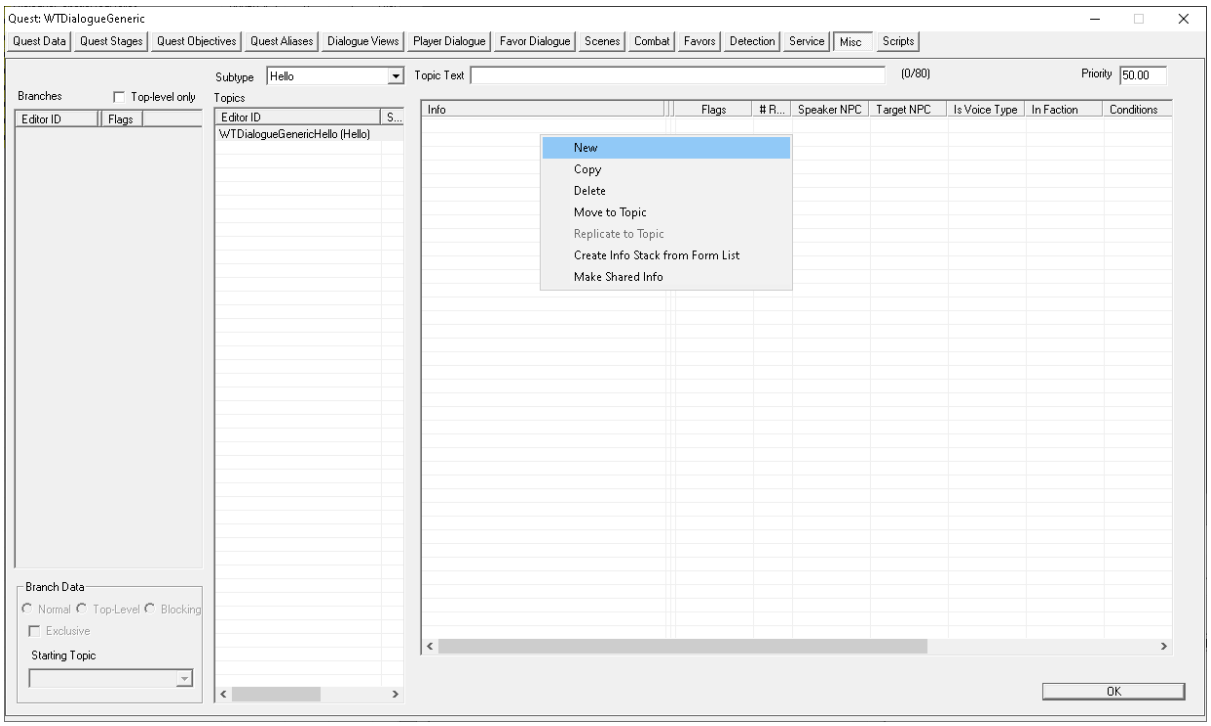


Figure 797 - Adding a new hello.

Enter in the line of dialogue in the Response Text field.

Set the Emotion Type and Emotion Value if necessary then click OK.

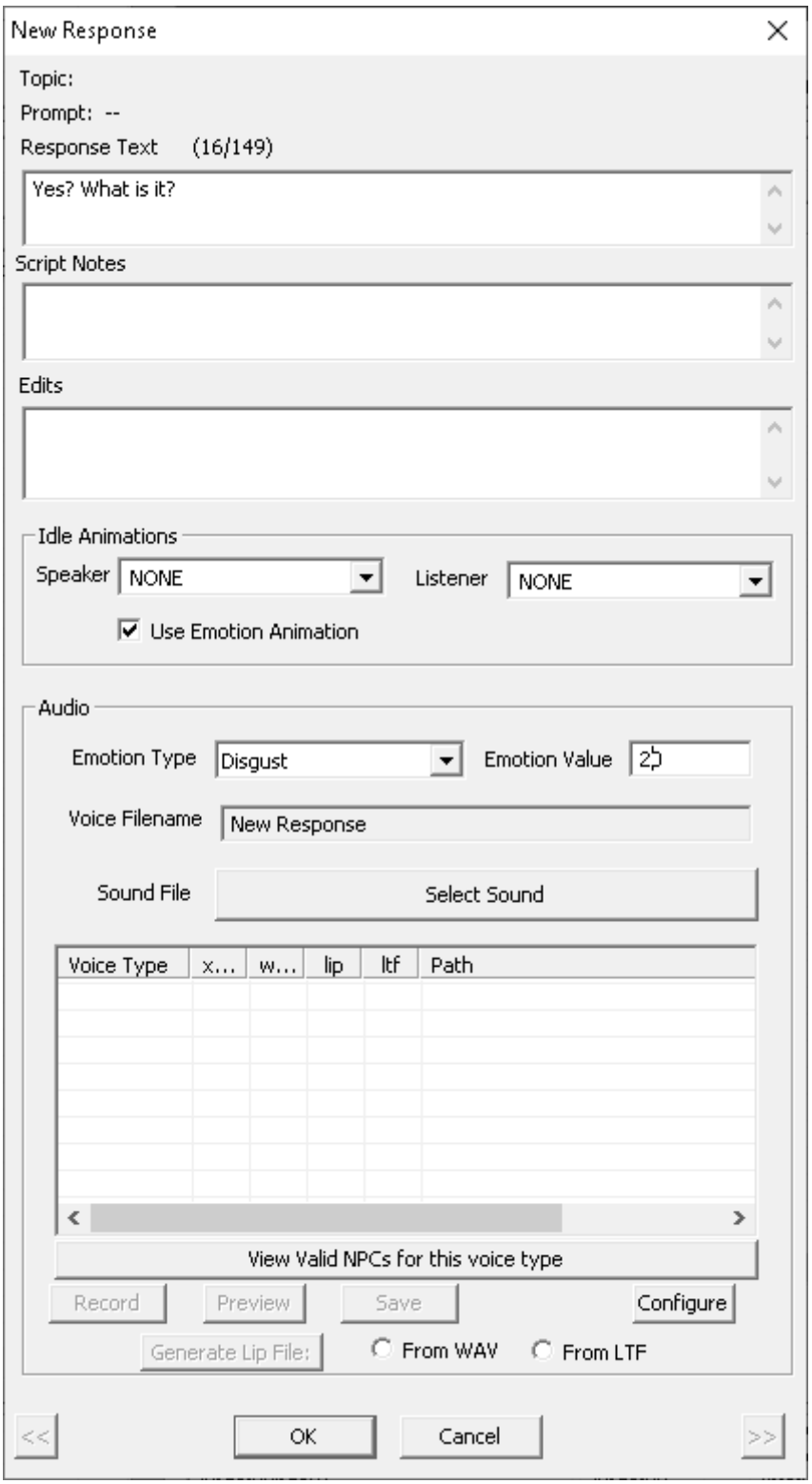


Figure 798 - Adding response text.

Now we need to add a condition to determine which NPCs will say this line.

Right-click in the Conditions list and select New.

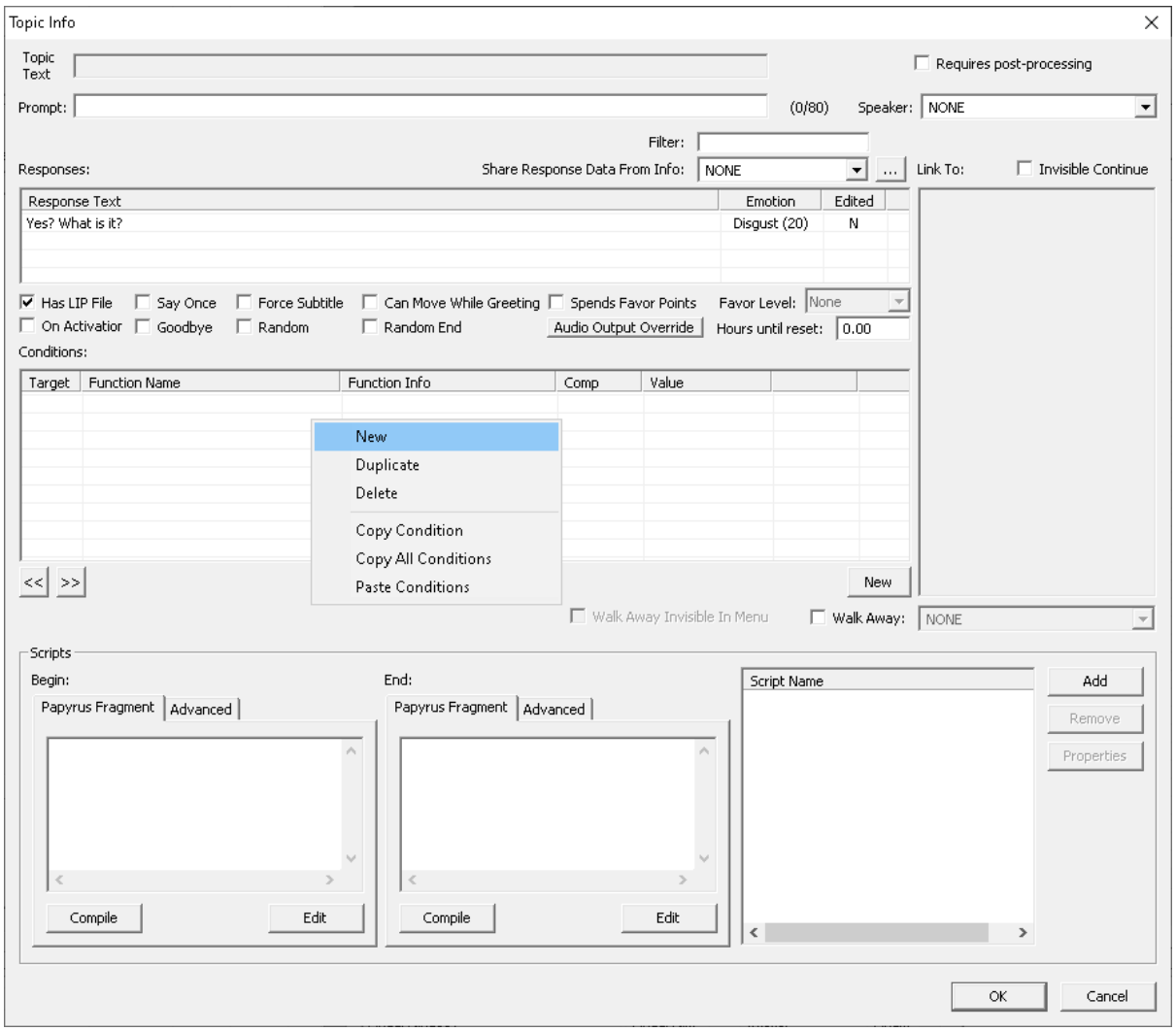


Figure 799 - Adding a condition to determine which NPC(s) will say this line.

One way to do that is by using the GetIsID condition to limit the line to a specific NPC. This is fine if you don't want to share lines between NPCs, if for example, your NPCs are all voiced by different voice actors.

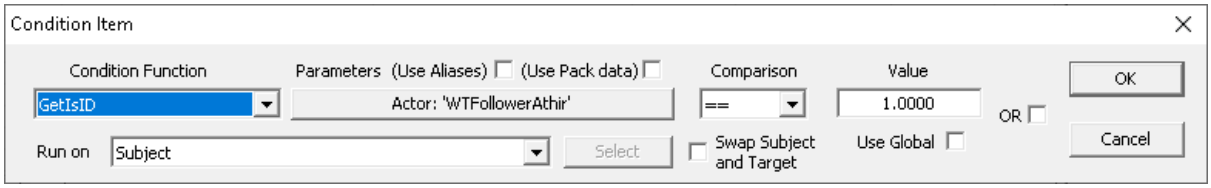


Figure 800 - Adding a GetIsID condition.

Another option is to limit the scope by voice type. This is done in the base game where a single voice actor is responsible for voice acting multiple NPCs.

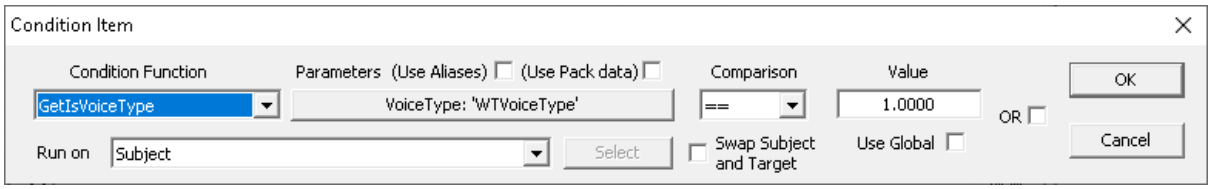


Figure 801 - Adding a GetIsVoiceType condition.

You can use a combination of conditions as well. The GetInFaction condition can be used to limit a line of dialogue only to members of a specific faction. For example, if you only want guards to say a specific line of dialogue, you can set the GetInFaction to the 'IsGuardFaction' faction which all guard NPCs are members of.

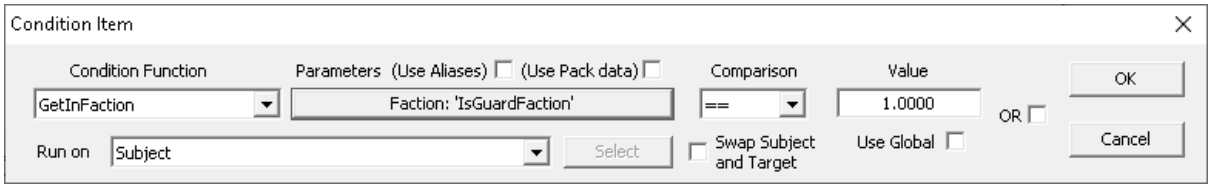


Figure 802 - Adding a GetInFaction condition.

For a full list of Condition Functions, see the article on the Creation Kit wiki.

Once your condition has been added, tick the 'Random' tickbox. By default, NPCs will cycle through all their hellos sequentially. By ticking random, NPCs will choose a hello at random instead.

Click OK to close out of Topic Info.

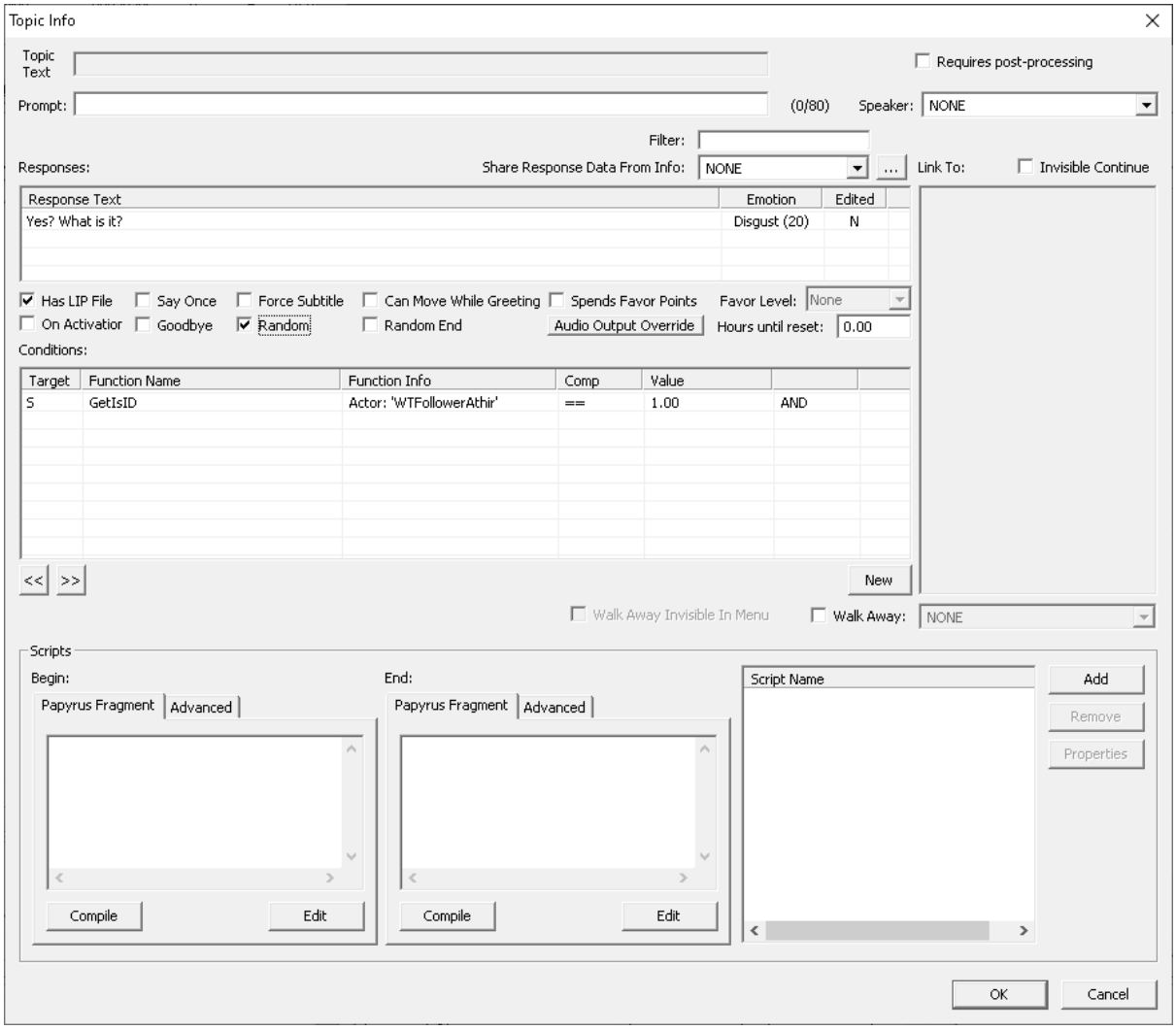


Figure 803 - Response condition added.

You should generally aim to have at least a dozen alternative lines for each NPC or voice type for the sake of variety.

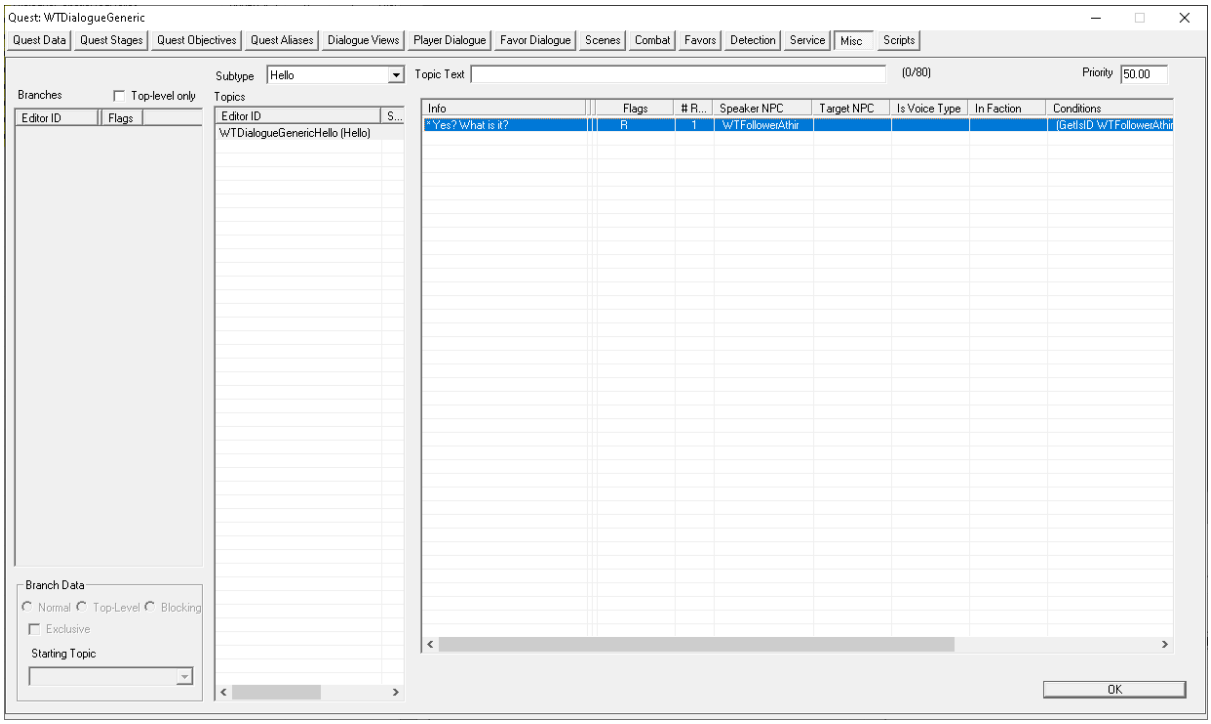


Figure 804 - The first hello added.

The following screenshot is from the WTDialogueIdle quest in Wyrmstooth showing you some of the hellos set up for NPCs.

As you can see some of the hellos are configured to only occur at certain quest stages using the GetStage condition function. Some hellos are only configured to occur at certain times of the day using the GetCurrentTime condition function.

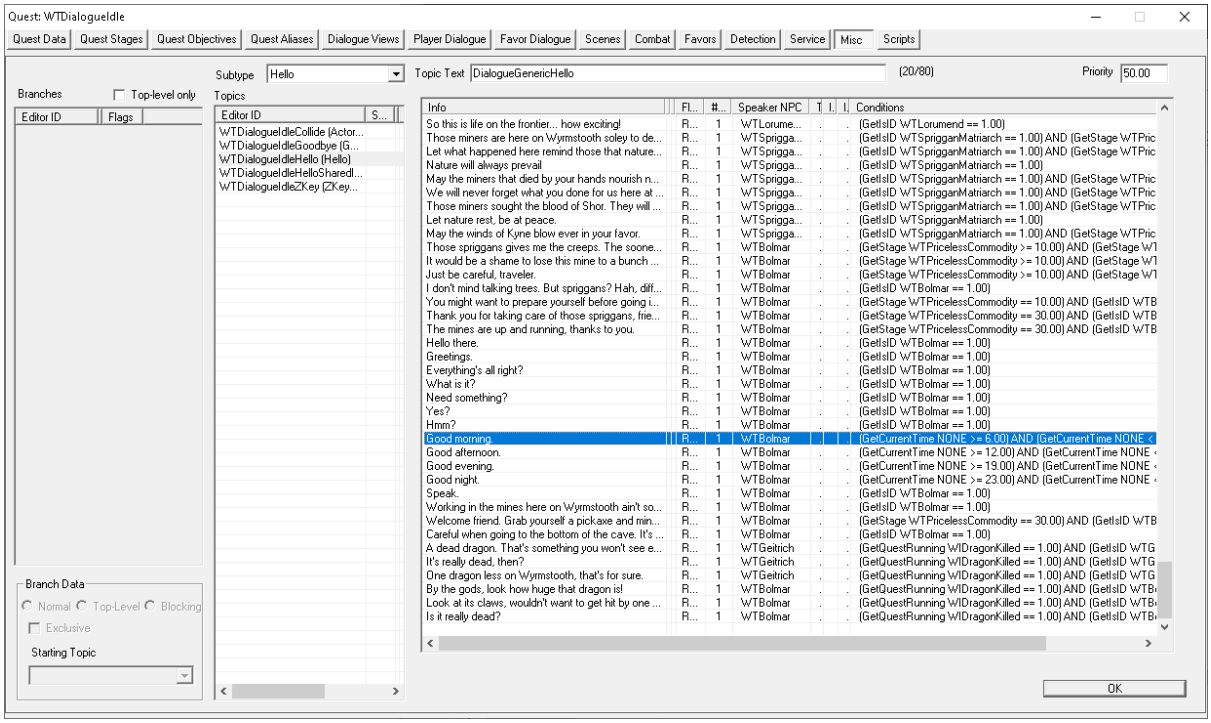


Figure 805 - List of hellos from Wyrmstooth.

Goodbyes are set up in the same way as hellos.

Right-click in the Topics list and select New Topic.

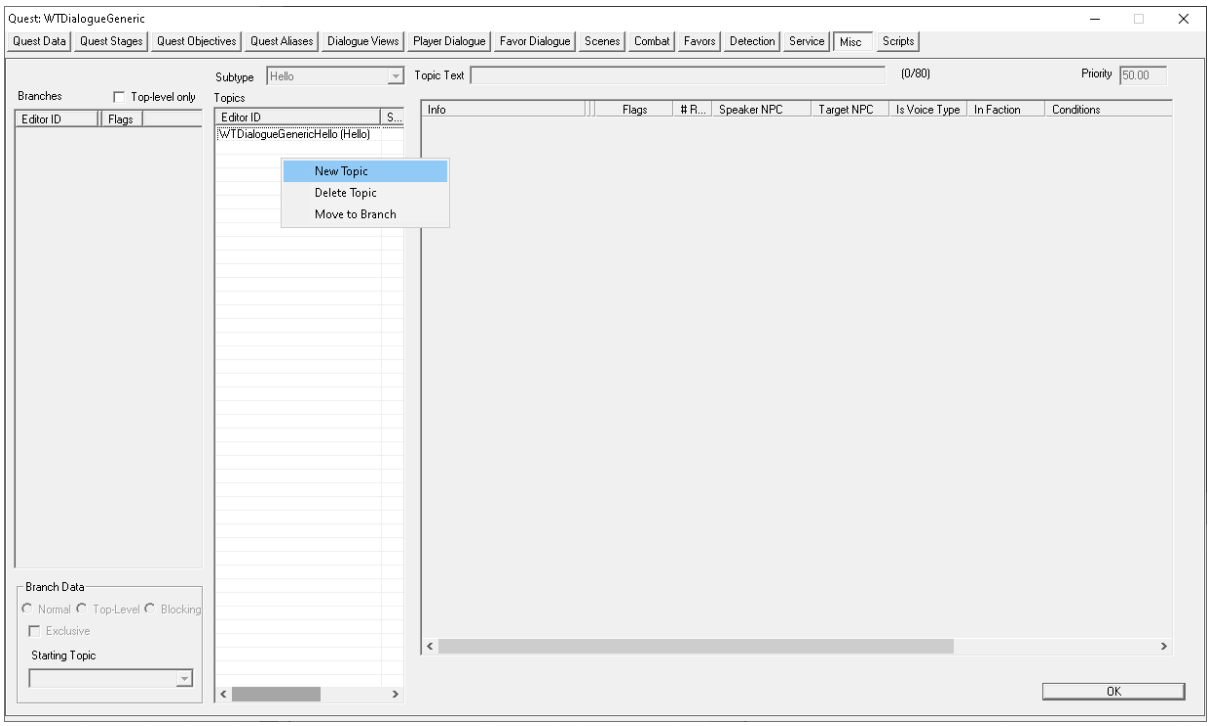


Figure 806 - Adding a goodbye topic.

This time, set the topic type to Goodbye then click OK.

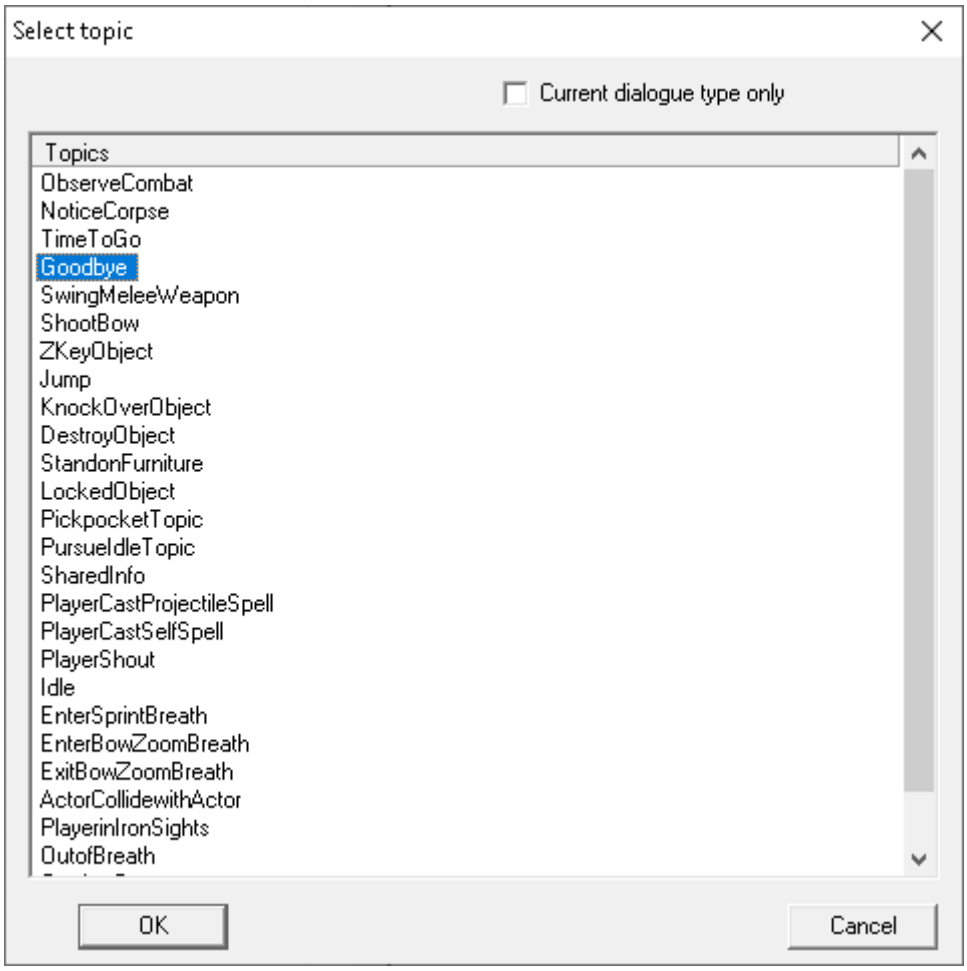


Figure 807 - Selecting the Goodbye topic type.

Enter in the Topic ID. For this example I just called it WTDialogueGenericGoodbye.

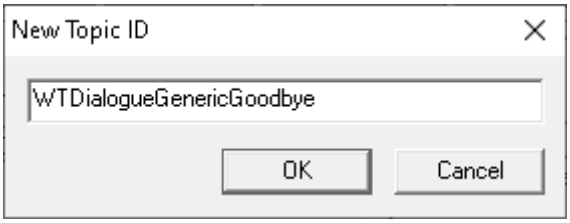


Figure 808 - Setting the topic ID.

To add a goodbye, click on the goodbye topic in the Topics list then right-click in the response list to the right and select New.

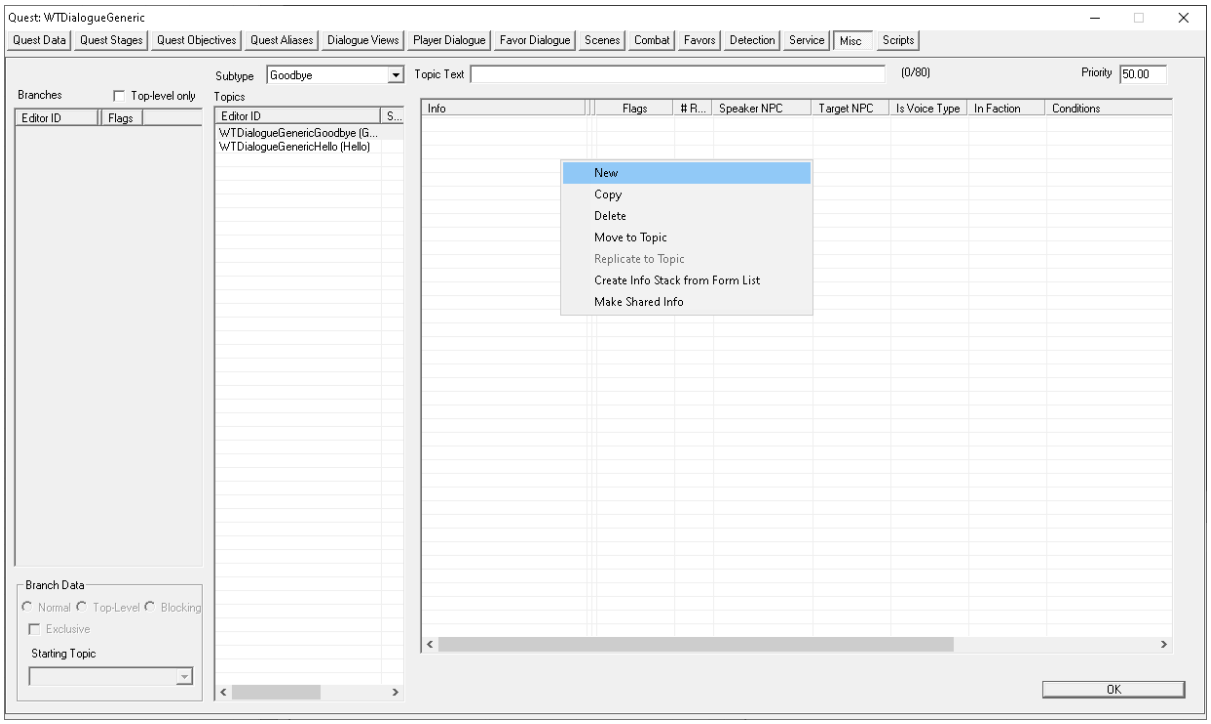


Figure 809 - Adding a new goodbye.

Goodbyes are set up in the same manner as hellos.

Note: You don't need to tick the 'Goodbye' tickbox for Goodbye dialogue.

Goodbyes are triggered when the player manually exits out of dialogue with an NPC.

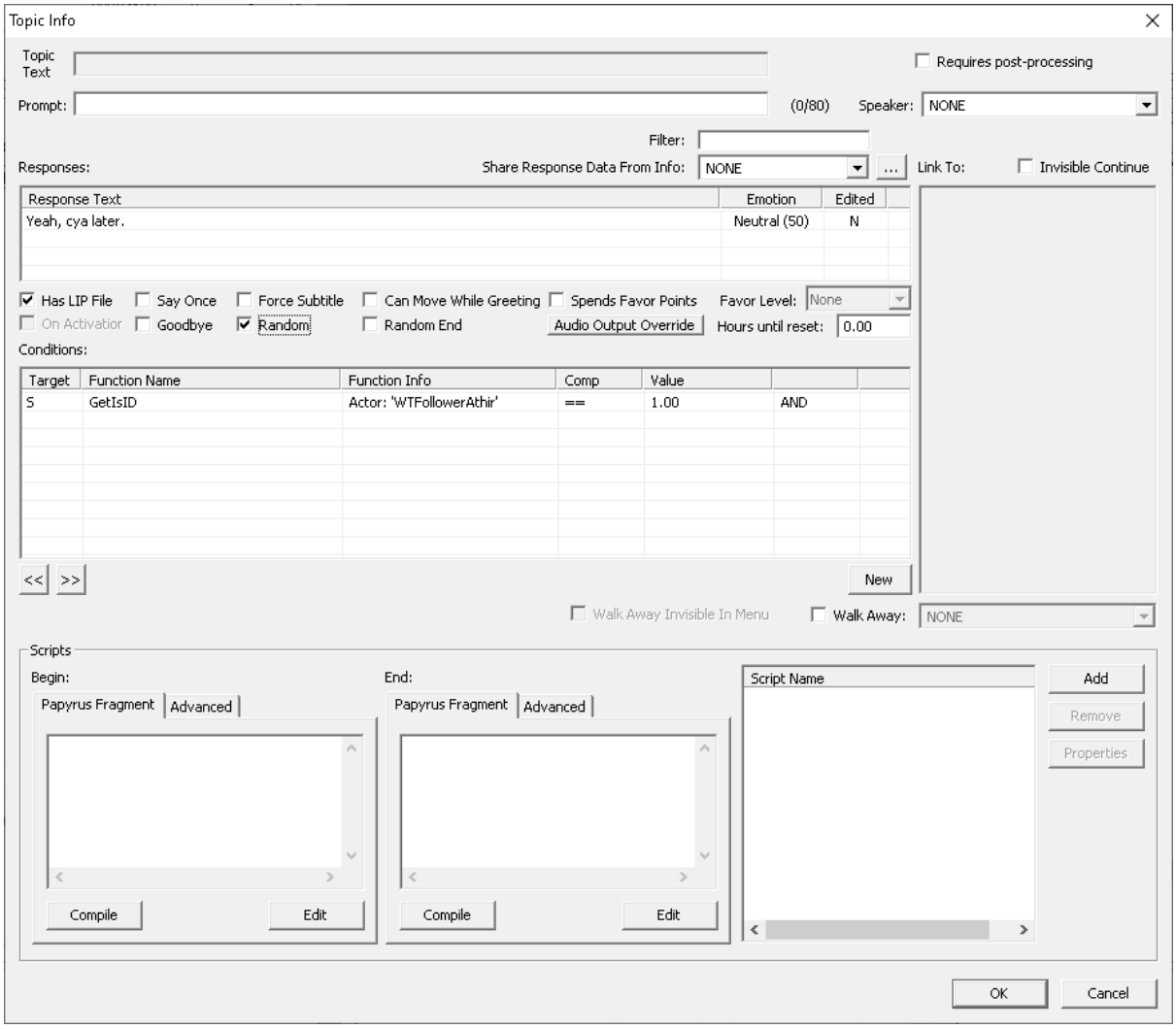


Figure 810 - A sample goodbye.

For more information on each topic type see the Misc tab article on the Creation Kit wiki.

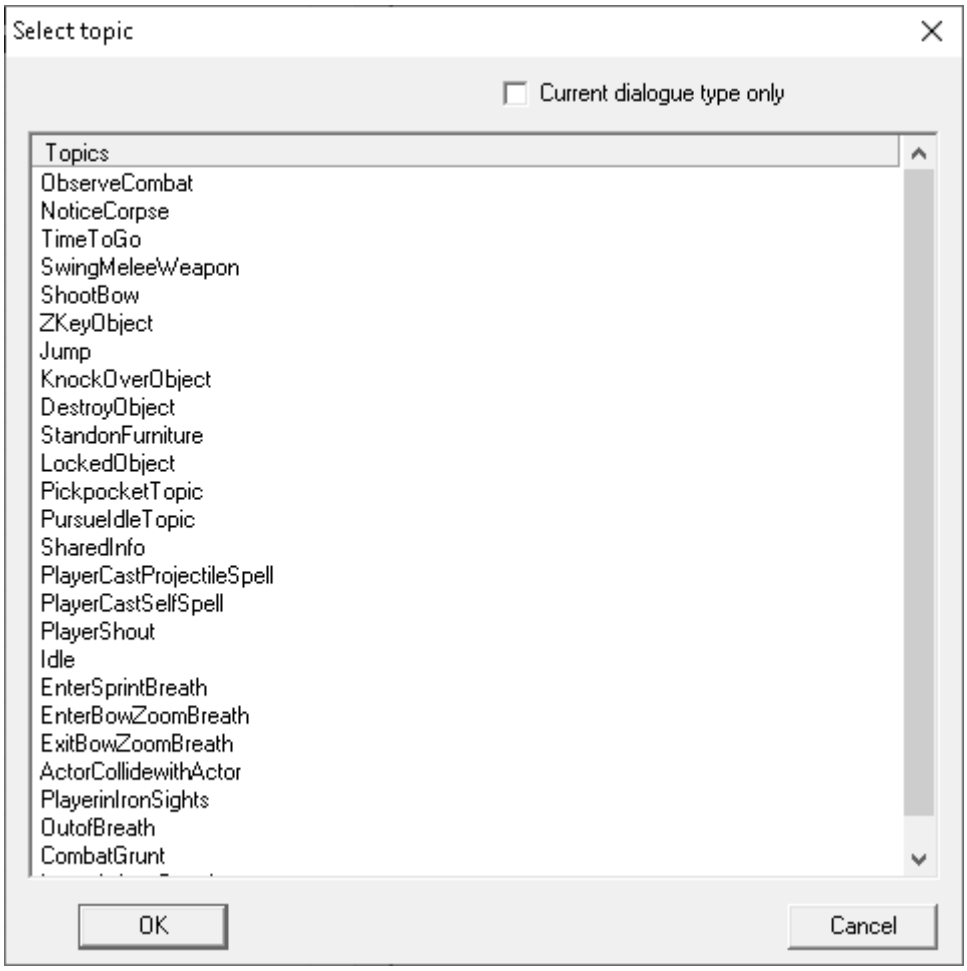


Figure 811 - List of topic types for the Misc tab.

TimeToGo is a trespass warning that doesn't lead to the guards getting called.

EnterSprintBreath, EnterBowZoomBreath, ExitBowZoomBreath, LeaveWaterBreath, OutofBreath, PlayerCastProjectileSpell, PlayerCastSelfSpell pertain to the player and don't need to be set up for NPCs.

Jump, DestroyObject, StandOnFurniture and PlayerIronSights aren't used.

Next, let's look at setting up topics under the Combat tab.

Topics defined here pertain to dialogue spoken by NPCs during combat, or upon witnessing the player commit a crime.

Let's start by adding an attack line. This line may be used when an NPC makes an attack with a melee weapon.

Right-click in the Topics list and select New Topic.

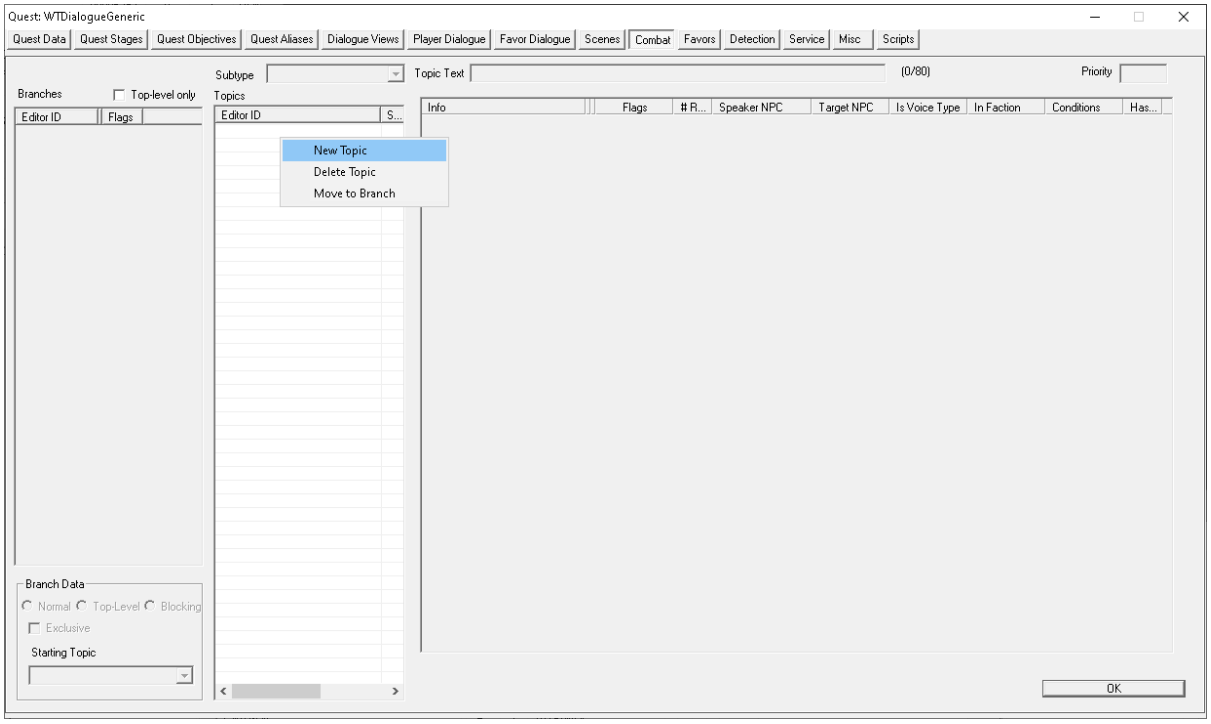


Figure 812 - Adding a new topic to the Combat tab.

Select Attack in the topic types list then click OK.

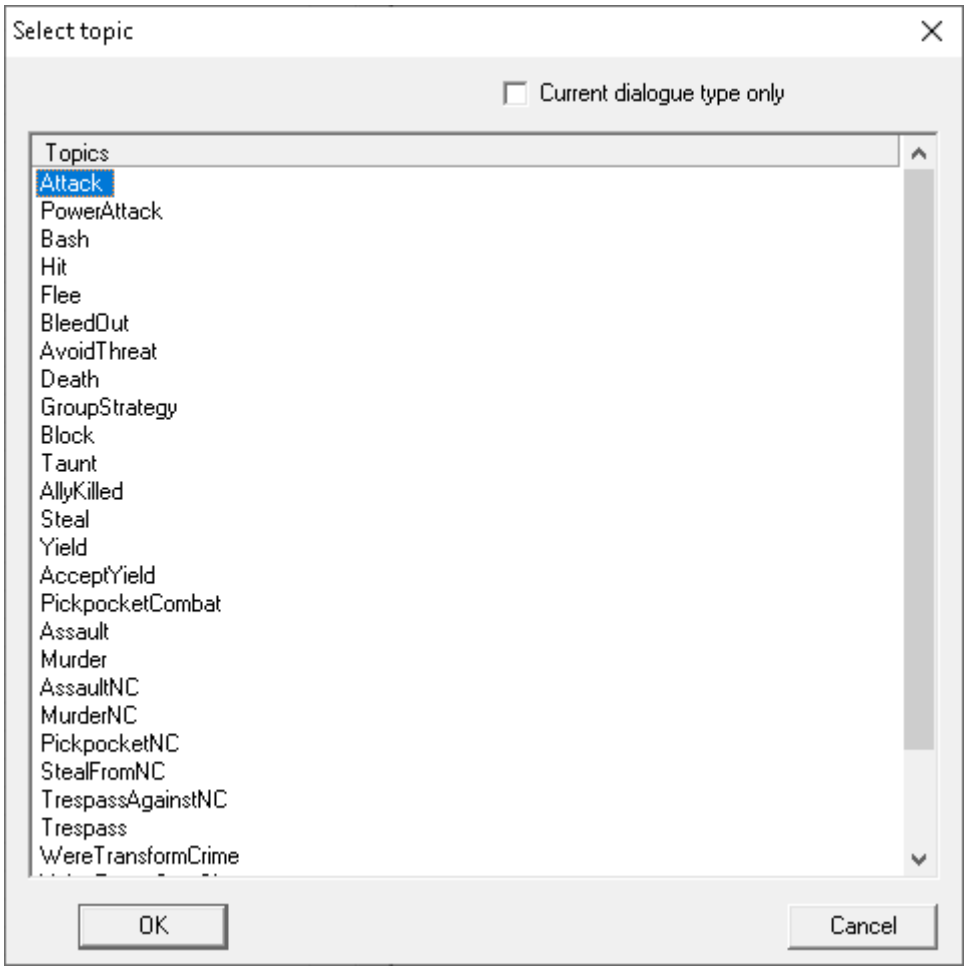


Figure 813 - List of combat topic types.

Set the Topic ID and click OK. For this example, I just set it to WTDialogueGenericAttack.

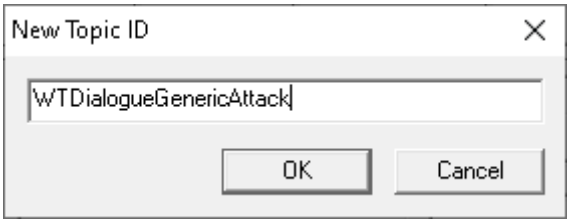


Figure 814 - Setting the topic ID for the attack topic.

Click on the attack topic to highlight it, then right-click in the section to the right and select New.

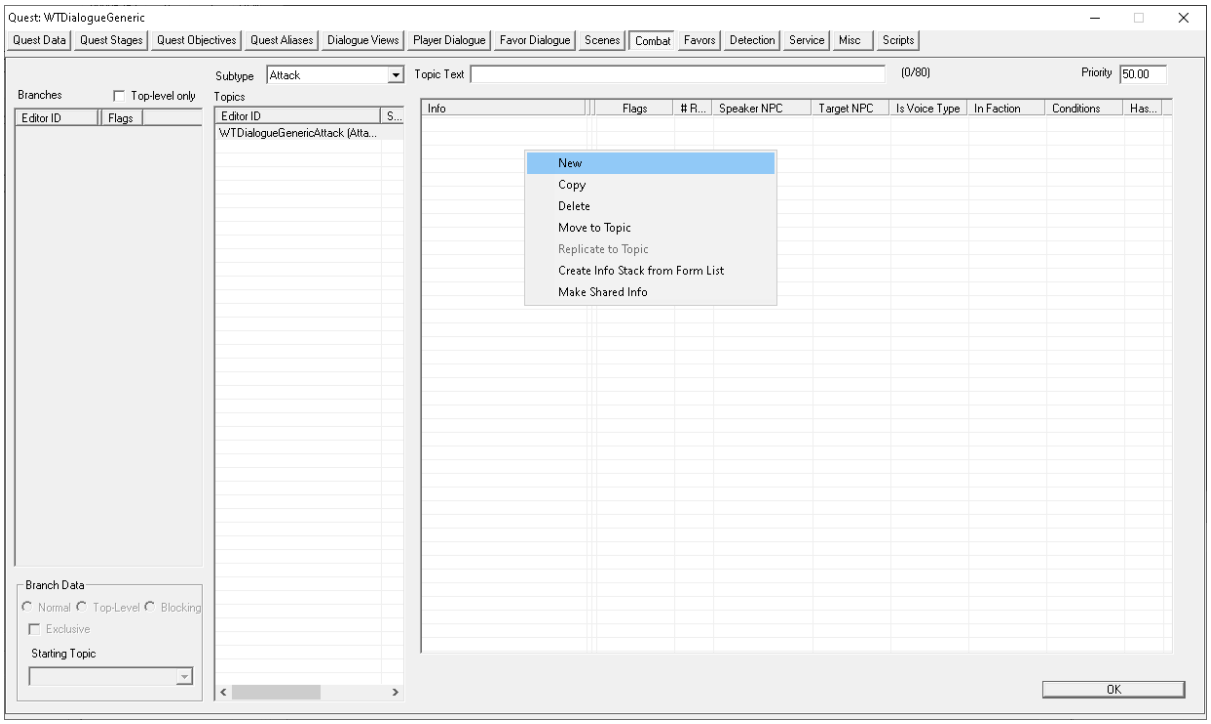


Figure 815 - Adding a new attack line.

Enter in the line of dialogue in the Response Text field. Generally you're going to want to set the Emotion Type to Anger for attack lines.

Click OK once you're done.

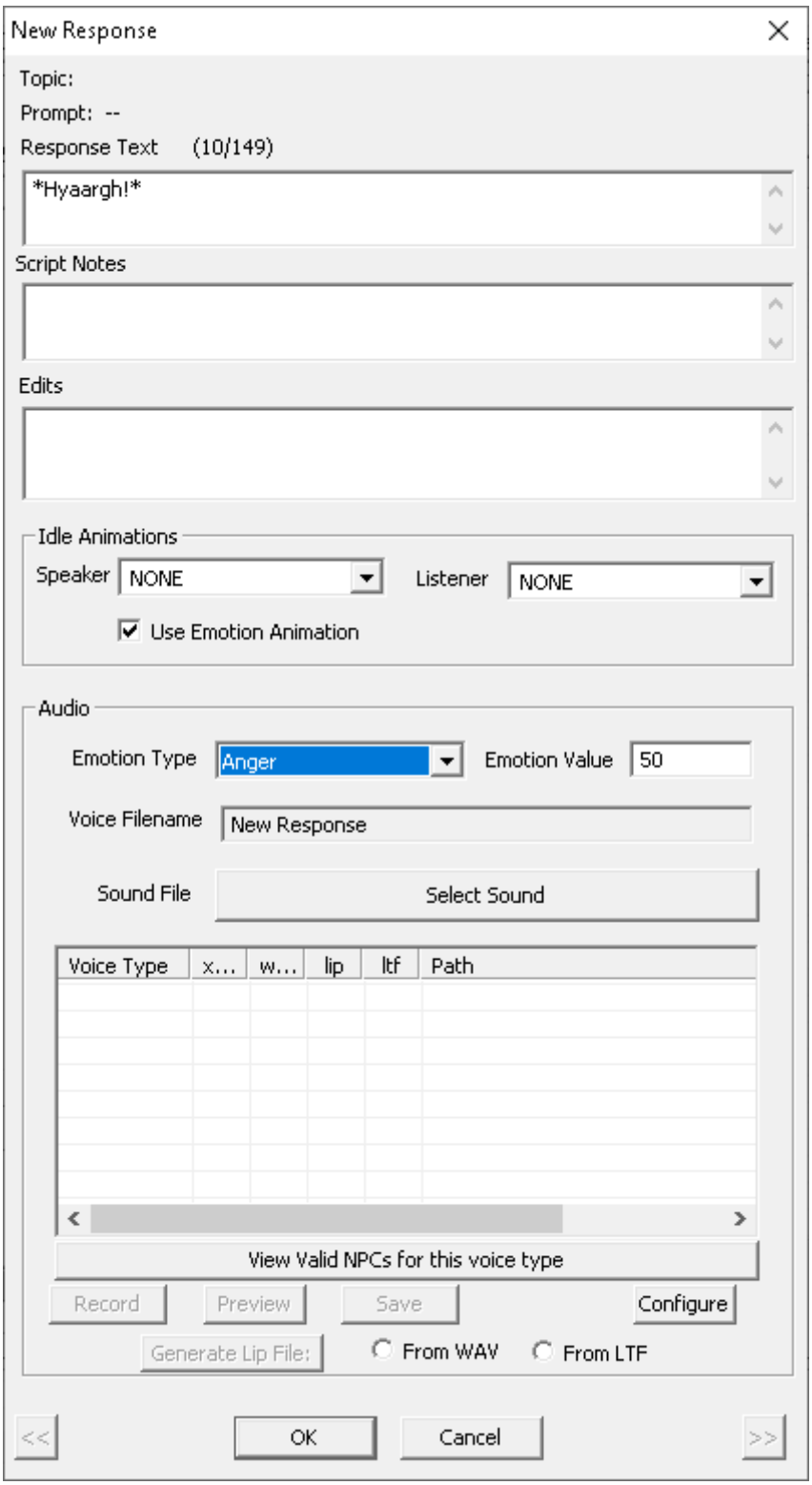


Figure 816 - Setting up an attack line.

Add a condition to determine which NPC(s) should use this line then click OK to close out of Topic Info.

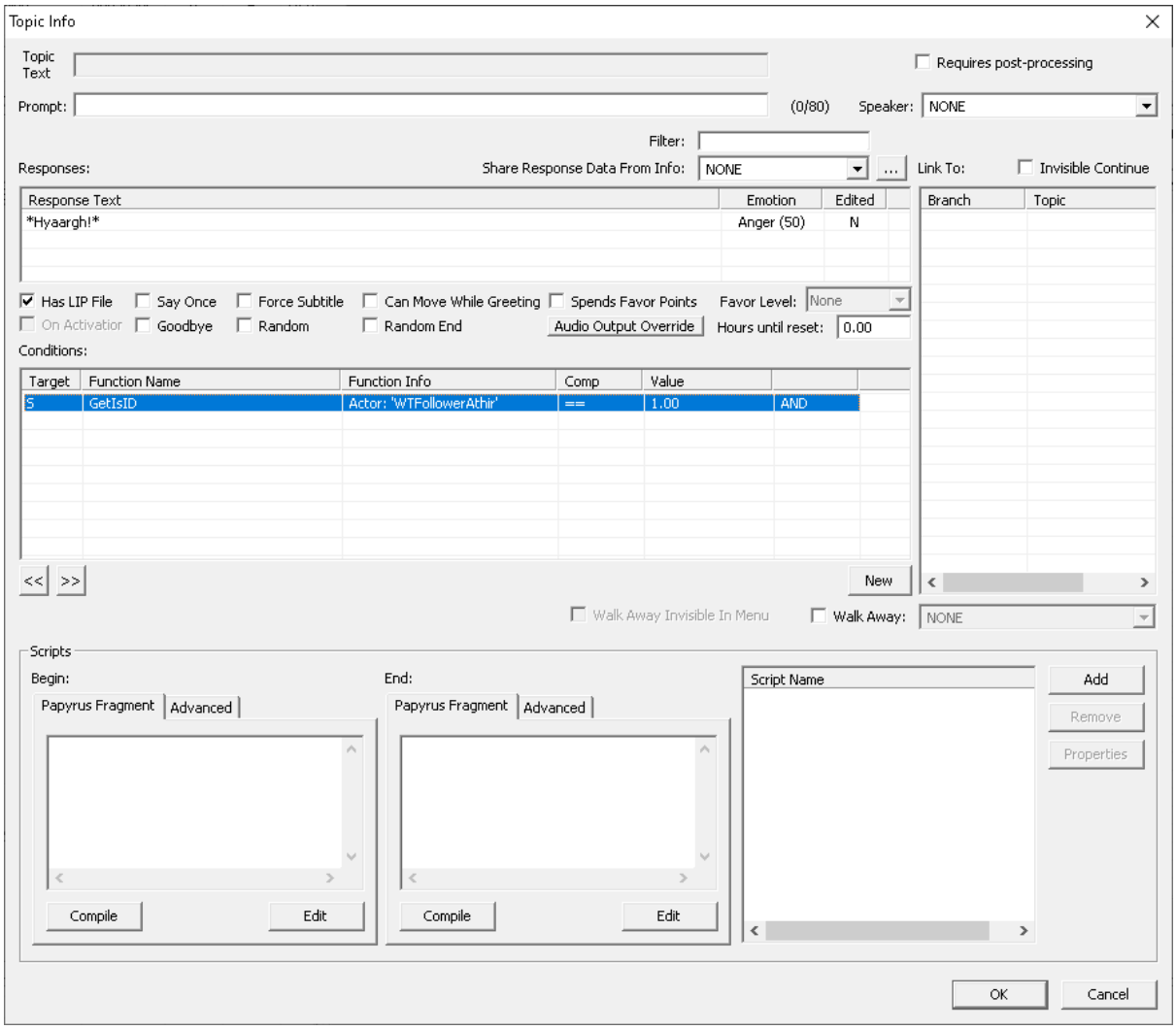


Figure 817 - Added a condition to the attack line.

You're going to want to have at least half a dozen or more different attack grunts for variation.

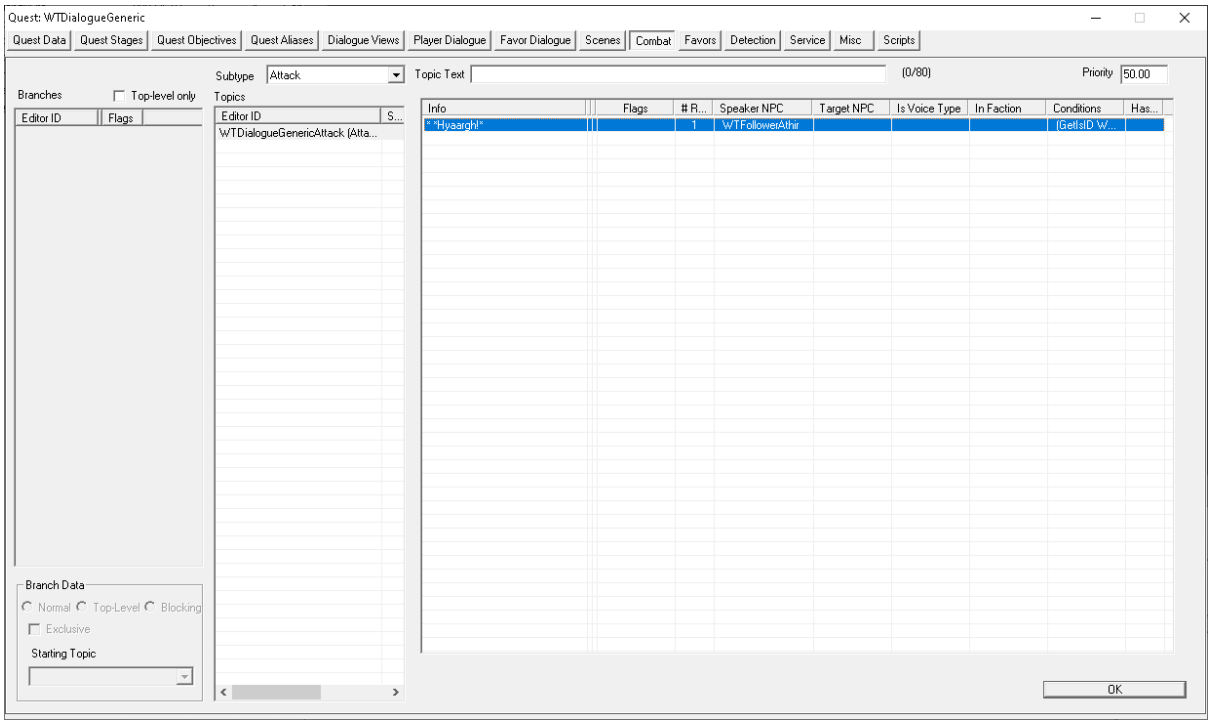


Figure 818 - The attack line has been added.

We can set up combat taunts in the same manner. These are phrases randomly spoken while an NPC is engaged in dialogue.

Right-click in the Topics list and select New Topic.

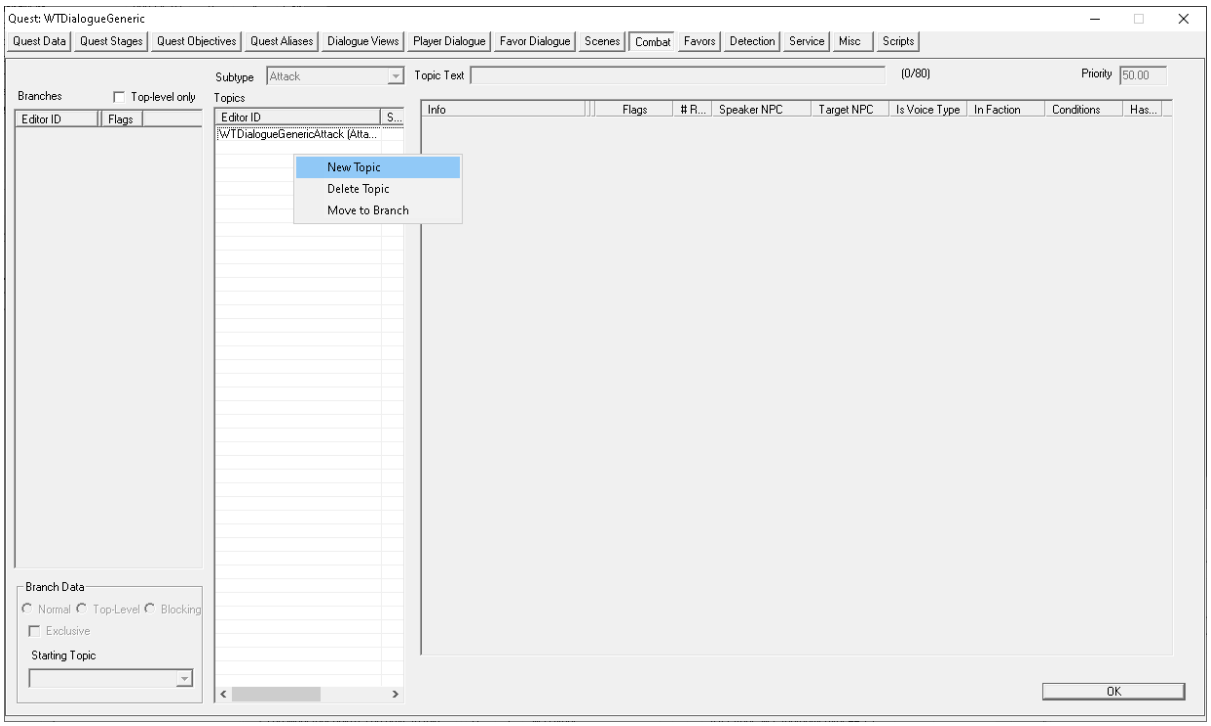


Figure 819 - Adding a taunts topic.

Click on the Taunt topic type to highlight it then click OK.

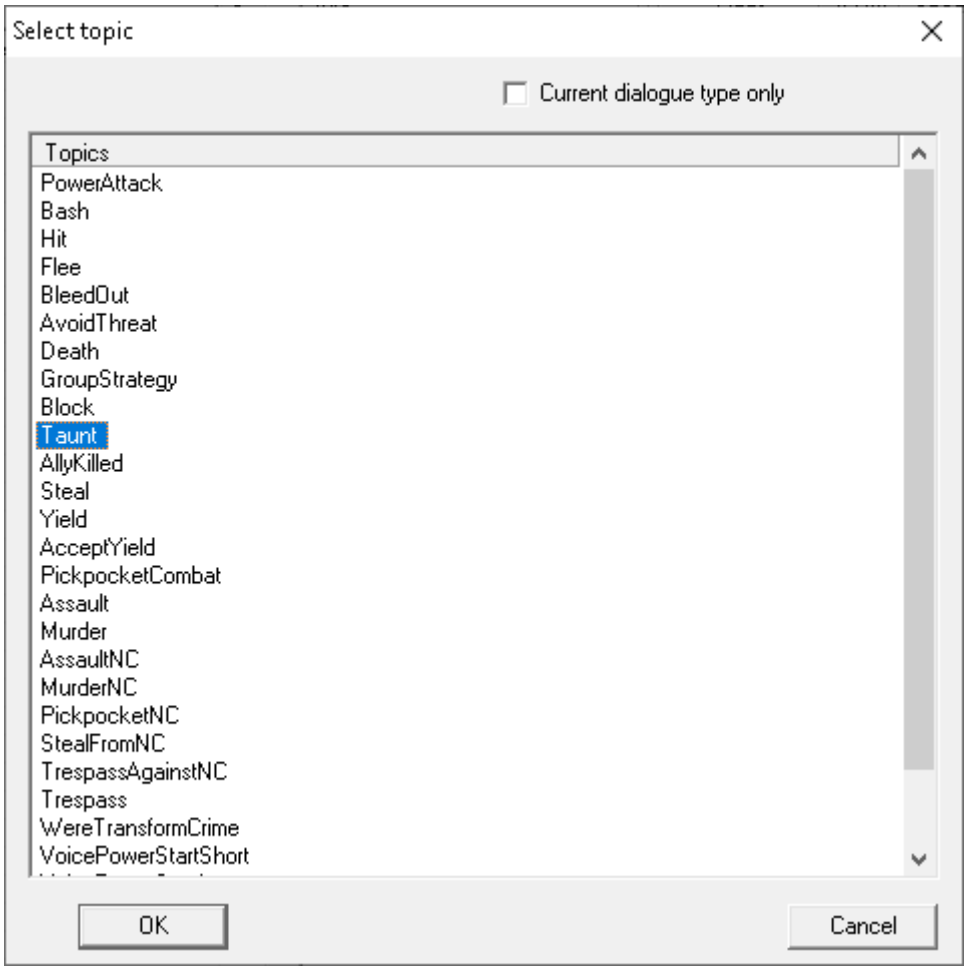


Figure 820 - Selecting the taunt topic type.

Enter in a Topic ID then click OK. For this example, I just went with WTDialogueGenericTaunts.

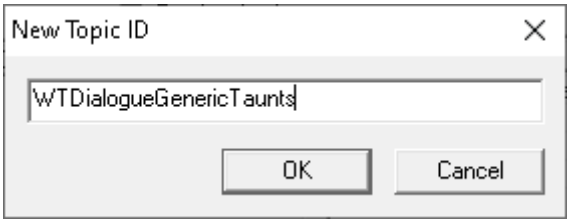


Figure 821 - Entering in a topic ID for the taunt topic.

Click on the taunts topic to highlight it then right-click in the area to the right and select New.

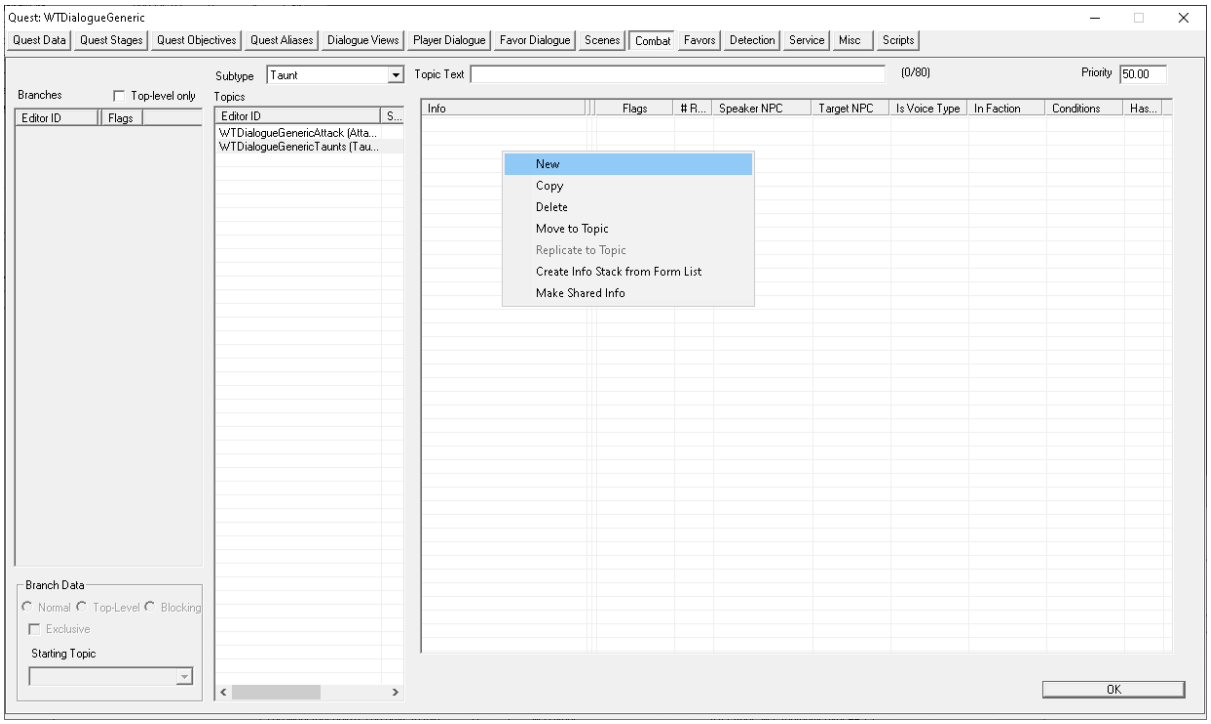


Figure 822 - Adding a new combat taunt.

Enter in the line of dialogue in the Response Text field. Again, you're probably going to want to set the Emotion Type to Anger.

Click OK to close out of New Response.

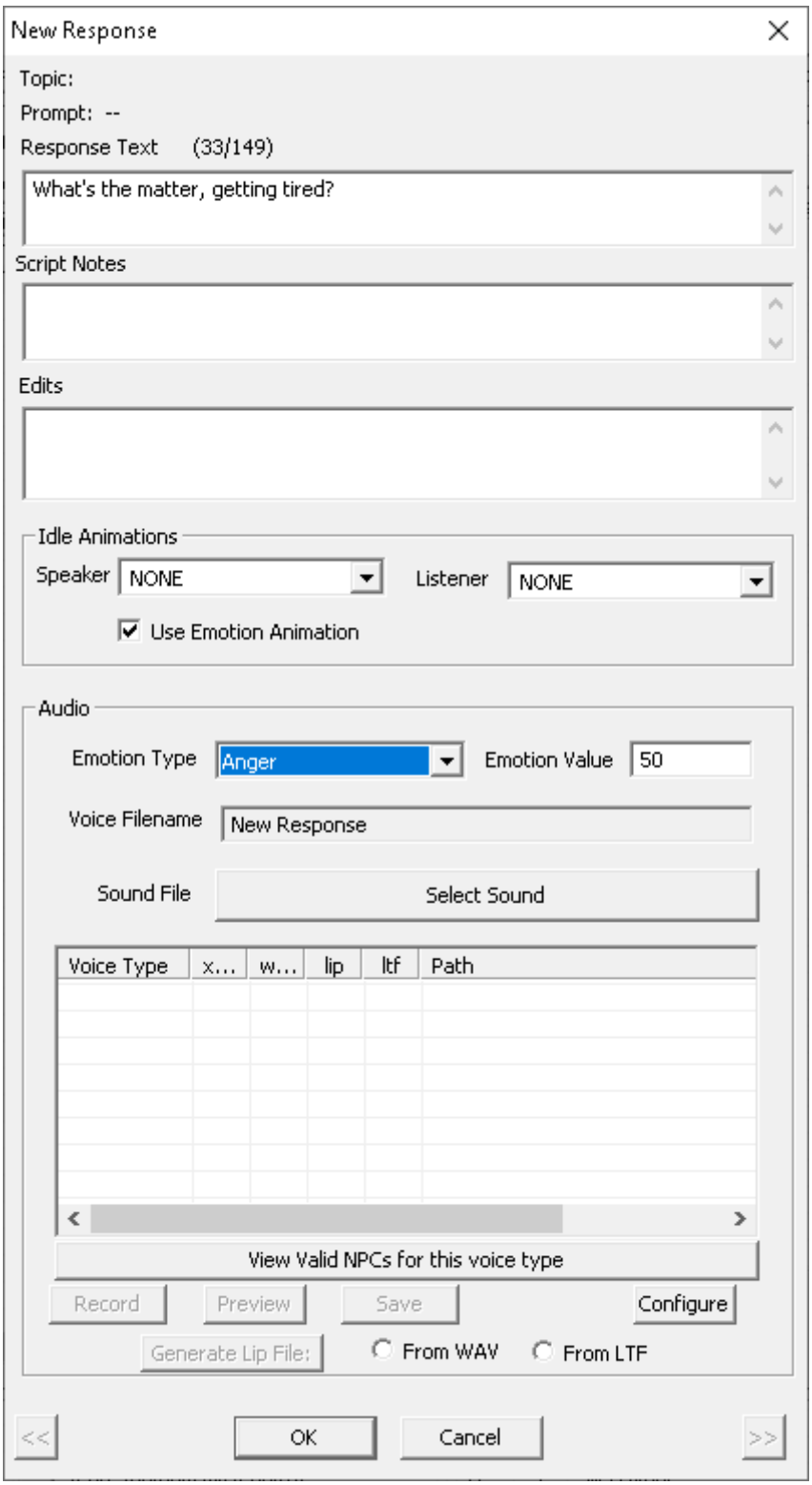


Figure 823 - Setting up a combat taunt.

Again, tick Random and add a condition to determine which NPC(s) can use this line of dialogue during combat then click OK.

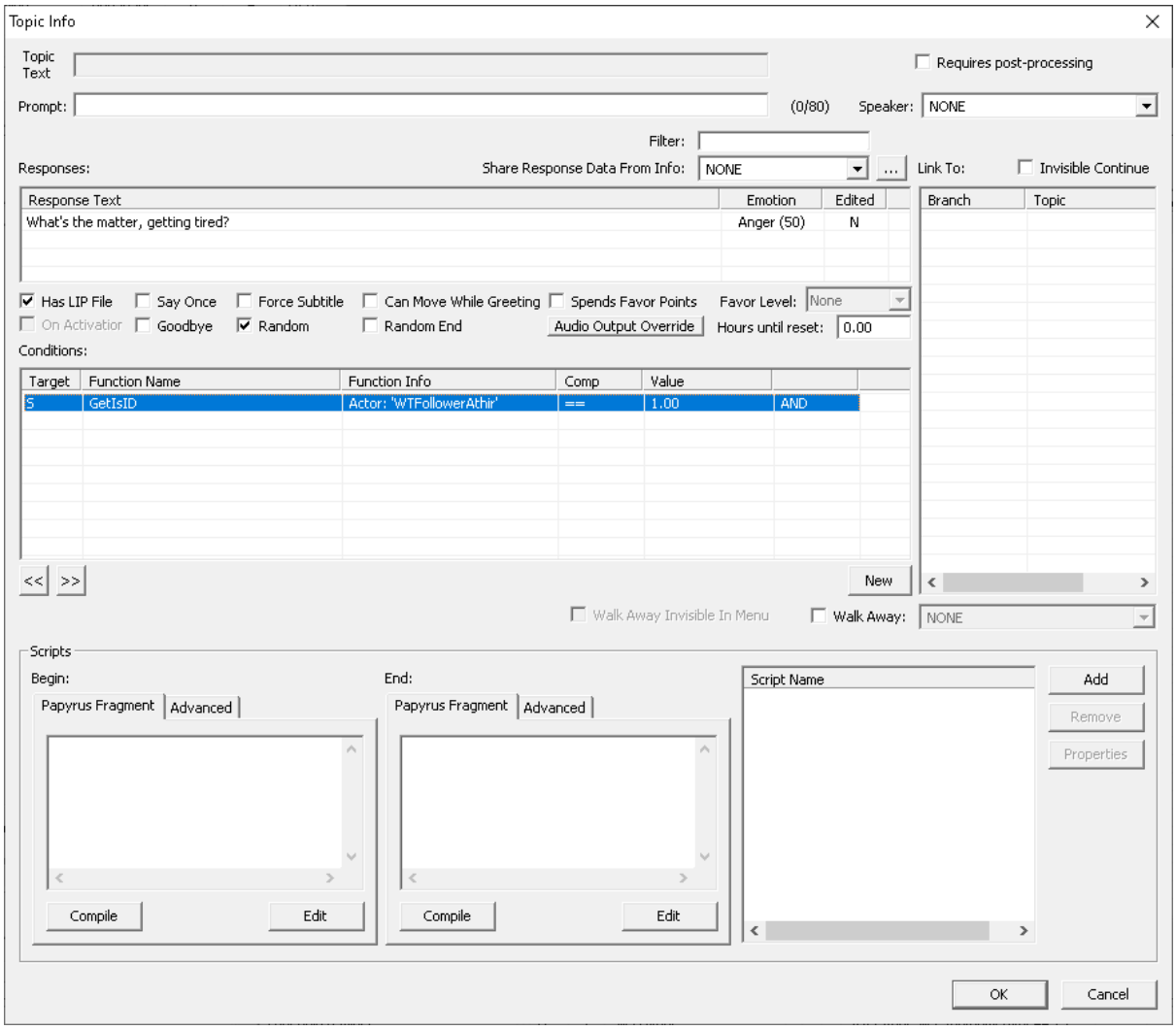


Figure 824 - Combat taunt topic info properties.

Here are some combat taunts I set up for Wyrmstooth:

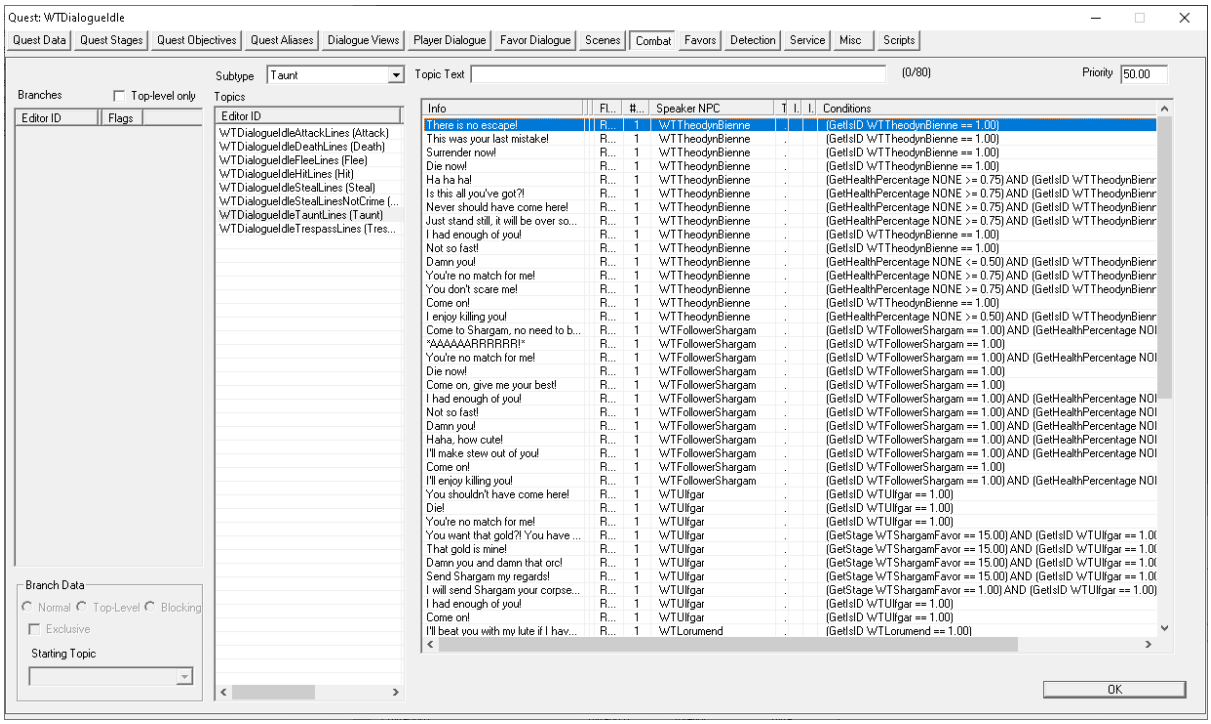


Figure 825 - Combat taunts in Wyrmstooth.

Some of the combat taunts make use of the GetHealthPercentage condition function so that optimistic liens are only spoken if the NPC is in good health, while more pessimistic lines are only spoken at low health when the NPC is closer to death.

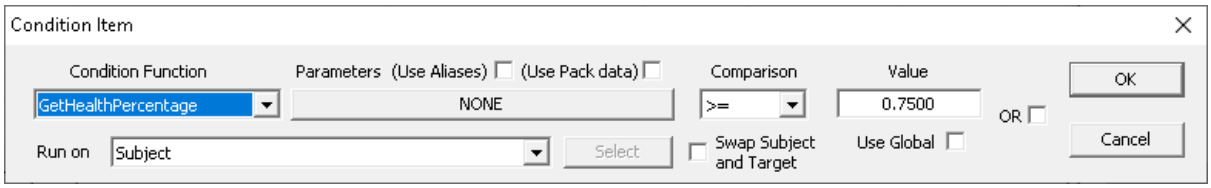


Figure 826 - GetHealthPercentage function checking whether the NPC has 75% of their health or more.

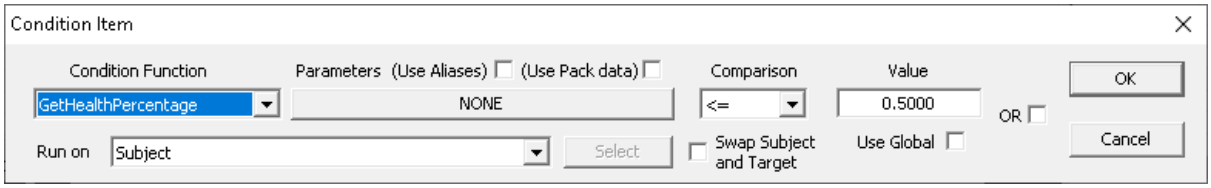


Figure 827 - GetHealthPercentage function checking whether the NPC is down to 50% health or less.

In Wyrmstooth, I also make use of the GetStage condition function to only play certain combat taunts if the NPC has learned specific information from the player.

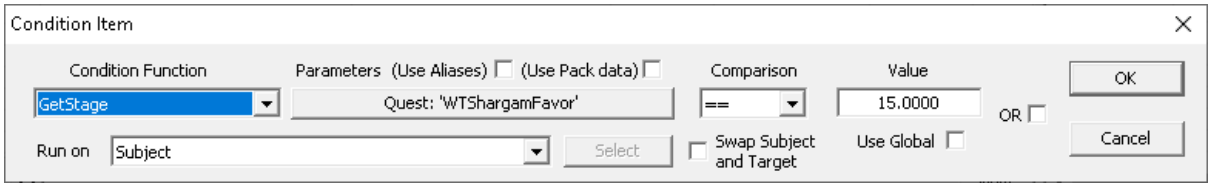


Figure 828 - GetStage condition used to play specific taunts only at a specific quest stage.

Let's look at crime lines next.

Firstly it's important to note that crime topics also have a 'No Care' variant. They're the ones that have 'NC' appended to the end.

No Care lines are used when the player is spotted committing a crime, but the NPC witnessing the player is not in or is not friendly with the faction that the crime was committed against.

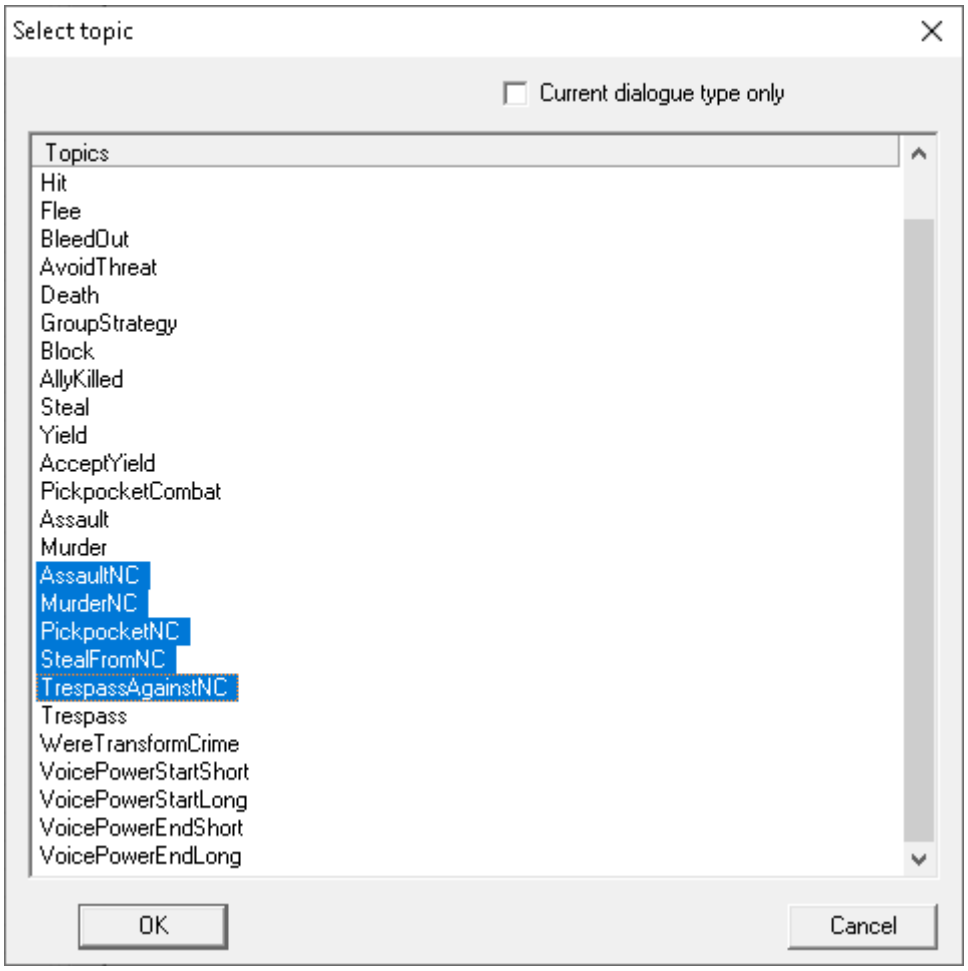


Figure 829 - No Care topics highlighted.

Crime topics are set up in the same manner as the Attack and Taunt topics, but Trespass is a bit different.

TrespassAgainstNC is the No Care version that does not result in guards getting called.

Trespass, on the other hand, escalates in several warning levels, with the last level resulting in guards getting called.

Let's take a look at setting up the Trespass topic.

Right-click in the Topics list and select New.

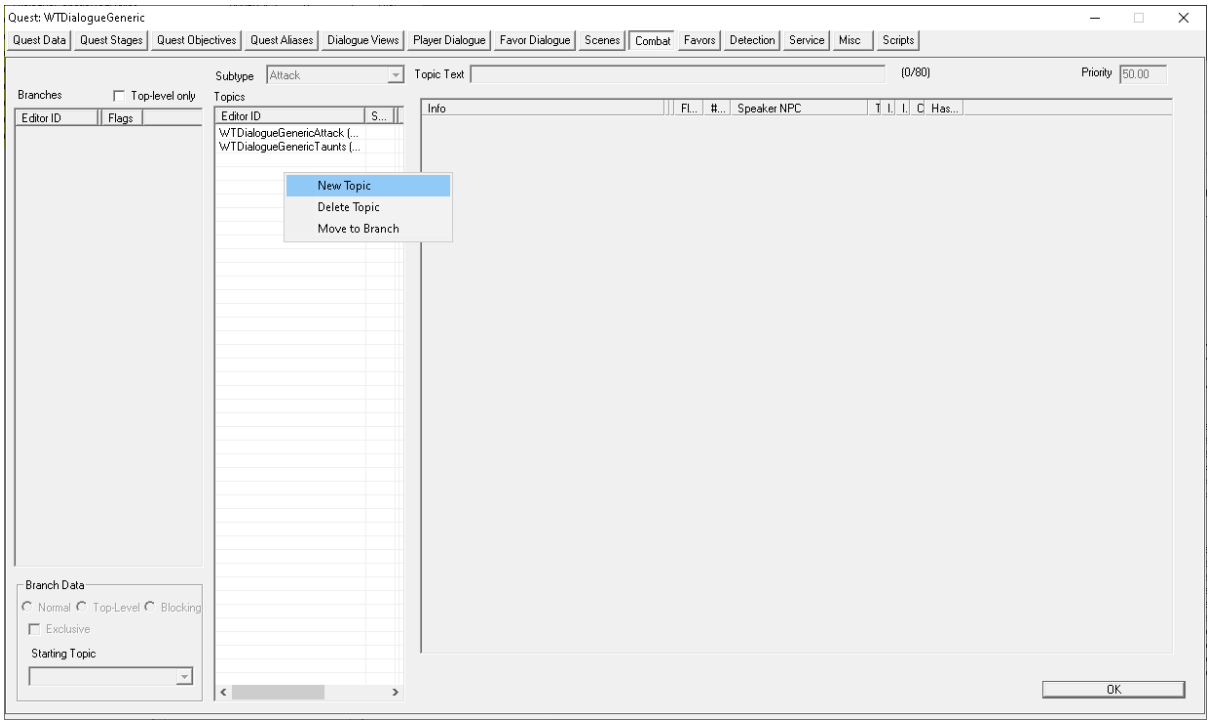


Figure 830 - Setting up the Trespass topic.

Click on Trespass to highlight it then click OK.

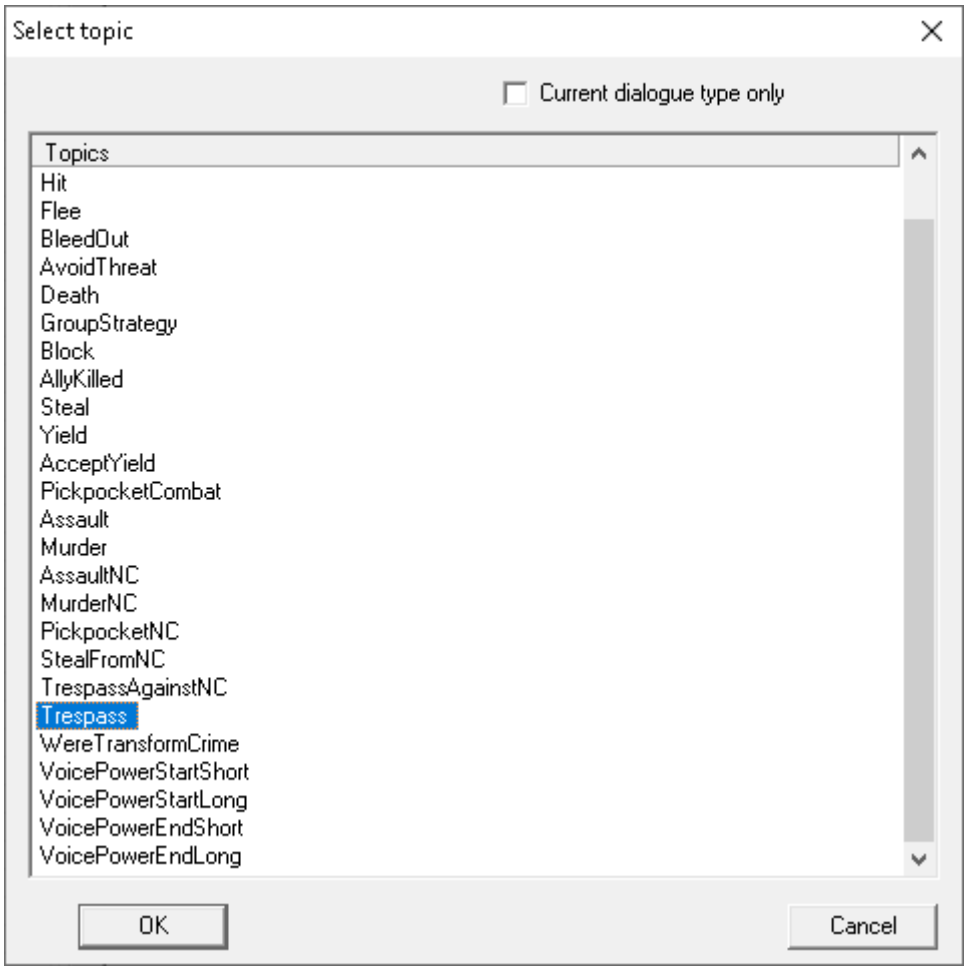


Figure 831 - Selecting the Trespass topic.

Enter in the topic ID then click OK. For this example I just went with WTDialogueGenericTrespass.

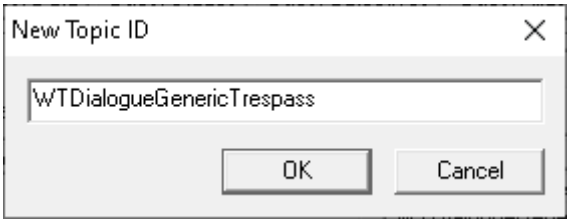


Figure 832 - Setting the topic ID.

Click on the Trespass topic to highlight it then right-click in the area to the right and select New.

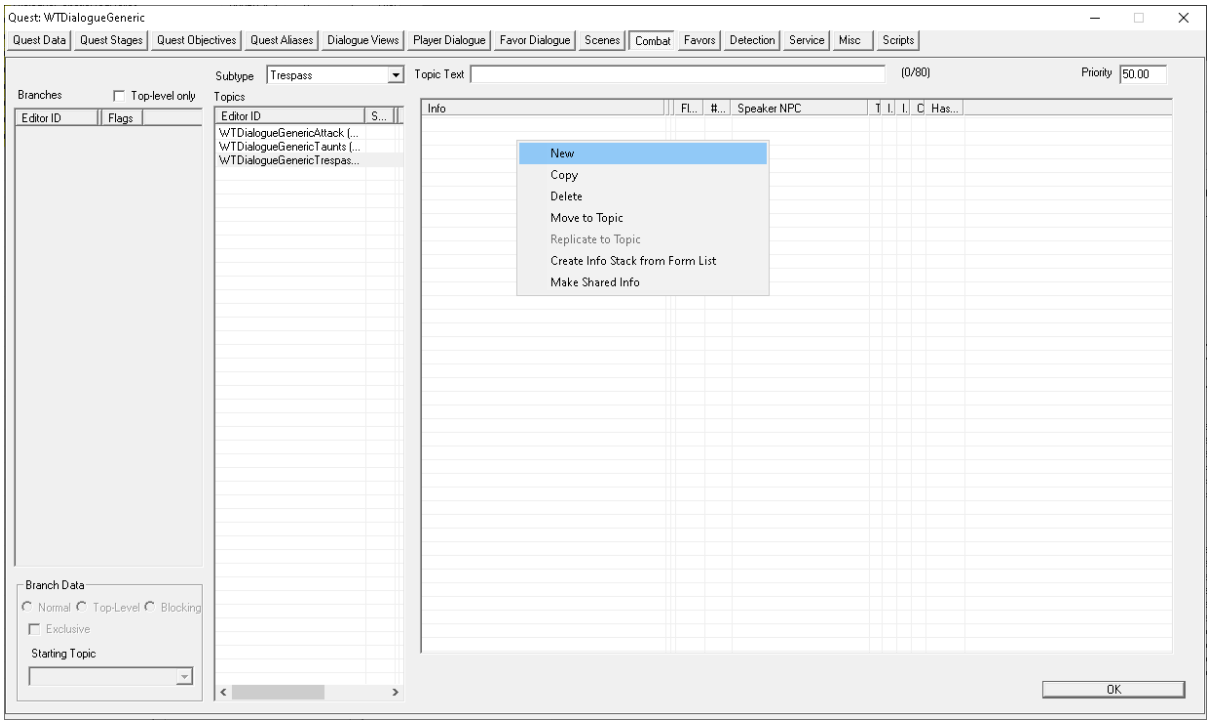


Figure 833 - Adding a trespass response.

Warning level 0 is the first line that's spoken by an NPC when the player is caught trespassing. It should go something along the lines of "You're not supposed to be here".

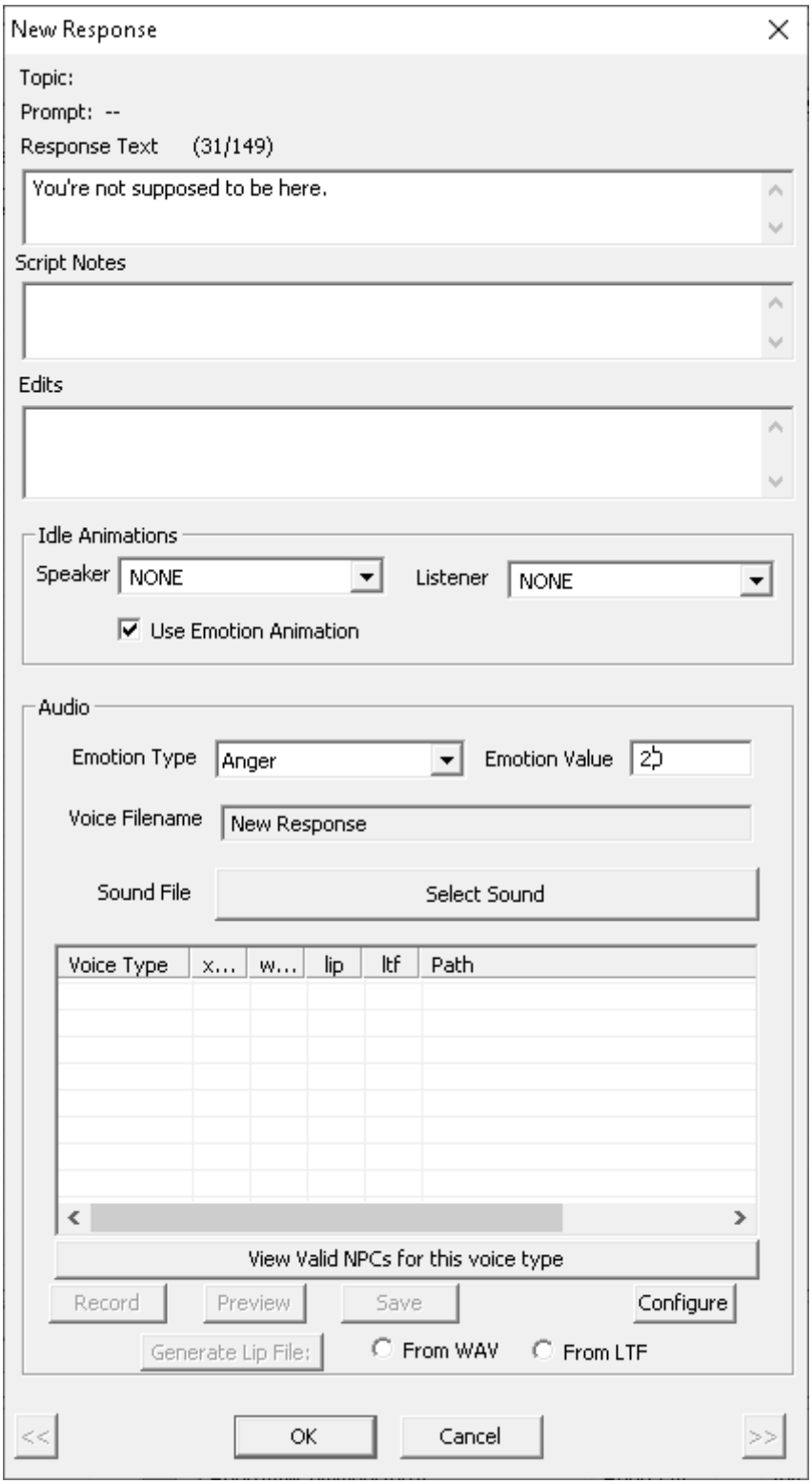


Figure 834 - Response Text for a level 0 trespass warning.

Generally we won't need to tick the Random button unless you're setting up multiple responses per warning level.

We'll need to add two additional conditions for each trespass warning.

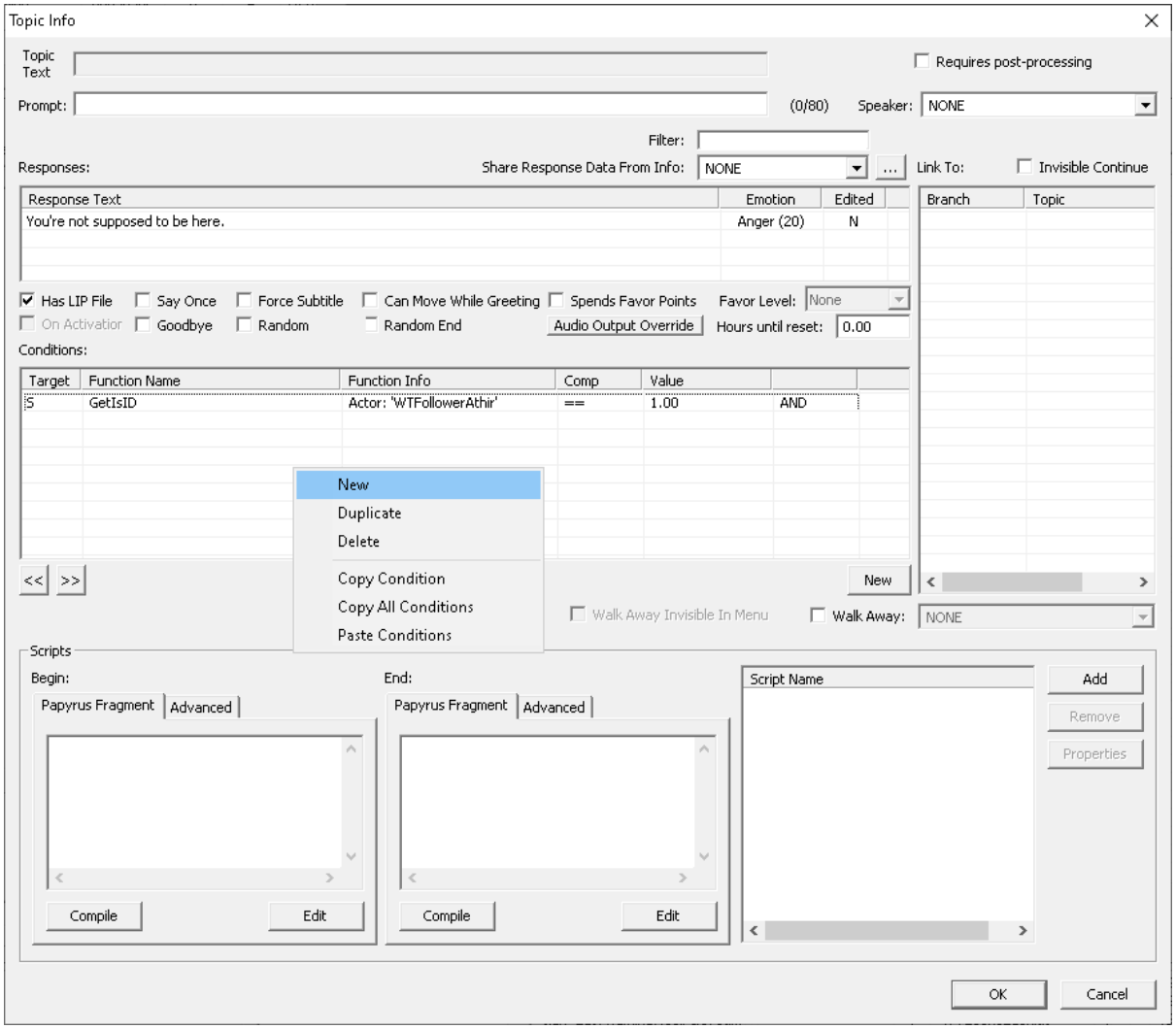


Figure 835 - Adding the first trespass condition.

The first condition to check whether the player is trespassing. We can use the IsTrespassing condition function to do this.

Set the Value field to 1 then click OK.

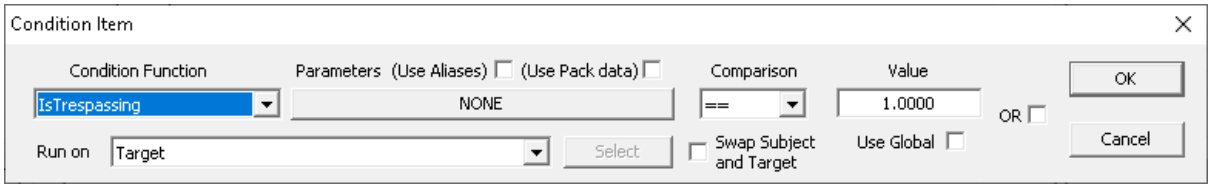


Figure 836 - IsTrespassing condition function.

The second condition is used to check the current trespass warning level. We can use the GetTrespassWarningLevel condition function to do this.

For the first trespass warning, set the Value field to 0 then click OK.

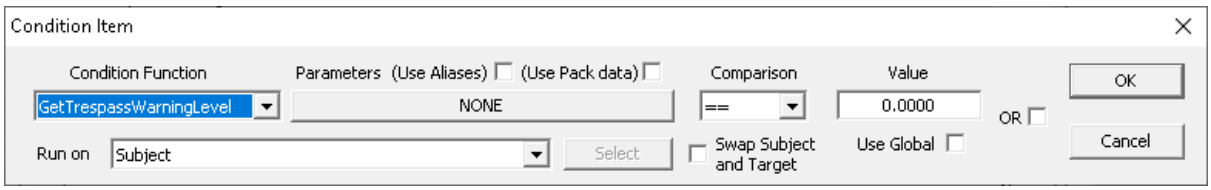


Figure 837 - GetTrespassWarningLevel condition function.

Our first trespass warning should look something like this:

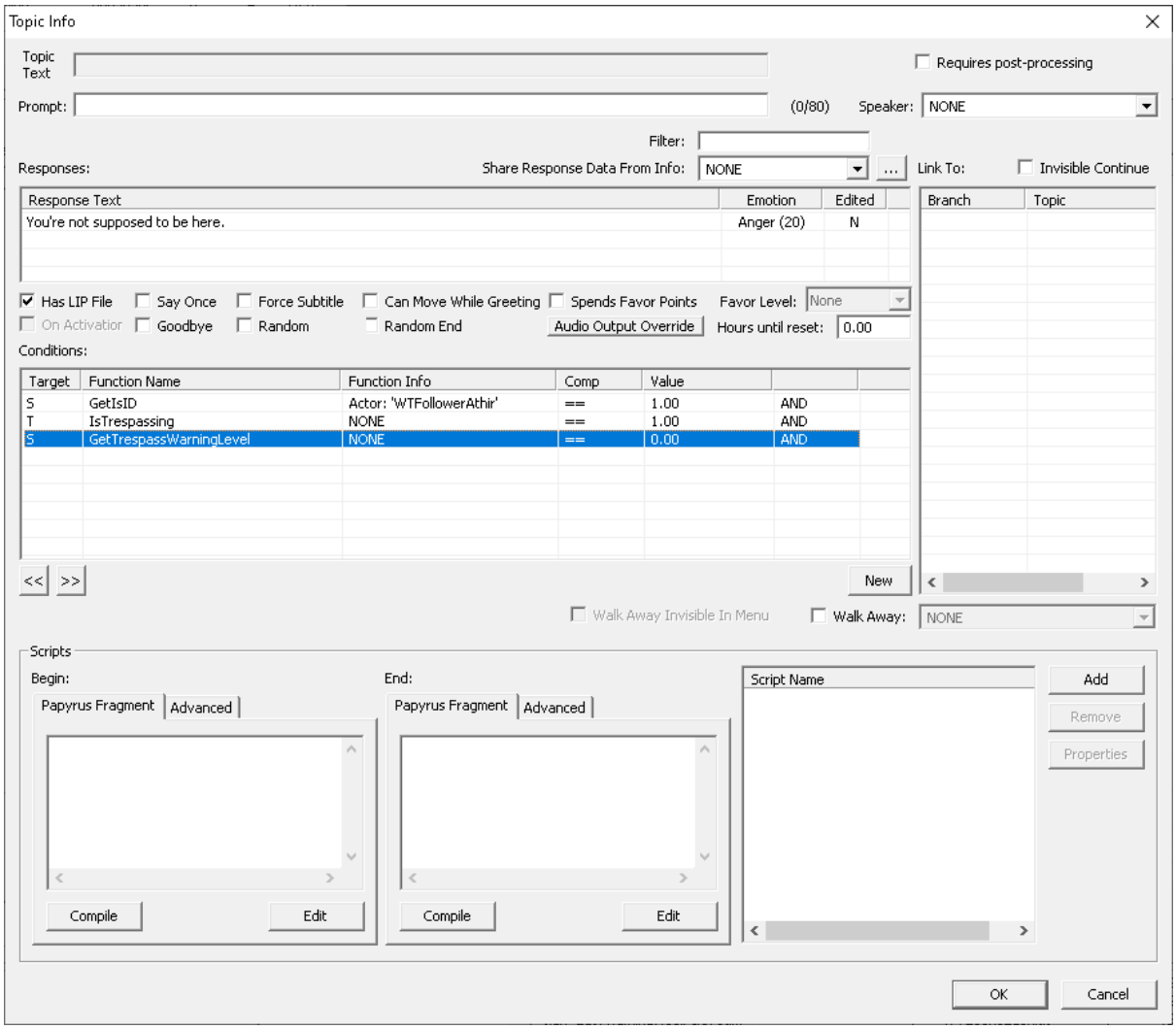


Figure 838 - Level 0 trespass warning set up.

The level 1 trespass warning comes a few moments after the first warning if the NPC detects that the player is still trespassing.

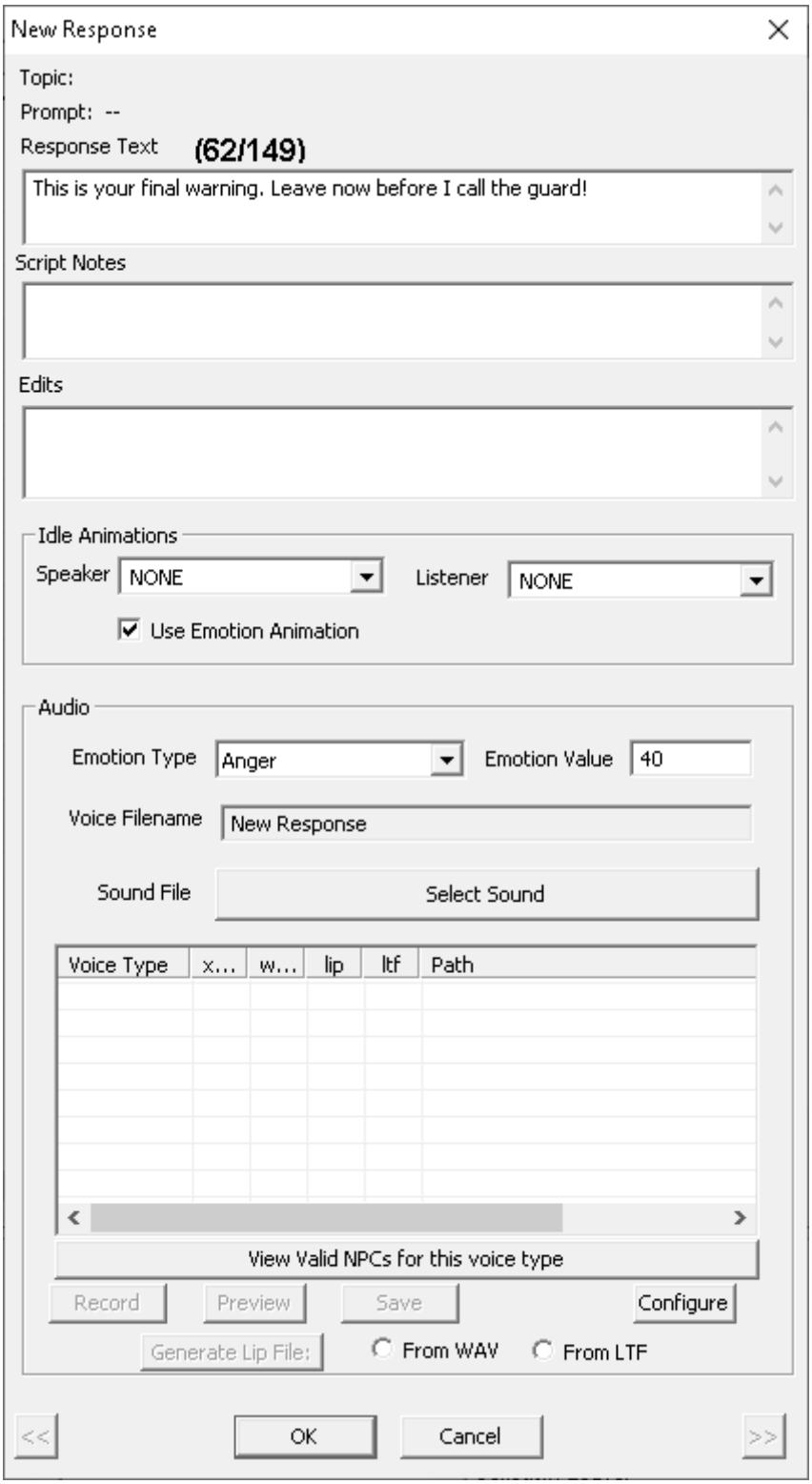


Figure 839 - A level 1 trespass warning.

For the level 1 warning, set the GetTrespassWarningLevel condition value to 1.

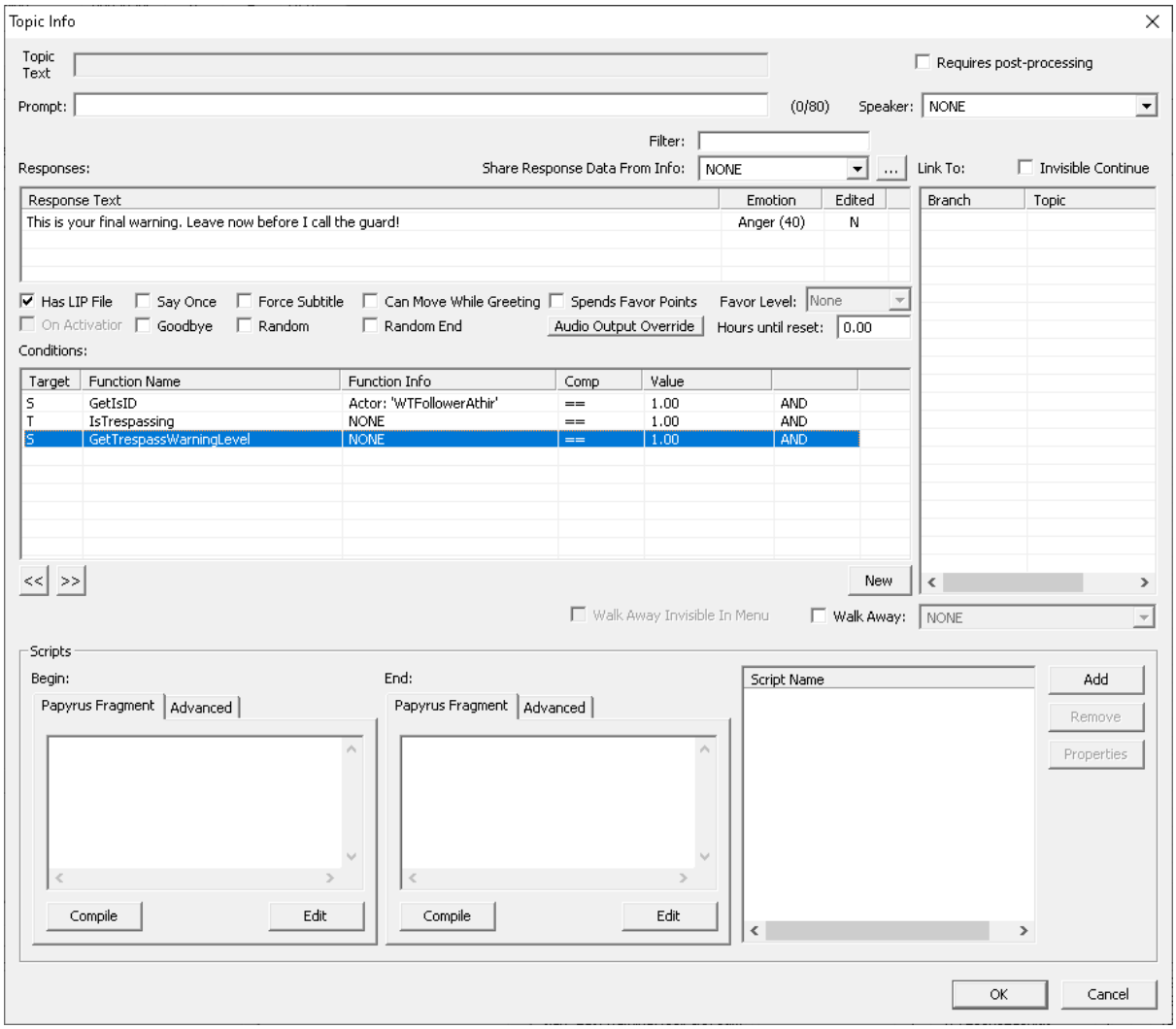


Figure 840 - Level 1 trespass warning set up.

Last is the level 2 warning at which point the guards are called to arrest the player.

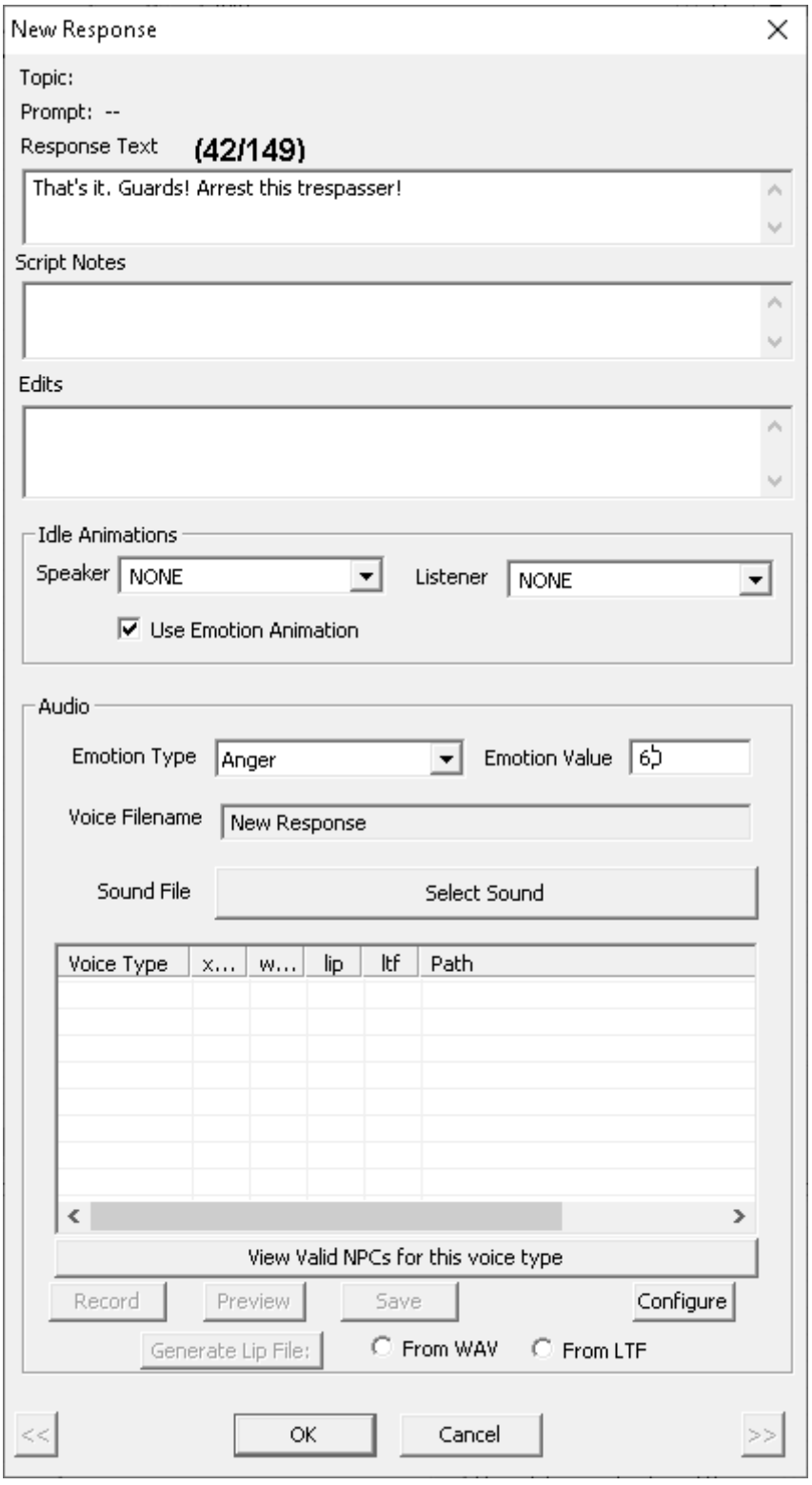


Figure 841 - Level 2 trespass warning.

Set the GetTrespassWarningLevel condition value to 2.

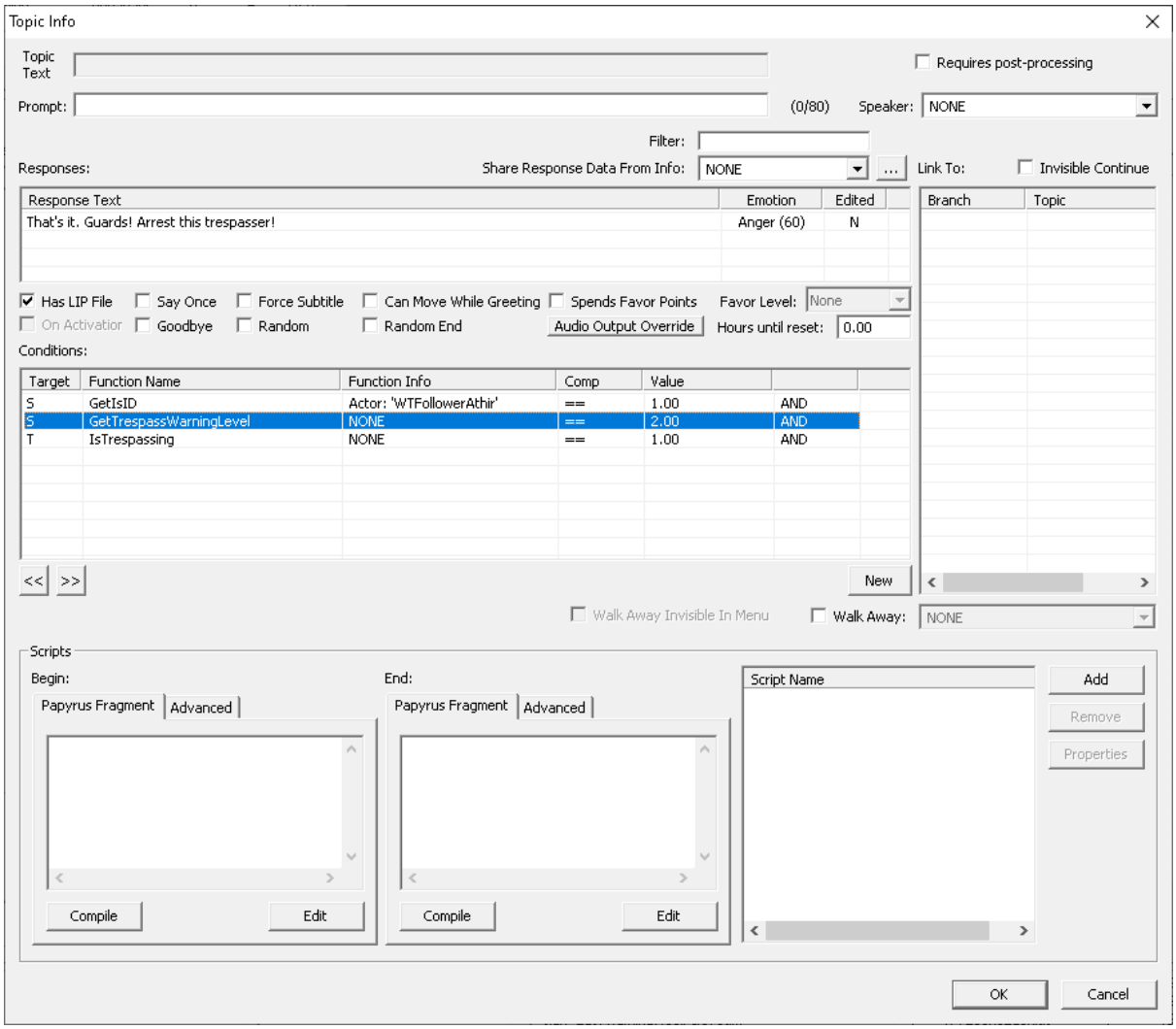


Figure 842 - Level 2 trespass warning set up.

The screenshot below shows trespass warnings set up for NPCs in Wyrmstooth. Each NPC has three trespass lines, one for each warning level.

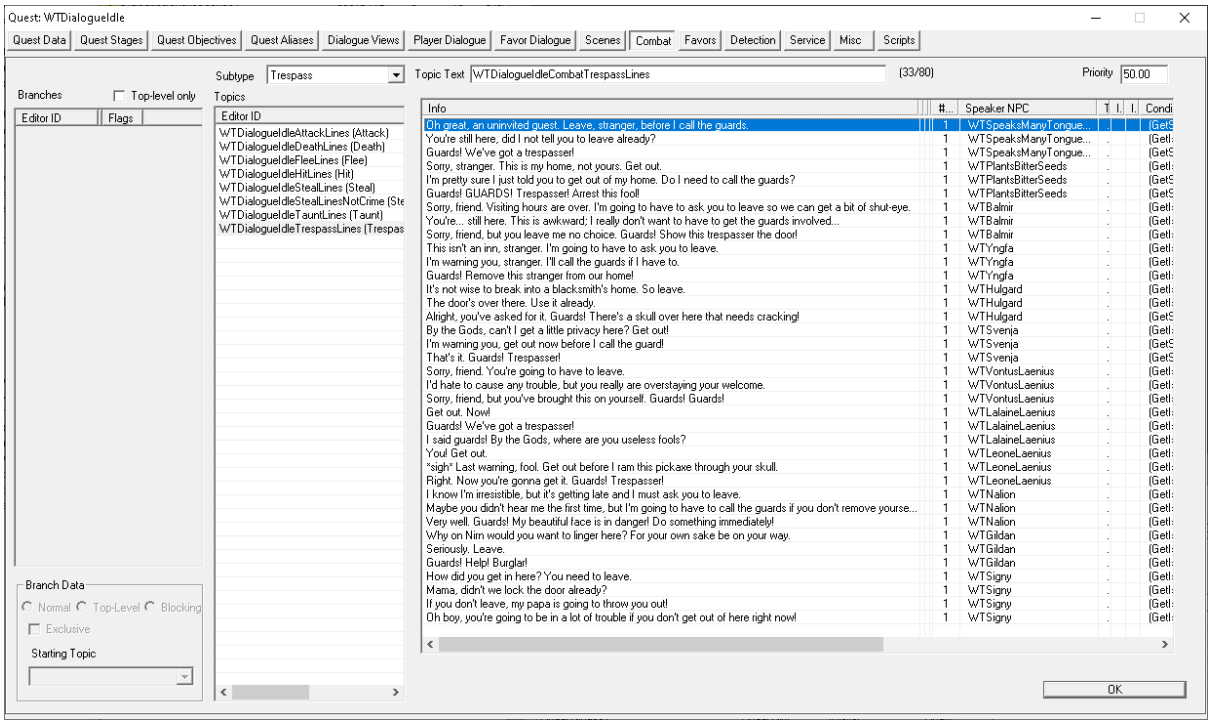


Figure 843 - Sample trespass warnings.

For more information on each topic type see the Combat tab article on the Creation Kit wiki.

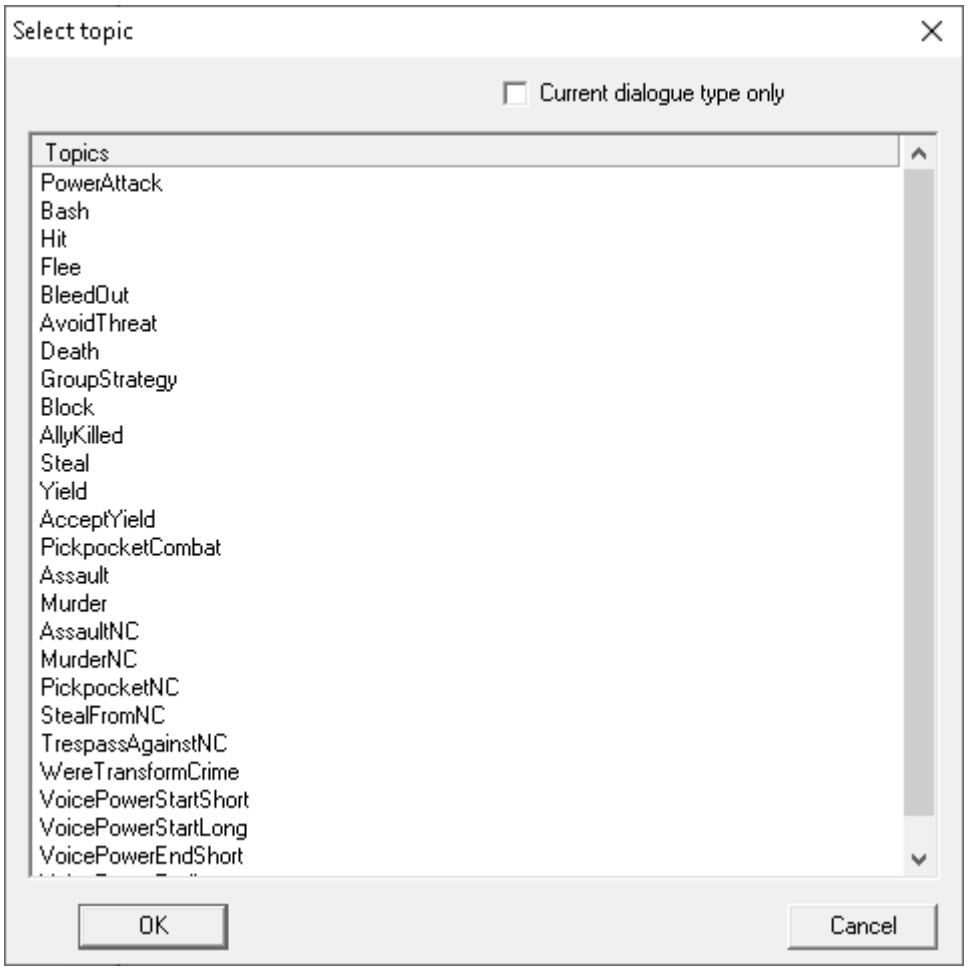


Figure 844 - List of topic types for the Misc tab.

AllyKilled, AvoidThreat, GroupStrategy and Yield aren't used. If you want the NPC to respond to a yield request with custom dialogue, use AcceptYield instead.

ADDING SCENES

Scenes are used to carefully choreograph NPC actions. Scenes are divided into phases and an NPC can be assigned a different package or dialogue action on each phase.

Before a phase can end, all actions defined in a phase need to complete.

For example, the intro carriage ride at the start of the game is set up using a scene.

In Wyrmstooth, a scene is triggered when you encounter Vulthurkrah for the first time on Ancient's Ascent.

In this section I'll be showing you how to set up a simple scene involving a couple NPCs.

Note: I'll be reusing the same sample quest WTTestQuest that I set up in the previous sections of this guide. If you want to create a new quest for this scene, just be sure to generate a new .seq file once this is done.

Go to the Scenes tab in Quest properties.

Right-click in the scenes list on the left hand side and select New.

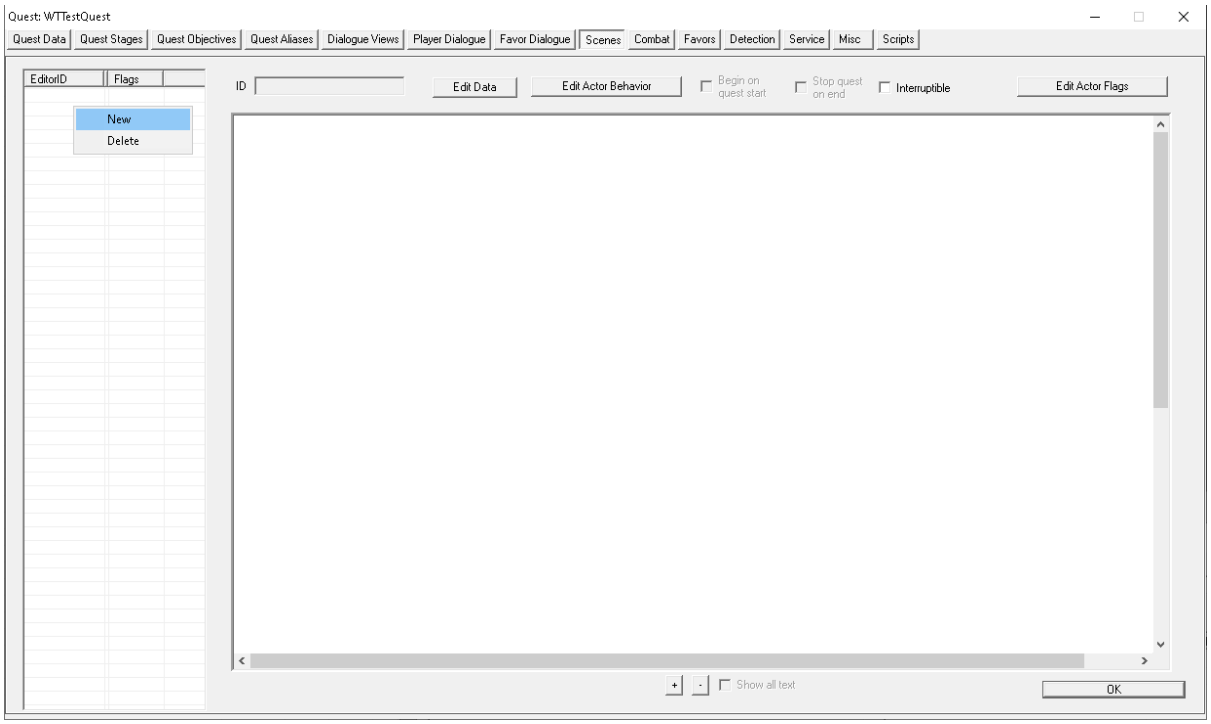


Figure 845 - Scenes tab.

Enter in a scene ID then click OK. For this example, I'm just going to call it WTTestQuestSampleScene.

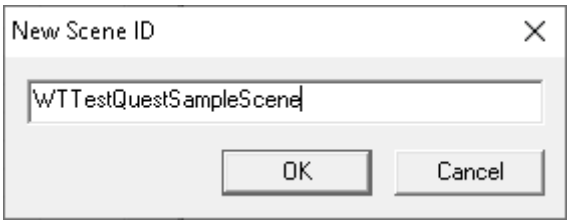


Figure 846 - Scene ID.

Click on the new scene in the scenes list to highlight it, then right-click in the scene layout section to the right and select New Actor.

Note: Only NPCs added to your Quest Alias tab can be chosen here.

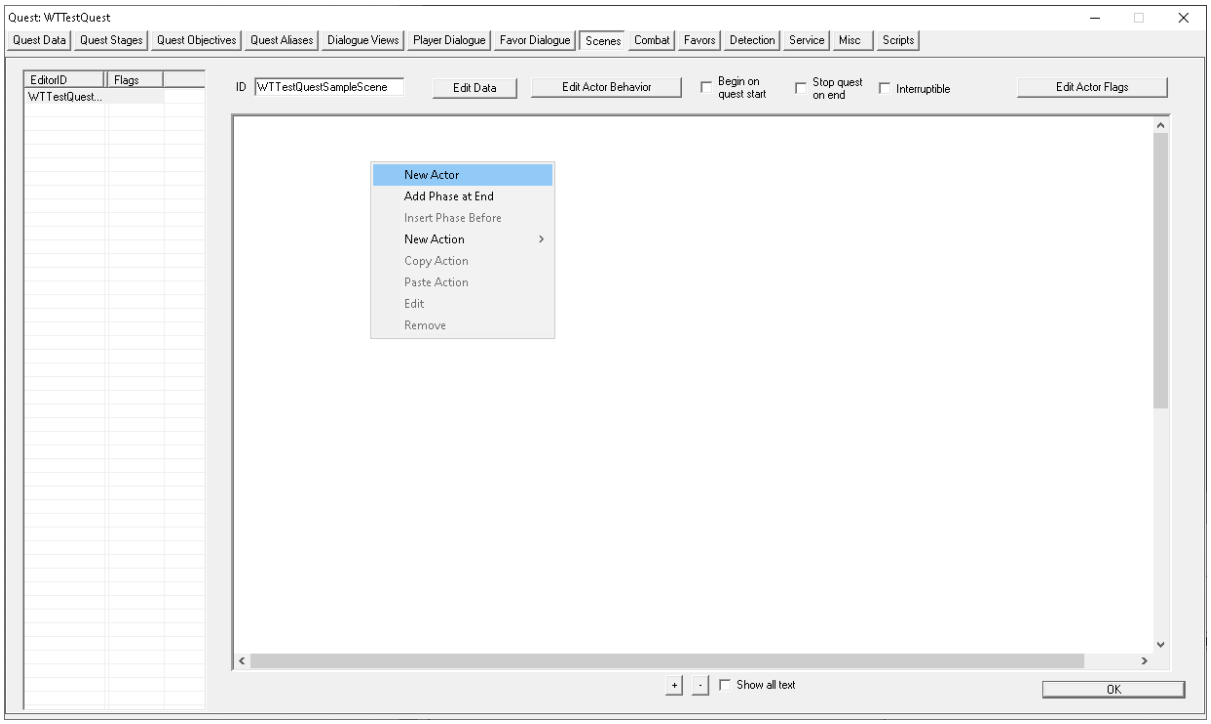


Figure 847 - Adding a new actor.

Select the NPC to add from the list of quest aliases then click OK.

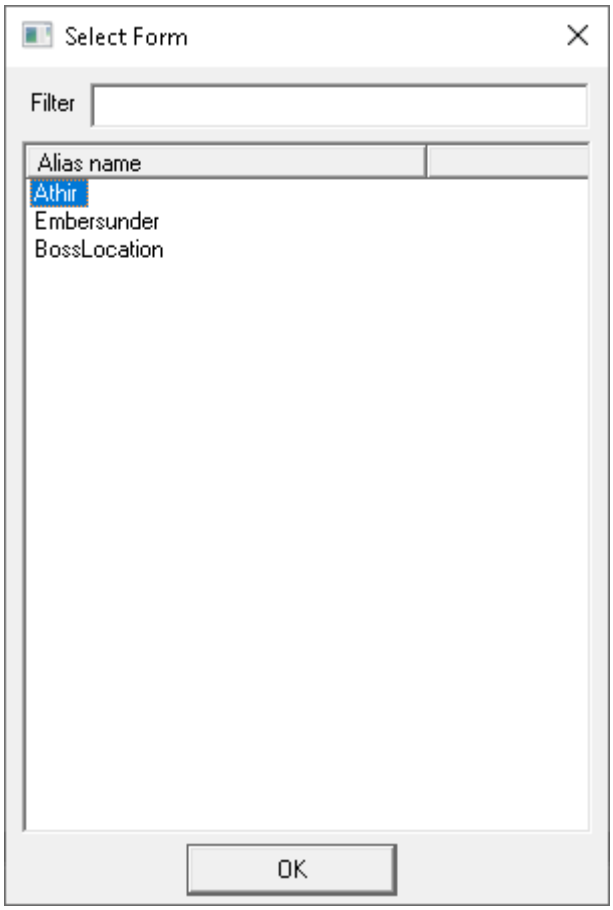


Figure 848 - Selecting the first NPC.

The NPC should appear in the scene layout as per the screenshot below:

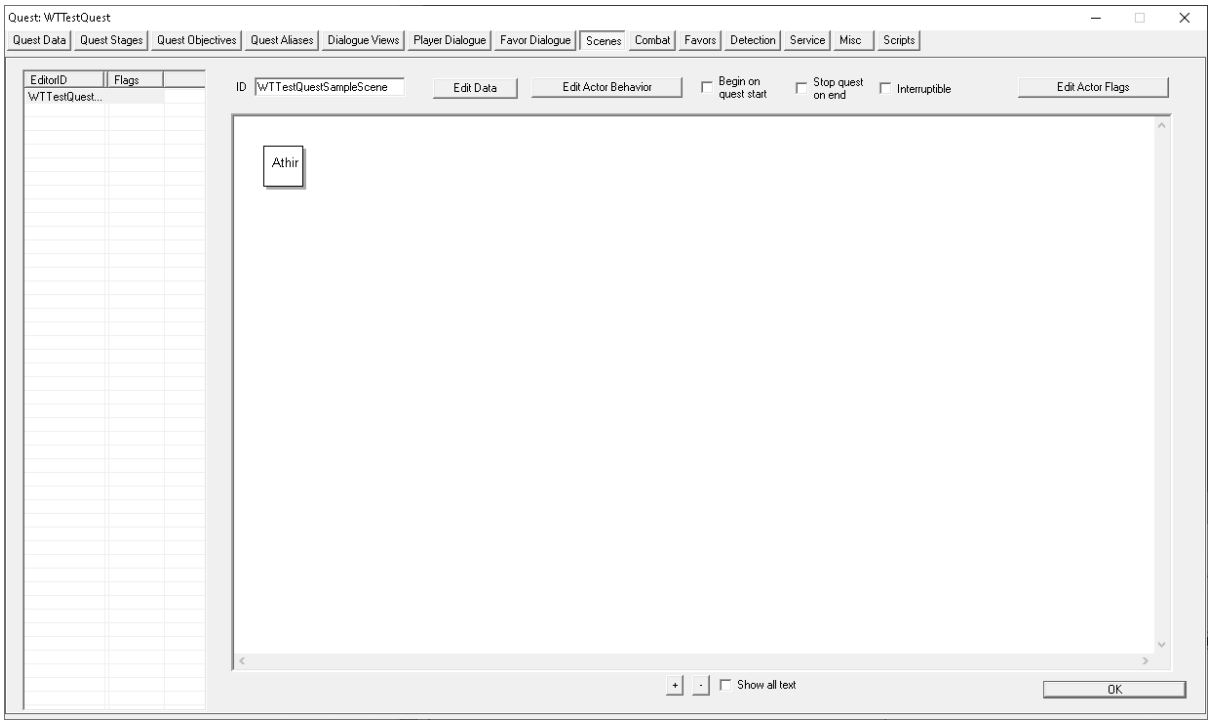


Figure 849 - An NPC added to the current scene.

Our scene will need to consist of at least one phase. To add a phase, right-click and select 'Add Phase at End'.

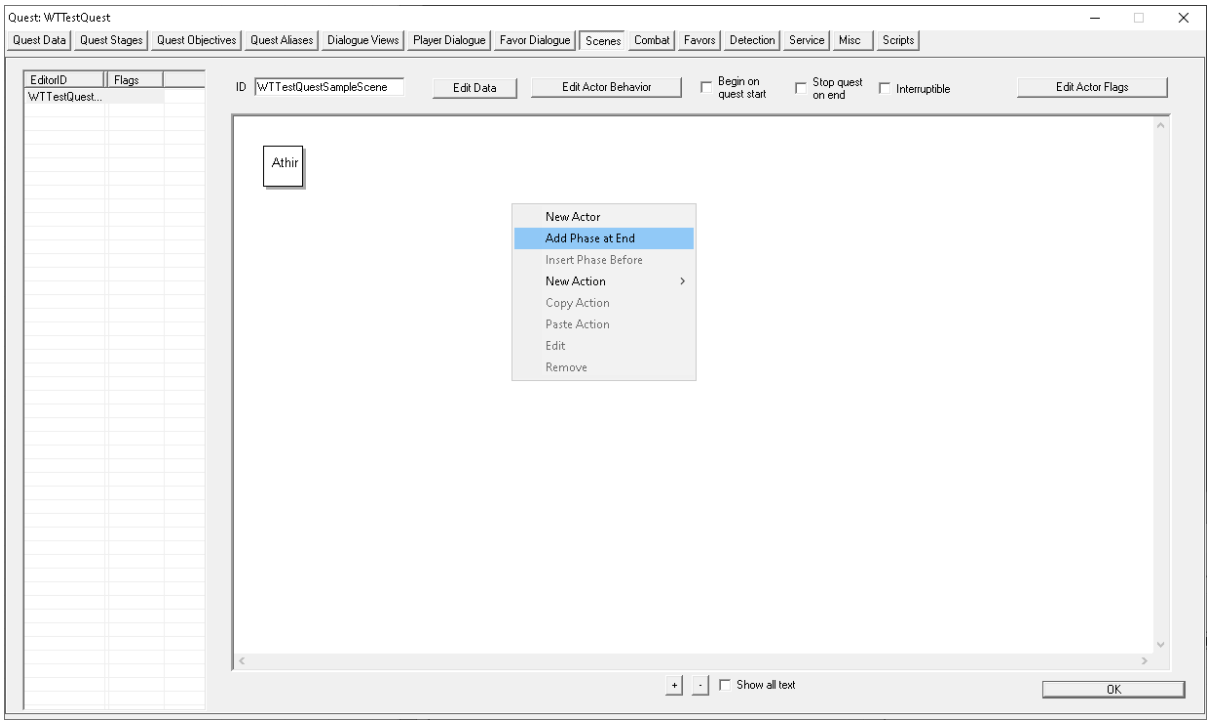


Figure 850 - Adding the first phase.

Our scene layout should now look like this.

Currently we haven't assigned any actions to Athir, so let's give him something to do next.

In this example, I'm going to give him some banter with the other mercenaries following the player through Wyrmstooth Barrow.

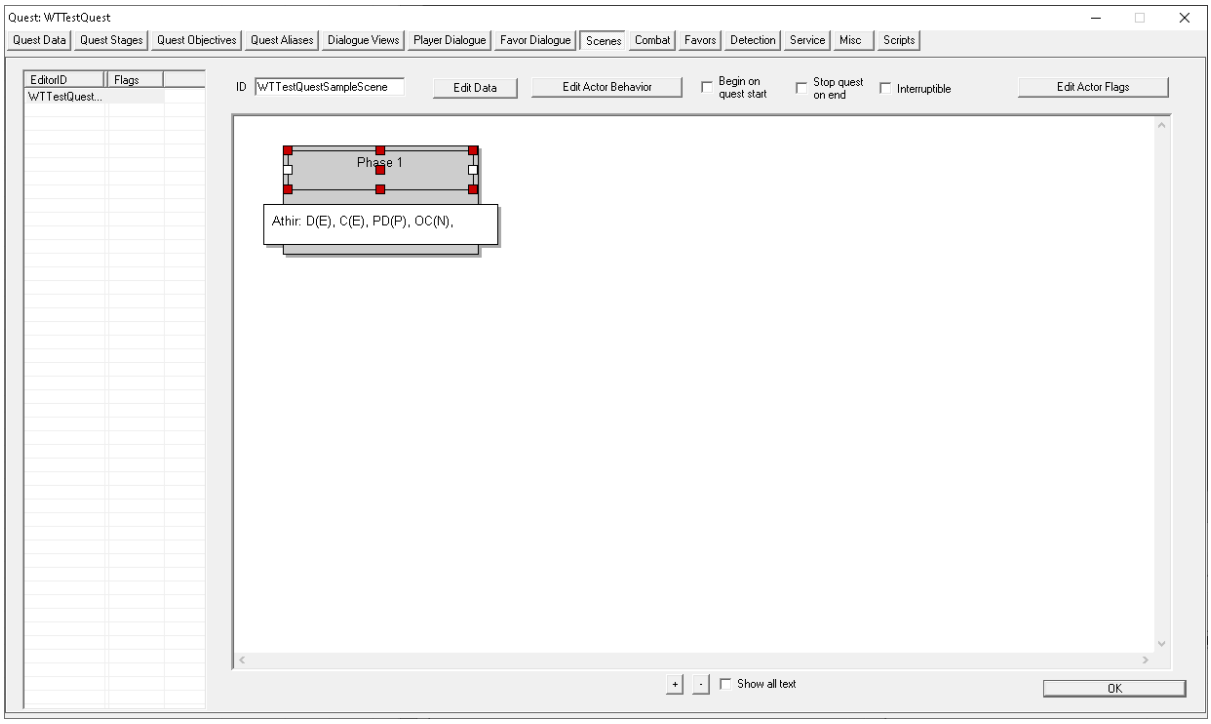


Figure 851 - First phase added.

I added a couple XMarkerHeadings in the render window that I'll be using in this example scene to tell the NPCs involved where to stand.



Figure 852 - XMarkerHeadings that'll be used for the example scene.

To tell Athir where to stand when the scene starts up, right-click on him in the scene layout and go to New Action > Package.

Important: Avoid assigning non-completing packages to NPCs in a scene. Some packages such as Sandbox never complete, so the phase they're assigned on won't complete and the scene will stall.

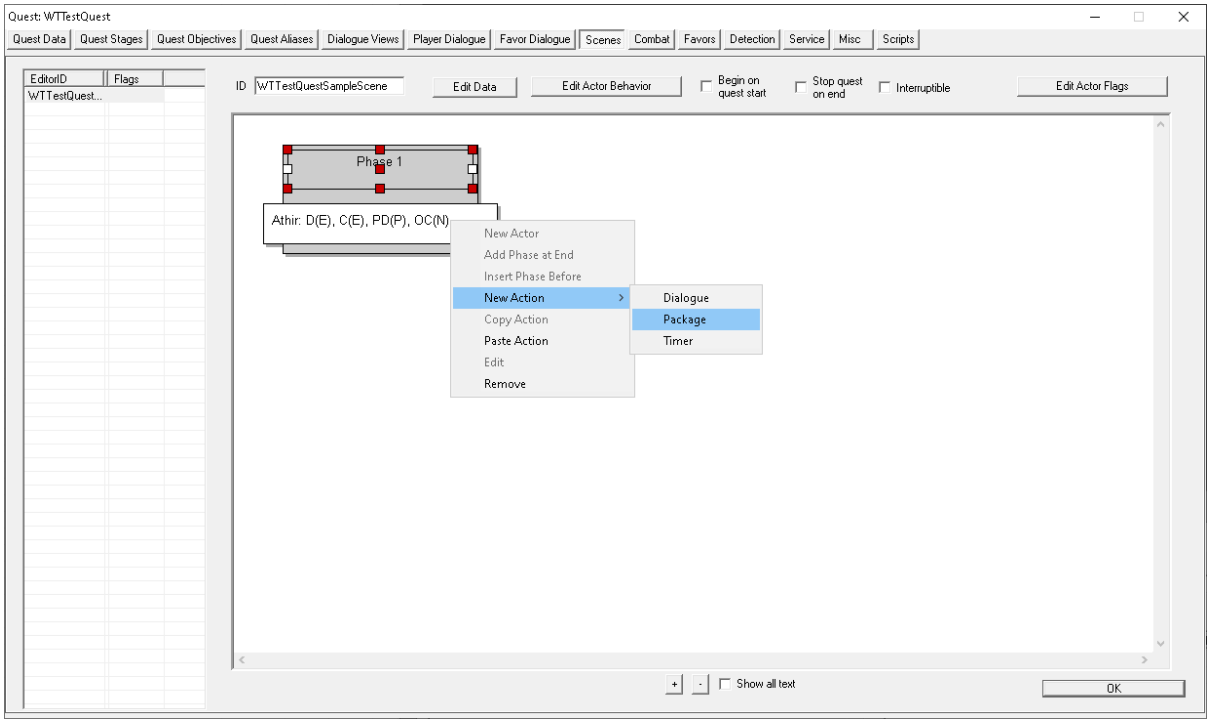


Figure 853 - Adding a package to Athir for phase 1.

Right-click in the package list and select New.

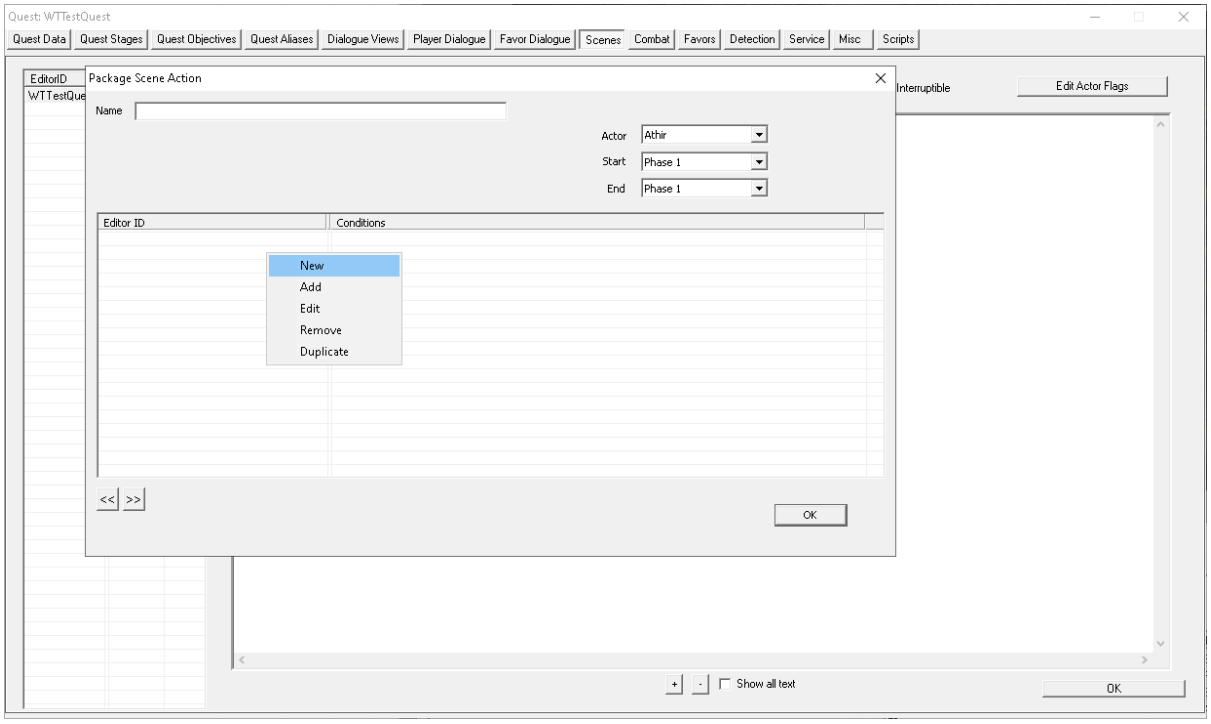


Figure 854 - Creating a new package.

Enter in the package ID. I just called it WTTestSceneAthirWalkToPackage for this example.

Make sure the 'Owner quest' drop-down is set to the name of the quest you're setting up the scene in.

In this example, I set up a new package using the Travel template and made Athir travel to the XMarkerHeading facing the left wall.

Click OK to add the new package to Athir for phase 1 of the scene.

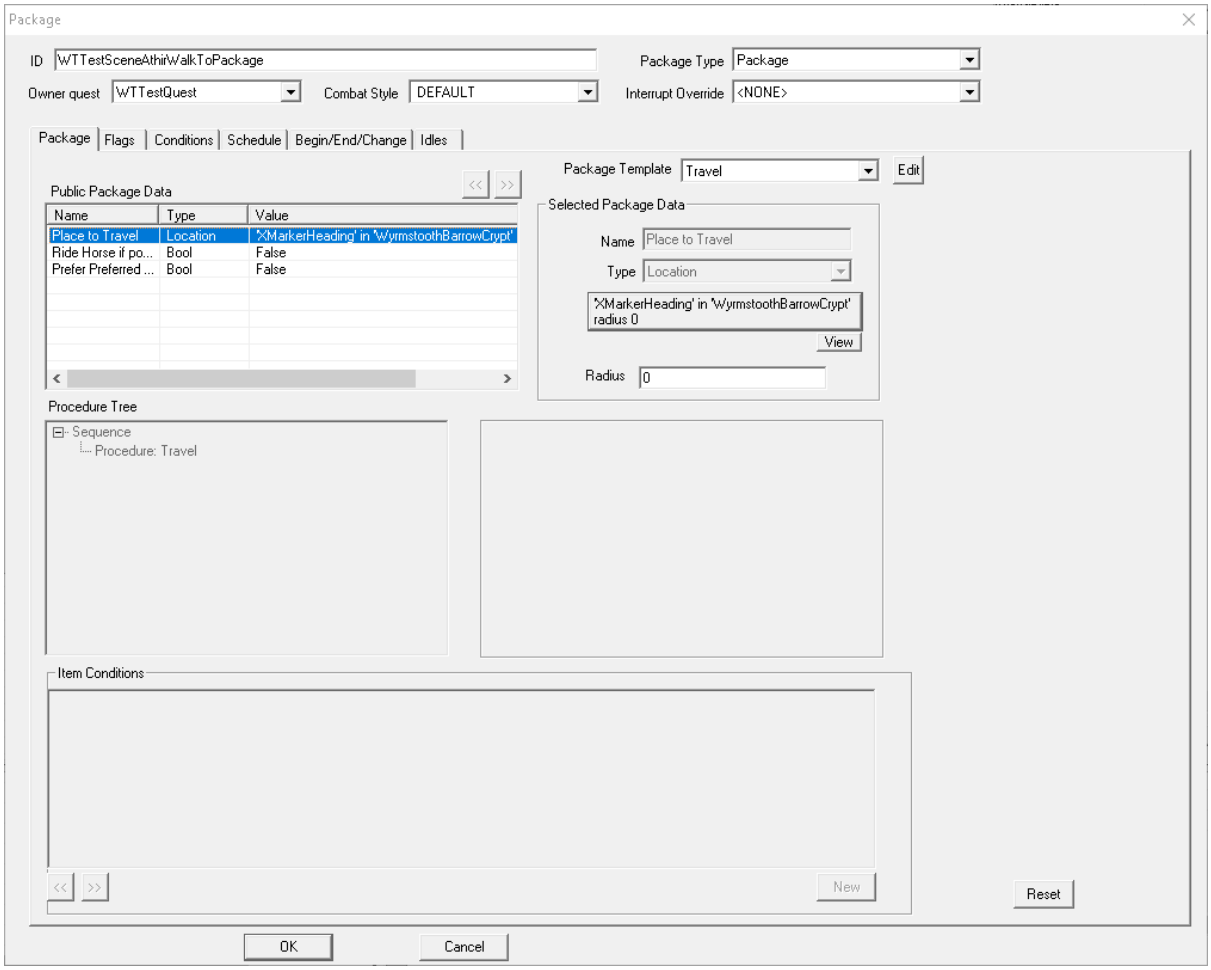


Figure 855 - Athir's scene package.

Click OK to close out of Package Scene Action properties.

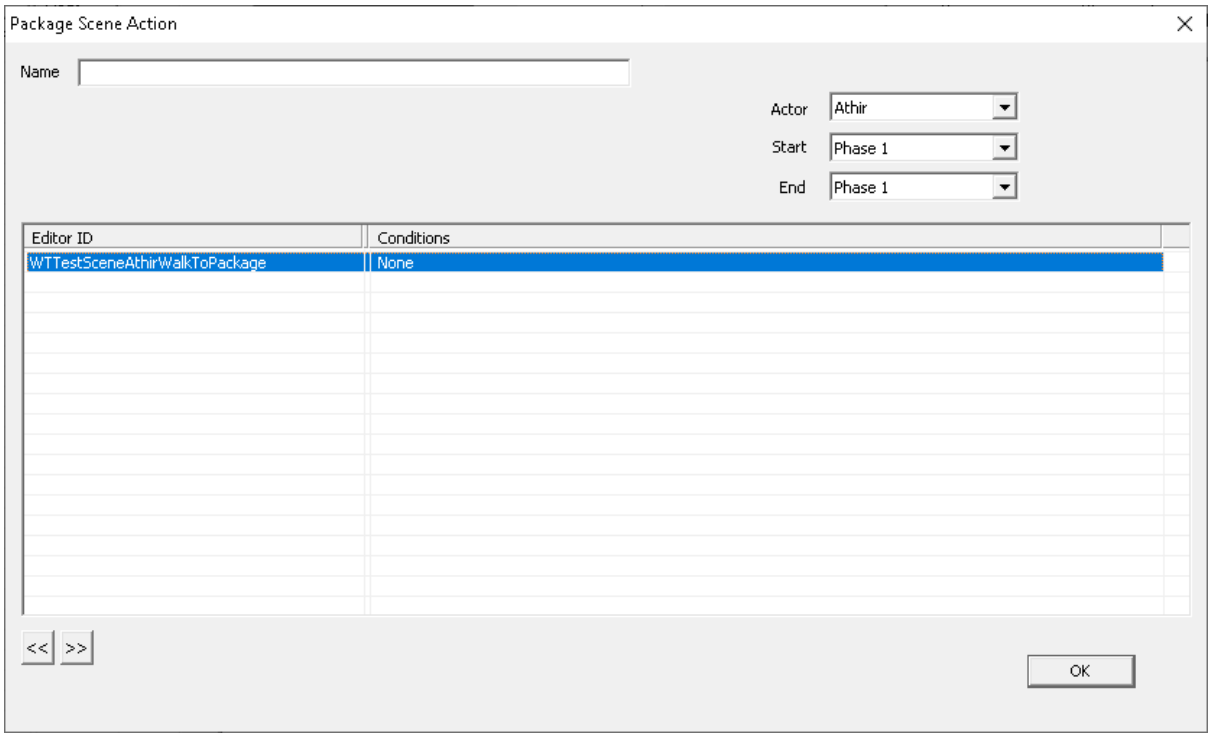


Figure 856 - Package Scene Action properties.

Now Athir has a package for phase 1.

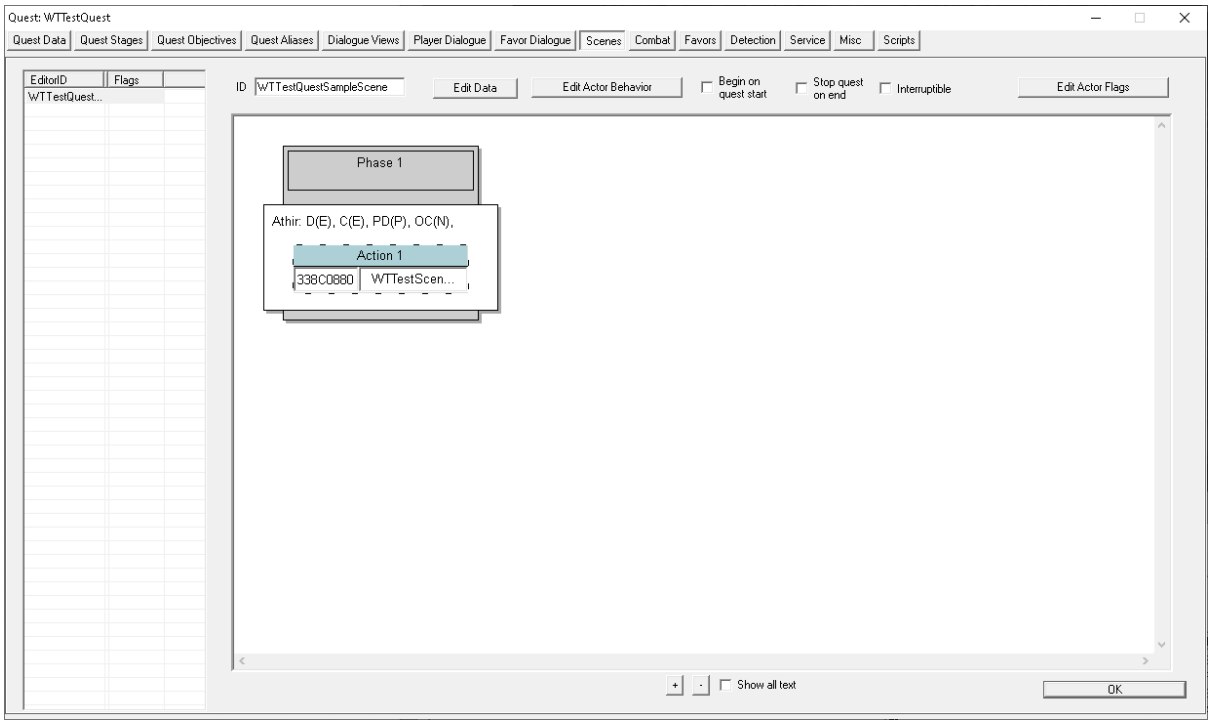


Figure 857 - Athir with a package set up for phase 1.

Next, let's give him something to say while he walks over to the XMarkerHeading.

Right-click on Athir again. This time select New Action > Dialogue.

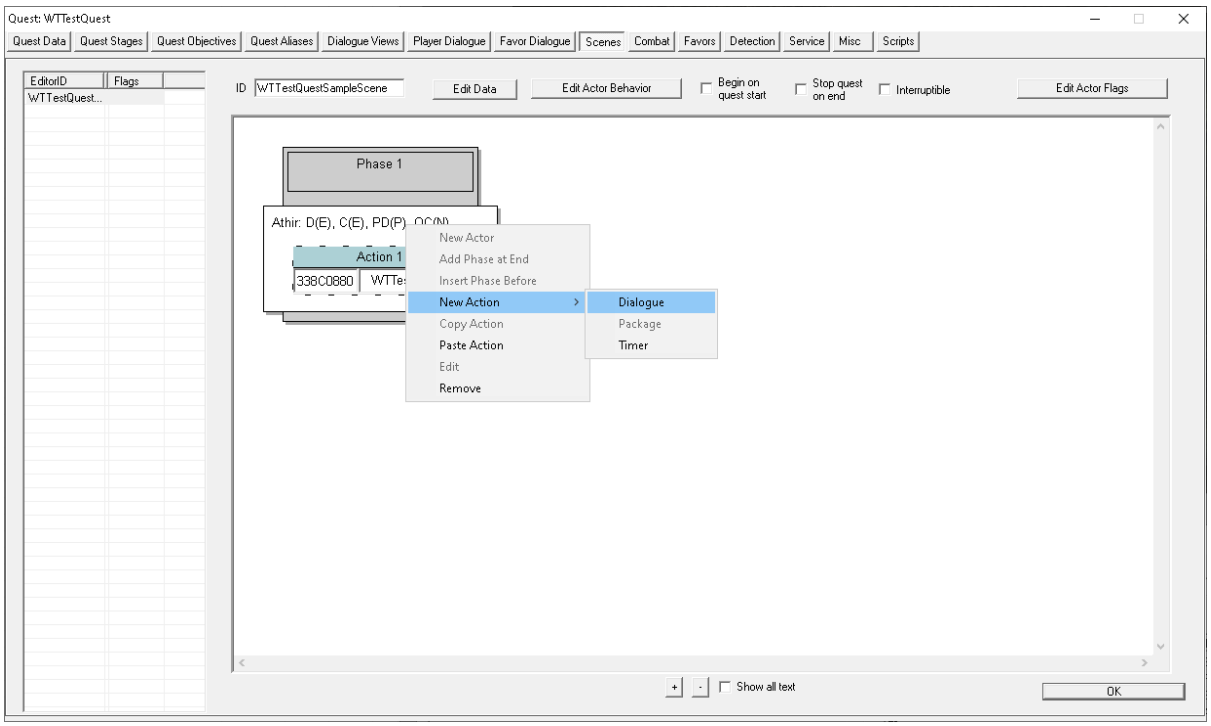


Figure 858 - Adding a new dialogue scene action.

Right-click in the dialogue list and select New.

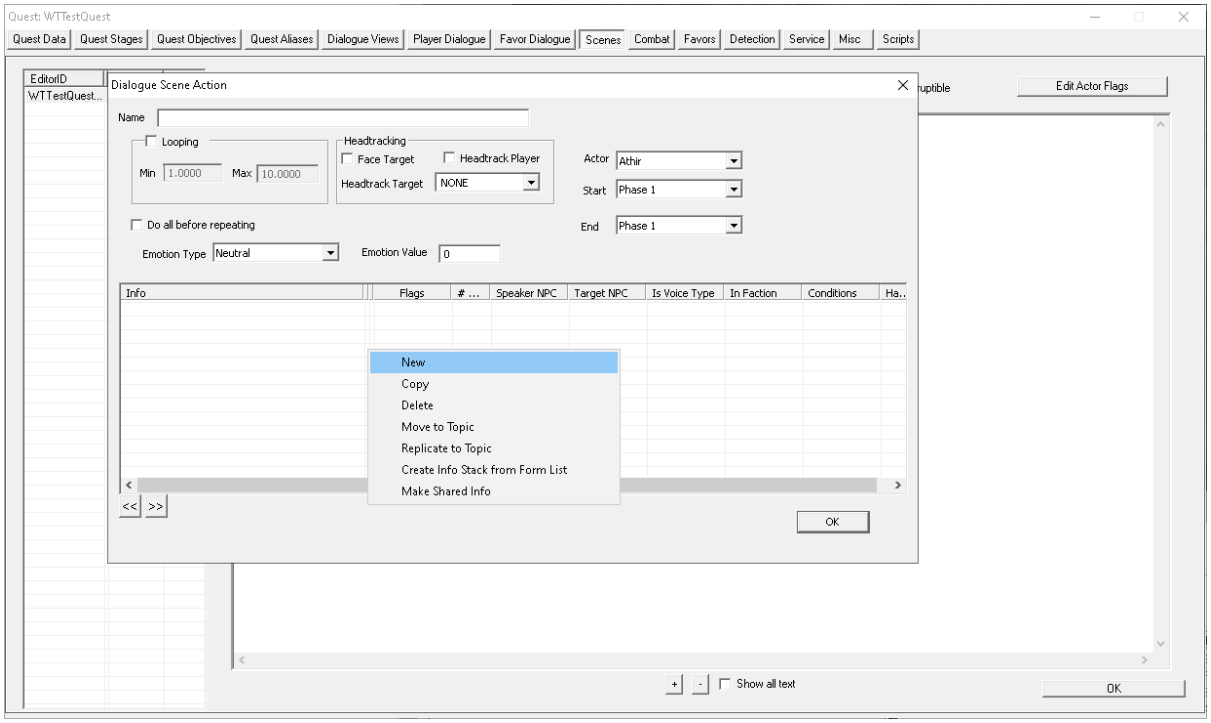


Figure 859 - Adding a new line of dialogue for phase 1.

Enter in dialogue in the Response Text field. Change the Emotion Type and Emotion Value if necessary then click OK.

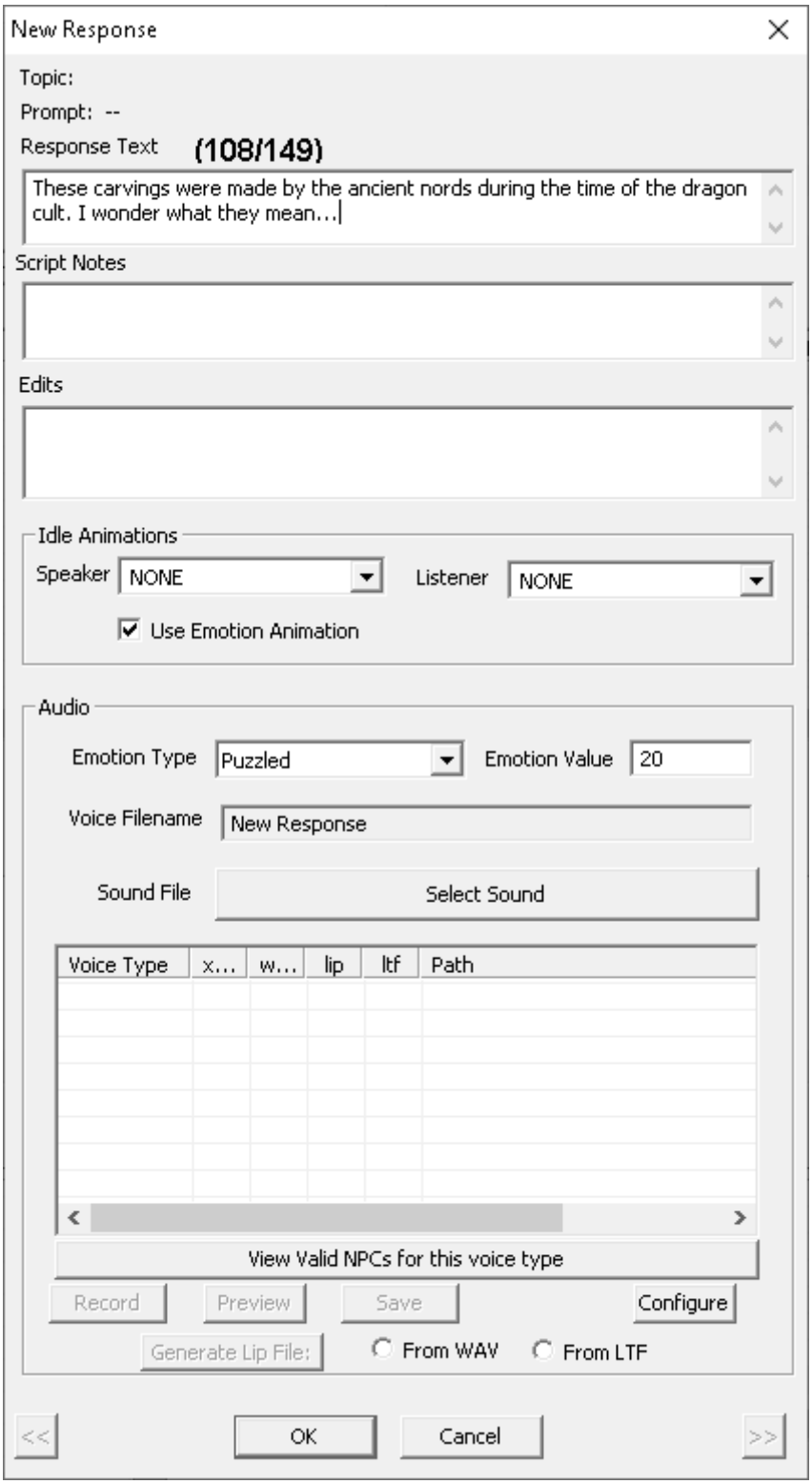


Figure 860 - Adding response text.

Right-click in the conditions list and select New.

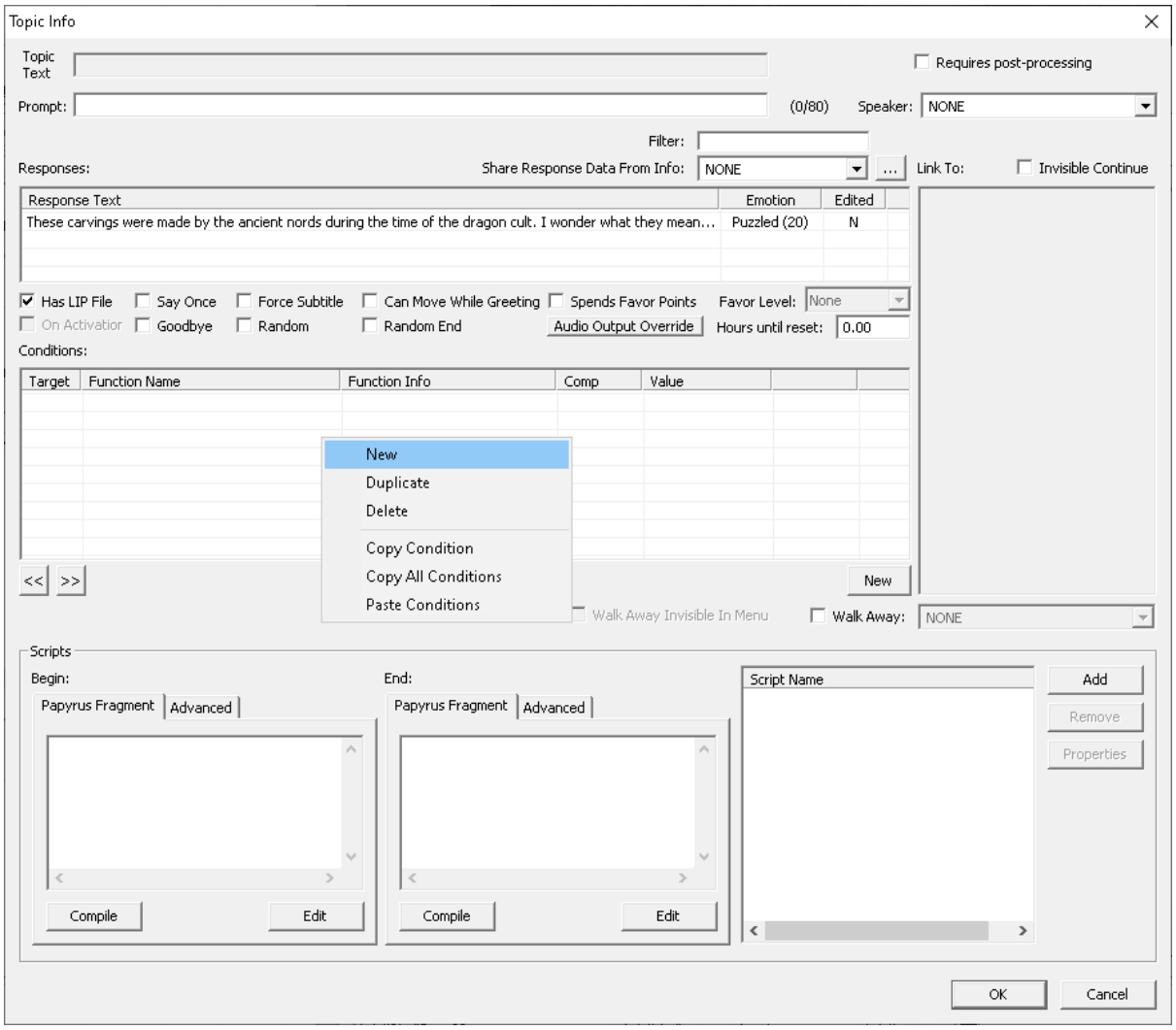


Figure 861 - Adding a new dialogue condition.

I added a GetIsID condition to ensure only Athir can speak this line of dialogue.

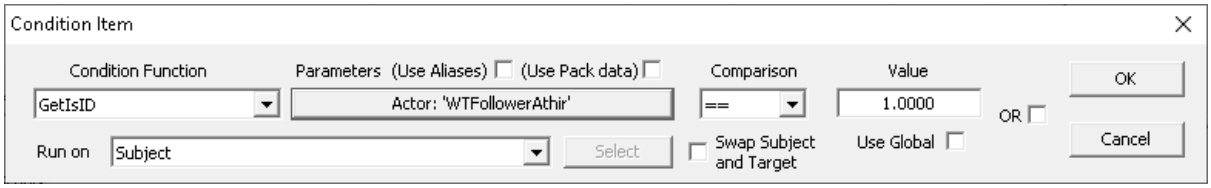


Figure 862 - GetIsID condition.

Once the condition has been added, click OK.

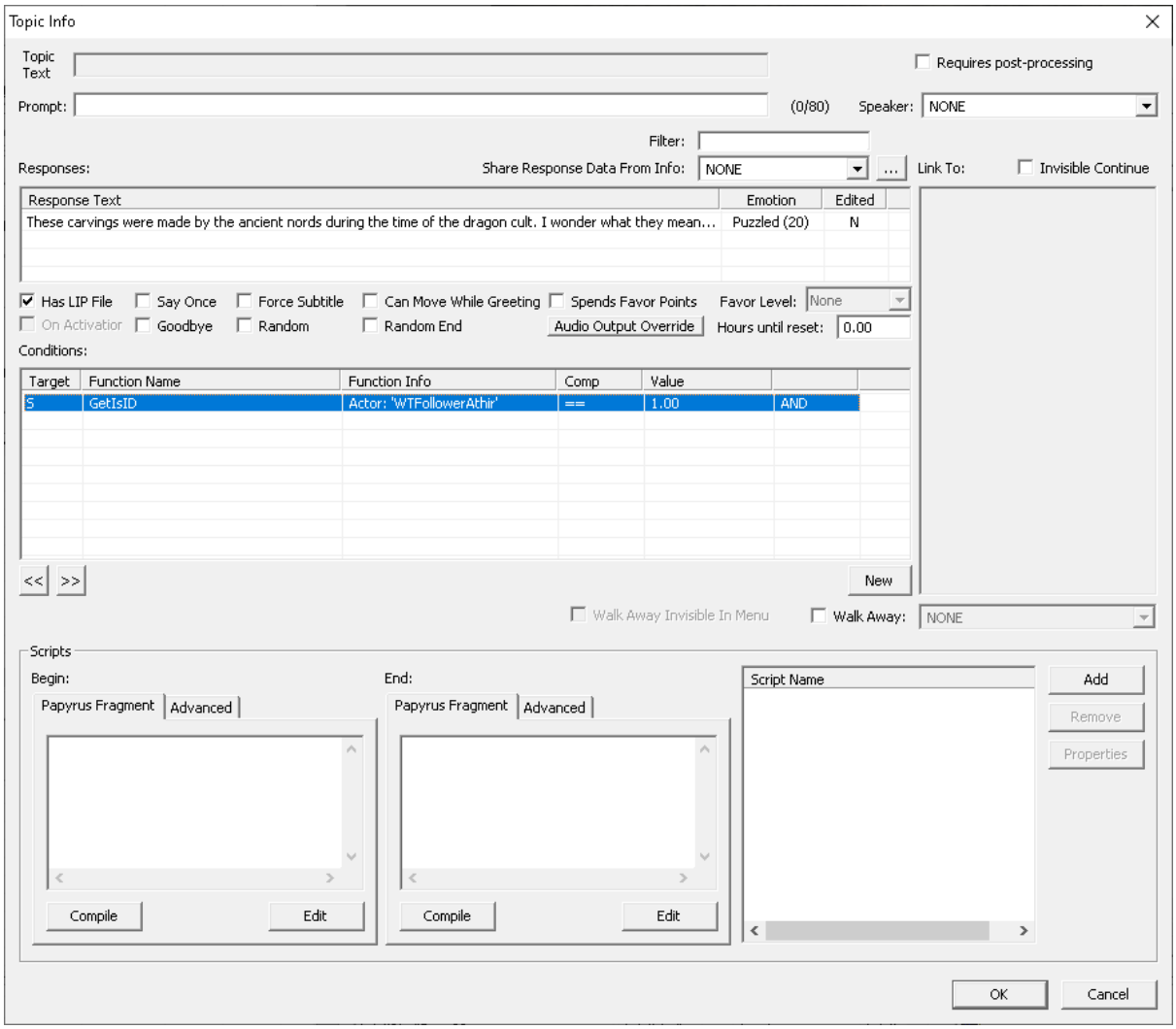


Figure 863 - The first line of dialogue for the scene set up.

Click OK to close out of Dialogue Scene Action properties.

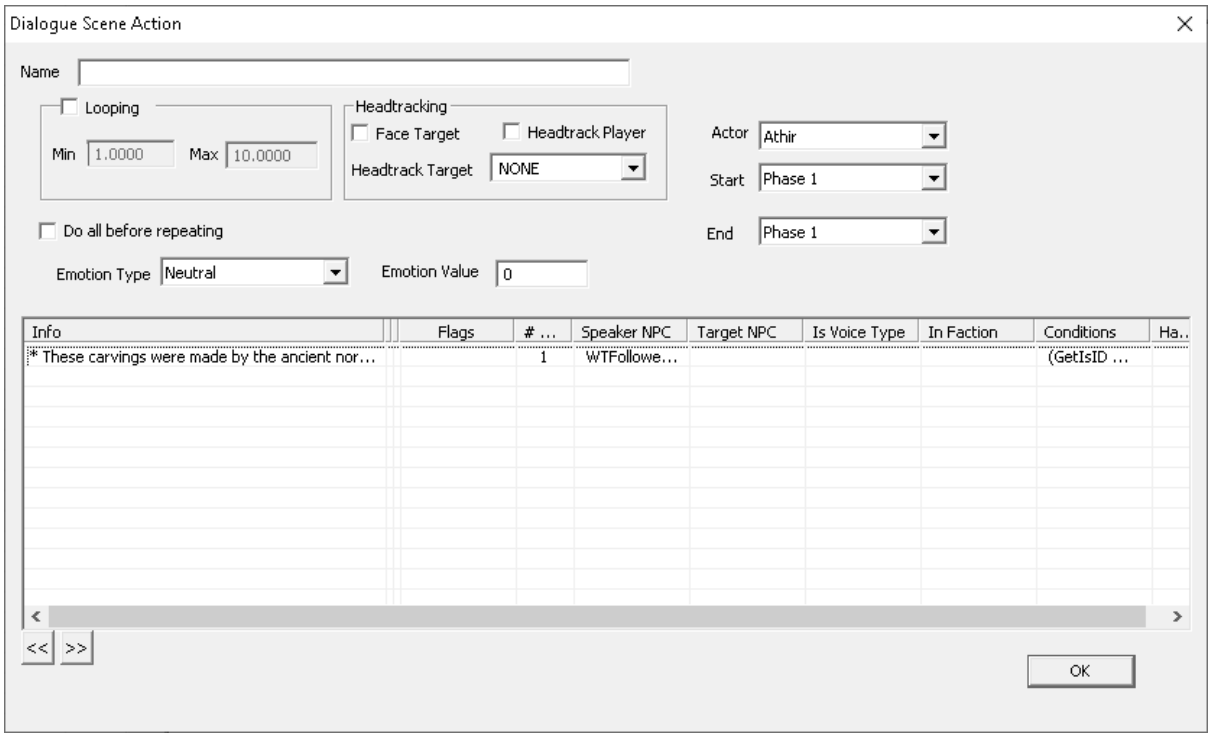


Figure 864 - Dialogue Scene Action with a line of dialogue set up.

When this scene starts up, Athir should now walk over to the XMarkerHeading on the left while commenting about the carvings.

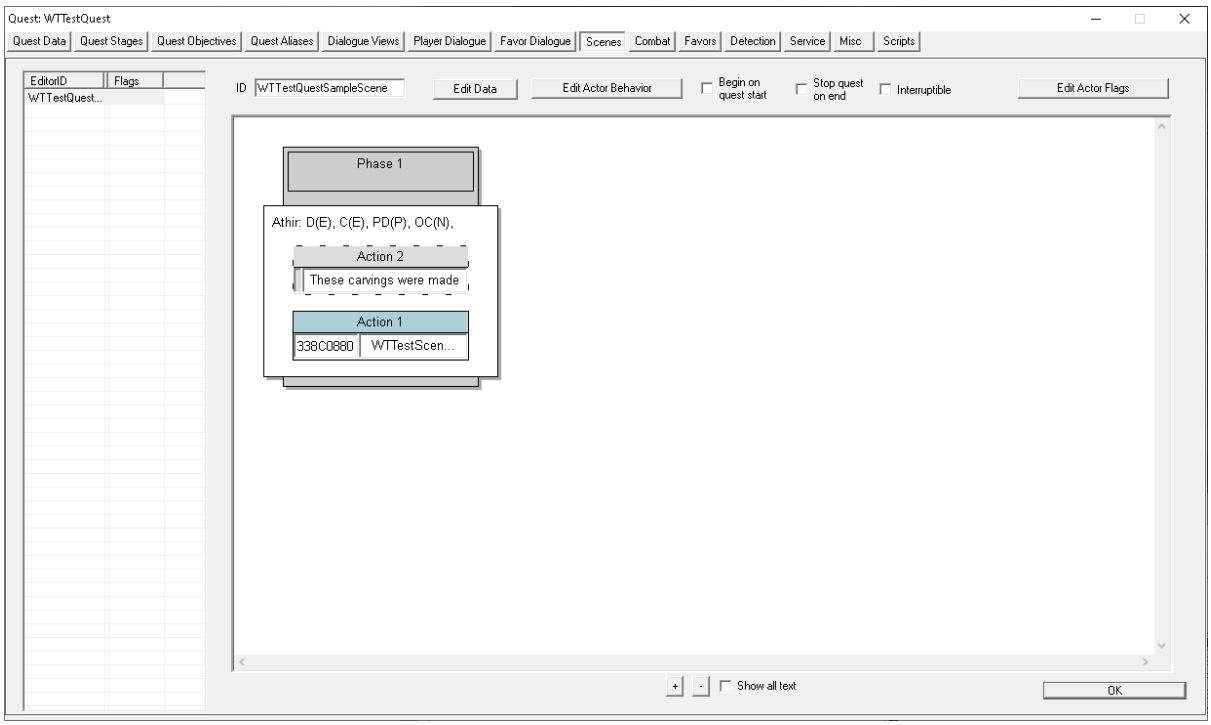


Figure 865 - Phase 1 set up with a line of dialogue and a package.

Let's add in another NPC to get a bit more banter going.

Go to the Quest Aliases tab, right-click in the aliases list and select New Reference Alias.

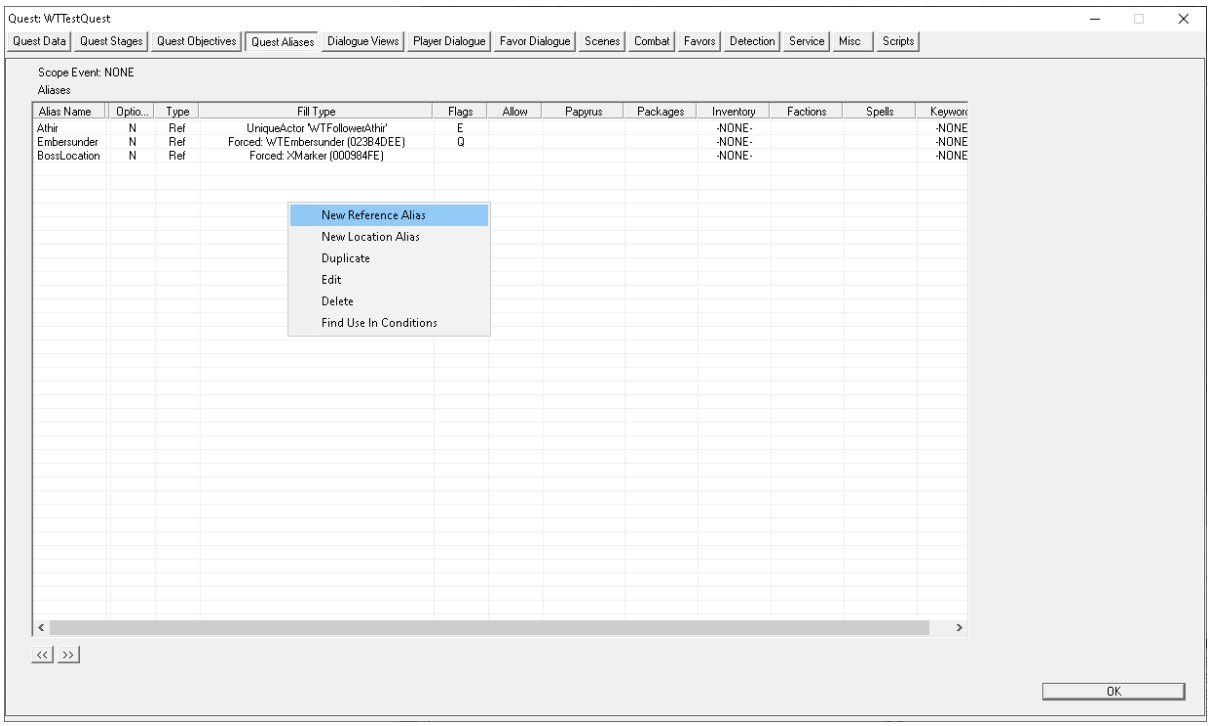


Figure 866 - Creating a new reference alias.

I'm going to add in Daenlit so she has something to say and flag her as essential.

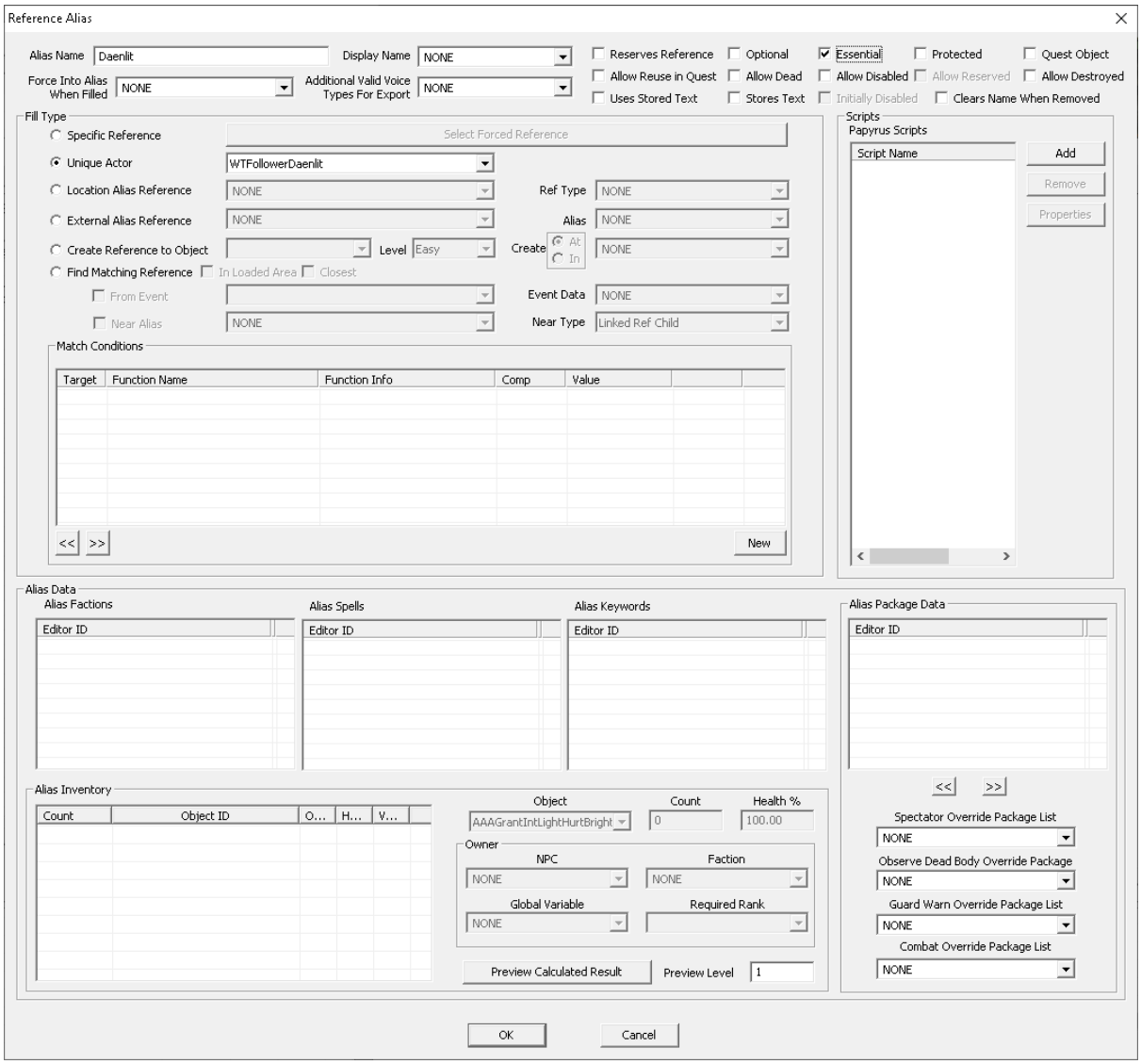


Figure 867 - Setting Daenlit as the reference in the new reference alias.

Click OK to close out of the Reference Alias properties.

Go back to our scene in the Scenes tab.

Right-click to the right of Phase 1 and select 'Add Phase at End'.

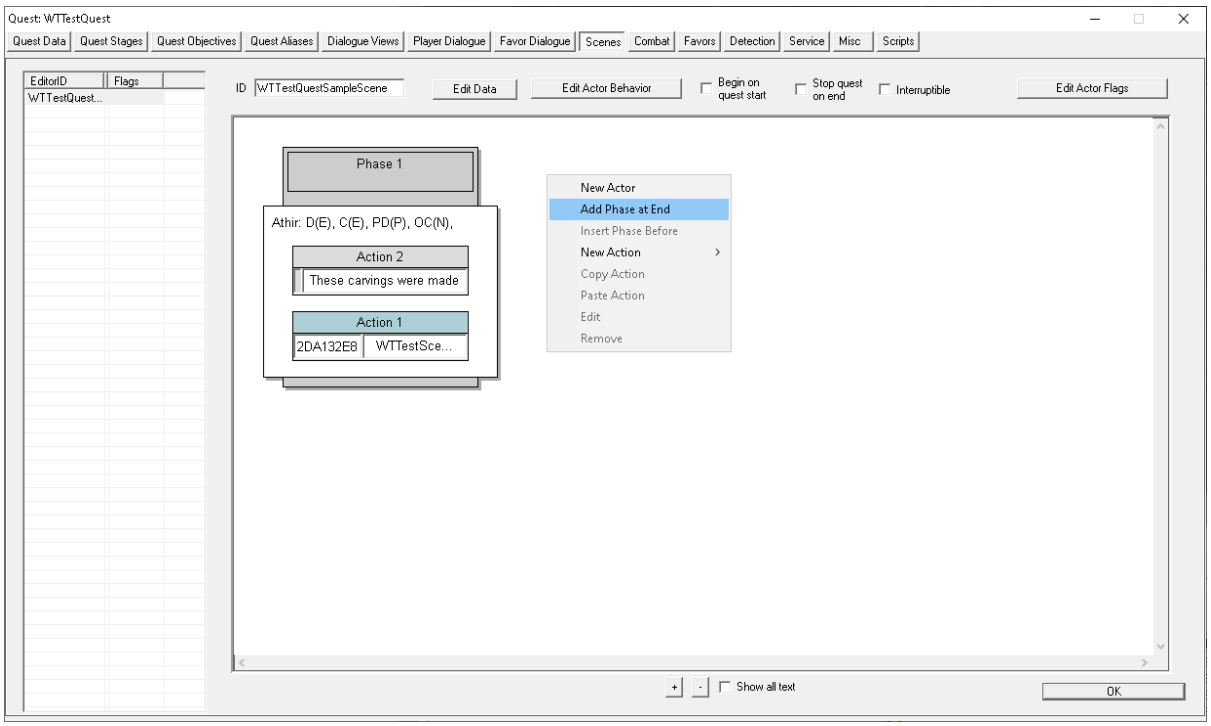


Figure 868 - Adding in a second phase.

You should now see Phase 2 after Phase 1 as per the screenshot below:

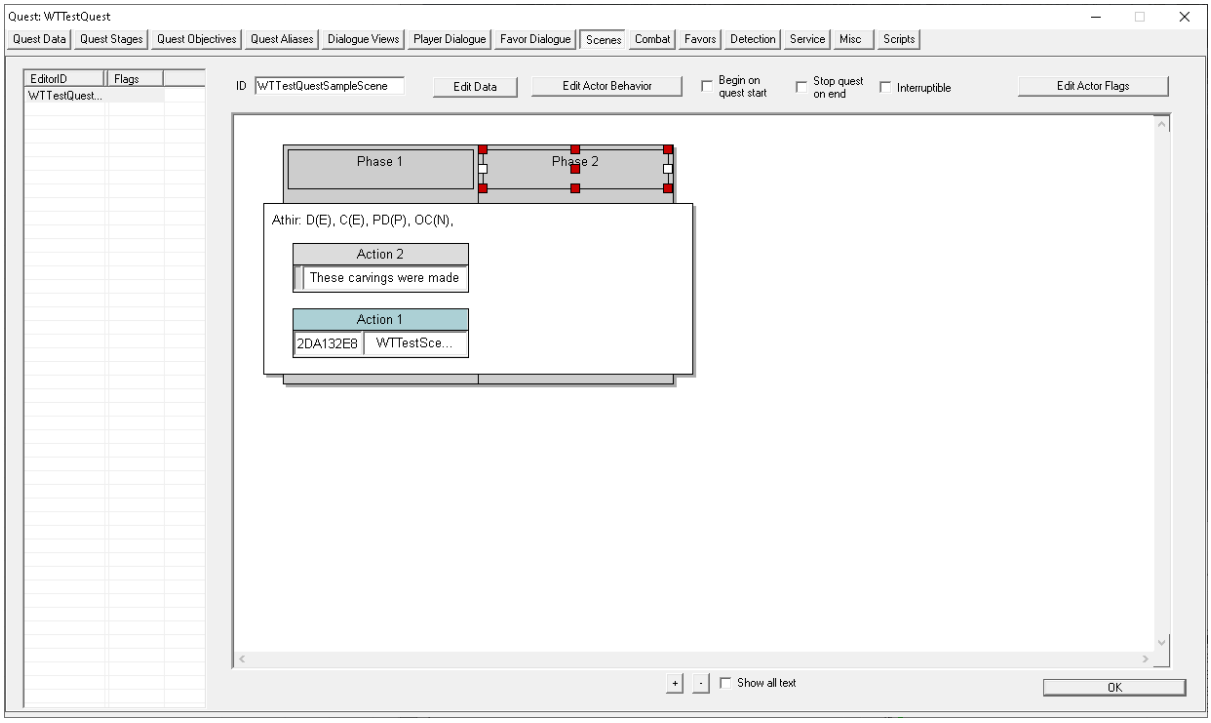


Figure 869 - Second phase added.

Right-click outside of the phases and select New Actor.

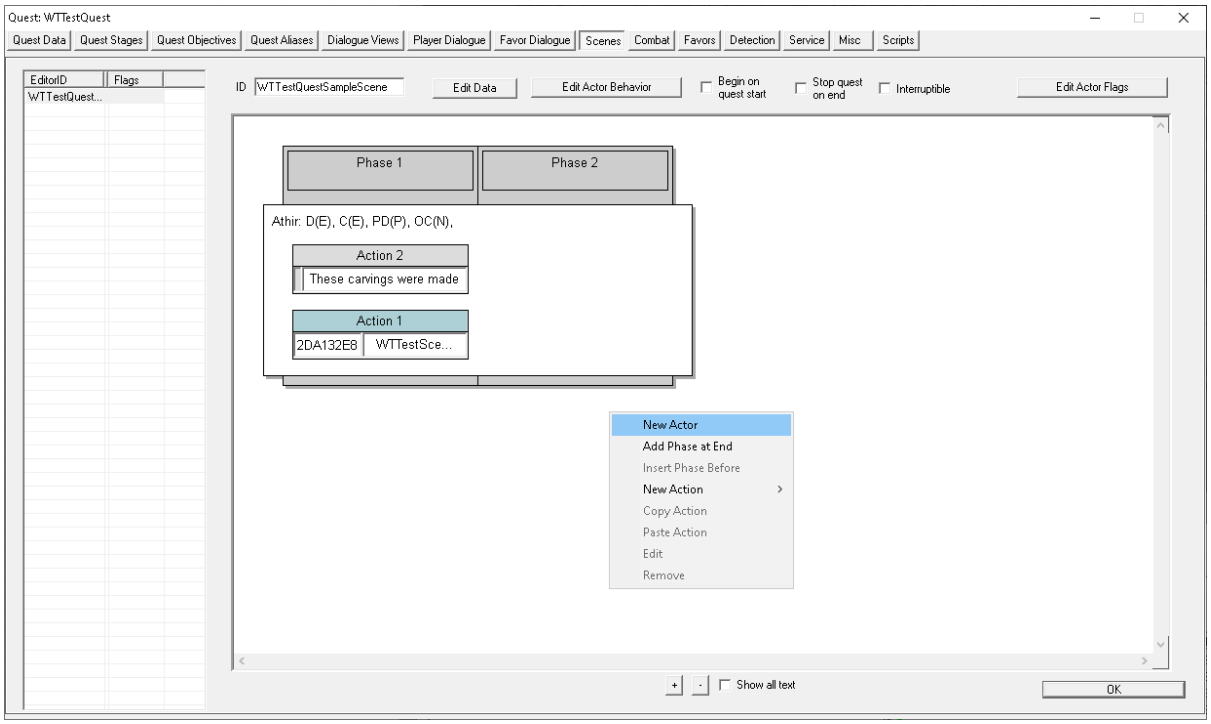


Figure 870 - Adding a second actor into our scene.

Select Daenlit in the Select Form pop-up then click OK.

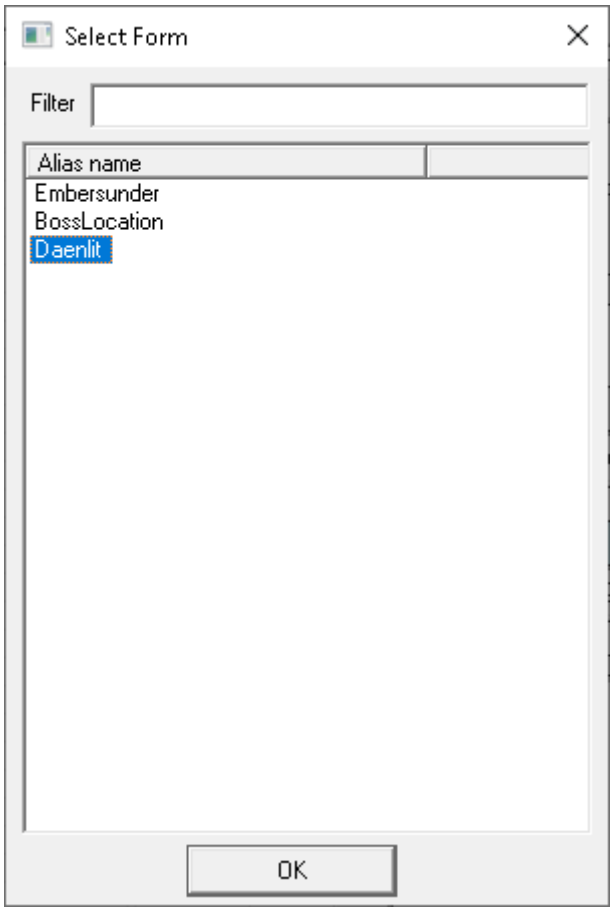


Figure 871 - Selecting Daenlit's alias.

We should now see Daenlit listed below Athir.

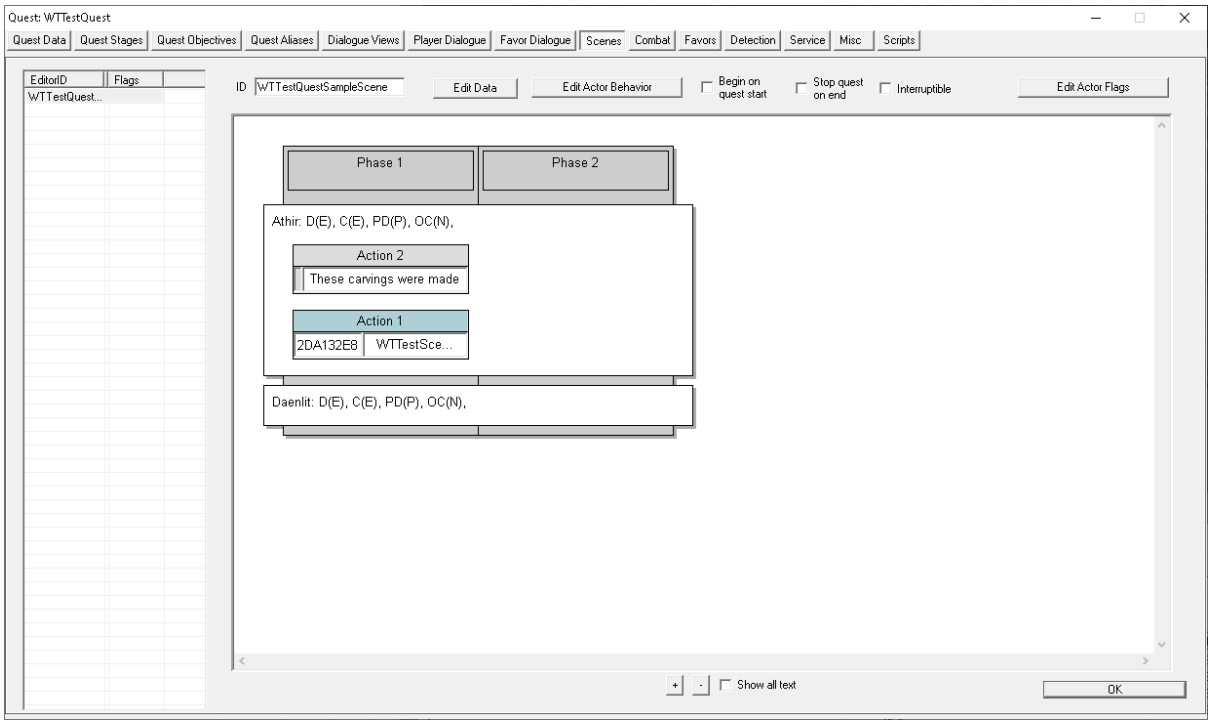


Figure 872 - Scene with two NPCs.

Now let's give her something to say. Right-click under Phase 2 beside Daenlit and select New Action > Dialogue.

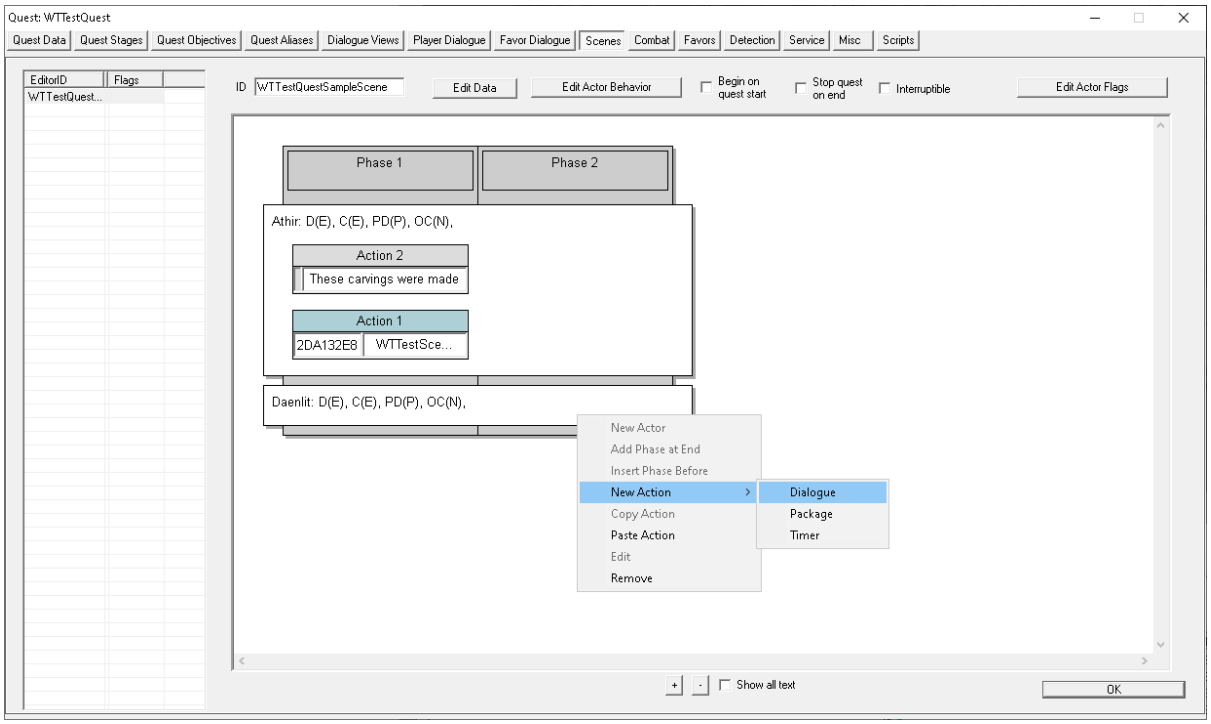


Figure 873 - Assigning scene dialogue to Daenlit.

Right-click in the dialogue list and select New.

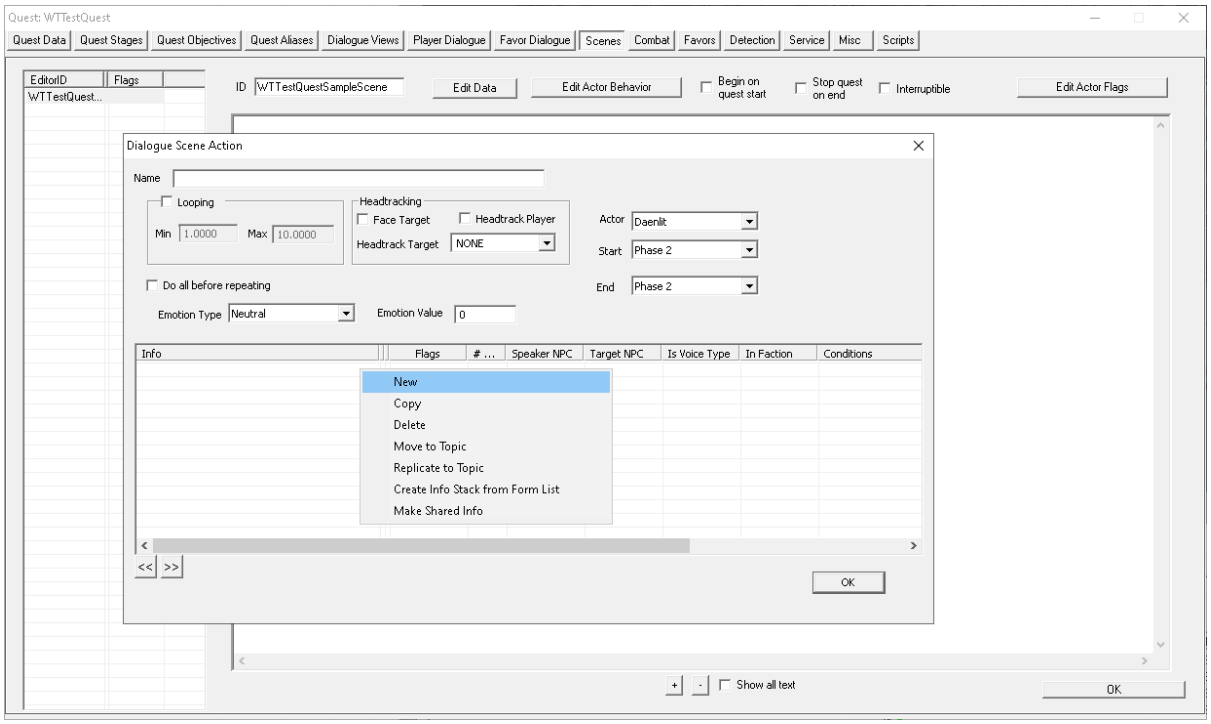


Figure 874 - Adding dialogue to the Dialogue Scene Action.

Enter in her dialogue into the Response Text field. I set her Emotion Type to Fear and the Emotion Value to 20 to match her dialogue.

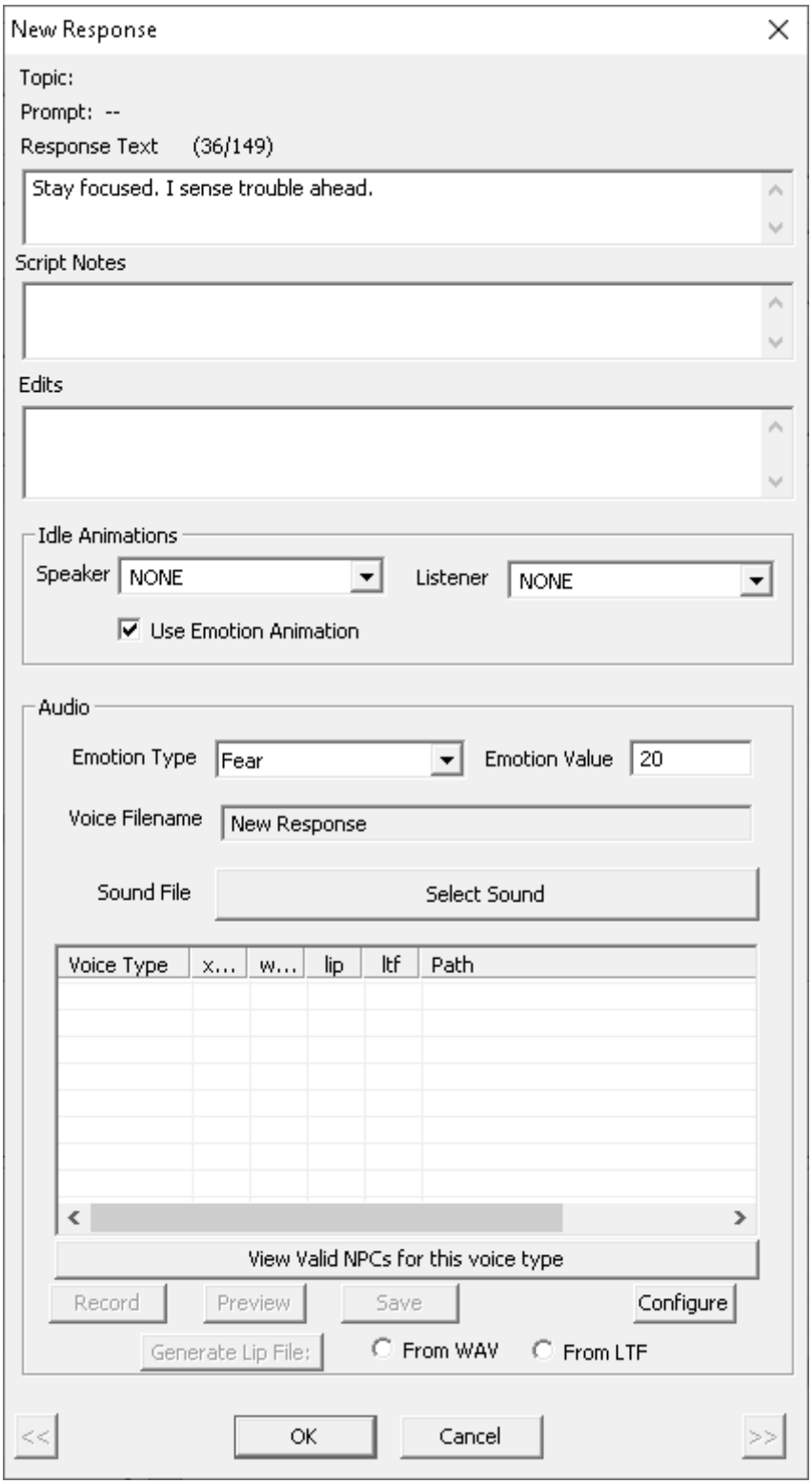


Figure 875 - Adding response text.

Click OK to close out of New Response.

Right-click in the conditions list and select New.

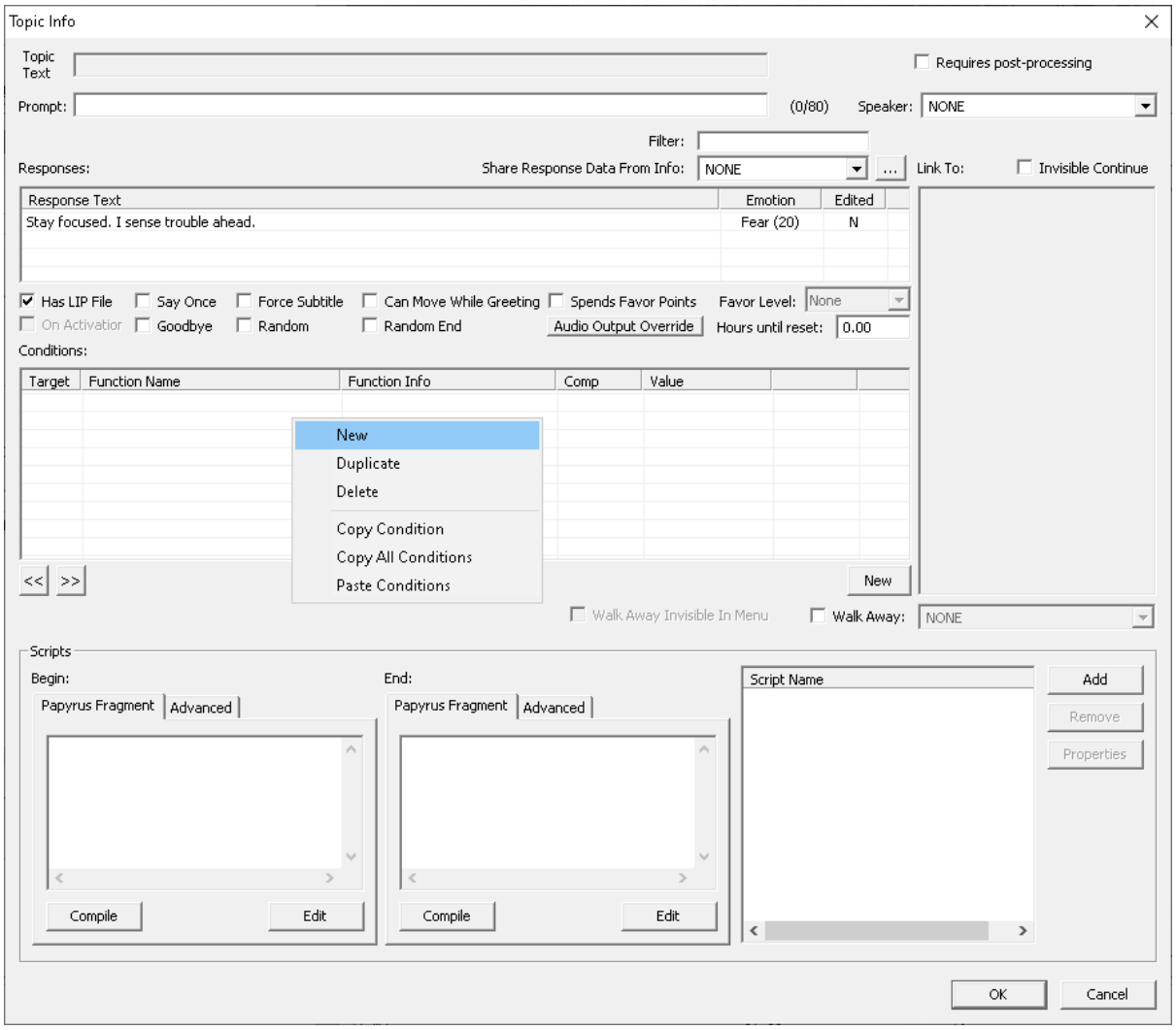


Figure 876 - Adding a condition to Daenlit's response.

Again, I set a GetIsID condition to ensure Daenlit is the only actor able to deliver this line.

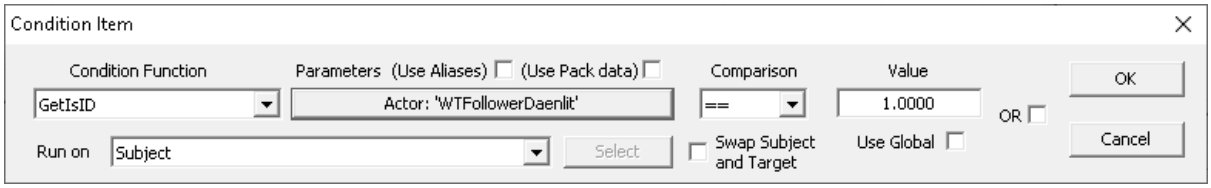


Figure 877 - GetIsID condition.

Click OK to close out of Condition Item properties.

Click OK to close out of Topic Info.

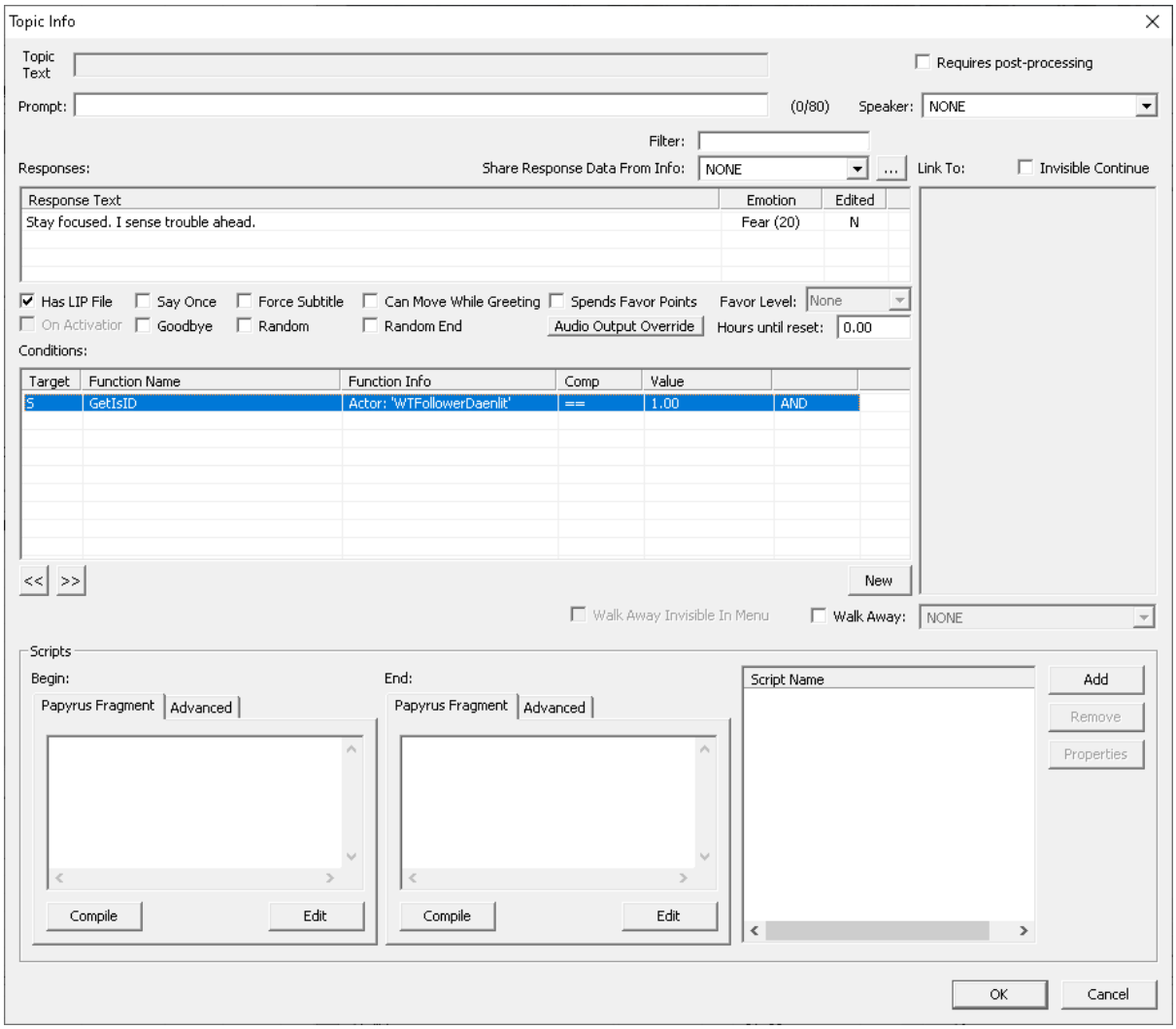


Figure 878 - Topic Info with condition.

Set Headtrack Target to Athir, then click OK to close out of Dialogue Scene Action.

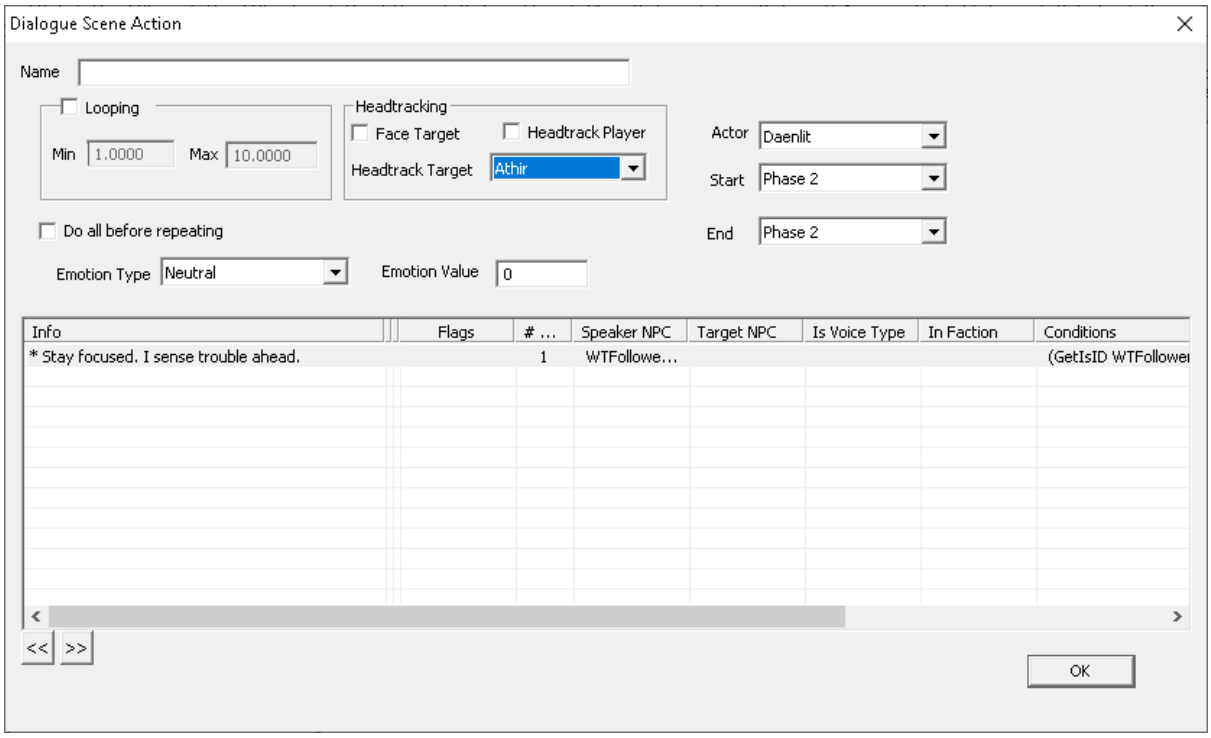


Figure 879 - Dialogue Scene Action with response text.

We should now see our dialogue action in Phase 2. After Athir delivers his line and walks over to the glyphs on the wall, Daenlit should deliver her line. We're not overriding her package so she should continue to follow the player while speaking.

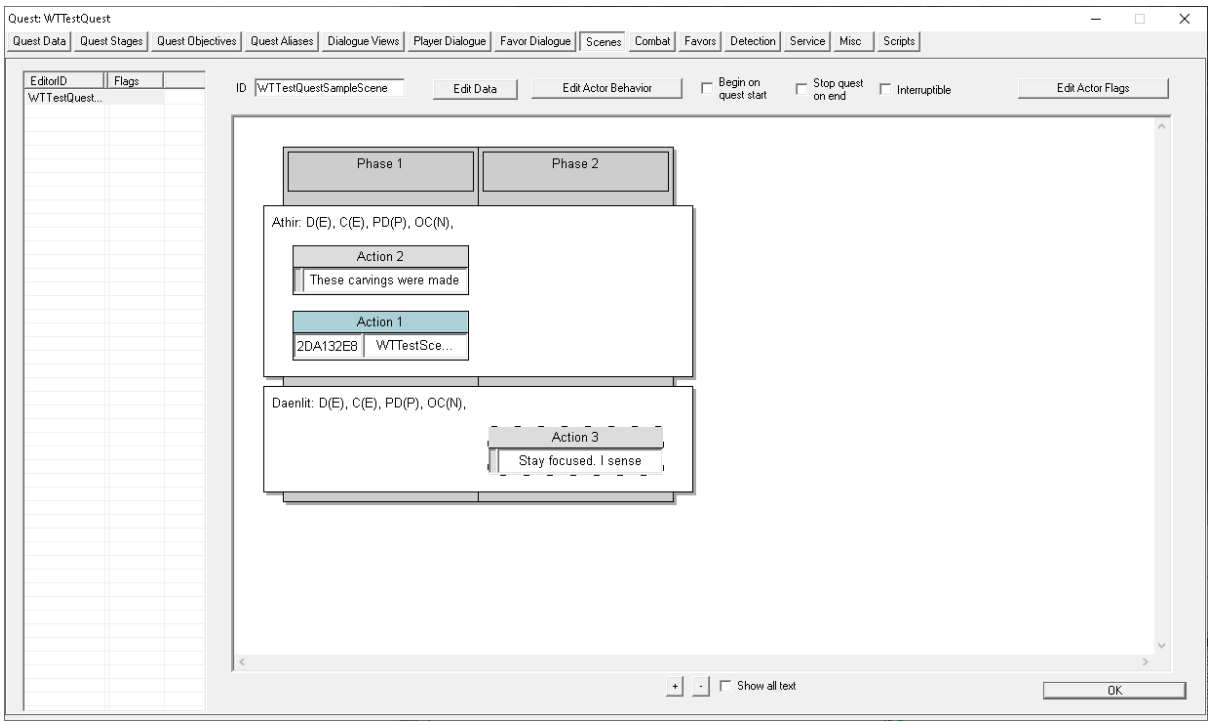


Figure 880 - Phase 2 with a dialogue action.

Let's add in a quick delay before the next action.

Right-click in the empty space after Phase 2 and select 'Add Phase at End'.

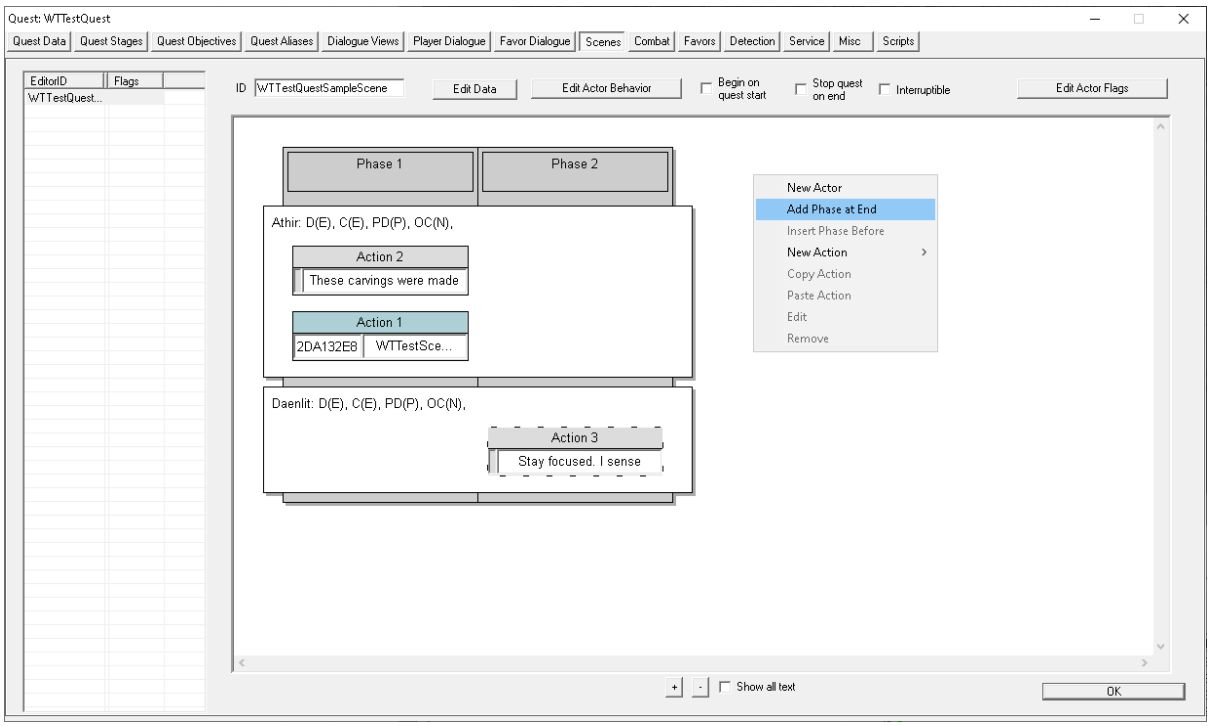


Figure 881 - Adding a third phase.

We should now have three phases as per the following screenshot:

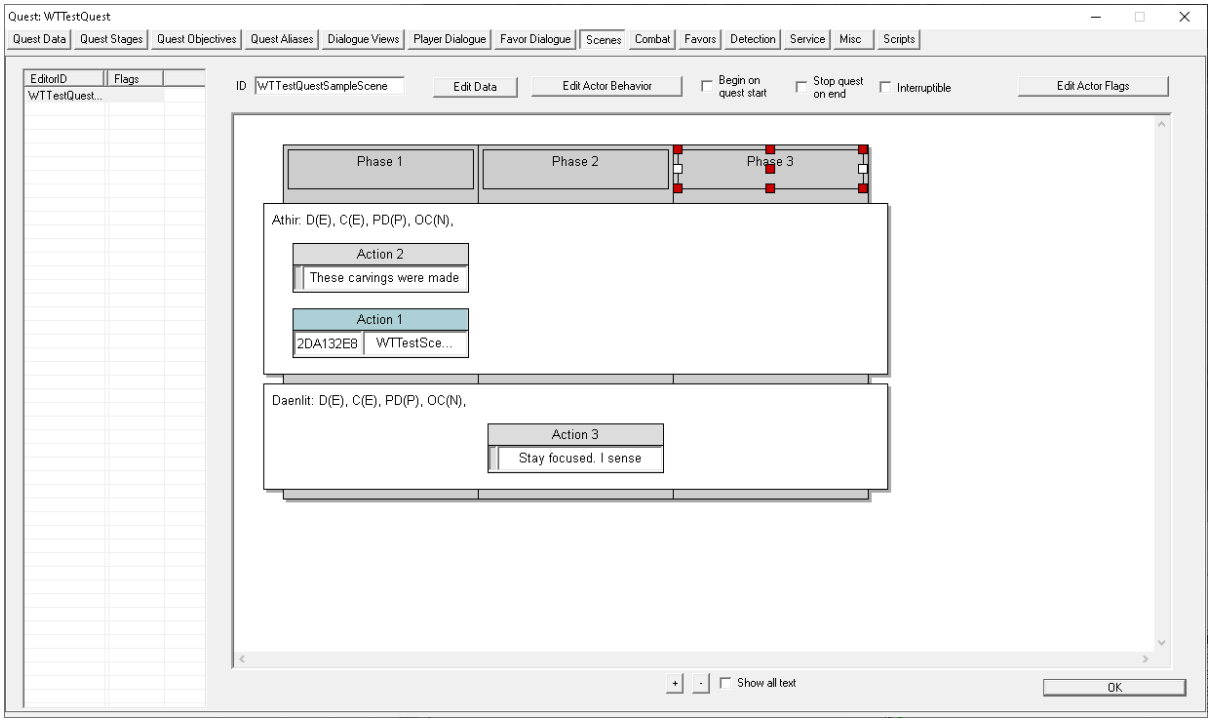


Figure 882 - Our scene with three phases.

Right-click under Phase 3 beside Daenlit and select New Action > Timer.

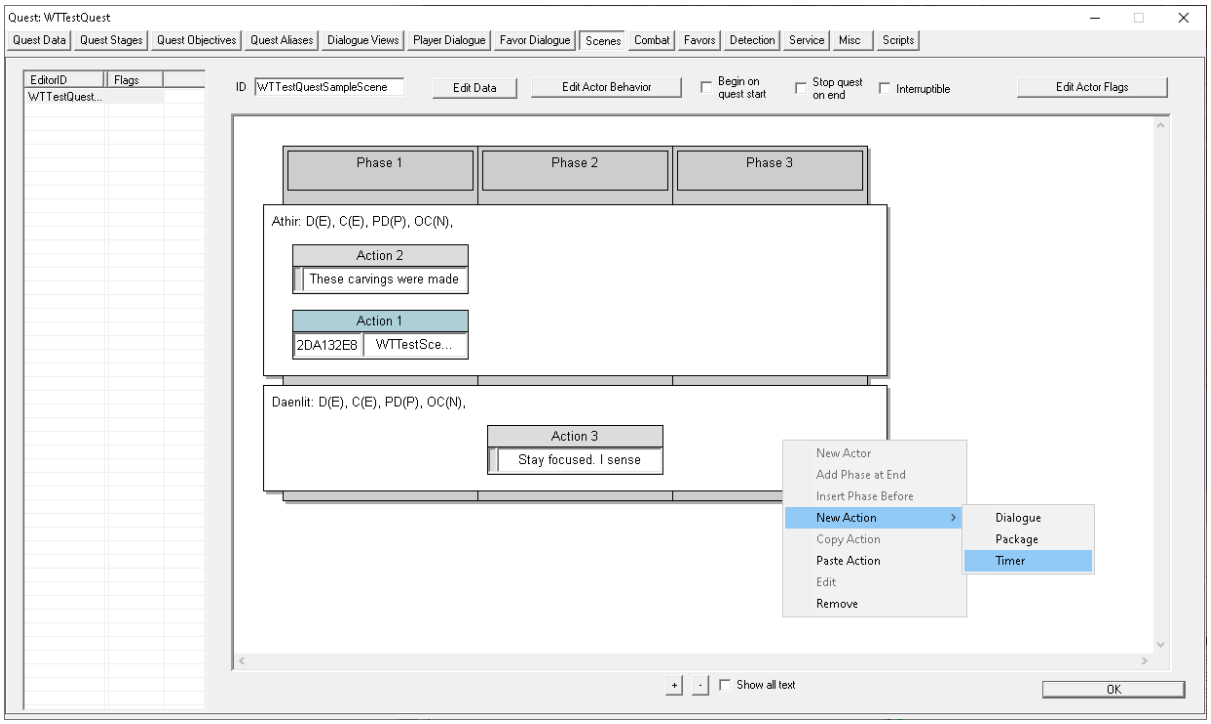


Figure 883 - Adding in a timer.

Set the Seconds field to 3.0000 then click OK.

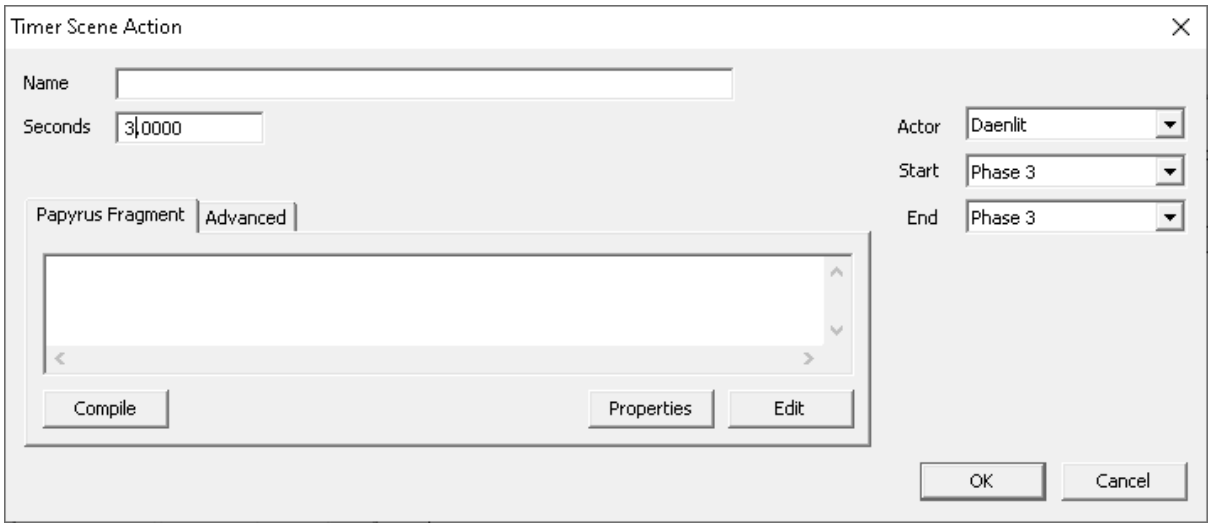


Figure 884 - Adding a three second delay before the next action.

A three second delay should now appear highlighted in yellow under Phase 3.

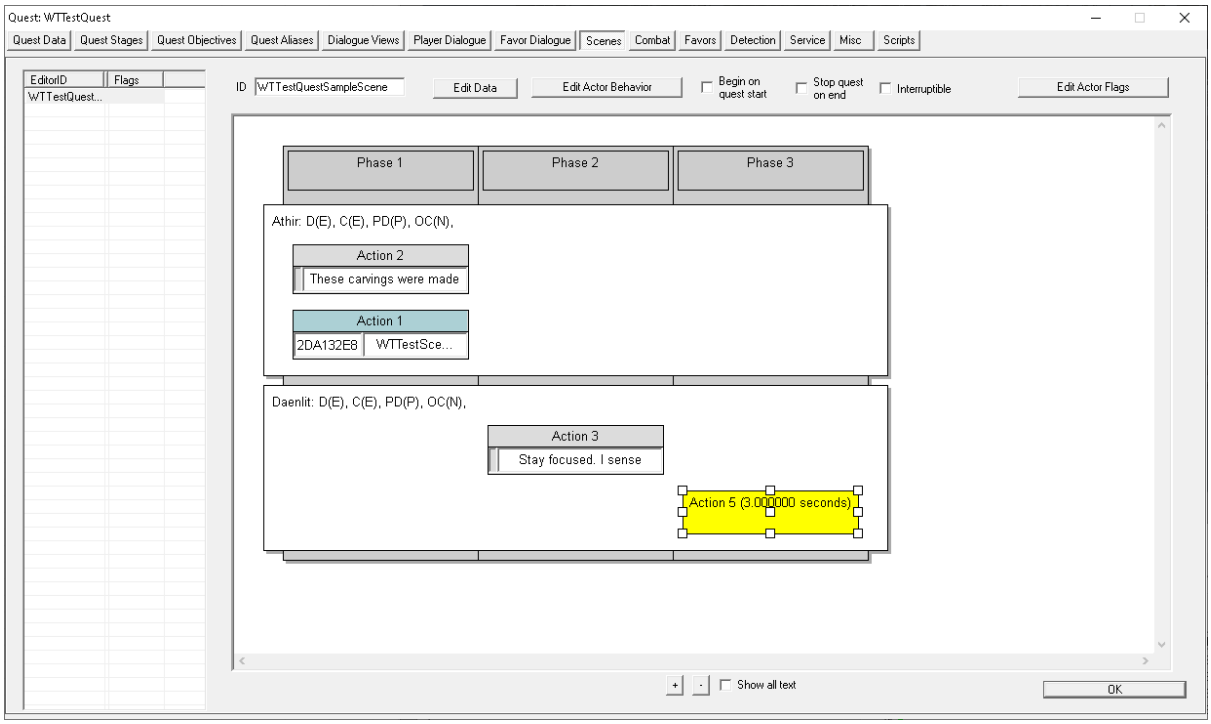


Figure 885 - A delay added to Phase 3.

Now let's add in a third NPC.

Go back to the Quest Aliases tab, right-click in the aliases list and select New Reference Alias.

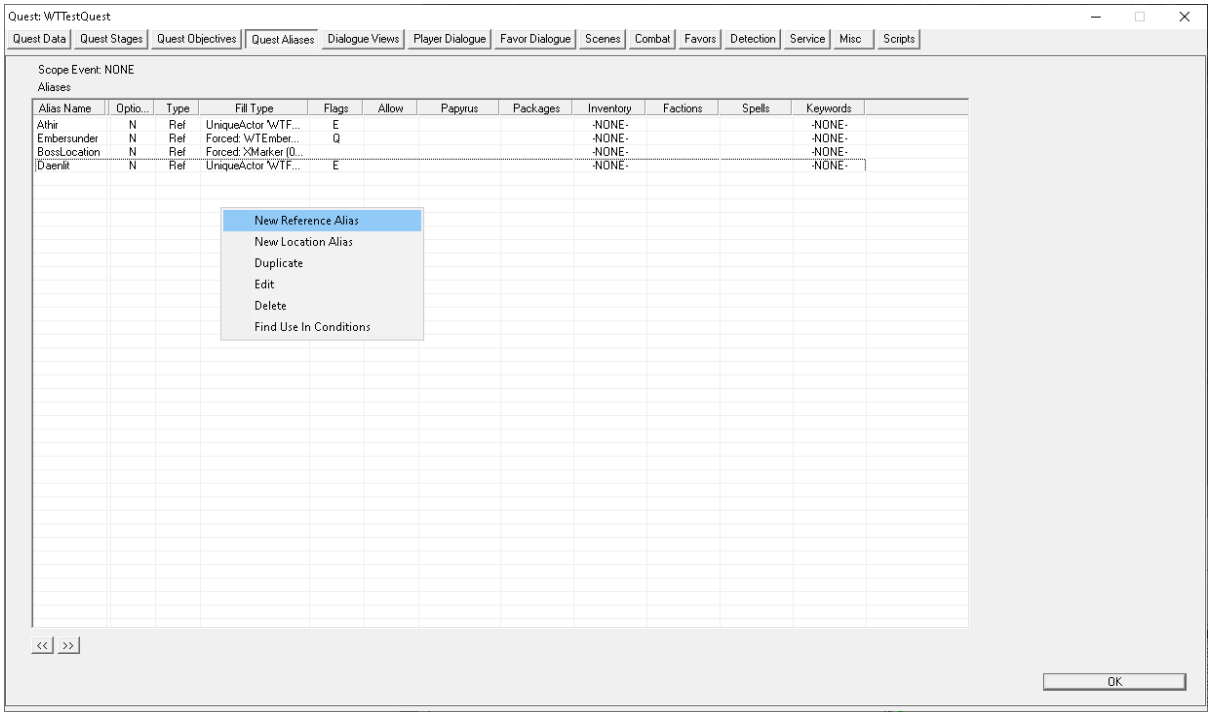


Figure 886 - Adding in a third NPC alias.

For the third NPC alias, I select Shargam and flagged him as essential.

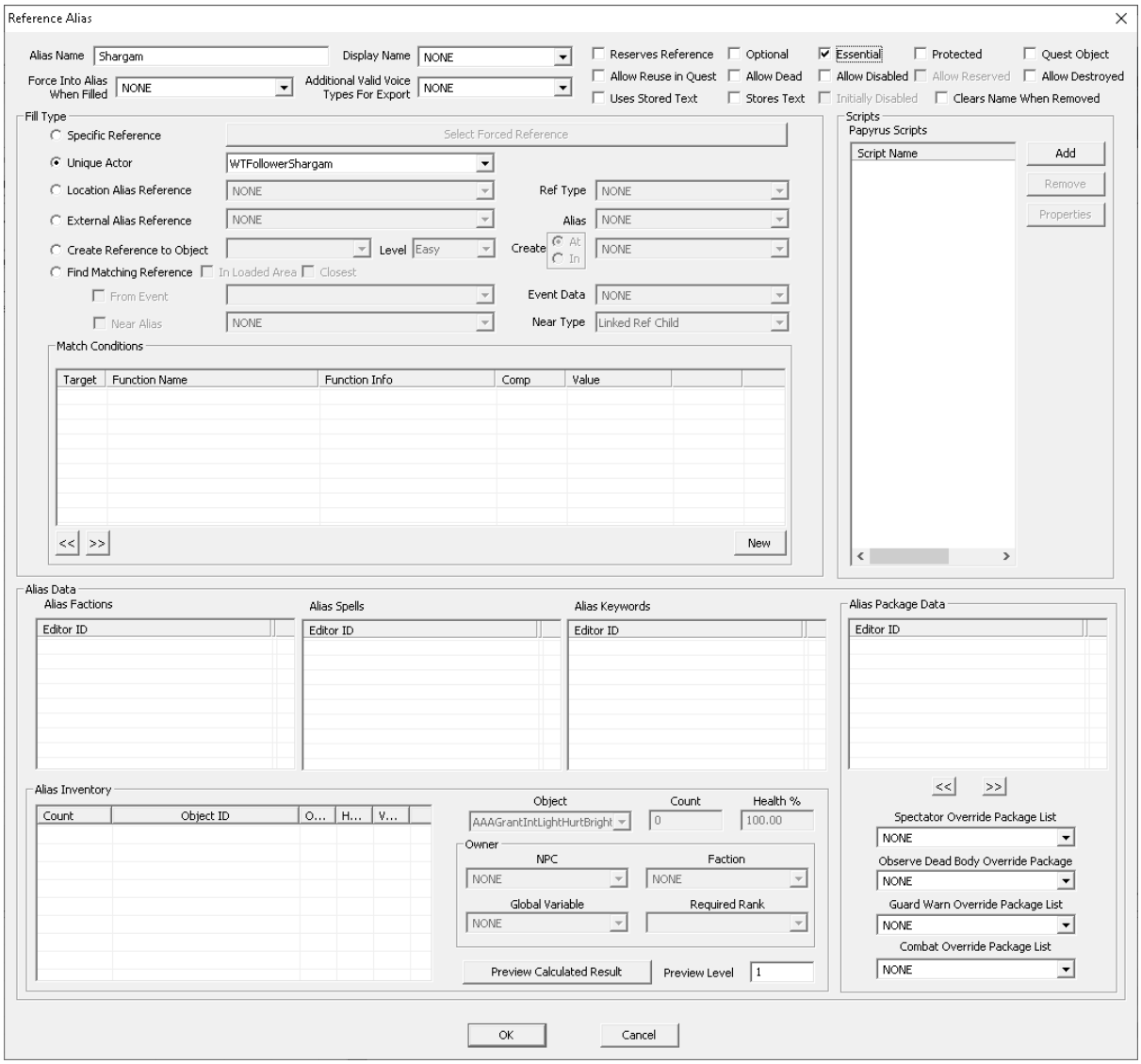


Figure 887 - Setting up our third NPC reference alias.

Click OK to close out of Reference Alias properties.

Go back to our scene in the Scenes tab.

Right-click in the space after Phase 3 and select 'Add Phase at End'.

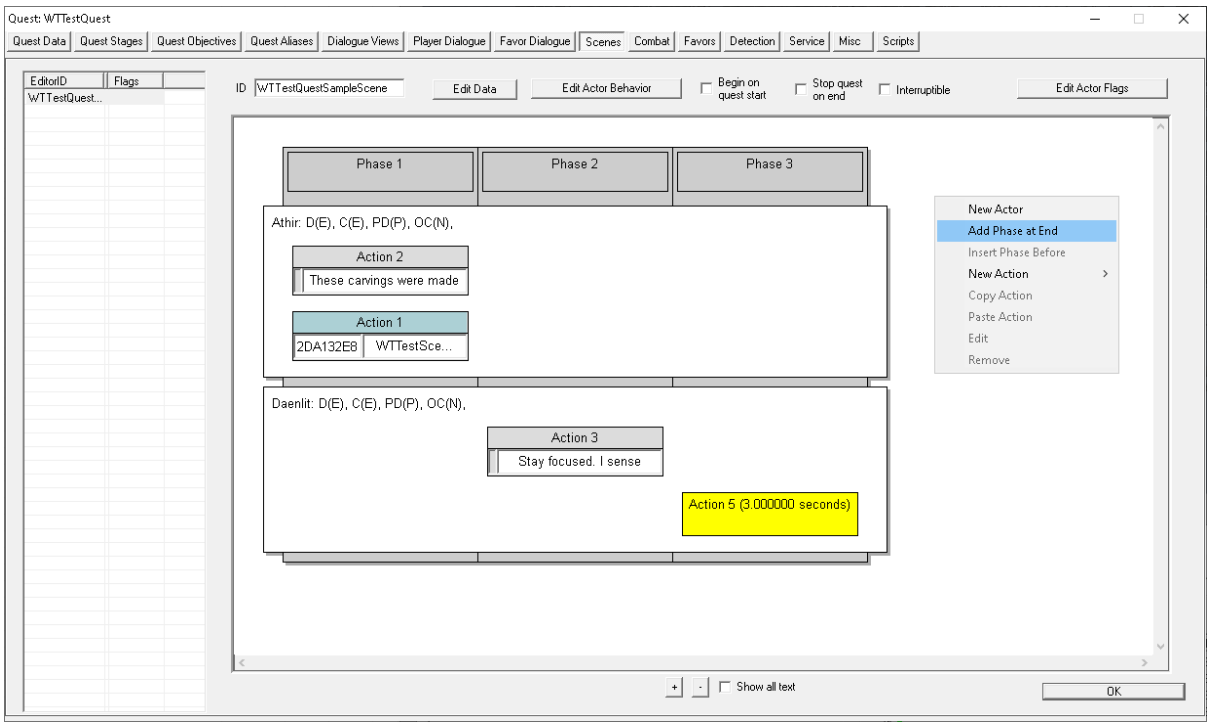


Figure 888 - Adding a fourth phase.

We should now see four phases in our scene as per the screenshot below:

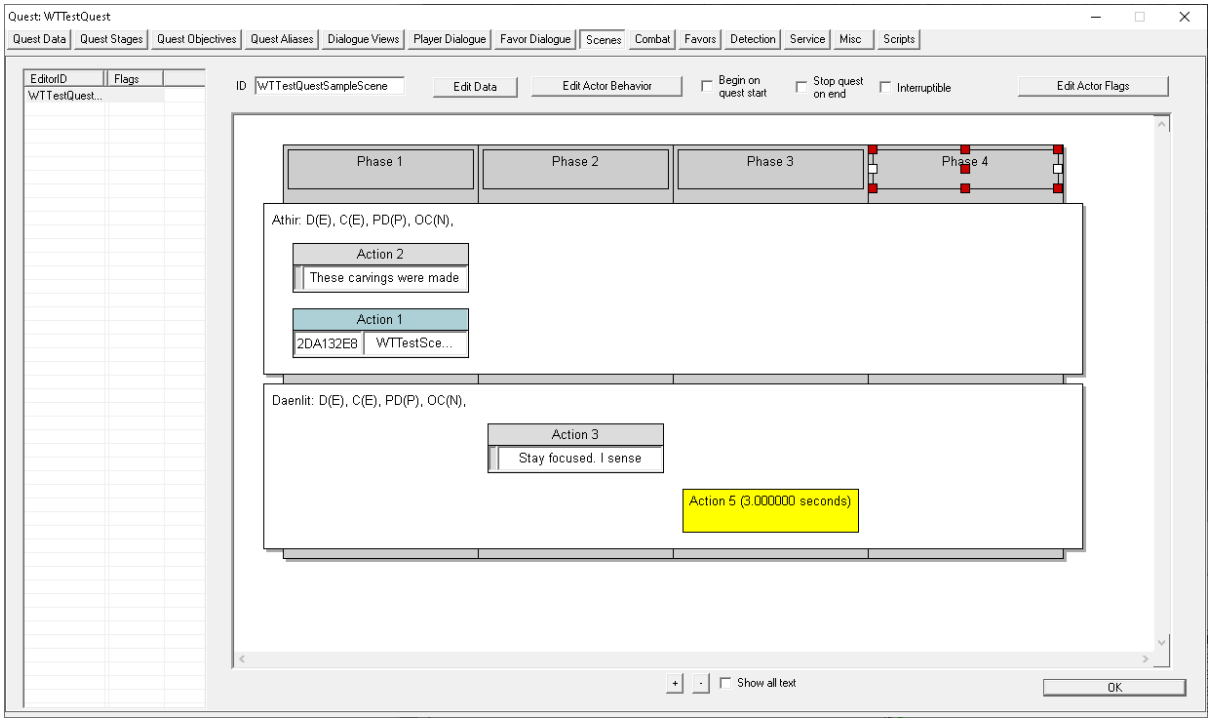


Figure 889 - Our scene with four phases.

Right-click in the area outside of the phases and select New Actor.

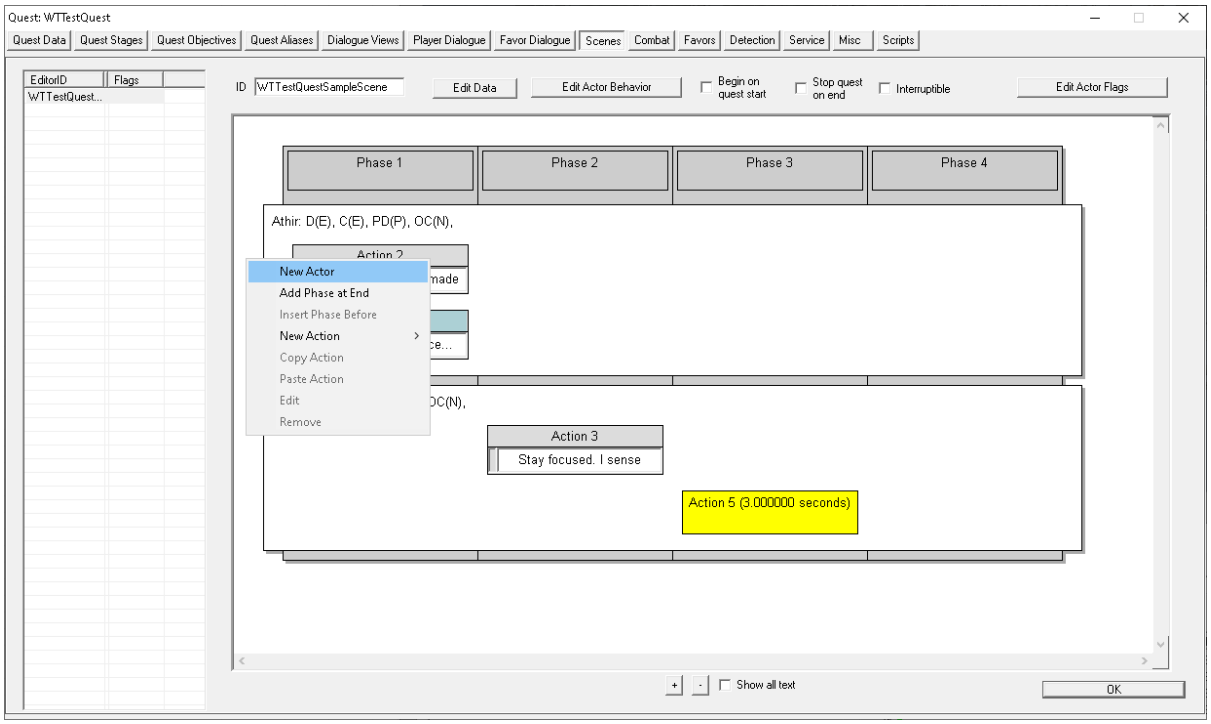


Figure 890 - Adding in our third NPC.

Select Shargam in the alias list then click OK.

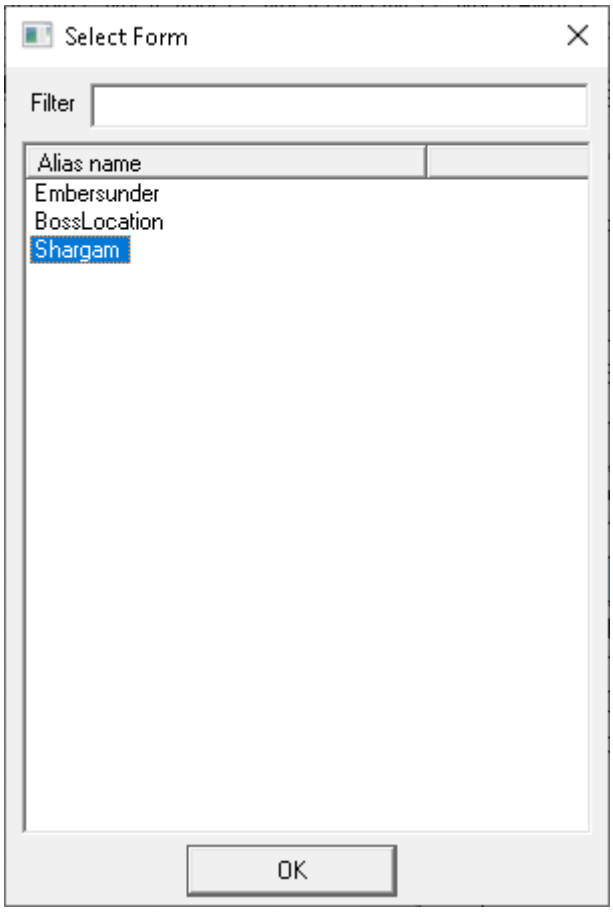


Figure 891 - Selecting Shargam's alias.

Let's set up a travel package for Shargam first. Right-click under Phase 4 next to Shargam and select New Action > Package.

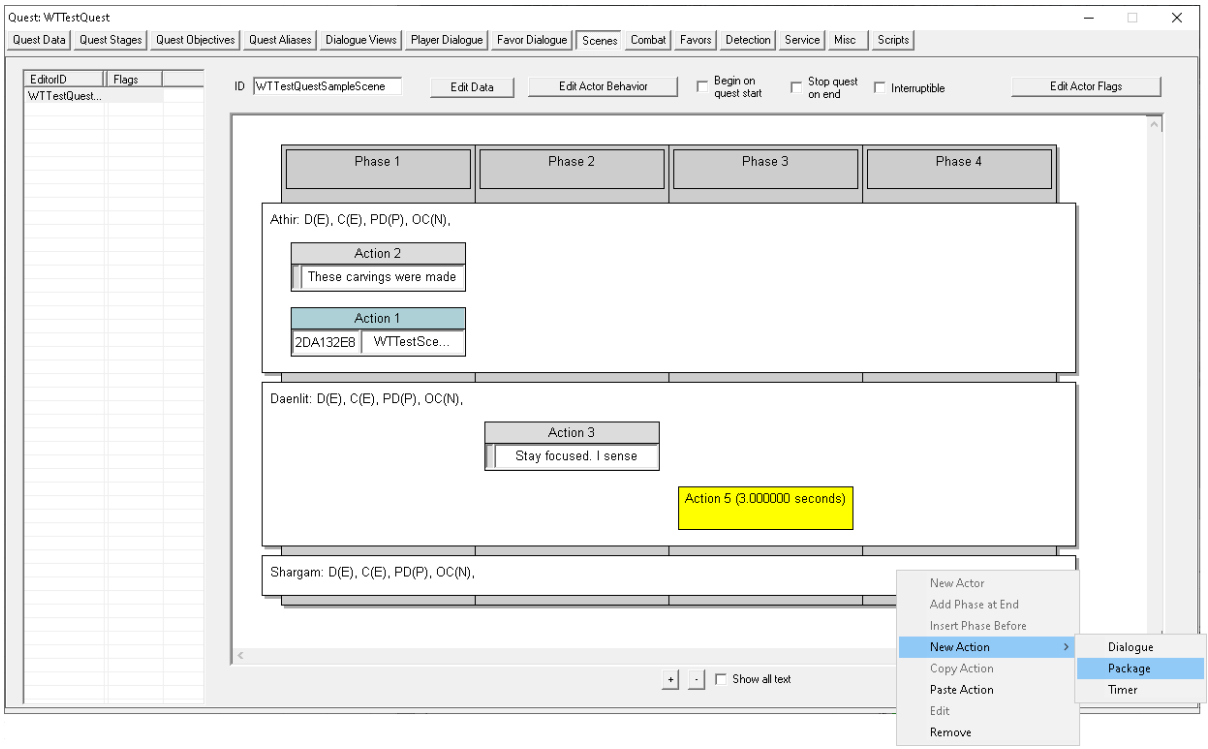


Figure 892 - Creating a new scene package for Shargam.

Right-click in the packages list and select New.

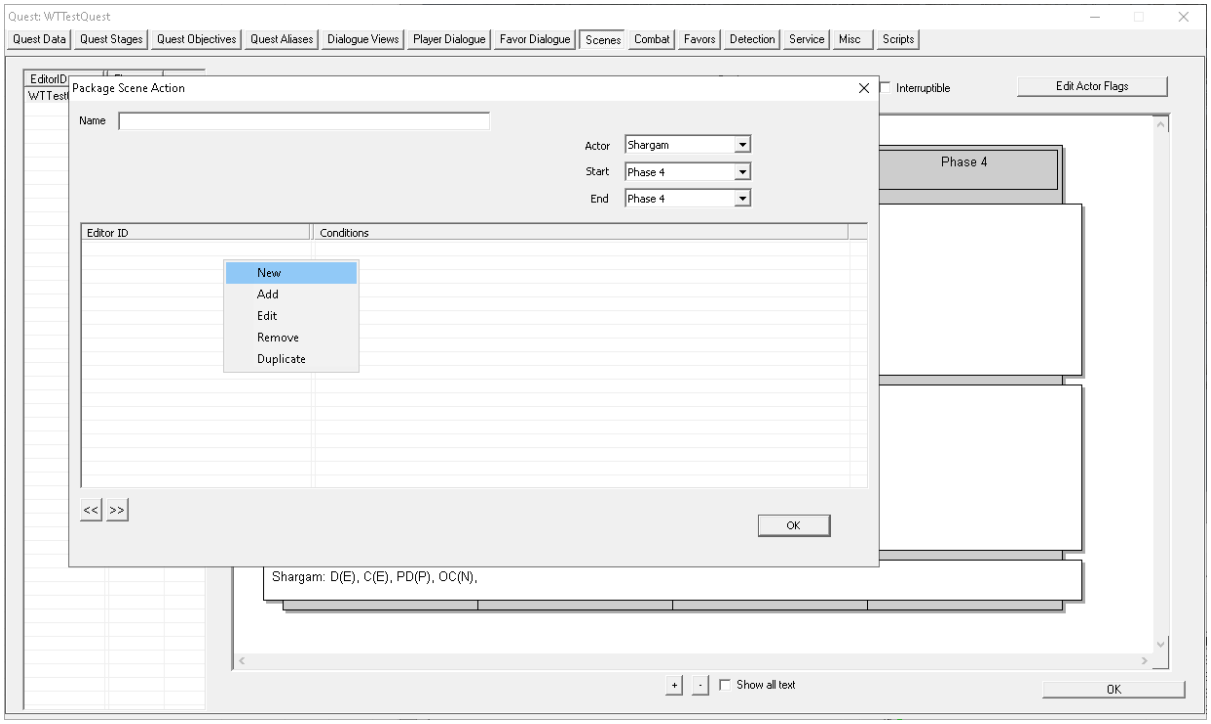


Figure 893 - Adding a new package to Package Scene Action.

Enter in the package ID. I just called it WTTestSceneShargamWalkToPackage for this example.

Make sure the 'Owner quest' drop-down is set to the name of the quest you're setting up the scene in.

In this example, I set up a new package using the Travel template and made Shargam travel to the XMarkerHeading facing the puzzle door.

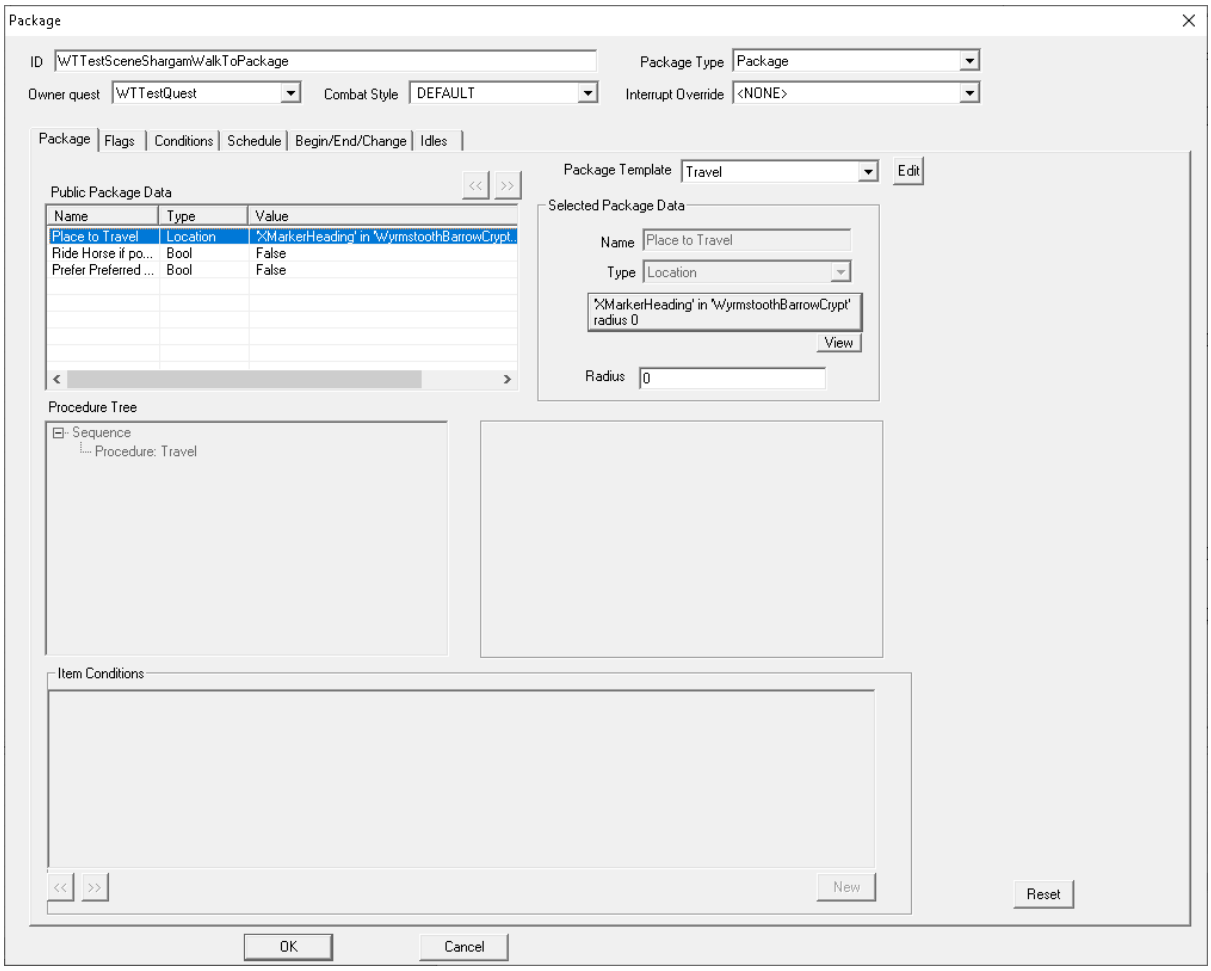


Figure 894 - Shargam's package for phase 4.

Go to the Flags tab.

Tick Preferred Speed and set the drop-down beside it to Run.

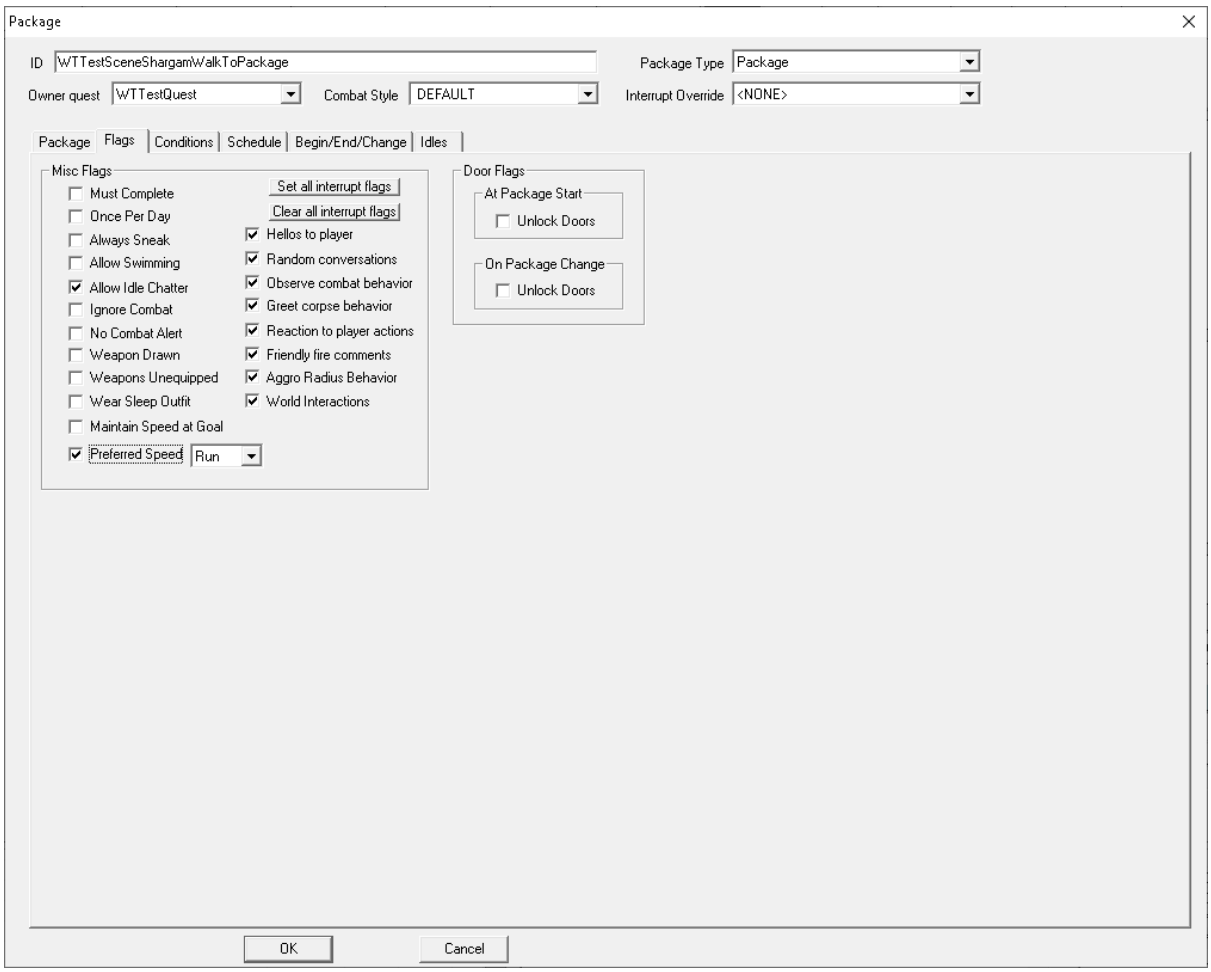


Figure 895 - Setting Shargam's movement speed for this package.

Click OK to add the new package to Shargam for phase 4 of the scene.

Click OK to close out of Package Scene Action.

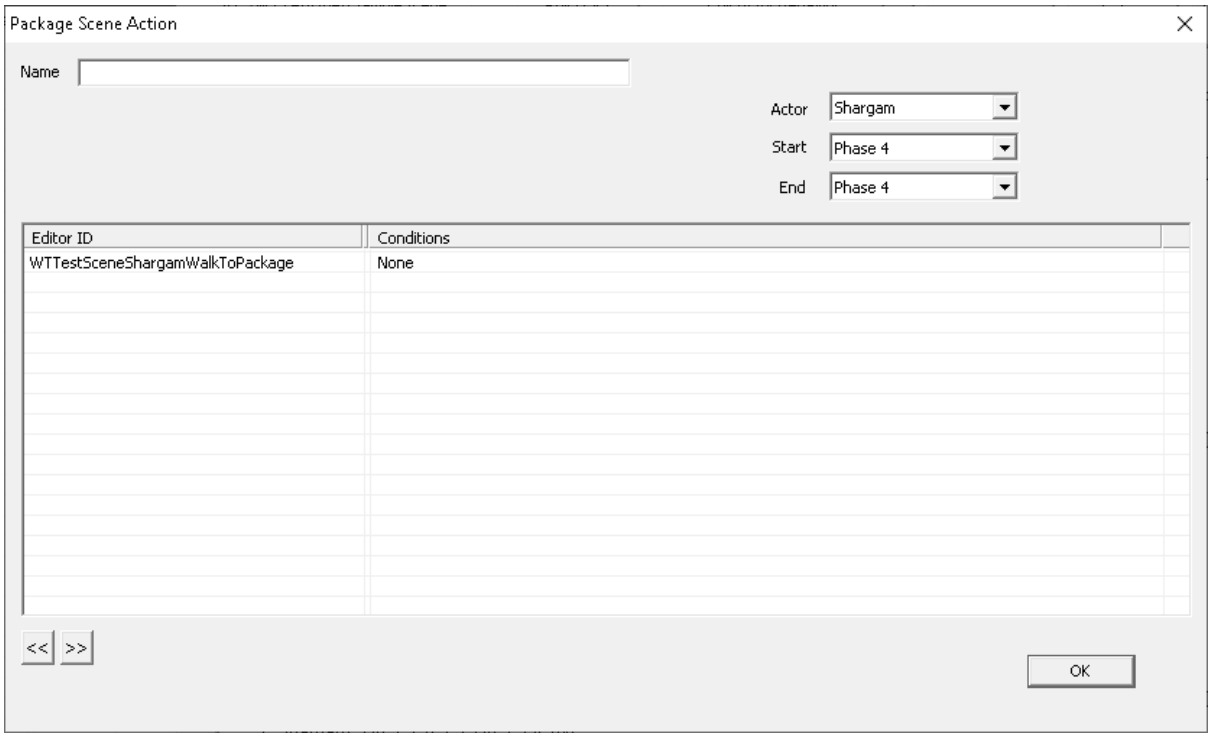


Figure 896 - Package Scene Action with a package set up for phase 4.

We should now see Shargam's phase 4 package.

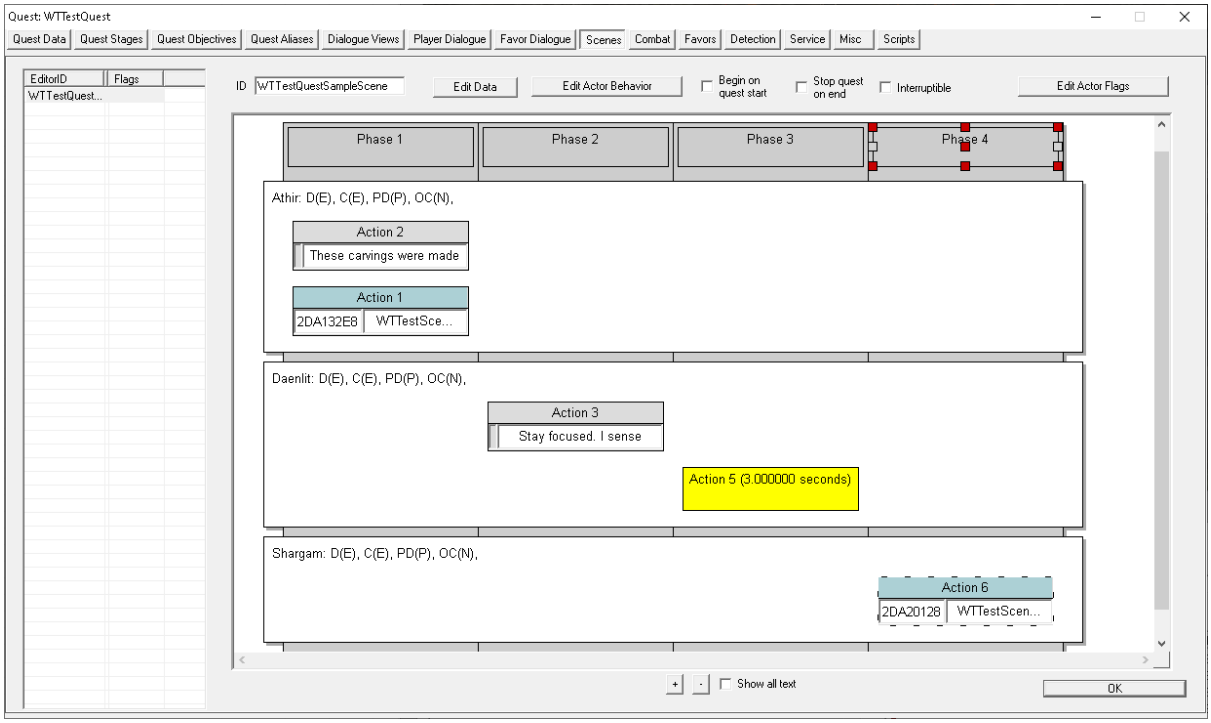


Figure 897 - Phase 4 with a package set up for Shargam.

Lastly, let's give him something to say.

Right-click under Phase 4 next to Shargam and select New Action > Dialogue.

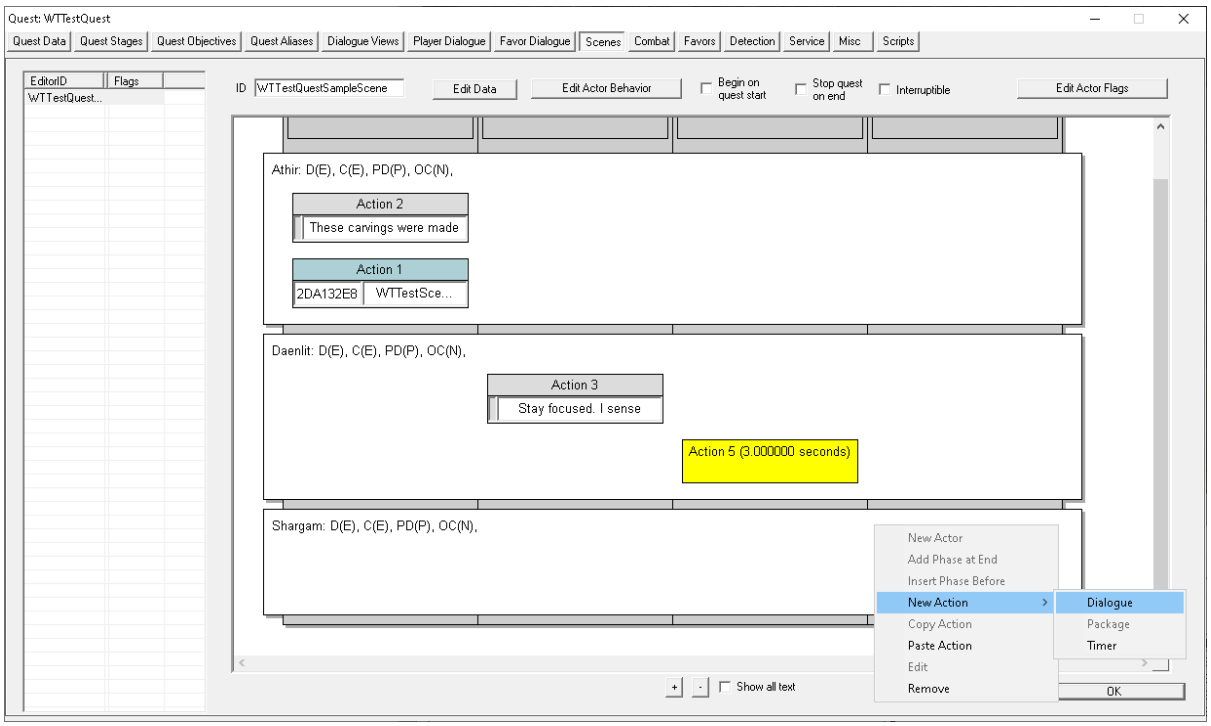


Figure 898 - Adding a dialogue action for Shargam in Phase 4.

Right-click in the responses list and select New.

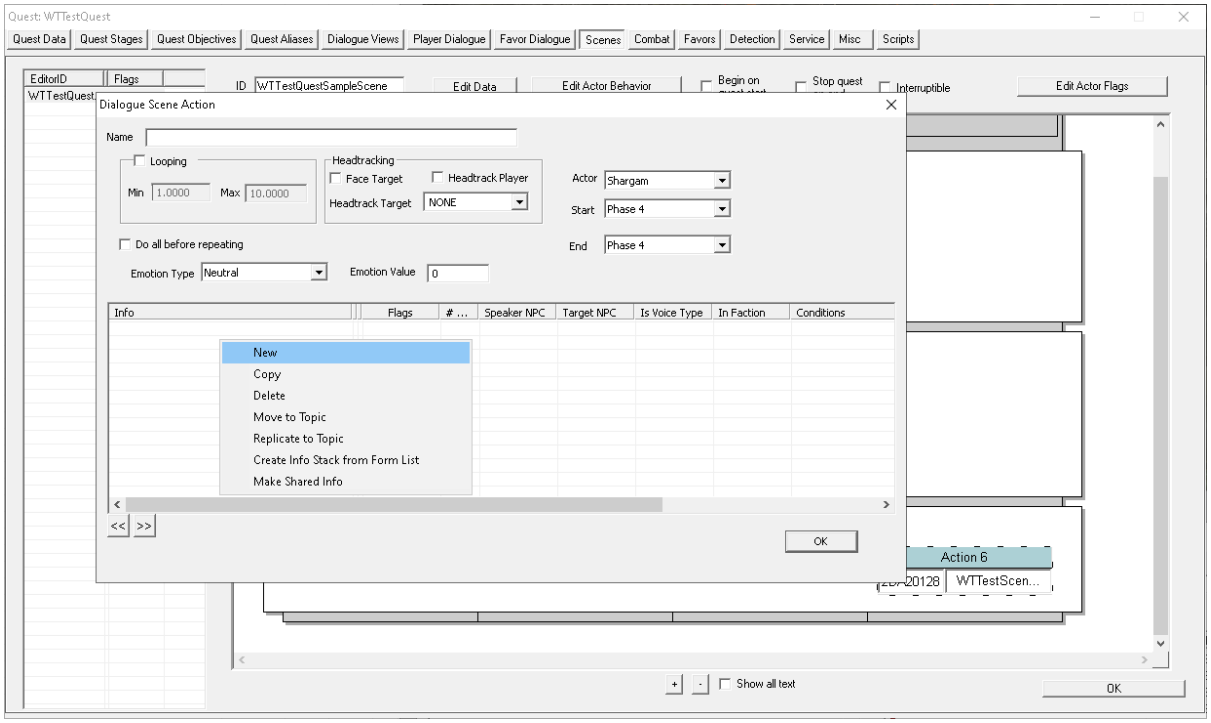


Figure 899 - Adding a line of dialogue to the Dialogue Scene Action.

Enter Shargam's line in the Response Text field. I also set his Emotion Type to Anger and the Emotion Value to 40.

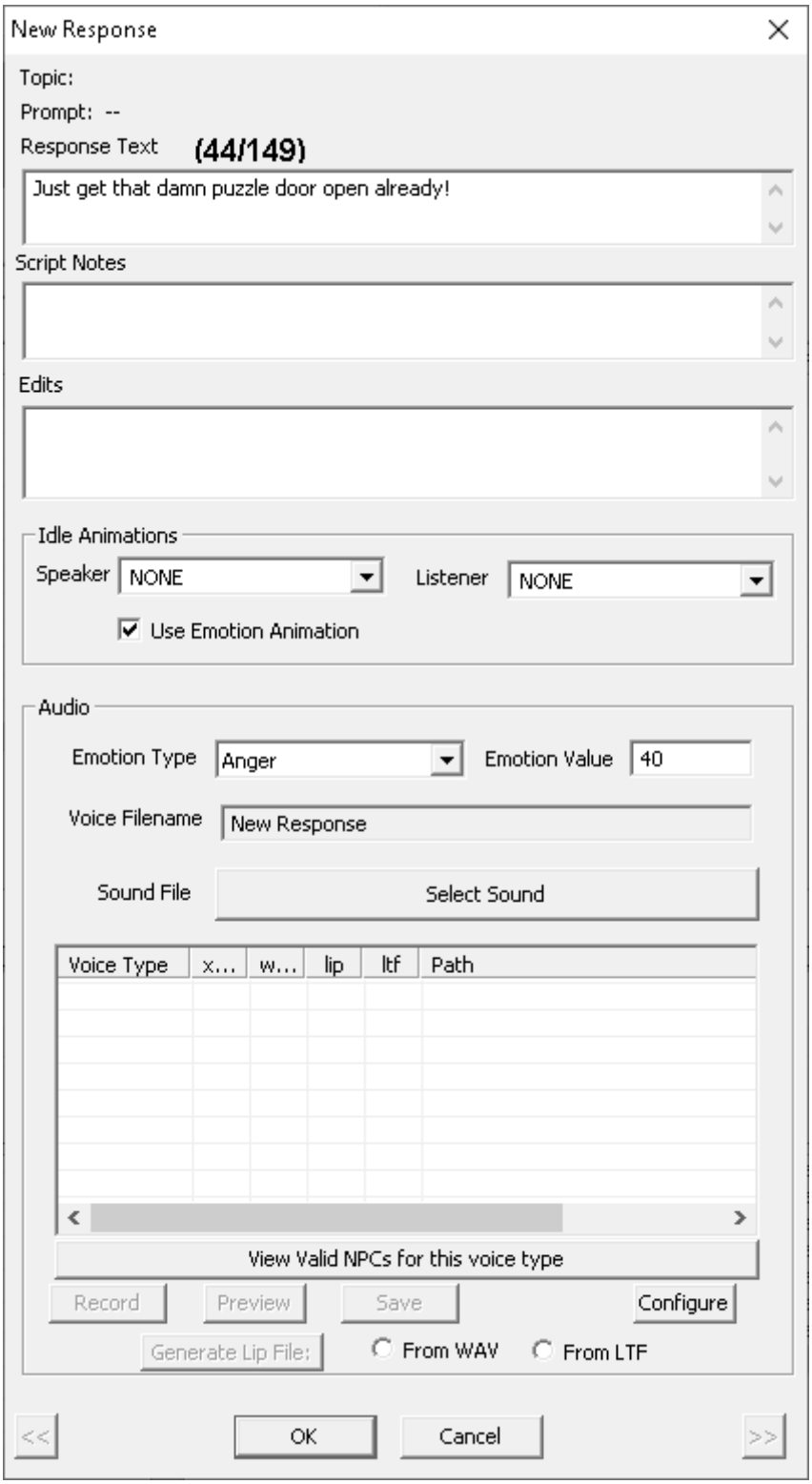


Figure 900 - Shargam's response for phase 4.

Click OK to close out of New Response.

Right-click in the conditions list and select New.

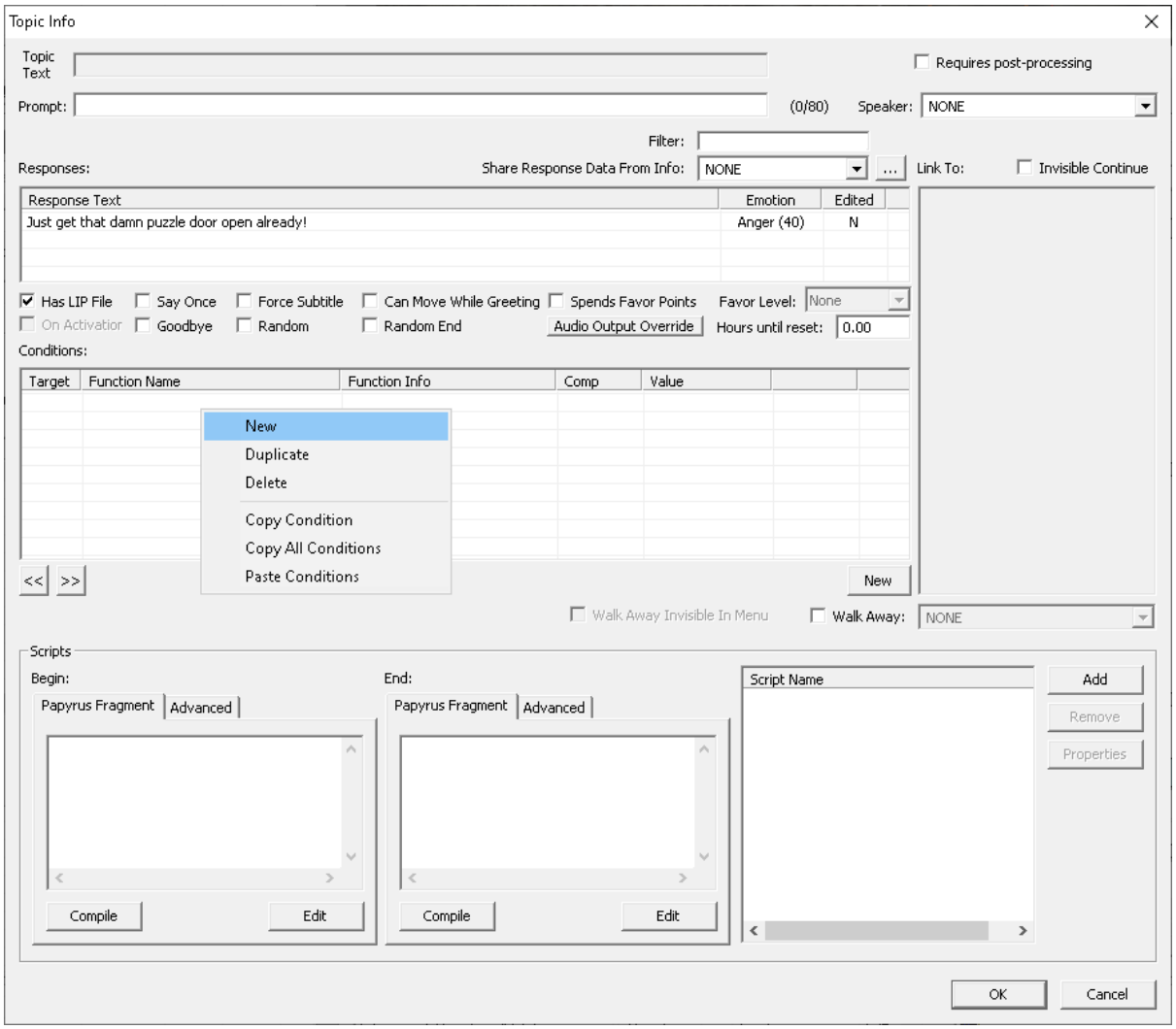


Figure 901 - Adding a new condition to Shargam's line of dialogue.

Again I set the Condition Function to GetIsID and set the actor to Shargam to ensure only Shargam can say this line of dialogue.

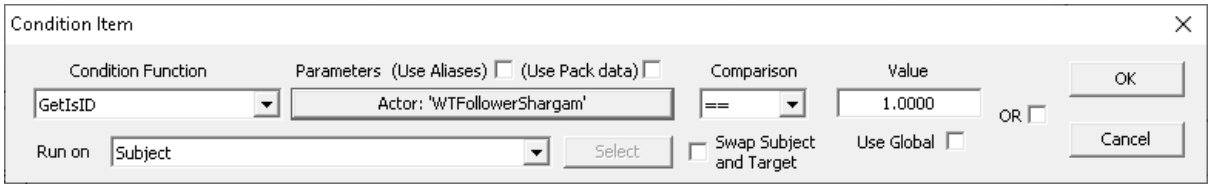


Figure 902 - GetIsID condition set to Shargam.

Click OK to close out of Condition Item.

Click OK to close out of Topic Info.

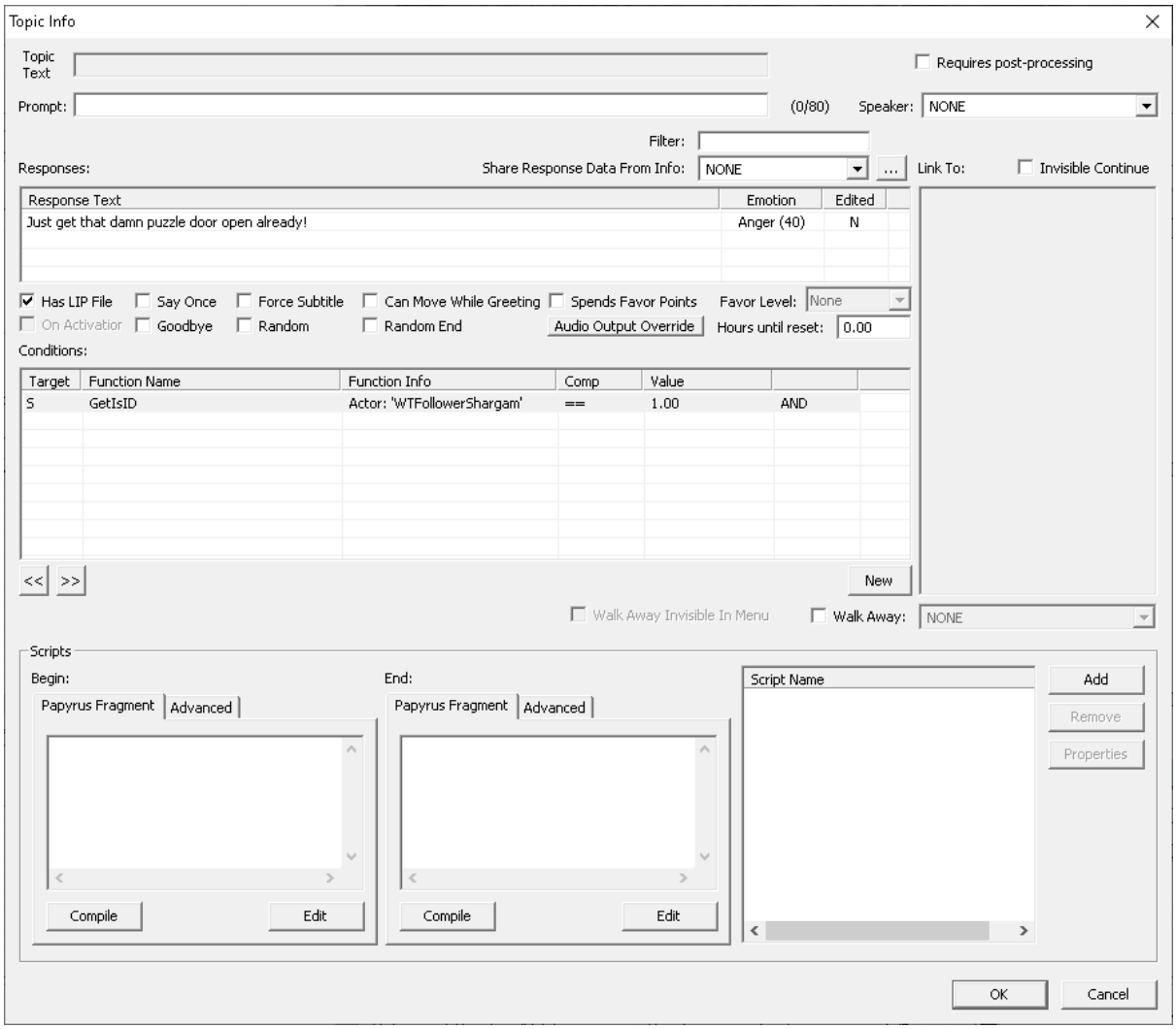


Figure 903 - Topic Info with its condition set up.

Tick Headtrack Player so Shargam looks at the player while saying this line, then click OK to close out of Dialogue Scene Action.

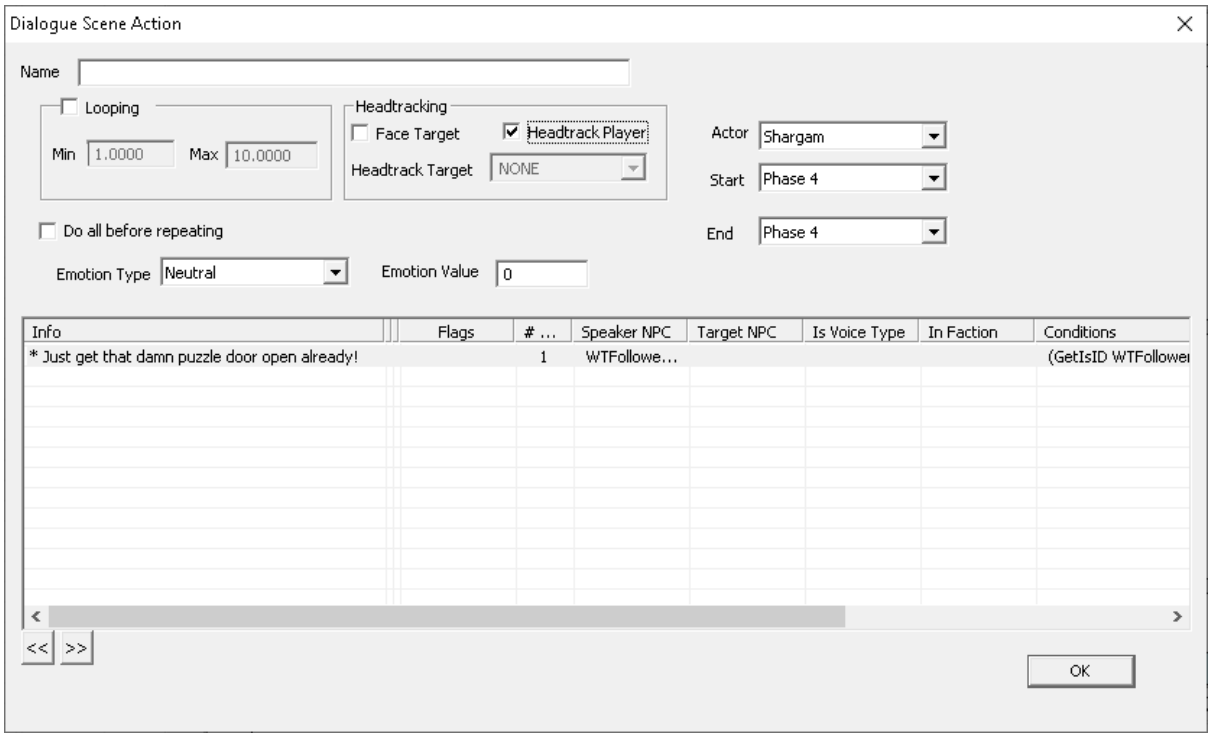


Figure 904 - Dialogue Scene Action with a line of dialogue.

So now after Daenlit speaks, there'll be a three second pause before Shargam runs over to the puzzle door and demands that the player hurries up and opens it.

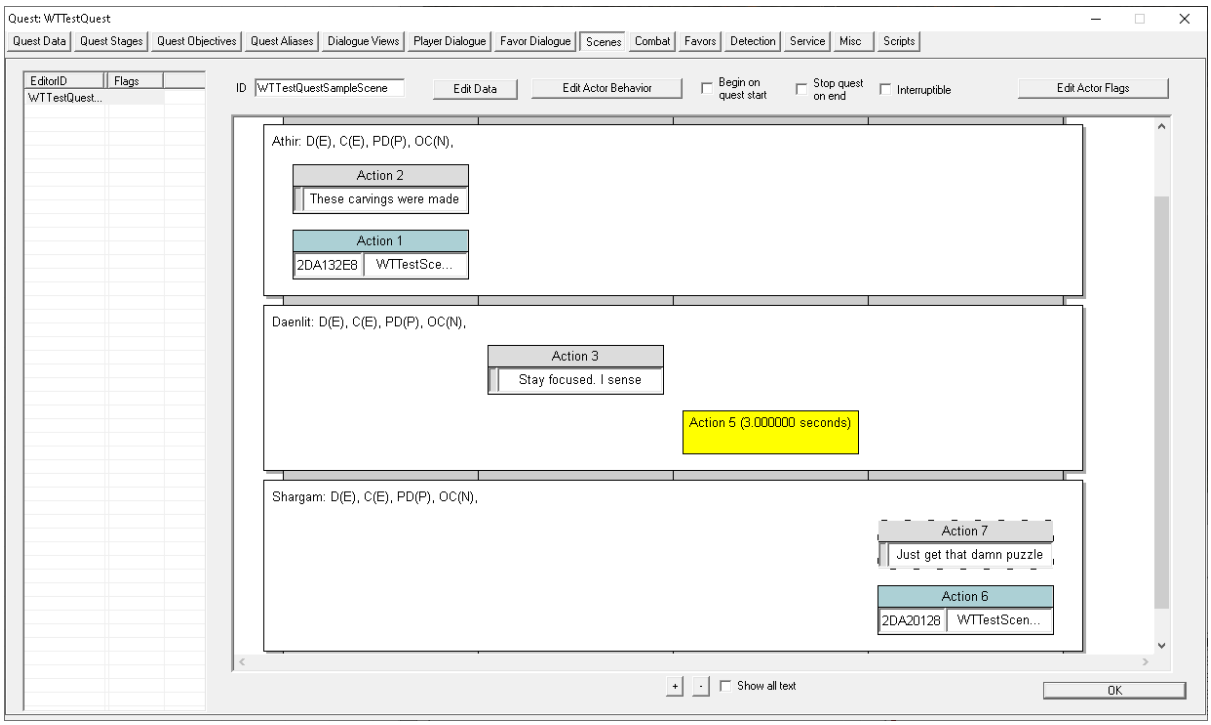


Figure 905 - Four phases set up.

To set up voice acting for this scene, follow the steps in the section Adding Voice Acting and Lip Synching.

We can configure interrupt behaviour for this scene by clicking on the Edit Actor Behaviour button.

By default, the scene will end automatically if any NPCs involved die or engage in combat. You can change this here if you wish but for this example, I'm just going to leave it at its defaults.

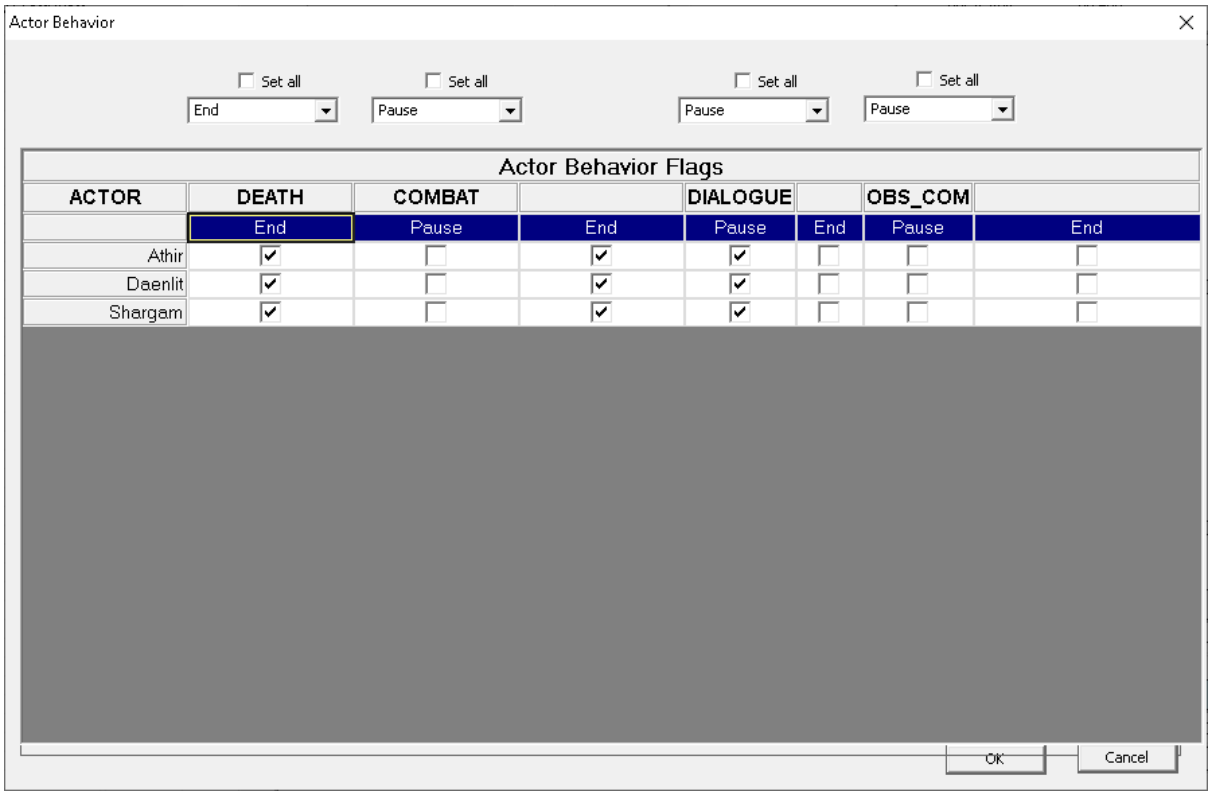


Figure 906 - Configuring interrupt behaviour.

The final thing we need to do is add in a trigger to begin this scene. In the render window, I highlighted one of the Hall Of Stories tileset pieces to set where I want a new trigger to be placed when I click on the Create Trigger button.

To add in a new trigger, click on the Create Trigger button in the toolbar at the top.

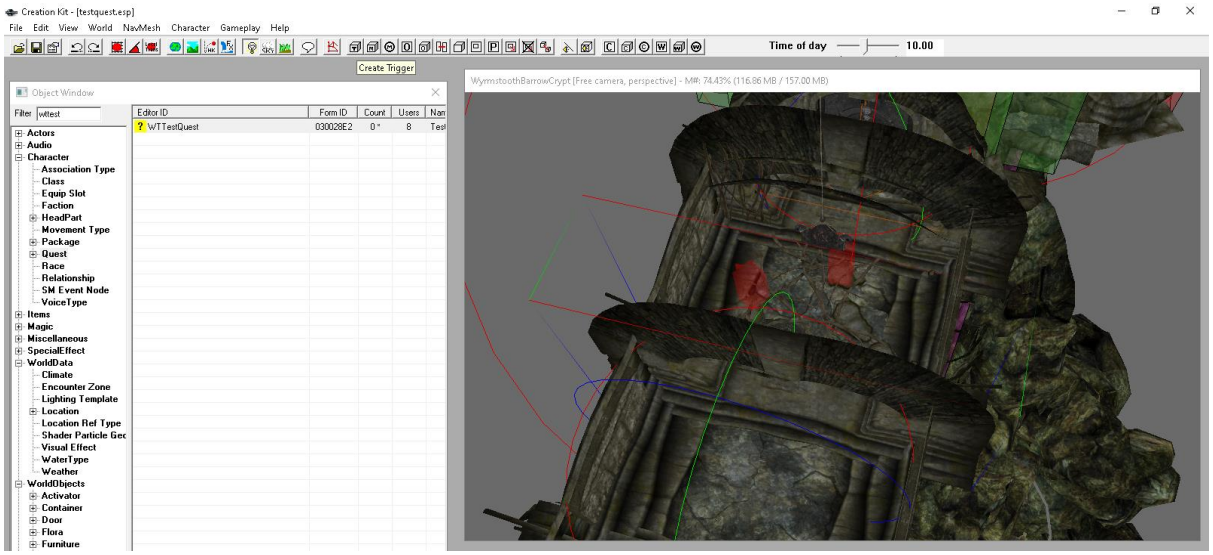


Figure 907 - Create trigger button.

Filter by 'defaultstartscene', select defaultStartSceneTrigger then click OK.

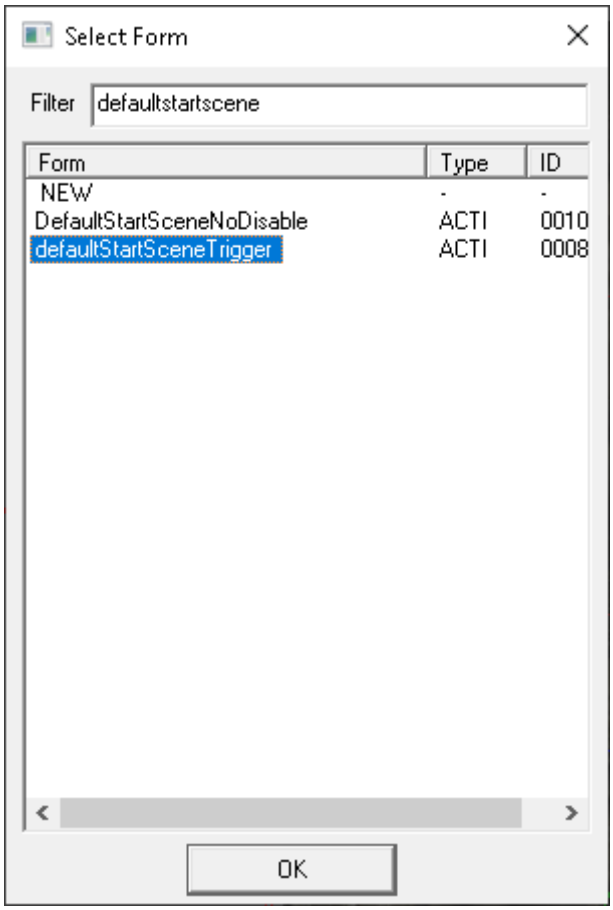


Figure 908 - Finding the defaultStartSceneTrigger script.

Our defaultStartSceneTrigger box should appear in the render window. You can resize it as necessary. Just ensure that the player can't bypass it by walking around it.

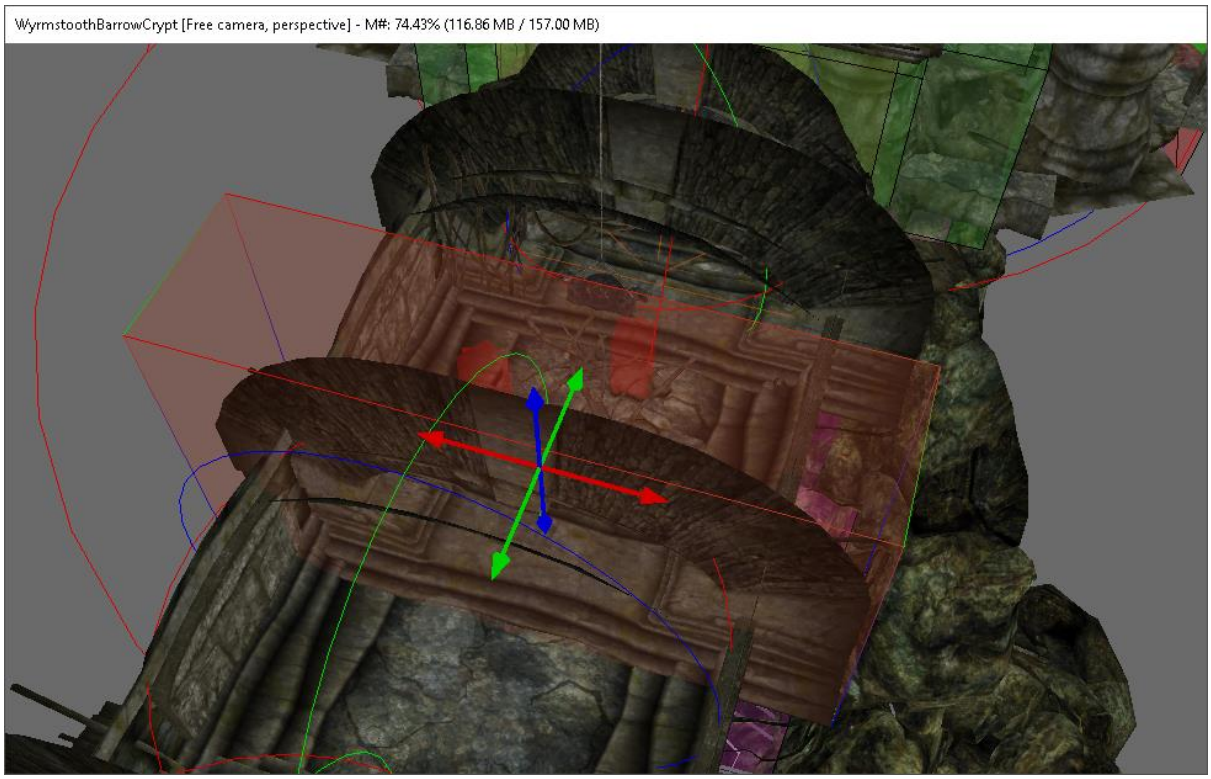


Figure 909 - defaultStartSceneTrigger box.

Double-click or right-click on the trigger box in the render window and select Edit.

Go to the Scripts tab.

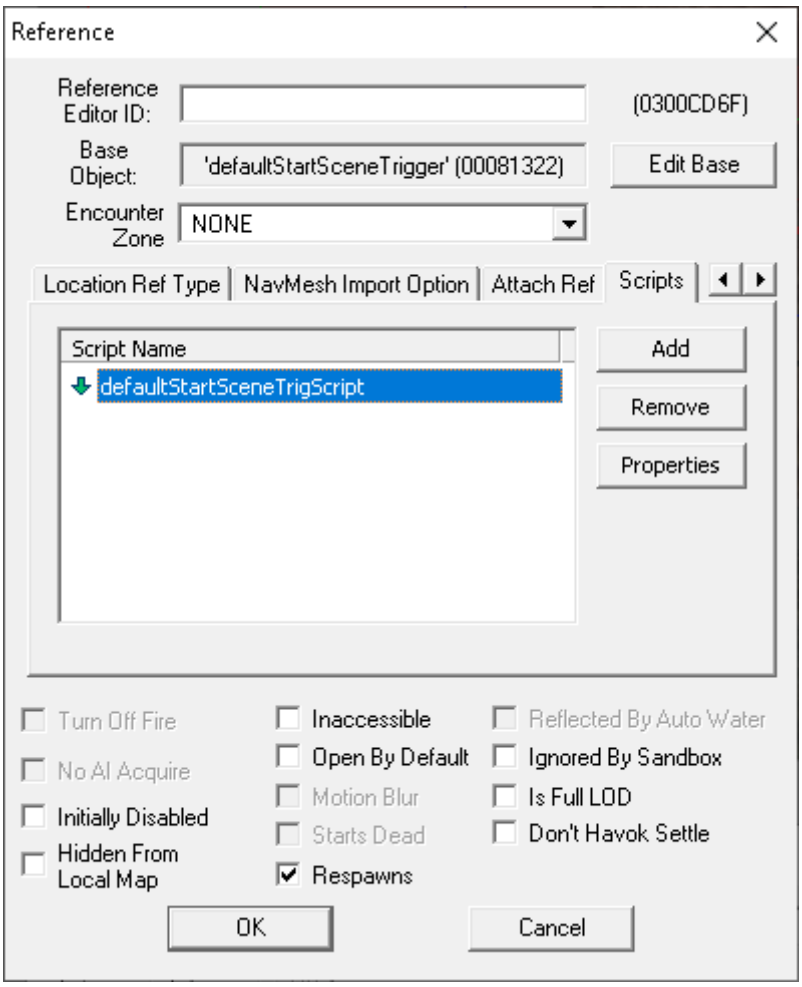


Figure 910 - Scripts tab in Reference properties.

Highlight the defaultStartSceneTrigScript script and select Properties.

First, click on doOnce then click on the Edit Value button.

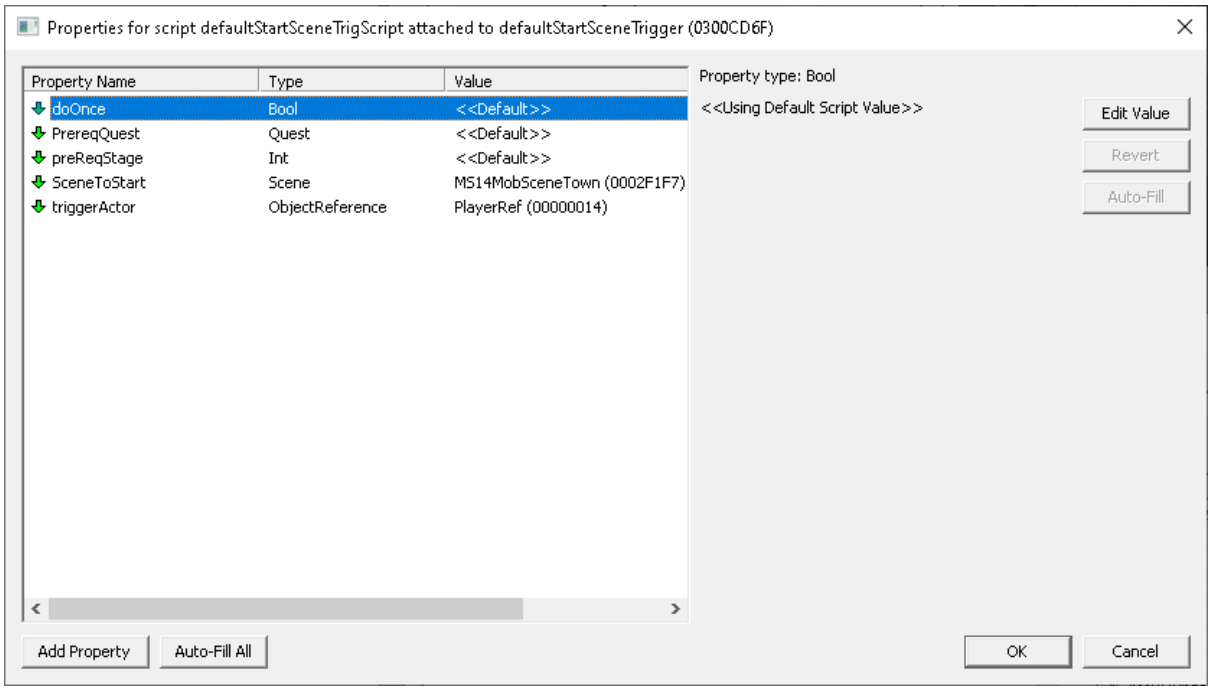


Figure 911 - Changing the default value on doOnce.

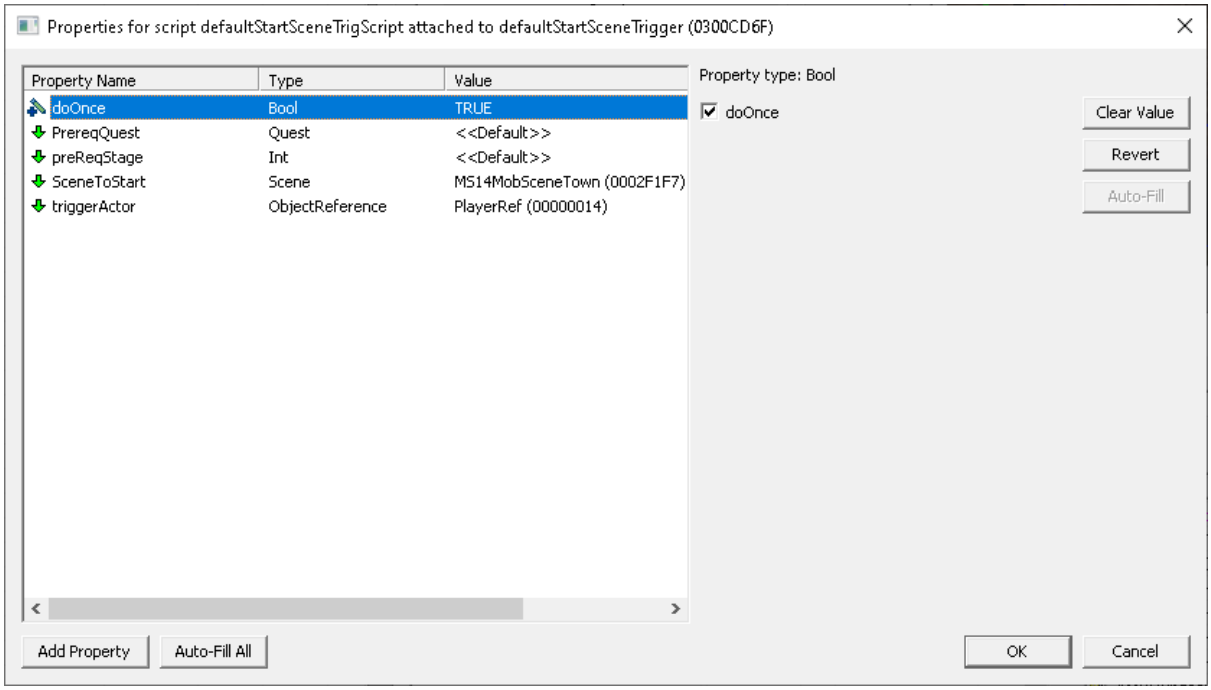


Figure 912 - doOnce set to TRUE.

Next, go to SceneToStart and set the Pick Object drop-down to the ID of scene we want to run when this script is triggered.

For our example, that's going to be WTTestQuestSampleScene.

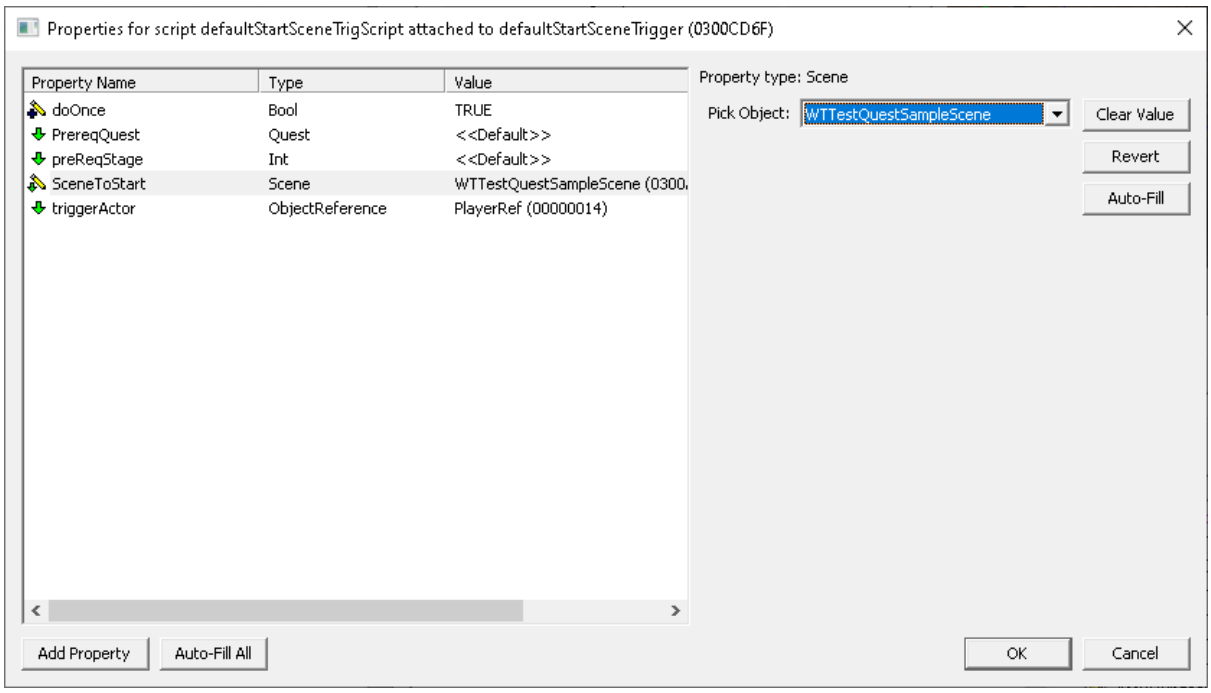


Figure 913 - Setting SceneToStart to WTTestQuestSampleScene.

By default, the script will be triggered when the player enters the trigger box. But what if the NPCs are lagging far behind? We can set this scene to occur when they enter the trigger box instead.

Go to triggerActor and set it to WTFollowerAthir. His in-game reference is in the WyrmstoothHolderCell interior cell.

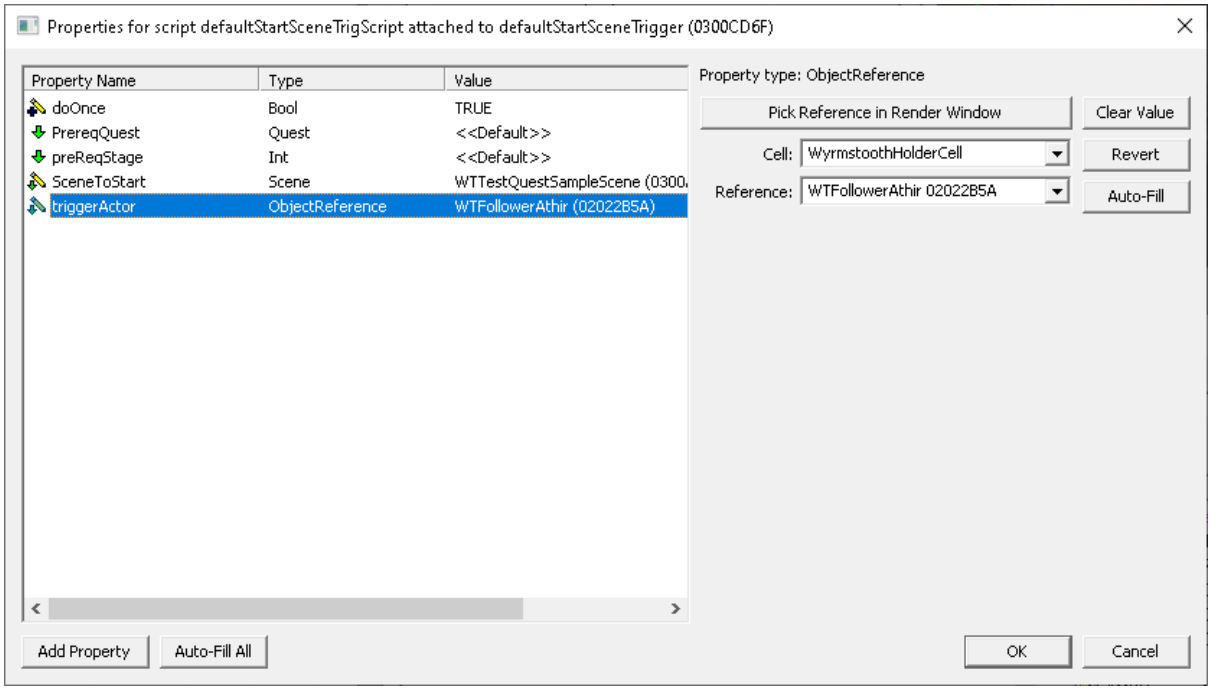


Figure 914 - Setting the NPC to trigger the script.

Click OK to close out of script properties.

Click OK to close out of Reference properties.

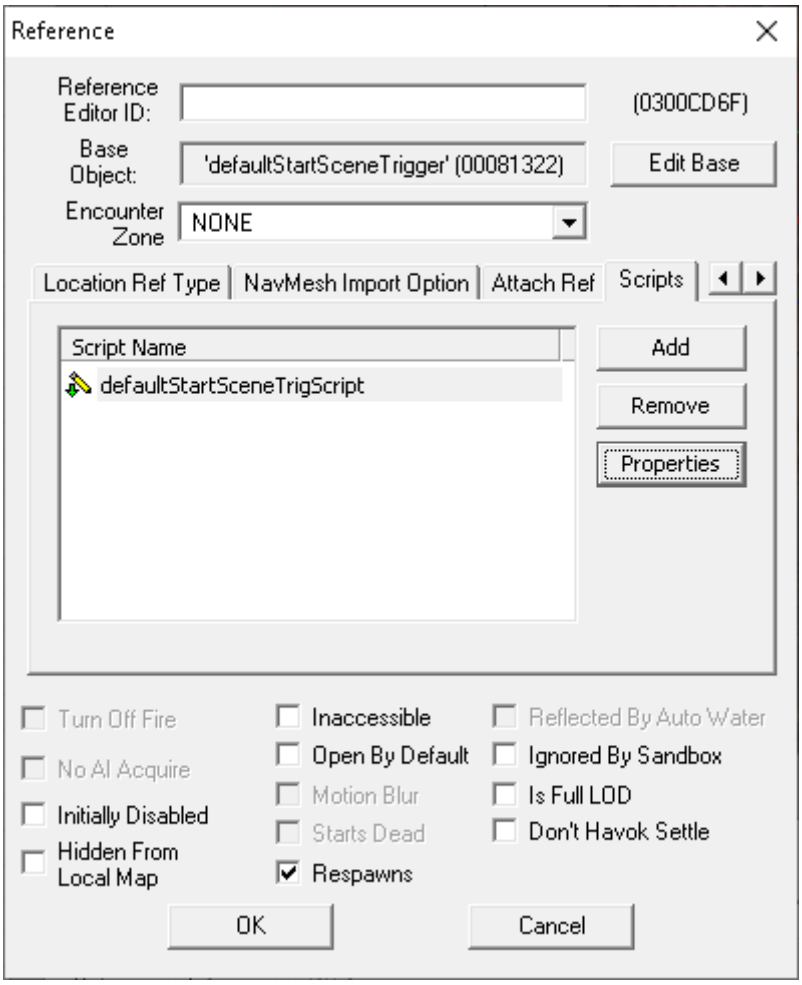


Figure 915 - defaultStartSceneTrigScript properties now flagged as modified.

Our scene will now start once Athir passes through the trigger box instead.

USING TALKING ACTIVATORS

Talking Activators can be used when dialogue needs to be delivered by an inanimate object.

For example, in Wyrmstooth talking activators are used by Vulom both as a voice in the player's head and when talking to the skull. When Lurius Liore delivers his lines during the intro boat ride, a talking activator is used and animated alongside the ship.

Talking Activators can be found in the Object Window under Actors > TalkingActivator.

The easiest way to create a new Talking Activator is to duplicate an existing one. Right-click on, say, DA01VoiceofAzura and select Duplicate.

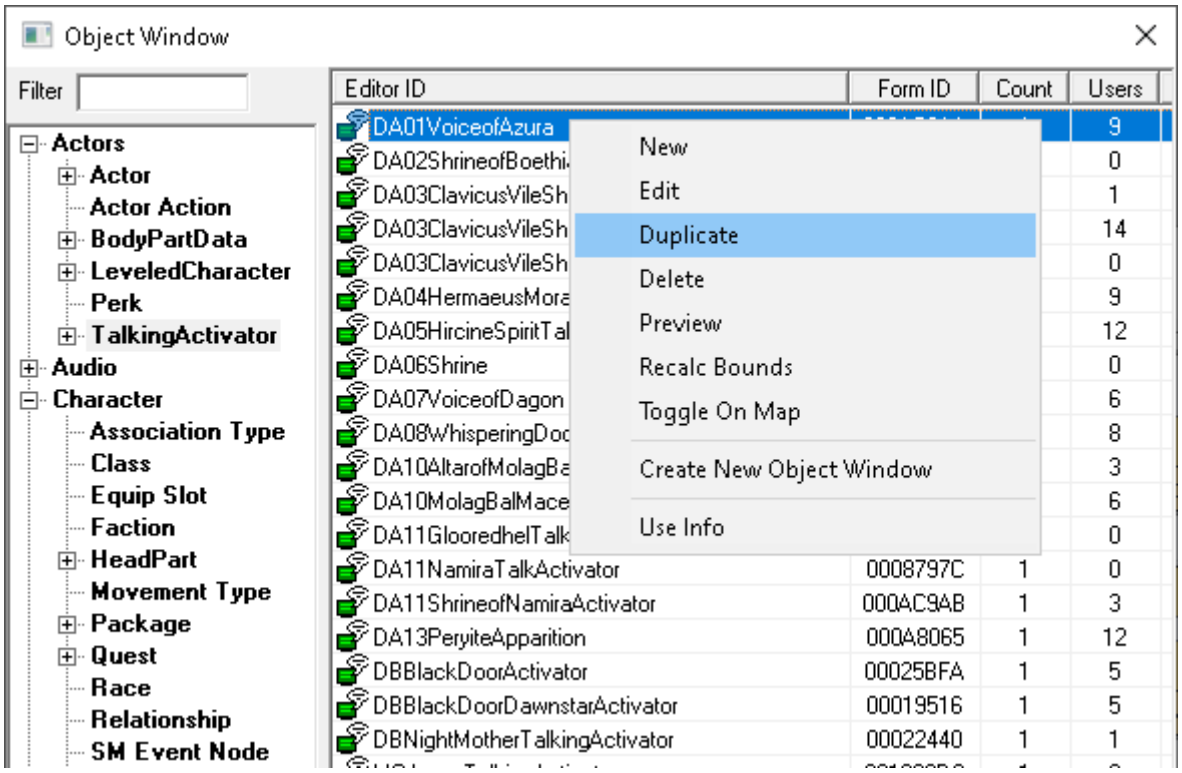


Figure 916 - Duplicating an existing talking activator.

Set a new form ID in the ID field. For this example I'm going to show you how to set up the Vulom talking activators, both for the voice in the player's head and the skull that the player can talk to. I'm going to start with the voice in the player's head so I just set the ID to WTVulomSceneTalkActivator since we'll be using this talking activator in a scene.

Enter a name into the Name field. This name is used when the player interacts with the talking activator.

The Model field defines the type of object we'll be placing in the world. Since we're copying the DA01VoiceofAzura form, this will be set to Markers\Misc\InvisibleCollision01.nif which will be an invisible marker.

It's important to set the Voice Type drop-down to match the voice type of the voice acting. For this example, I'm going to set it to WTVoiceType.

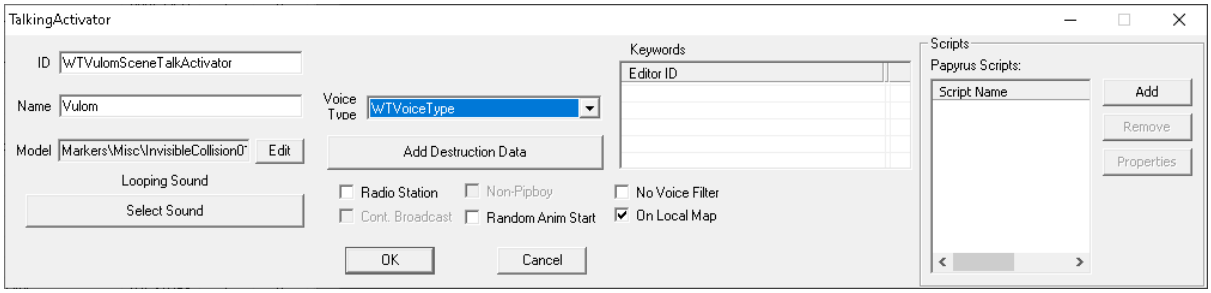


Figure 917 - First talking activator configuration.

Click OK to close out of Talking Activator properties.

When prompted to create a new form, select No.

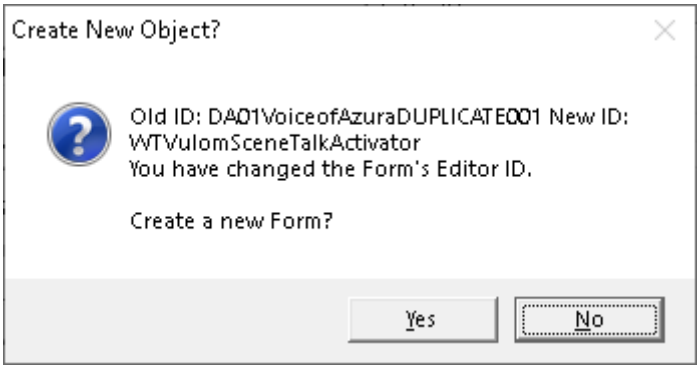


Figure 918 - Create New Object pop-up.

When prompted to confirm rename, select Yes.

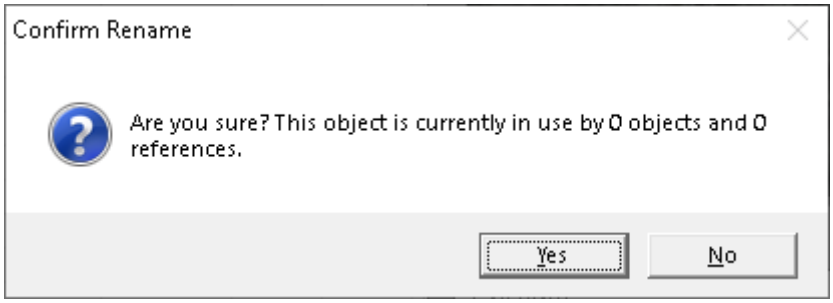


Figure 919 - Confirm Rename pop-up.

Drag and drop the talking activator from the Object Window into the render window to place it in the world.

I placed the WTVulomSceneTalkActivator near Vulom's sarcophagus, beneath the ground. Later I'll be changing the audio override on the voice acting that the talking activator will deliver to make it hearable from any location in this cell regardless of the player's position.

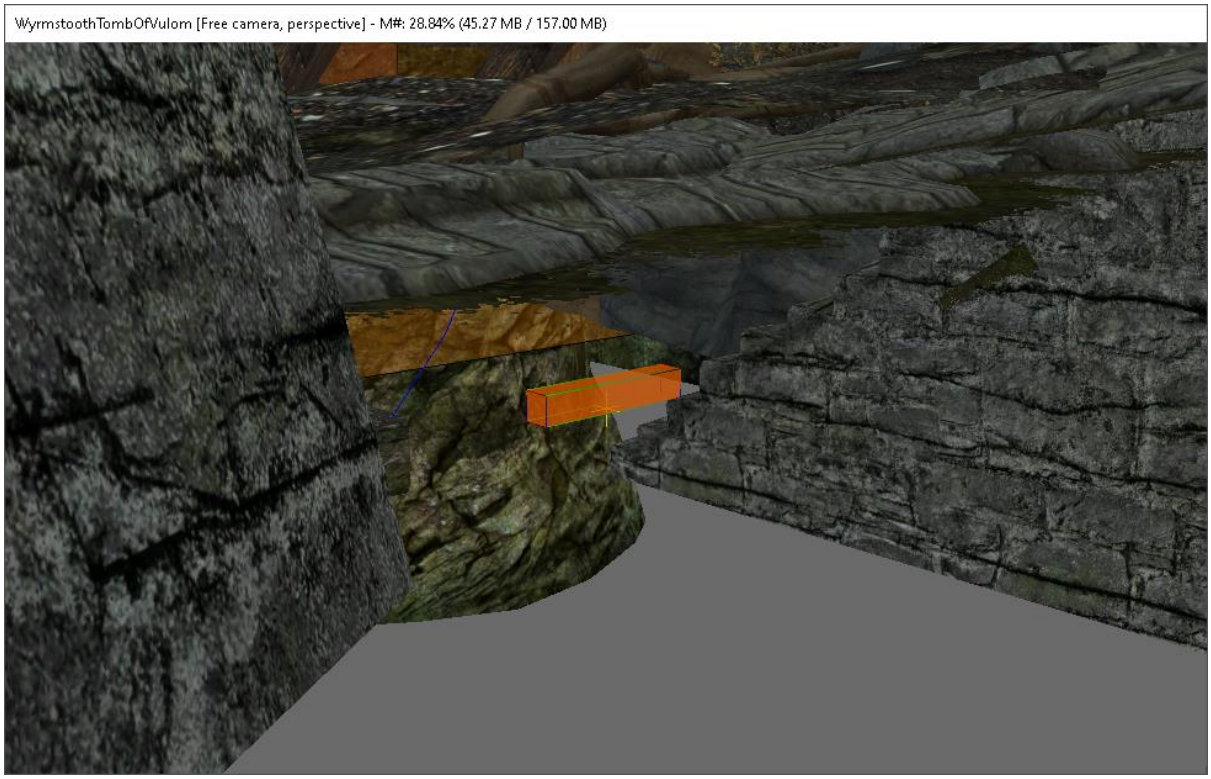


Figure 920 - Placing the first talking activator.

Next, let's create the second talking activator we'll need for this example; the skull that players can interact with directly.

Go back to the Object Window, right-click on one of the existing talking activators and select Duplicate.

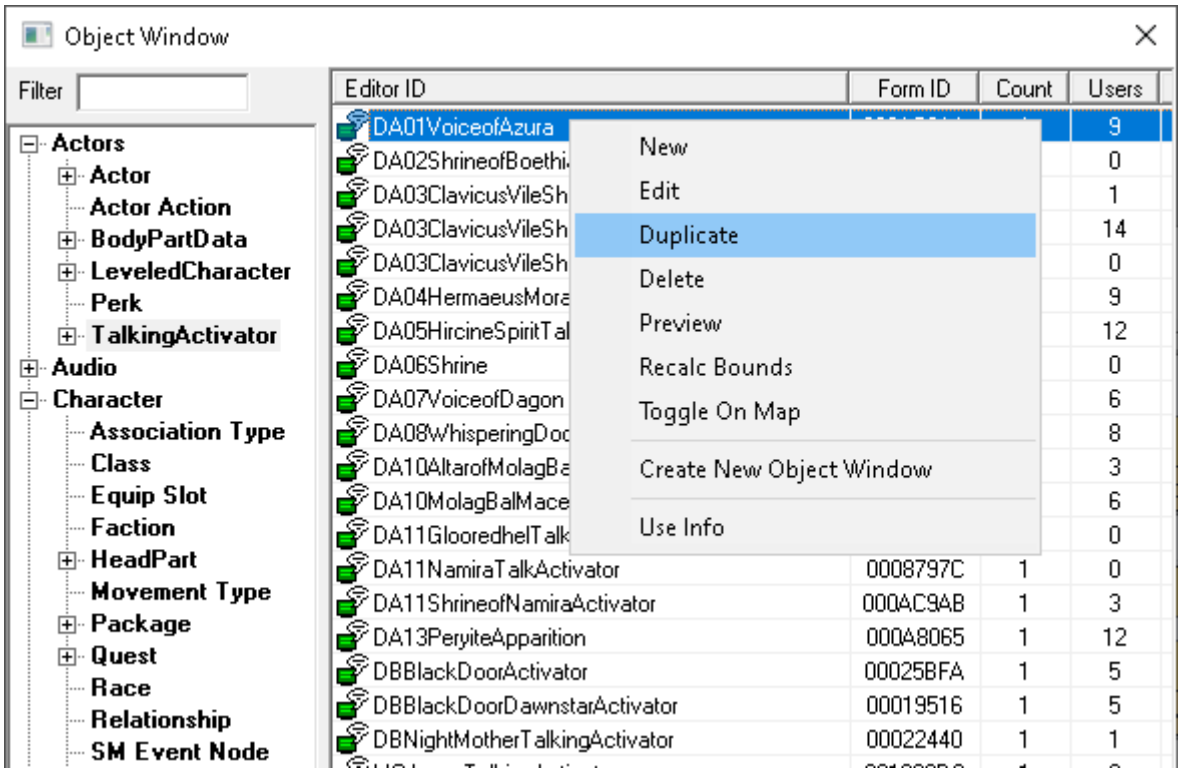


Figure 921 - Duplicating an existing talking activator.

For the second talking activator, I set its ID to WTVulomSkullTalkActivator.

Set the name in the Name field. Again, I set this to Vulom.

This time, I set the Model to Clutter\Bones\HumanSkullFull.nif.

The Voice Type was also set to WTVoiceType.

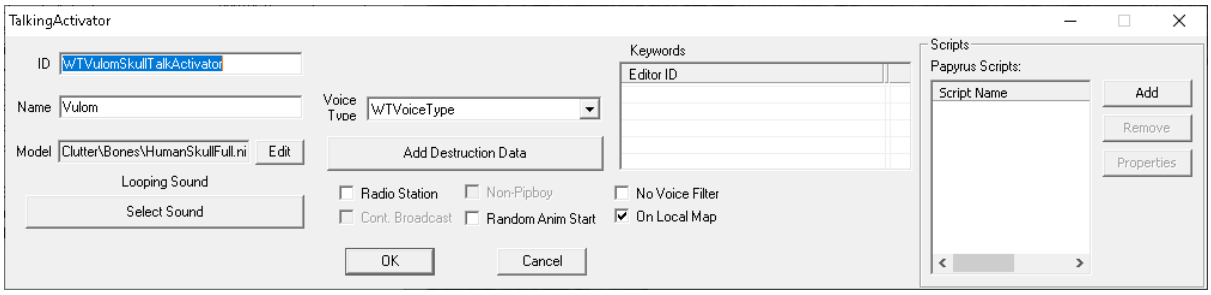


Figure 922 - Second talking activator configuration.

Note: You can retexture the .nif you set here as the talking activator's world model if you want to by clicking on Edit.

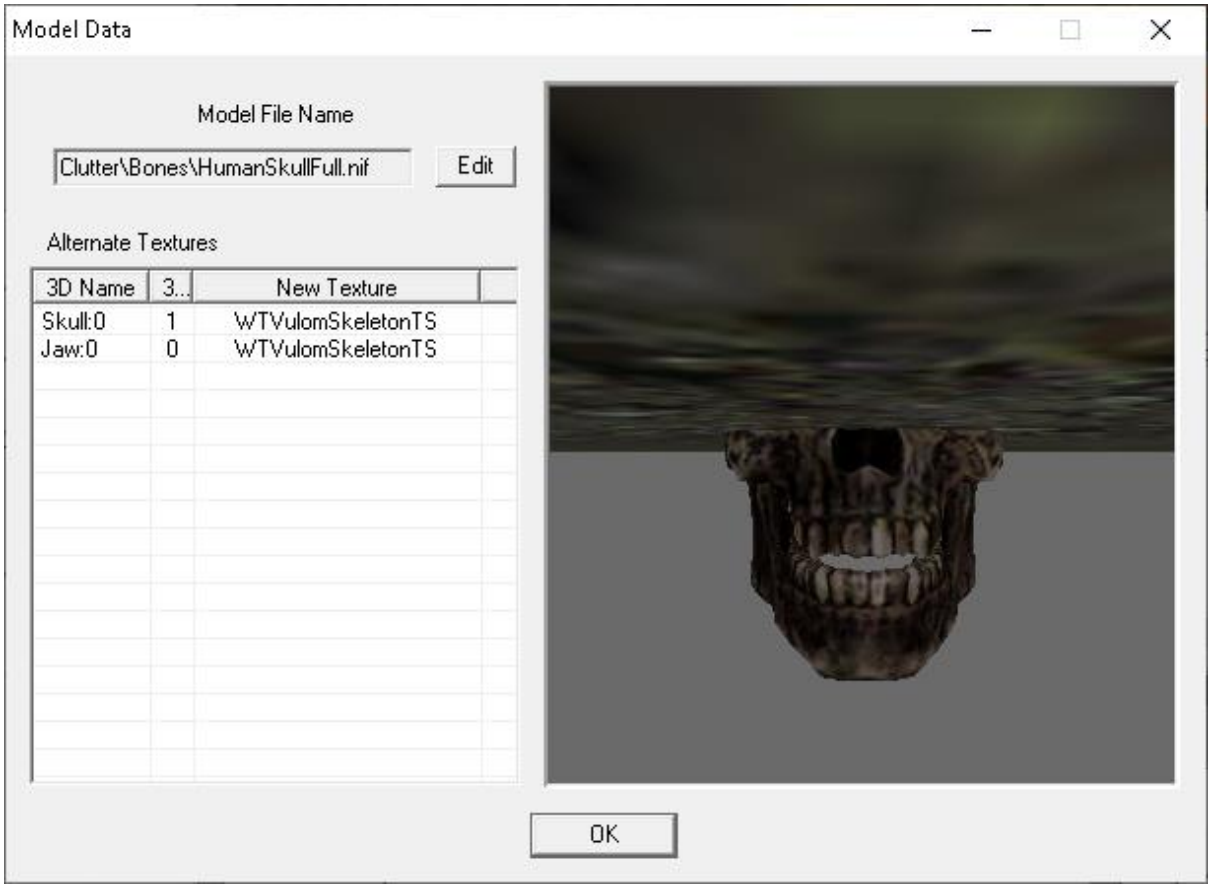


Figure 923 - Retexturing a .nif by defining a texture set.

To add a texture set to a specific part of the .nif, double-click on Skull:0 or Jaw:0.

Highlight the texture set in the form list then click OK.

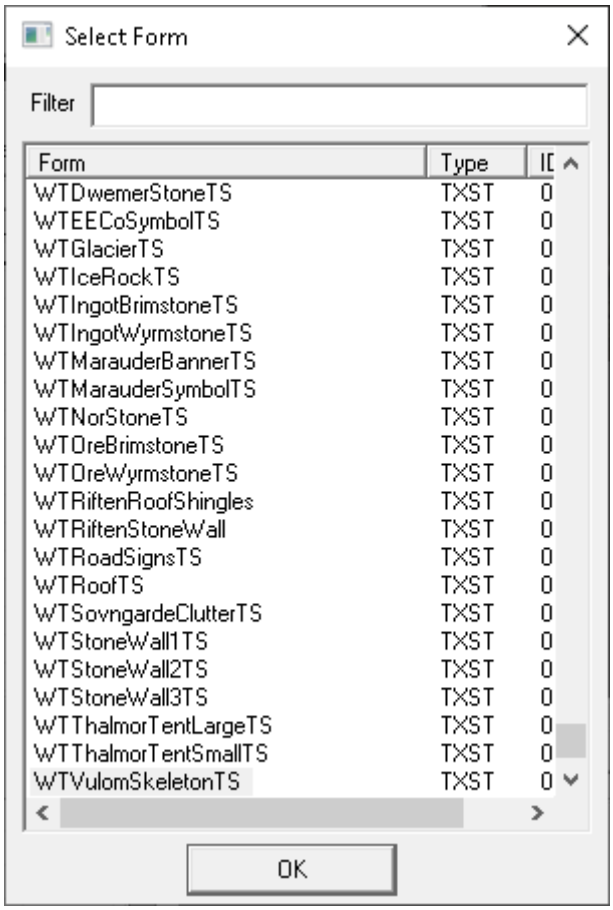


Figure 924 - Texture set form list.

Click OK to close out of Model Data properties if you changed a texture, then click OK to close out of Talking Activator properties.

Again, when prompted to create a new form, select No.

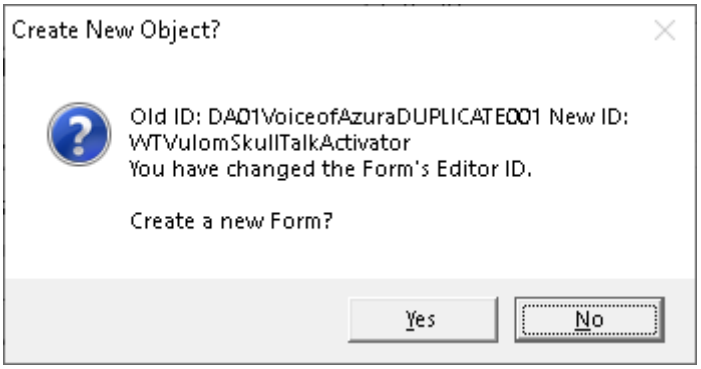


Figure 925 - Create New Object pop-up.

And when prompted to confirm rename, select Yes.

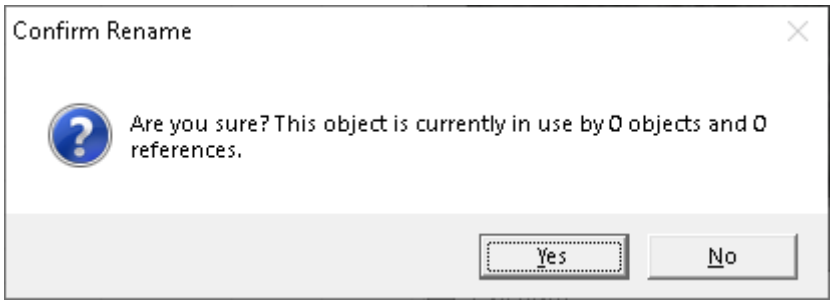


Figure 926 - Confirm Rename pop-up.

Drag and drop the second talking activator to place it. It's important to ensure that it is not clipping with the collision of any surrounding objects so the player can activate it from their position.

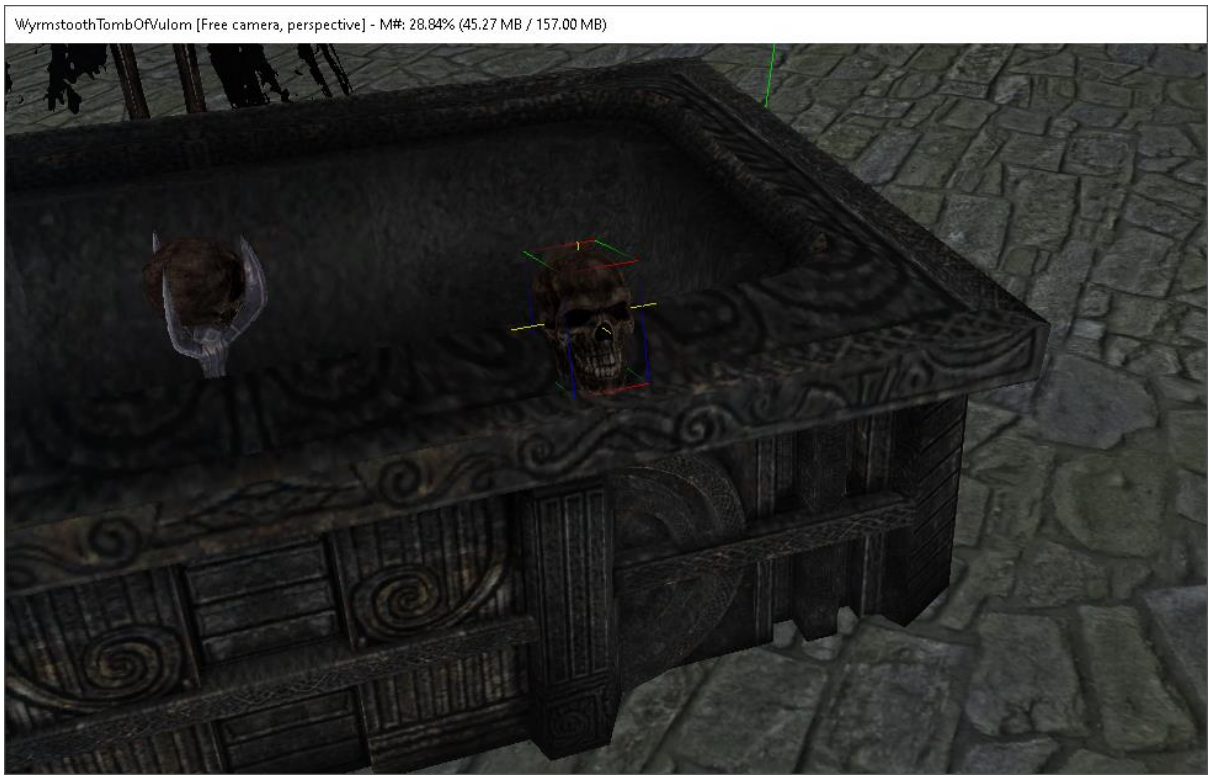


Figure 927 - Placing the second talking activator.

Next, we'll take a look at how Vulom's dialogue was set up in his associated side-quest.

If you need step-by-step instructions on setting up the quest dialogue, see the sections Adding Dialogue to a Quest, Adding Scenes, and Adding Voice Acting and Lip Synching.

The first thing I did was add both talking activators into Vulom's quest as quest aliases.

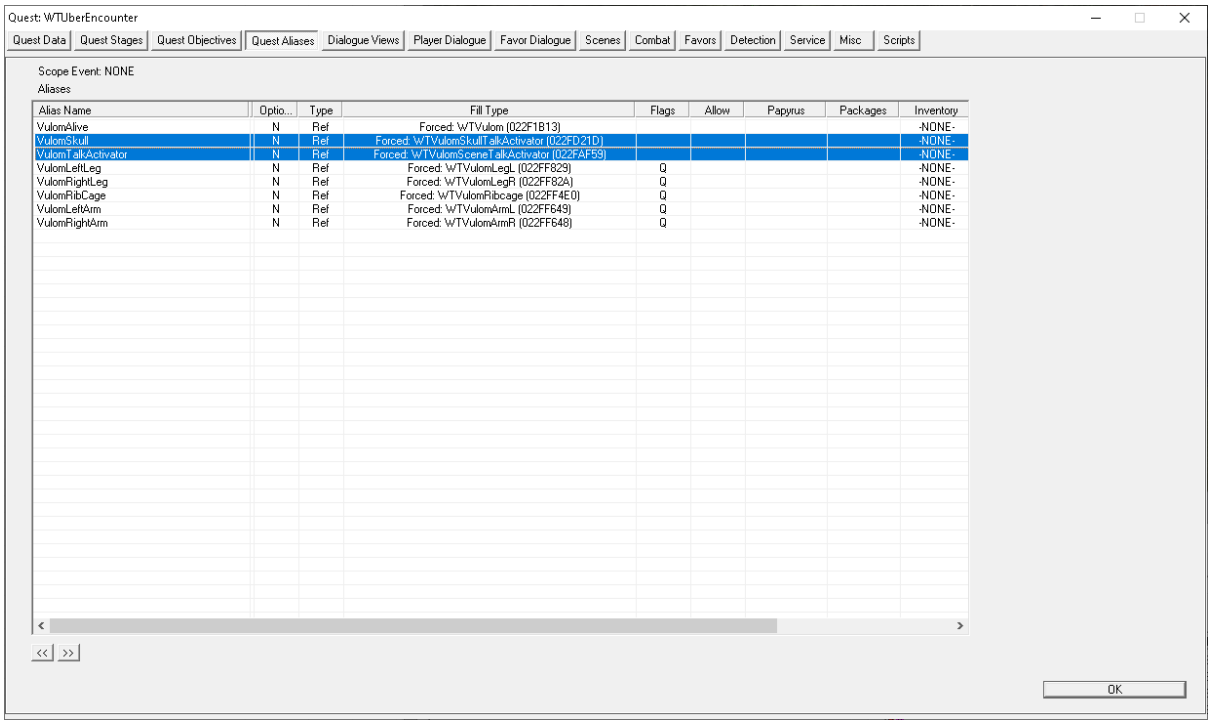


Figure 928 - Talking activators added as quest aliases.

When setting up the alias, just set the Fill Type to Specific Reference and select the reference in the render window. For talking activators, there's no need to tick any other options.

The first talking activator uses a scene which is triggered by the player when they enter Vulom's tomb for the first time.

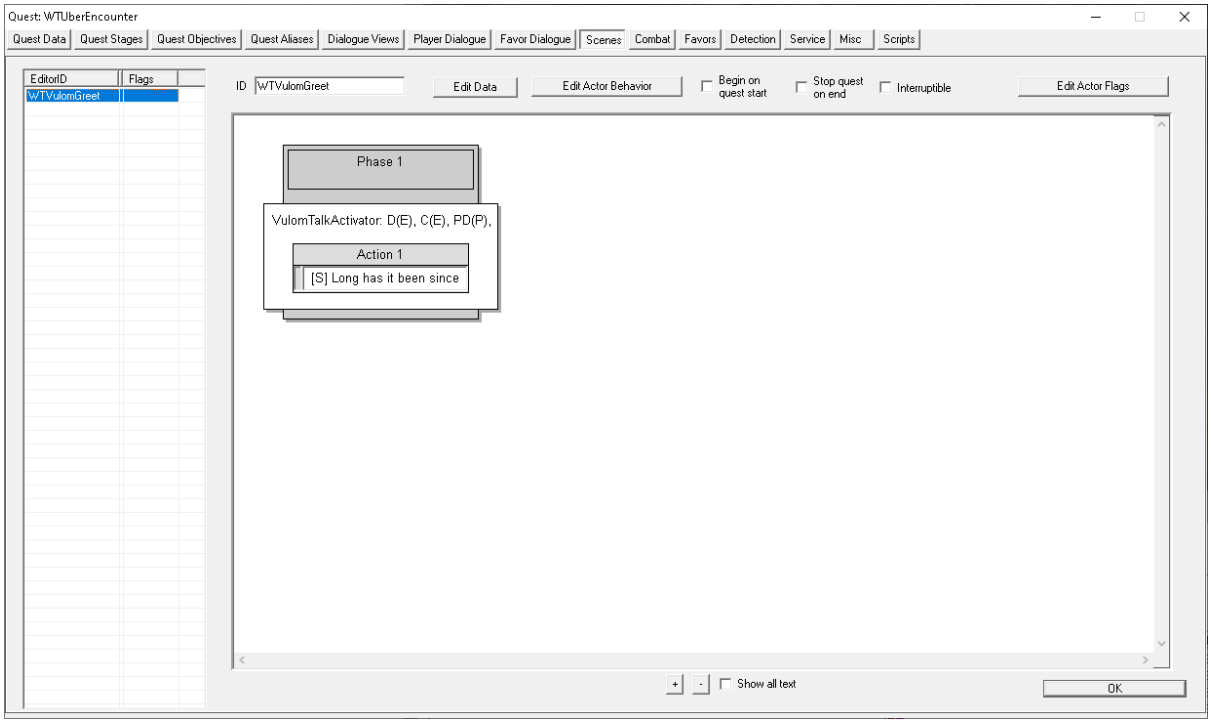


Figure 929 - Scene for the first talking activator.

This scene consists of a single phase with the first talking activator set up to deliver a single line of dialogue.

Because the talking activator itself is far away from the player when it's activated, I ticked 'Force Subtitle' to ensure subtitles are always shown regardless of proximity to the speaker.

To ensure the voice acting can be heard in the player's head, I set the Audio Override to SOMDialogue2D.

Note: There's no need to set any conditions.

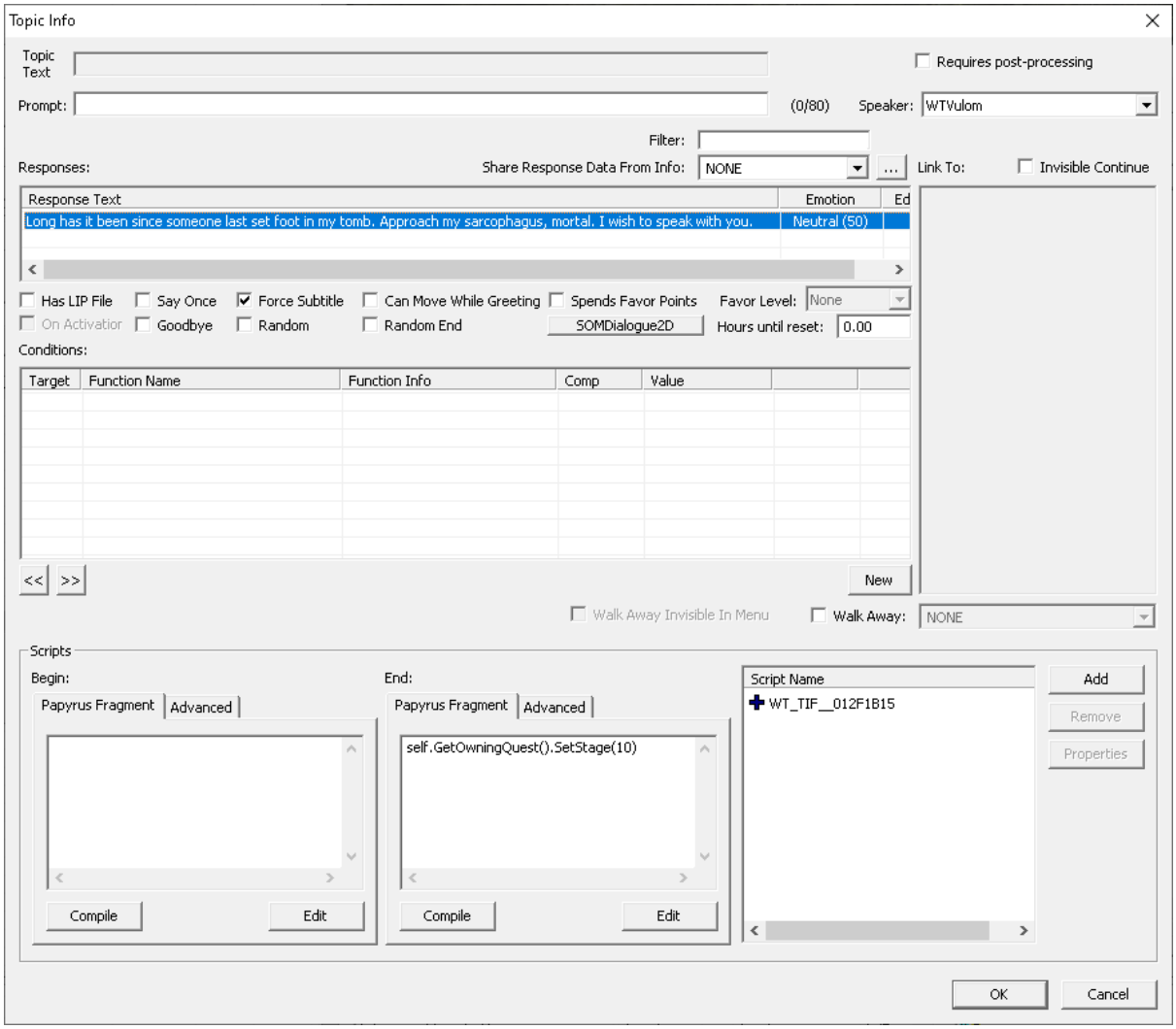


Figure 930 - Topic Info setup for the first talking activator.

Once the line of dialogue has been spoken, Vulom's quest is advanced to stage 10 which puts an objective in the player's journal.

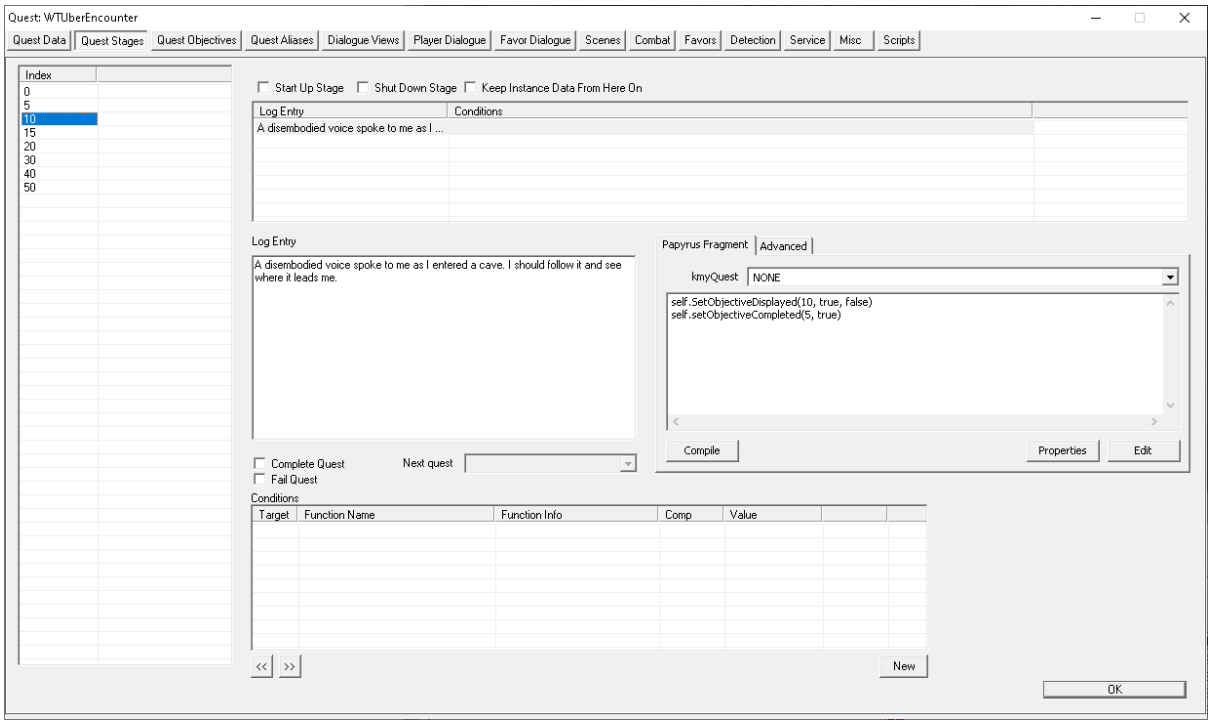


Figure 931 - Stage 10 of Vulom's quest.

A defaultStartSceneTrigger trigger is placed at the entrance to the large cavern containing Vulom's sarcophagus to start the scene.

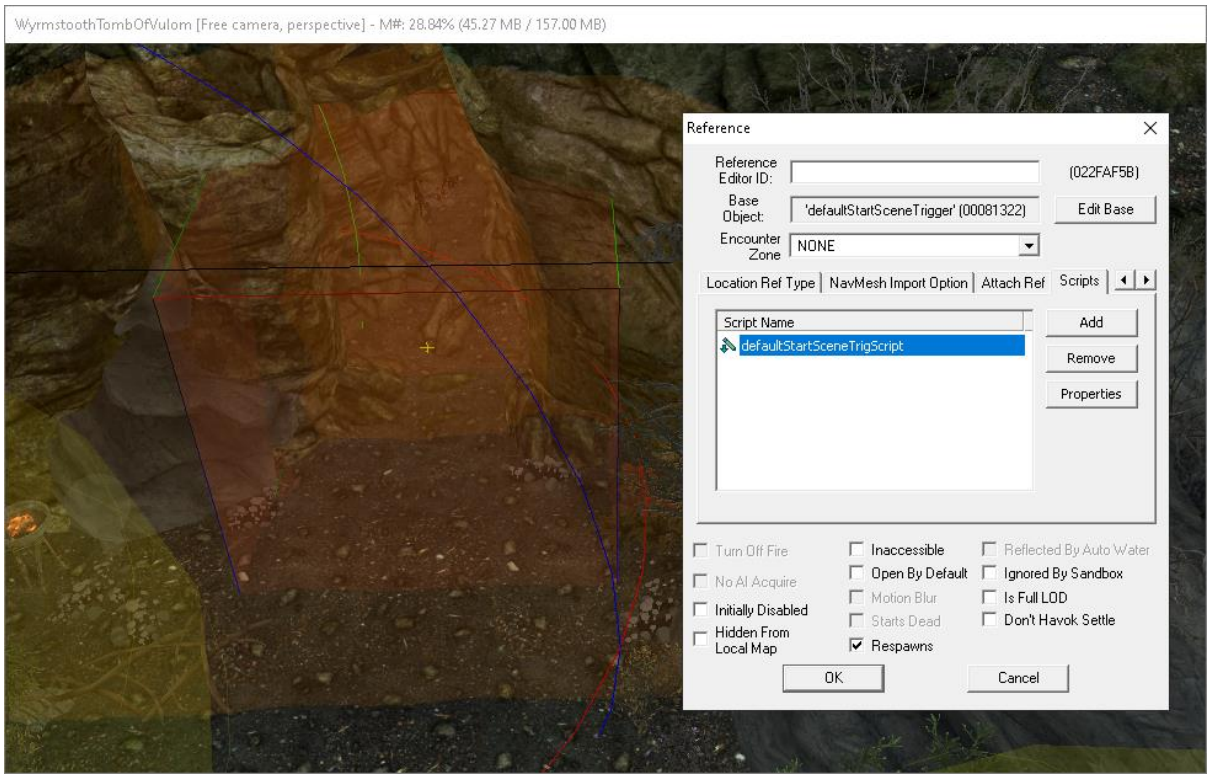


Figure 932 - defaultStartSceneTrigger used to start the first talking activator's scene.

This script is configured to run only once and to run the WTVulomGreet scene. By default, this trigger can only be activated by the player.

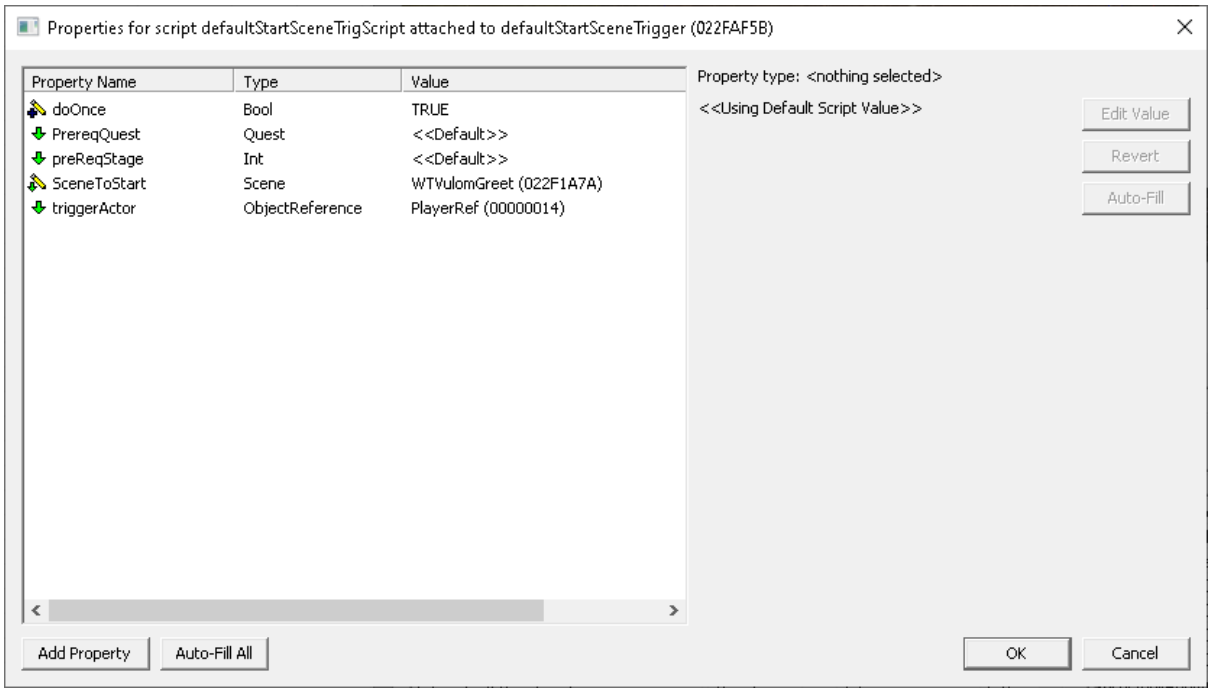


Figure 933 - Script properties on the defaultStartSceneTrigScript script.

Dialogue for the second talking activator is set up in Dialogue Views.

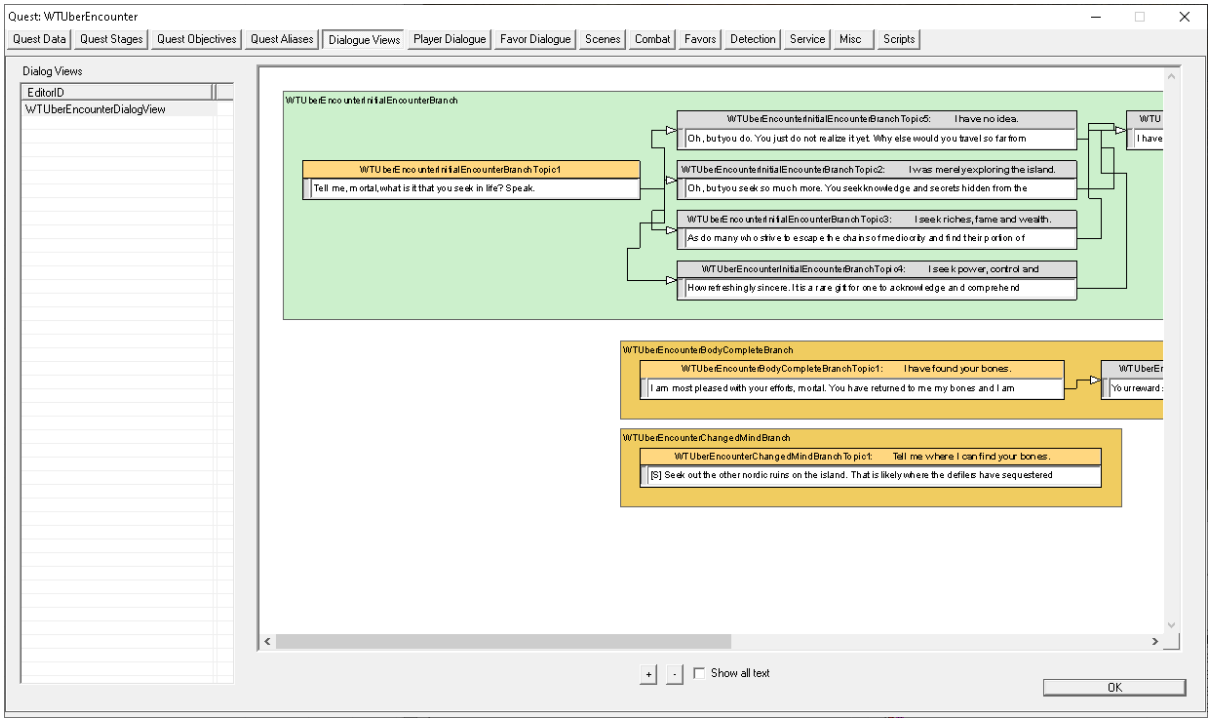


Figure 934 - The second talking activator's dialogue.

Each topic has two conditions; one to ensure the player is speaking with the second talking activator, Vulom's skull, and the other to check the current stage of Vulom's quest.

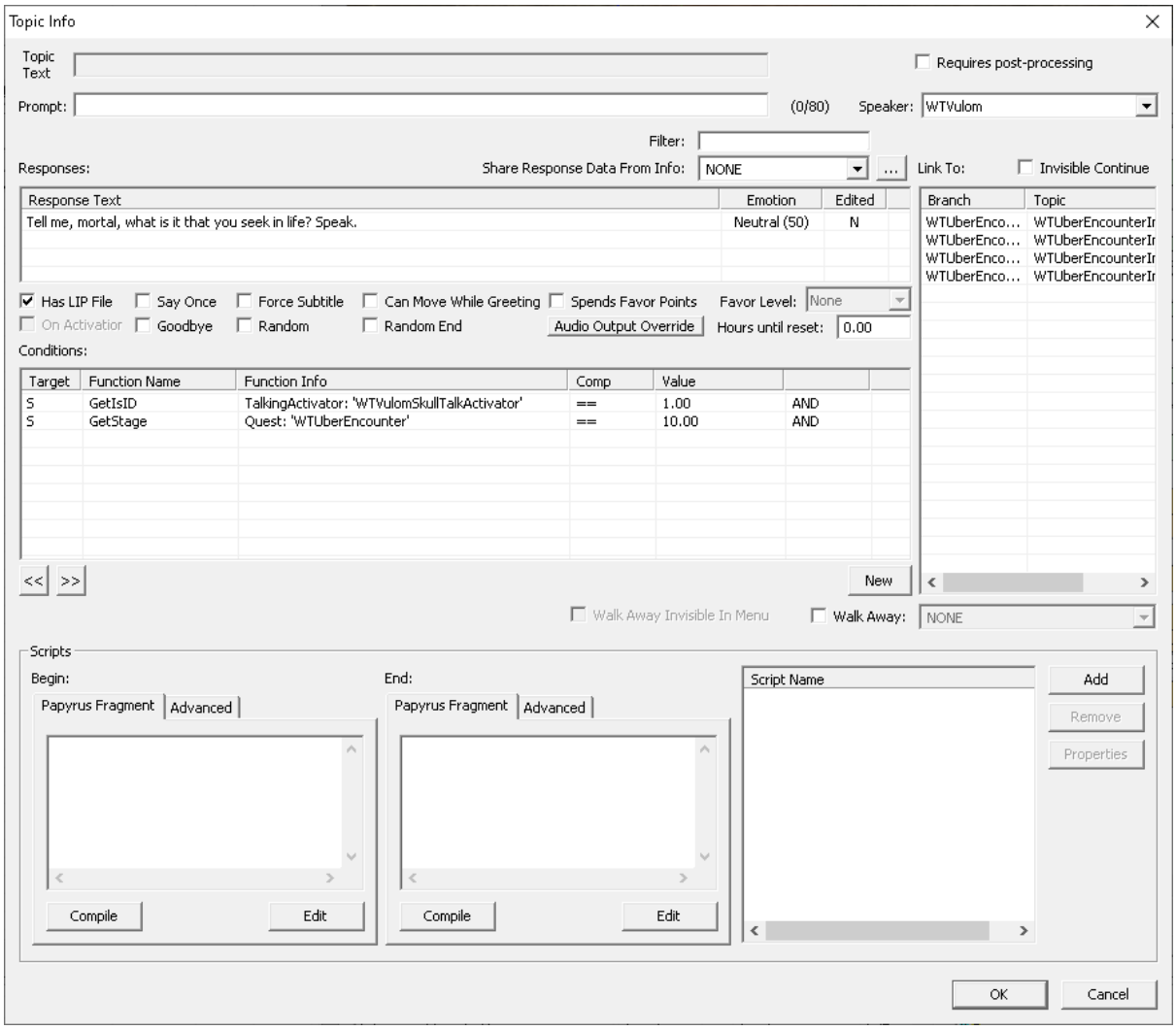


Figure 935 - Topic Info setup for the second talking activator's dialogue.

CREATING AN MCM

In this section I'll be going over the basics for setting up an MCM for your mod.

Note: You will need to install SKSE and SkyUI (LE/SSE).

This section will require a fair bit of papyrus scripting. If scripting makes you squeamish, you're probably not going to enjoy this. That said, I've tried to explain each line of code to make it as easy as possible to understand.

Let's start by setting up a new quest for the MCM script.

In the Object Window, go to Character > Quest, right-click on one of the existing quests and select New.

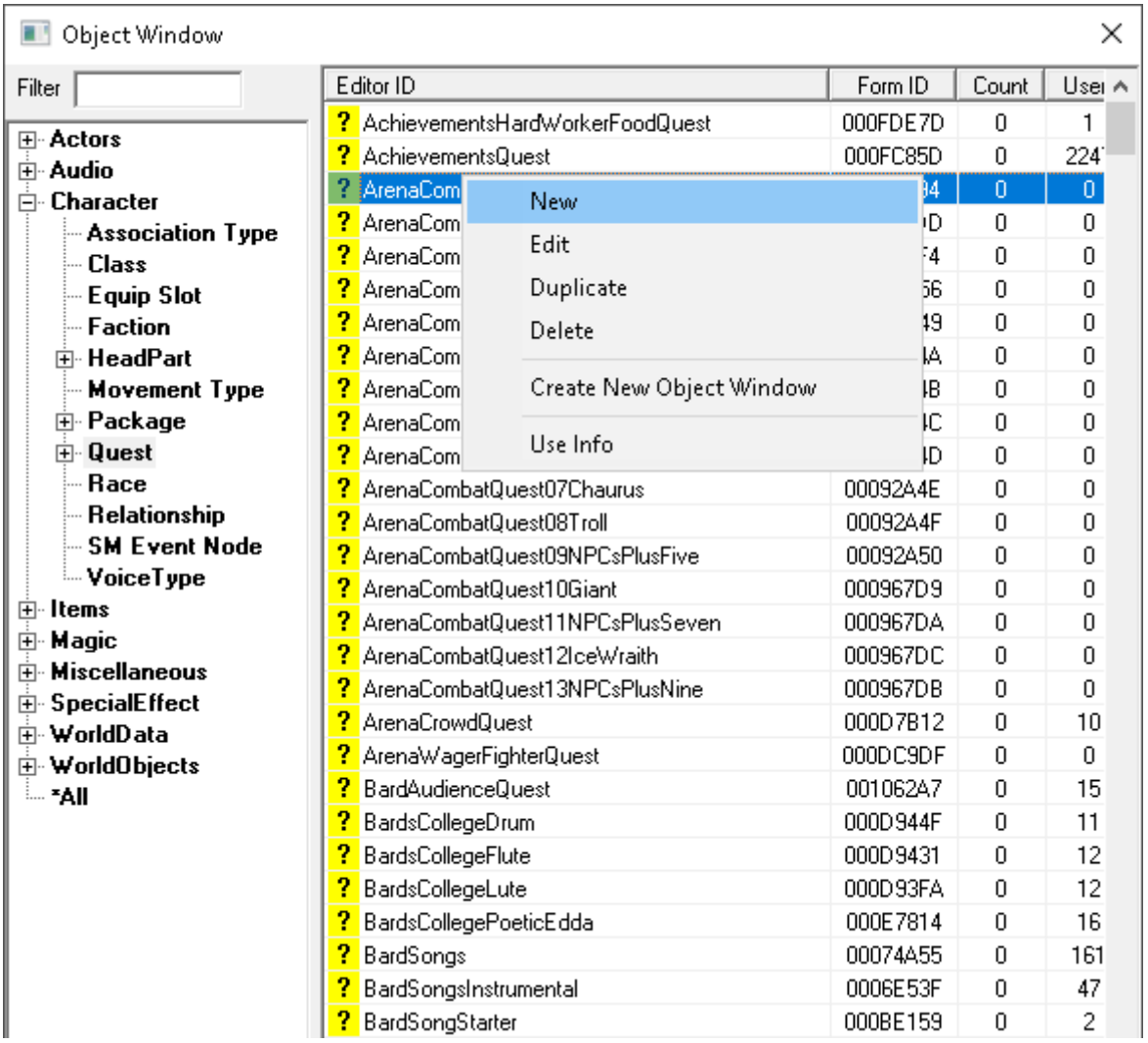


Figure 936 - Creating a new quest for the MCM.

Enter in a quest ID. For this example I entered WTTestMCM.

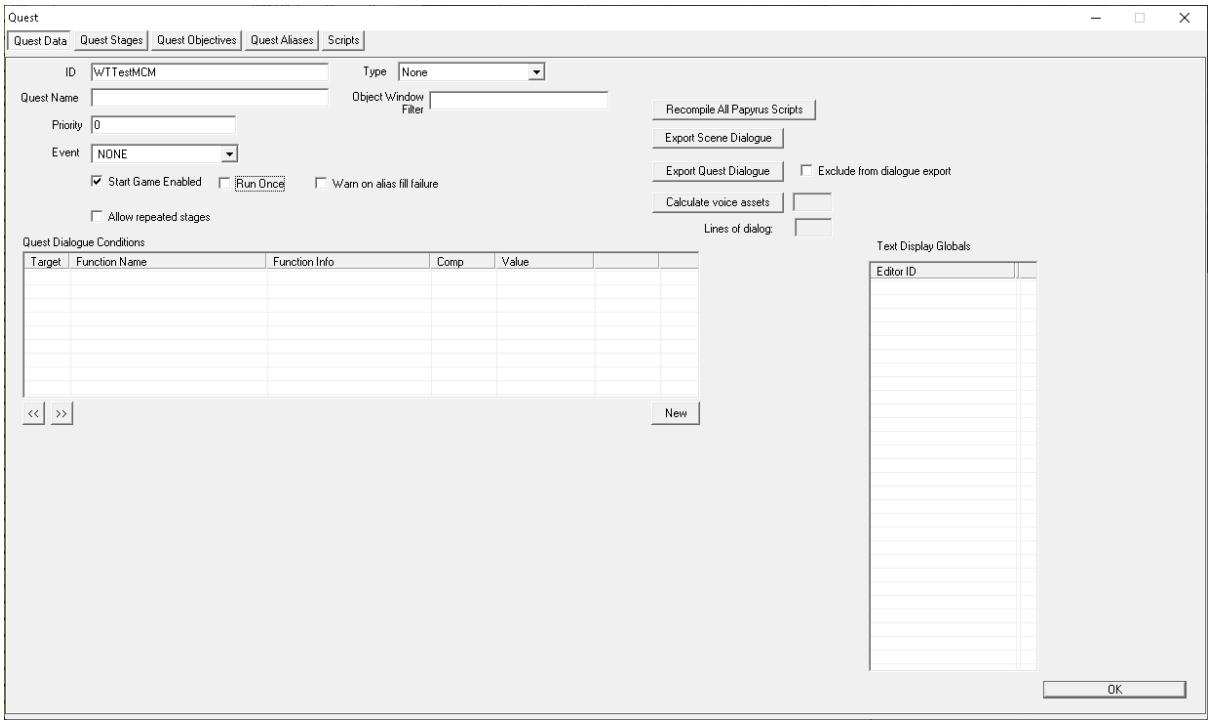


Figure 937 - Quest Data tab.

Before continuing, click OK to close out of Quest properties. Right-click on the quest we just created and select Edit to re-open its properties.

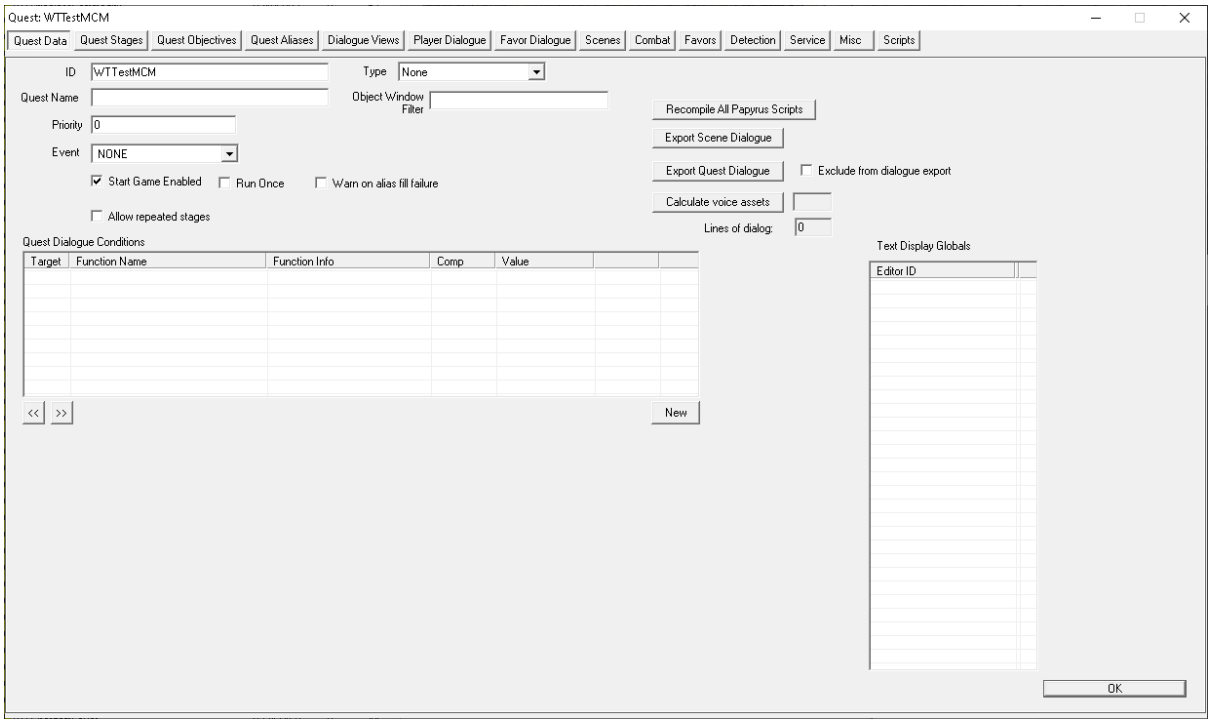


Figure 938 - Quest properties after closing and re-opening it.

We should now see more tabs then before.

Go to the Quest Aliases tab, right-click in the aliases list and select New Reference Alias.

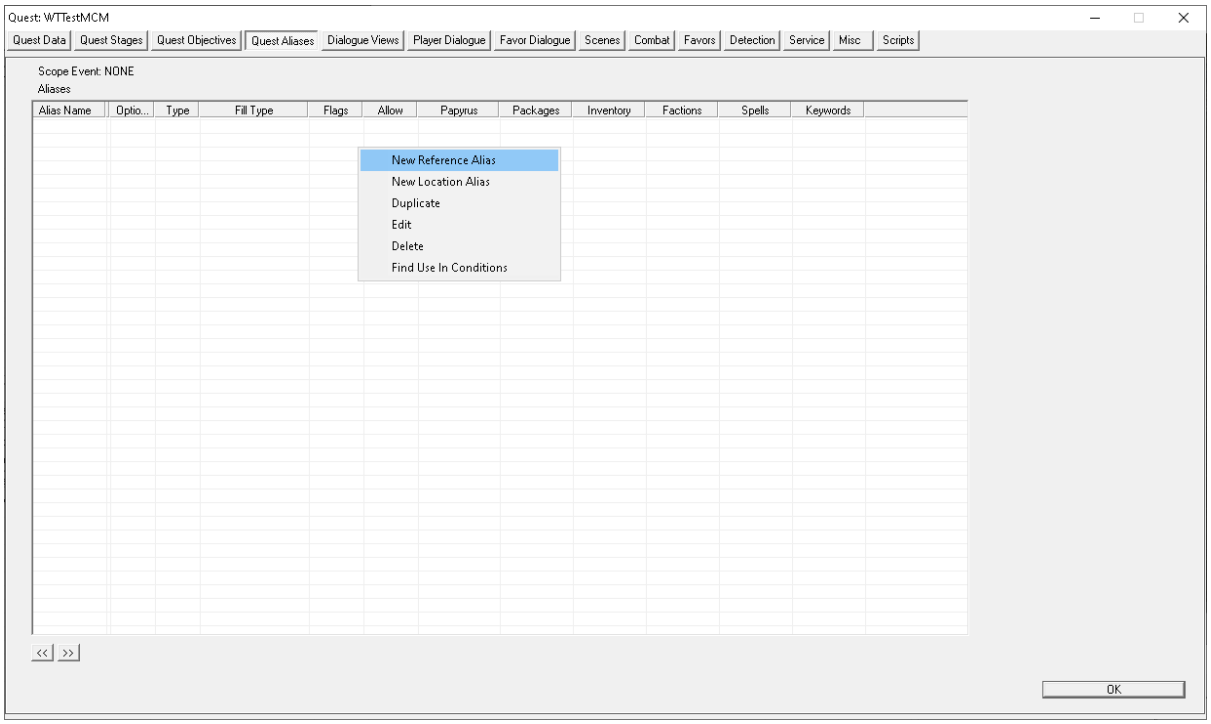


Figure 939 - Creating a new reference alias.

Set the Alias Name field to PlayerREF.

Under the Scripts section, click on the Add button.

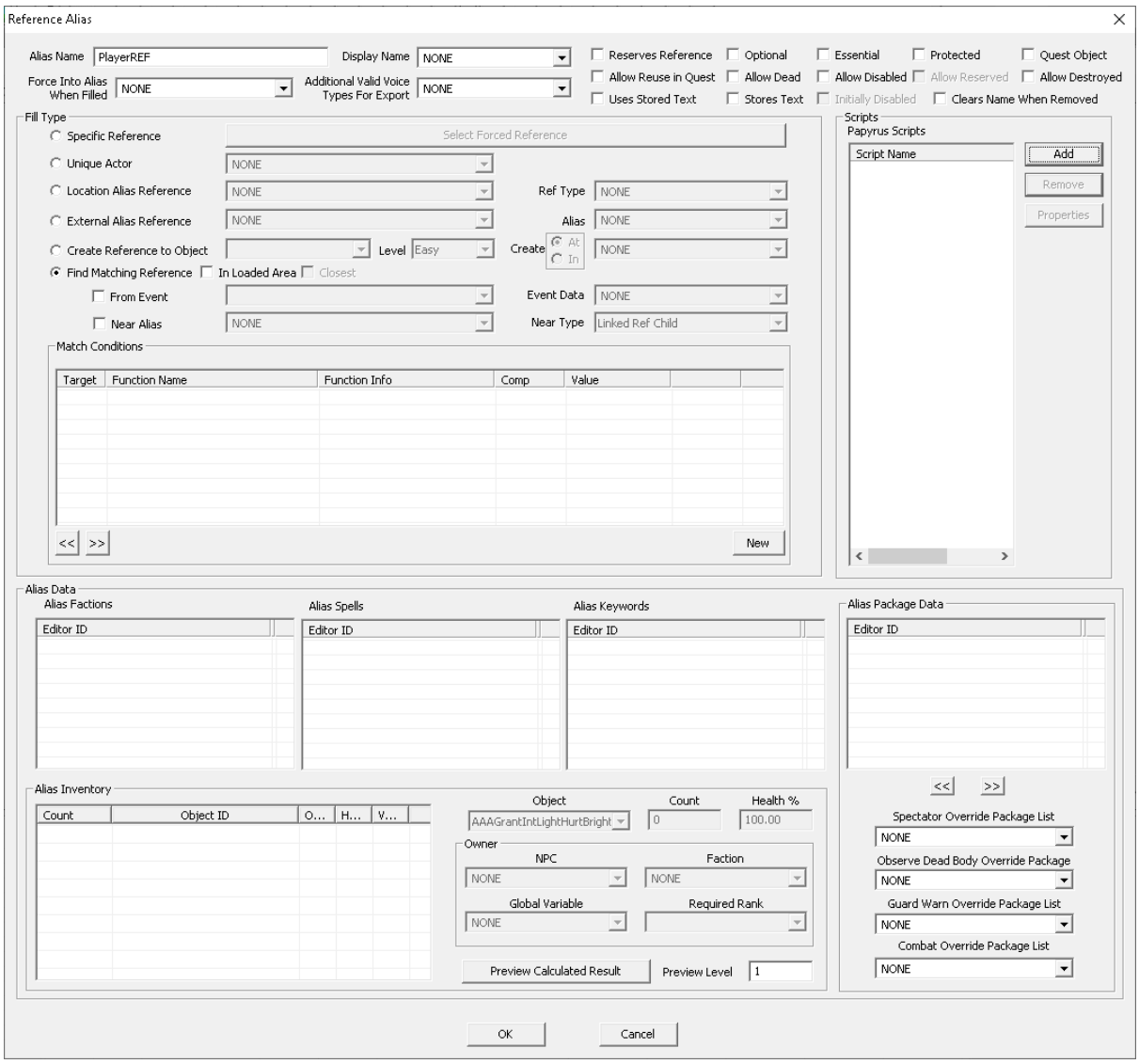


Figure 940 - Setting up the PlayerREF alias.

Filter by 'ski_player', click on SKI_PlayerLoadGameAlias in the script list to highlight it then click OK.

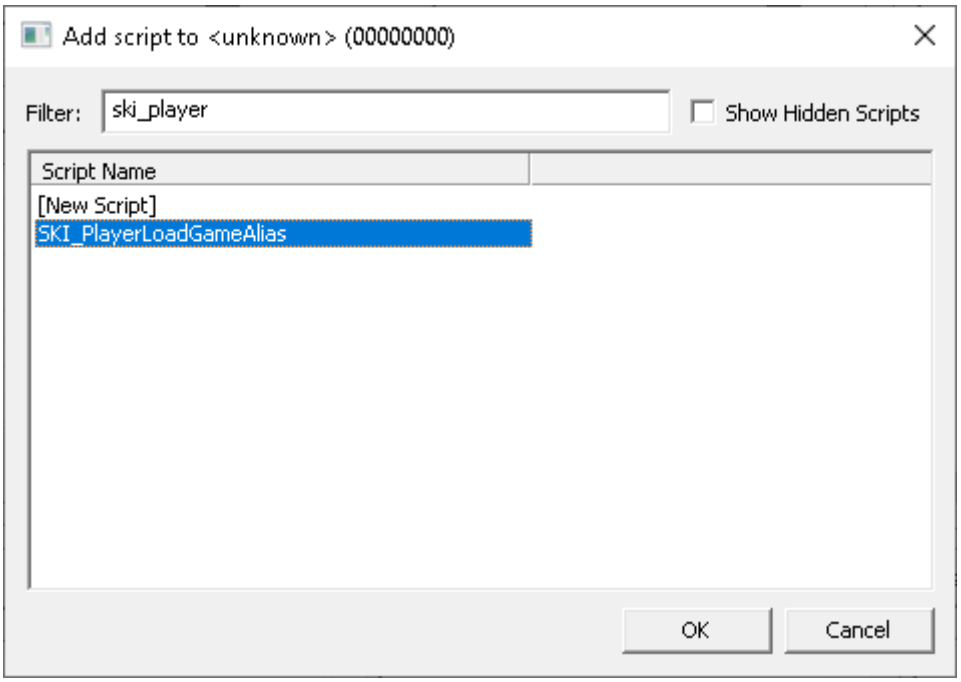


Figure 941 - Adding SKI_PlayerLoadGameAlias to the PlayerREF alias.

Click OK to close out of Reference Alias properties.

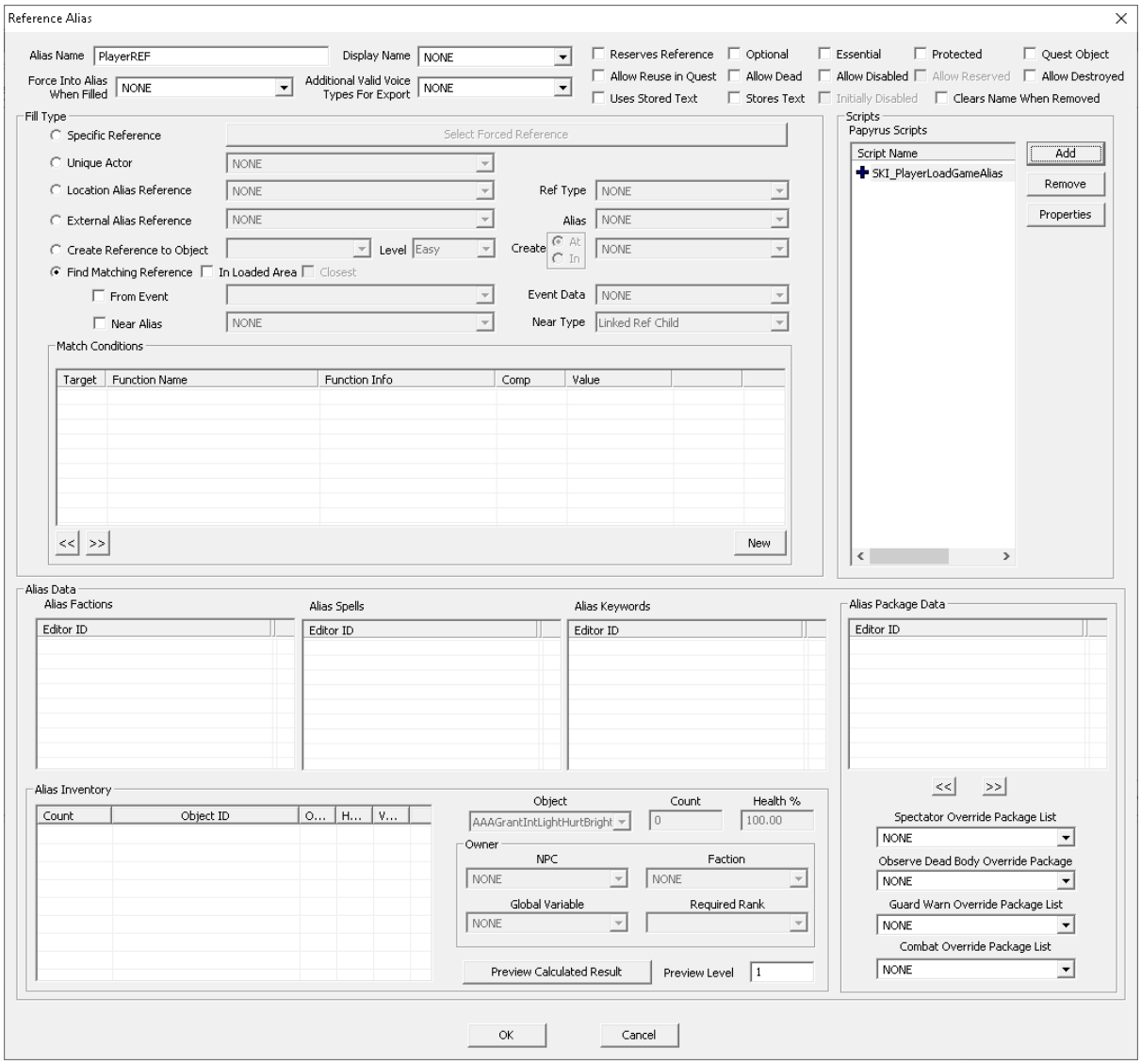


Figure 942 - PlayerREF alias with SKI_PlayerLoadGameAlias script attached.

Go to the Scripts tab and click on the Add button.

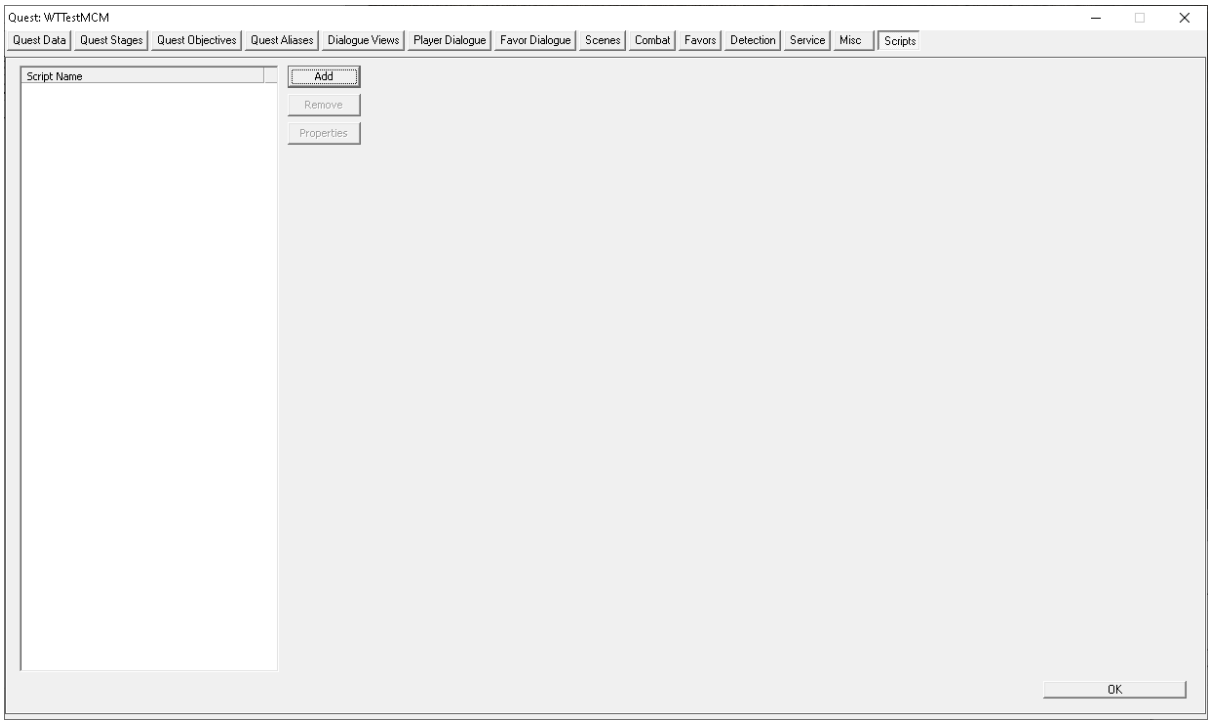


Figure 943 - Adding a new quest script.

Click on [New Script] to highlight it then click OK.

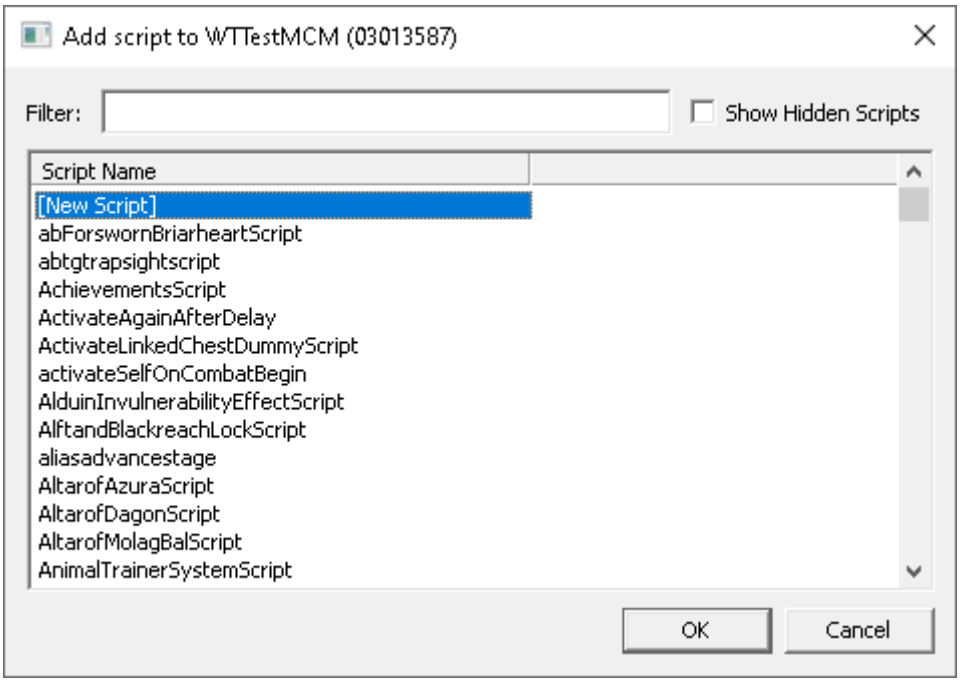


Figure 944 - Creating a new script.

Enter the name of the new script in the Name field.

In the Extends field, enter 'ski_configbase' then click OK.

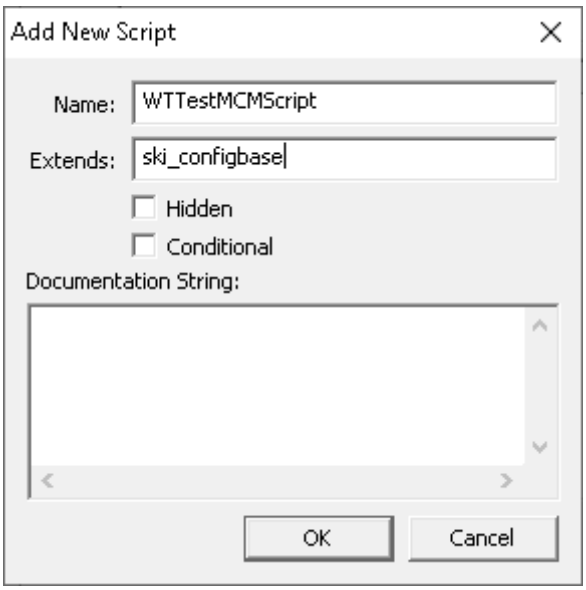


Figure 945 - New script configuration.

Right-click on the new script in the script list and select Edit Source.

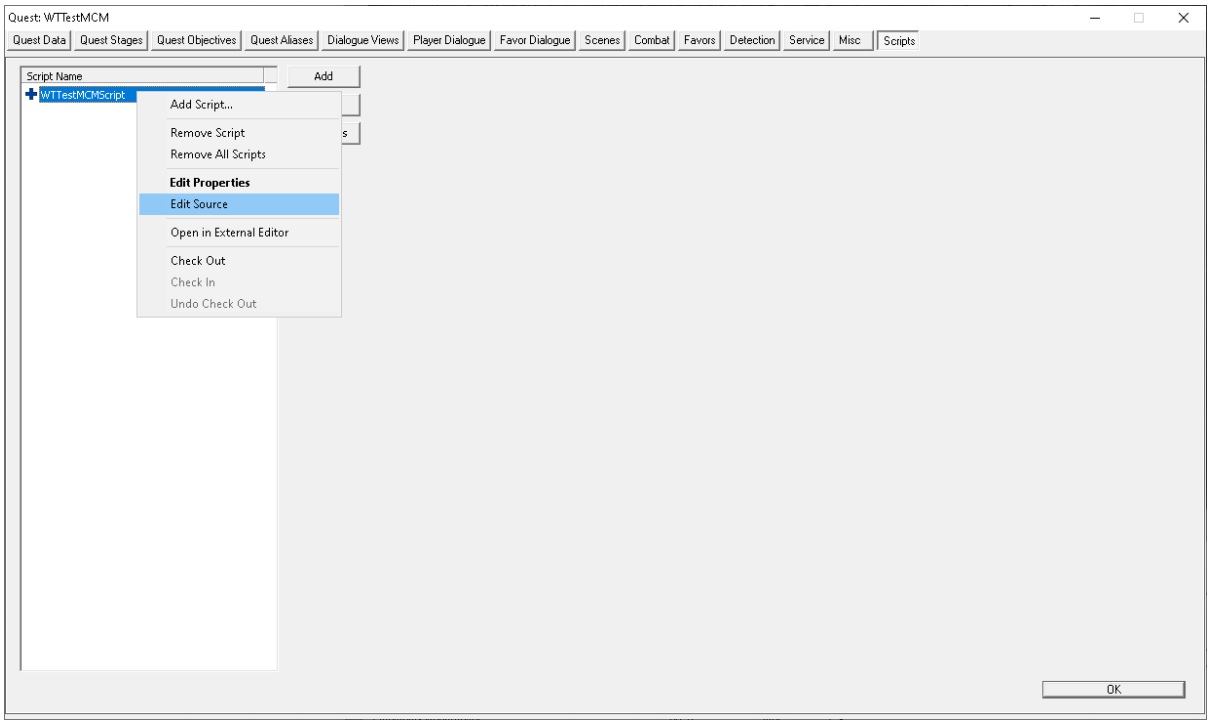


Figure 946 - Editing the script source of our new script.

Right now, our script only consists of the scriptname declaration and the script template we're extending from. ski_configbase is part of SKSE.



Figure 947 - Empty script.

Before we start on our script, we'll need to create some global variables. In the Object Window, go to Miscellaneous > Global.

Right-click on one of the existing global variables and select New.

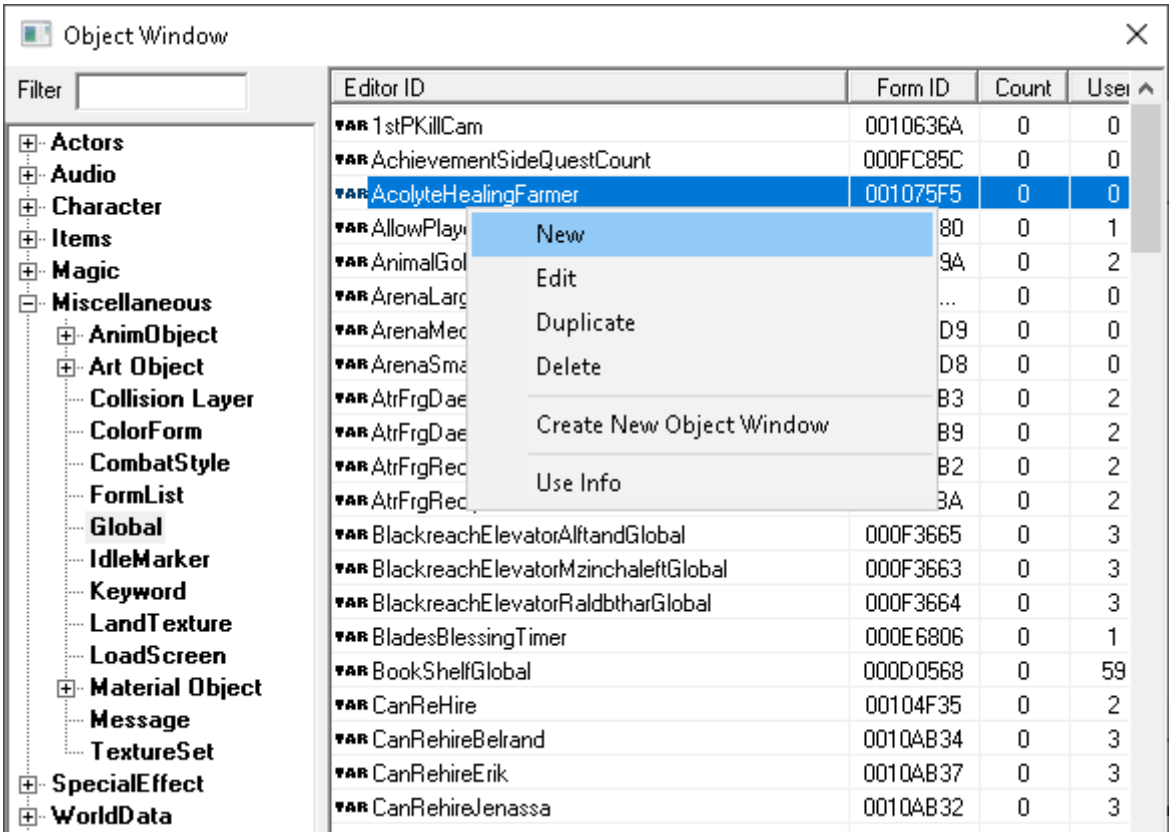


Figure 948 - Creating a new global variable.

Set the ID field. The first one is going to be called WTSampleSlider1.

Leave the Variable Type set to Short and leave the Value field set to 0 then click OK.

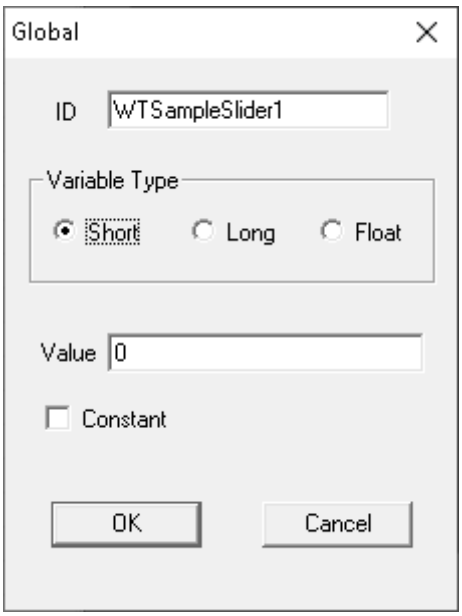


Figure 949 - Global variable setup.

Repeat these steps to create the rest of the global variables we'll need: WTSampleSlider2, WTSampleSlider3, WTSampleTickbox1, WTSampleTickbox2, WTSampleTickbox3, and WTSampleDropDown.

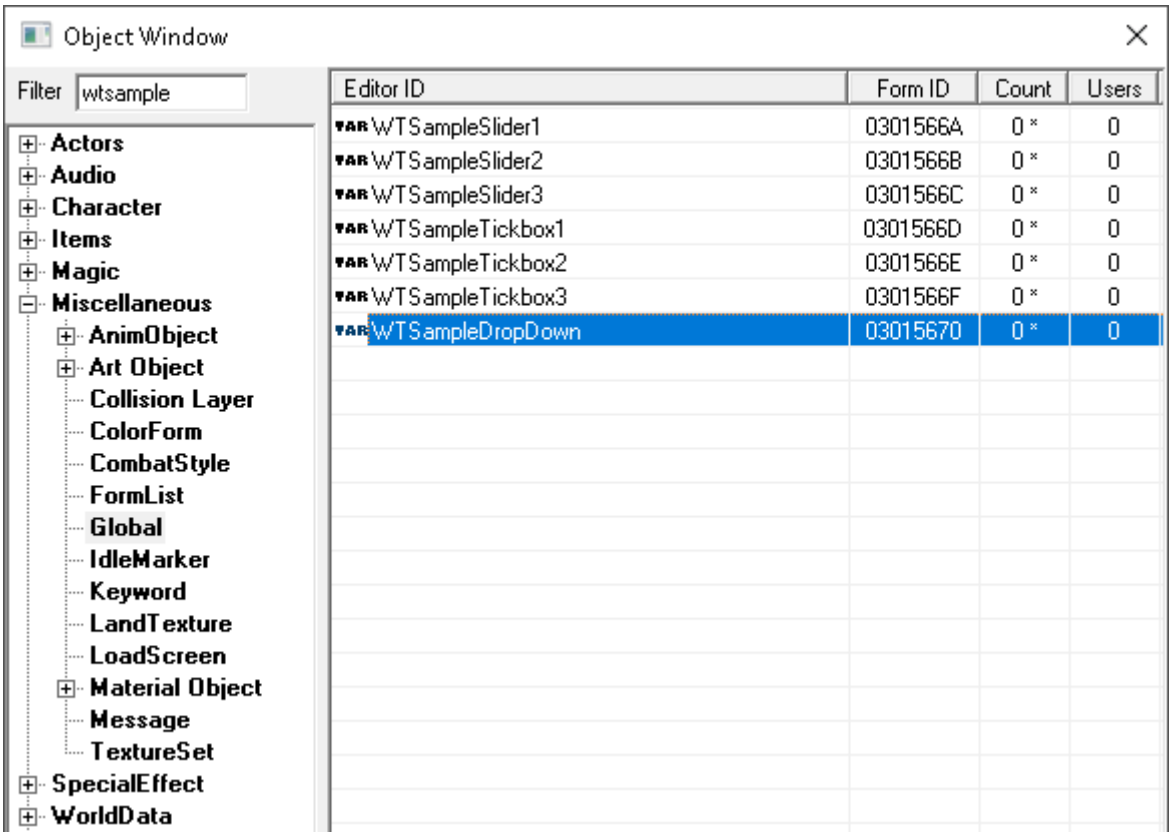


Figure 950 - The global variables created.

Let's start adding those global variables to the script, and the option ID variables used by the interface. These variables should go directly beneath the Scriptname declaration.

```
GlobalVariable Property WTSampleSlider1  Auto
GlobalVariable Property WTSampleSlider2  Auto
GlobalVariable Property WTSampleSlider3  Auto
GlobalVariable Property WTSampleTickbox1  Auto
GlobalVariable Property WTSampleTickbox2  Auto
GlobalVariable Property WTSampleTickbox3  Auto
GlobalVariable Property WTSampleDropDownGlobal  Auto

int OID_WTSampleSlider1
int OID_WTSampleSlider2
int OID_WTSampleSlider3
int OID_WTSampleTickbox1
int OID_WTSampleTickbox2
int OID_WTSampleTickbox3
int OID_WTSampleDropDownMenu
```



Figure 951 - Adding in global variables.

So basically our General submenu will have three tickboxes, three sliders and one drop-down menu.

Press CTRL + S to compile the script.



Figure 952 - Compiling the script.

Back in the Scripts tab, click on WTTestMCMScript to highlight it then click on the Properties button.

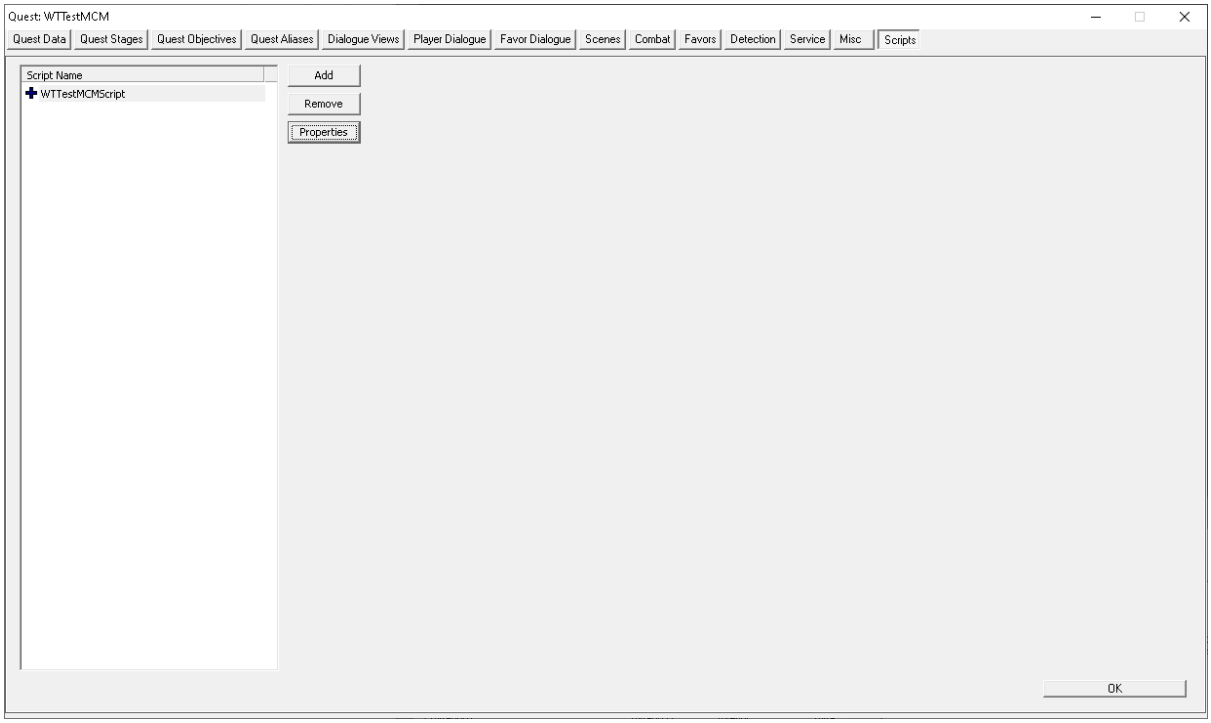


Figure 953 - Linking the script properties to the global variables.

Click on the Auto-Fill All button. This should automatically link our script properties with the global variables we just created.

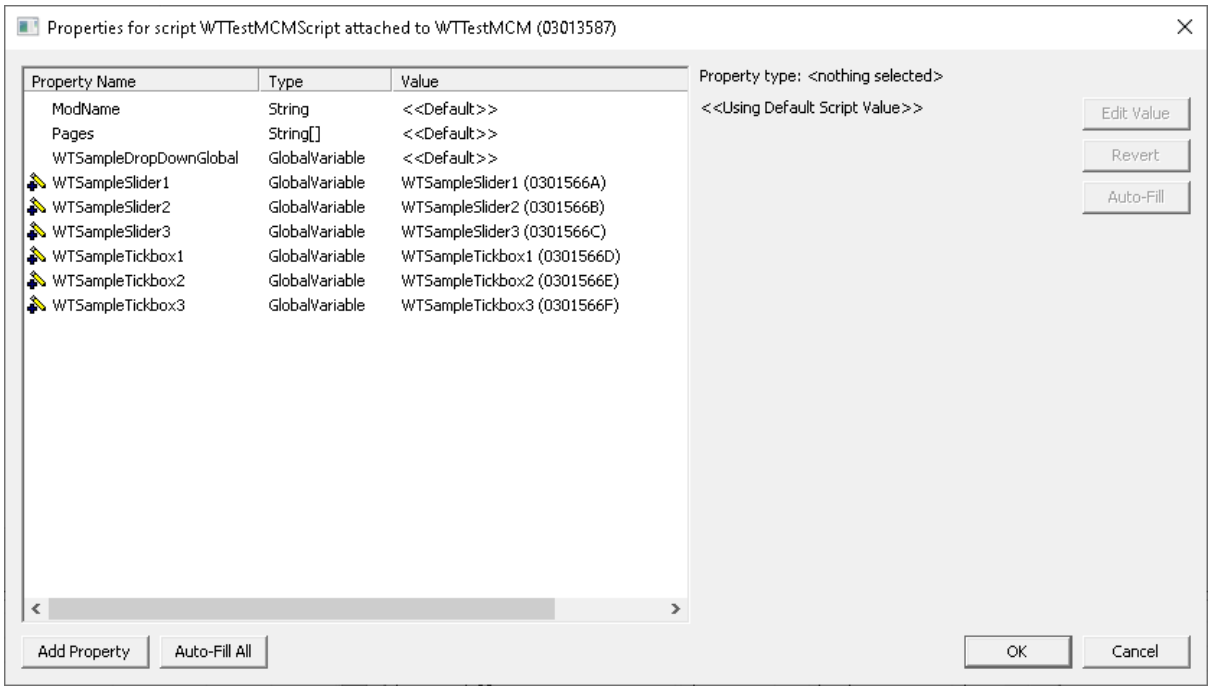


Figure 954 - Auto-Fill All.

The properties that have an identical name to their global variable ID will be linked automatically. The ones that have mismatching names can be linked manually. Set WTSampleDropDownGlobal to WTSampleDropDown.

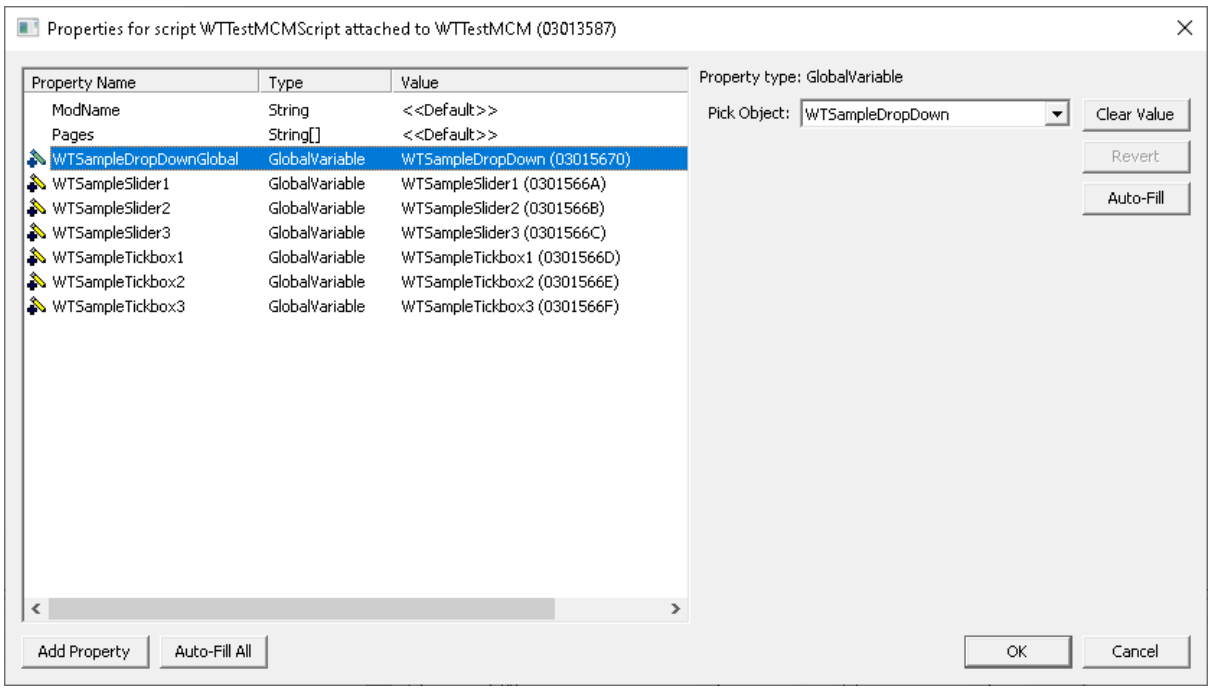


Figure 955 - Setting the global variable for WTSampleDropDownGlobal.

Next, click on ModName in the properties list and click the Edit Value button to the right. Enter in the name of the mod as you want it to appear in the MCM menu.

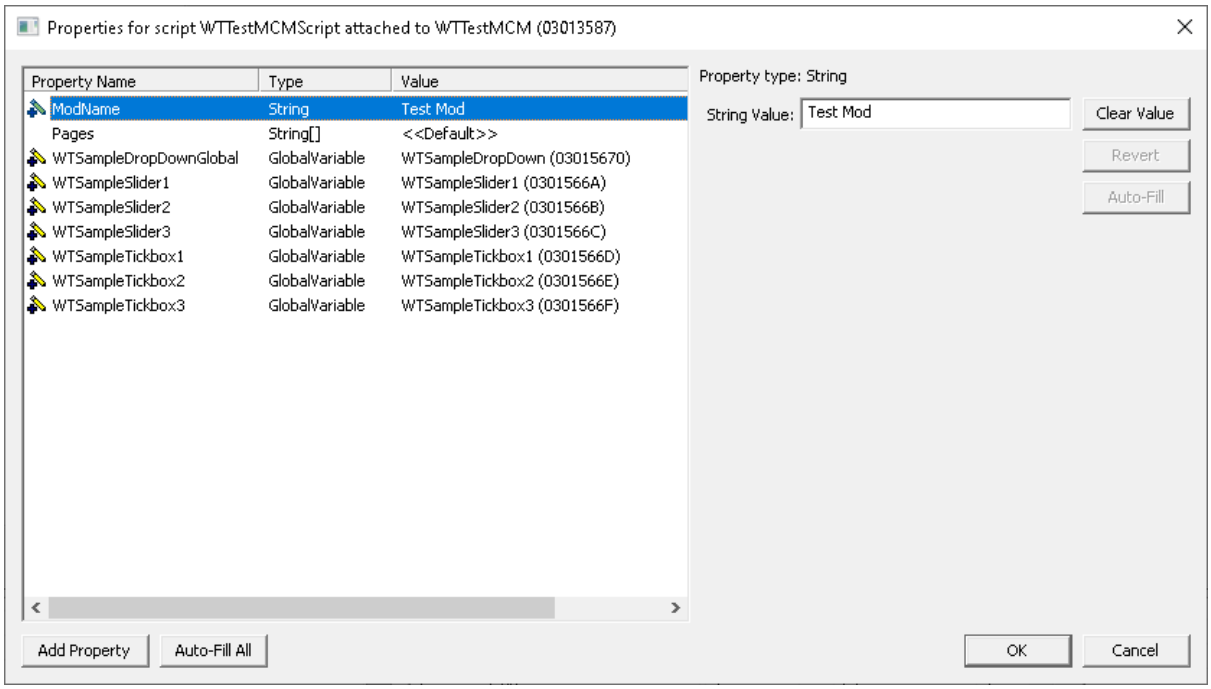


Figure 956 - Setting the mod name for the MCM.

Click on the Pages property and click on the Edit Value button.

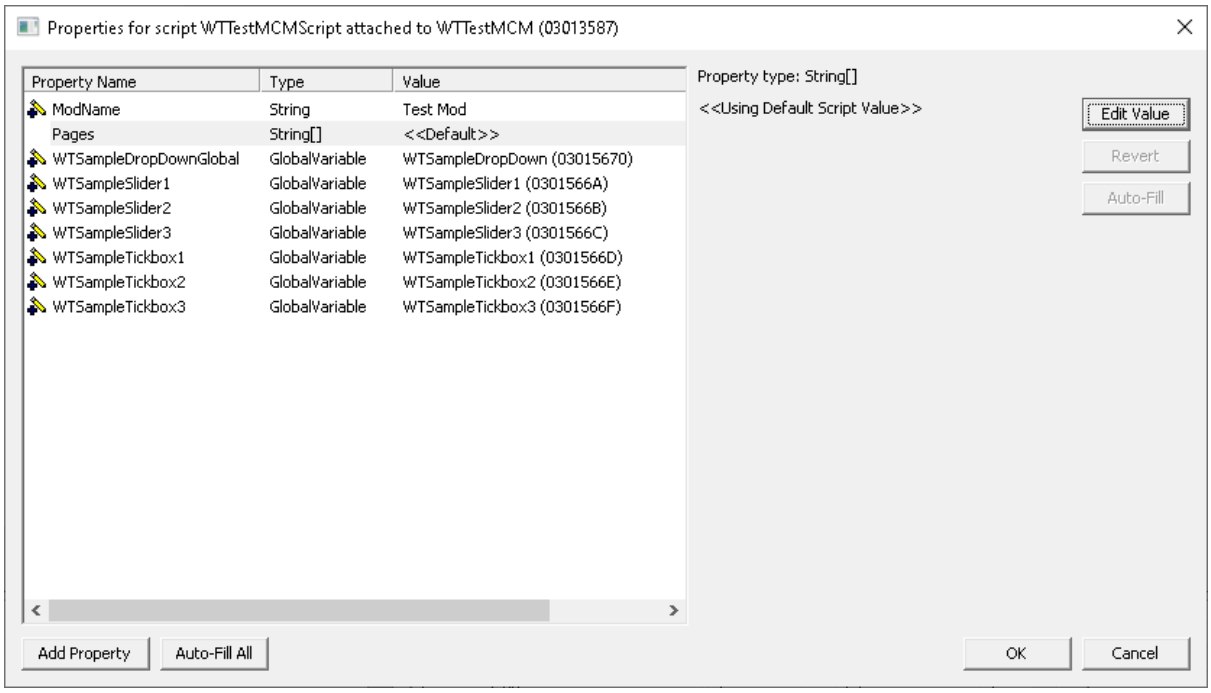


Figure 957 - Adding a submenu to our MCM.

Click on the Add button.

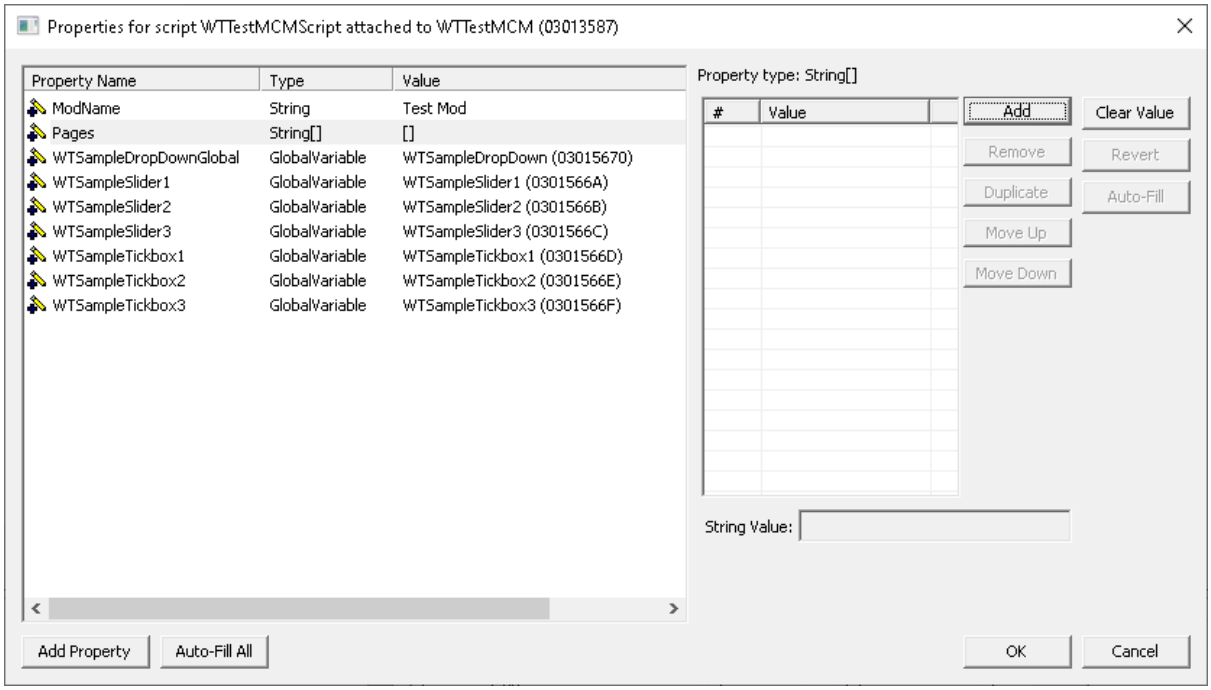


Figure 958 - Adding a new page.

For this example I'm going to add one submenu. Set the string value for the new page to General.

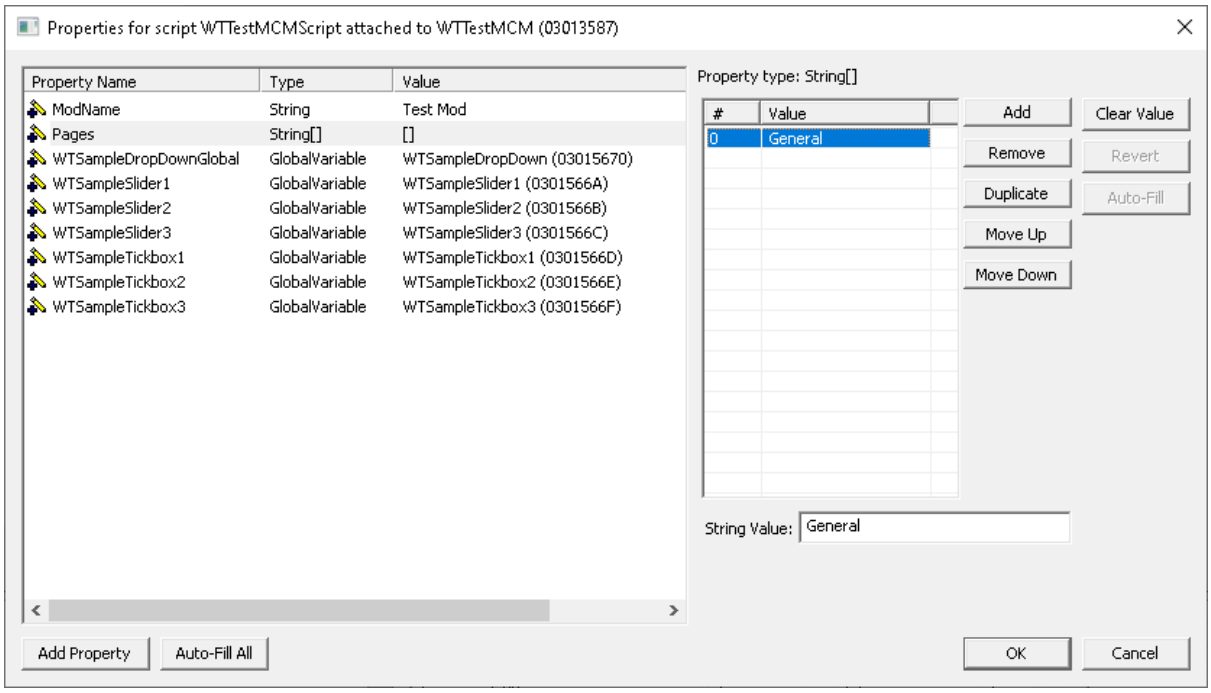


Figure 959 - Adding a General page.

Click OK to close out of script properties. Let's go back to the script source.

The first thing we need to do is initialize the script with the following commands:

```
event OnConfigInit()
parent.OnInit()
endEvent
```



Figure 960 - Initializing the MCM script.

Most MCMs include a splash screen which is displayed when the user clicks on the mod in the MCM menu. The first page loaded by the MCM doesn't have a page name, hence "".

```
Event OnPageReset(string page)
if page == ""
self.LoadCustomContent("testmod/testmodsplash.dds", 0.000000,
0.000000)
return
else
self.UnloadCustomContent()
endIf
EndEvent
```



Figure 961 - Adding a splash screen.

The path defined in the LoadCustomContent command should exist in the Skyrim\Data\Interface or Skyrim Special Edition\Data\Interface folder.



Figure 962 - MCM splash screen for Wyrmstooth.

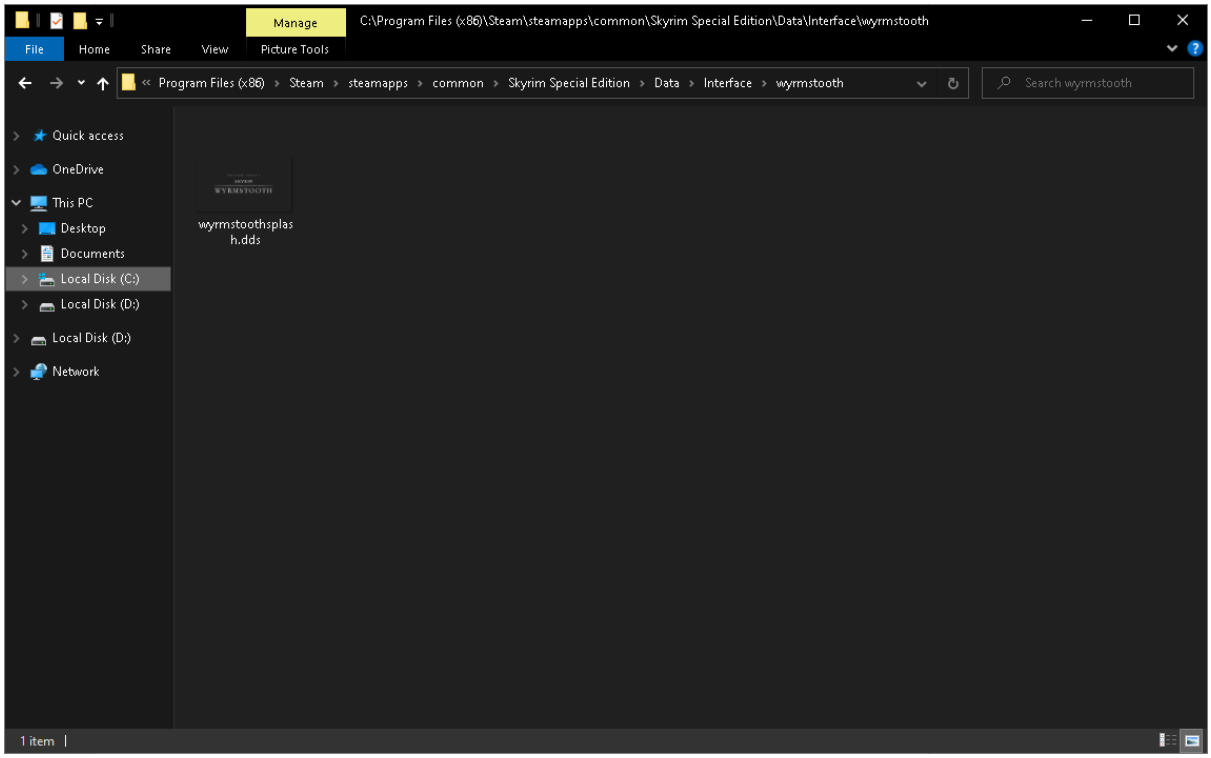


Figure 963 - Full path to splash screen .dds file.

For Wyrmstooth, the path used by LoadCustomContent for its splash screen would be:

```
self.LoadCustomContent("wyrmstooth/wyrmstoothsplash.dds", 0.000000, 0.000000)
```

The numbers after the path to the .dds file specify an X/Y offset for the image. We can leave them both set to 0.000000 for now.

Next, let's add a General tab with some sample sliders, tick-boxes and a drop-down selection.

```
if page == "General"
SetCursorFillMode(TOP_TO_BOTTOM)

AddHeaderOption("Section 1")
OID_WTSampleSlider1 = AddSliderOption("Sample Slider 1",
WTSampleSlider1.GetValue(),"{0}")
OID_WTSampleSlider2 = AddSliderOption("Sample Slider 2",
WTSampleSlider2.GetValue(),"{0}")
OID_WTSampleSlider3 = AddSliderOption("Sample Slider 3",
WTSampleSlider3.GetValue(),"{0}")
AddEmptyOption()

AddHeaderOption("Section 2")
OID_WTSampleTickbox1 = AddToggleOption("Sample Tickbox 1",
WTSampleTickbox1.GetValue())
OID_WTSampleTickbox2 = AddToggleOption("Sample Tickbox 2",
WTSampleTickbox2.GetValue())
OID_WTSampleTickbox3 = AddToggleOption("Sample Tickbox 3",
WTSampleTickbox3.GetValue())

SetCursorPosition(1)
AddHeaderOption("Section 3")
OID_WTSampleDropDownMenu = AddMenuOption("",
WTSampleDropDownList[WTSampleDropDownSelection])
Endif
```

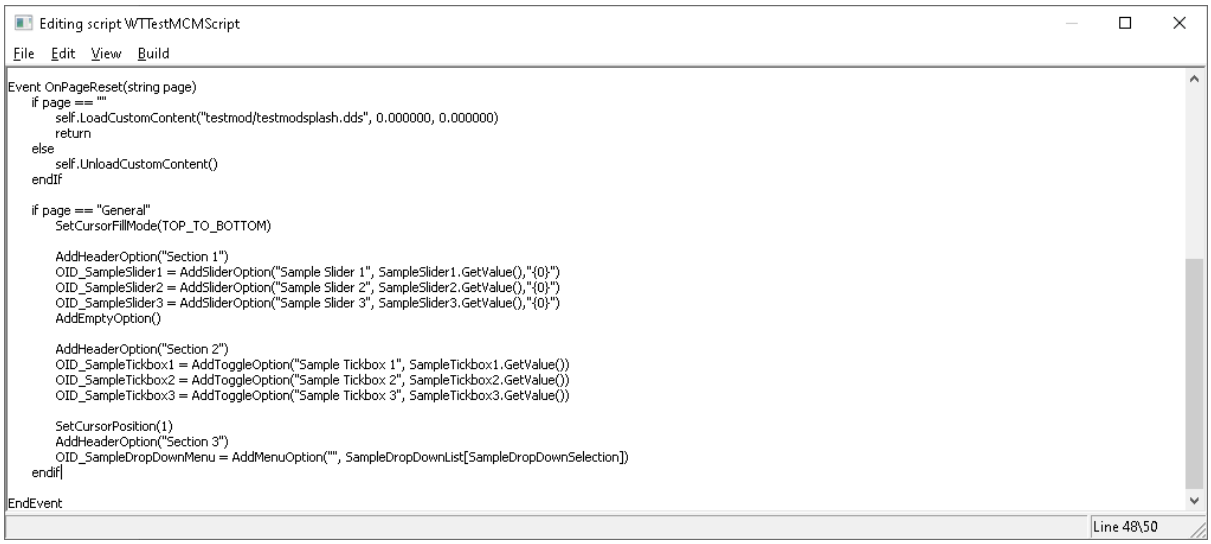


Figure 964 - General tab scripting added.

If you added more submenus in the script properties, you can add UI elements to them by adding elseif page == "My Second Submenu", elseif page == "My Third Submenu" before the Endif.

If the player clicks on the General tab, we want these interface elements to appear instead of the splash screen. We can separate different pages of UI elements using an if/elseif/else statement.

SetCursorFillMode(TOP_TO_BOTTOM) means the interface elements we're adding to the UI will be added vertically starting from the top-left. The alternative arrangement is LEFT_TO_RIGHT which would mean the interface elements would be added to the UI horizontally starting from the top-left. I set it to TOP_TO_BOTTOM so we can define exactly at which element we start the second column.

AddHeaderOption("Section 1") adds a header to the UI. We can use headers to group similar interface elements together.

Right now if we try and compile we'll get an error because we're missing a couple variables needed for the drop-down menu. So let's add those in next.

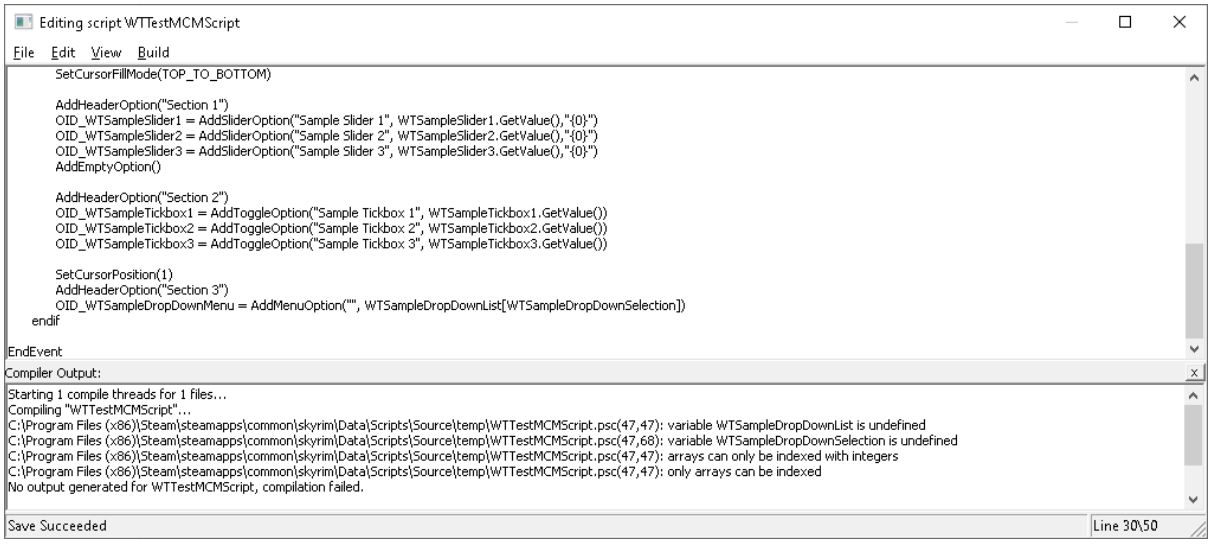


Figure 965 - Compilation error.

Enter the following variable declarations beneath the int OID variables:

```
string[] WTSampleDropDownList
int WTSampleDropDownSelection
```

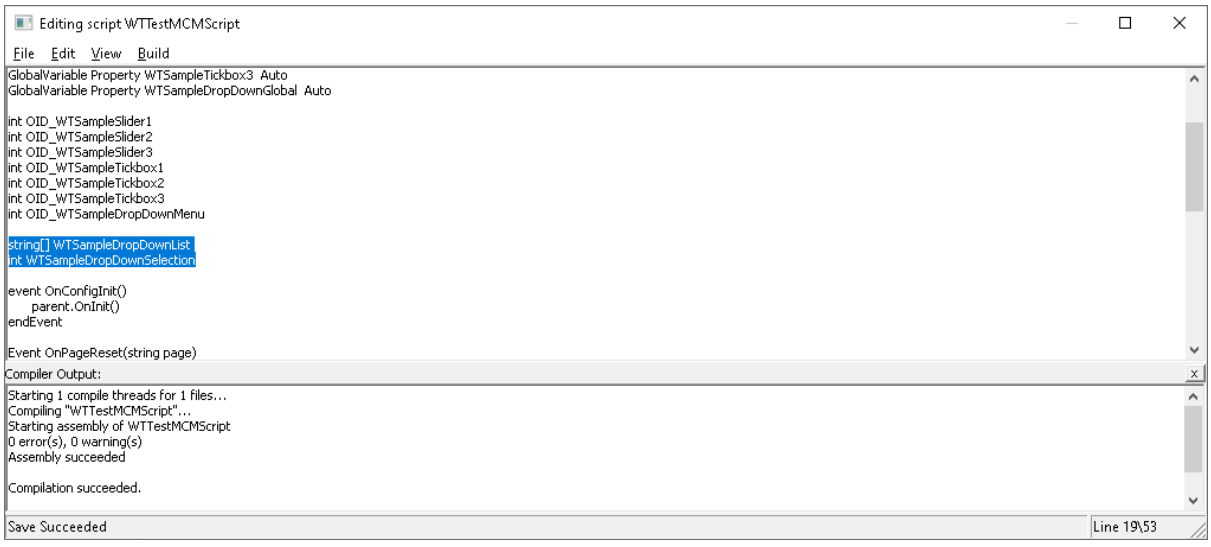


Figure 966 - Adding the string and int variables for the drop-down list.

If you compile the script it should now compile successfully.

Next, let's add in an array with some selectable options for the drop-down menu. These options will appear when the player clicks on the drop-down menu in the MCM.

In the OnConfigInit() event, add the following lines beneath parent.OnInit():

```
WTSampleDropDownList = new string[15]
WTSampleDropDownList[0] = "Option 1"
WTSampleDropDownList[1] = "Option 2"
WTSampleDropDownList[2] = "Option 3"
WTSampleDropDownList[3] = "Option 4"
WTSampleDropDownList[4] = "Option 5"
```



Figure 967 - Adding drop-down list options.

To show the options in a drop-down list when the player clicks on it, we'll need to add an OnOptionMenuOpen() event.

```
Event OnOptionMenuOpen(int option)
if (option == OID_WTSampleDropDownMenu)
SetMenuDialogOptions(WTSampleDropDownList)
SetMenuDialogStartIndex(WTSampleDropDownSelection)
SetMenuDialogDefaultIndex(0)
endIf
EndEvent
```

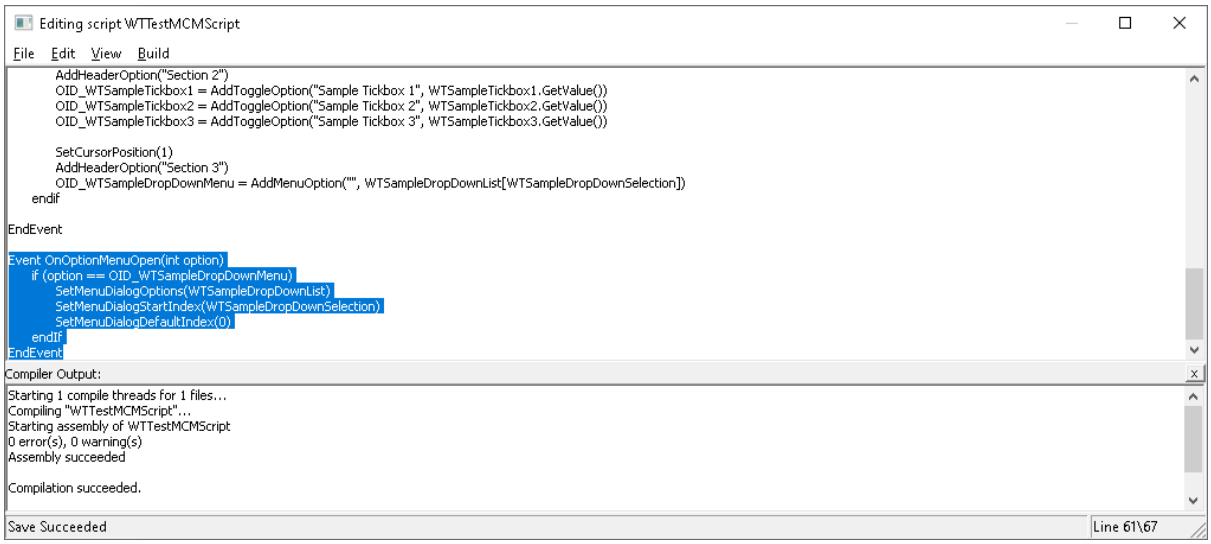


Figure 968 - Event to handle opening a drop-down menu.

The value defined in SetMenuDialogueDefaultIndex() specifies the default value for this drop-down menu. For this example I'm just going to leave it set to '0'.

We need to add an OnOptionMenuAccept() event to handle the player making a selection from a drop-down menu.

```
Event OnOptionMenuAccept(int option, int index)
if (option == OID_WTSampleDropDownMenu)
WTSampleDropDownSelection = index
SetMenuOptionValue(OID_WTSampleDropDownMenu,
WTSampleDropDownList[WTSampleDropDownSelection])
WTSampleDropDownGlobal.SetValue(WTSampleDropDownSelection)
endIf
EndEvent
```

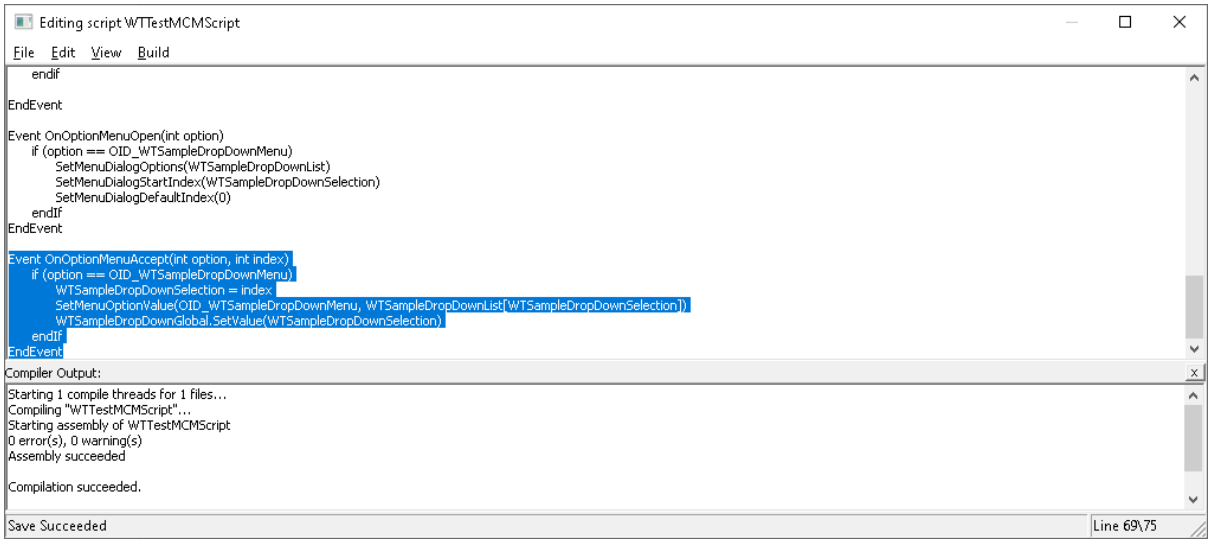


Figure 969 - Event to handle making a selection from a drop-down menu.

The option we choose here will change the value of the WTSampleDropDownGlobal global variable which we can reference in other scripts.

Like drop-down menus, we need to set up two events sliders. The OnOptionSliderOpen() event handles the opening of the slider menu when the player clicks on it in the MCM.

```
Event OnOptionSliderOpen(Int Option)
if(Option == OID_WTSampleSlider1)
SetSliderDialogStartValue(WTSampleSlider1.GetValue())
SetSliderDialogDefaultValue(0)
SetSliderDialogRange(0,99)
SetSliderDialogInterval(1)
elseif(Option == OID_WTSampleSlider2)
SetSliderDialogStartValue(WTSampleSlider2.GetValue())
SetSliderDialogDefaultValue(0)
SetSliderDialogRange(0,99)
SetSliderDialogInterval(1)
elseif(Option == OID_WTSampleSlider3)
SetSliderDialogStartValue(WTSampleSlider3.GetValue())
SetSliderDialogDefaultValue(0)
SetSliderDialogRange(0,99)
SetSliderDialogInterval(1)
endif
EndEvent
```



Figure 970 - Event to handle opening a slider.

SetSliderDialogueStartValue() sets the initial value of the slider. In the example above that value is read from the global variable. SetSliderDialogDefaultValue() sets the default value of the slider. SetSliderDialogRange() sets the minimum and maximum values for the slider. SetSliderDialogInterval() sets the amount by which the player can increase or decrease the slider value. In the example above I set this to '1' so the player can increase or decrease the slider value by a full integer.

To save the slider value set by the player we need to use the OnOptionSliderAccept() event.

```
Event OnOptionSliderAccept(Int Option, Float Value)
if(Option == OID_WTSampleSlider1)
WTSampleSlider1.setValue(Value)
SetSliderOptionValue(Option,Value,"{0}")
elseif(Option == OID_WTSampleSlider2)
WTSampleSlider2.setValue(Value)
SetSliderOptionValue(Option,Value,"{0}")
elseif(Option == OID_WTSampleSlider3)
WTSampleSlider3.setValue(Value)
SetSliderOptionValue(Option,Value,"{0}")
endif
EndEvent
```



Figure 971 - Event to handle setting a new value based on the slider selection.

We only need the OnOptionSelect() event to handle tickboxes.

```
event OnOptionSelect(int Option)
  if (Option == OID_WTSampleTickbox1)
    if WTSampleTickbox1.getValue() as int == 0
      WTSampleTickbox1.setValue(1)
      SetToggleOptionValue(Option, 1)
    else
      WTSampleTickbox1.setValue(0)
      SetToggleOptionValue(Option, 0)
    endif

  elseif (Option == OID_WTSampleTickbox2)
    if WTSampleTickbox2.getValue() as int == 0
      WTSampleTickbox2.setValue(1)
      SetToggleOptionValue(Option, 1)
    else
      WTSampleTickbox2.setValue(0)
      SetToggleOptionValue(Option, 0)
    endif

  elseif (Option == OID_WTSampleTickbox3)
    if WTSampleTickbox3.getValue() as int == 0
      WTSampleTickbox3.setValue(1)
      SetToggleOptionValue(Option, 1)
    else
      WTSampleTickbox3.setValue(0)
      SetToggleOptionValue(Option, 0)
    endif
  endif
EndEvent
```



Figure 972 - Event to handle ticking or unticking a tickbox.

Basically, if the player ticks a tickbox and the tickbox isn't ticked, we flag it as ticked and set its global variable value to 1. Otherwise, untick it and set its global variable value to 0. Use tickboxes for setting Boolean values. i.e.: when an option either needs to be on or off/yes or no.

Lastly, we need to add an OnoptionHighlight() event to handle text tips that appear at the bottom of the MCM screen when we hover over UI elements.

```
Event OnOptionHighlight(Int Option)
If (Option == OID_WTSampleSlider1)
SetInfoText("Sample slider 1 highlighted.")
ElseIf (Option == OID_WTSampleSlider2)
SetInfoText("Sample slider 2 highlighted.")
ElseIf (Option == OID_WTSampleSlider3)
SetInfoText("Sample slider 3 highlighted.")
ElseIf (Option == OID_WTSampleTickbox1)
SetInfoText("Sample tickbox 1 highlighted.")
ElseIf (Option == OID_WTSampleTickbox2)
SetInfoText("Sample tickbox 2 highlighted.")
ElseIf (Option == OID_WTSampleTickbox3)
SetInfoText("Sample tickbox 3 highlighted.")
ElseIf (Option == OID_WTSampleDropDownMenu)
SetInfoText("Sample drop down menu highlighted.")
EndIf
EndEvent
```

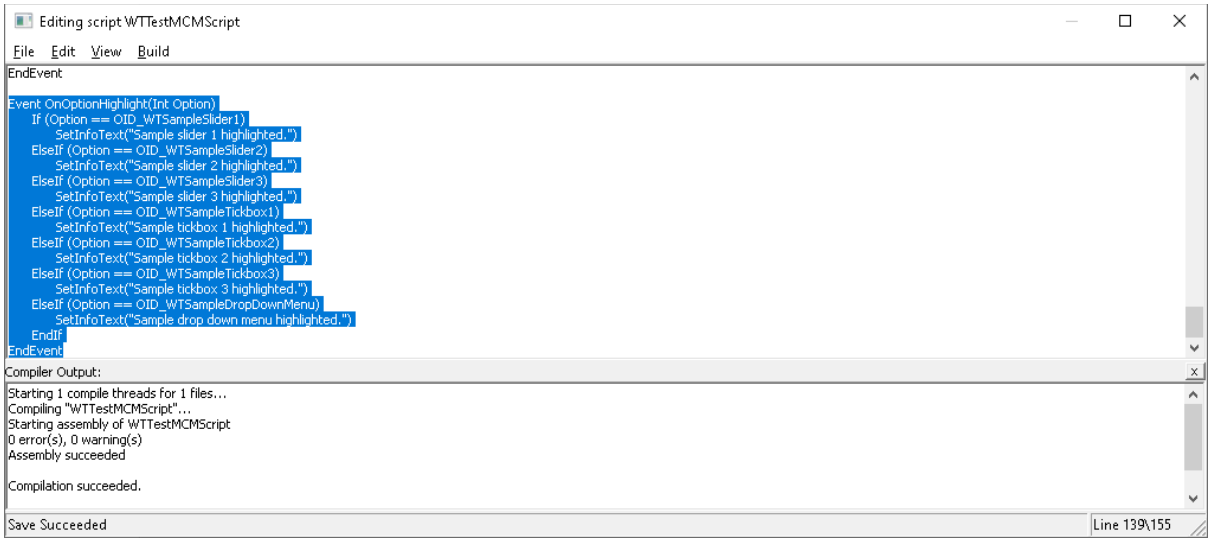


Figure 973 - Event to handle text tips.

The text to display at the bottom of the screen is defined by SetInfoText().

If we go in-game, our new MCM should register itself after a few seconds.



Figure 974 - Test Mod MCM.

Take a few moments to confirm all the options are present and the text tips for each UI element are correct.

Change a few settings then close and re-open the MCM to confirm that the settings were successfully saved.



Figure 975 - Options saved after closing the menu.

For more information I'd recommend Schlangster's MCM Quickstart guide on Github.

CHAPTER 7: BUILDING LOD AND GENERATING FILES

BUILDING LAND LOD USING OSCAPE

Land LOD is a simplified representation of the world space landscape shown in unloaded cells.

To generate land LOD I'm going to use a utility called Oscape which you can download from Nexusmods.

Important: Before we can generate object and tree LOD in xEdit, we need to generate land LOD first to create a .lod file for our world space. Without a .lod file, the world space won't be listed in xEdit.

Launch Oscape x32/Oscape x64.

Go to Game and set it to Skyrim.

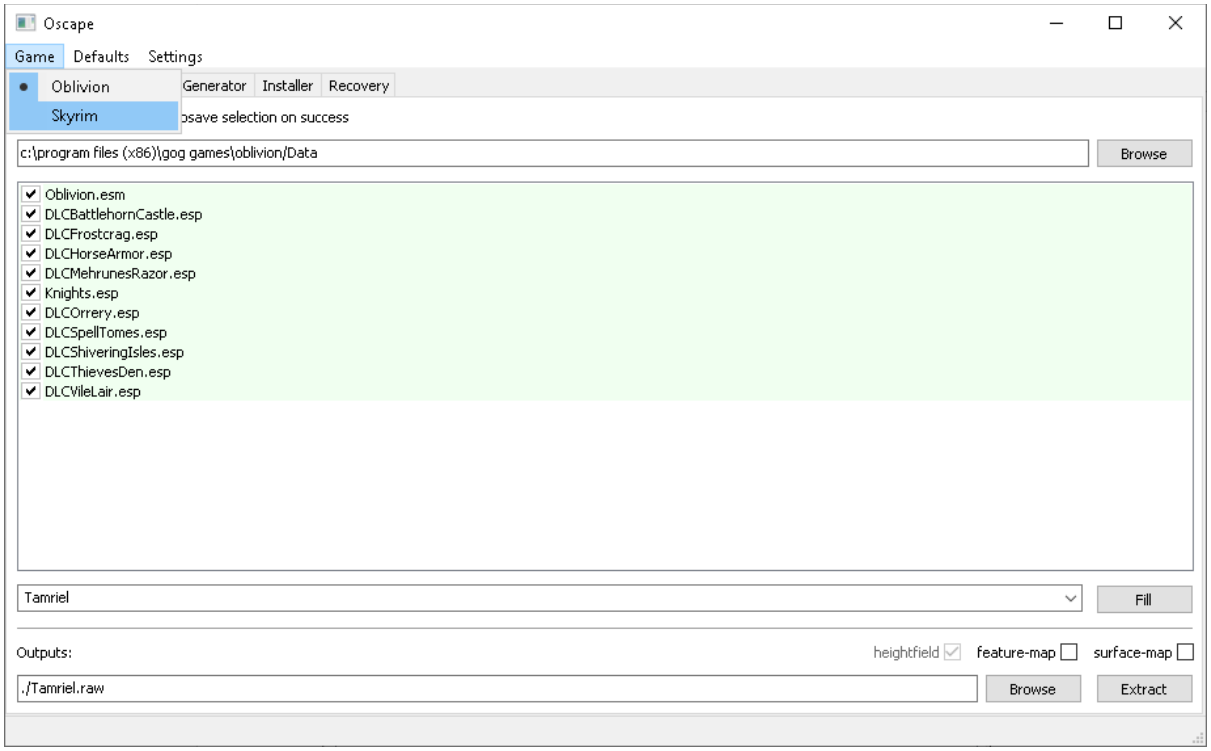


Figure 976 - Set Game to Skyrim.

Tick your mod, tick any dependencies such as Skyrim.esm, then click on the Fill button.

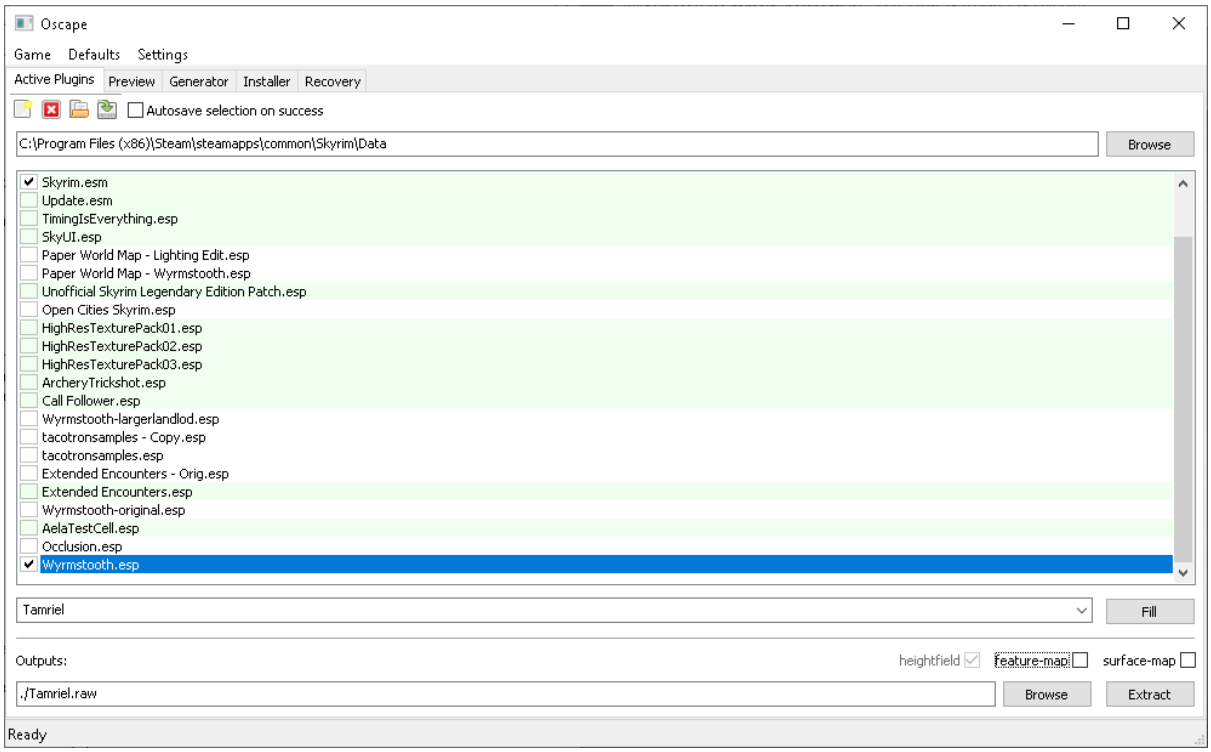


Figure 977 - Selecting our mod.

Select your world space in the drop-down menu next to the Fill button, and change the name of the .raw file in the Outputs text field.

Tick Feature-Map and Surface-Map then click Extract.

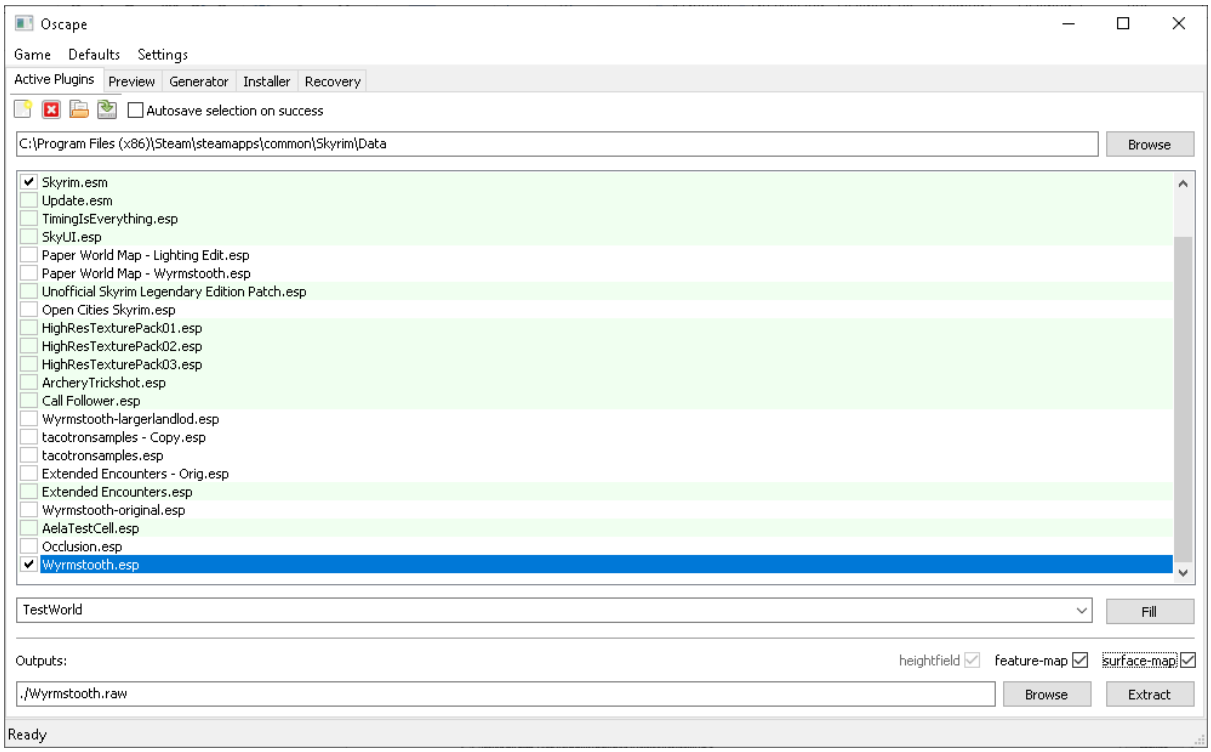


Figure 978 - Selecting our world space.

In the Preview tab, click on the second Browse button and choose a directory to save the generated files to. In my example I just set this to 'C:\Users\Jonx0r\Desktop\Generated Files' for now. You can always move this folder to a better location later.

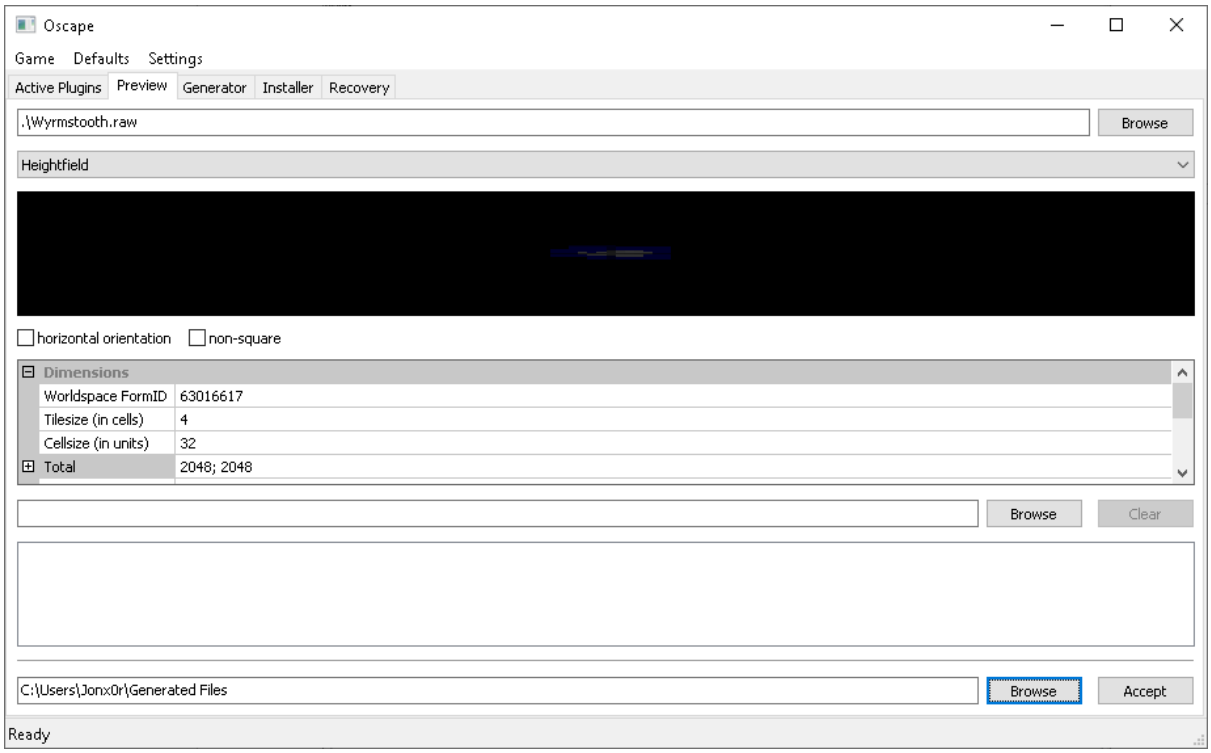


Figure 979 - Preview tab.

Click Accept.

Alright, now to configure our LOD generation settings.

So you should see three tick boxes in the Generator tab; Meshes, Normals and Colors. Meshes and Normals should already be ticked. Go ahead and tick Colors as well.

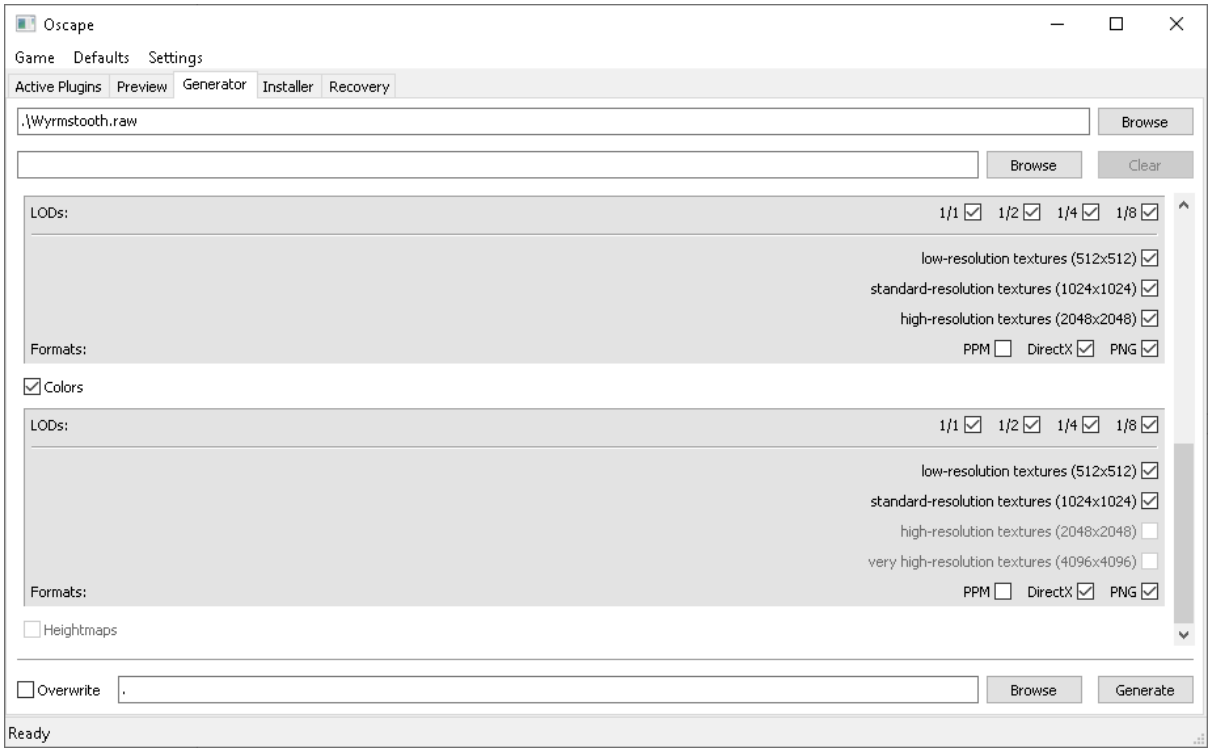


Figure 980 - Colors ticked.

Next, I'm going to set the 'Target of max. resolution' drop-down to '256000 (1/1 of 1000 per tile)'.

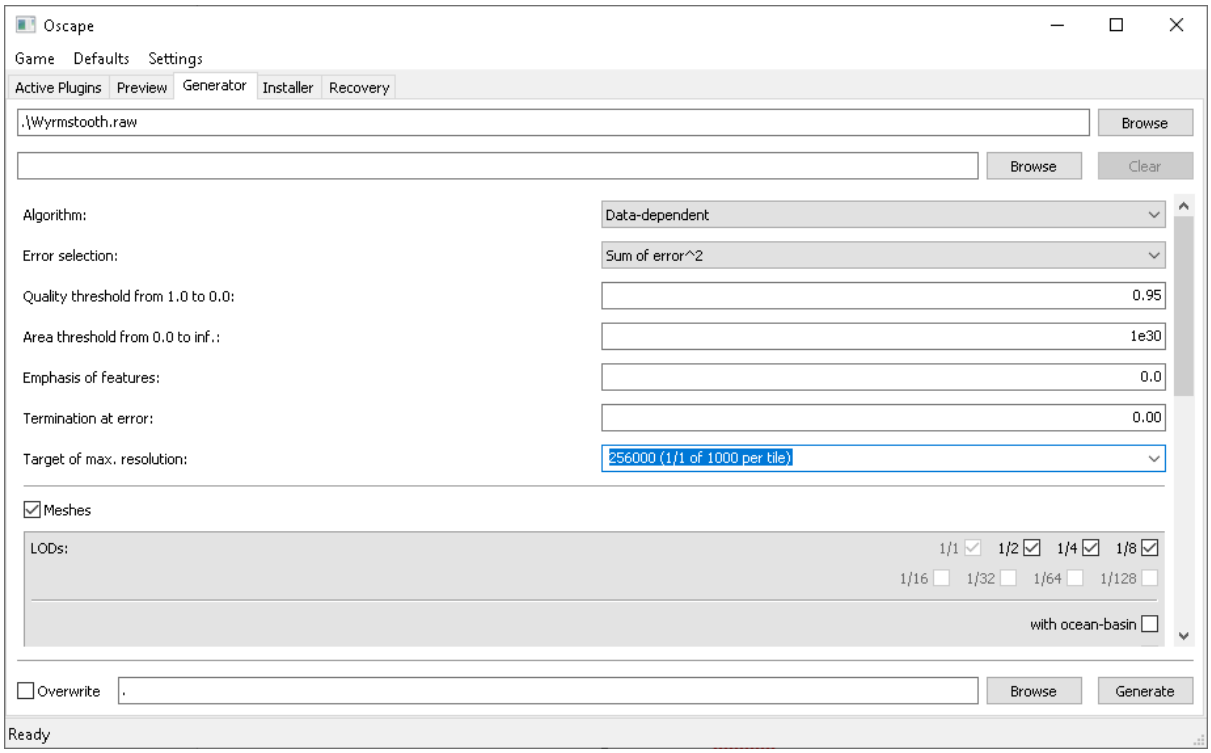


Figure 981 - Target of max. resolution set to 256000.

Important: When your land LOD is building, if Oscape crashes, you'll need to lower the 'Target of max. resolution' value.

Under Meshes, tick 'with ocean-basin'. Make sure only the NIF format is ticked.

Under Normals, untick PNG.

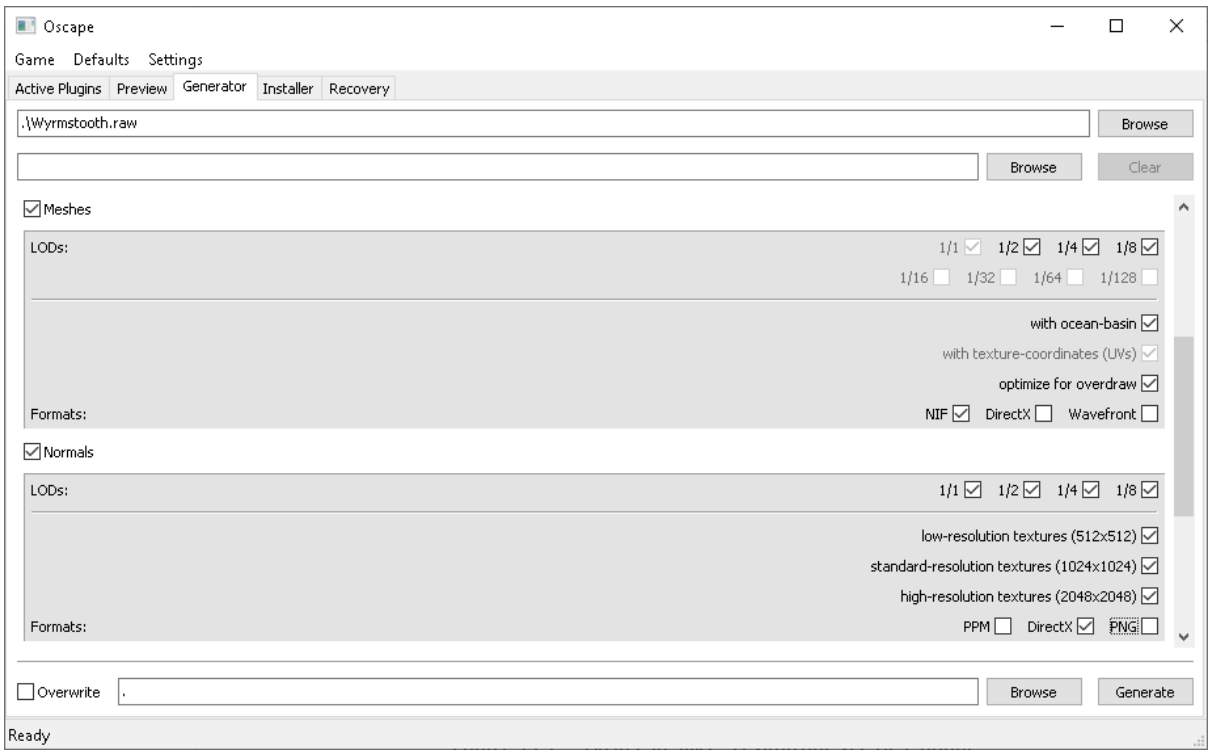


Figure 982 - Configuring Meshes and Normals.

Under Colors, untick PNG.

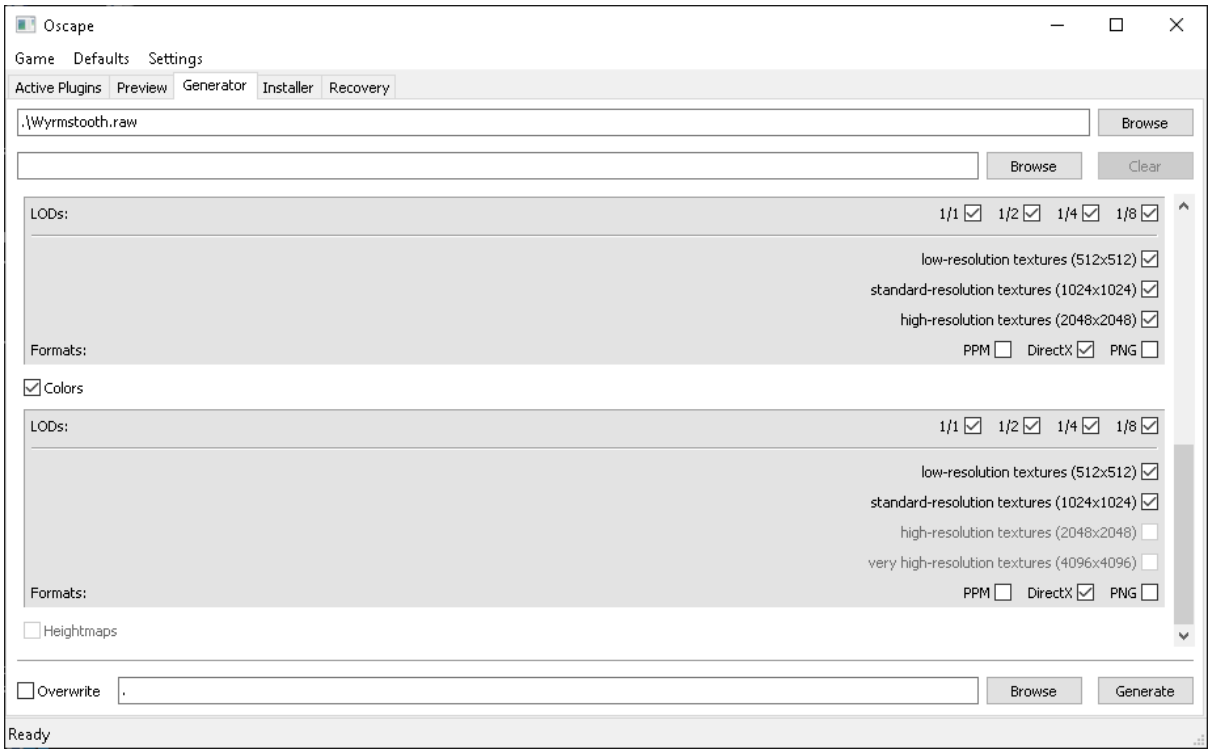


Figure 983 - Configuring Colors.

Click Browse and choose a folder to generate the LOD files in. Again, I just set this to 'C:\Users\Jonx0r\Desktop\Generated Files'

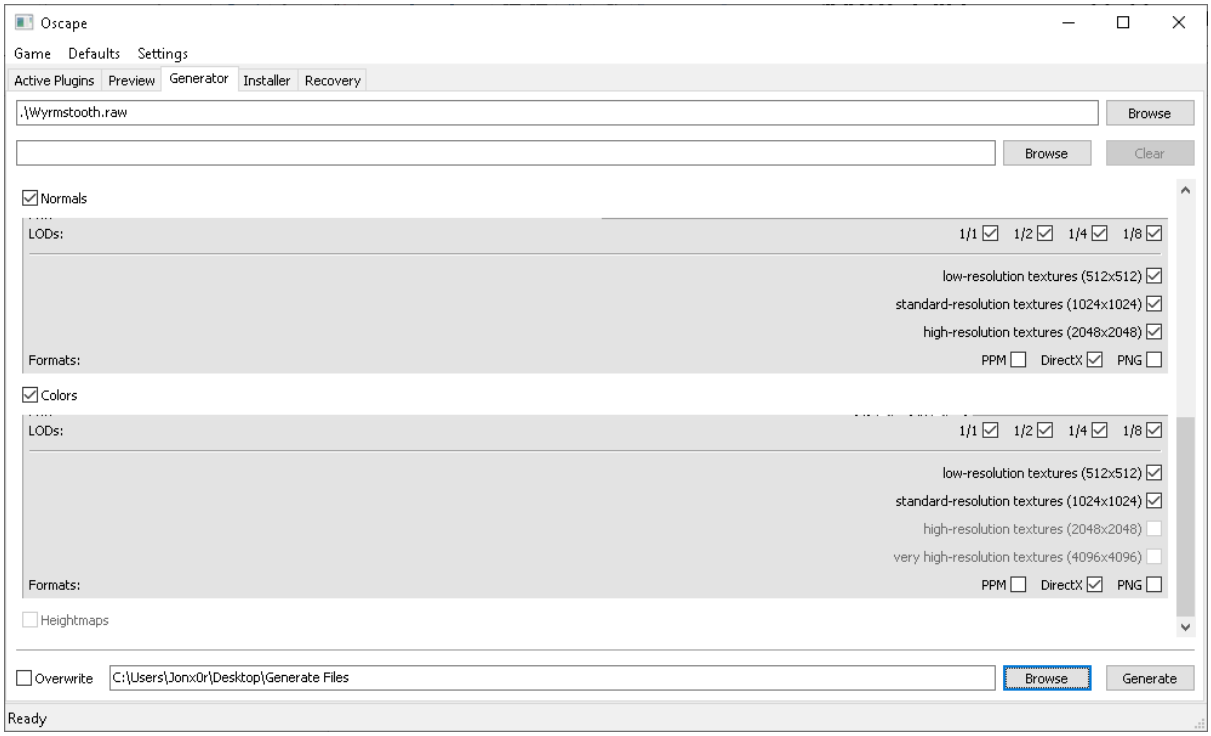


Figure 984 - Choosing a folder to generate the LOD files in.

If you've done this before and that folder already exists, tick Overwrite.

Click Generate to start creating your land LOD files.

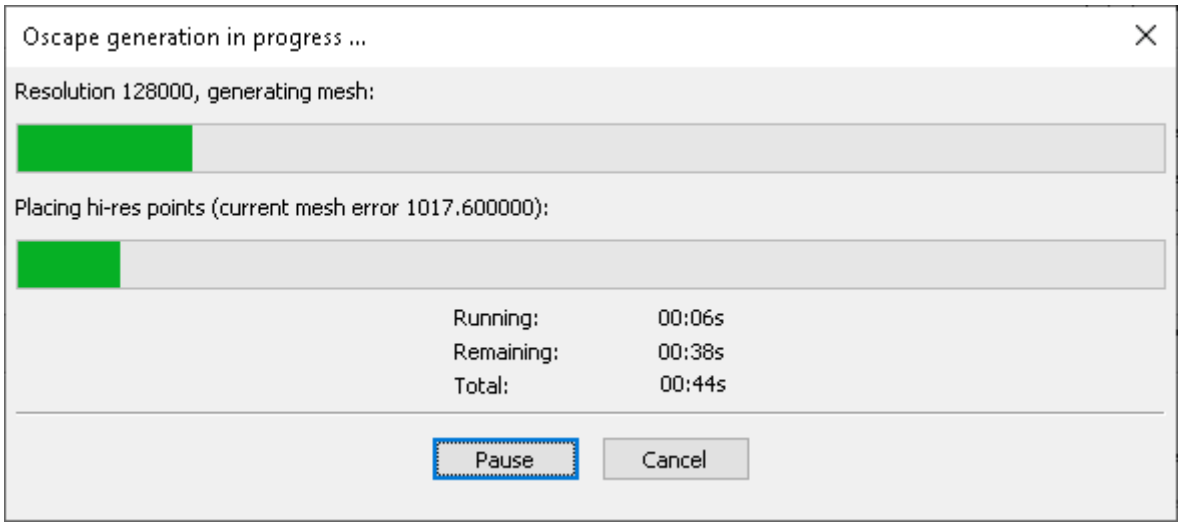


Figure 985 - Oscape generation in progress.

Once it's done generating, you'll be taken to the Installer tab.

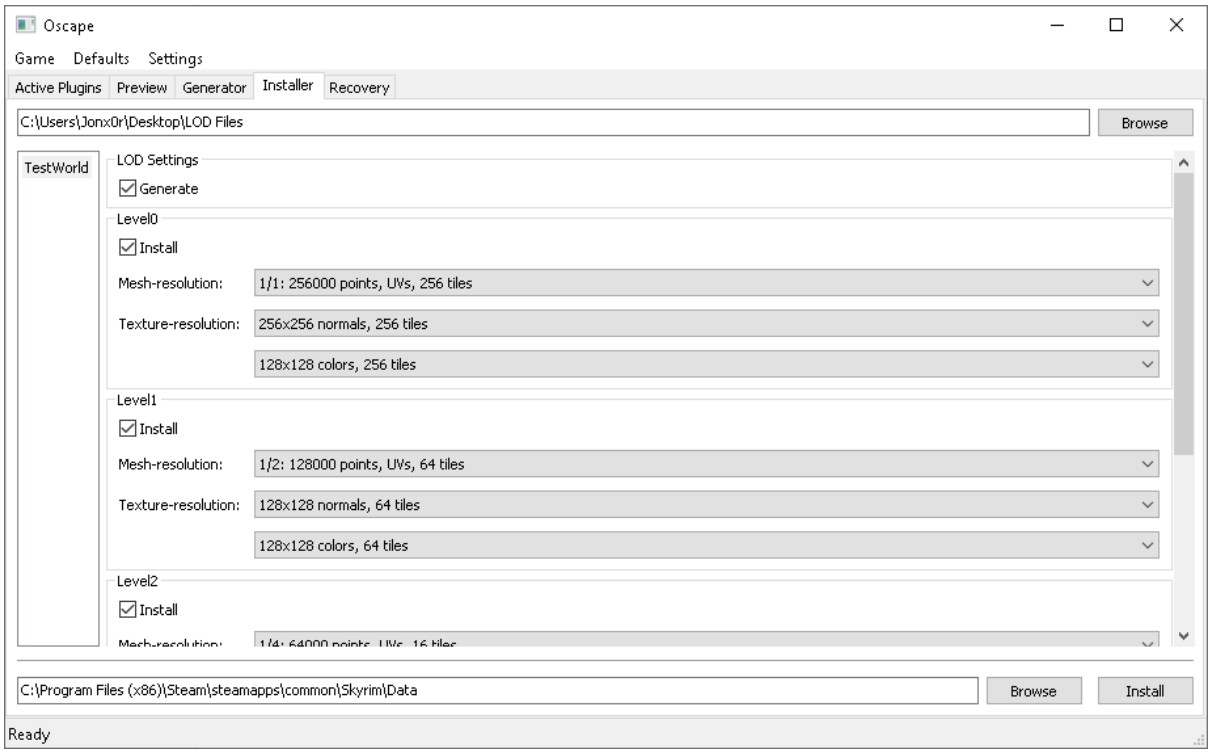


Figure 986 - Installing land LOD files.

Under LOD Settings, tick Generate.

Tick the Install tick box for Level0, Level1, Level2 and Level3.

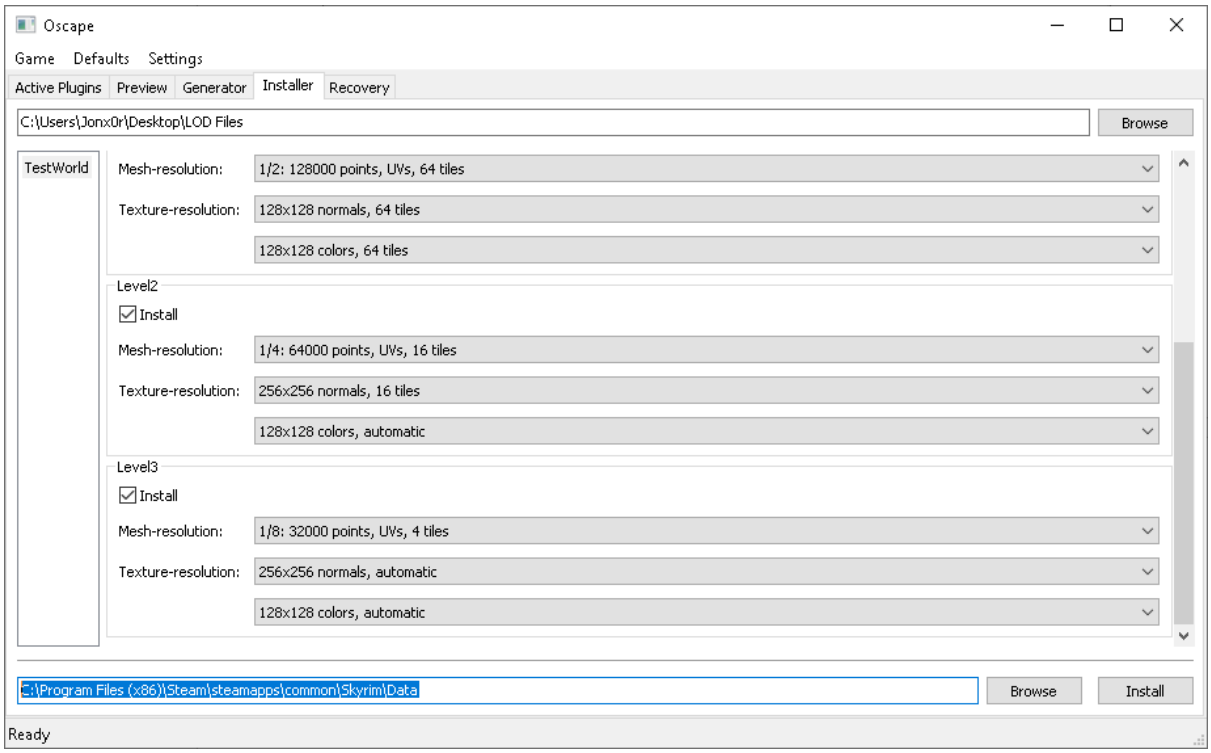


Figure 987 - Setting the install directory.

Make sure the install directory is set to the location of your Skyrim\Data folder.

Click Install to copy the land LOD files to your Skyrim\Data folder.

Confirm the files have been copied to your Skyrim\Data folder.

Go to your Skyrim\Data\Meshes\Terrain\<YourWorldSpaceName> folder and confirm that it contains .btr files prefixed with the name of your world space.

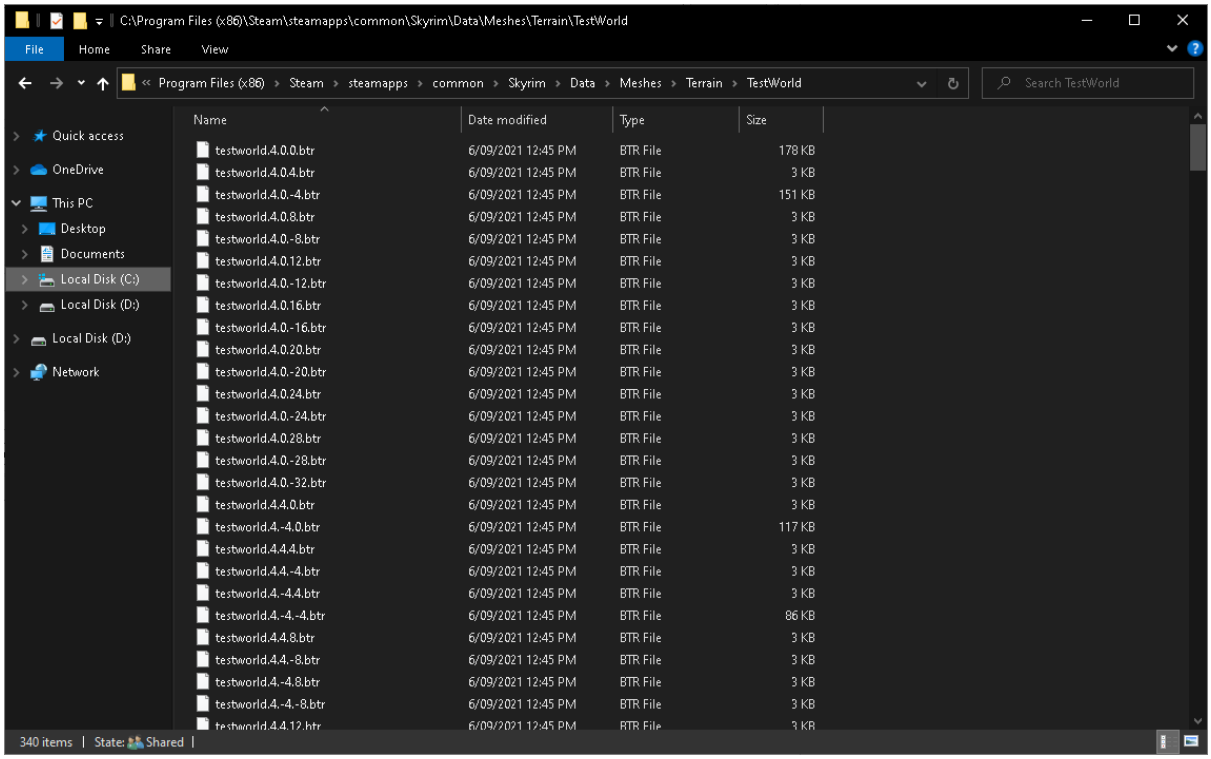


Figure 988 - Land LOD .btr files.

Confirm that the .dds textures were copied to Skyrim\Data\Textures\Terrain\<YourWorldSpaceName>.

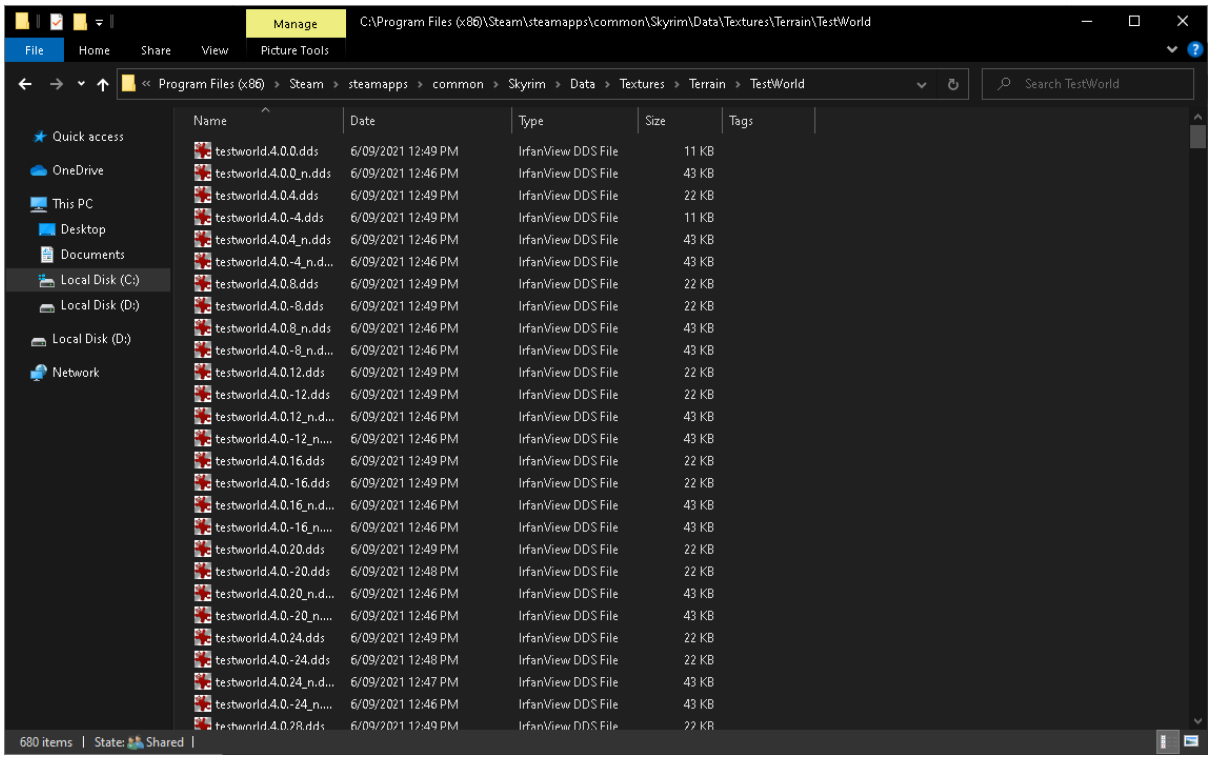


Figure 989 - Land LOD .dds files.

Lastly, confirm a new .lod file exists in the Skyrim\Data\LODSettings folder and is named after your world space.

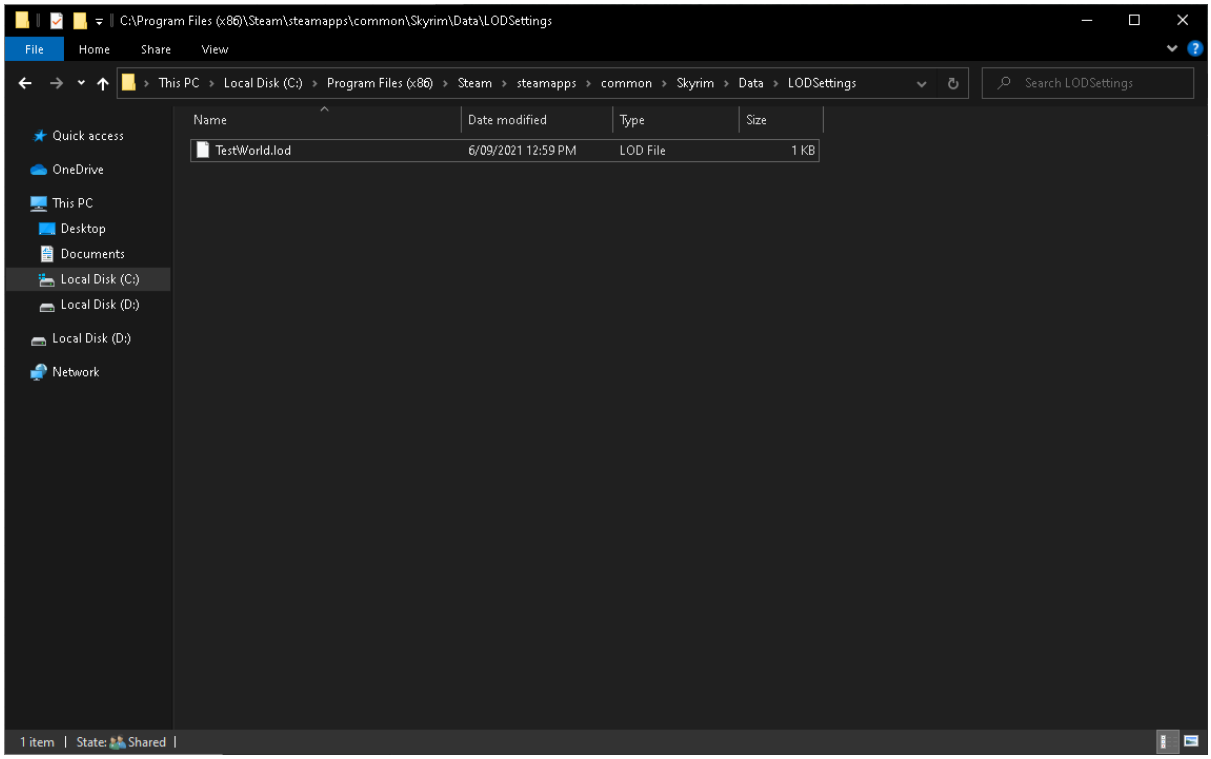


Figure 990 - New LOD Settings file.

Go to your world space in-game and confirm land LOD is present and is textured correctly.



Figure 991 - The test world island from a distance, land LOD only.

We should also be able to see it in the map screen.



Figure 992 - The test world island from the map screen.

Just ignore the hard edges in the clouds on the map screen for now; that can be fixed later.

BUILDING OBJECT LOD USING XEDIT

Object LOD includes objects such as mountain peaks, buildings and city walls seen in the distance.

To generate object LOD we're going to use TES5Edit/SSEEdit which you can download from Nexusmods.

Important: Before you can generate object LOD in xEdit, you need to have made a .lod file for your world space. This file can be created by Oscape when you generate land LOD.

Launch xEdit.

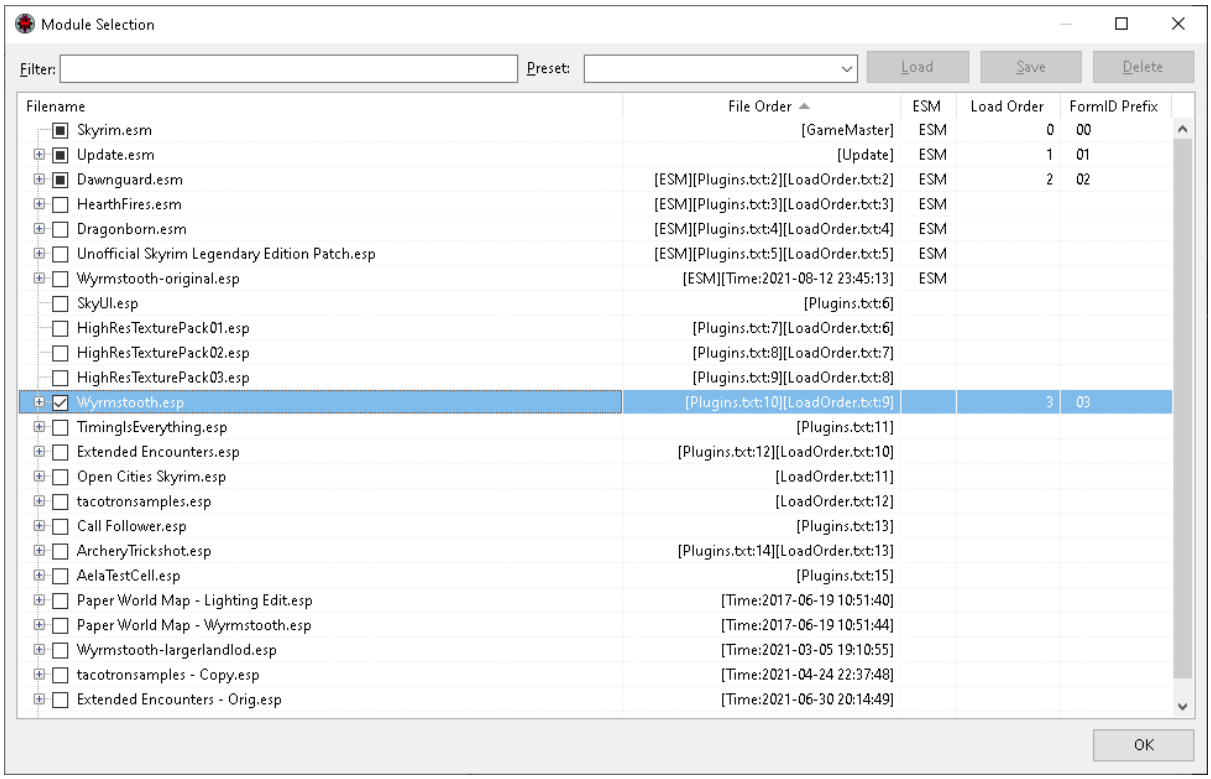


Figure 993 - Select your mod in xEdit.

Select your mod and click OK.

Wait for the background loader to finish.

Right-click on your mod and go to Other > Generate LOD.

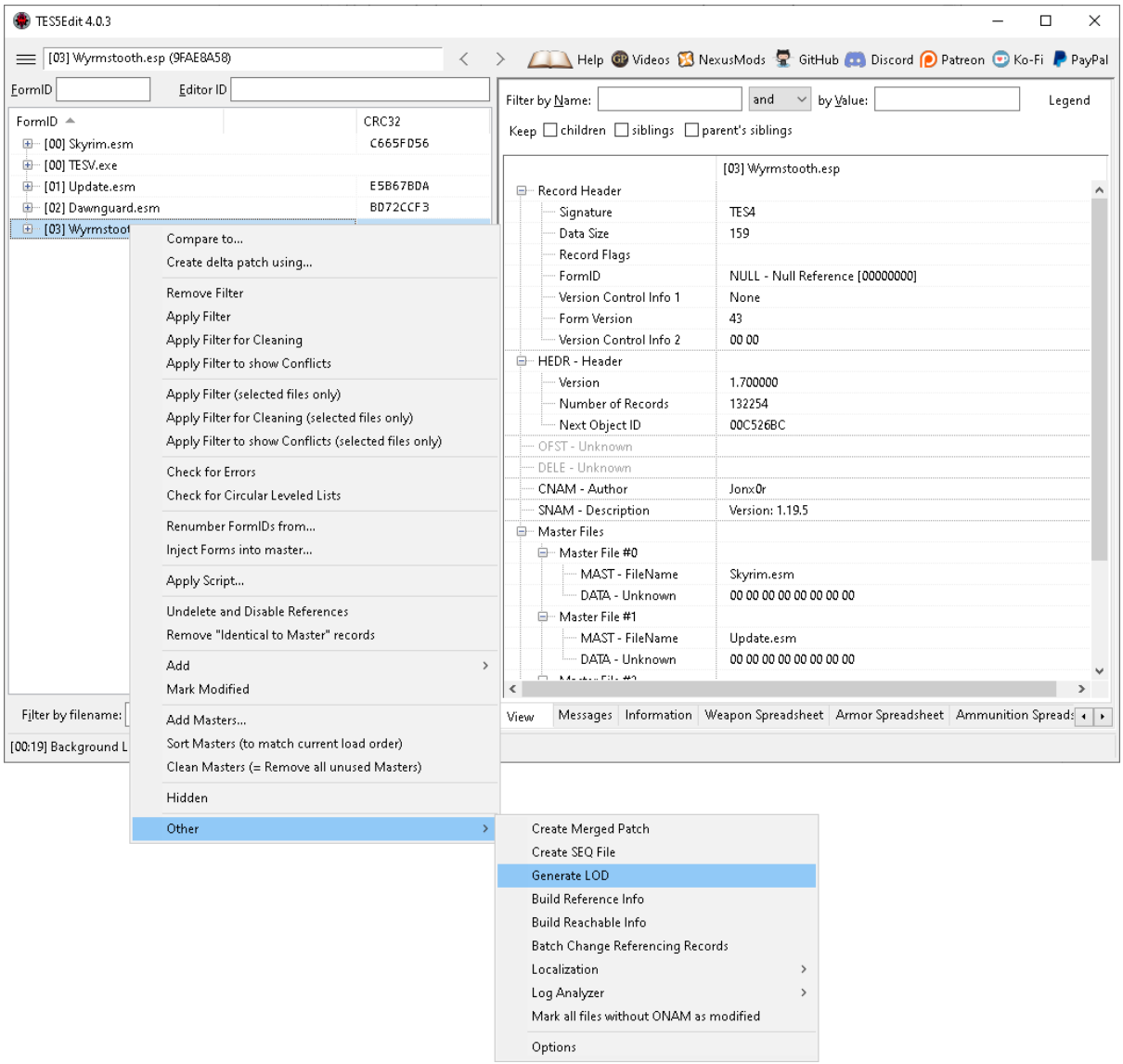


Figure 994 - Generate LOD menu.

Tick the world space(s) you want to generate object LOD for.

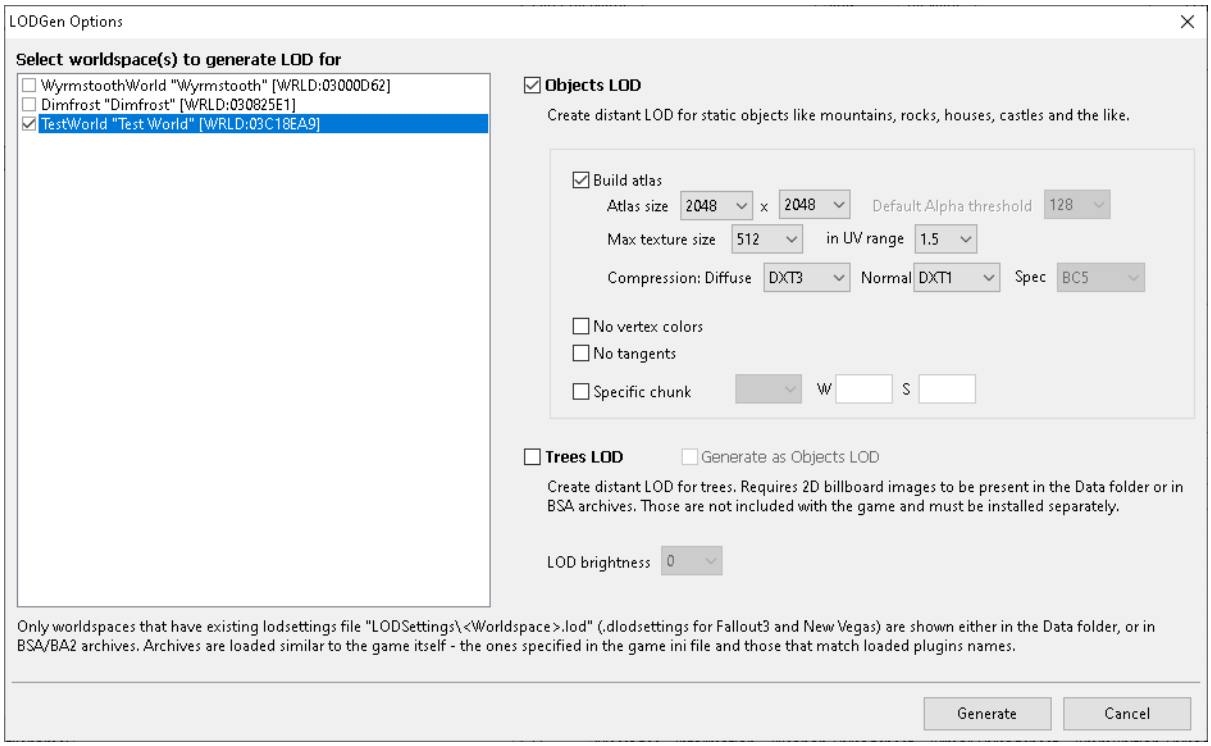


Figure 995 - Selecting the world space to build object LOD for.

Tick Objects LOD.

Click Generate to start building object LOD.

Once it's done, browse to your Skyrim\Meshes\Terrain\<YourWorldspaceName>\Objects folder and confirm that it contains .bto files.

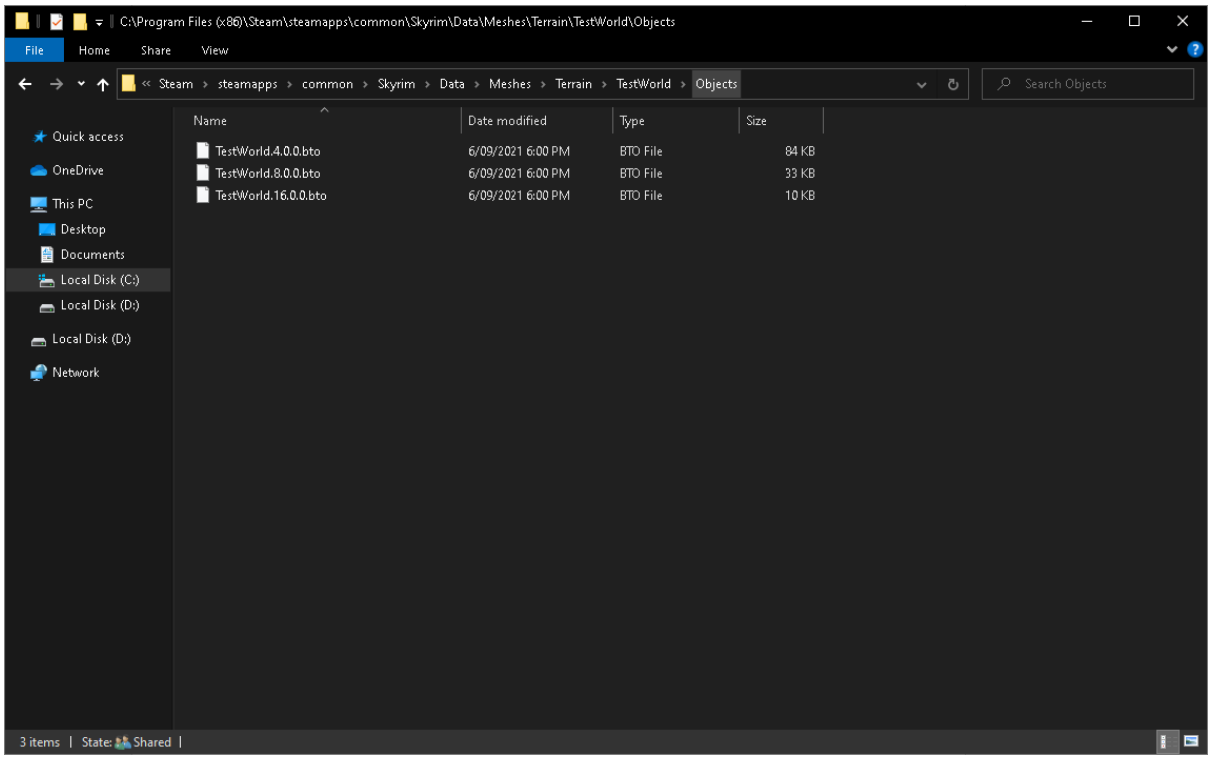


Figure 996 - Object LOD .bto files.

In my TestWorld example, I only have a couple objects with LOD so only a couple files were generated.

Navigate to your Skyrim\Data\Textures\Terrain\<YourWorldspaceName>\Objects folder and confirm it contains two .dds files.

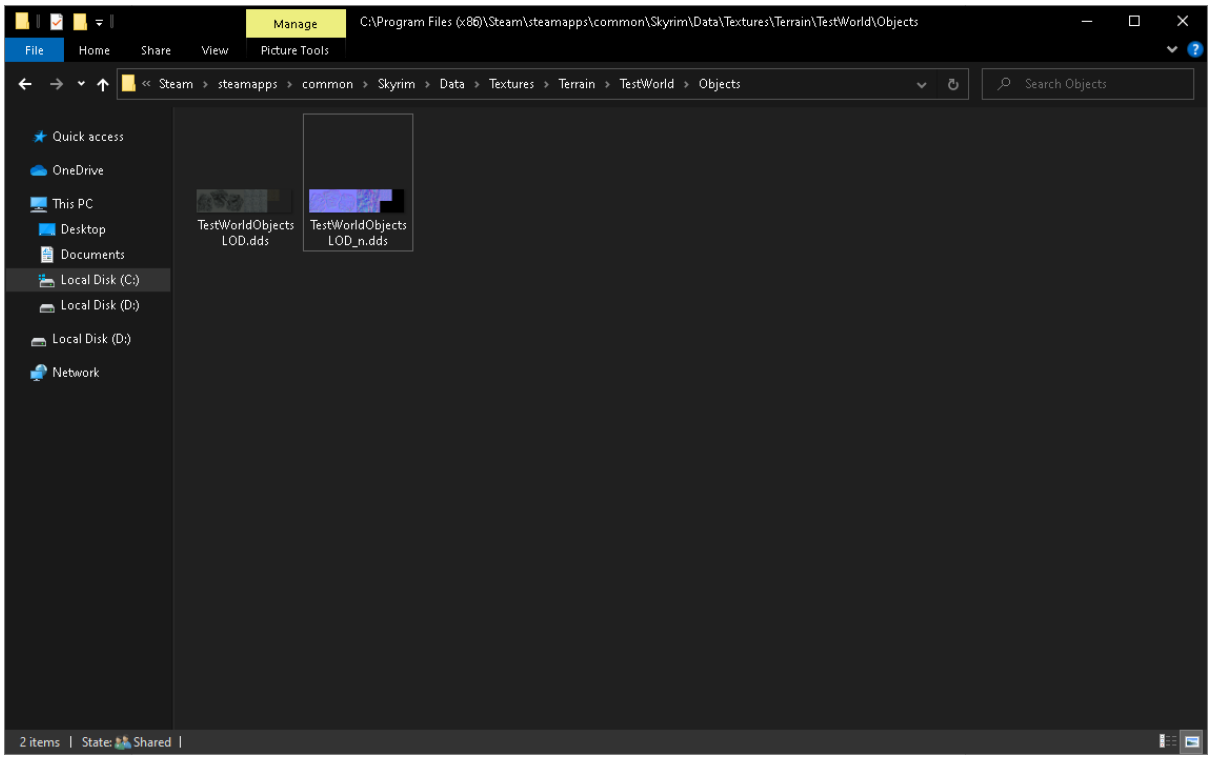


Figure 997 - Object LOD texture index.

Go to your world space in-game and confirm object LOD is present and is textured correctly.



Figure 998 - The mountain peak on our island can now be seen from a distance.

As you can see from the screenshot above, the mountain peak can now be seen from a distance.

It can also be seen on the world map.



Figure 999 - Object LOD also visible on the world map.

BUILDING TREE LOD USING XEDIT

Similar to object LOD, tree LOD shows distant trees in unloaded cells.

To generate object LOD we're going to use TES5Edit/SSEEdit which you can download from Nexusmods.

Important: Before you can generate object LOD in xEdit, you need to have made a .lod file for your world space. This file can be created by Oscape when you generate land LOD.

The first thing we'll need to do is download the tree billboards which you can download from the TES5LODGen page on Nexusmods.

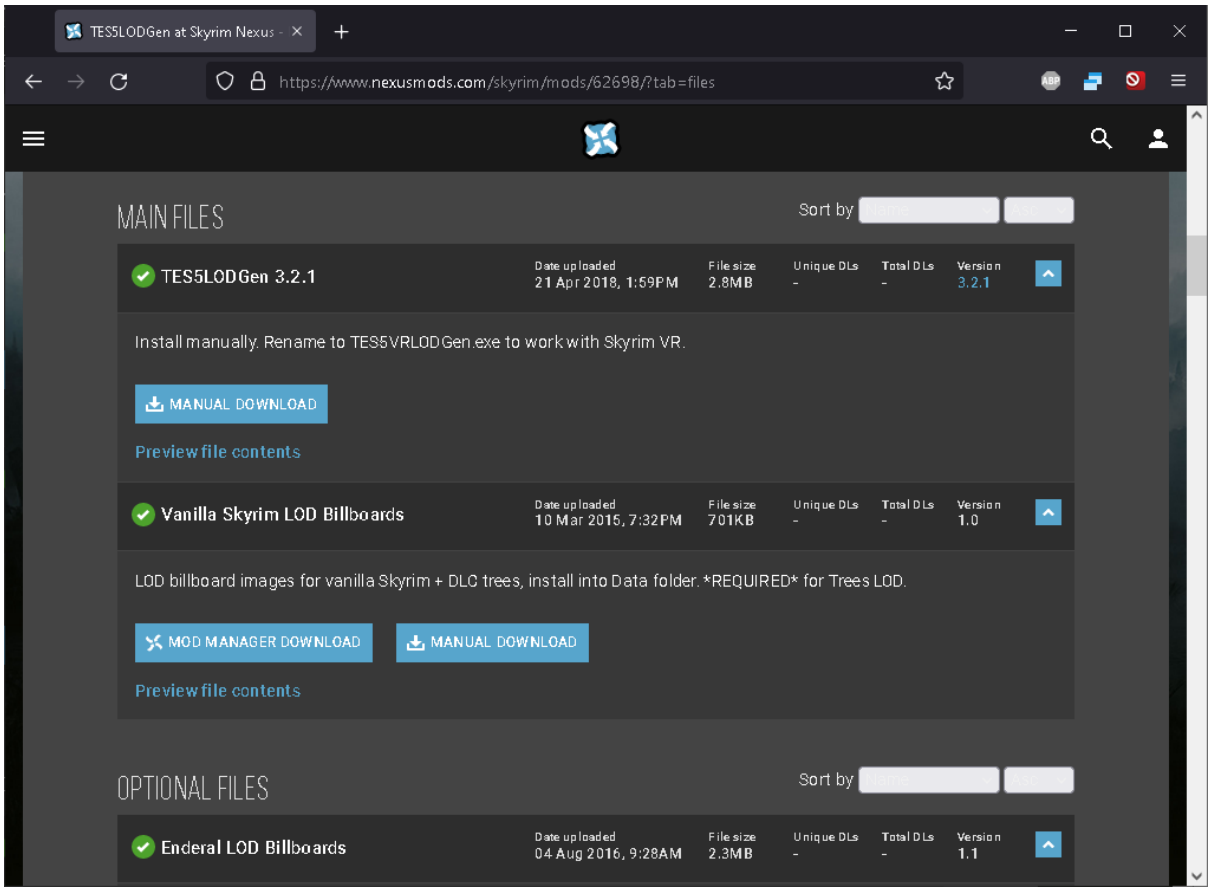


Figure 1000 - Vanilla Skyrim LOD Billboards download source.

Open the .zip archive and extract the textures folder to your Skyrim\Data or Skyrim Special Edition\Data folder.

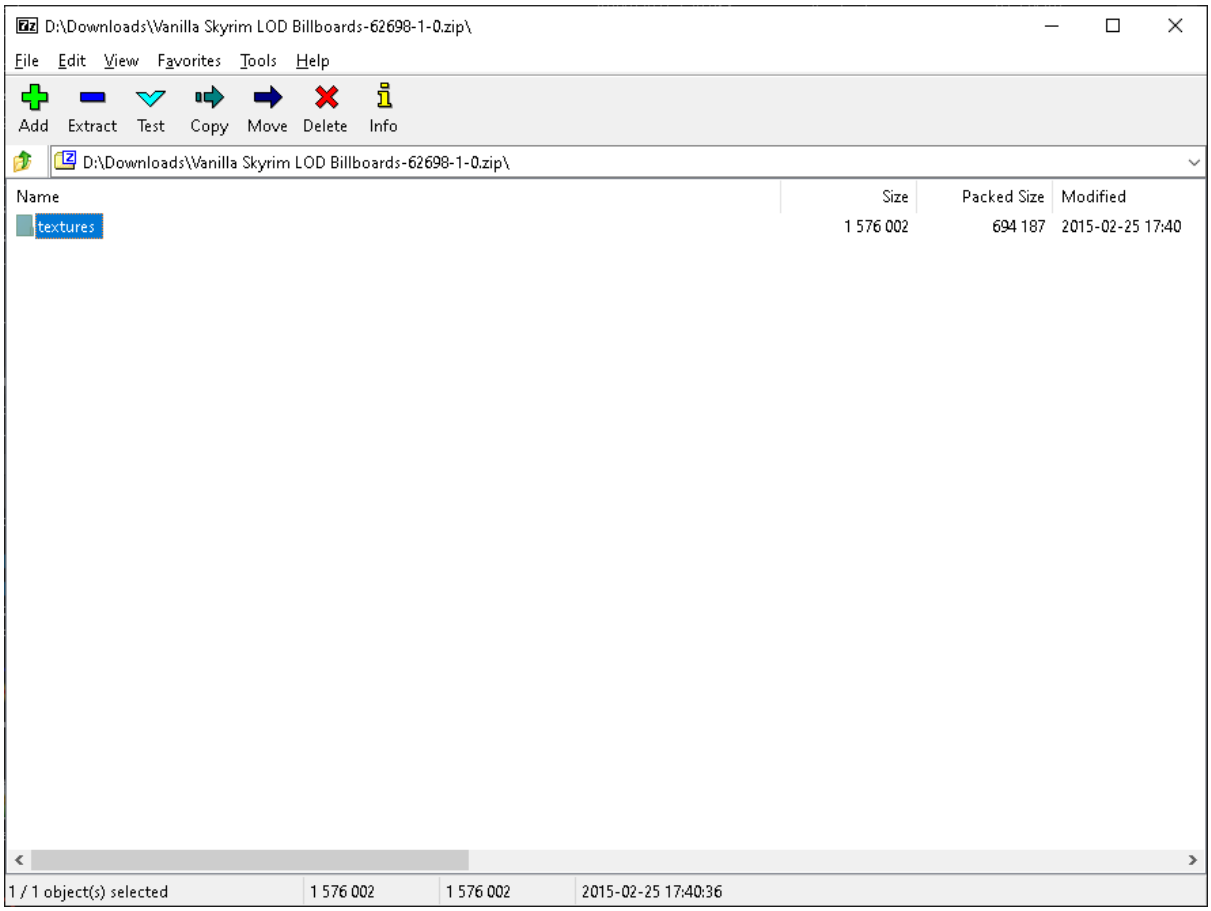


Figure 1001 - Vanilla Skyrim LOD Billboards .zip contents.

Now we can launch xEdit.

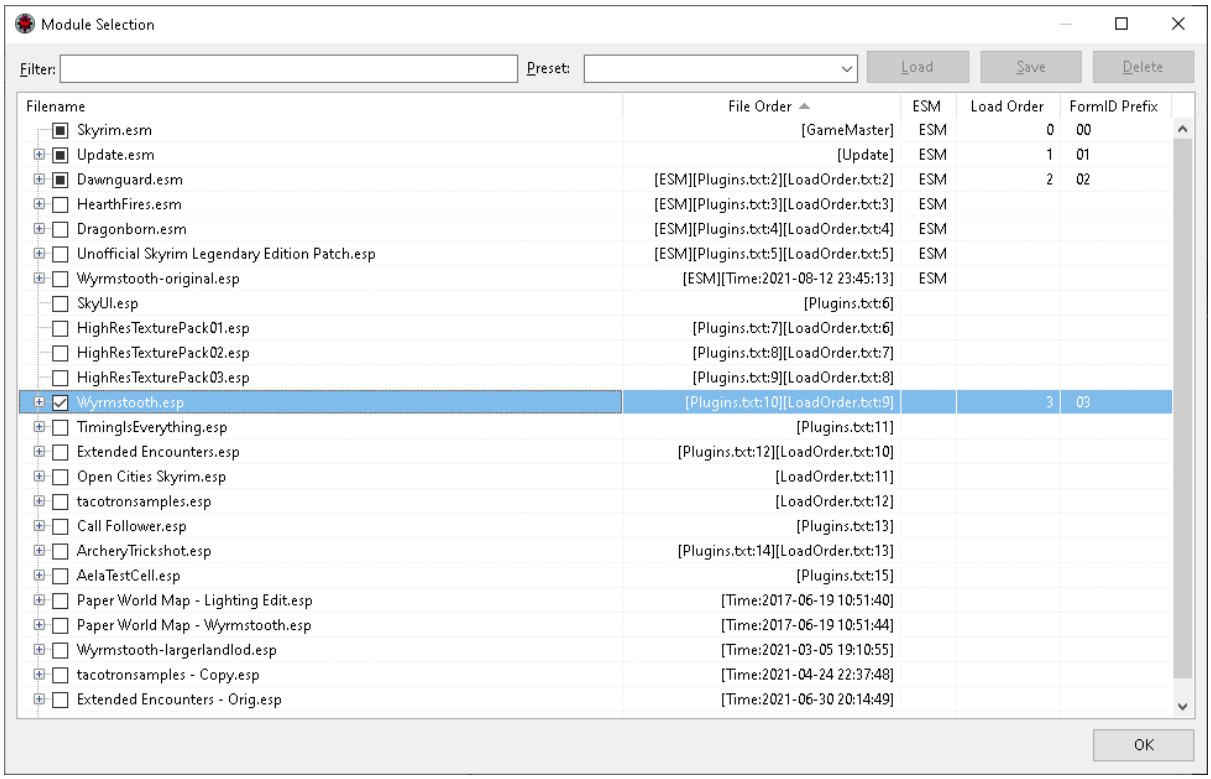


Figure 1002 - Select your mod in xEdit.

Select your mod and click OK.

Wait for the background loader to finish.

Right-click on your mod and go to Other > Generate LOD.

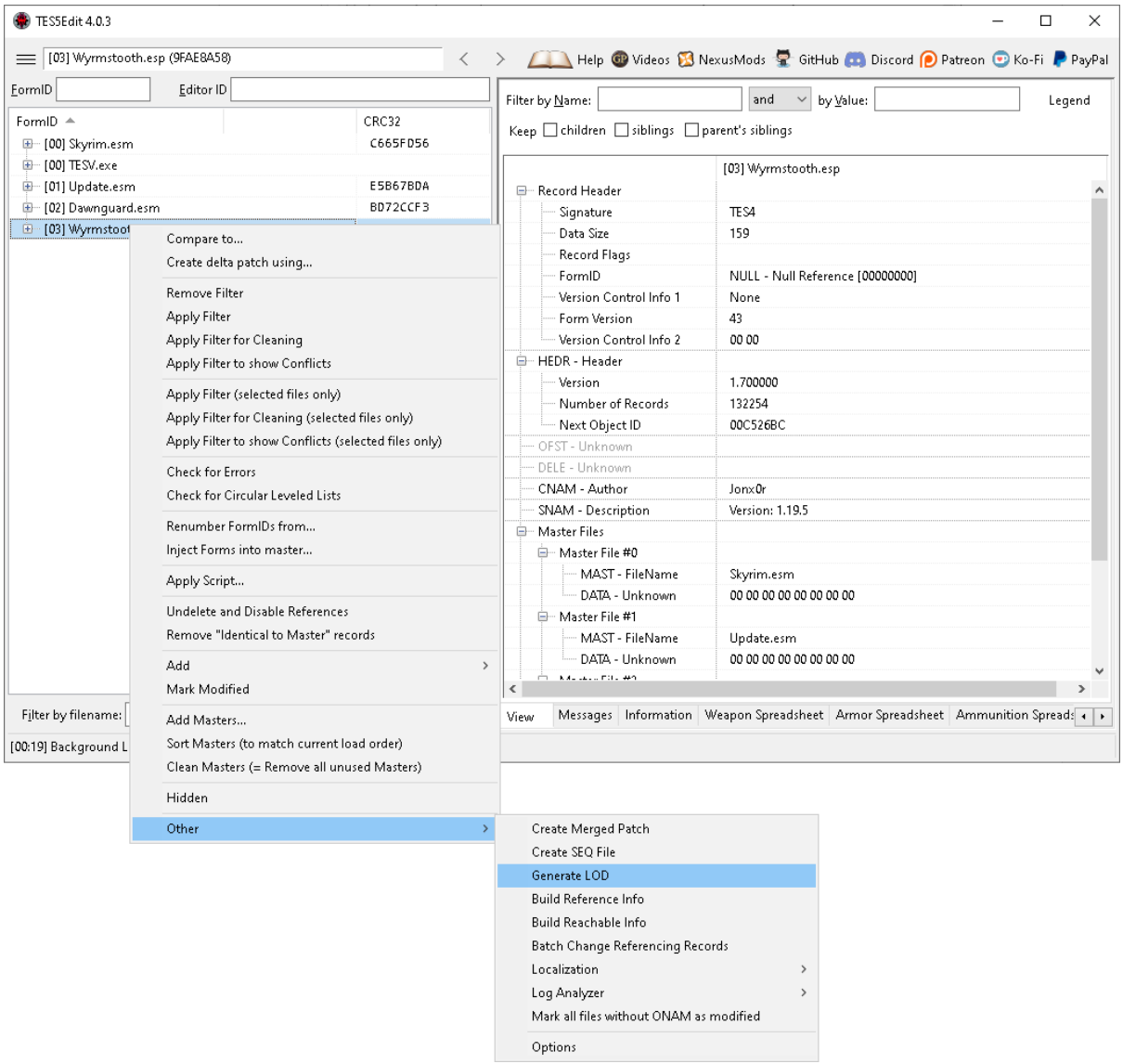


Figure 1003 - Generate LOD menu.

Tick the world space(s) you want to generate object LOD for.

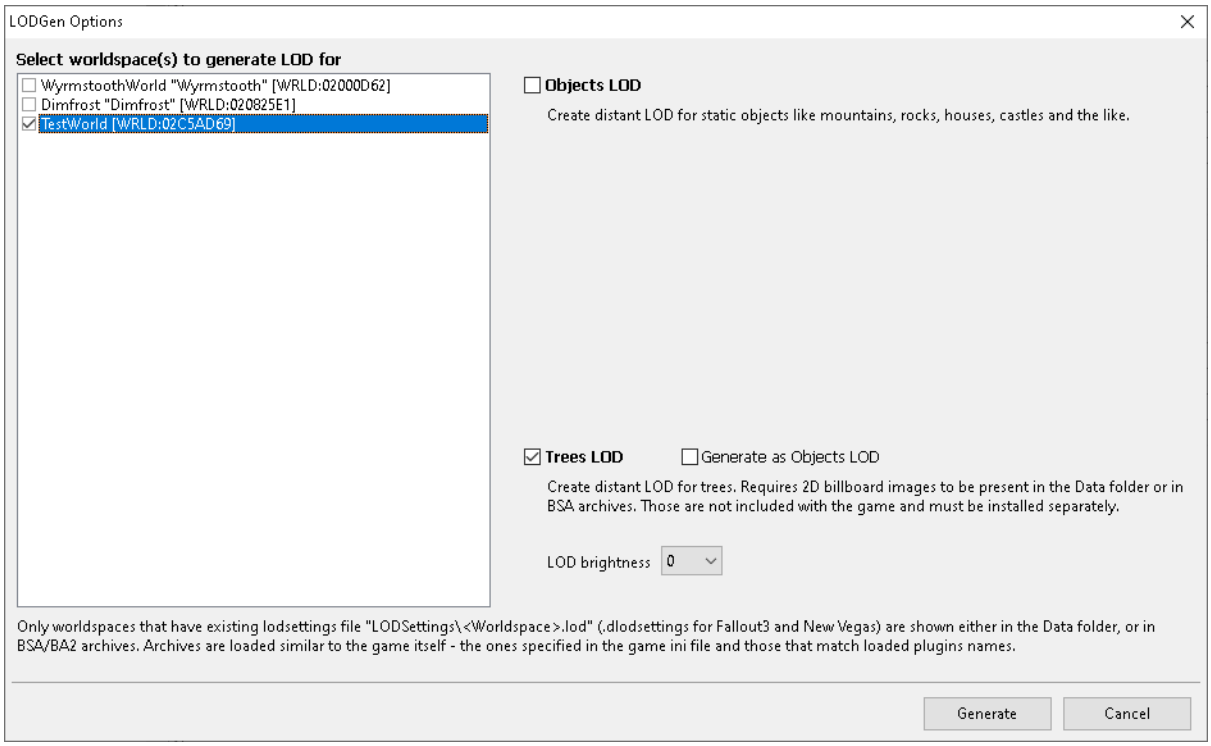


Figure 1004 - Generating tree LOD.

Click Generate to begin generating tree LOD.

Now navigate to your Skyrim\Data\Meshes\Terrain\<YourWorldspaceName>\Trees folder and confirm it contains .btt files.

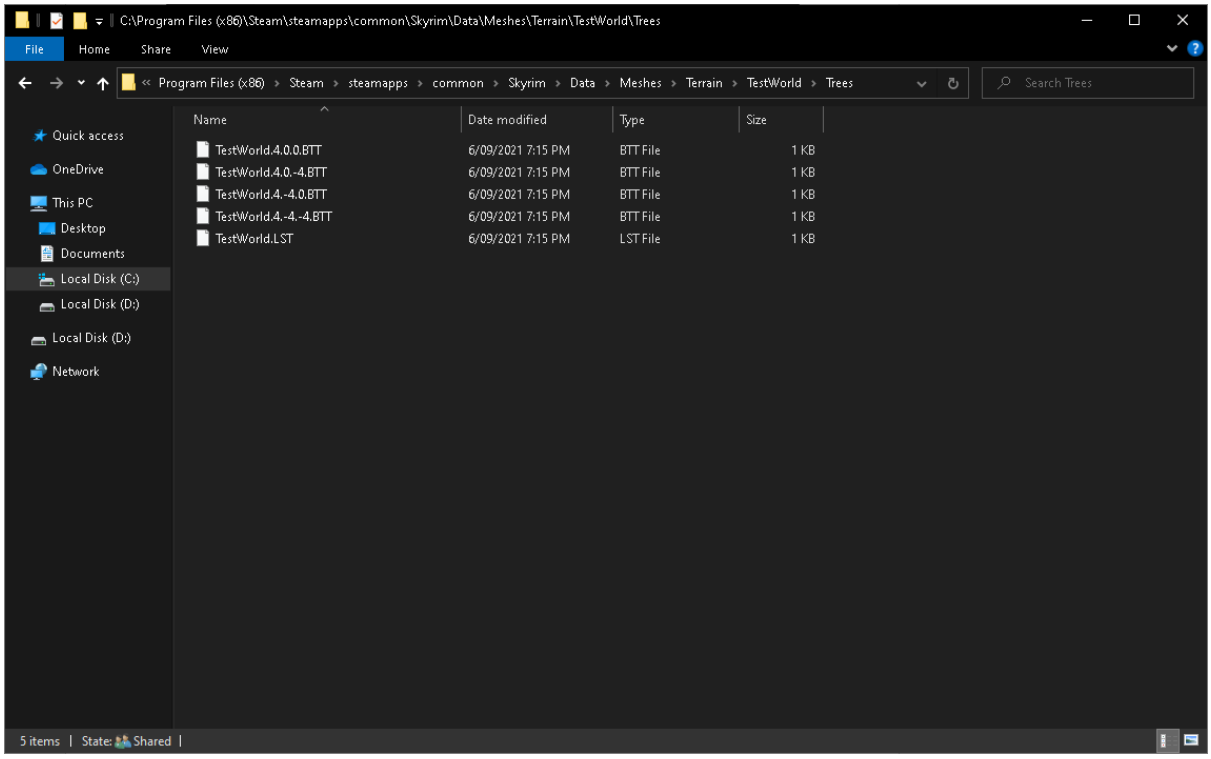


Figure 1005 - Tree LOD .btt files.

Again, I only have a handful of .btt files in the screenshot above because the sample world space I made is very small.

Browse to your Skyrim\Data\Textures\Terrain\<YourWorldspaceName>\Trees folder and confirm you have a <YourWorldspaceName>treelod.dds file here.

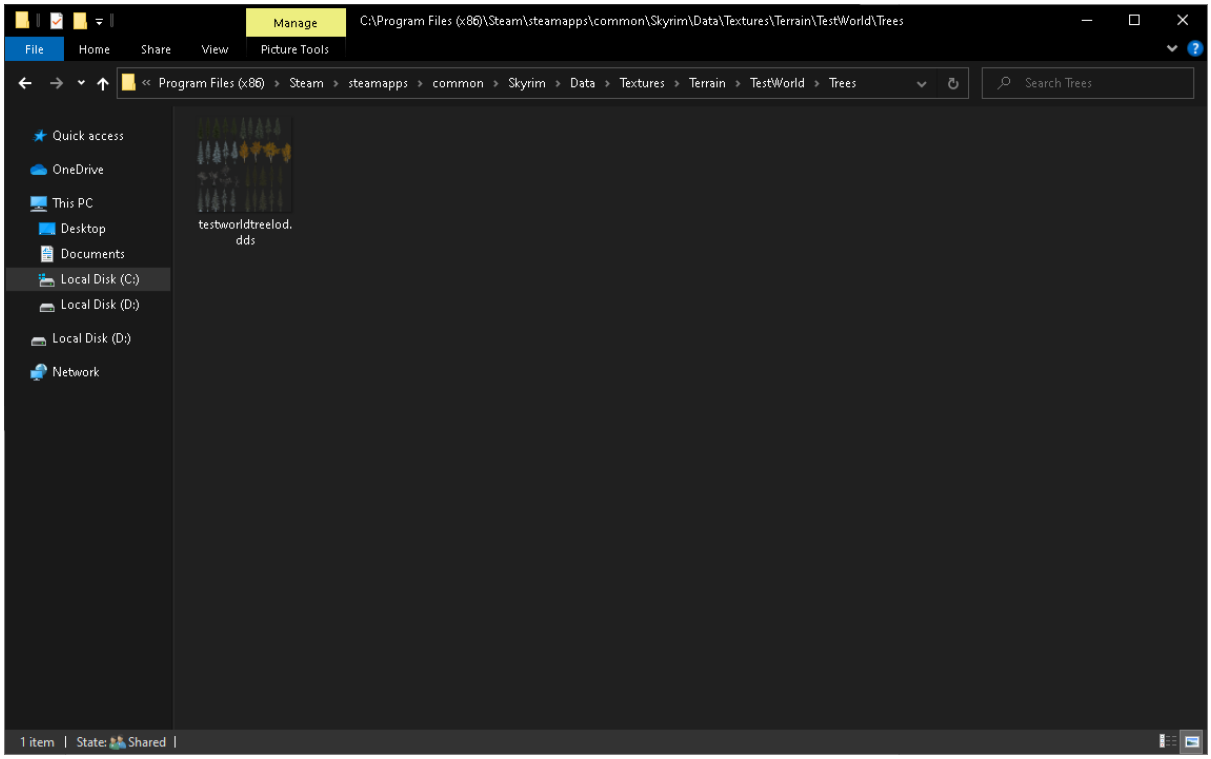


Figure 1006 - Pasting in and renaming tamrieltreelod.dds.

Go to your world space in-game and confirm tree LOD is present and is textured correctly.



Figure 1007 - Trees can now be seen in the distance on our little island.

You should be able to see trees on the island now from far away.

Important: For Skyrim Special Edition, I would recommend ticking 'Generate as Objects LOD'. The Creation Kit can't pack .lst files into a .bsa archive.

Tree LOD requires an .lst file that's created when generating tree LOD. The problem, however, is that the Creation Kit can't pack this .lst file into a .bsa archive. To work around this problem, you can generate tree LOD as part of object LOD which requires no .lst file.

In order to do this you will need to tick 'Objects LOD', 'Trees LOD' and 'Generate as Objects LOD' as per the screenshot below:

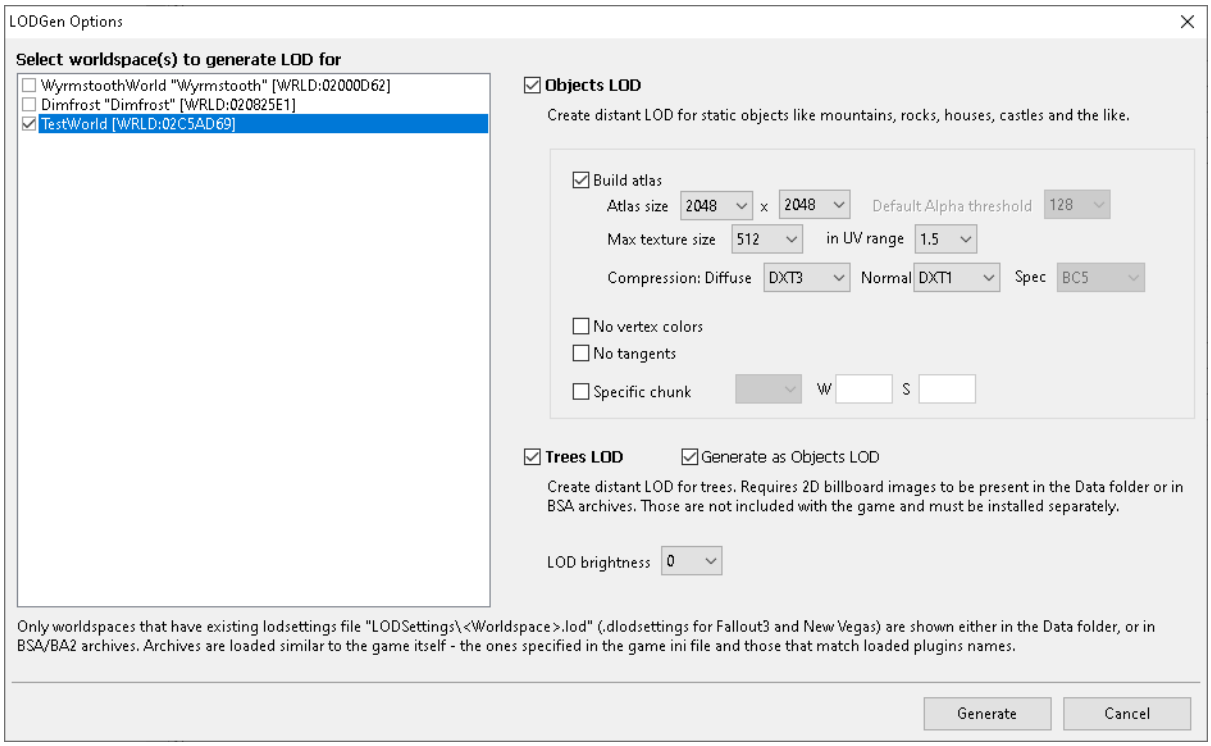


Figure 1008 - Generating tree LOD as object LOD.

FINAL ADJUSTMENTS TO WORLD SPACE SETTINGS

In this section I'll be covering a few final things that need to be changed on a new world space.

The first thing we're going to need to do is create a new cloud layer for our world map. The default cloud map for the Tamriel world space conforms to the shape of that world space, so this needs to be fixed.

The default skyrimworldmapcloudbank.nif cloud layer can be extracted from the sky folder in the 'Skyrim - Meshes.bsa' archive.

See Unpacking the BSA Archives for instructions on exporting assets from the BSA archives.

We'll also need NifSkope, which you can download from their website.

In NifSkope, it will look like this:

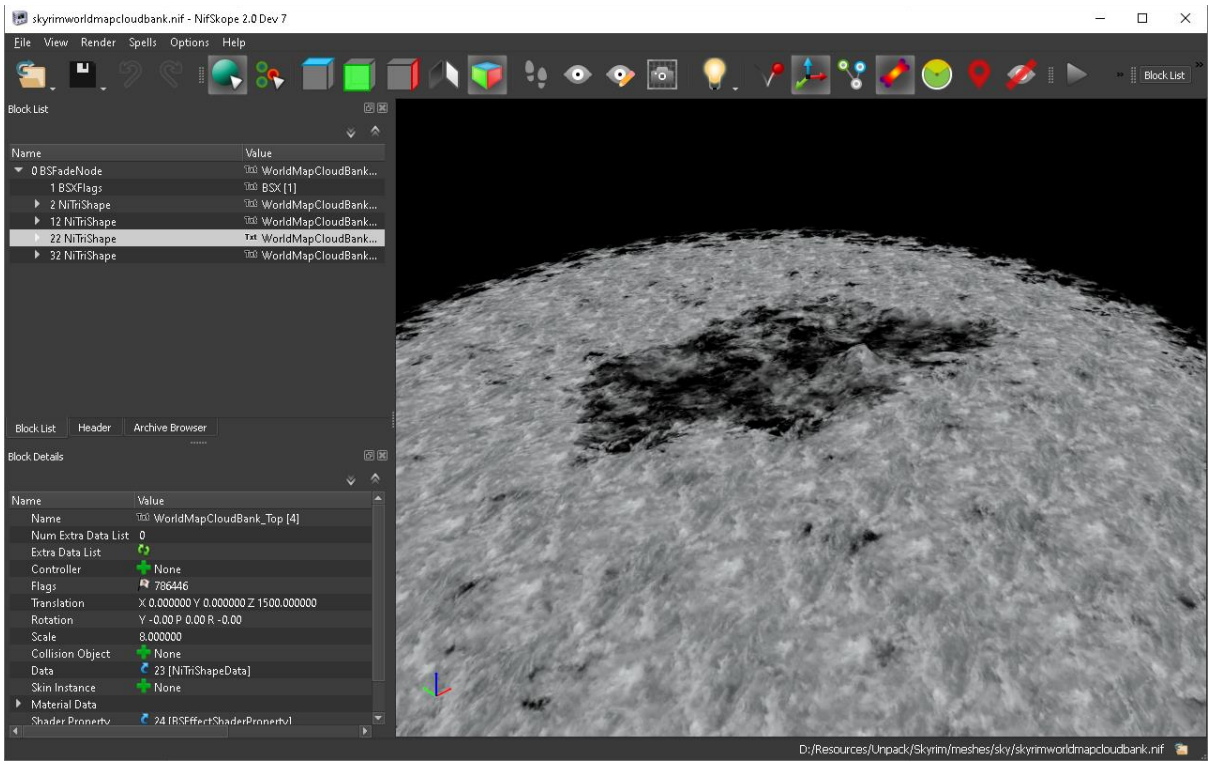


Figure 1009 - Base game cloud layer in NifSkope.

The easiest way to fix this is to weld the vertices. To accomplish this I'll be working with 3ds Max 2017.

The NIF Export/Import Plugin for 3ds Max 2017 is available on Nexusmods. For other versions of 3ds Max, try NIF Plugin for 3ds Max 2015-2018 available on Nexusmods.

Launch 3ds Max.

Go to Import.

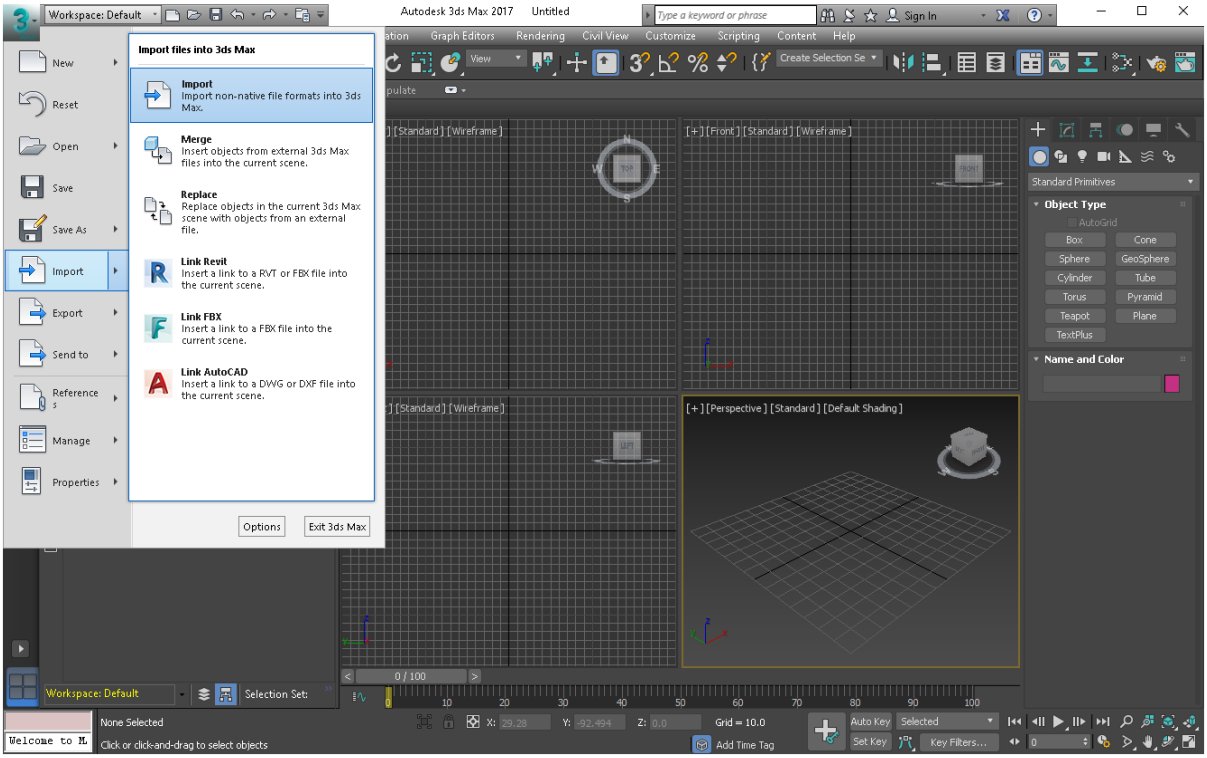


Figure 1010 - Importing a file.

Set the 'Files of type' to 'NetImmerse/Gamebryo (*.NIF, *.KF)' and select the skyrimworldmapcloudbank.nif file.

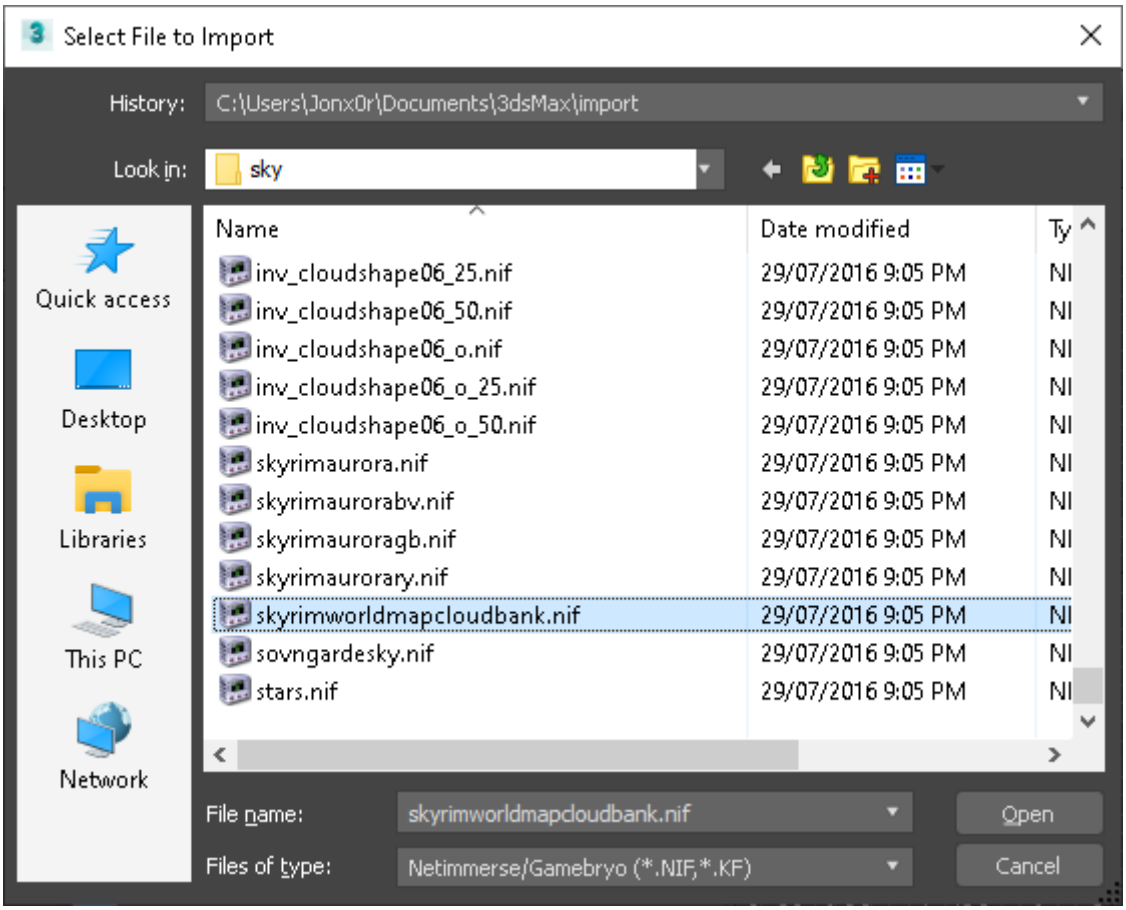


Figure 1011 - Selecting the default cloud bank .nif.

Click Open to import the .nif into 3ds Max.

Set Game to Skyrim.

I unticked the options under Geometry, Animation and Skeleton. I also unticked Flip UV, Ignore User Prop Buffers and Ignore Root Node in Miscellaneous.

Click Import.

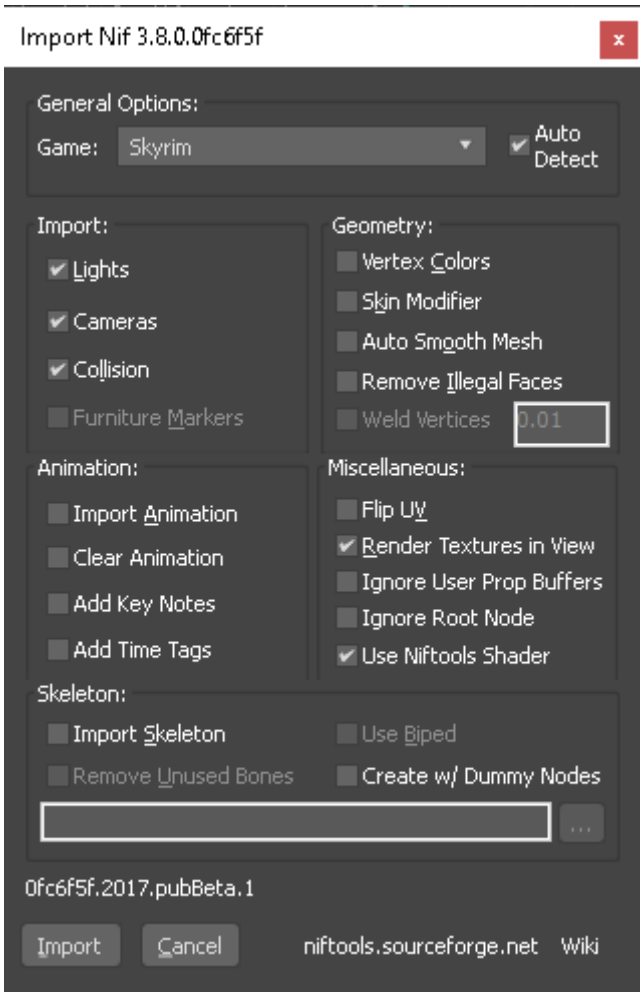


Figure 1012 - Import Nif screen.

The imported .nif should look like this.

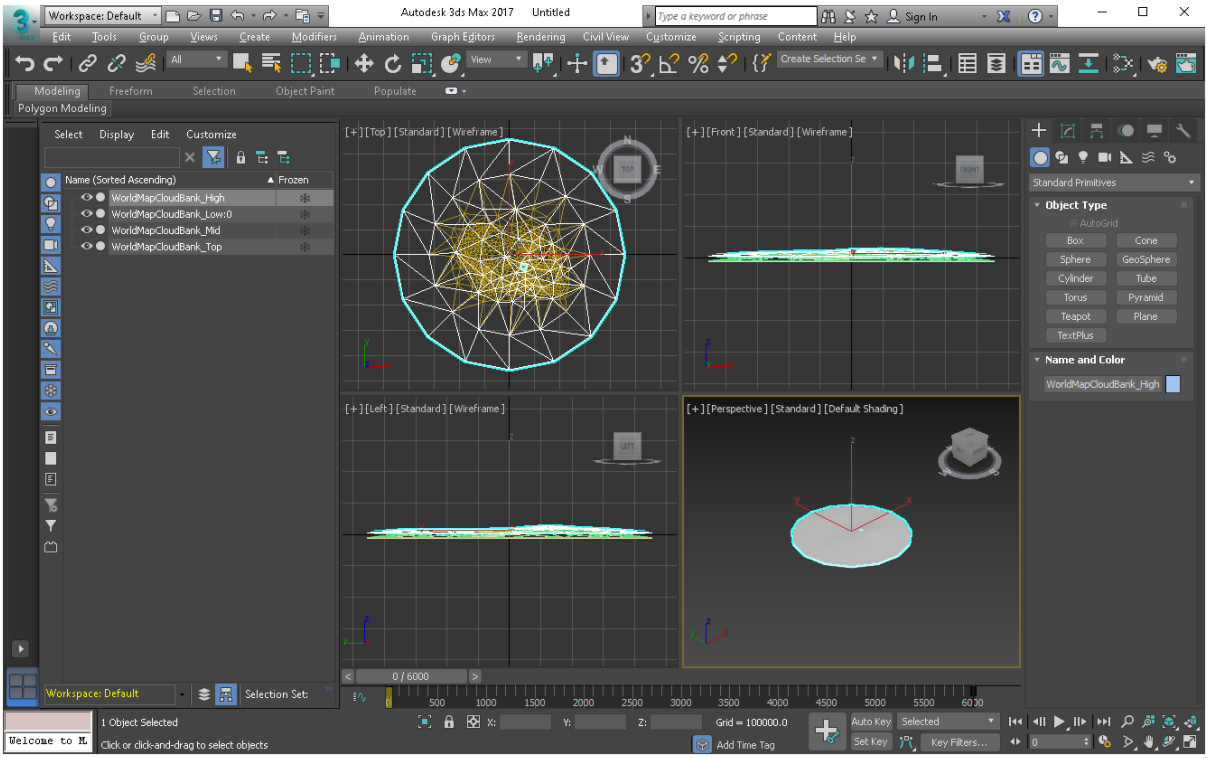


Figure 1013 - The cloudbank .nif imported into 3ds max.

The object is comprised of four circular planes to add multiple layers of clouds to the world map.

We can fix the clipping by welding the vertices within the cloud planes together.

Select the first plane, click on Modify, then click on Vertex mode.

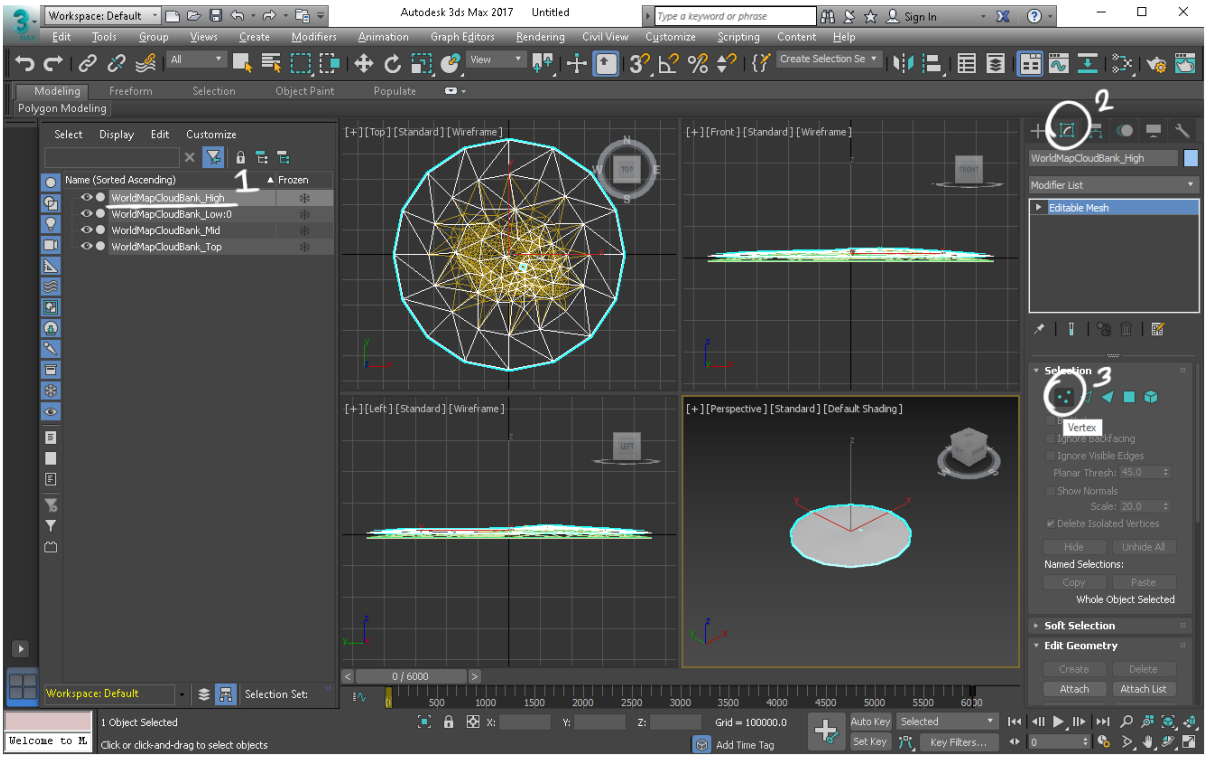


Figure 1014 - Entering vertex mode.

Set the Selection Region tool to lasso.

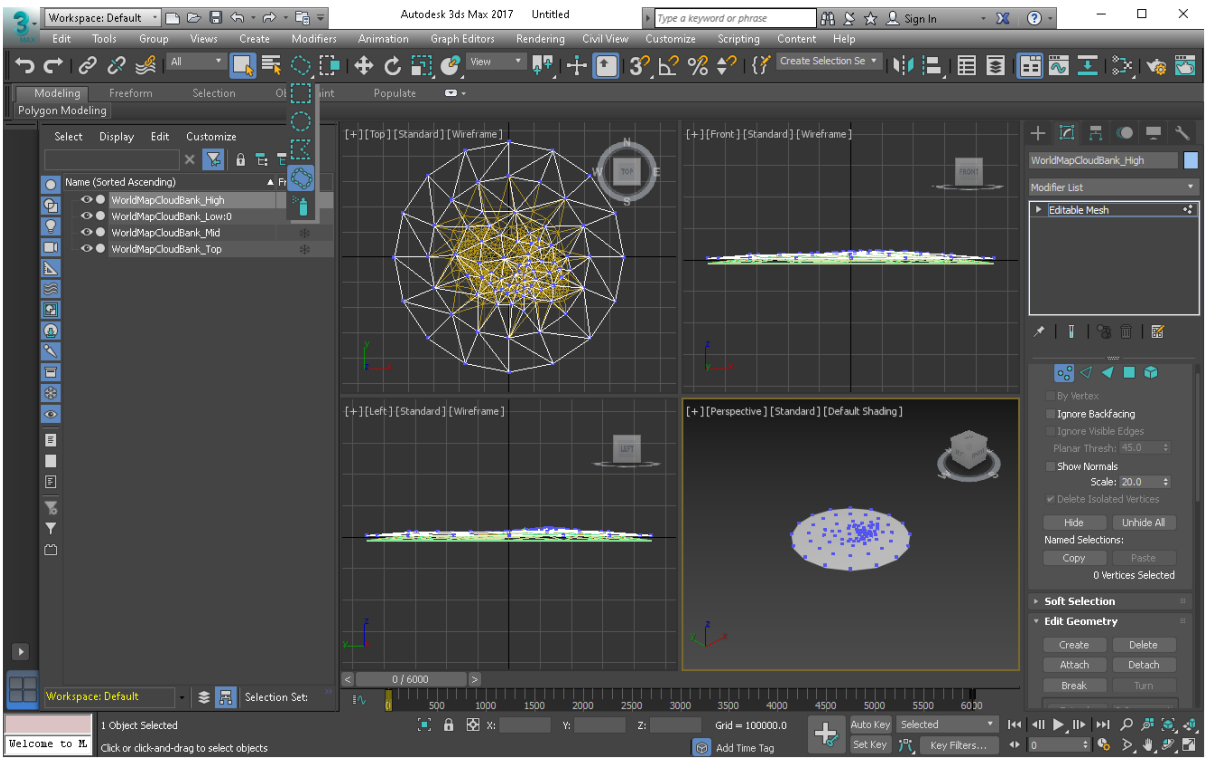


Figure 1015 - Setting the selection region tool to lasso.

In the Top view, lasso the vertices within the plane. Don't lasso the outside edge vertices.

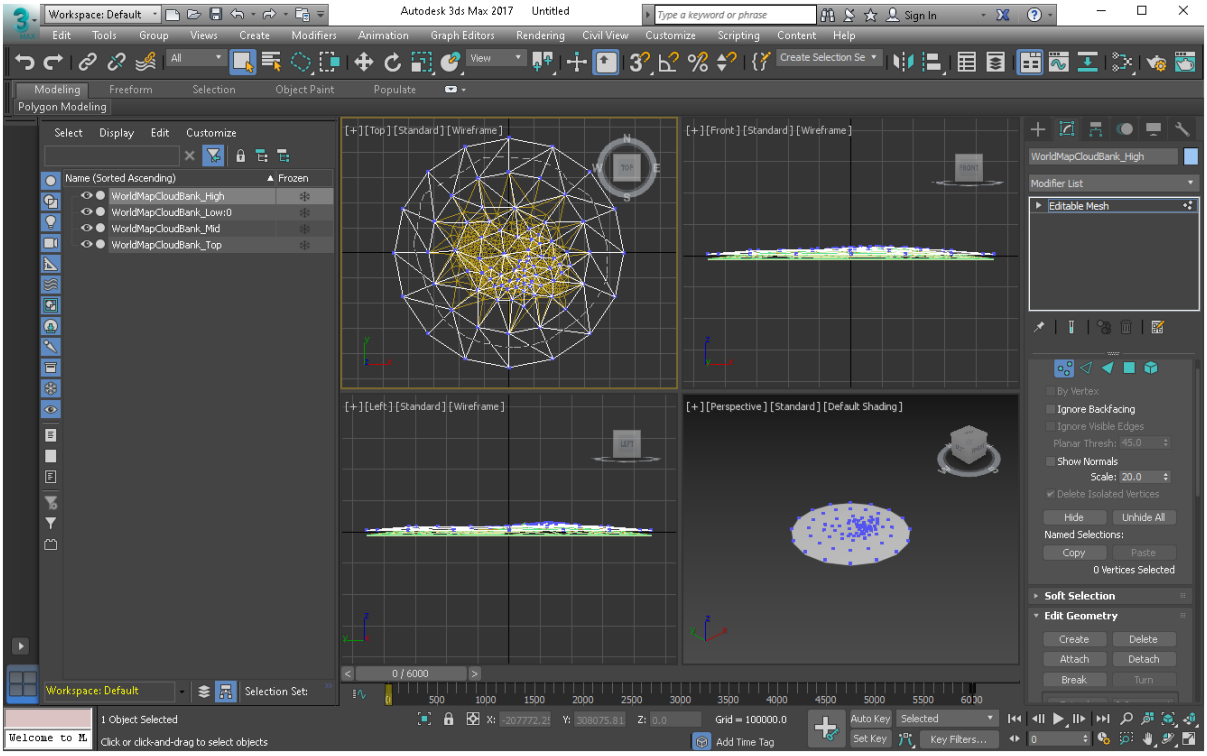


Figure 1016 - Lassoing the vertices.

If you try to weld, you'll probably see a message saying 'No vertices within weld threshold'.

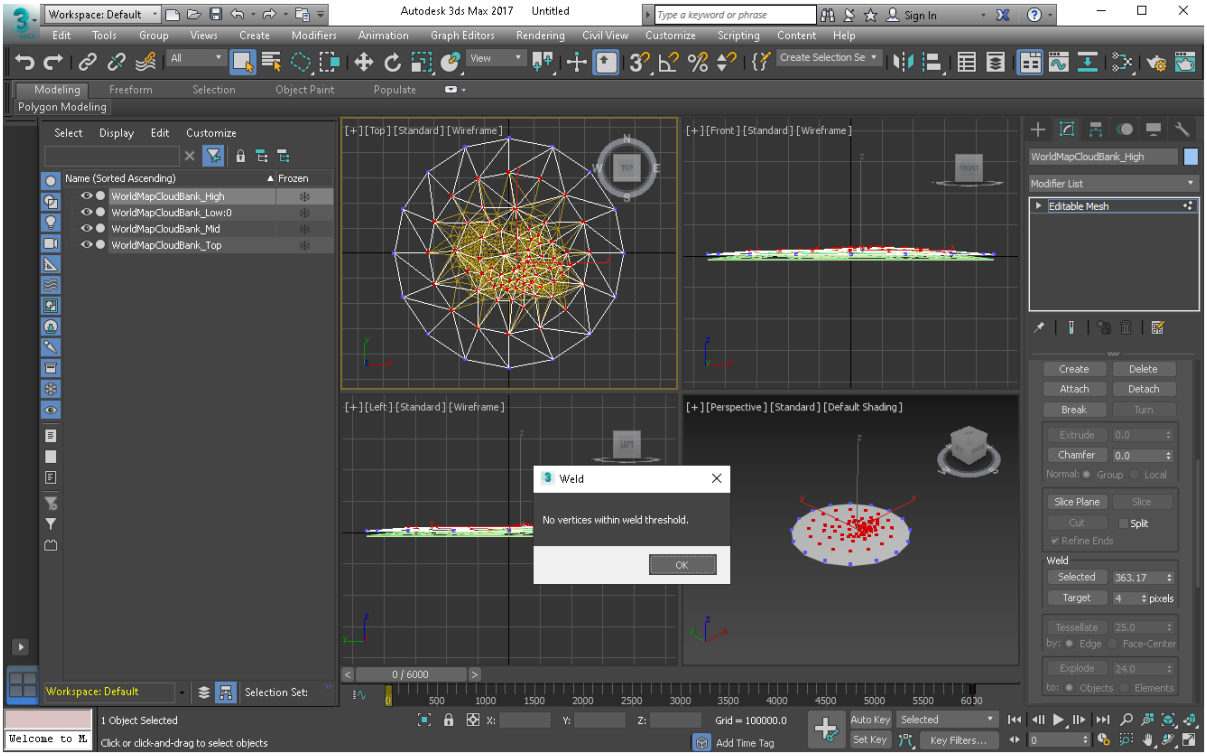


Figure 1017 - Weld message.

Increase the weld value then click Selected to weld all the selected vertices together to a single point.

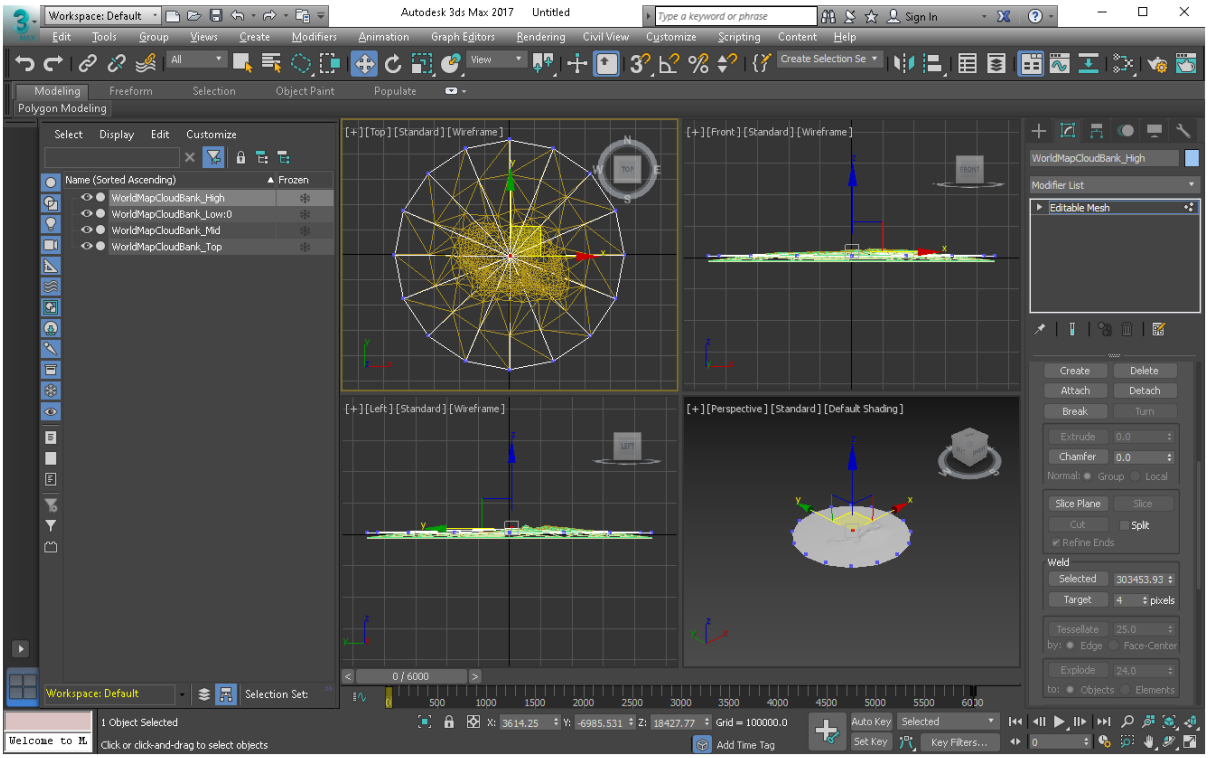


Figure 1018 - Welding all selected vertices.

You can then use the move tool to reposition the central vertex in the centre of the plane.

Repeat these steps for the other three layers.

Next, we'll need to rework the UV map.

If you haven't already, you'll need to add the texture paths to 3ds Max in order to see textures in the viewports. See the section on Adding Texture Paths to 3ds Max for steps on how to do this.

Select WorldMapCloudBank_High in the object list.

Hide the other layers by clicking on the 'eye' icon next to them.

Then go to Modifiers, click on the Modifier List drop-down and add UVW Map.

Now click on the UVW Map modifier to go to its settings.

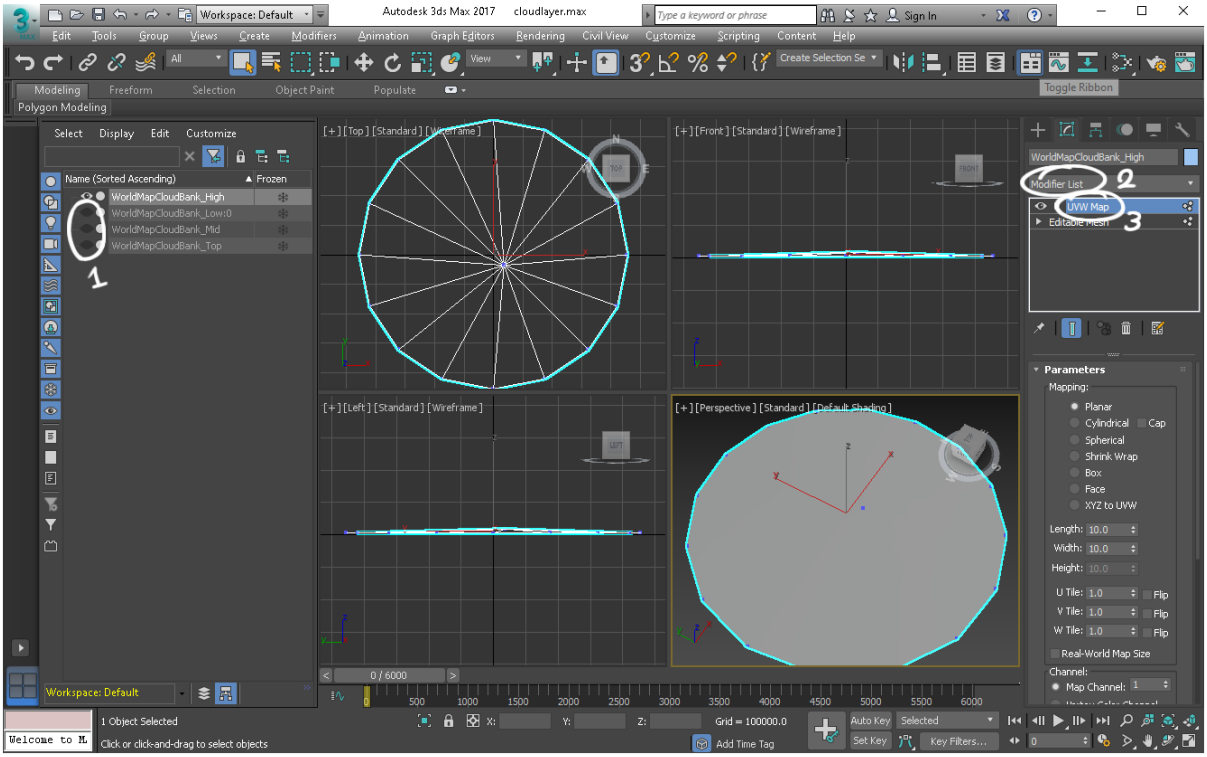


Figure 1019 - Adding a UVW Map modifier.

Firstly, make sure Mapping is set to Planar.

Go down and look for Alignment and set it to 'Z'.

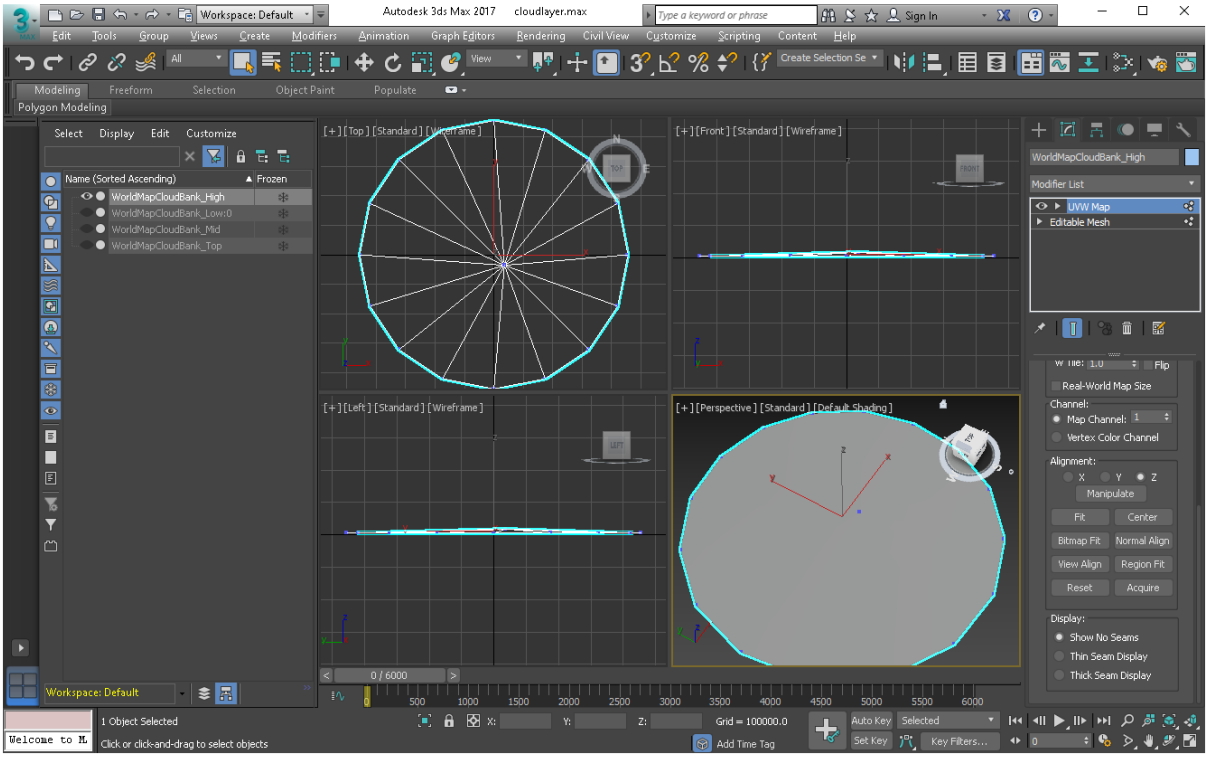


Figure 1020 - Setting UV map alignment.

Now go back up and set Length and Width to 10000.0.

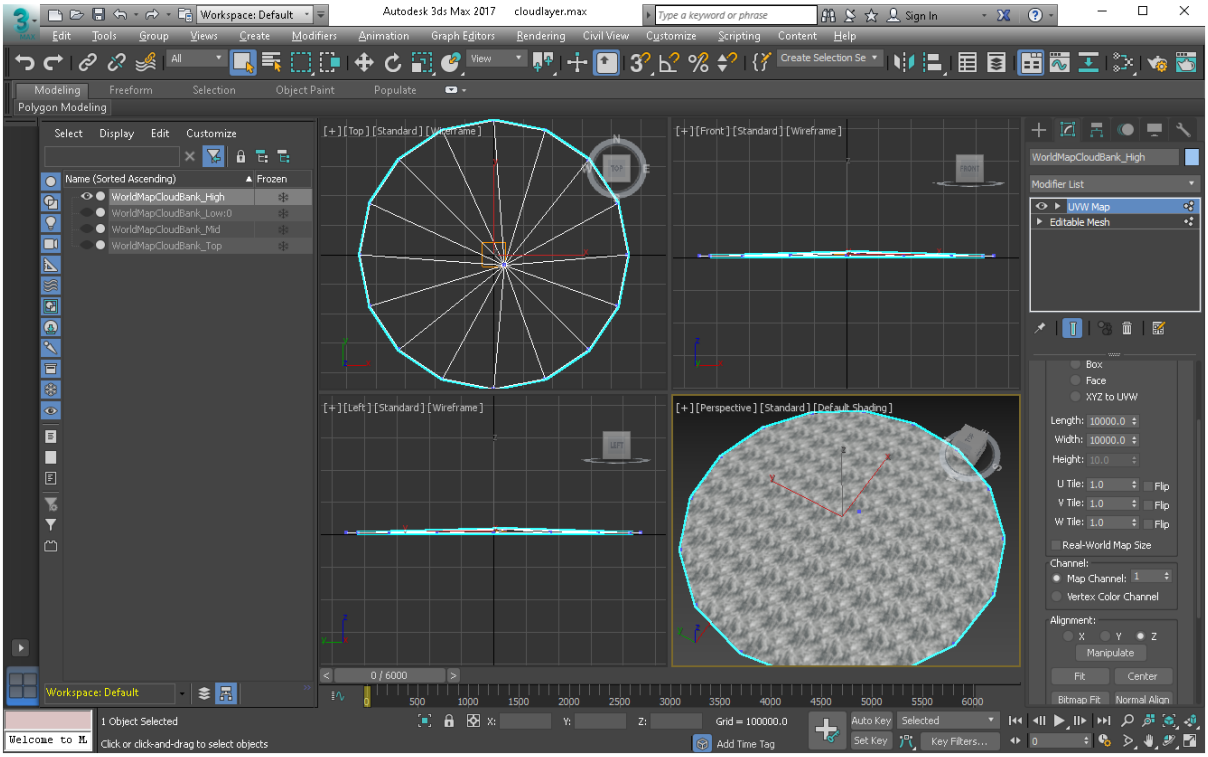


Figure 1021 - Setting length and width.

To export our changes, go to Export.

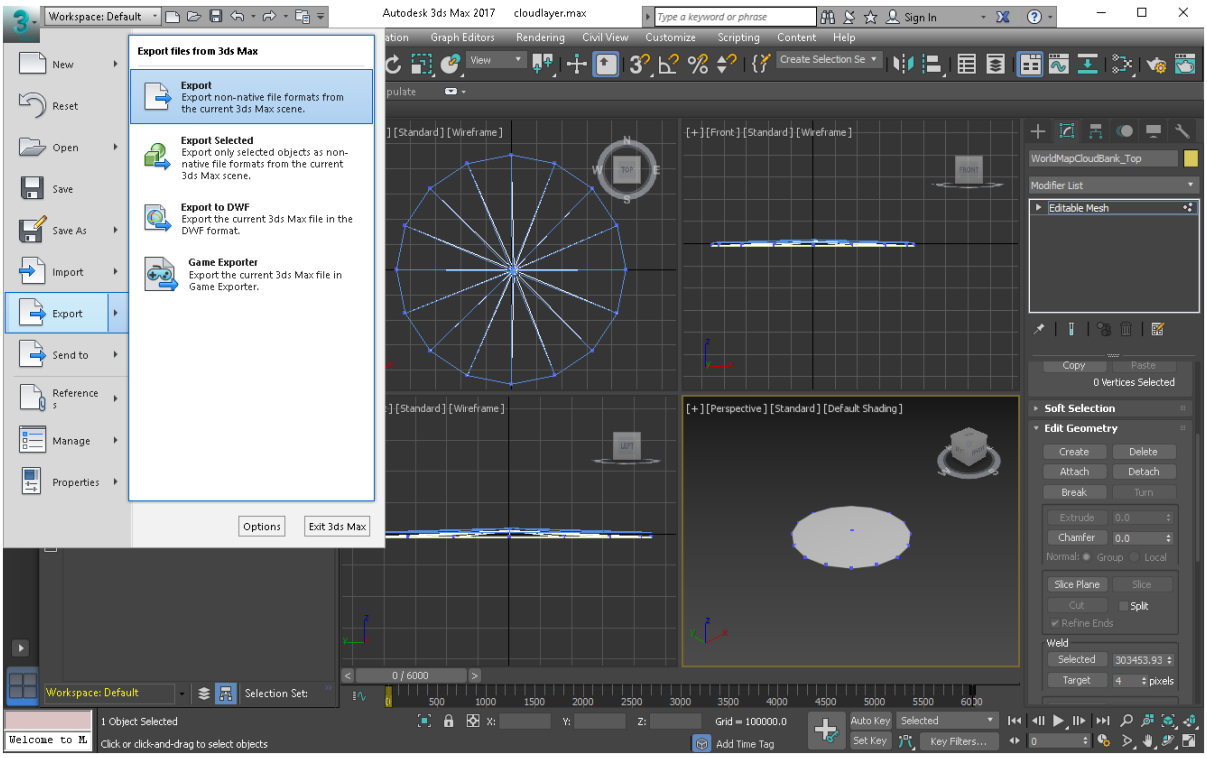


Figure 1022 - Exporting our changes to a new .nif file.

Set the location you want to save the .nif to and set 'Save as type' to 'NetImmerse/Gamebryo (*.KF, *.NIF)'. In my example I just set the file name to testworldmapcloudbank_export.nif.

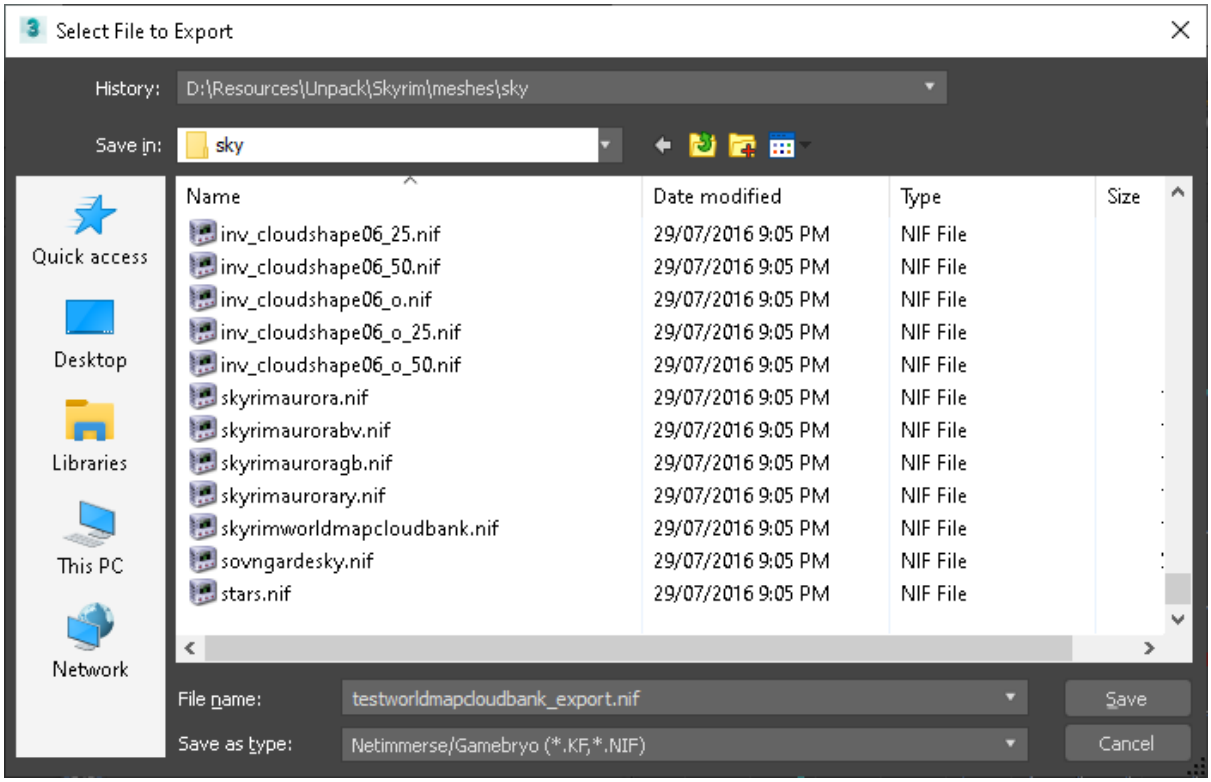


Figure 1023 - Setting the file type and file location.

Click Save.

Make sure Game is set to Skyrim.

Set the Root Node Type to BSFadeNode.

Click Export.

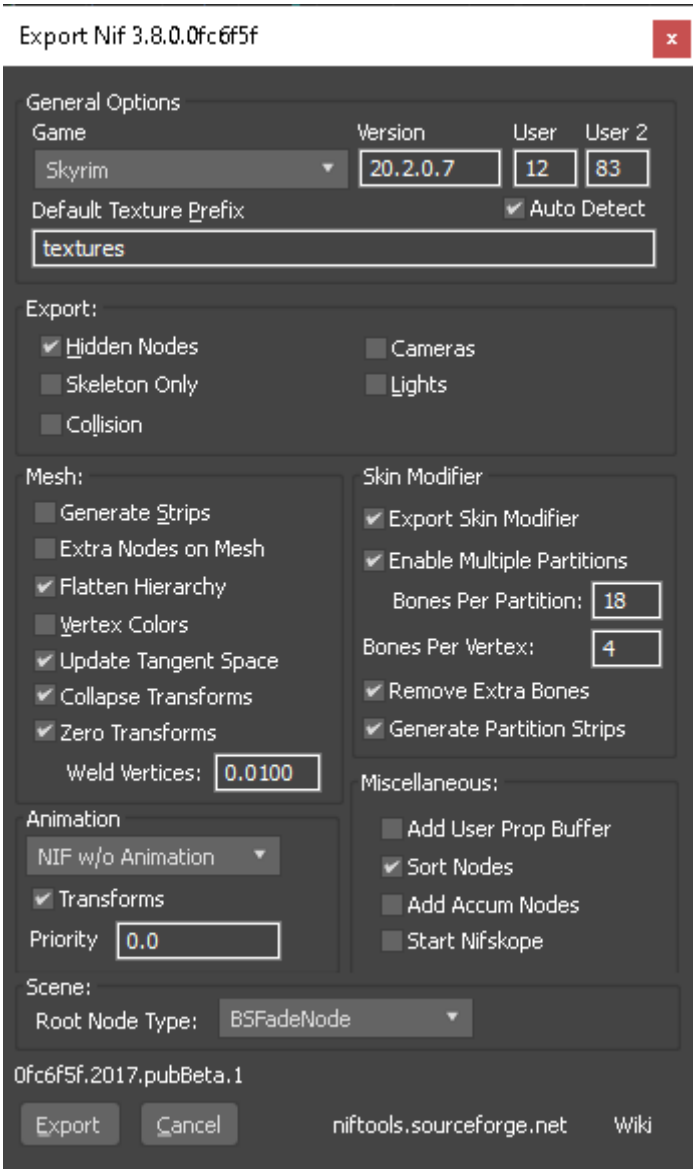


Figure 1024 - Configuring export settings.

Now we can close out of 3ds Max.

If we open our exported .nif in NifSkope, it'll look like this:

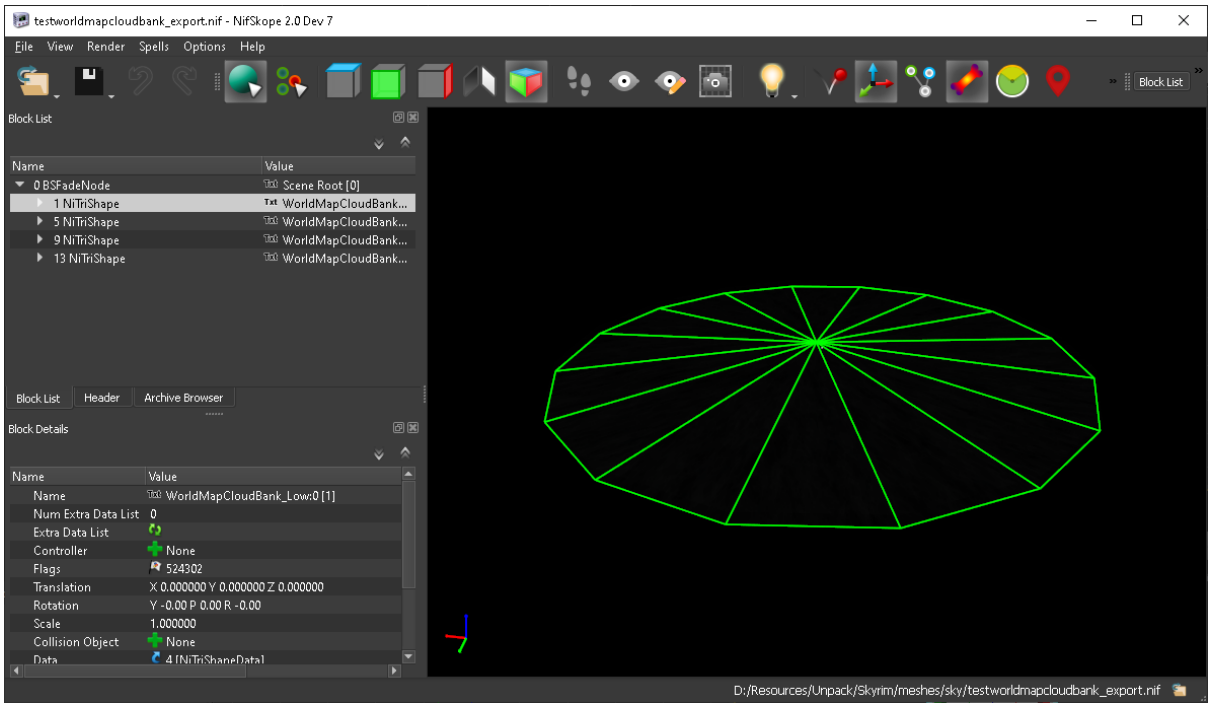


Figure 1025 - Our exported .nif in NifSkope.

It doesn't look like the exporter is able to create a few blocks we need, such as the NiAlphaProperty block. There are also some shader flags that weren't set, but we can fix this with NifSkope.

The easiest way to rectify this situation is to make a copy of the original skyrimworldmapcloudbank.nif.

For my example, I renamed it to testworldmapcloudbank.nif.

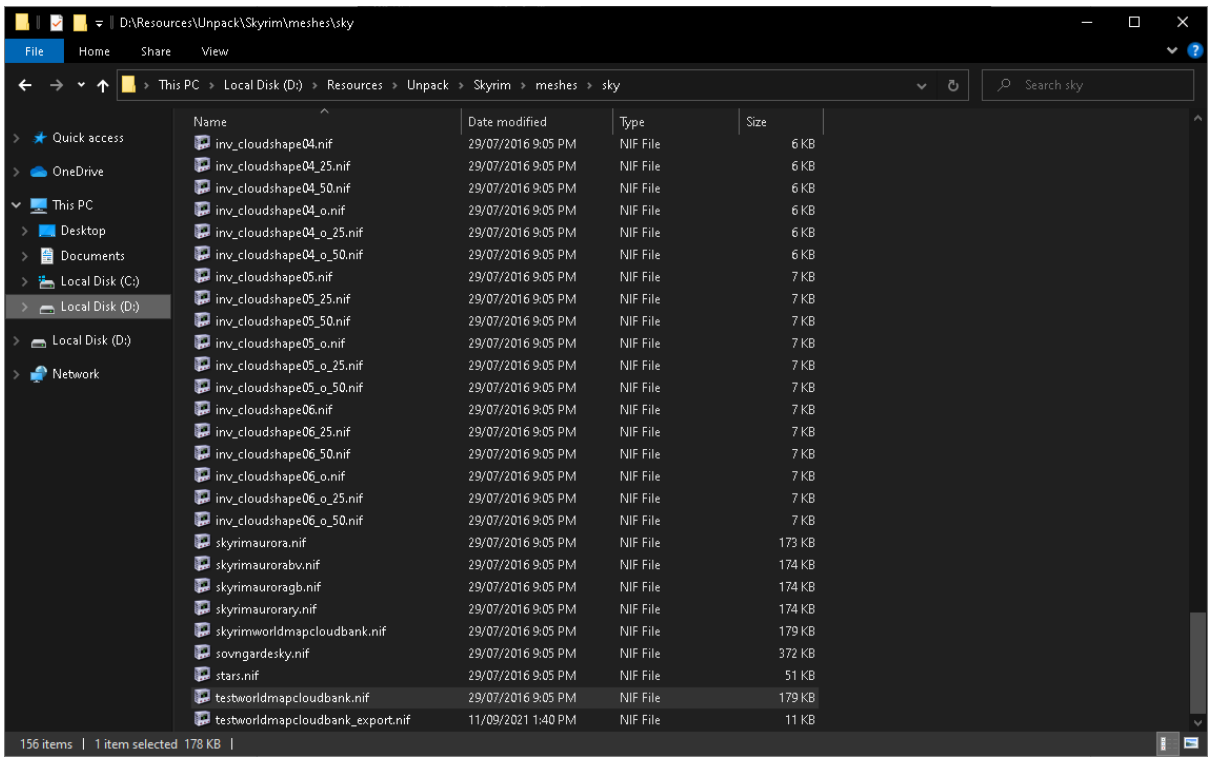


Figure 1026 - Made a copy of the skyrimworldmapcloudbank.nif.

Alright, open both testworldmapcloudbank.nif and testworldmapcloudbank_export.nif in NifSkope.

In testworldmapcloudbank_export.nif, navigate to 0 BSFadeNode > 13 NiTriShape > 16 NiTriShapeData.

Right-click on this block and go to Block > Copy.

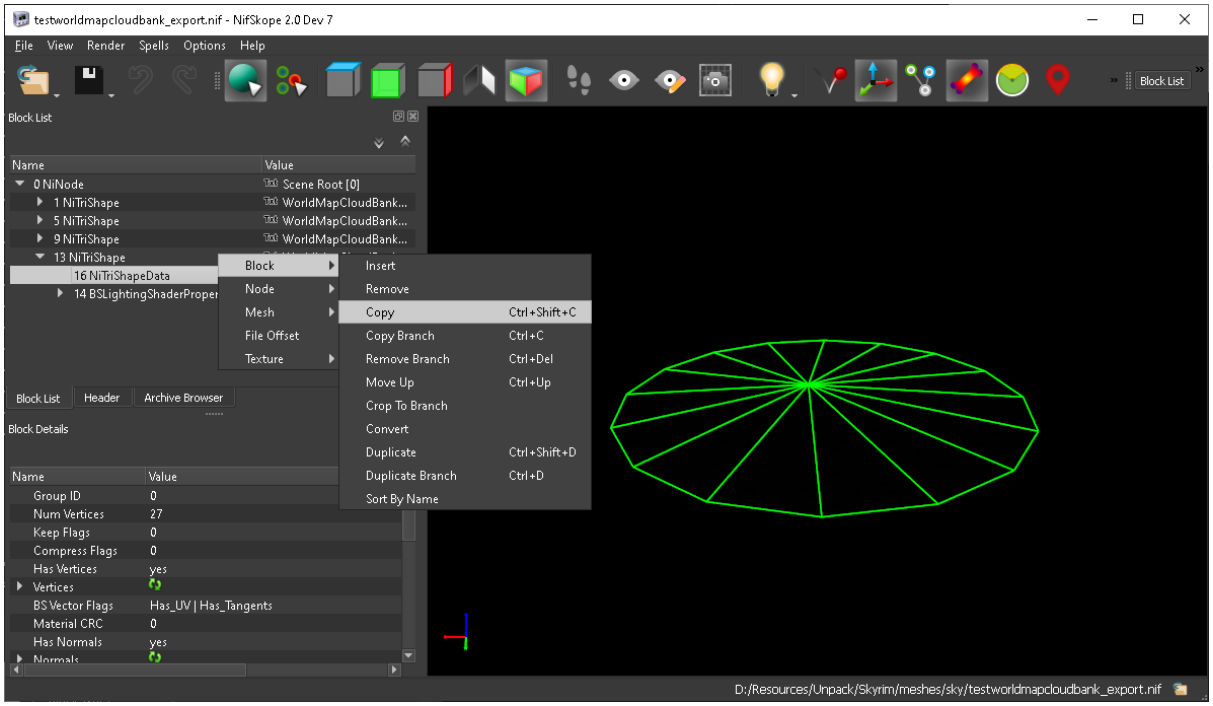


Figure 1027 - Copying the NiTriShapeData block.

Go to testworldmapcloudbank.nif, navigate to 0 BSFadeNode > 2 NiTriShape > 3 NiTriShapeData.

Right-click on this block and go to Block > Paste Over.

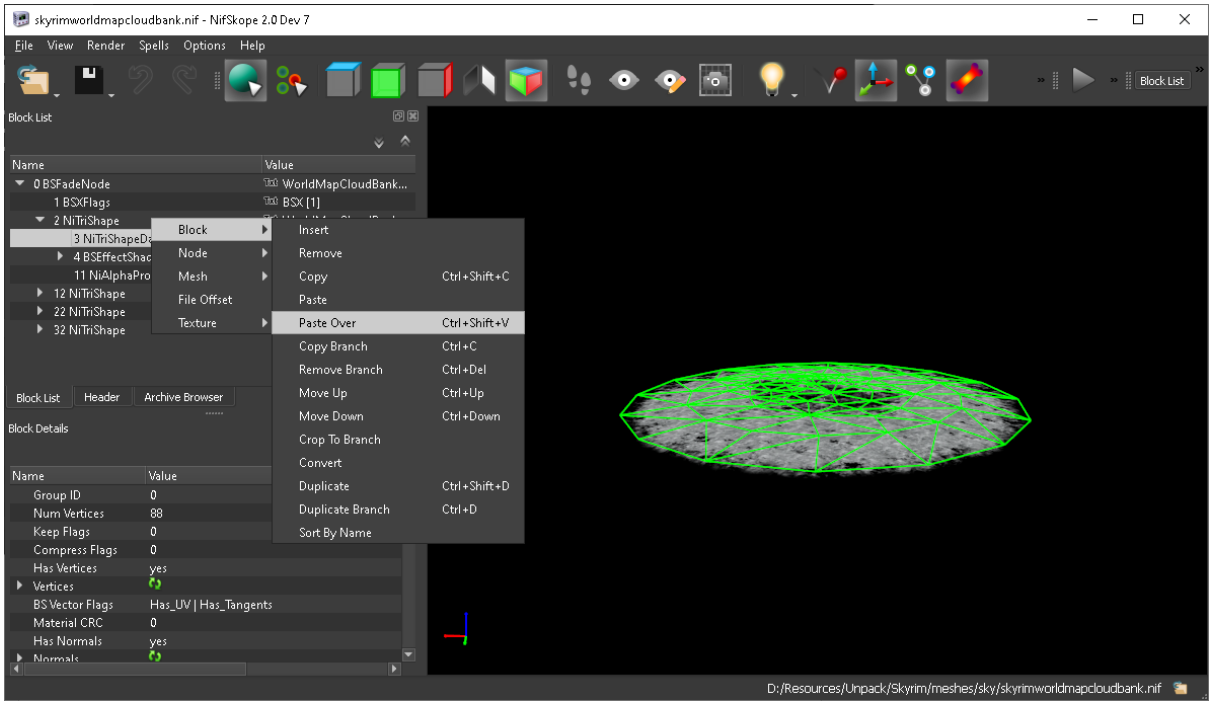


Figure 1028 - Pasting NiTriShapeData from one .nif to another.

It'll appear enormous in the viewport.

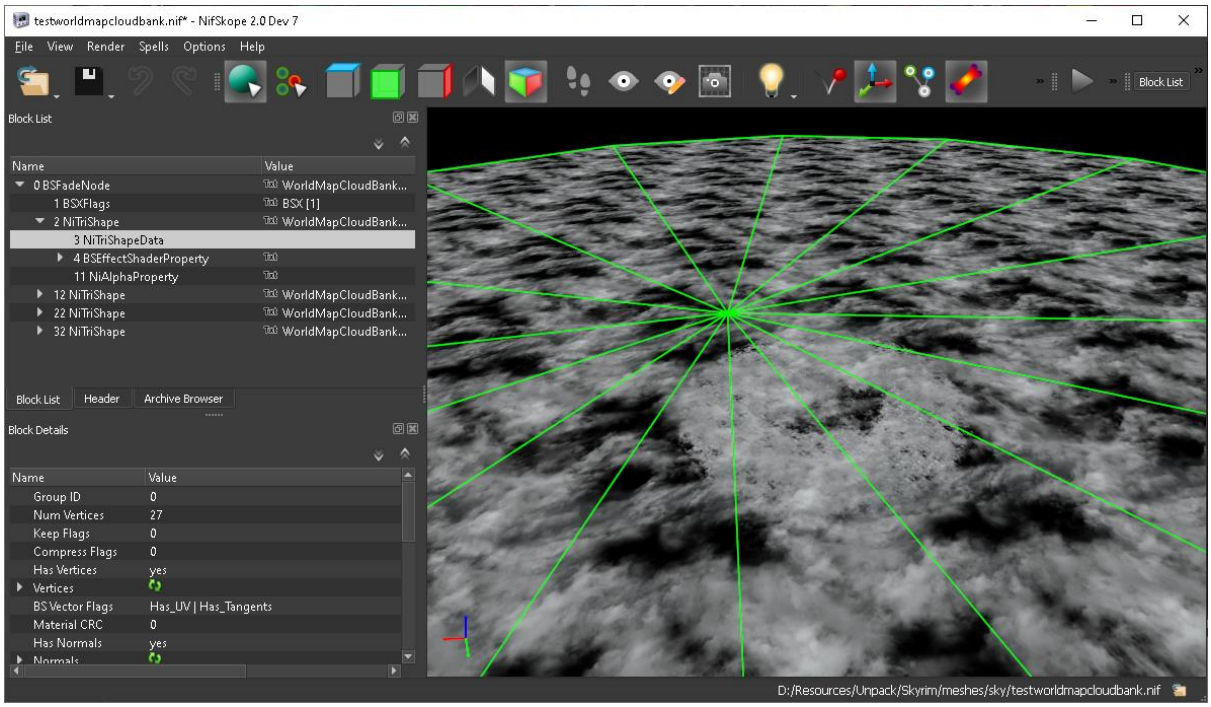


Figure 1029 - Replaced the existing NiTriShapeData block with the one from our export.

To fix that, click on the NiTriShape block above it and in the properties set the Scale to 1.000000.

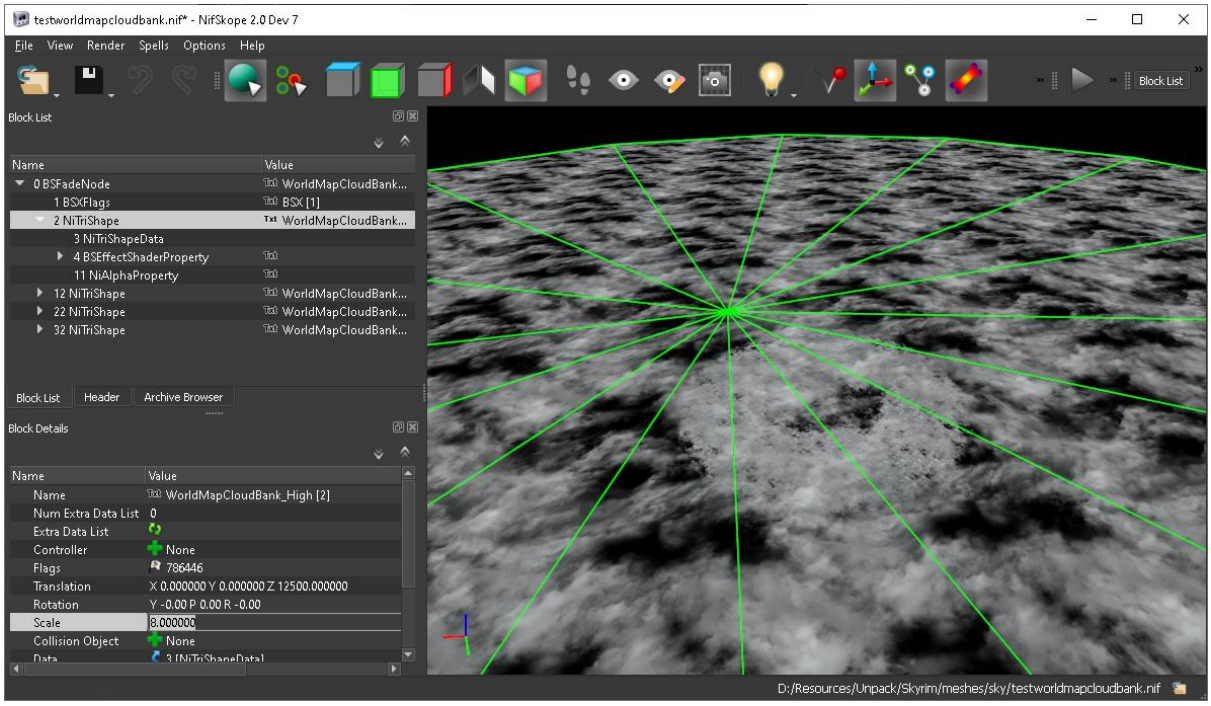


Figure 1030 - Changing the scale of the NiTriShape block.

It's now a more reasonable size.

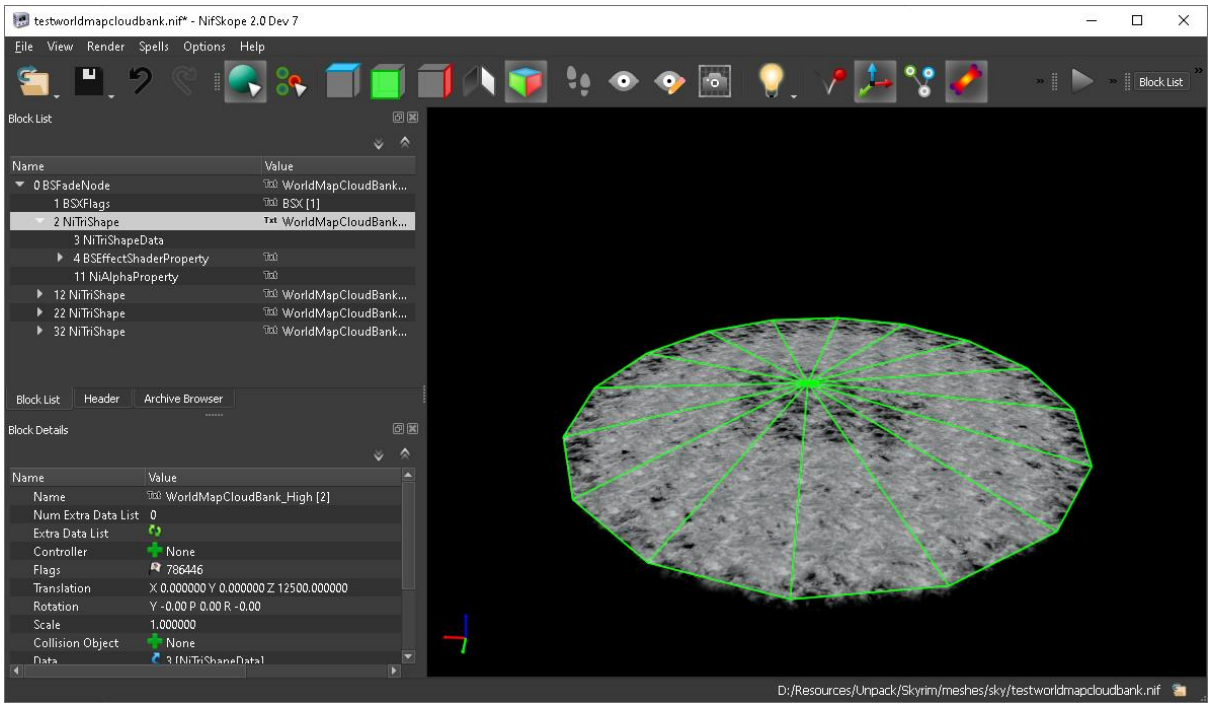


Figure 1031 - Resized the NiTriShape node.

Repeat these steps for the other three NiTriShapeData blocks.

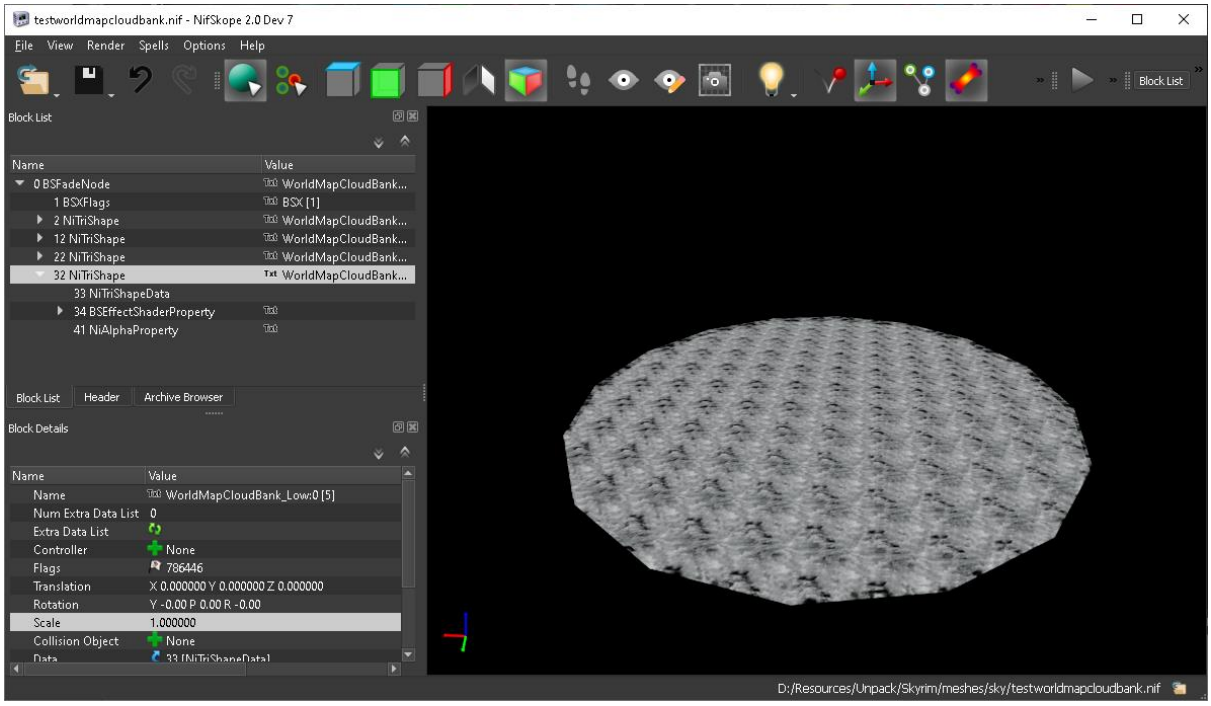


Figure 1032 - All four NiTriShapeData blocks replaced.

To break up that repeating pattern, I added some rotation to each NiTriShape block.

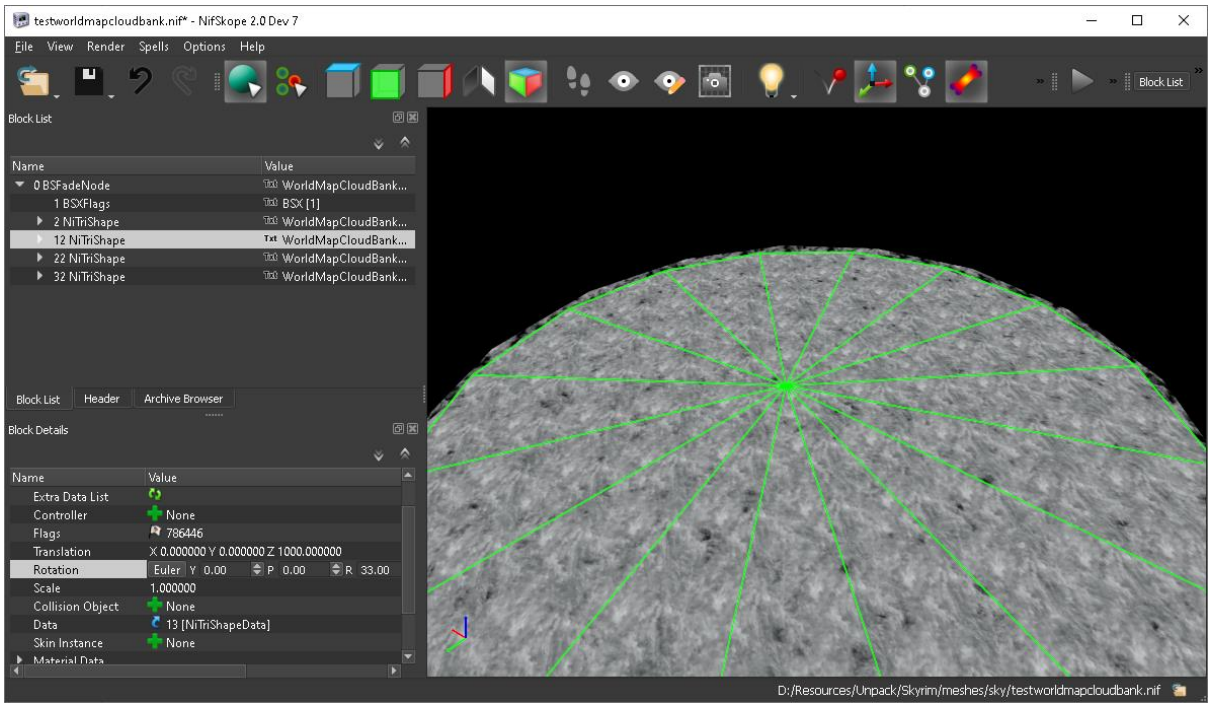


Figure 1033 - Adding rotation.

Double-click on the values next to Rotation and set R to a random number. In my example I set R to 33.00.

Repeat this step for each NiTriShape block.

Let's not completely cover our map with clouds. To clear up the central part of the plane, we'll need to modify the vertex colours.

Browse to 0 BSFadeNode > 2 NiTriShape > 3 NiTriShapeData.

In the properties, expand Vertex Colours.

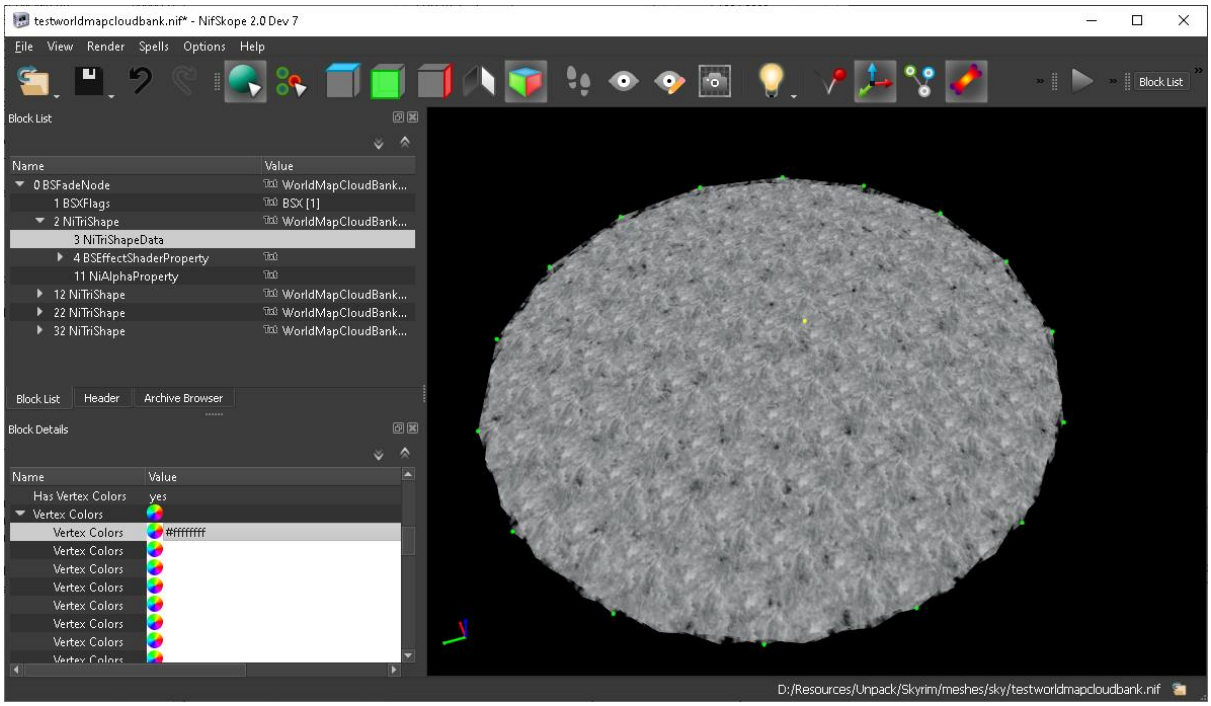


Figure 1034 - Vertex colours.

Vertices are green. The currently selected vertex is yellow. Browse down the Vertex Colours list until the central vertex is highlighted yellow.

Double-click on the value to the right to edit it. Currently it shows #ffffffff.

Set the 'A' value to 0.000. This will make the currently selected vertex transparent.

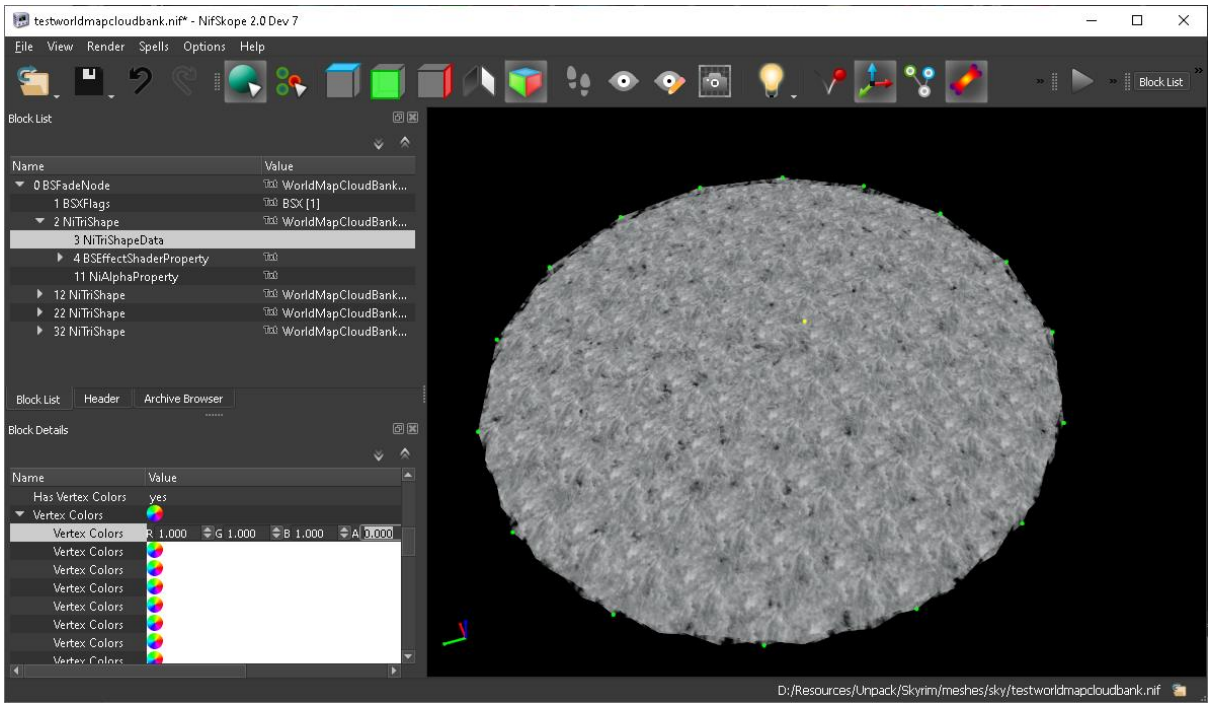


Figure 1035 - Setting the alpha to 0 for the currently selected vertex.

Go down through the list and set the alpha to 0 for the other Vertex Colour entries that highlight the central vertex.

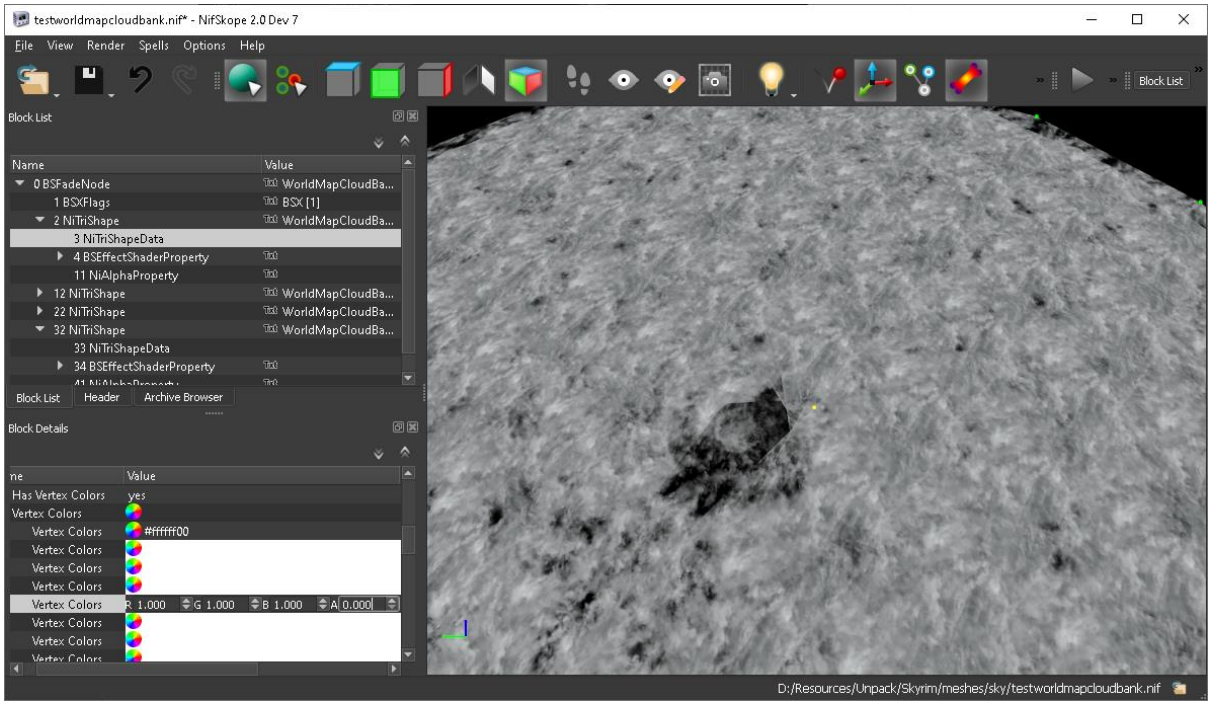


Figure 1036 - Changing the next vertex.

Repeat these steps for the other three NiTriShapeData blocks until all the central vertices for all four layers have their alpha set to 0.

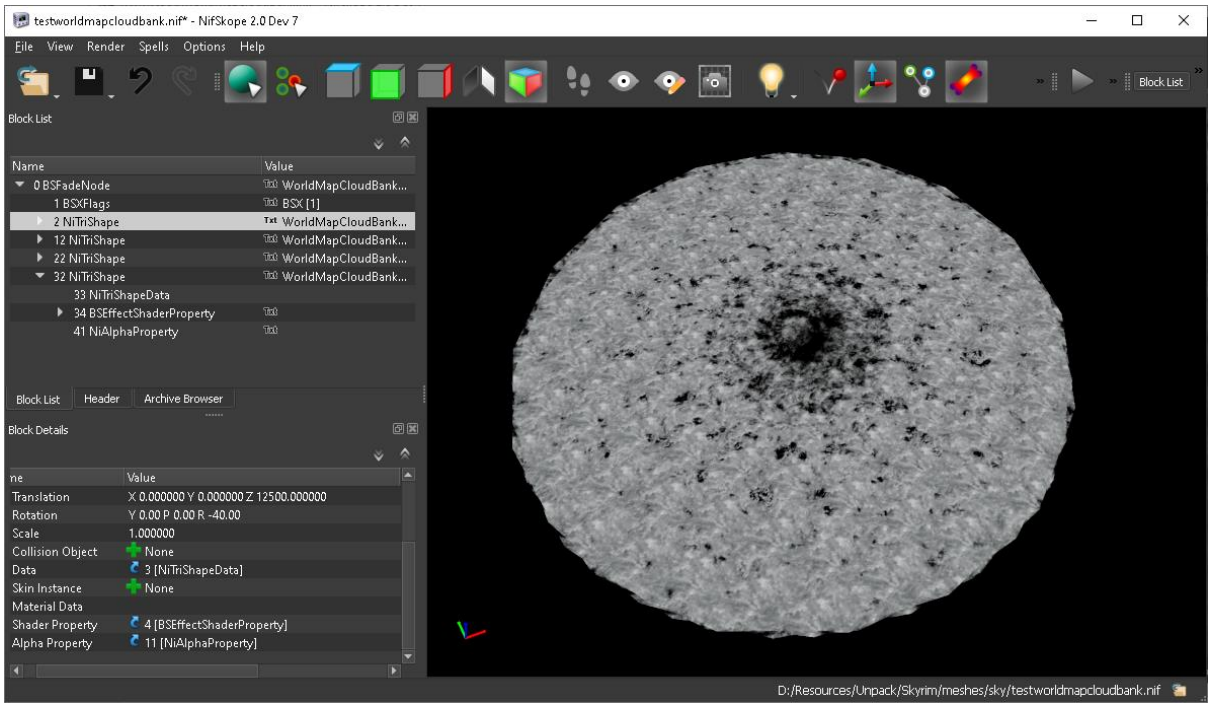


Figure 1037 - A hole in the clouds.

We should now have a hole in the clouds near the centre.

The rest of the cloud plane looks a bit thick, so I'm going to go back at reduce the alpha for the remaining vertices for all four NiTriShapeData blocks as well.

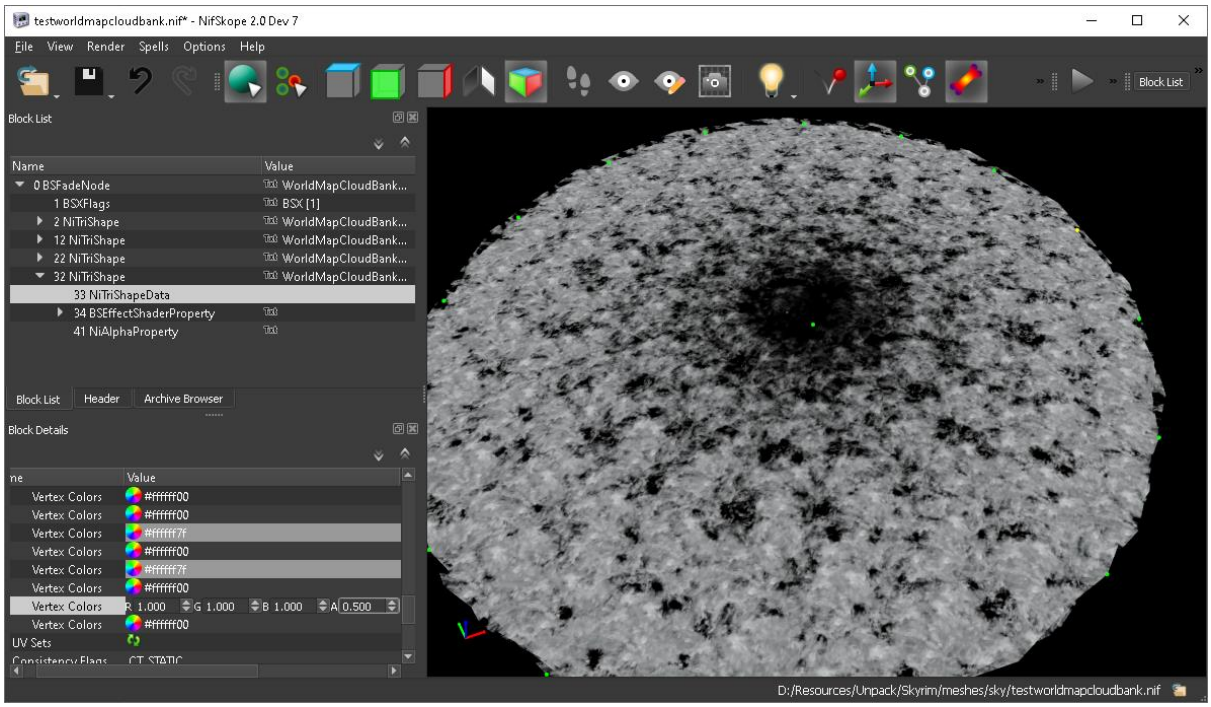


Figure 1038 - Reduced the alpha on the other vertices.

Alright, this looks good enough to try out.

Go to File > Save As.

Save .nif to your Skyrim\Data folder. I'm saving my .nif to Skyrim\Data\Meshes\sky.

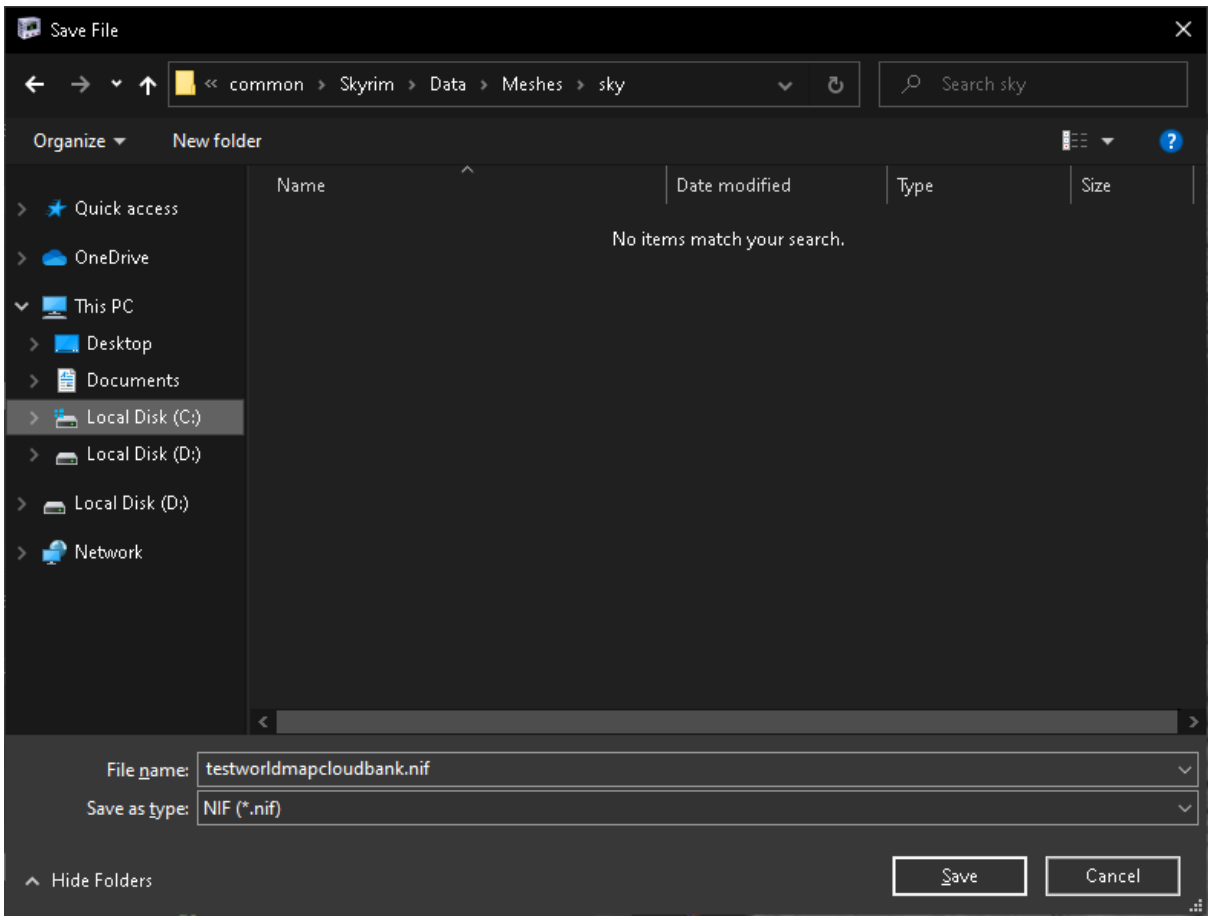


Figure 1039 - Saving our cloud plane to the Skyrim data folder.

Our world space is using testworldmapcloudbank.nif, so the testworldmapcloudbank_export.nif exported from 3ds Max can now be deleted as we don't need it anymore.

Back in the Creation Kit, go to World > World Spaces.

Highlight your world space in the list and click on the Edit button next to Cloud Model.

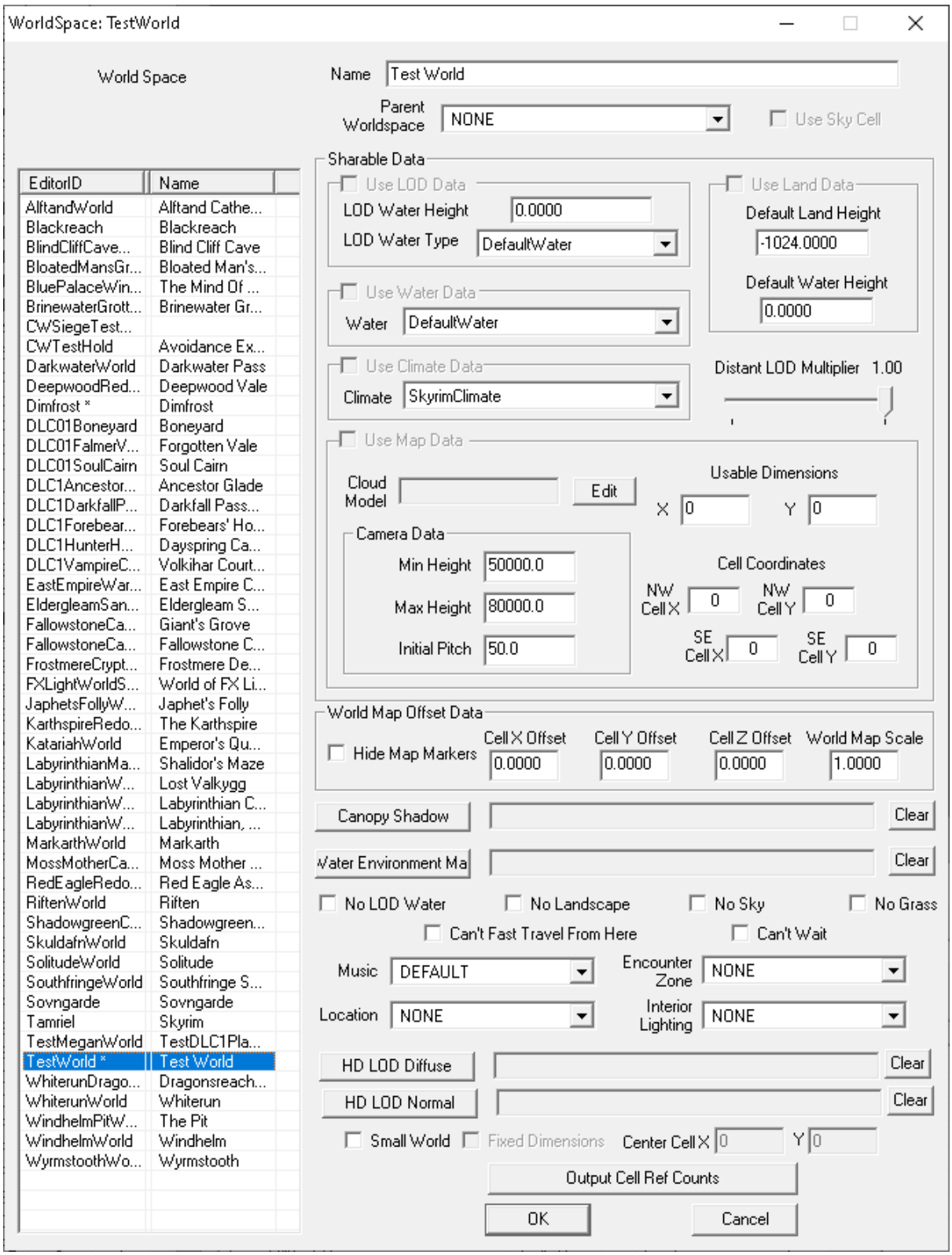


Figure 1040 - Adding a cloud model.

Select the cloud bank .nif you just saved.

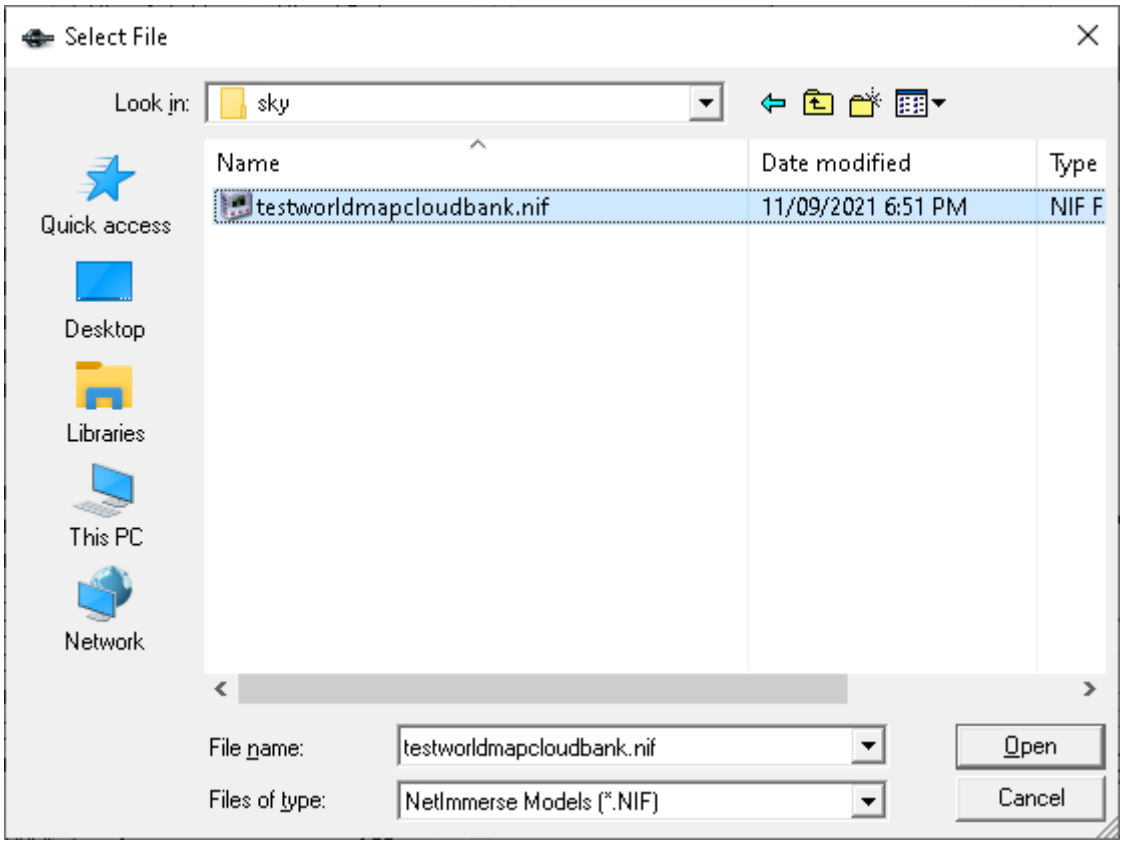


Figure 1041 - Selecting our cloud model.

Click OK to close out of world space properties.

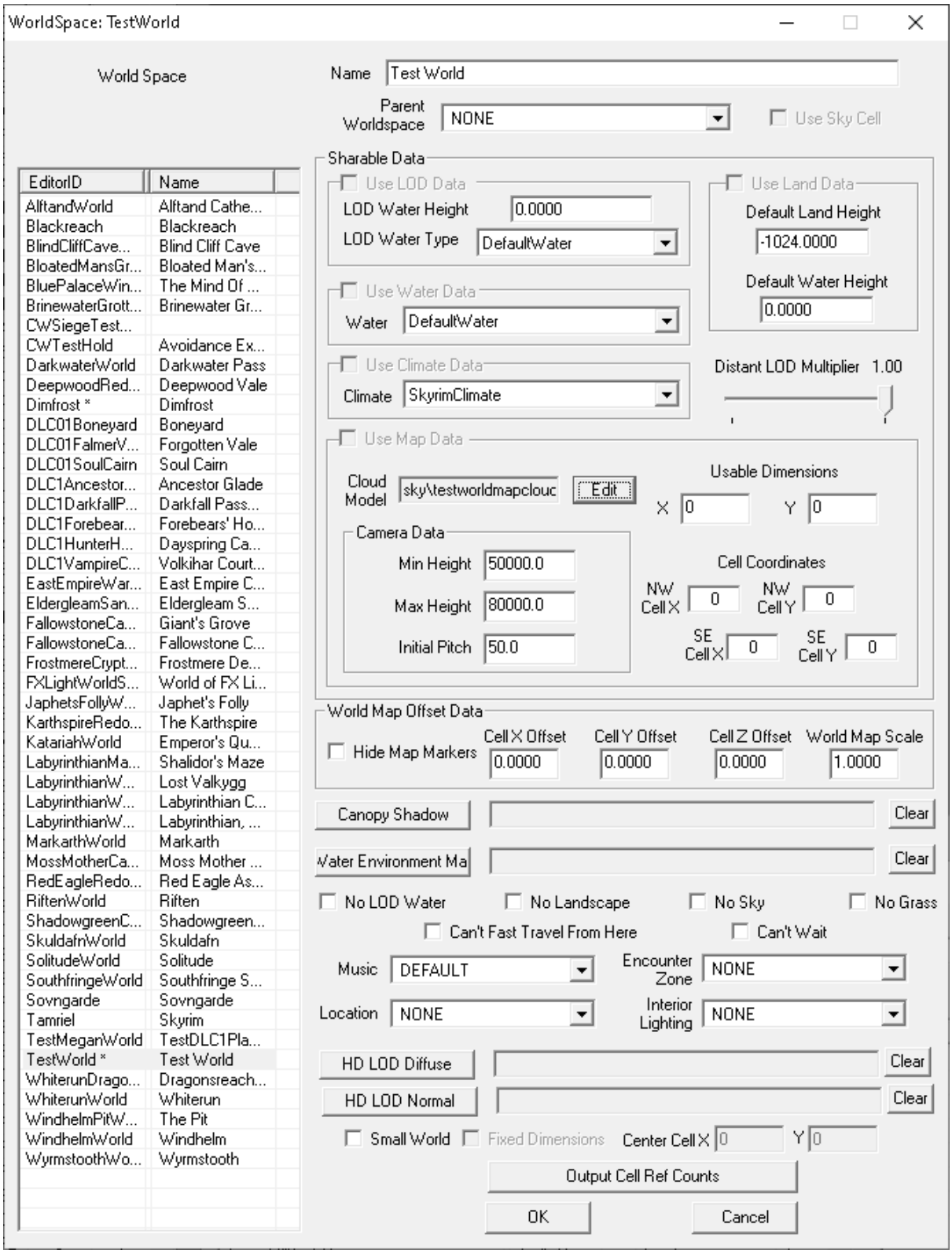


Figure 1042 - Cloud model added.

Now if we go in-game we'll be able to see our new cloud layer on the map screen. No more cloud seams. The clouds are also sparse near the centre of the map so we can see our world space clearly.



Figure 1043 - Test world map with custom cloud layer.

Lastly, let's look at a couple extra things you might want to change.

If you have a default climate you want to add to your world space, you had add it in the Climate drop-down.

For Wyrmstooth, I created a WTWyrmstoothLocation for the WyrmstoothWorld world space and a WTDimfrostLocation for the DimfrostWorld world space for scripting purposes so I can check the player's current location.

One final thing you might want to do here is tweak the camera data. By Default, the Min Height, Max Height and Initial Pitch values are set up for the Tamriel world space.

For smaller world spaces, you should bring the camera a bit closer to the ground, especially if you have a lot of locations close to each other. Having the camera too high up might make it hard for players to select certain locations to fast travel to.

Note: It also makes the world space seem a bit larger than it actually is, but in my defence Bethesda did it too with the Solstheim world space! ☺

For Wyrmstooth, I set the Min Height to 32000, the Max Height to 50000 and the Initial Pitch to 60 which I think is close to how the Solstheim world space is set up in the Dragonborn DLC.

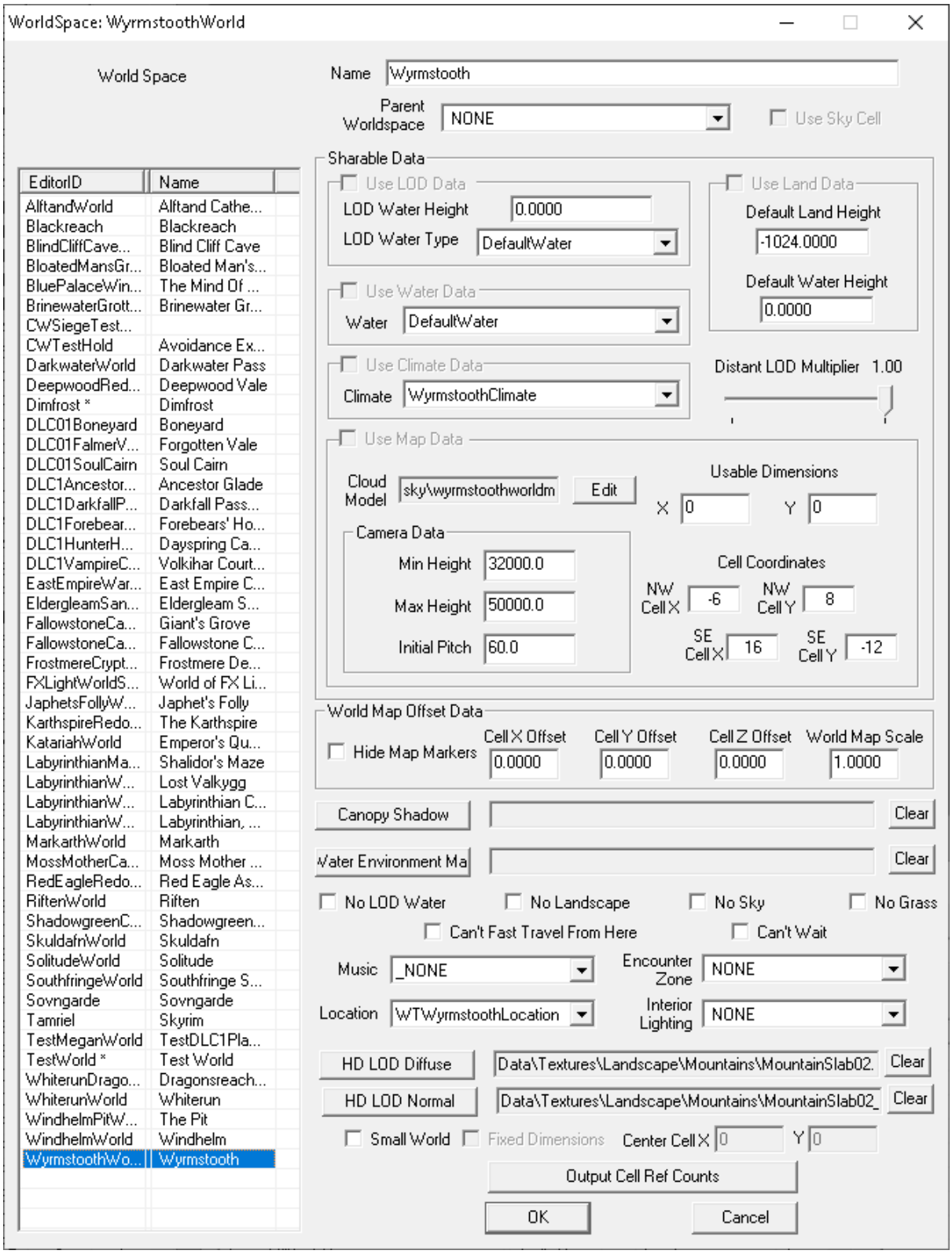


Figure 1044 - Setting the camera data values.

GENERATING MAX HEIGHT DATA

Max height data is used by dragons to prevent them from clipping through objects like mountains and buildings when flying around.

To generate max height data, go to World > Generate Max Height Data for World.

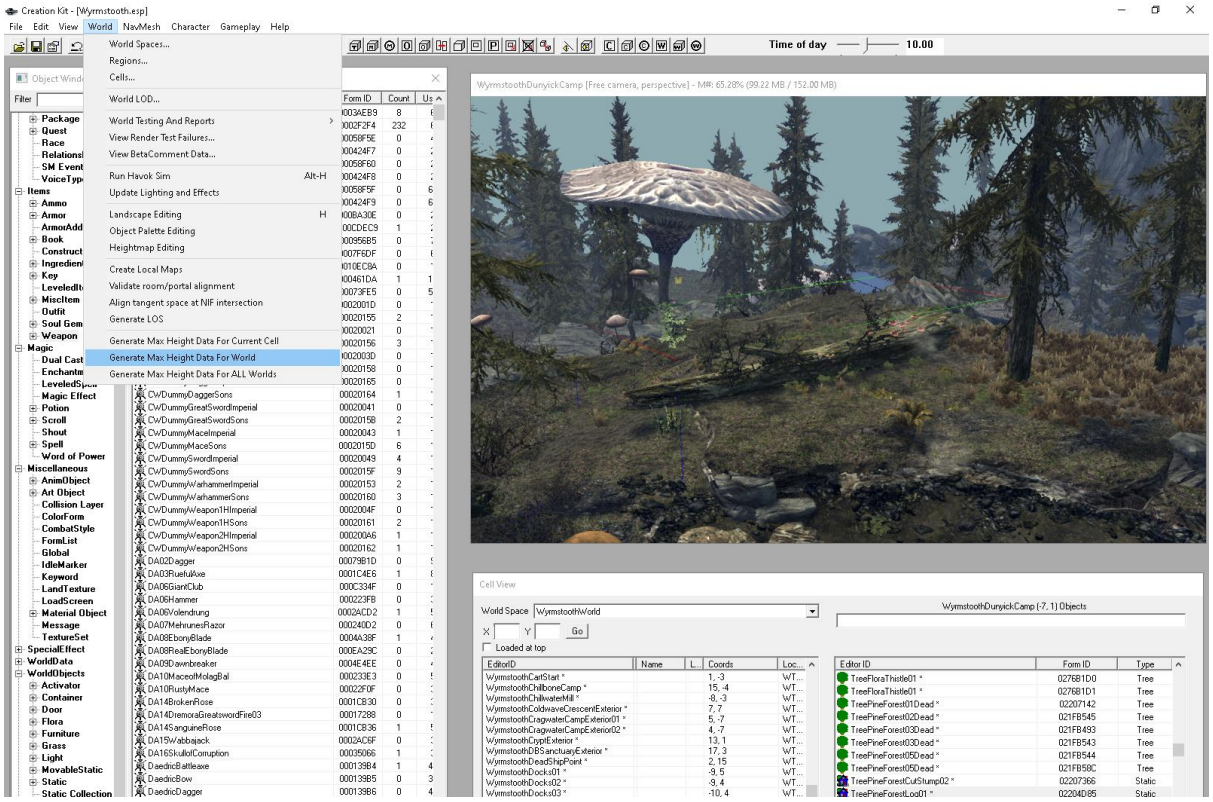


Figure 1045 - Generate Max Height Data.

Select your world space then click OK.

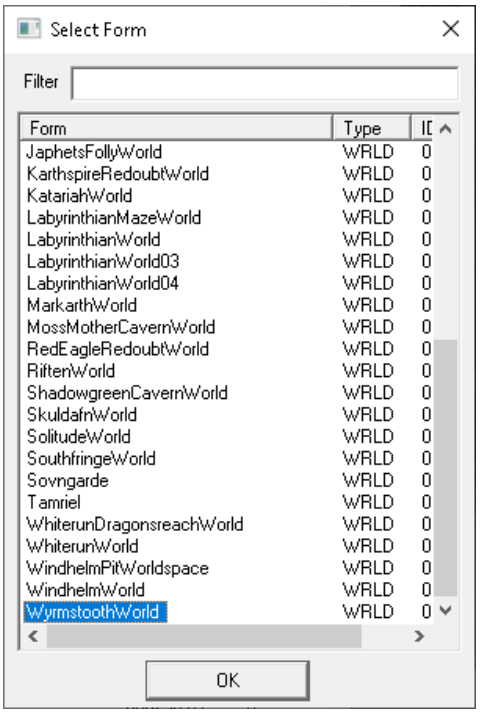


Figure 1046 - Generate Max Height Data world space selection.

Click Yes on the confirmation pop-up to begin the process.

Important: The Creation Kit will lock-up until the process is complete and that may take quite some time depending on the size and complexity of your world space.

Also be aware that this process may sometimes not complete at all. If it's been more than an hour and the Creation Kit is still locked up, use Task Manager to close CreationKit.exe and try again.

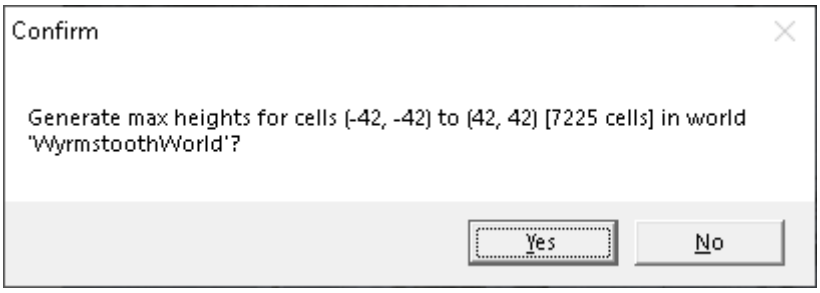


Figure 1047 - Generate Max Height Data world space confirmation.

If you have previously built Max Height Data for that world space, you'll also see the following pop-up window. I'd recommend choosing No to regenerate the data from scratch.

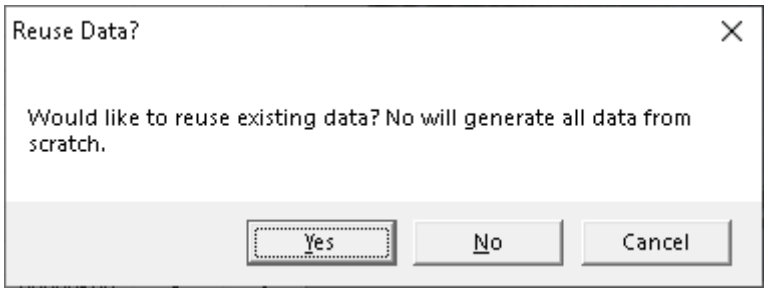


Figure 1048 - Generate Max Height Data reuse existing data confirmation.

If there are multiple world spaces in your mod, I'd recommend doing each one separately, saving in between as this procedure may also potentially crash the Creation Kit.



Figure 1049 - Done.

When the process is done, you'll see a pop-up simply saying Done. Click OK.

GENERATING TVDT OCCLUSION

This type of data is used to hide distant terrain models that are occluded by other distant terrain models to improve performance.

Important: While this data *can* be generated in the Creation Kit, from what I've seen it doesn't generate it accurately and will lead to LOD being occluded when it shouldn't be.

So we'll need a copy of the xLODGen beta.

Download and extract it to a folder. Make a copy of xLODGen.exe and rename it TES5Edit.exe/SSEEdit.exe.

Launch xEdit.

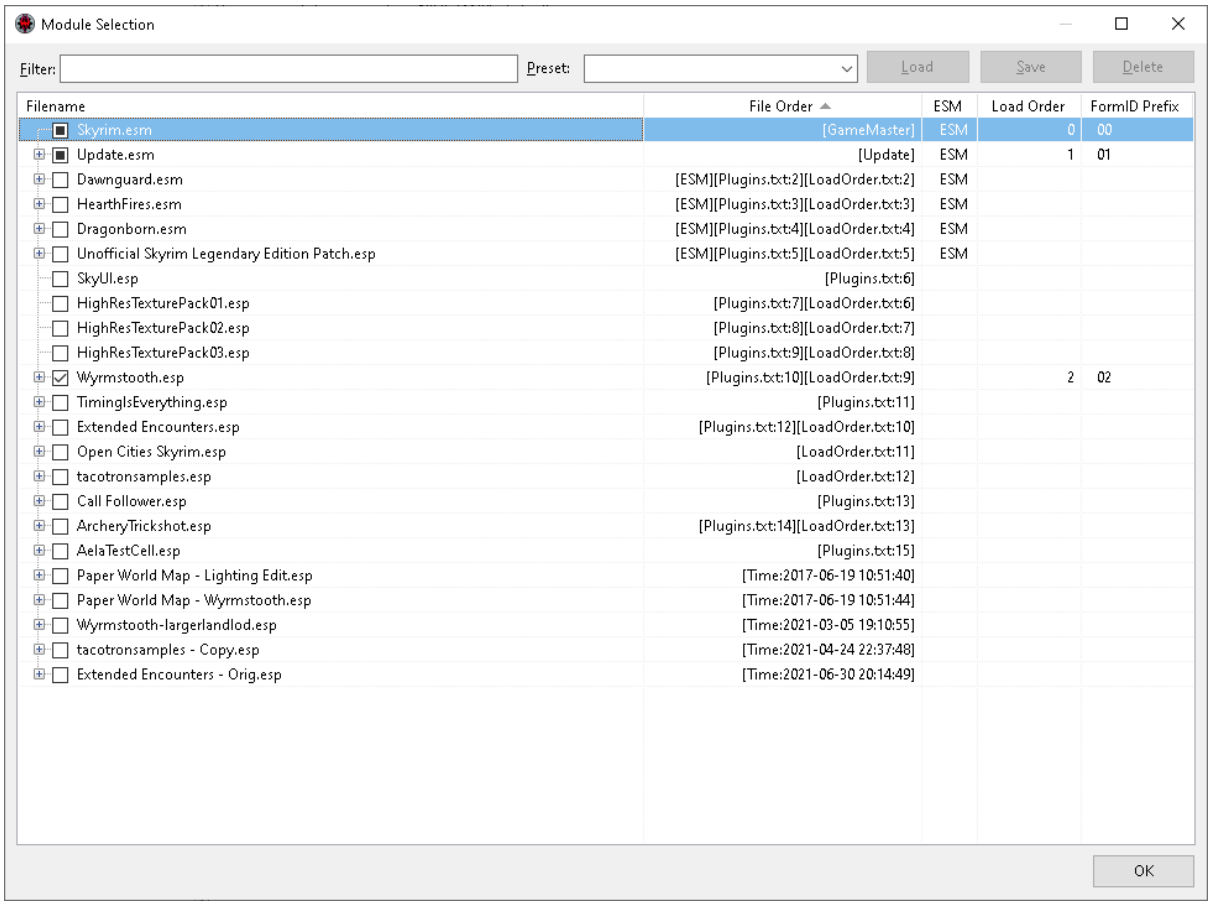


Figure 1050 - TES5Edit Module Selection.

Tick your mod and click OK.

Once the background loader has finished, right-click on your mod and select Other > Generate LOD.

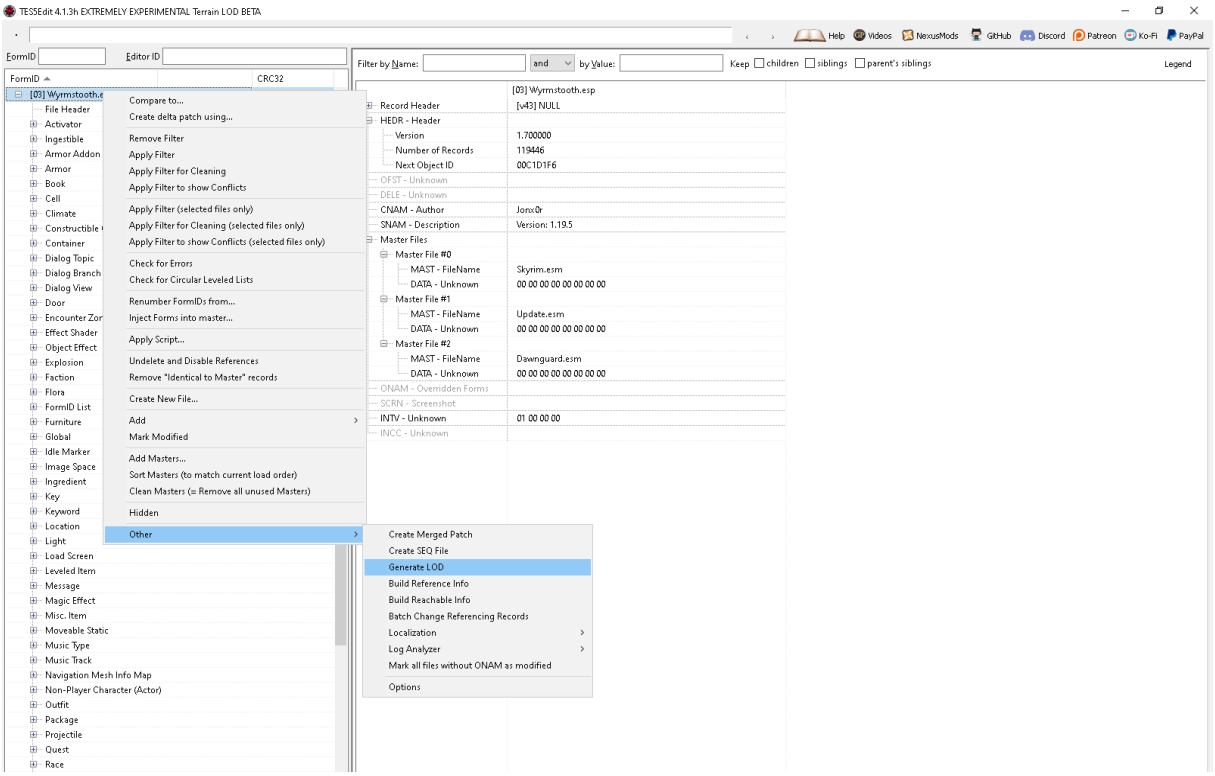


Figure 1051 - Generate LOD menu.

If Objects LOD, Trees LOD or Terrain LOD are ticked, untick them.

Select the world space you wish to generate occlusion for and tick Occlusion.

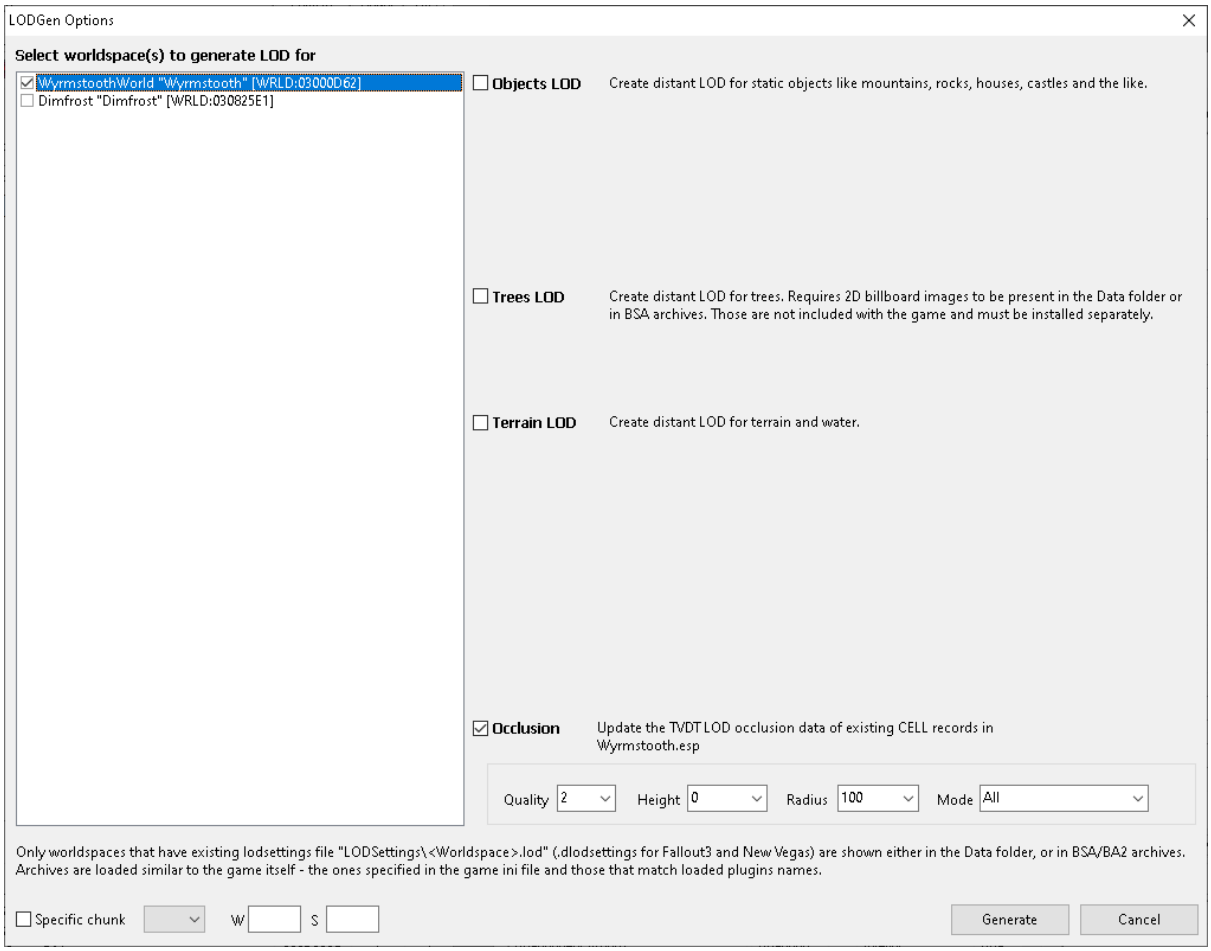


Figure 1052 - World space and occlusion selected.

Click Generate to begin generating occlusion data.

When the process is done, you'll see 'LOD generation complete' in the output.

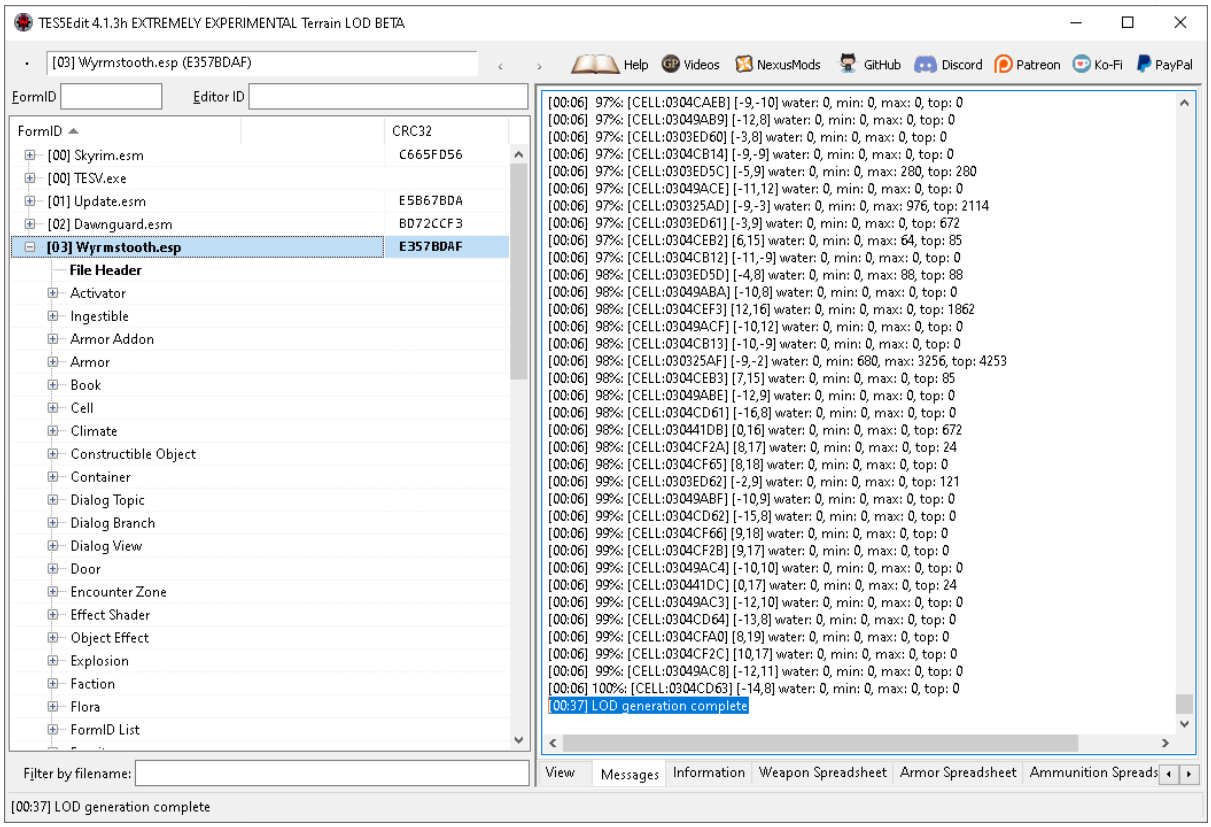


Figure 1053 - Occlusion generation complete.

Go to the Burger Menu at the top left and select Save, or press CTRL + S to save the changes to your mod.

Below is an example of TVDT occlusion at work.

In this screenshot, I used the 'tfc' console command to detach the camera from the player character and took the screenshot from high in the air looking down at a low-lying area that was obscured behind a mountain.



Figure 1054 - LOD objects being occluded.

As you can see, the landscape and object LOD in occluded cells are not being rendered.

Once the player moves into a position where they'll have line of sight to the cell, the objects within it will be rendered.

From the player's perspective, this process of hiding and un-hiding objects should never be noticeable.

GENERATING GRASS .GID FILES

By default, grass in your world space won't show up in-game unless you've either generated .gid files for each cell containing grass, or have bAllowCreateGrass=1 added to the [Grass] section of your Skyrim.ini file.

The process of generating .gid files is a bit… different to generating LOD.

To generate the .gid files, open your Skyrim.ini file, go to the [Grass] section and add bGenerateGrassDataFiles=1.



Figure 1055 - Skyrim.ini file.

Now launch Skyrim, teleport to your world space, and walk through every cell containing grass. The .gid files are created as each cell loads, so you'll need to do a bit of walking. ☺



Figure 1056 - Generating .gid files.

I would recommend using the console command 'player.setav speedmult 300' to speed up this process.

Once done, you should have a Grass folder in your Skyrim Data directory filled with .gid files prefixed with the name of your world space.

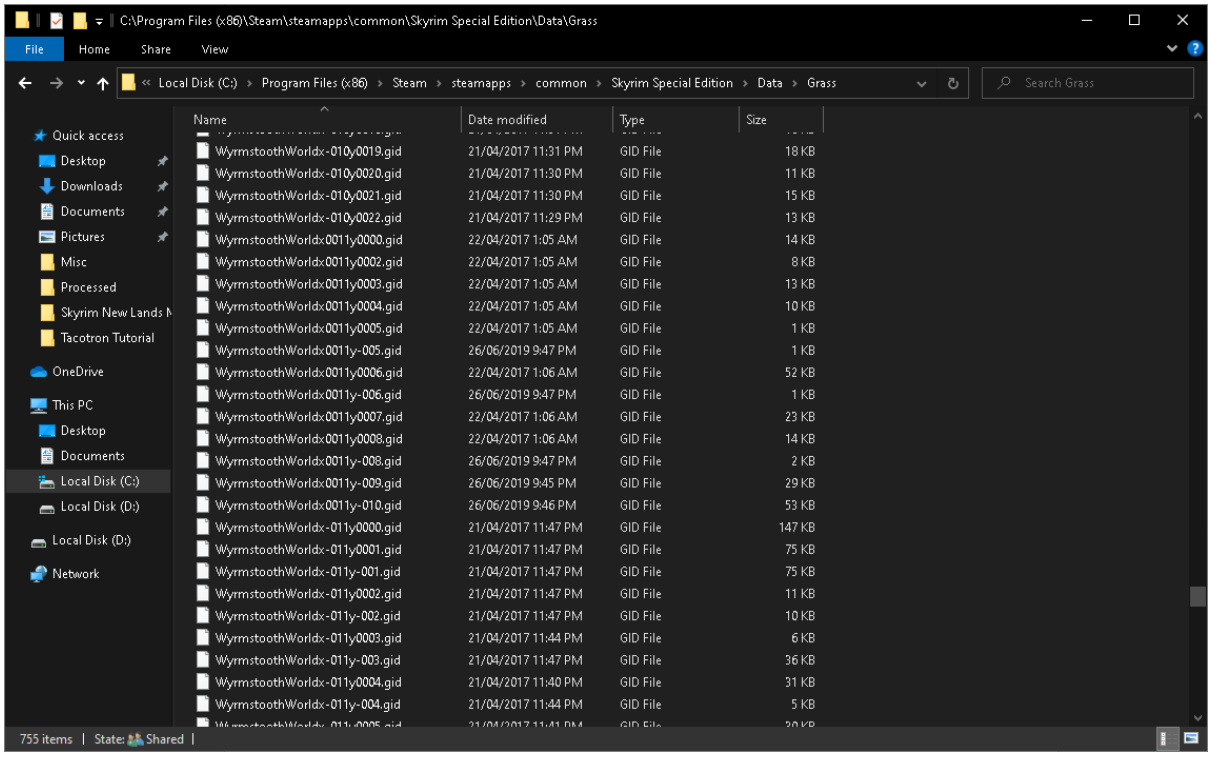


Figure 1057 - Grass folder.

Don't forget to remove bGenerateGrassDataFiles=1 from your Skyrim.ini file when you're done.

CHAPTER 8: ASSET PIPELINE

IMPORTING ASSETS FROM SKYRIM IN 3DS MAX

This section will cover importing an asset from Skyrim into 3ds Max to tweak or remodel.

For this section I'll be working with 3ds Max 2017. The NIF Export/Import Plugin for 3ds Max 2017 is available on Nexusmods. For other versions of 3ds Max, try NIF Plugin for 3ds Max 2015-2018 available on Nexusmods.

You can download a trial version of 3ds Max from the Autodesk website.

The first thing you need to do is export the .nif file from the .bsa archive. See the section on Unpacking the BSA Archives for step by step instructions on how to do this.

In this example, I'm going to be importing winebottle01a.nif into 3ds Max. This asset was exported from the Skyrim - Meshes.bsa file and is located under meshes\clutter\wine.

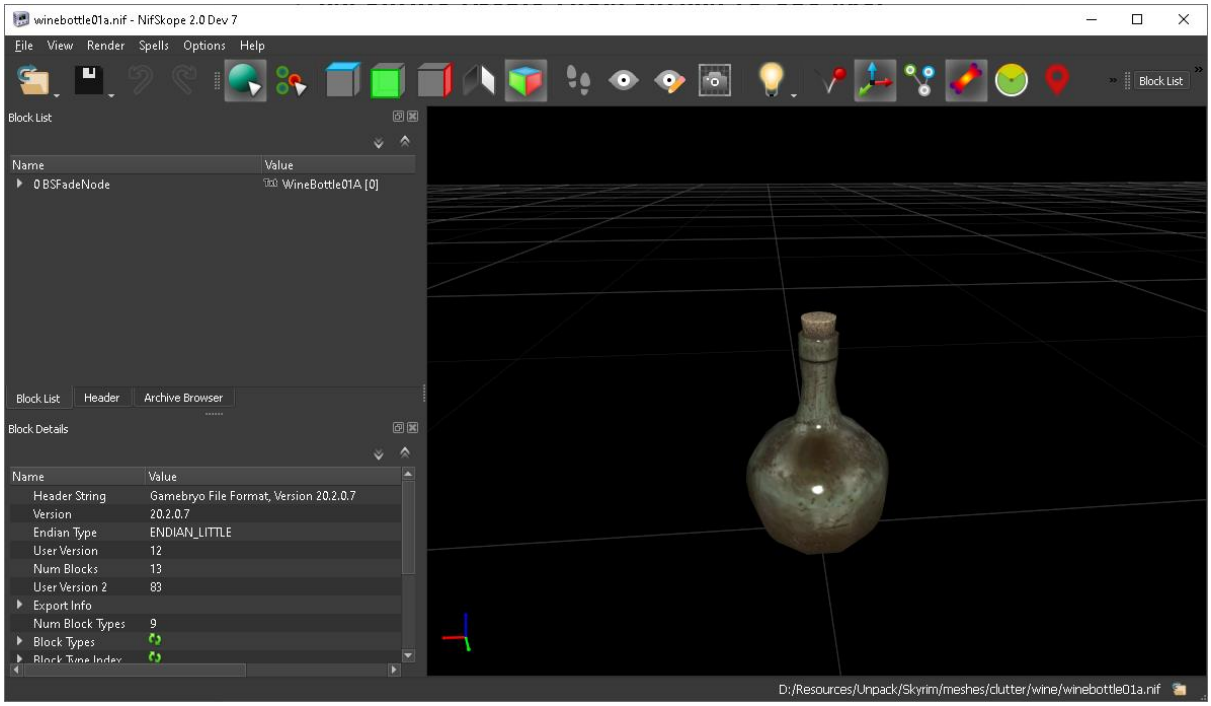


Figure 1058 - winebottle01a in NifSkope.

In 3ds Max, go to Import.

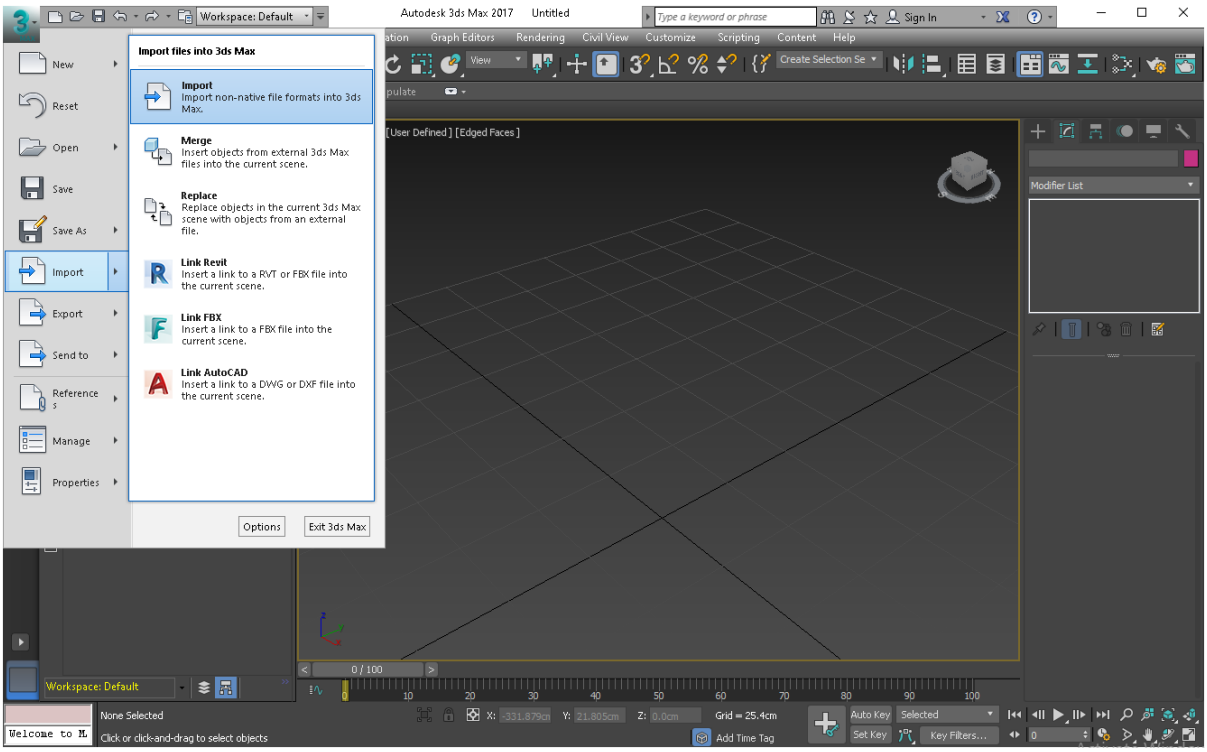


Figure 1059 - Importing an external asset.

Set the 'Files of type' to 'NetImmersse/Gamebryo (*.NIF, *.KF)'. Select the .nif file to import and click Open.

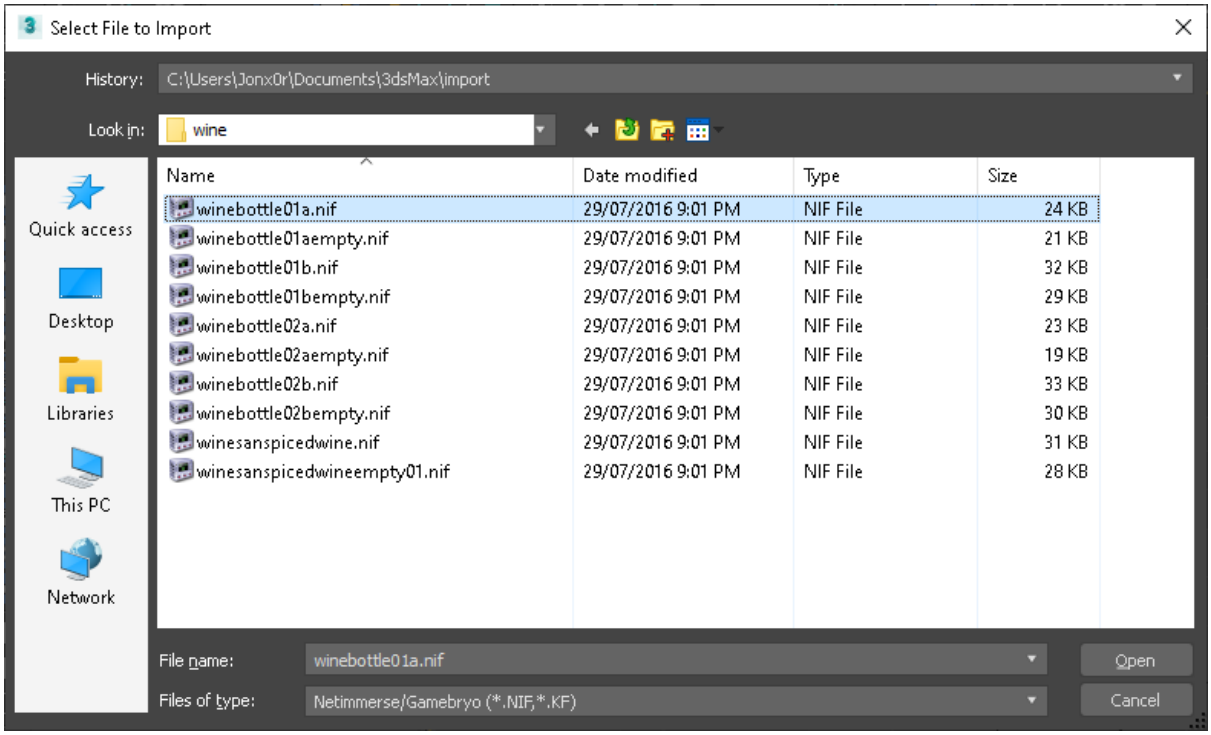


Figure 1060 - Selecting the .nif file to import.

Set the Game to Skyrim.

Under Import, tick Collision. The object doesn't have any lights or cameras, so they don't need to be ticked.

This particular object doesn't have any animation, so untick everything under the Animation section.

Under Geometry, make sure Vertex Colours is ticked. This particular object doesn't use the Skin Modifier so we can untick that checkbox. I also don't want to make any alterations to the model so I unticked 'Auto Smooth Mesh', 'Remove Illegal Faces' and 'Weld Vertices'.

Under Miscellaneous, tick 'Flip UV'. Make sure 'Render Textures in View', 'Ignore Root Node' and 'Use Niftools Shader' are also ticked.

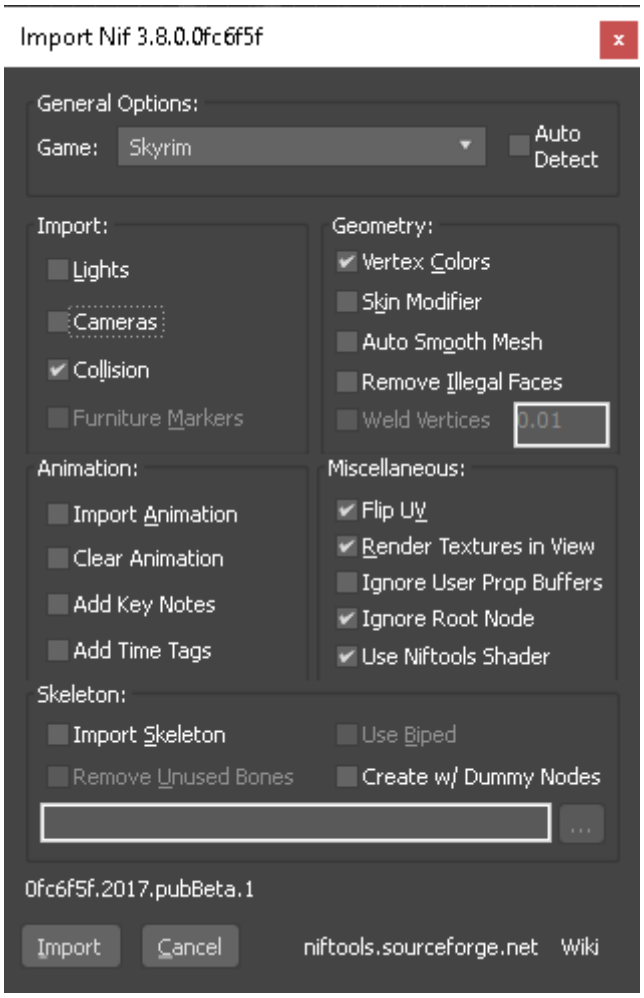


Figure 1061 - NIF import settings.

Click on the Import button.

And there's our wine bottle imported into 3ds Max.

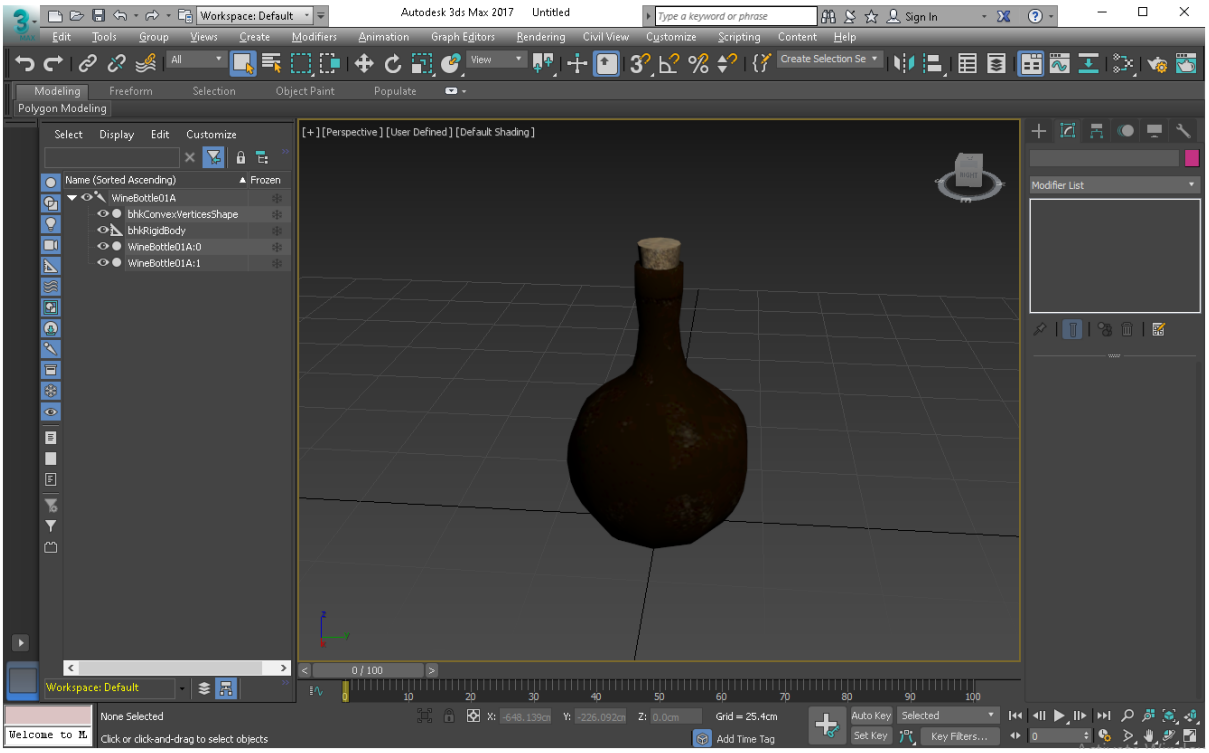


Figure 1062 - winebottle01a imported into 3ds Max.

EXPORTING ASSETS FROM SPEEDTREE TO 3DS MAX

SpeedTree is used by many game studios to create game-ready trees and vegetation.

This section covers the process of exporting a SpeedTree tree to 3ds Max to then bring over to Skyrim.

You can download a trial version of SpeedTree from the SpeedTree website.

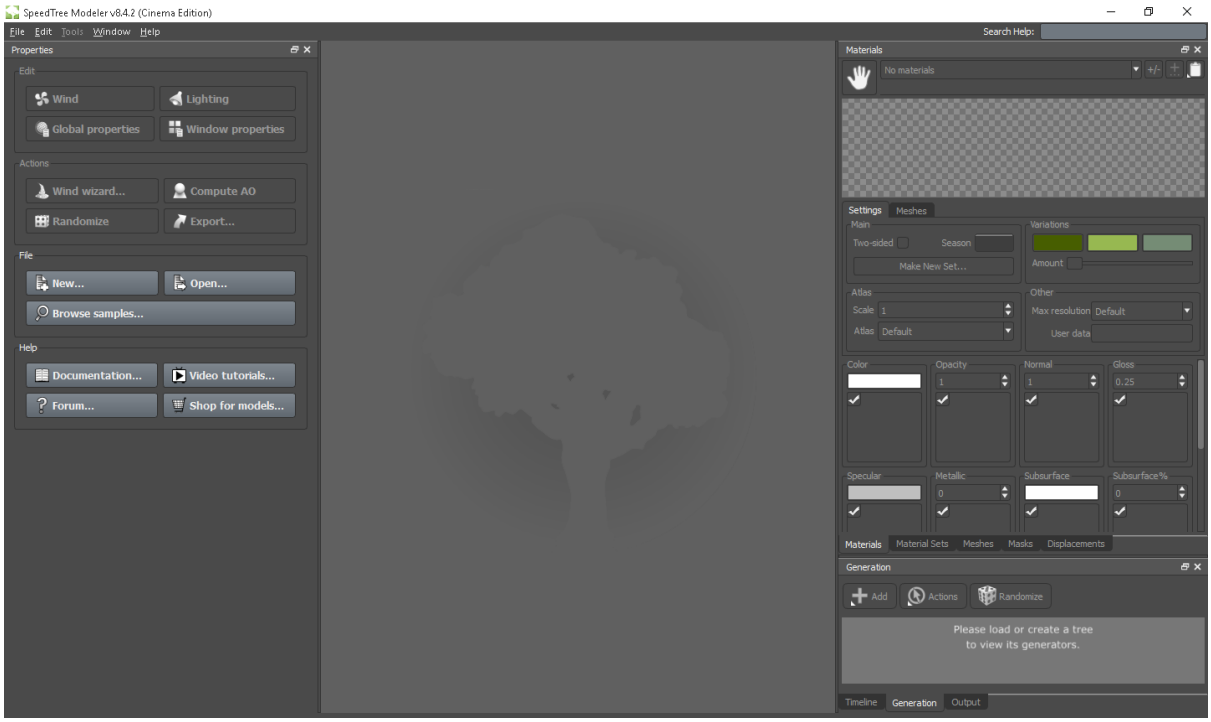


Figure 1063 - SpeedTree 8 Cinema Edition.

If we have a look at one of the trees from Skyrim in 3ds Max, we'll see that it's comprised of three main components; the trunk, large branches, and small branch & leaf cards.

Looking at an existing model will also give us a good idea on how many polygons our new tree should roughly consist of, for the sake of performance. treeaspen01.nif, for example, consists of 1,256 polys.

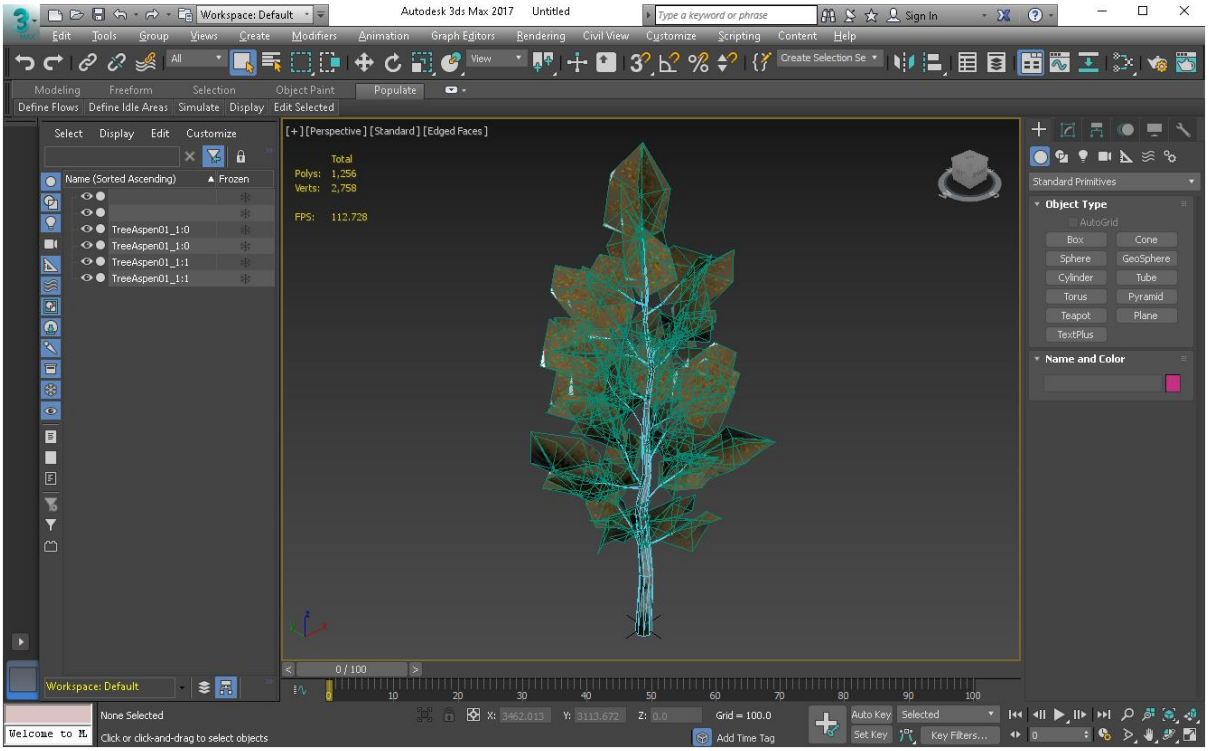


Figure 1064 - Tree in 3ds Max.

Before we export a tree from SpeedTree, we should reduce its poly count to roughly match the poly count of existing in-game trees.

Go to File > Open.

Select the tree you want to export then click Open. For this example I'll be working with the 'DouglasFir Low' tree from the sample library.

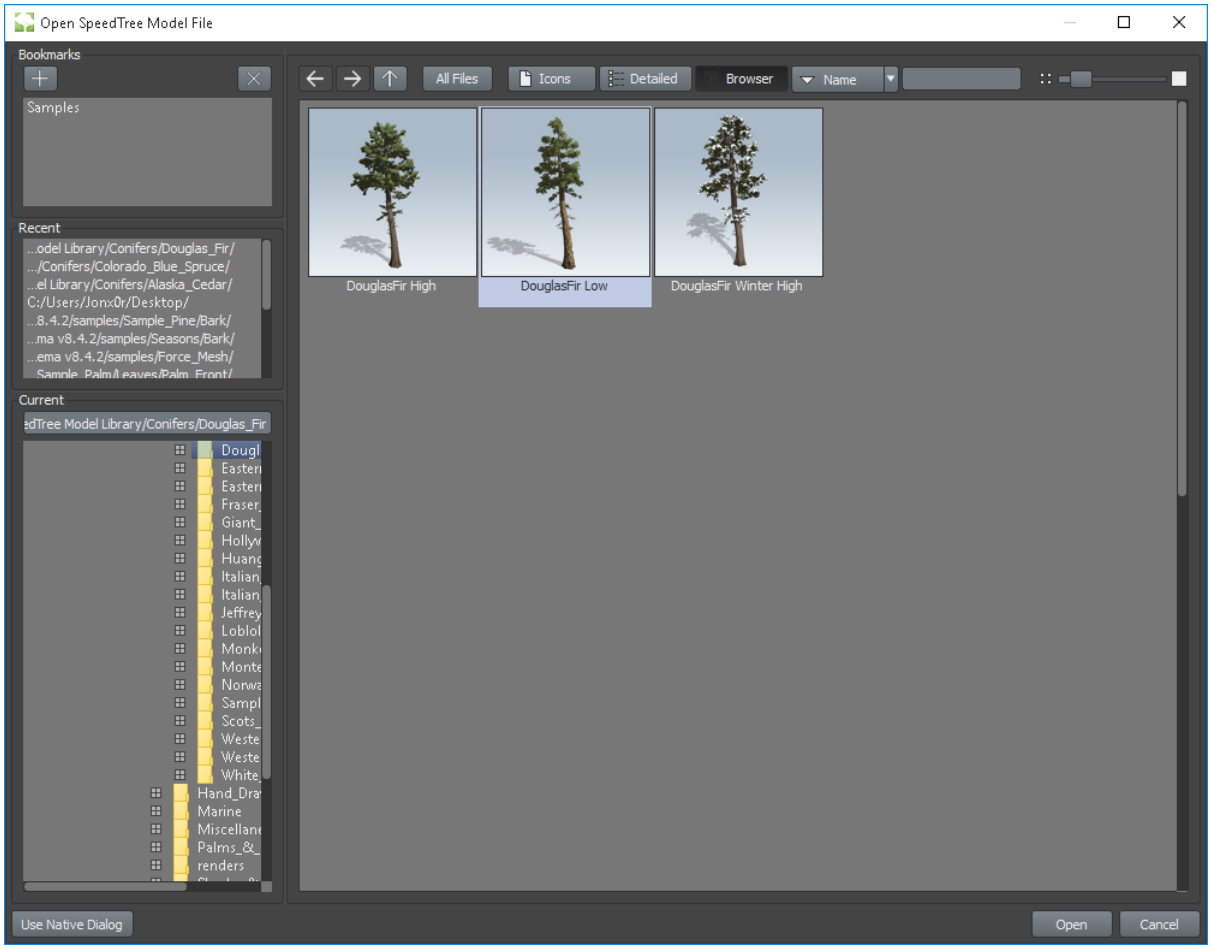


Figure 1065 - Loading DouglasFir Low.

Currently our tree has 3,781 polys. Let's reduce that a bit.

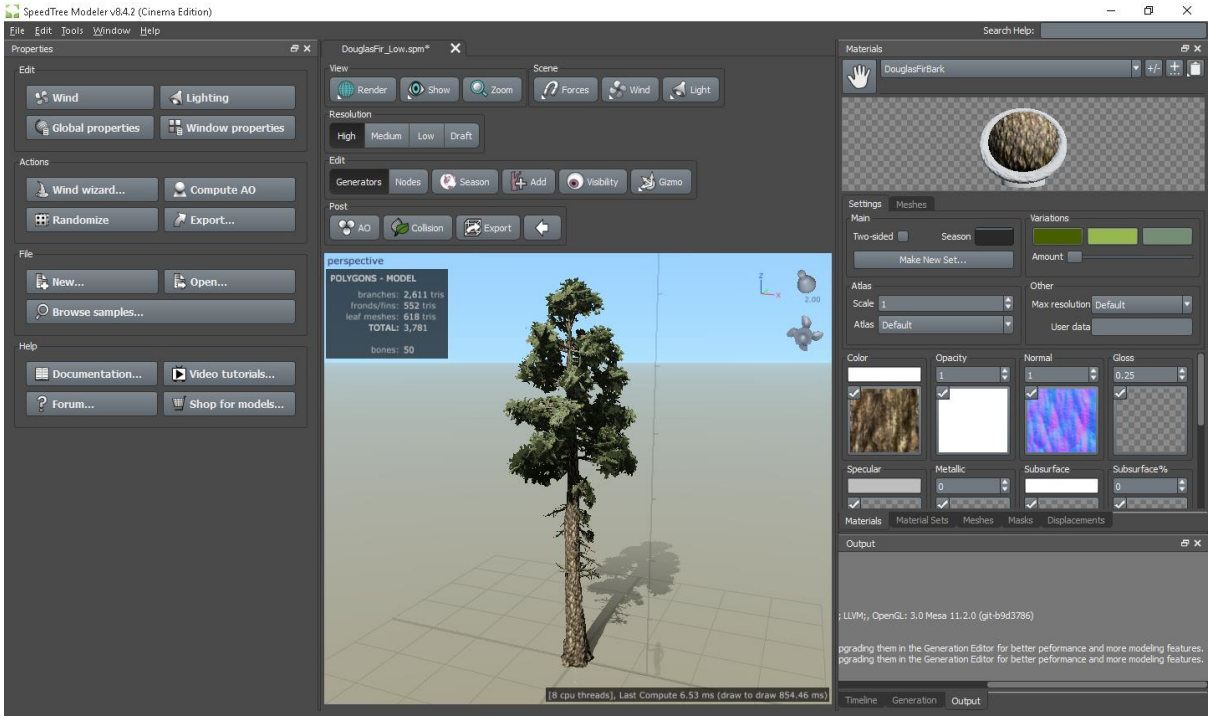


Figure 1066 - DouglasFir Low imported into SpeedTree.

Click on Render and select Scribed.

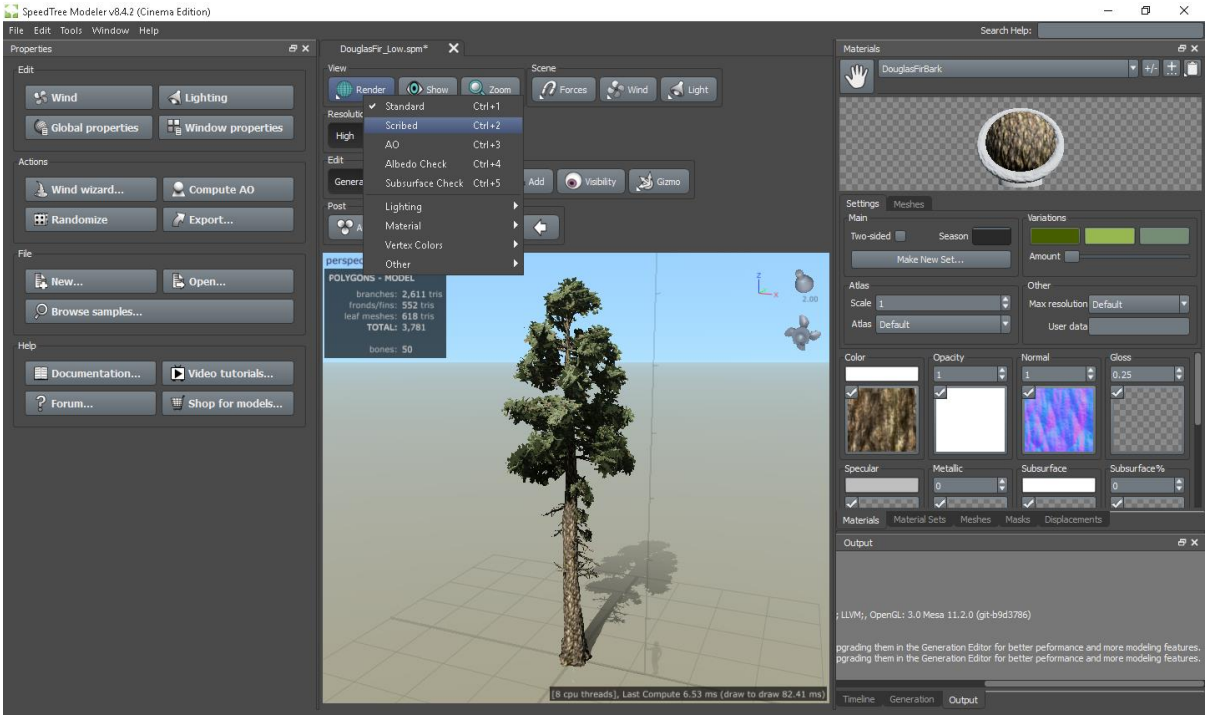


Figure 1067 - Scribed mode.

Let's start with the Trunk node. In the Generation tab, click on the Trunk node.

Go to the LOD section and reduce the Accuracy slider to simplify the trunk. In my example I lowered it from 50 all the way to 8. You should notice a simplification in the trunk geometry.

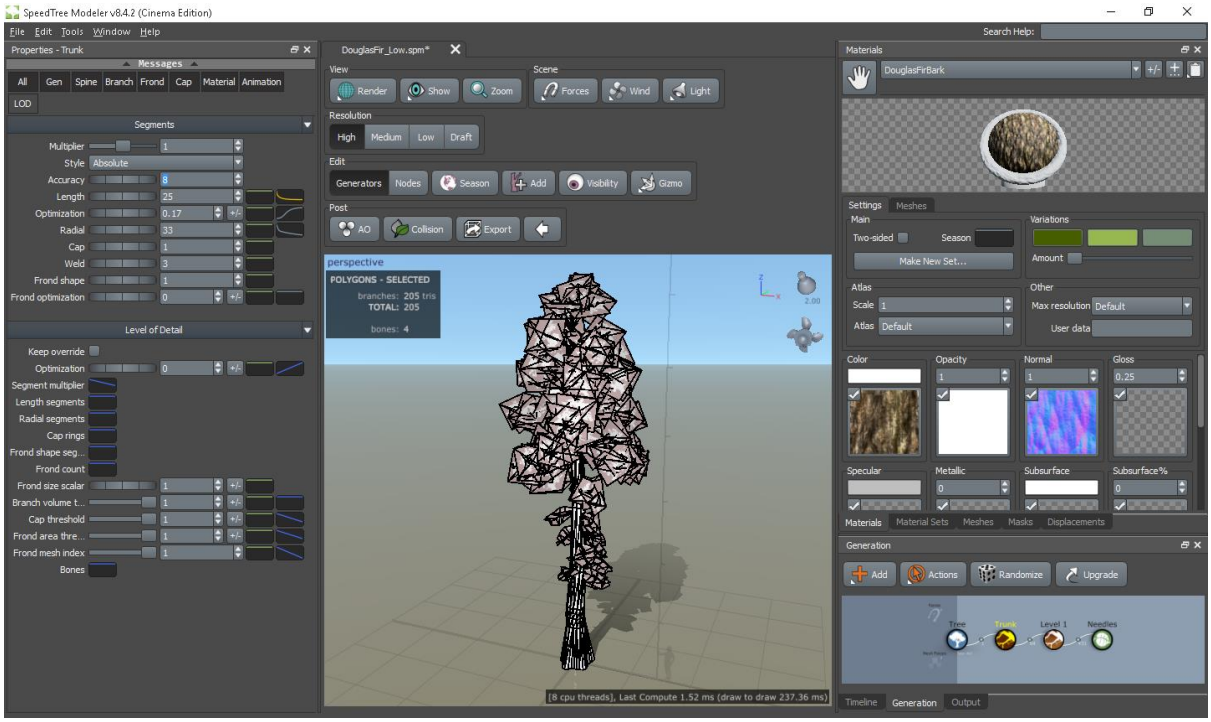


Figure 1068 - Simplifying the Trunk node.

Next, select the branches node above it. For the douglas fir example, that's going to be the node called level 1. We can see it's responsible for 2,622 polys.

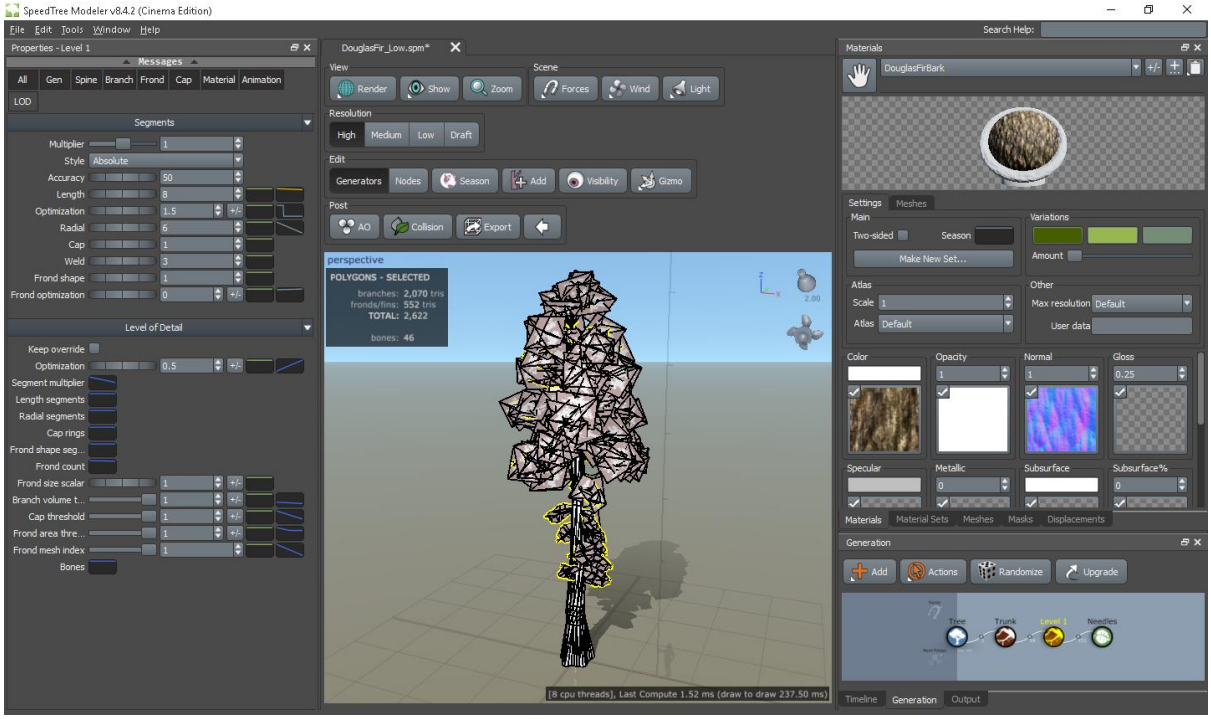


Figure 1069 - Level 1 node.

In the Gen section go to Knockout and increase the amount to randomly remove branches. But don't set it too high, just high enough to lower the density of branches without compromising the shape of the tree's foliage. In the screenshot below, I set this value to 0.25.

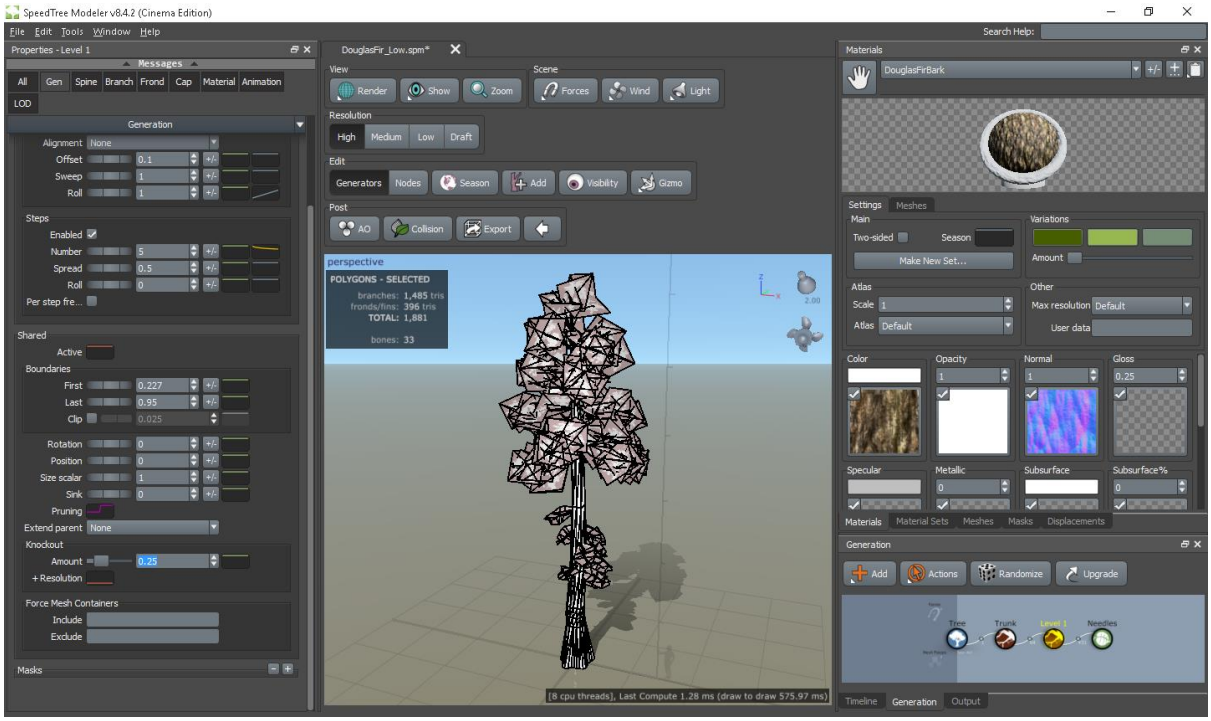


Figure 1070 - Reducing the branches poly count.

Alright, we're down to 2,526 polys.

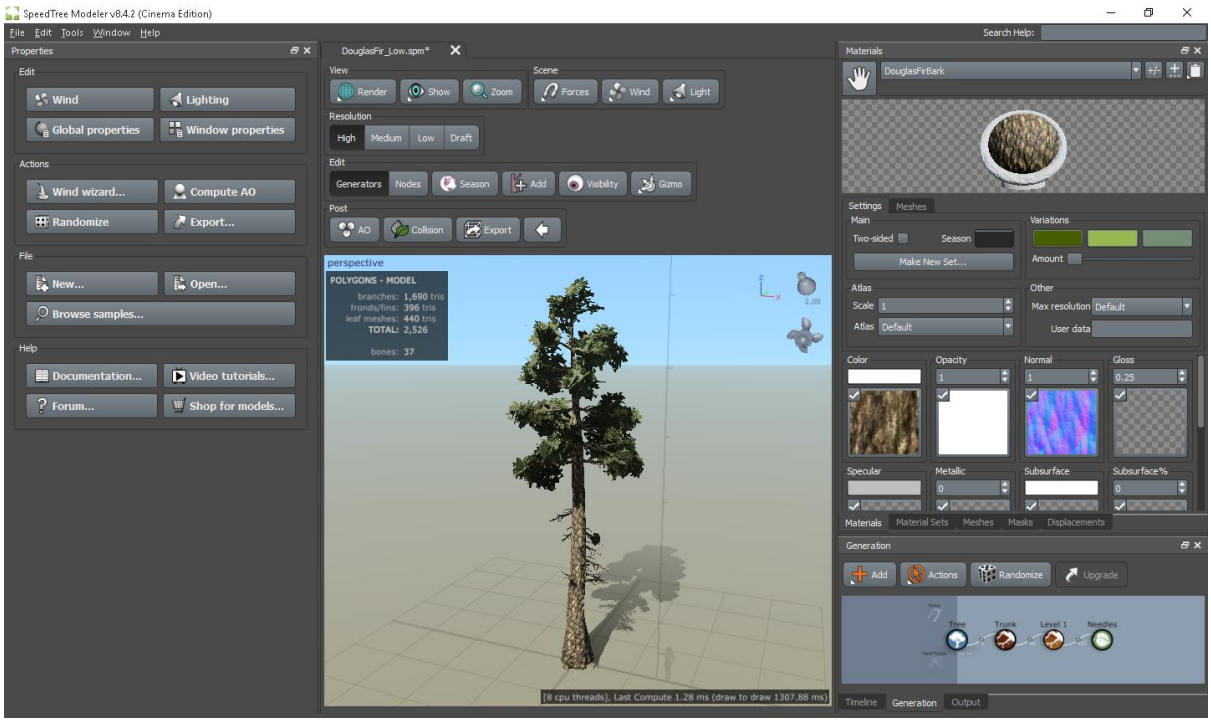


Figure 1071 - Our tree after poly reduction.

If you want a different trunk shape, click on the Trunk node in the Generator tab and click Randomize.

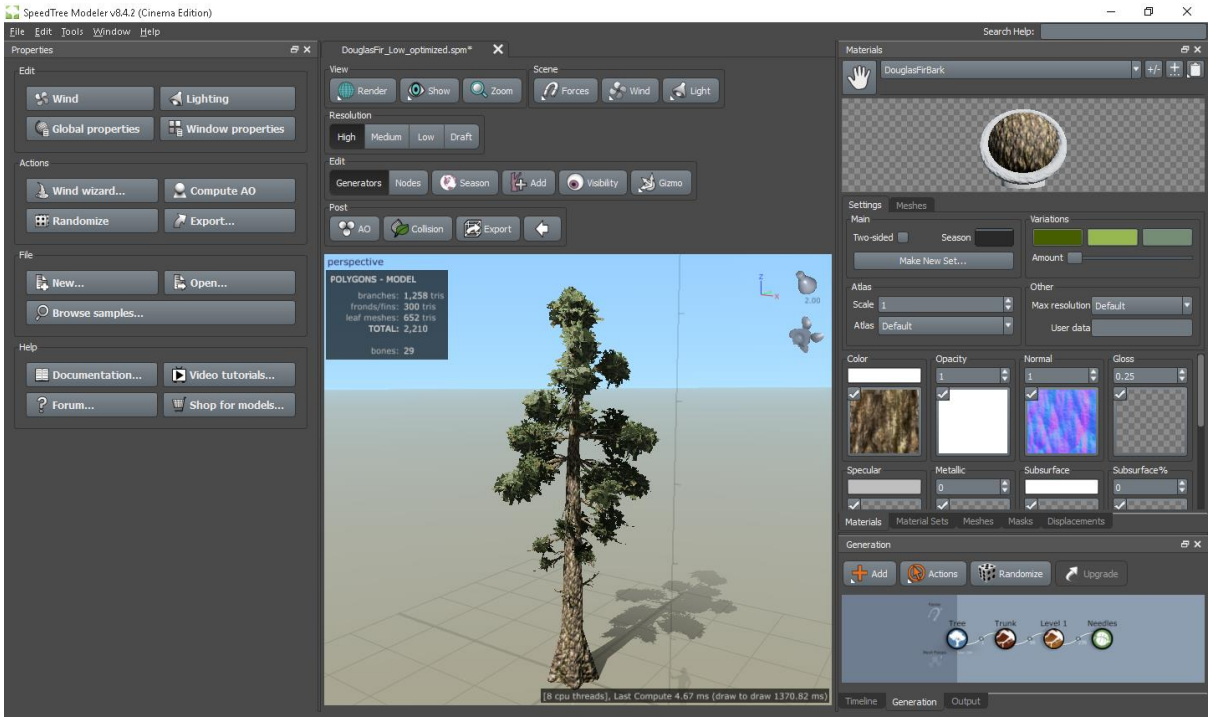


Figure 1072 - Randomized douglas fir.

I randomized until I got a tree that was 2,210 polys which should be low enough for us to use in-game.

To export the tree, go to File > Export Mesh.

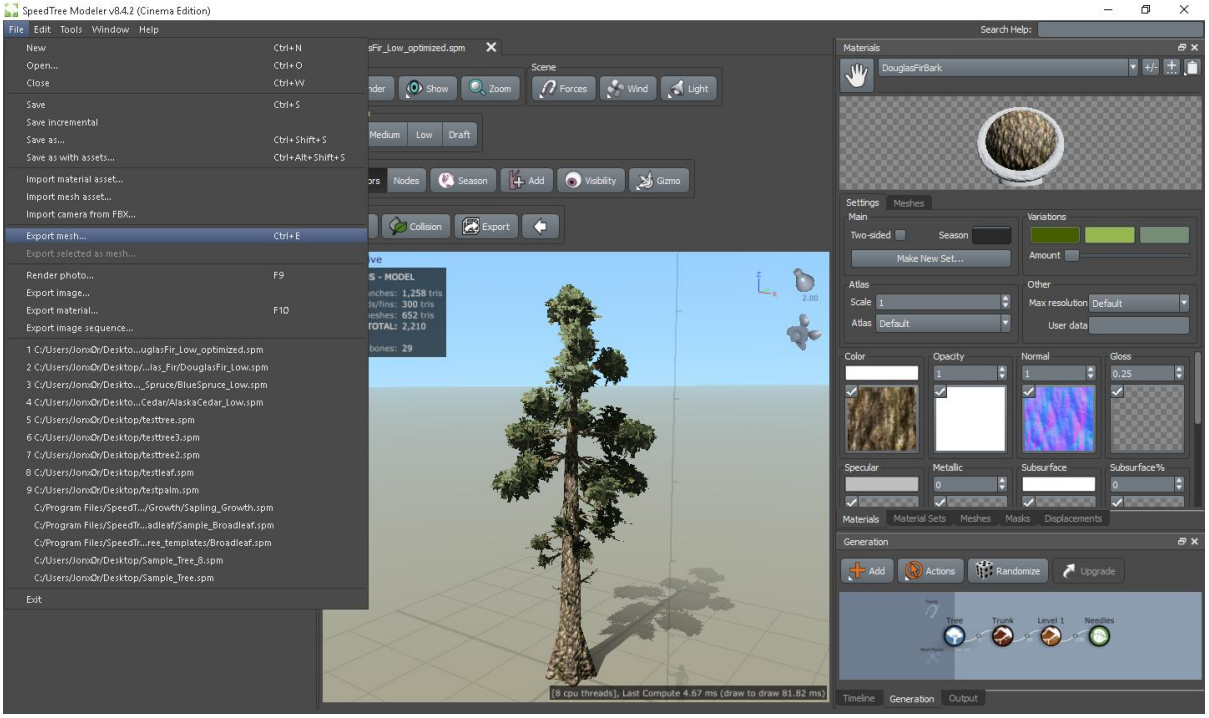


Figure 1073 - Export Mesh.

Select the folder to export the tree to and enter in a file name. Set the 'Save as type' field to 'Autodesk FBX (*.fbx)' then click Save.

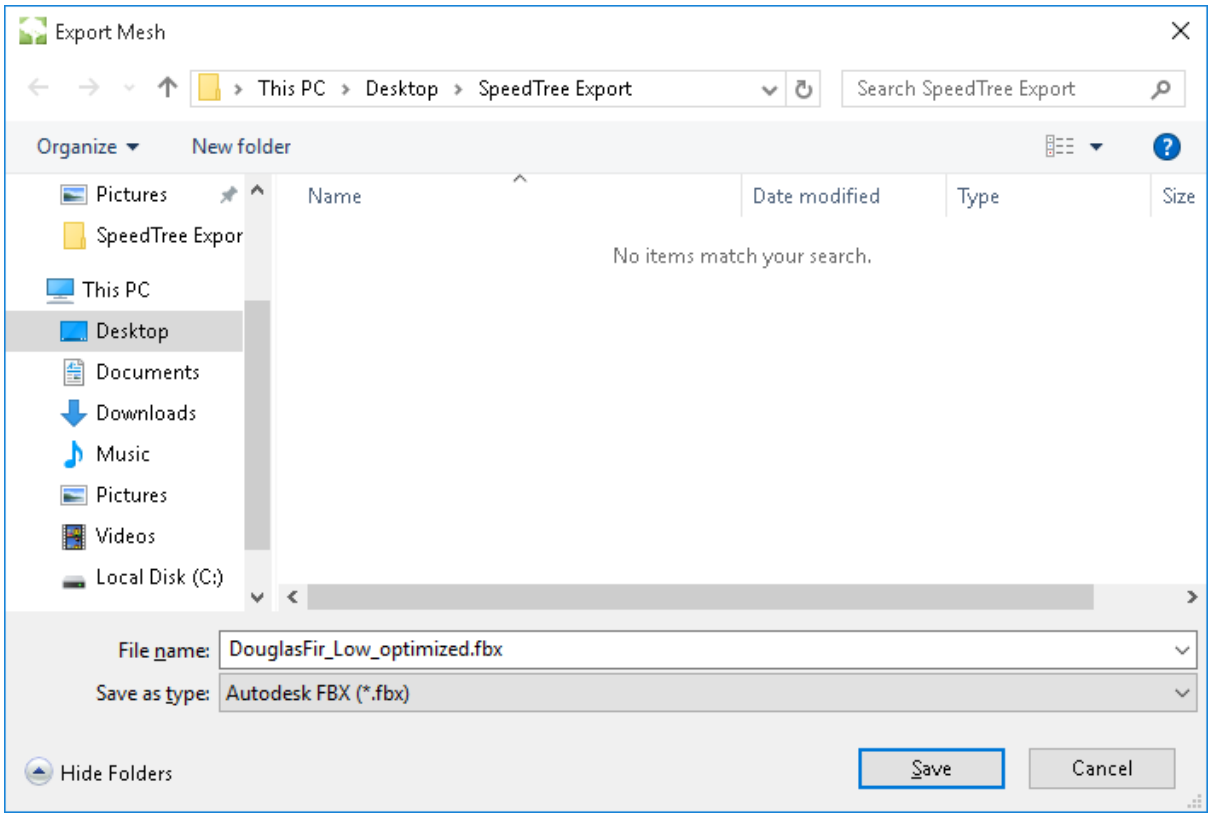


Figure 1074 - Filename and export folder.

In the Group By section, select Material. Under Transform, set the Scale to 30 and untick 'Swap YZ'. Under Textures, set Format to DDS and set 'Max size' to 1024.

Click OK to begin exporting.

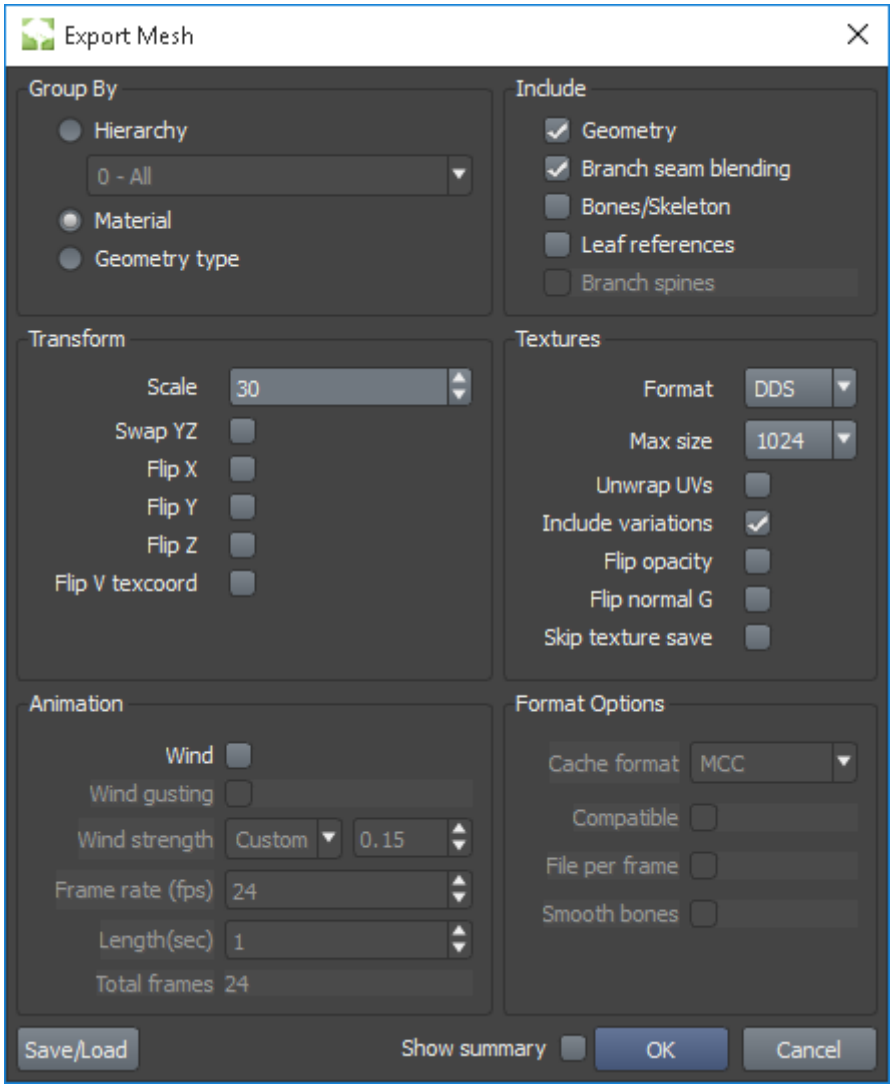


Figure 1075 - SpeedTree export mesh settings.

Now to import our asset into 3ds Max. For this, we'll need the SpeedTree Importer script which you can find in your SpeedTree installation directory under scripts\3ds Max.

Launch 3ds Max.

Go to Scripting > Run Script.

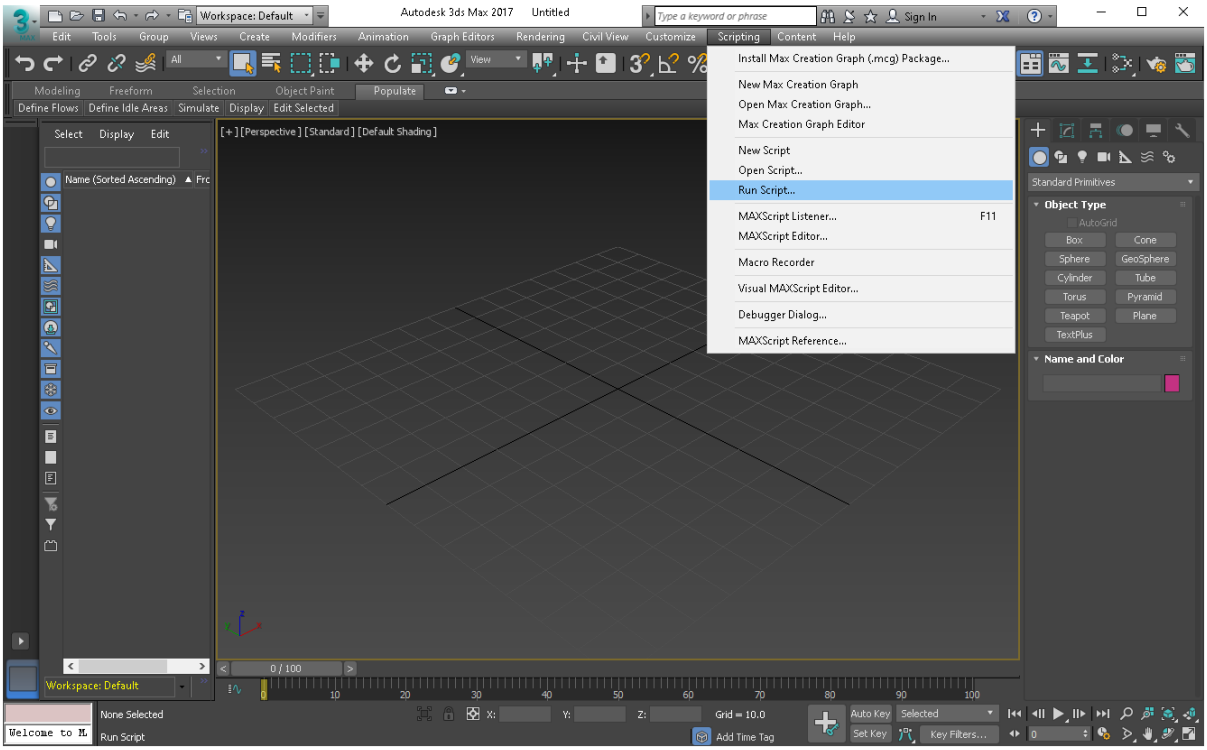


Figure 1076 - Running a script in 3ds Max.

Select SpeedTreeImporter.ms and click open.

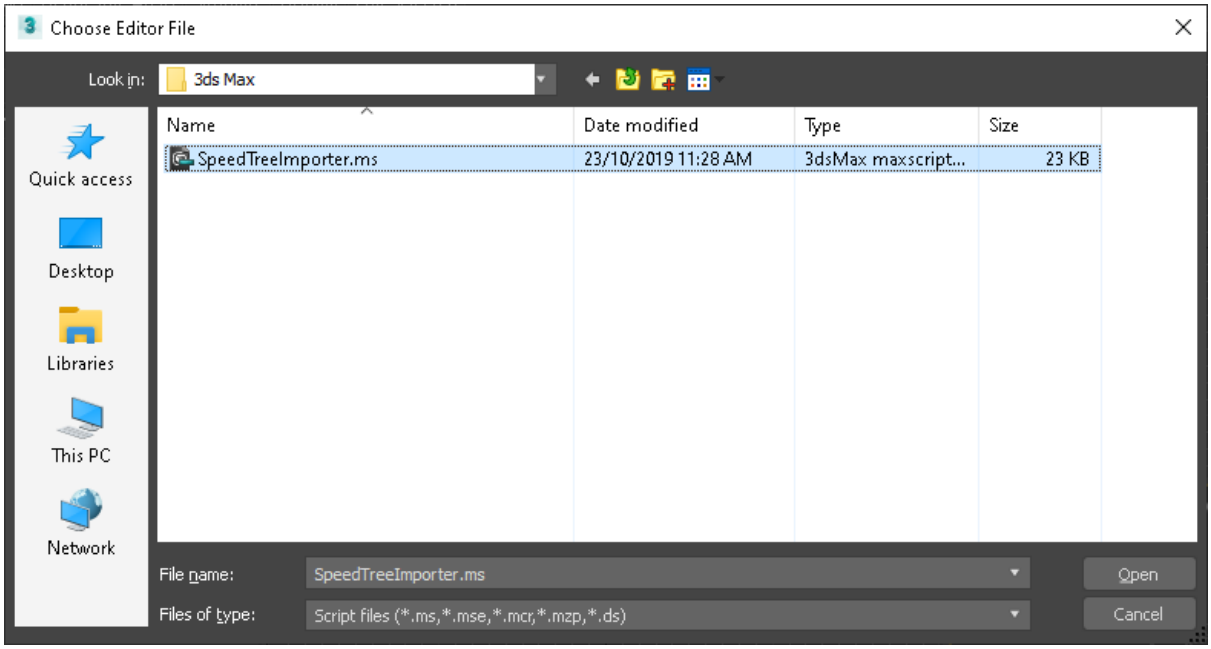


Figure 1077 - Running SpeedTreeImporter.ms.

You should see a new SpeedTree Importers panel appear on the right hand side.

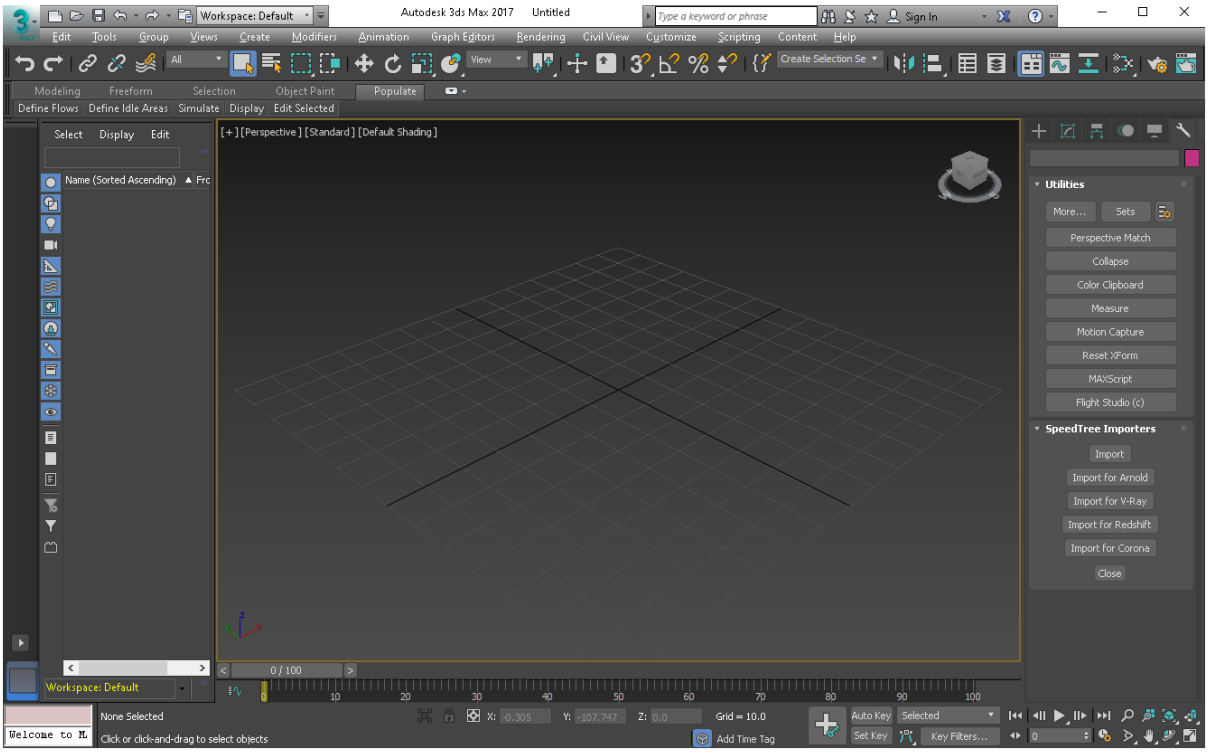


Figure 1078 - SpeedTree Importers panel.

Click on the Import button.

Select the .stmat file from the export folder we selected previously and click Open.

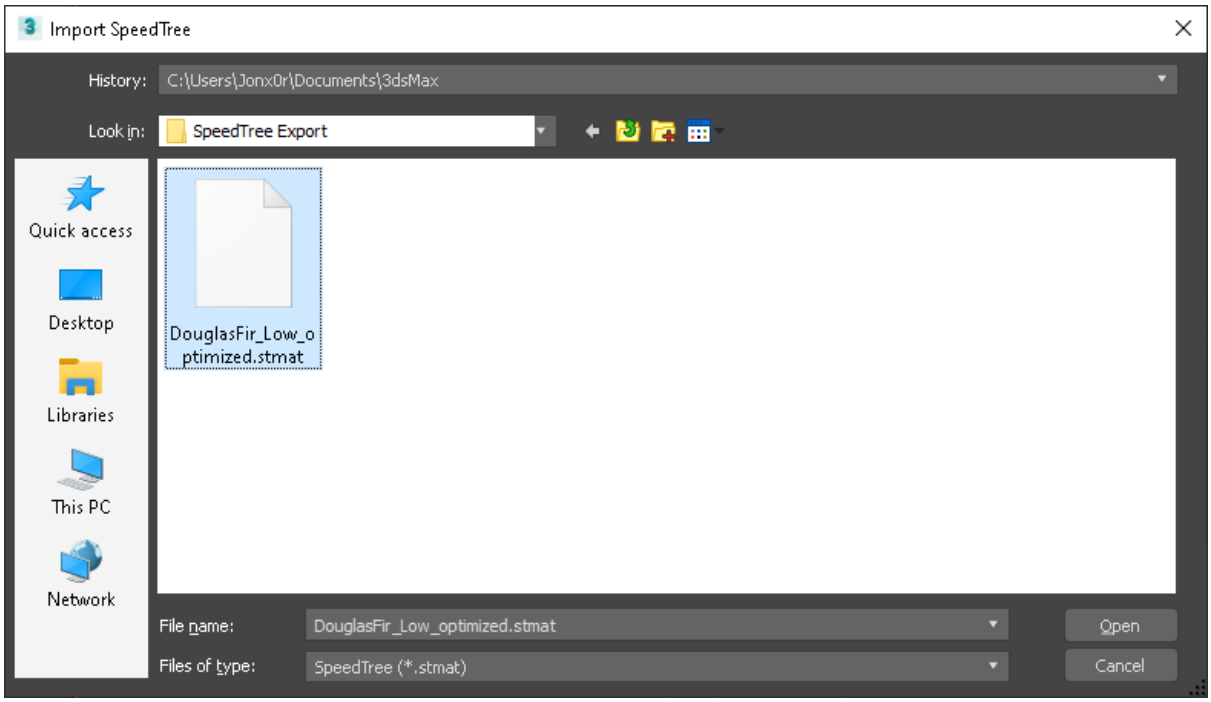


Figure 1079 - Selecting the .stmat file.

This should import the SpeedTree mesh and materials into 3ds Max.

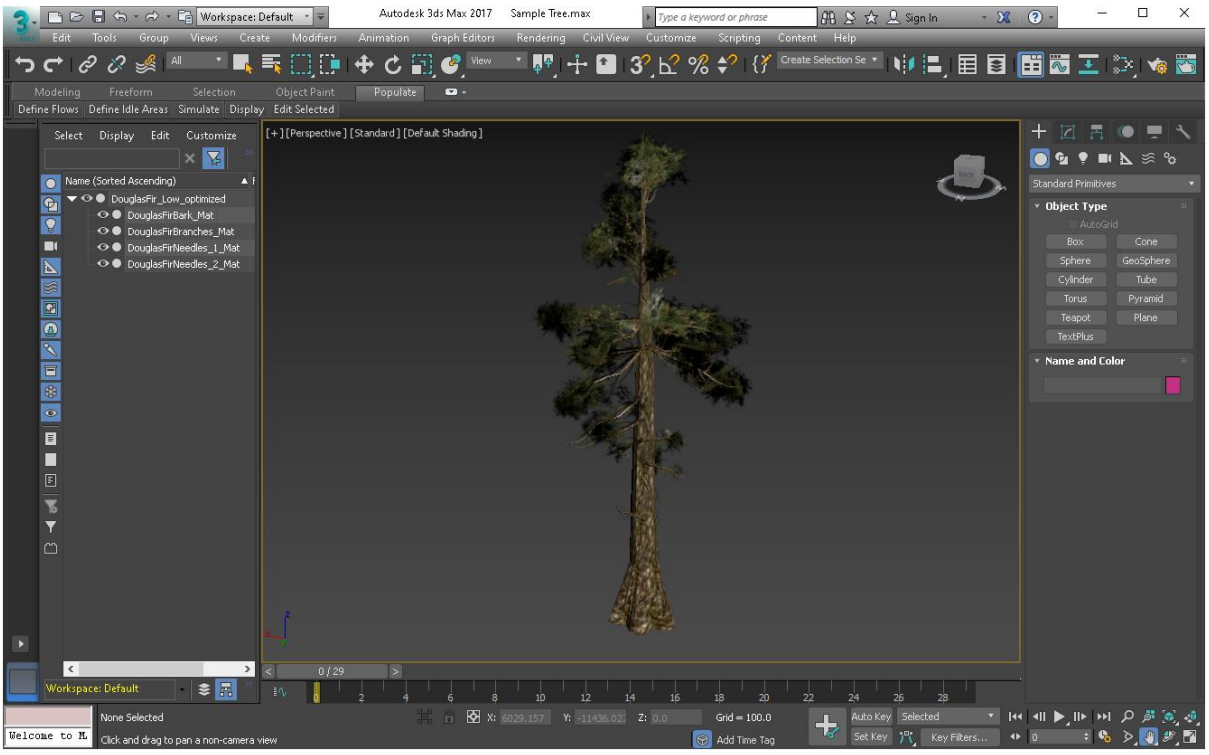


Figure 1080 - SpeedTree model in 3ds Max.

See the section Exporting Assets from 3ds Max to Skyrim for steps on exporting this asset in .nif format.

FIXING TRANSPARENCY IN NIFSKOPE

If we have a look at our tree in NifSkope, we'll see that there's a problem with the transparency on the leaf and twig cards.

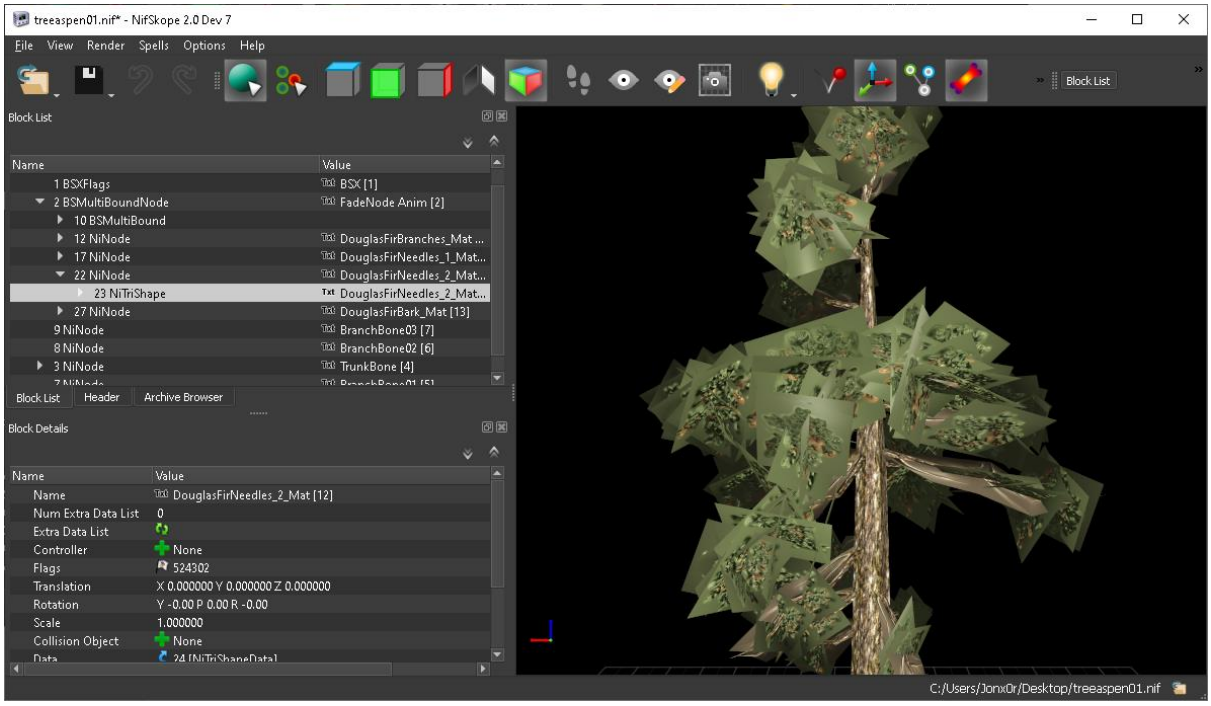


Figure 1081 - The douglas fir tree in NifSkope.

We need to make sure our alpha channels are set up properly so they appear transparent in-game.

Here's the problem. As per the screenshot below, the DouglasFirNeedles_1.dds texture doesn't have an alpha channel.

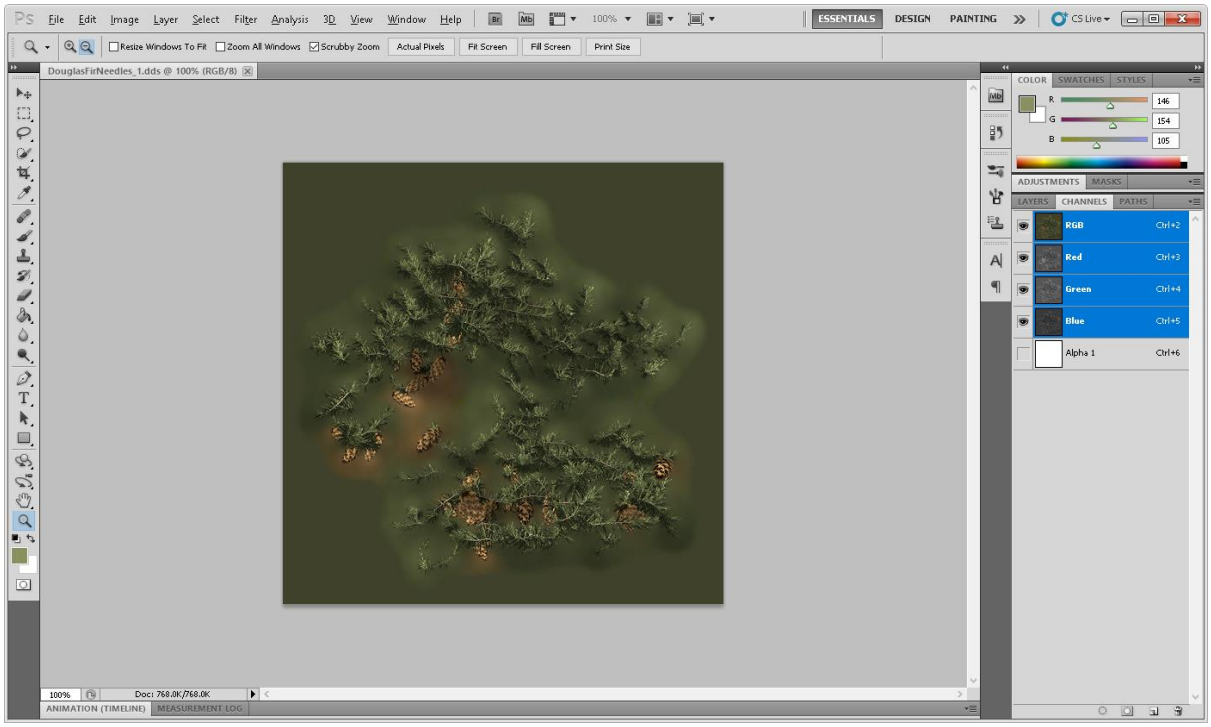


Figure 1082 - Blank alpha channel.

The leaves and twigs appear transparent in 3ds Max because we're getting our opacity from a separate opacity map, but this won't work in-game so we'll need to copy the alpha channel from our opacity map over to our diffuse map.

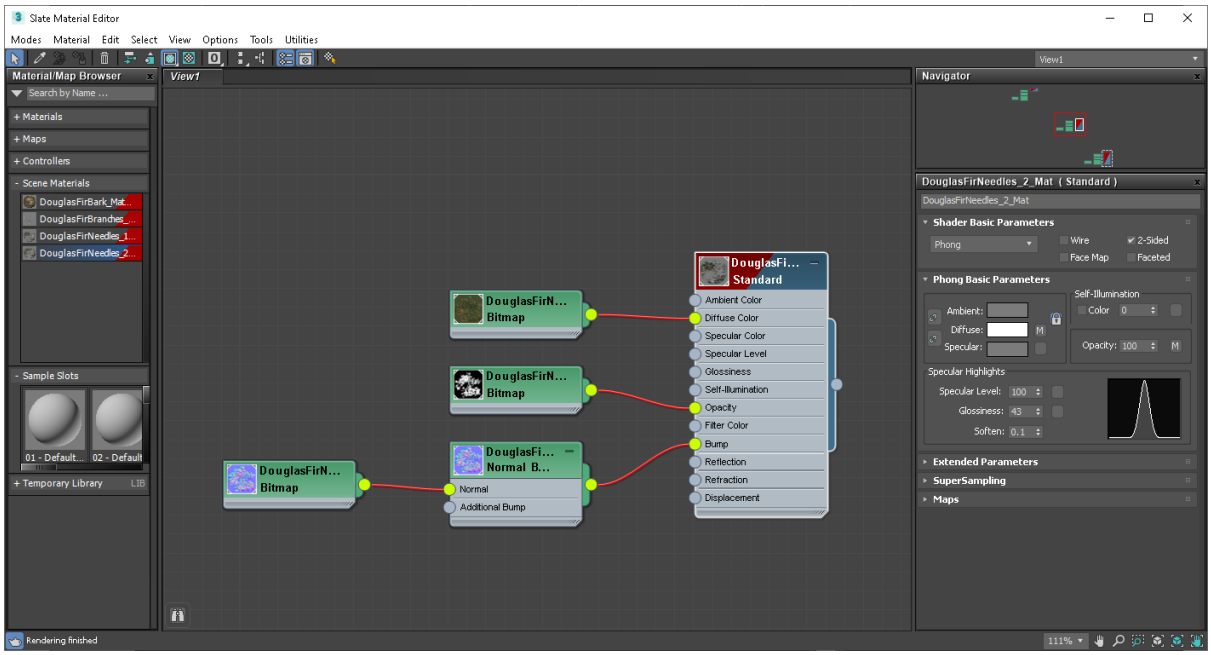


Figure 1083 - Douglas Fir needles material setup in 3ds Max.

Open the opacity map in Photoshop.

Press CTRL + A to select everything and CTRL + C to copy.

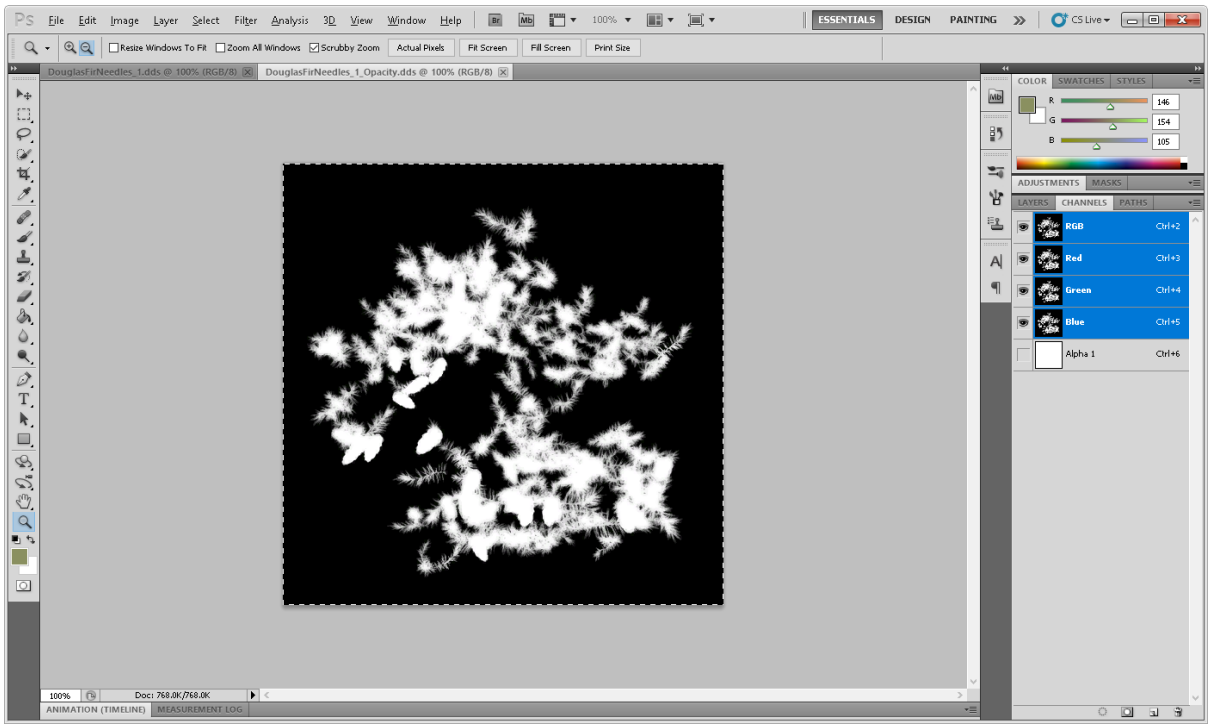


Figure 1084 - Copying the opacity map.

Now open the corresponding diffuse map and enable the alpha channel. It should be called Alpha 1.

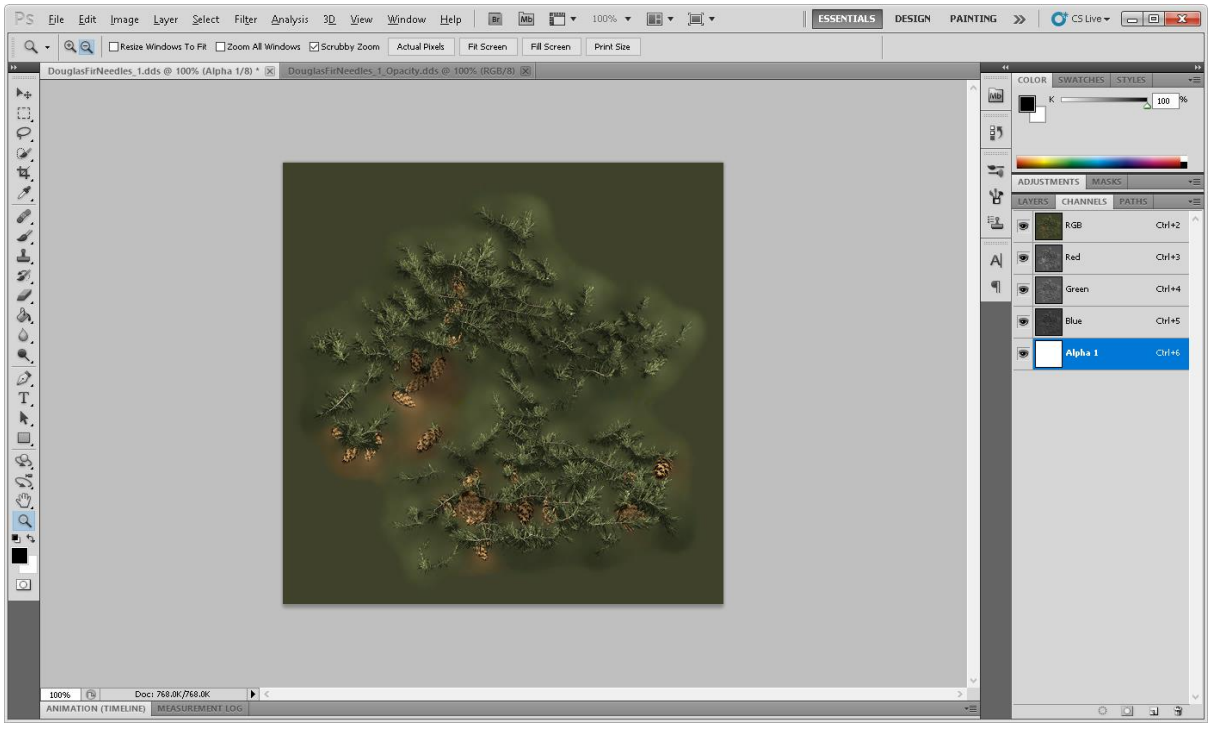


Figure 1085 - Diffuse map.

Select the alpha channel and press CTRL + V to paste in the opacity map.

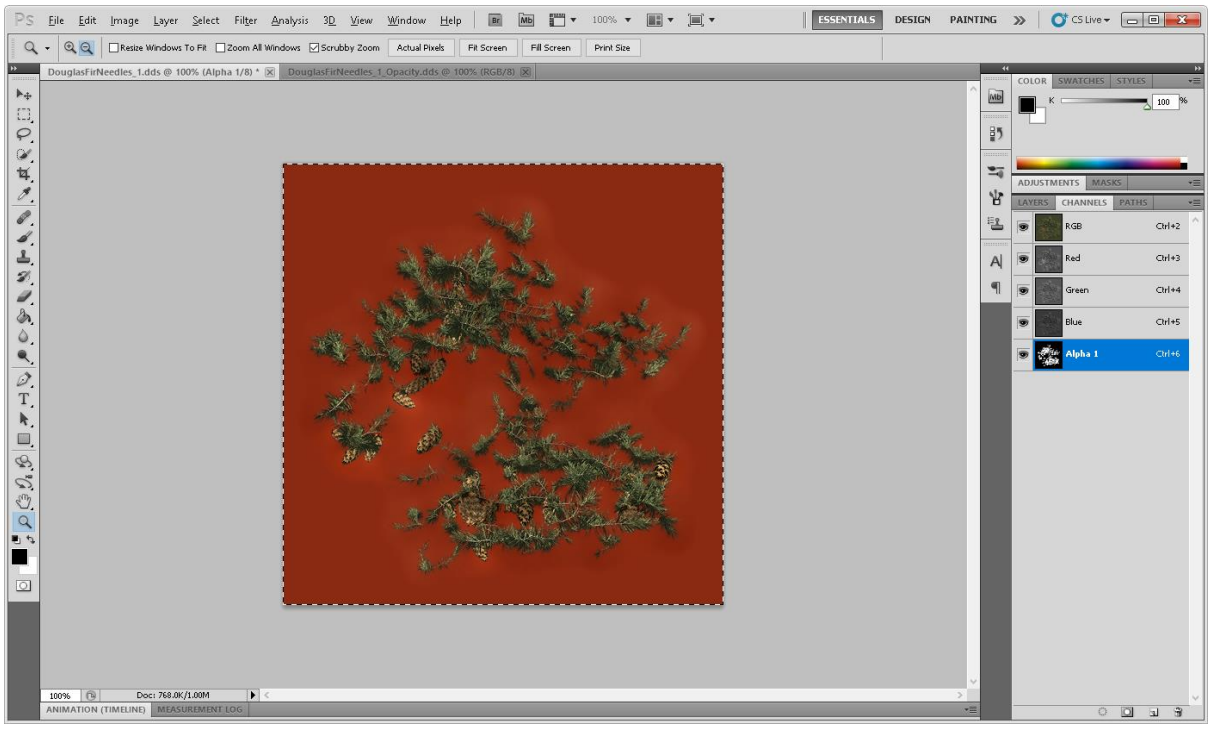


Figure 1086 - Pasting in opacity map in alpha channel.

Now save the diffuse map.

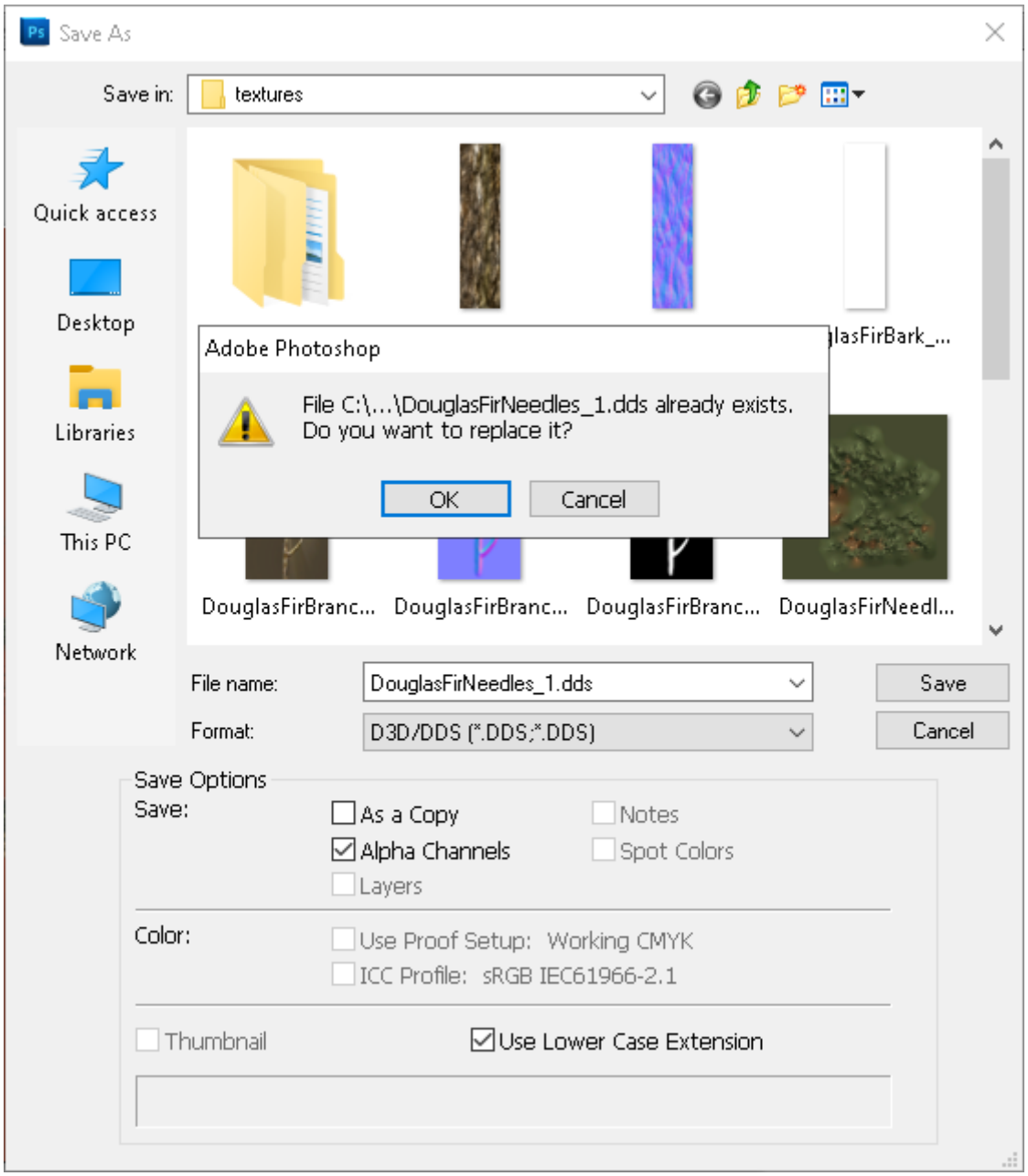


Figure 1087 - Saving the diffuse map.

Repeat these steps for any other textures that need an alpha channel.

Okay, back to NifSkope.

In order to enable the alpha channel, we need to attach a NiAlphaProperty node to each NiTriShape node that needs transparency.

Right-click on the NiTriShape node and go to Node > Attach Property.

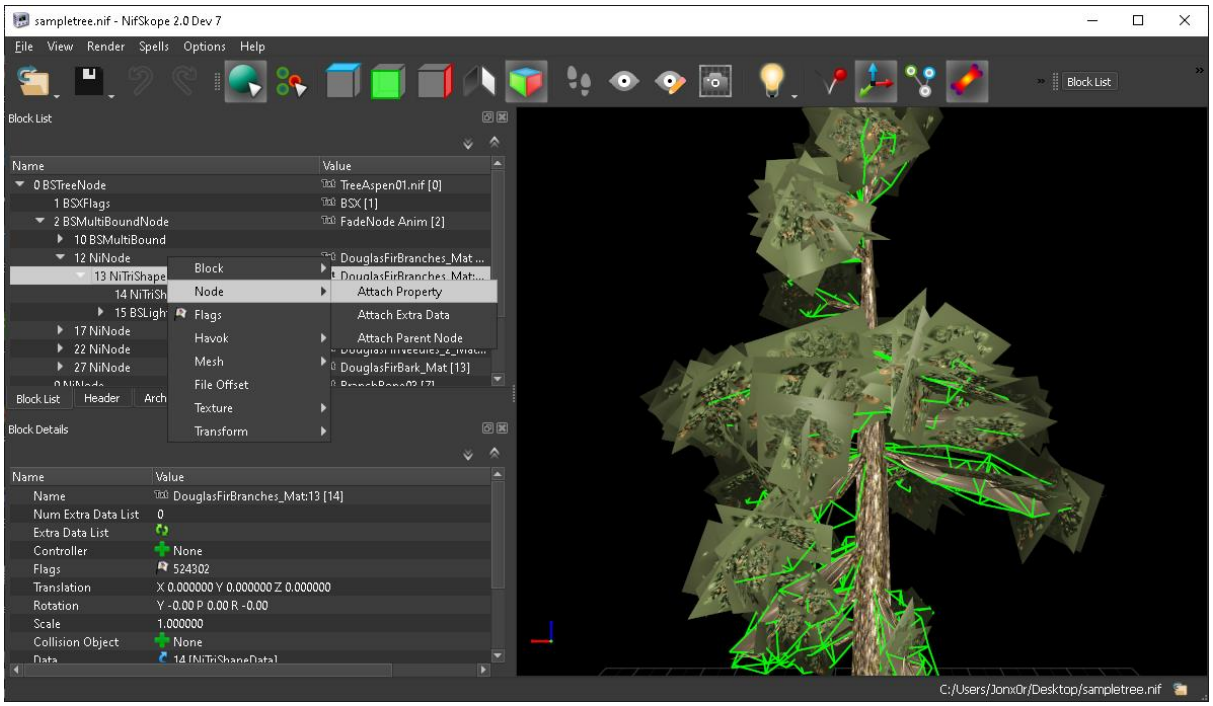


Figure 1088 - Attaching a new node.

Then select NiAlphaProperty.

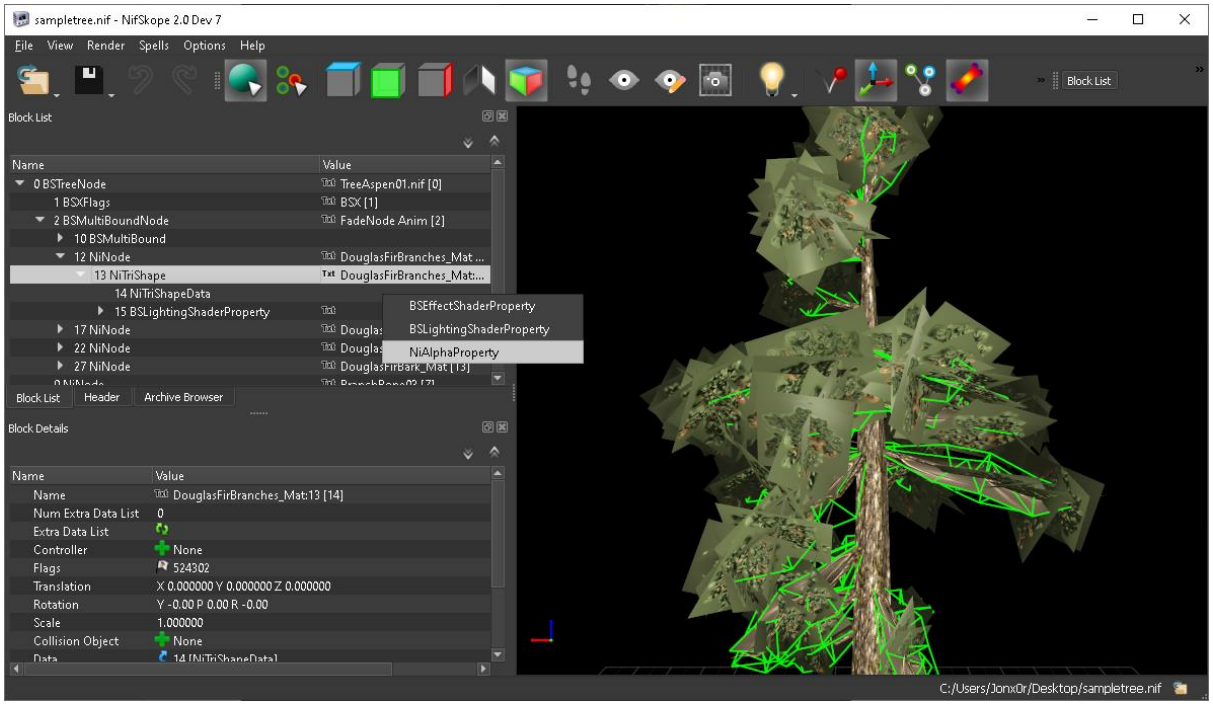


Figure 1089 - Attaching a NiAlphaProperty node.

Repeat these steps for the other NiTriShapes that need transparency.

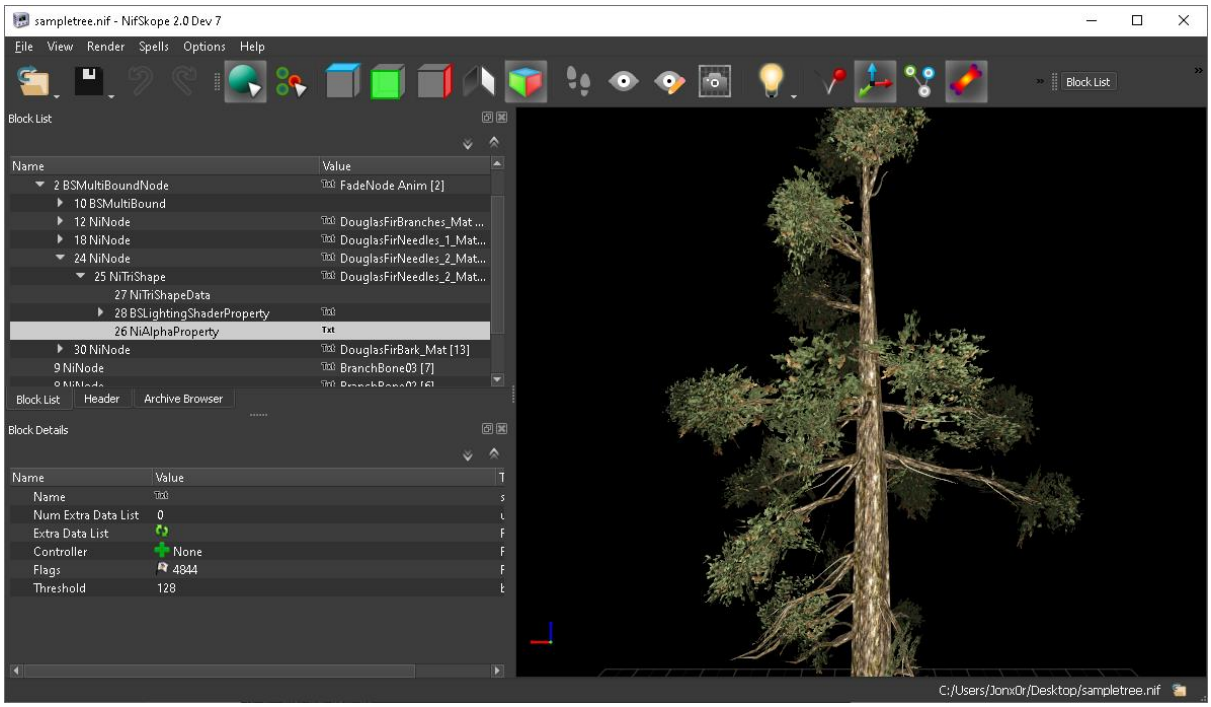


Figure 1090 - The douglas fir with transparency fixed.

Alright, so the transparency issue is fixed but our tree is looking a bit bare. This is because the exporter doesn't flag the 'Double_Sided' property on our two-sided materials. We need to do this manually.

Go to the BSLightingShaderProperty node beneath NiTriShape and scroll down to the Shader Flags 2 property.

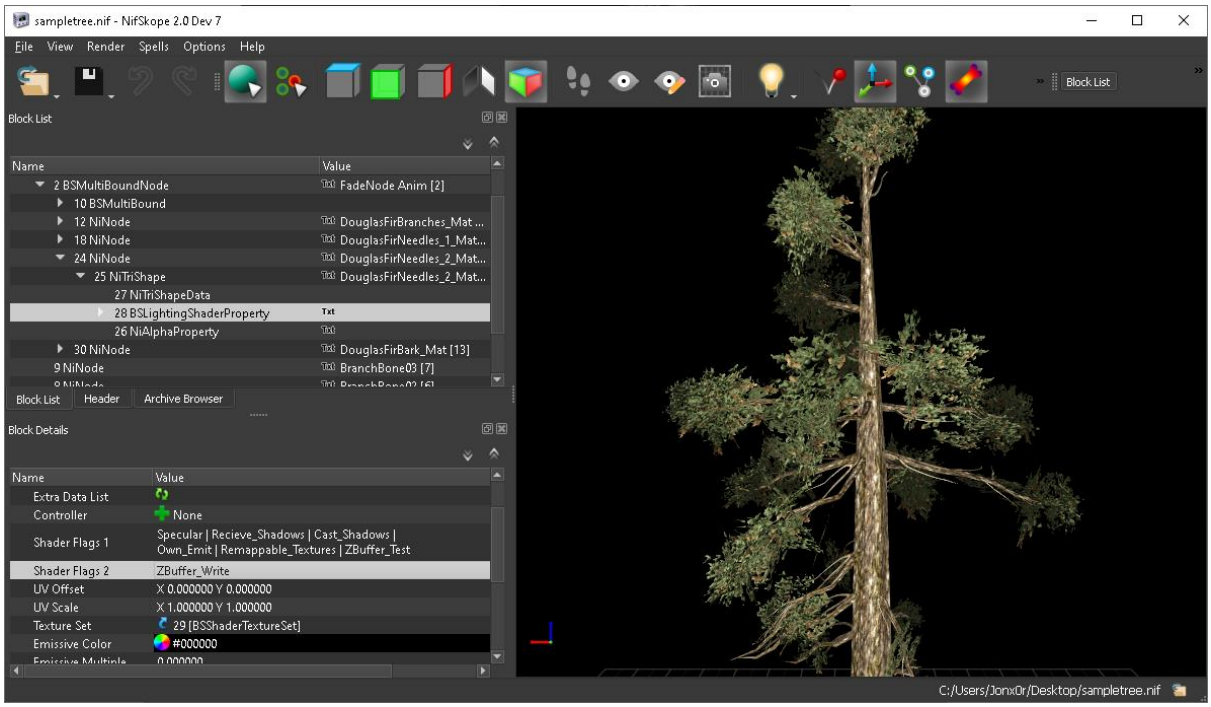


Figure 1091 - Shader Flags 2.

Right now it only has the ZBuffer_Write flag enabled.

Double-click on the Value field for Shader Flags 2 and click on the down arrow.

Tick Double_Sided.

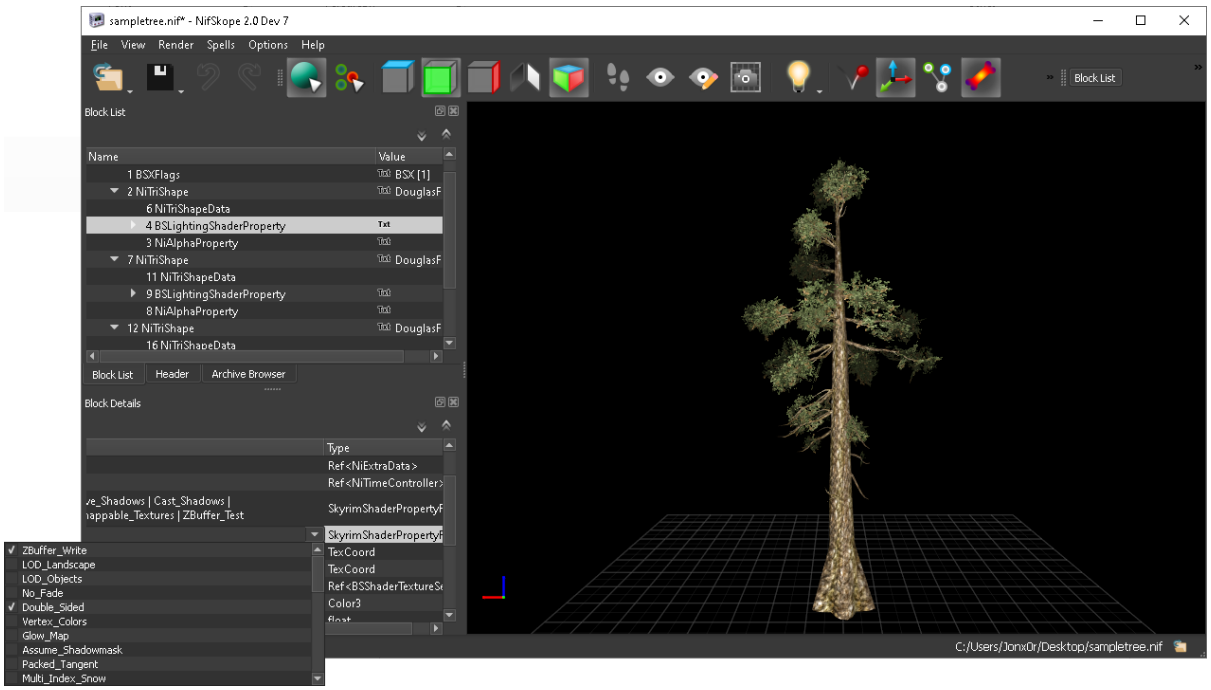


Figure 1092 - Making the leaf and twig cards double sided.

Repeat these steps for the other BSLightingShaderProperty nodes that need to be displayed as double-sided.

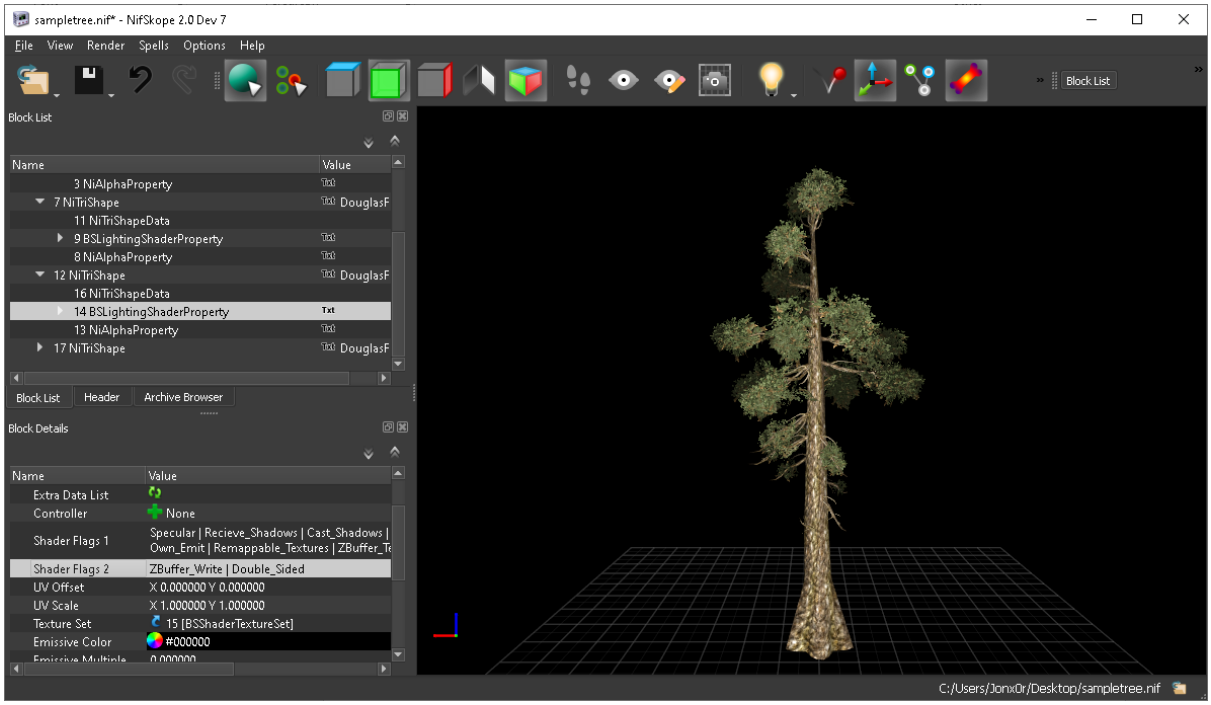


Figure 1093 - Tree with double-sided materials enabled.

EXPORTING ASSETS FROM ZBRUSH TO SUBSTANCE PAINTER TO 3DS MAX

This section is going to cover the general Zbrush to 3ds Max workflow. I'll be taking an asset made in Zbrush, texture it in Substance Painter, and import it into 3ds Max. I'll also be showing you how to fix UVs in 3ds Max.

In the following example I'll be working with Zbrush 2021 and Substance Painter 2018.

A trial version of Zbrush is available on the Pixologic website.

A trial version of Substance Painter is available on the Adobe website.

So for this example I just sculpted a simple boulder.

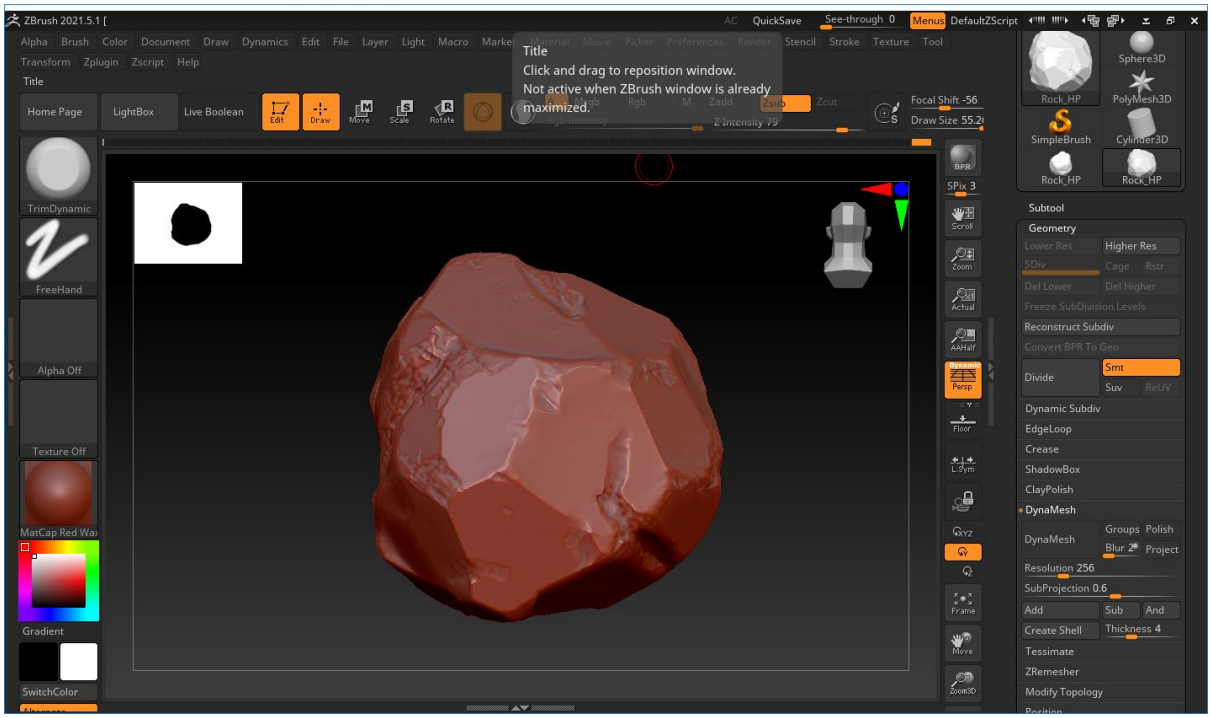


Figure 1094 - Rock sculpted in Zbrush.

Currently it consists of 205,854 points which is way too much for a game asset, so we need to create a low-poly version that we can bake the high poly version's details onto via a normal map.

But before we do that we need to save the high poly version of our asset.

Under Tool click on Export.

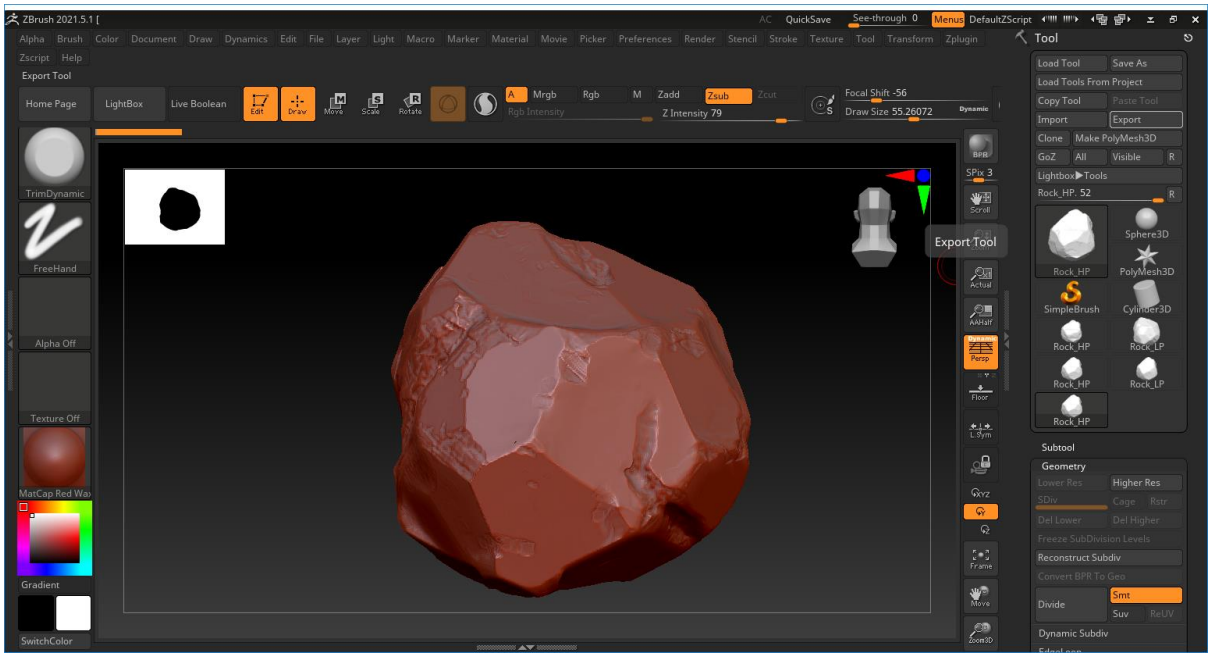


Figure 1095 - Saving the high poly asset.

Choose the folder to save to, enter in a file name and click Save. I appended '_HP' to the file name to differentiate it as the high poly version of our asset.

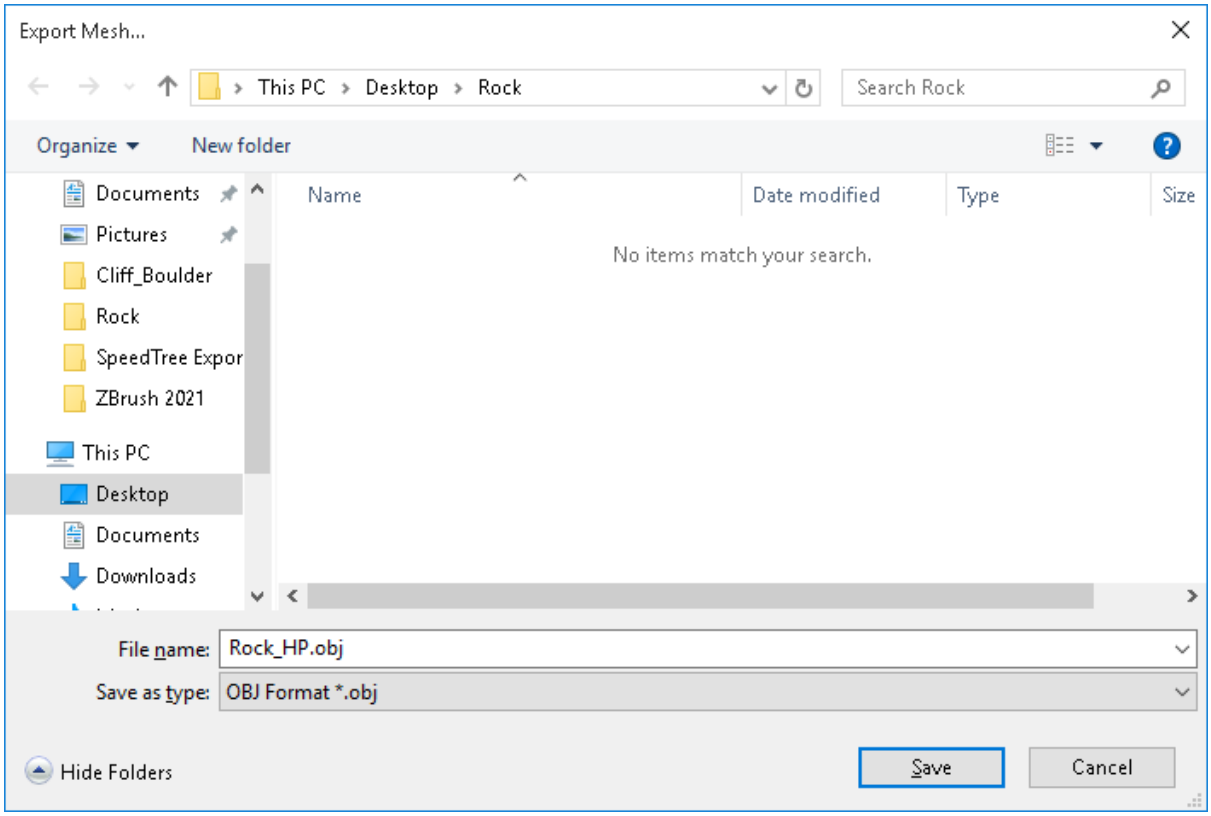


Figure 1096 - Setting the folder and file name.

Next, to create a low poly version I'm just going to use ZRemesher. I set Target Polygons Count to 0.3. You might need to play around with this value on your end to reduce the point count to something more reasonable.

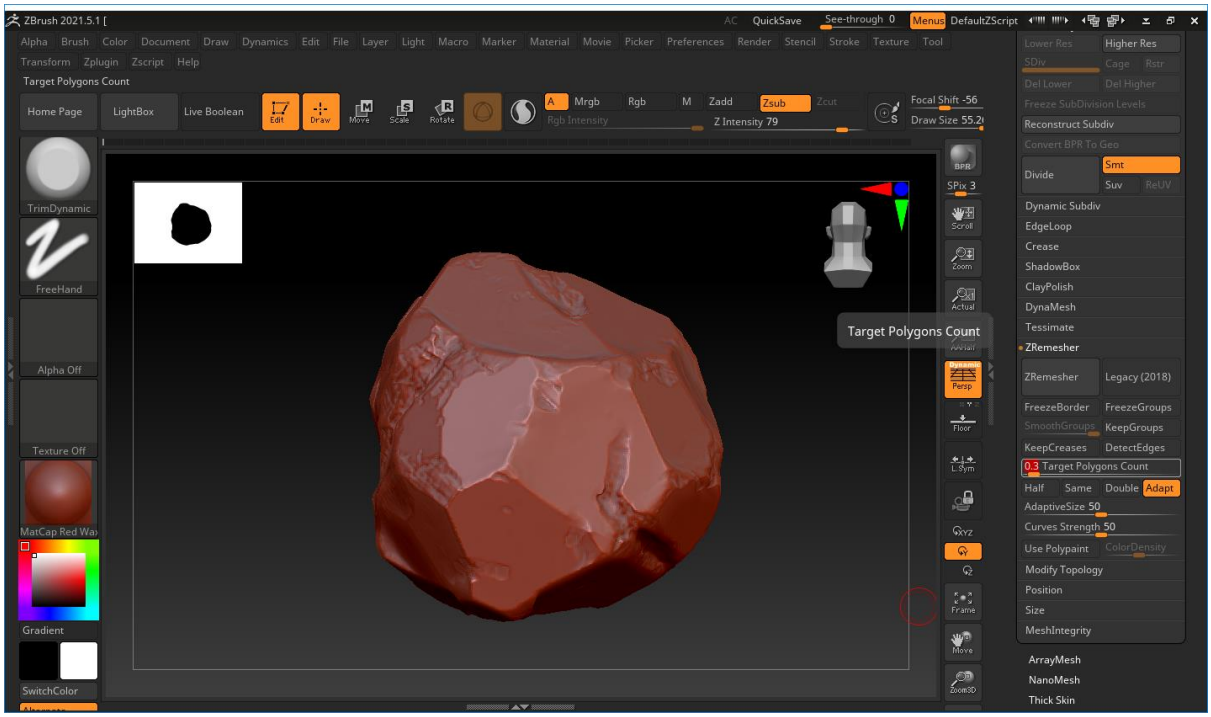


Figure 1097 - ZRemesher.

Click on ZRemesher to begin retopologizing. Depending on the type of model you may need to toggle KeepCreases to maintain hard edges, but for a natural object like this rock it isn't really necessary.

Once that's done, our model is now down to 600 points which is roughly on par with the vertex count of Skyrim boulders.

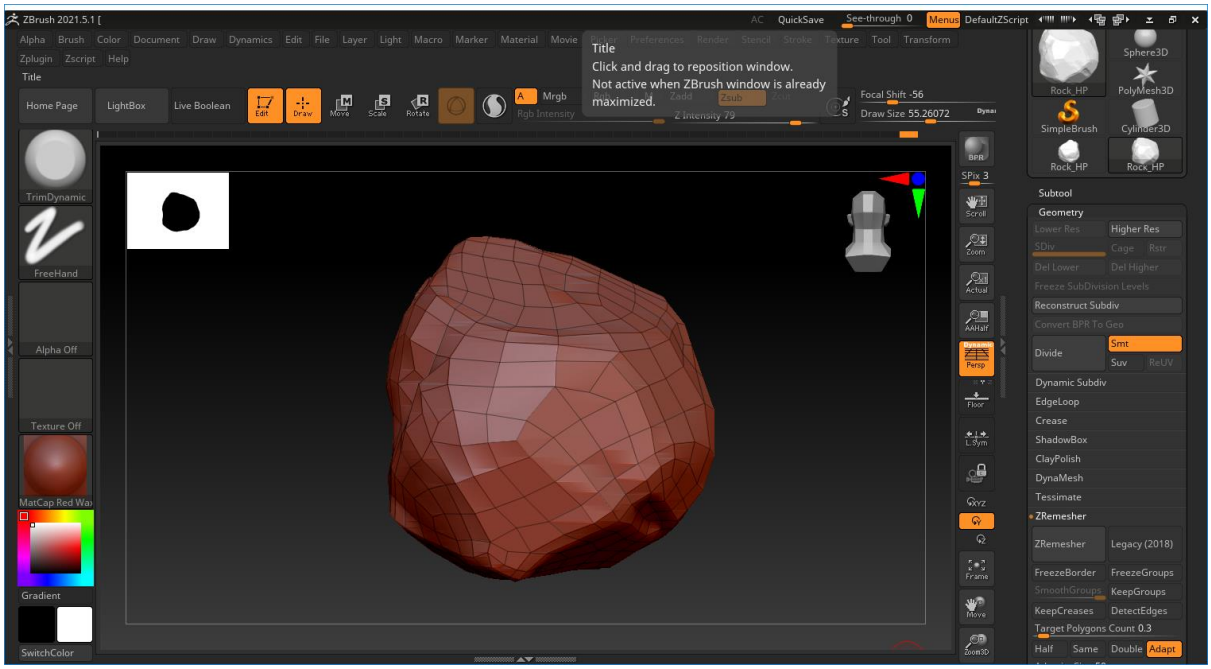


Figure 1098 - Retopologized mesh with a lower poly count.

Before we save our low poly version of the asset, we need to make sure the UVs have been unwrapped.

Go to Zplugin > UV Master.

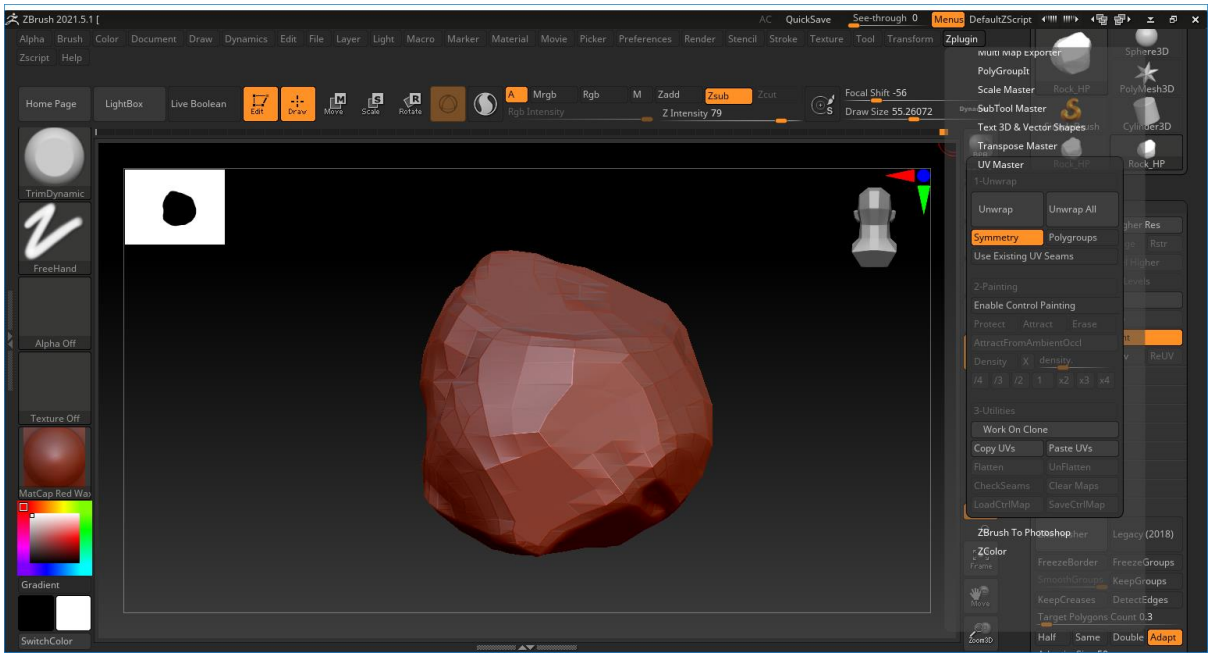


Figure 1099 - UV Master.

Click on Unwrap All.

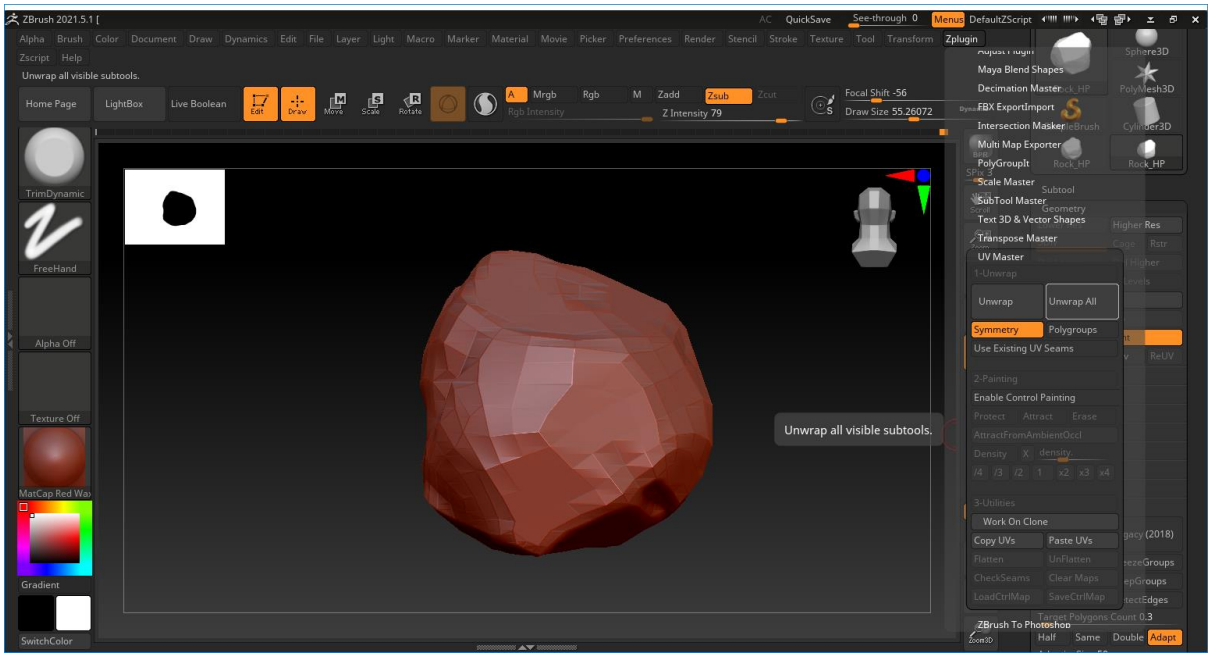


Figure 1100 - Unwrap All.

To view the UVs, go to Zplugin > UV Master and click on Flatten.

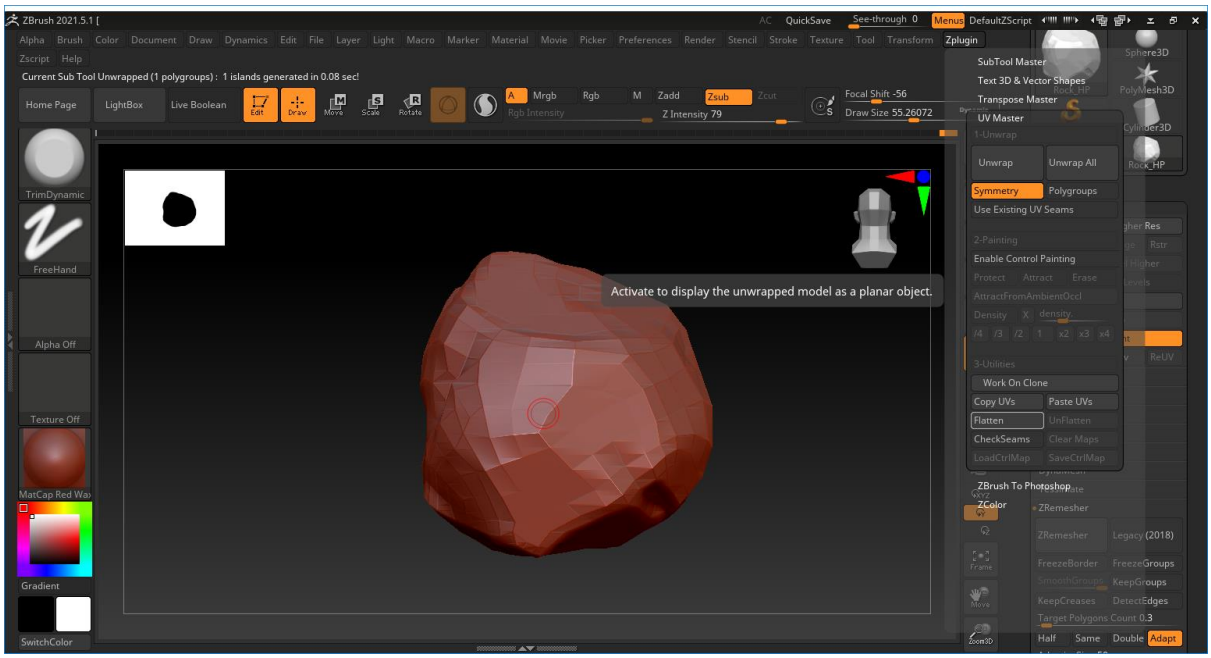


Figure 1101 - Flatten.

Here are our UVs. As you can see there are some problems with stretching and pinching that we'll need to fix.

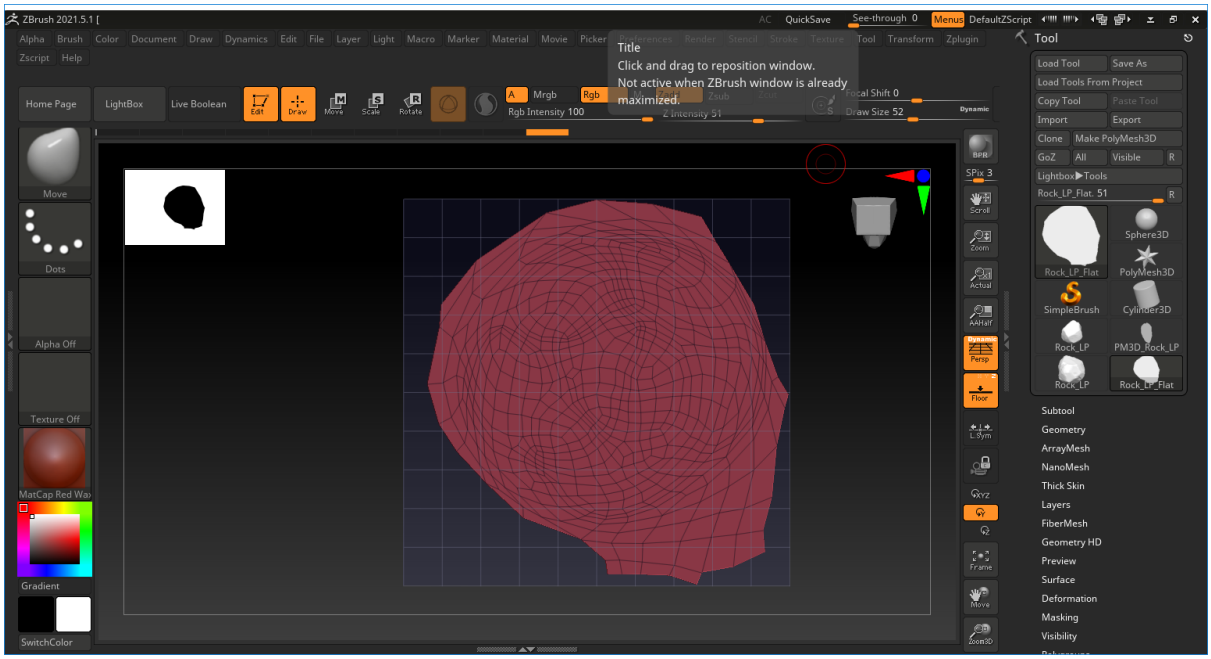


Figure 1102 - UVs.

There are several ways to go about doing this. You can use the Polish slider under Deformation to try and relax the UVs, or use the Move and Smooth brush to adjust and relax specific areas.

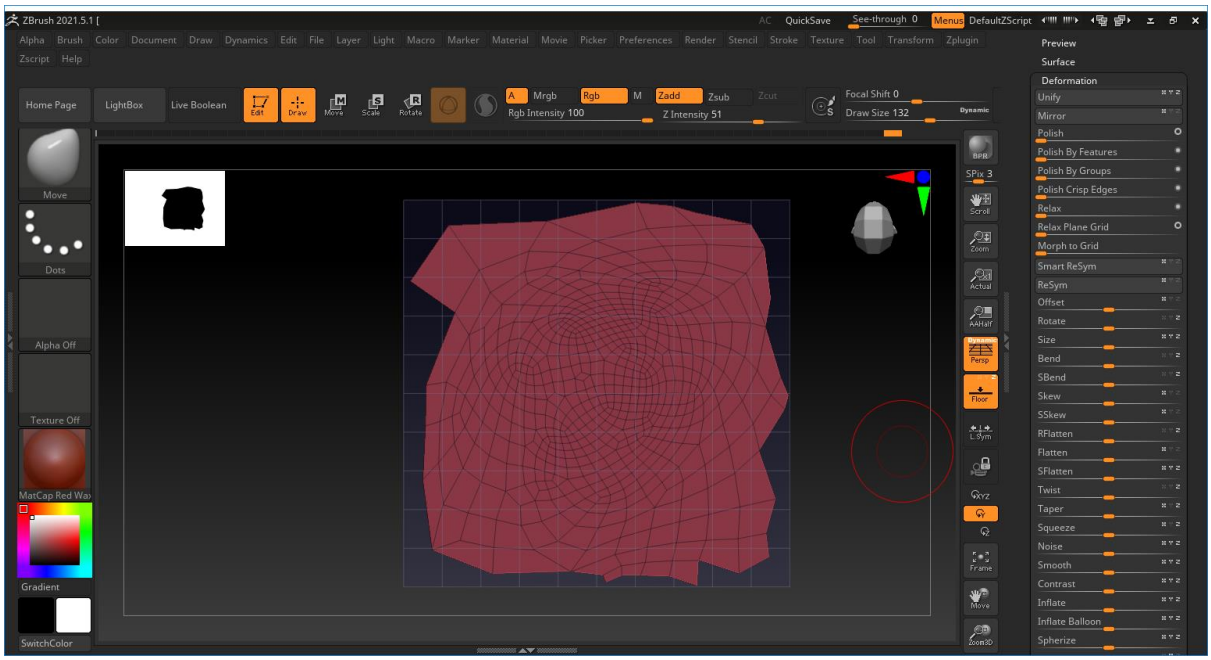


Figure 1103 - Relaxing the UVs.

This is still not *good*, but it's a slight improvement on what we had before.

We'll be adjusting UVs in 3ds Max shortly.

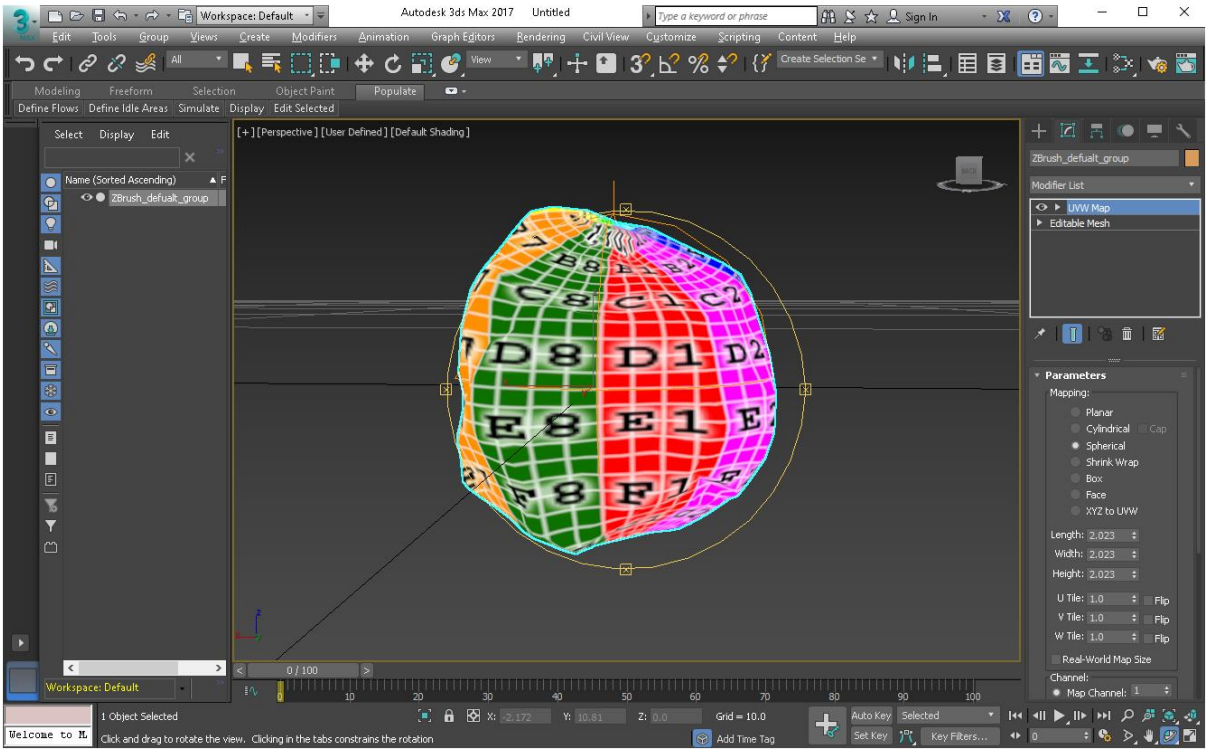


Figure 1104 - UVs in 3ds Max.

When you're done adjusting the UVs in Zbrush, go back to Zplugin > UV Master and click on Unflatten.

Click on Export again and save the low poly version of the asset. I added a '_LP' extension to differentiate it from the high poly version.

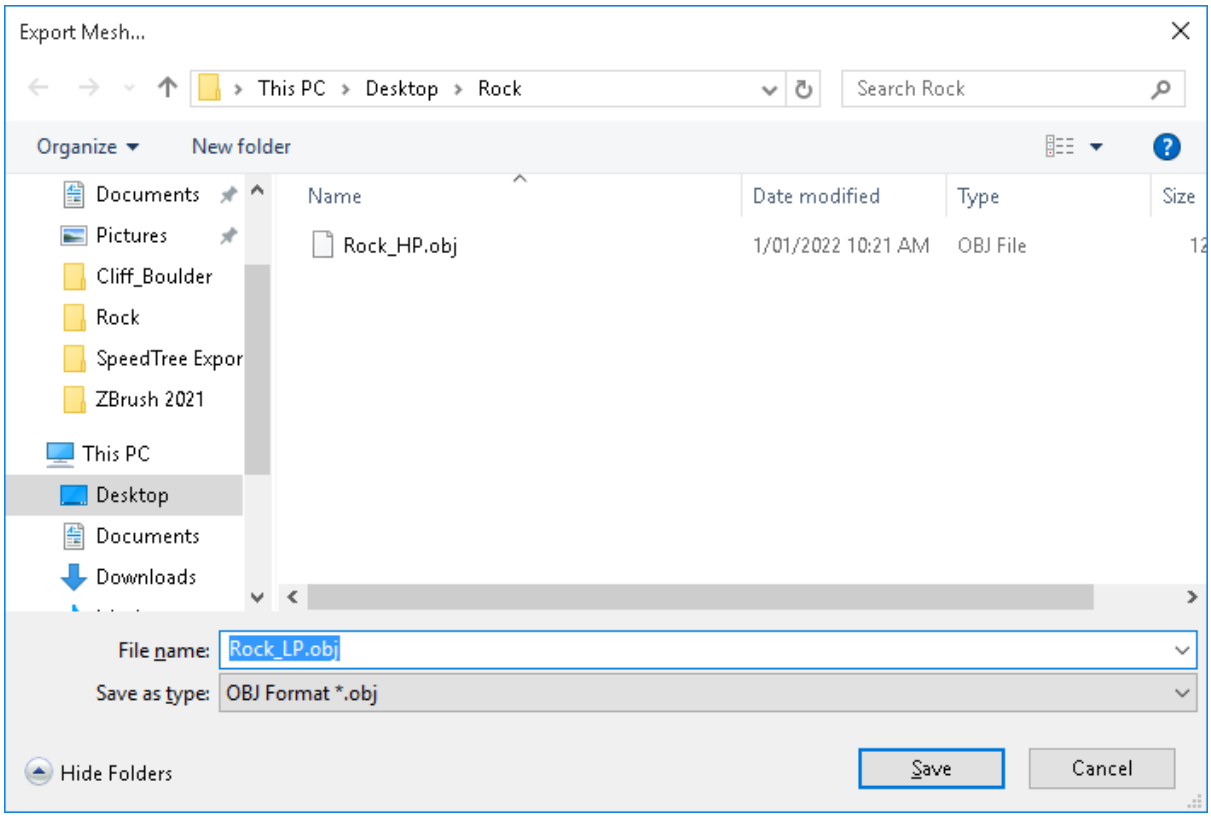


Figure 1105 - Saving the low poly asset.

Like I mentioned previously, we can adjust the UVs in another program which, for this example, will be necessary before we bring the asset into Substance Painter to apply a material. Organic shapes are generally a bit trickier to unwrap.

Note: I'm not an expert at 3d modelling and there are probably better methods for fixing UVs. I'd recommend taking a look around for plugins that simplify the process, but for the following example I'm just going to use the standard tools in 3ds Max.

So let's bring the model into 3ds Max first to try and tweak the UVs.

Open 3ds Max and go to Import.

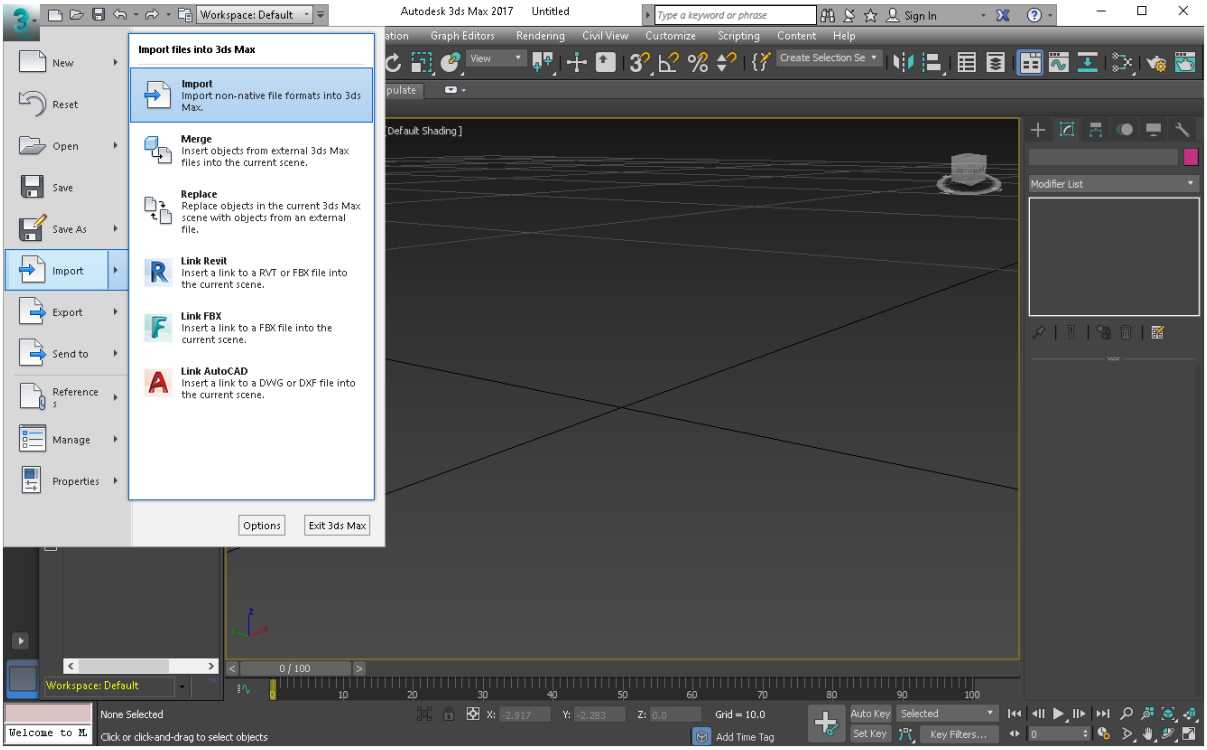


Figure 1106 - Import.

Select the low poly .obj and click Open.

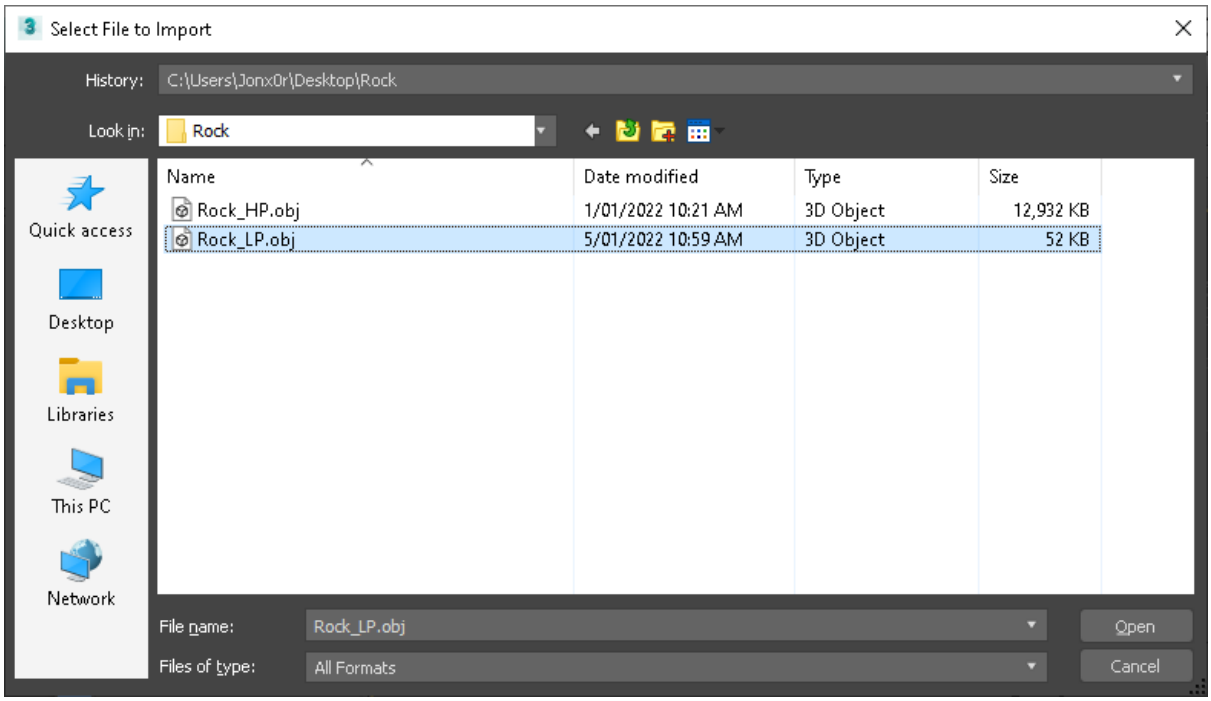


Figure 1107 - Importing the low poly .obj.

Tick 'Import as single mesh' and 'Import as Editable Poly'.

Under Geometry, ensure 'Flip ZY-axis' and 'Texture coordinates' are ticked then click Import.

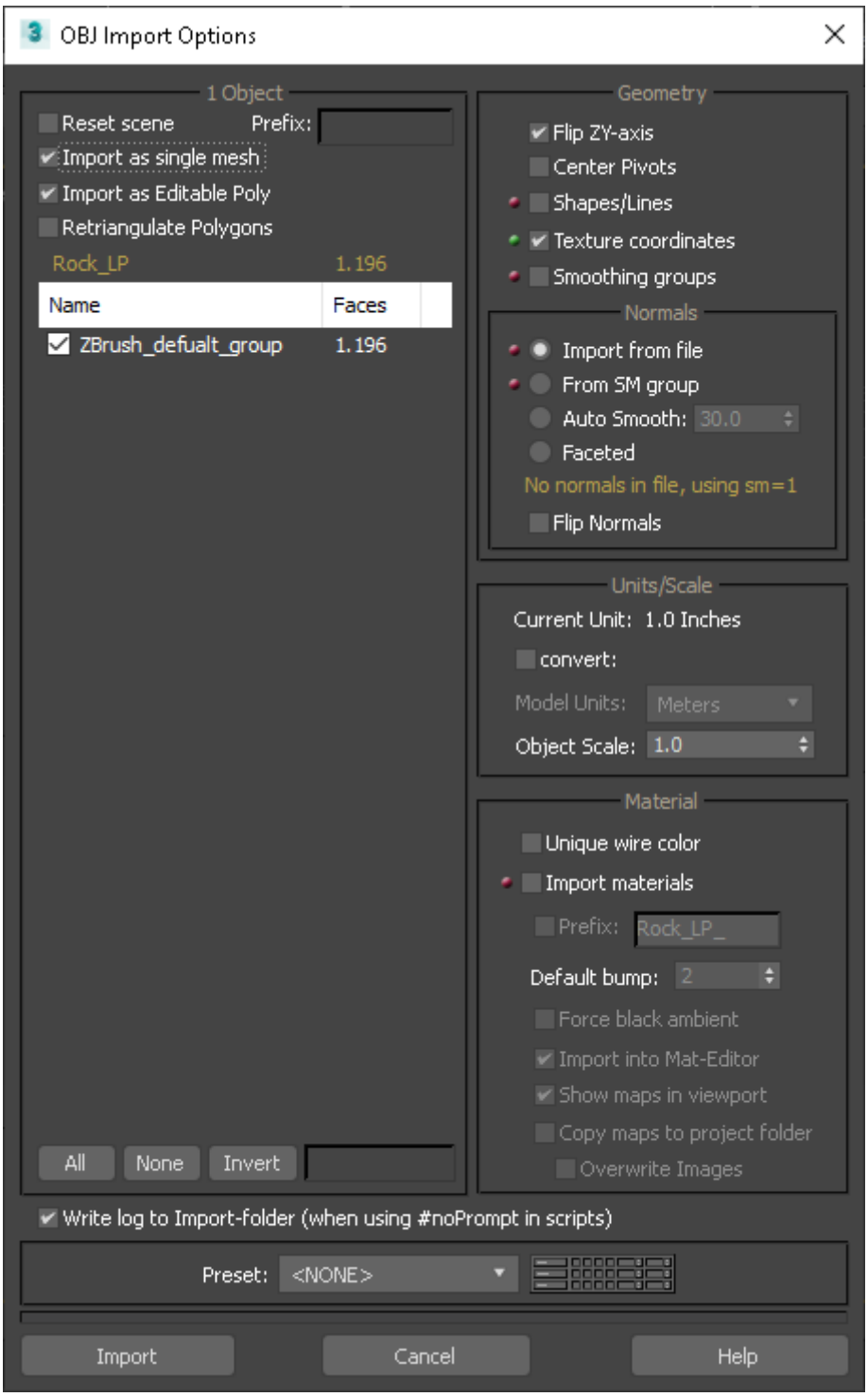


Figure 1108 - .obj import settings.

Here's our low poly asset in 3ds Max.

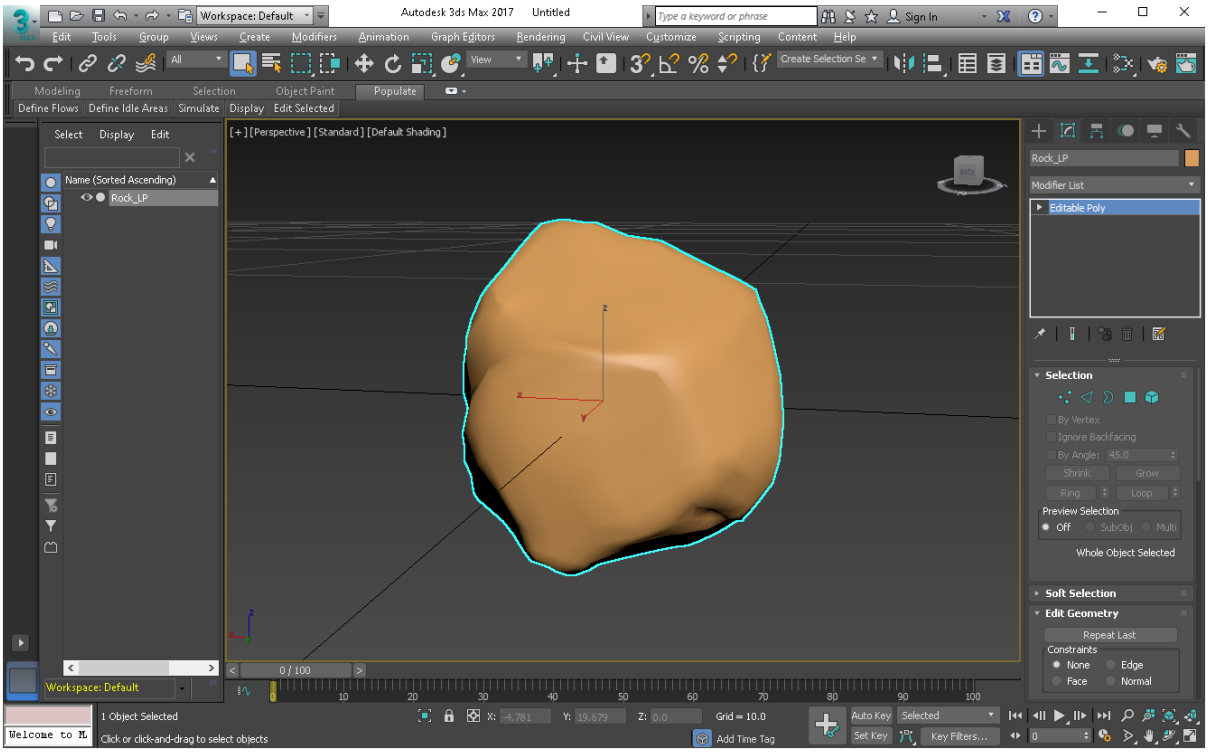


Figure 1109 - .obj imported into 3ds Max.

To quickly see UVs, click on Standard and go to Materials > Override with UV Checker.

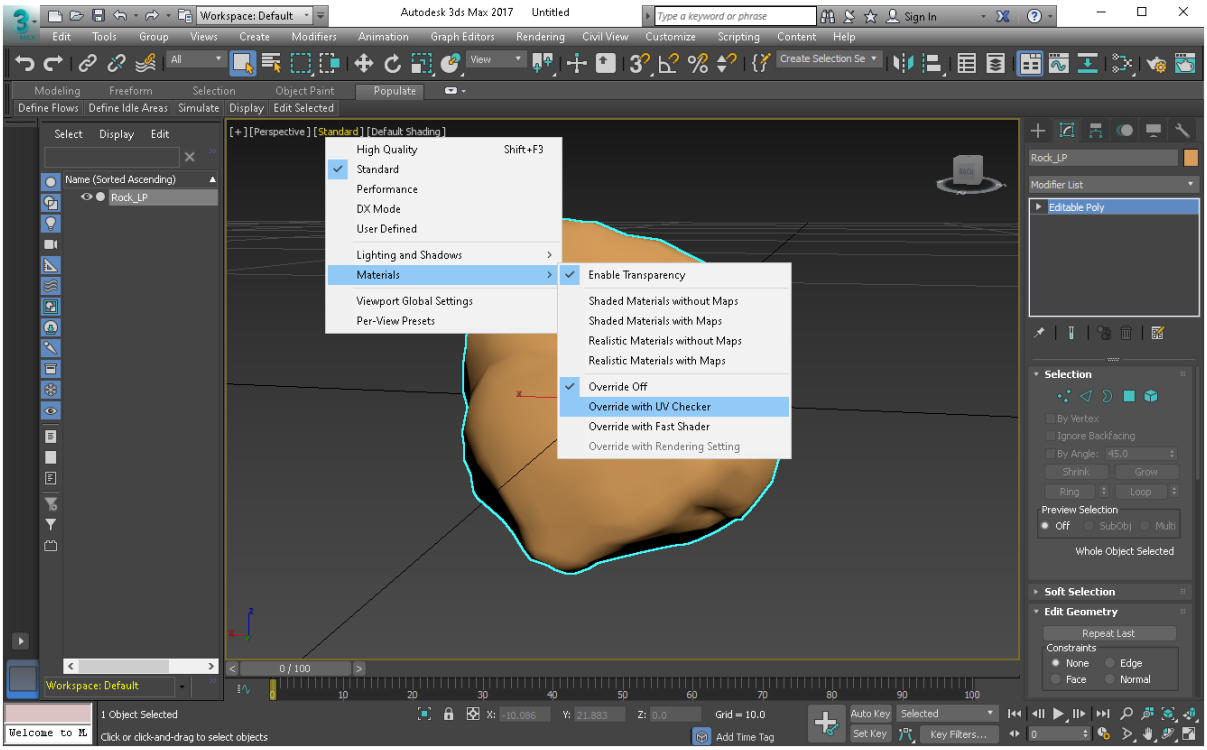


Figure 1110 - Enabling UV Checker material override.

We should now be able to see the UVs on our model. As you can see, some work needs to be done.

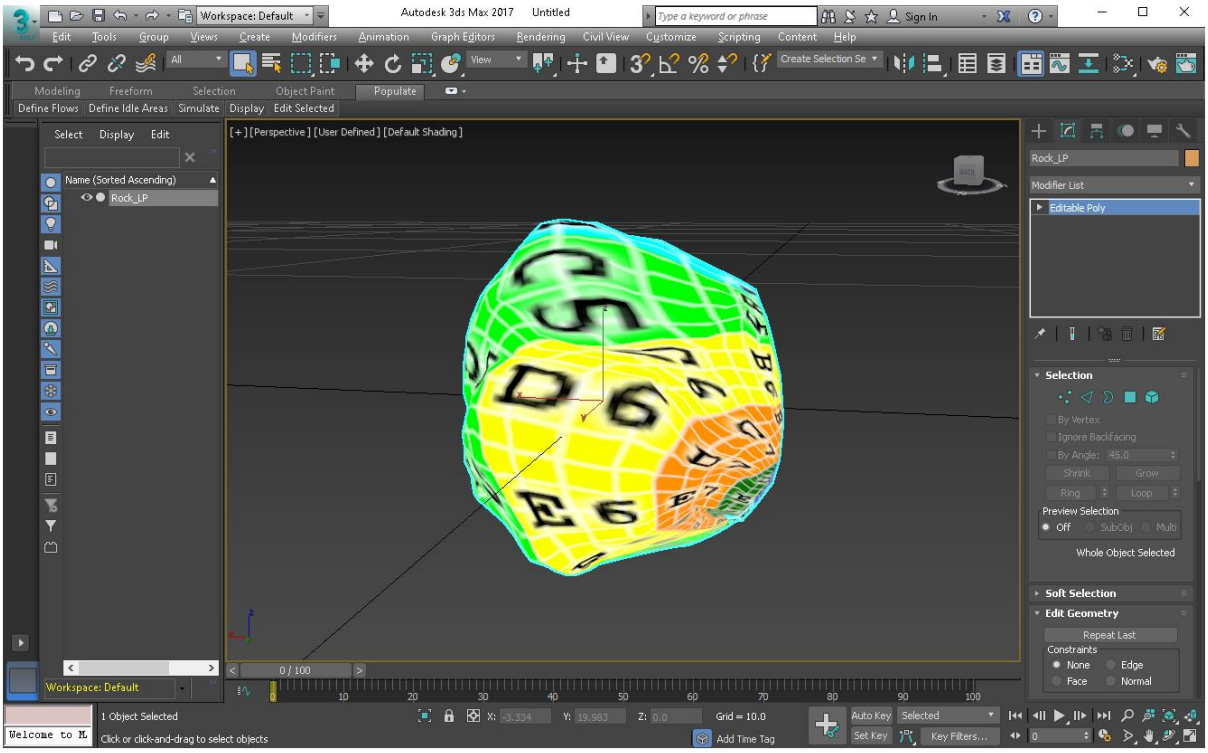


Figure 1111 - UV material override.

From the Modifier List, add an Unwrap UVW modifier.

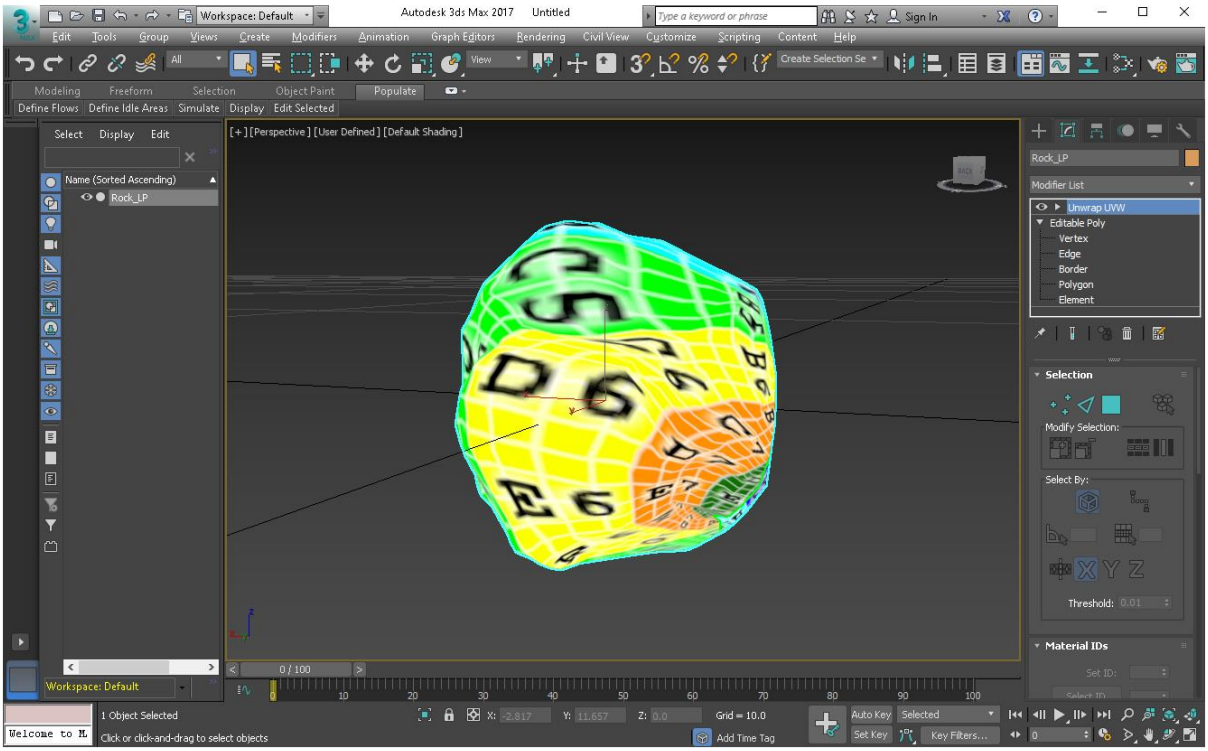


Figure 1112 - Unwrap UVW modifier.

Scroll down and click on the Open UVW Editor button.

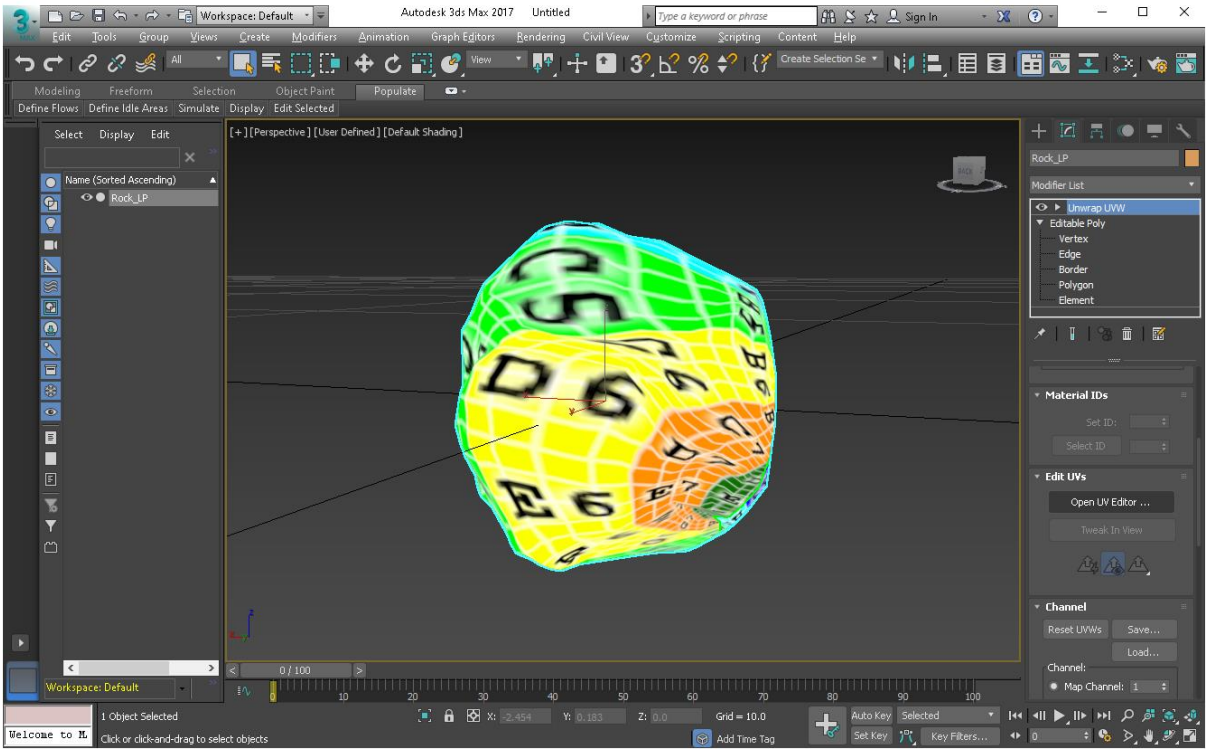


Figure 1113 - Open UVW Editor.

We should be able to see the UVW map here.

Under the Peel section, click on the Reset Peel button.

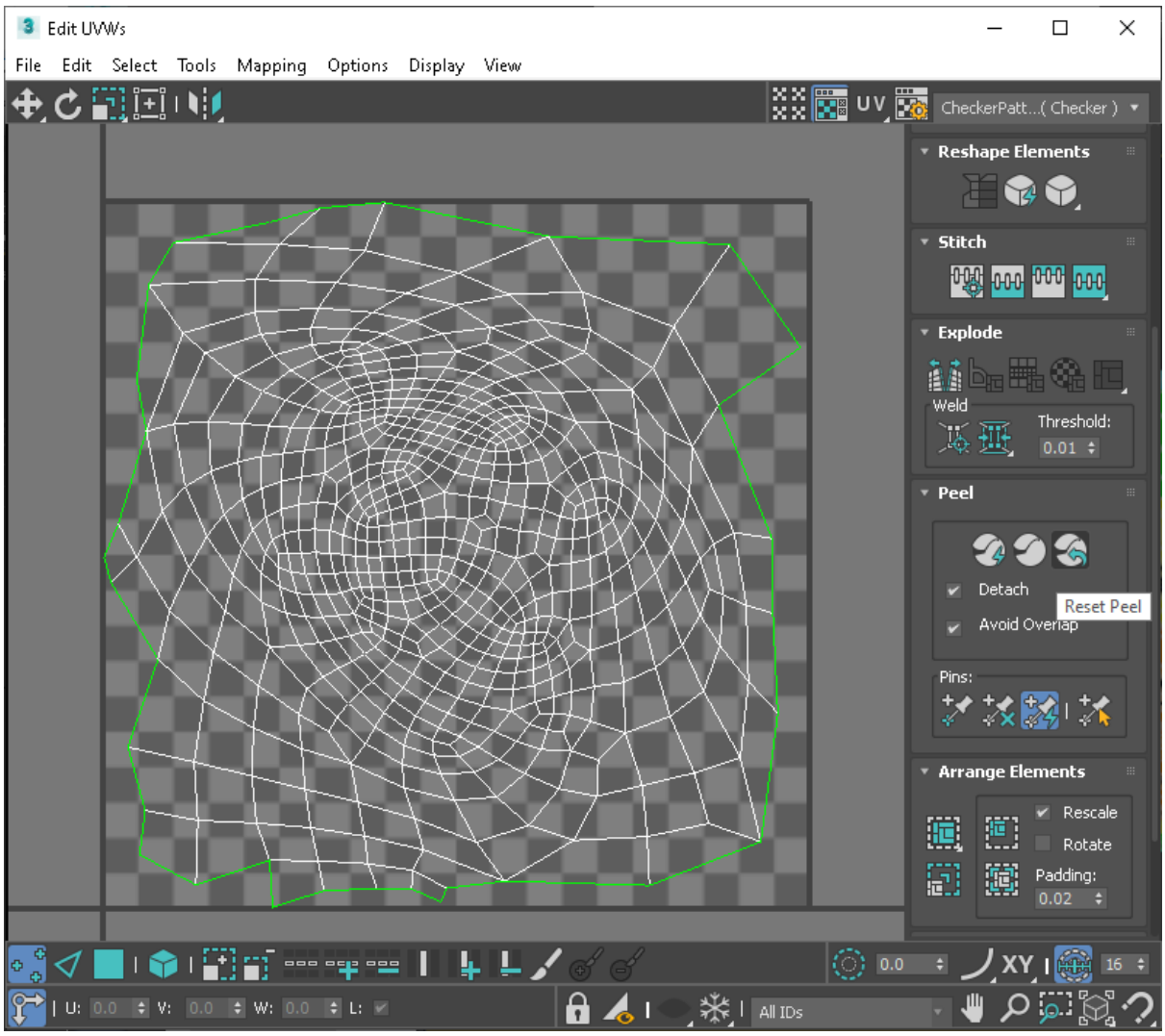


Figure 1114 - Reset Peel.

This should reset our UVs like so.

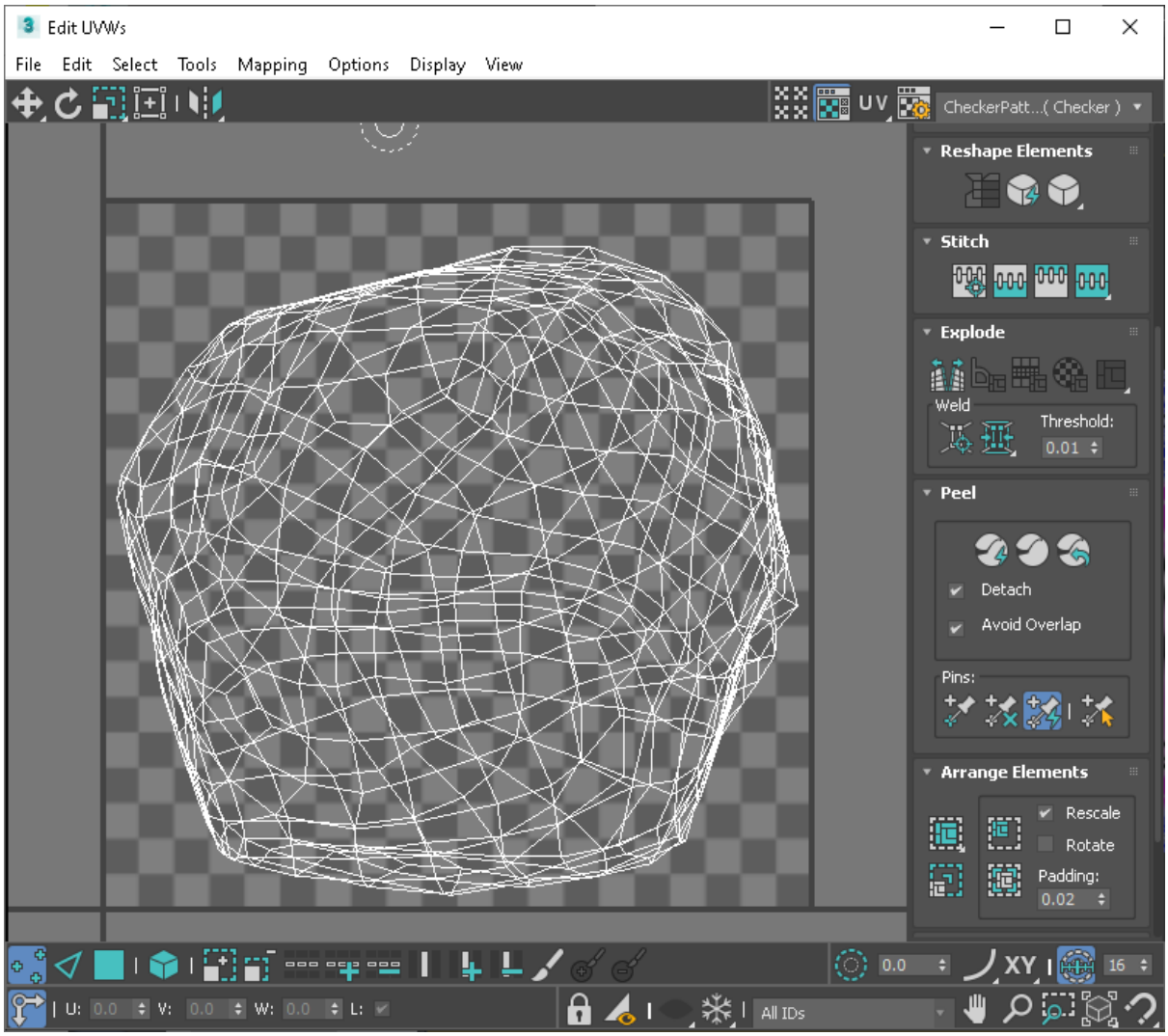


Figure 1115 - UVs reset.

For this example I'm going to cut the rock in half using the Point-to-Point Seams tool. Generally you'd want to follow hard edges, but for a naturally shaped object like a rock it's a bit less obvious where the seam should go.

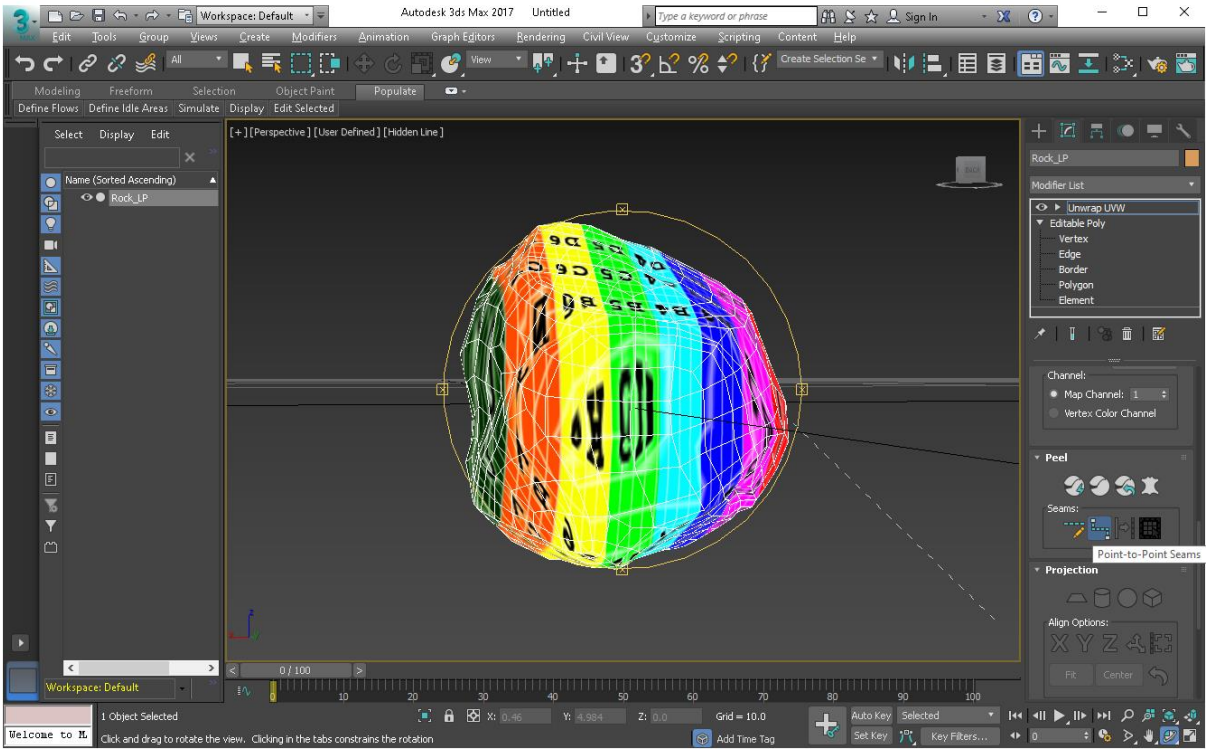


Figure 1116 - Point-to-Point Seams tool.

This seam will affect how the UVWs will unwrap when we peel them next.

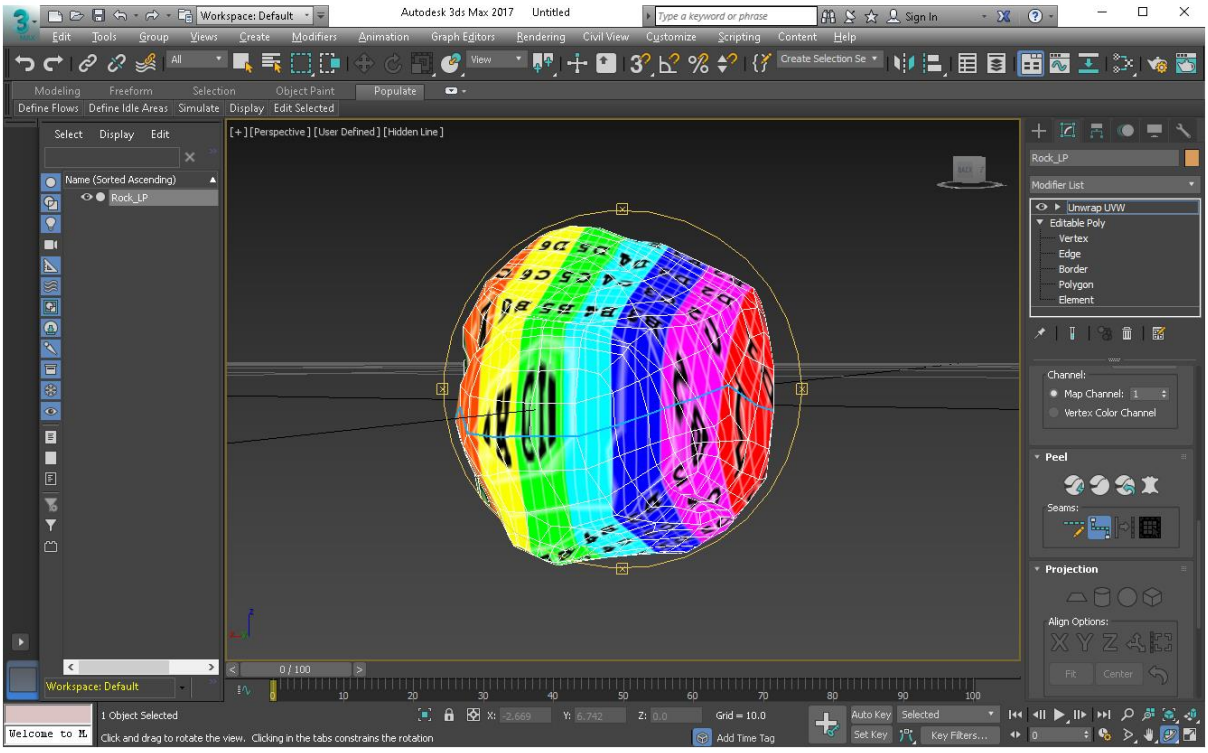


Figure 1117 - Seam added to UVW.

Once the seam has been added, click on Quick Peel.

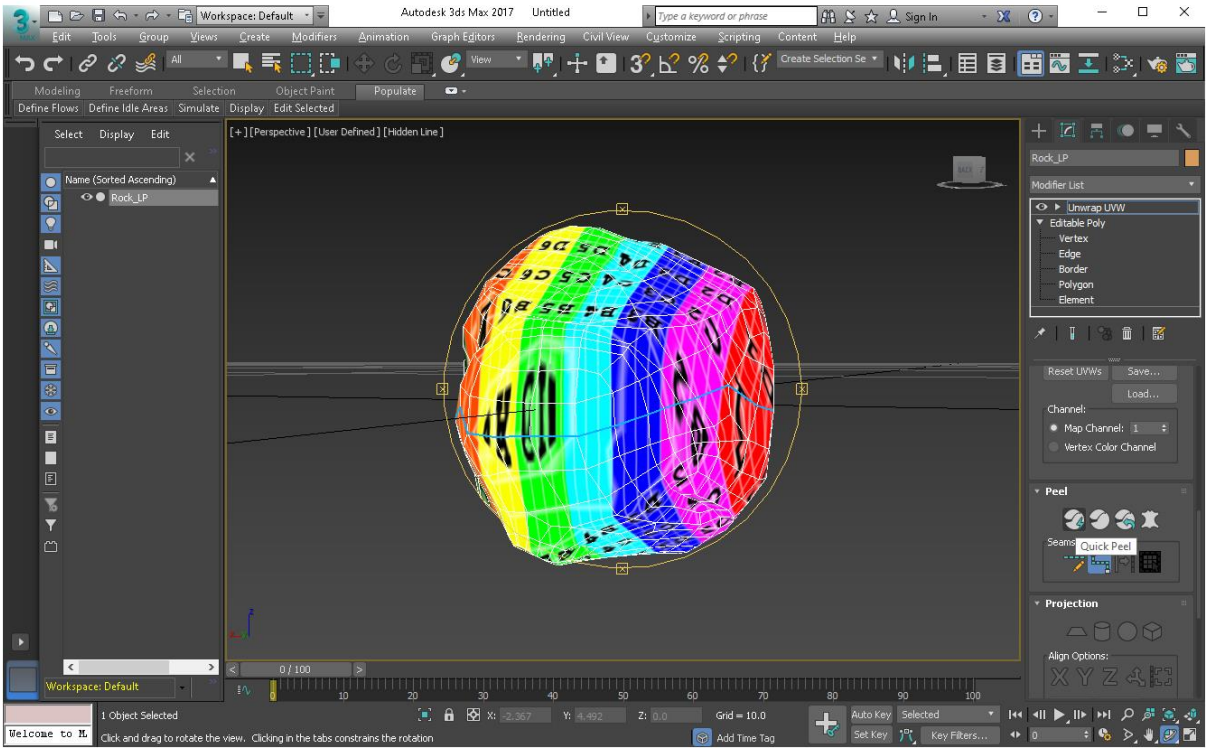


Figure 1118 - Quick Peel.

The seam should've split the UVWs into two separate islands like so.

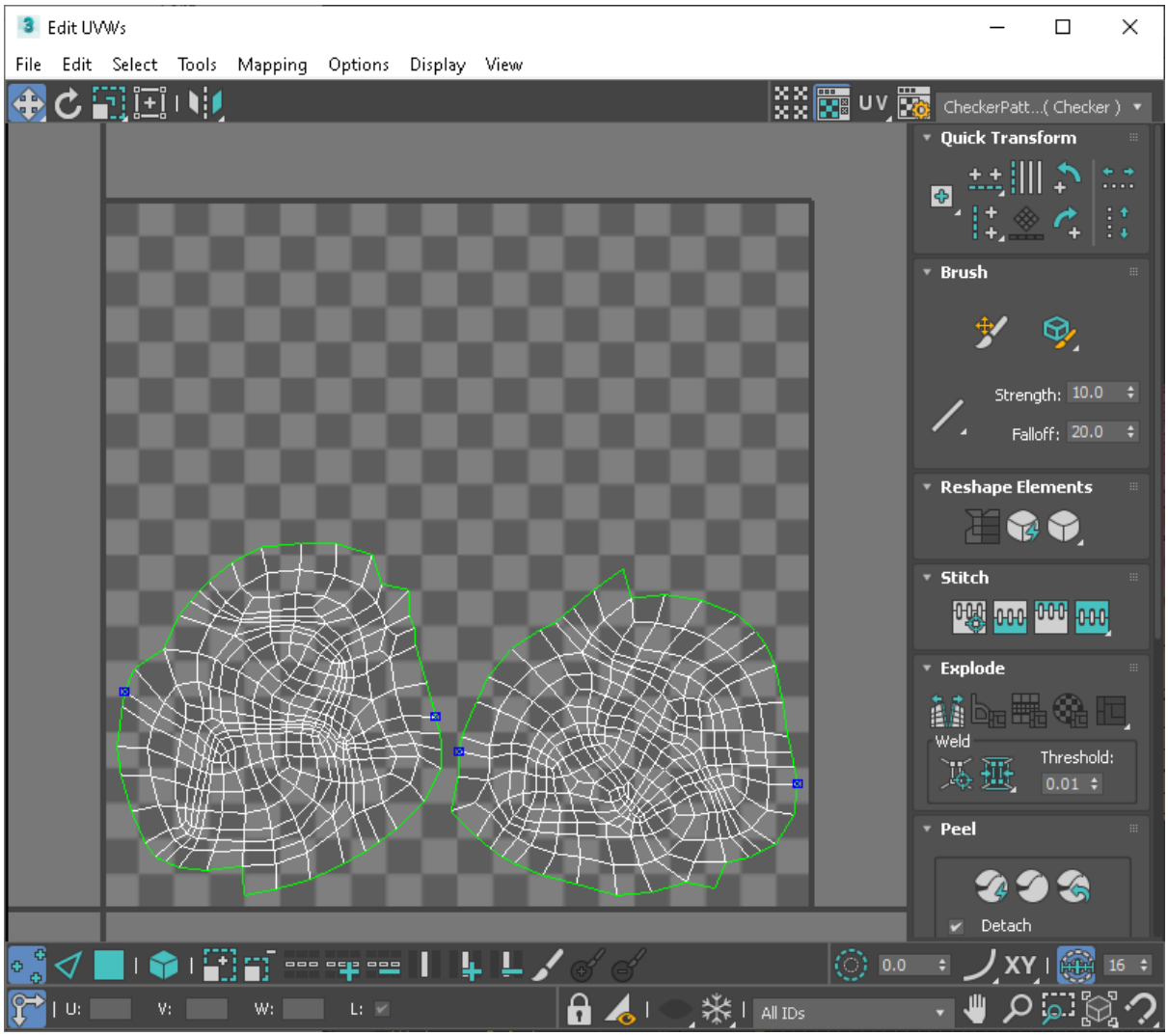


Figure 1119 - UVWs with seam.

Firstly, turn the viewport material override off by going to Materials > Override Off.

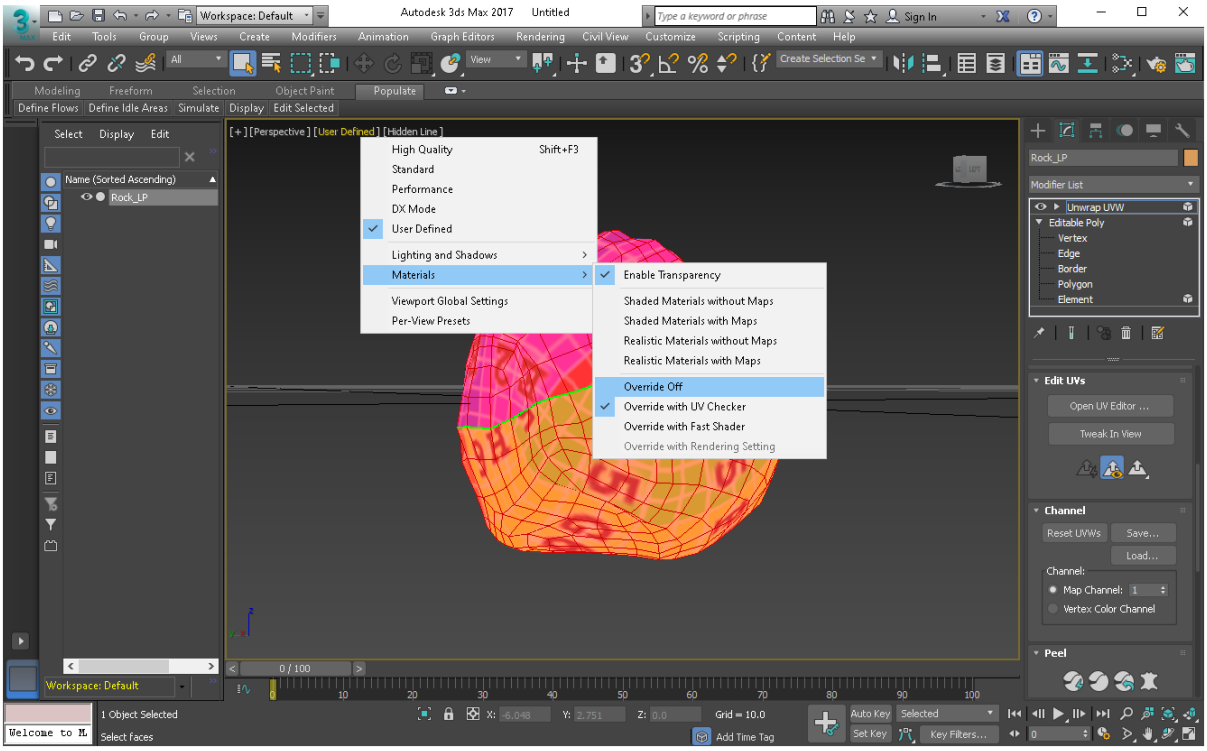


Figure 1120 - Turning off the material override.

In the UVW editor, select CheckerPattern. You might need to select Texture Checker first then switch back to CheckerPattern in order to update the texture on the model in the viewport.

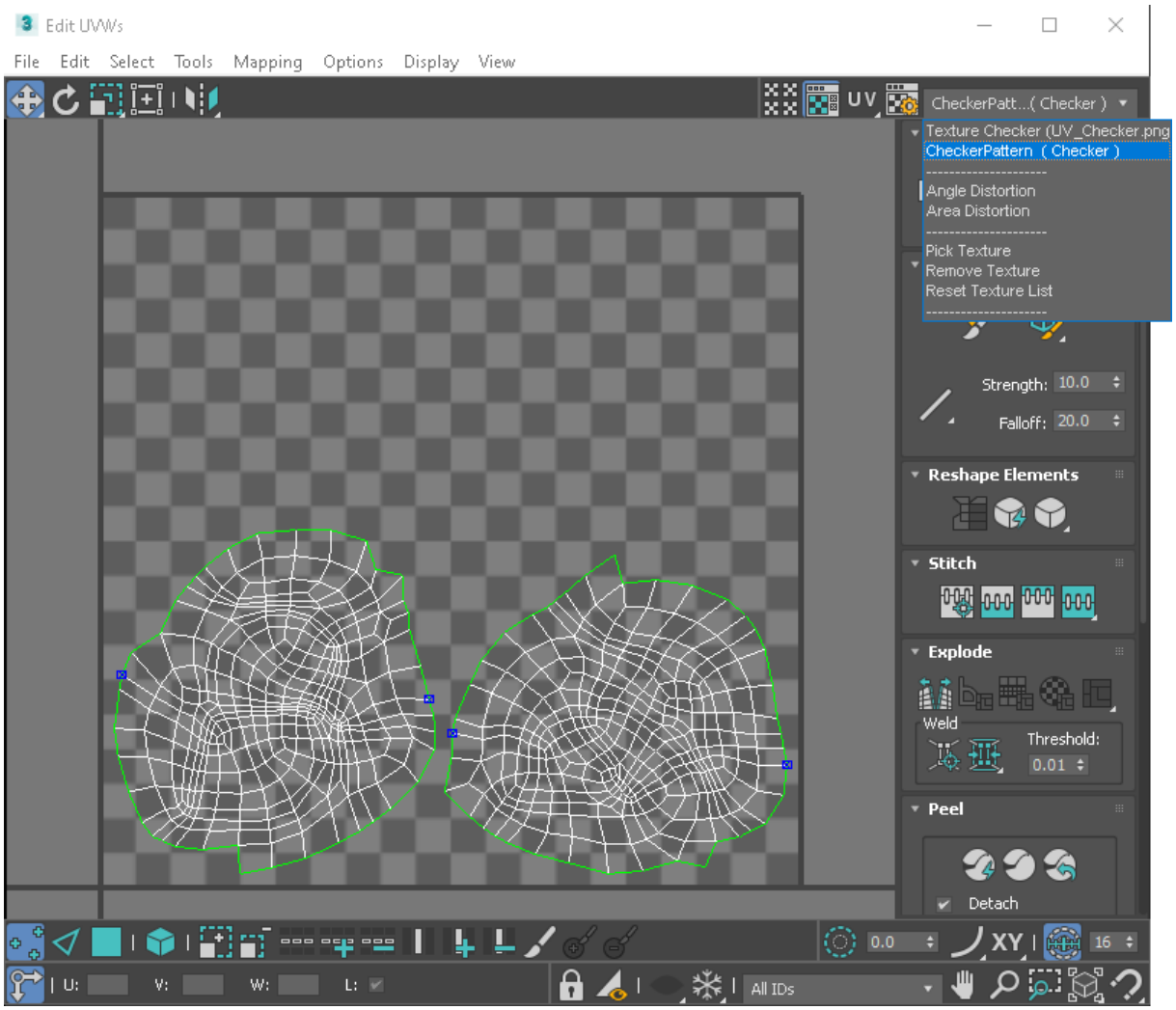


Figure 1121 - Checker Pattern.

The checker pattern should help us visually gauge whether there's any pinching or stretching that needs to be fixed in our UVWs.

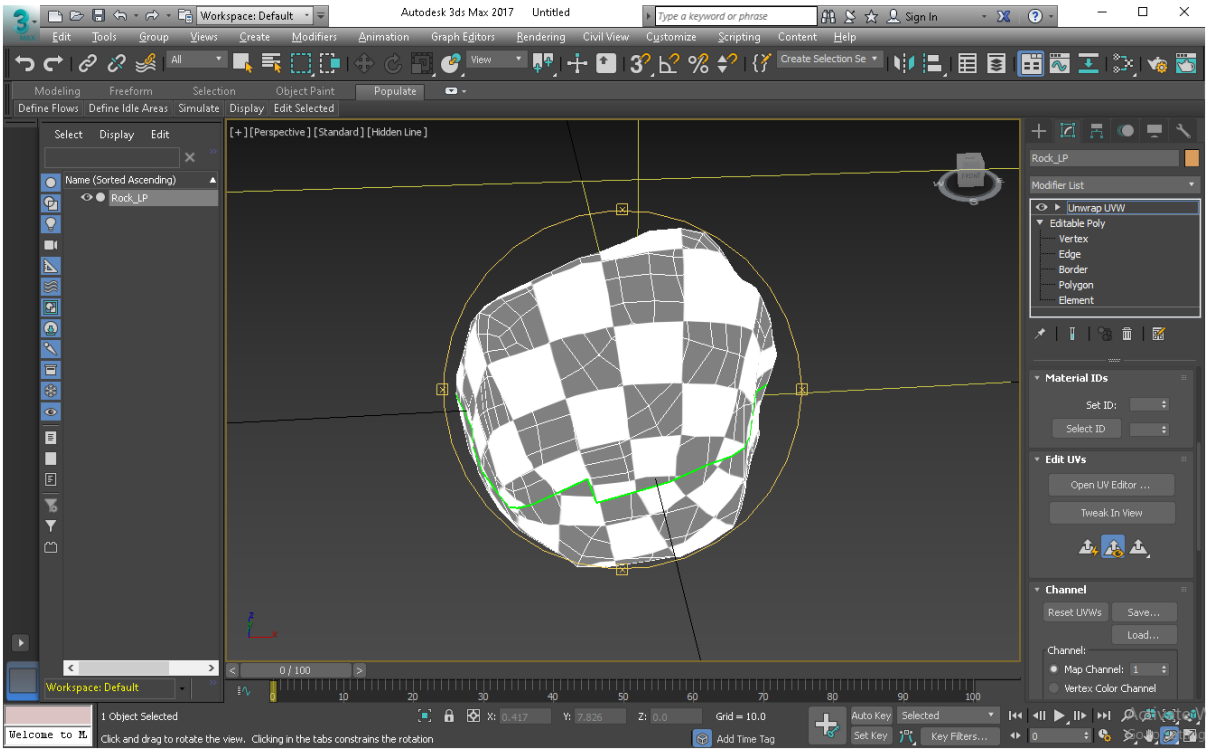


Figure 1122 - Checker pattern in viewport.

In the UVW editor, go to Tools > Relax.

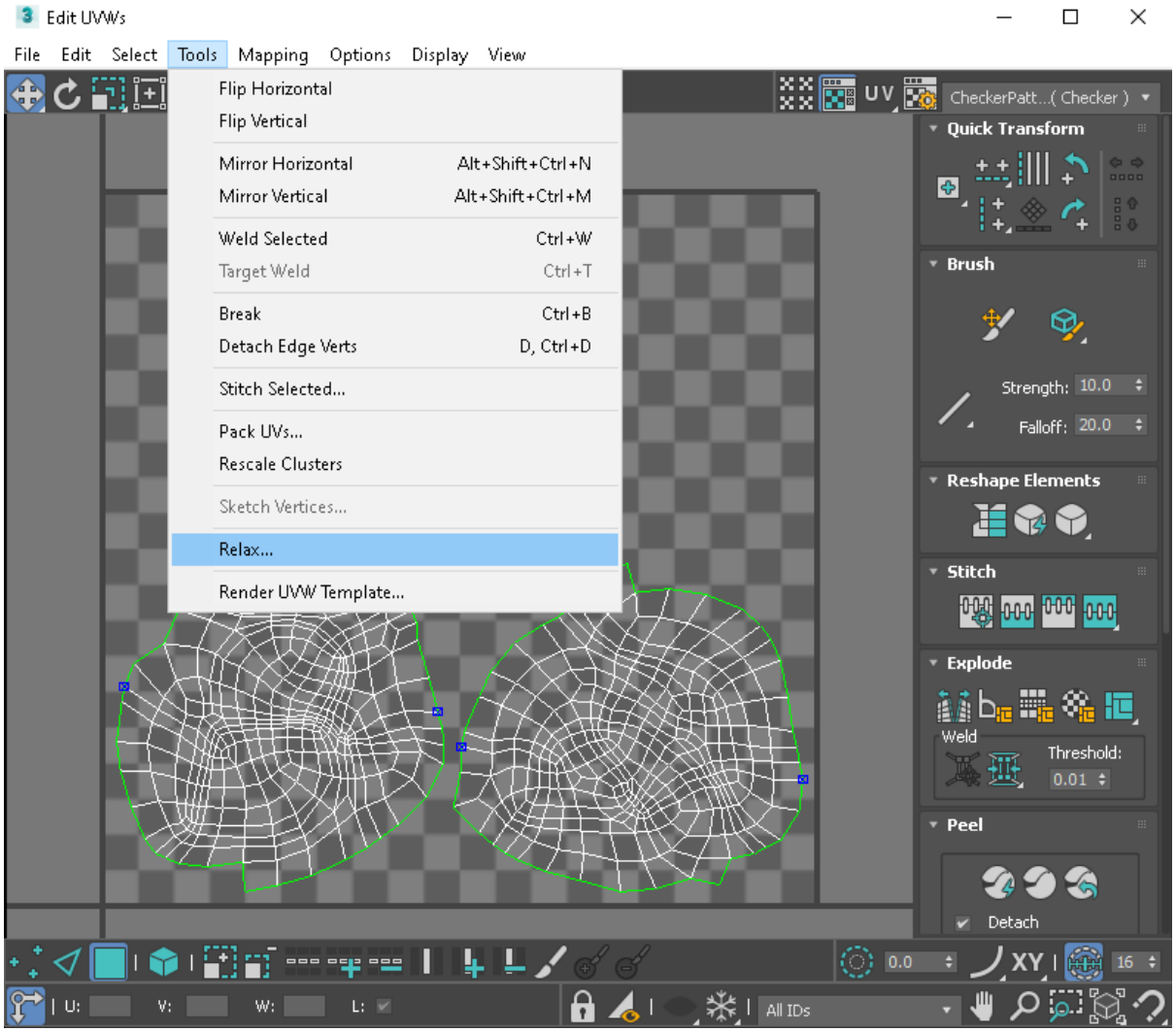


Figure 1123 - Relax UVW.

Ensure the drop-down is set to 'Relax By Polygon Angles' then click on Start Relax.

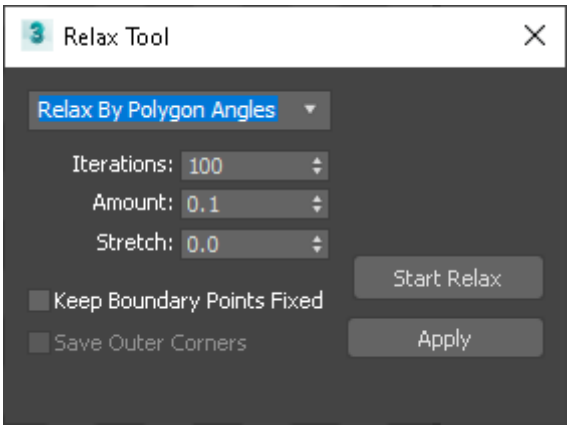


Figure 1124 - Relax settings.

Once the quads have spread themselves a bit more evenly, click Stop Relax.

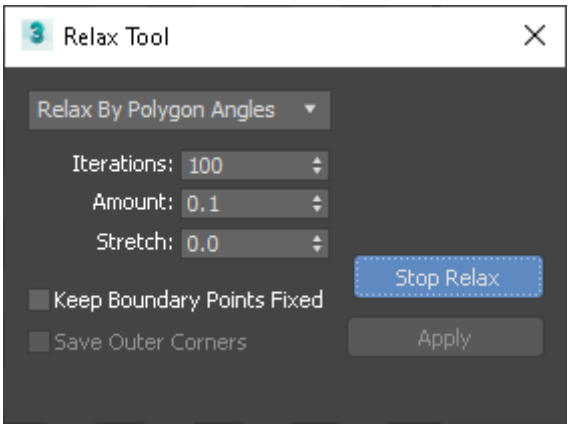


Figure 1125 - Stop Relax.

Close out of the Relax Tool.

Set the drop-down to Angle Distortion.

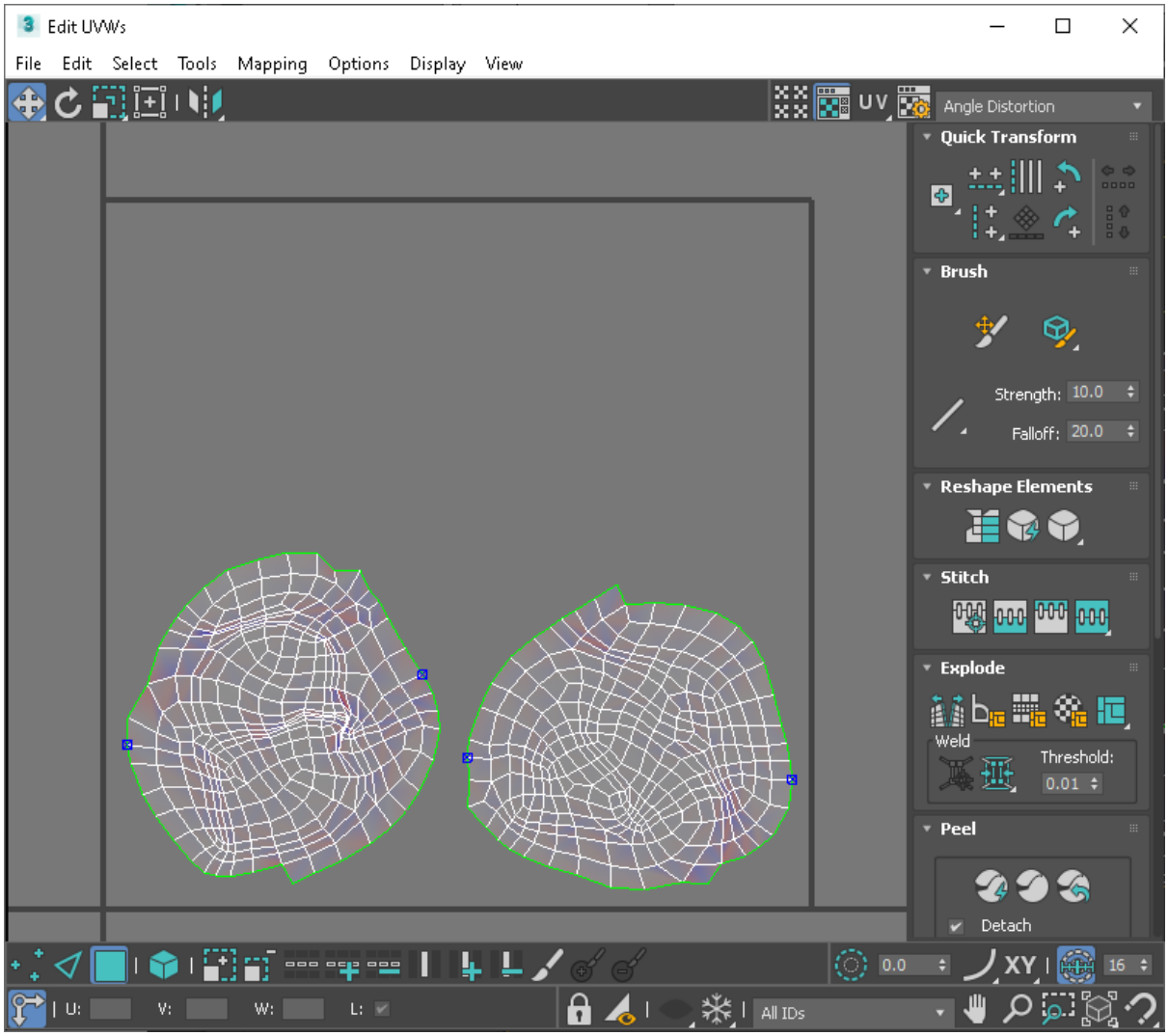


Figure 1126 - Angle distortion.

This will show us where the UVWs are getting pinched or stretched.

As you can tell by the red and blue, there's some stink happening in our UVWs. The brighter the colour, the stronger the distortion.

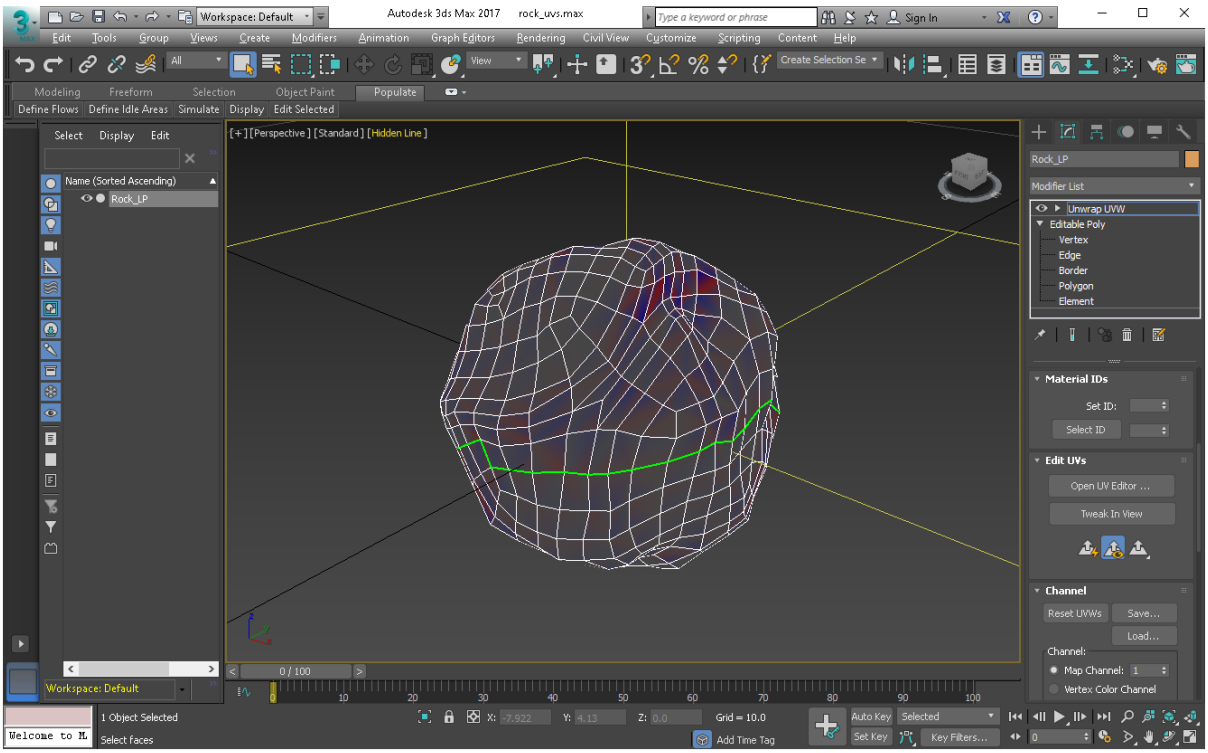


Figure 1127 - Angle distortion showing pinching and stretching.

You can use the UV Paint Movement tool to move vertices around to try and fix problematic areas manually. The colours will update automatically as you adjust things.

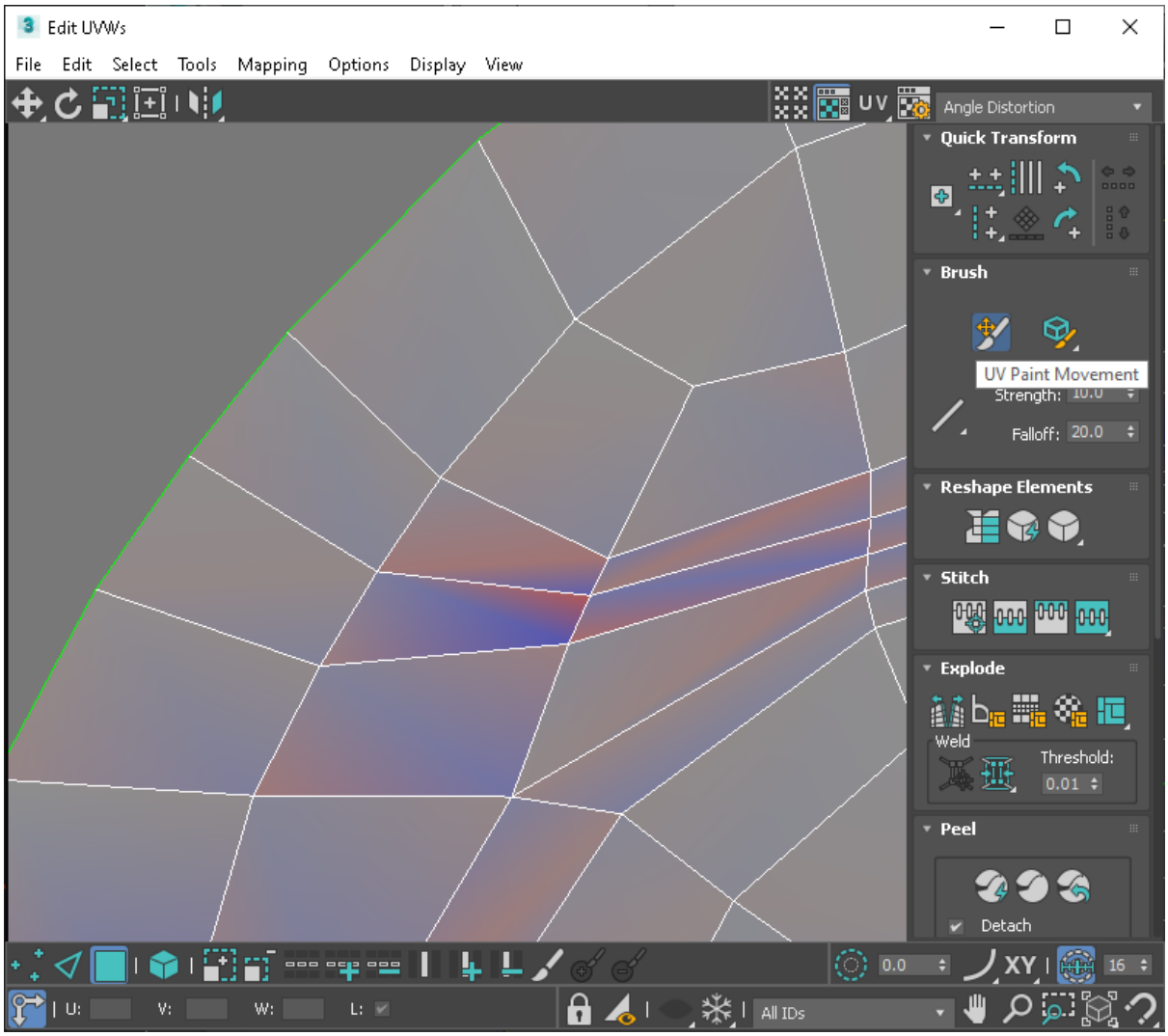


Figure 1128 - UV Paint Movement.

I added a couple extra seams near the most problematic areas and did another Quick Peel which seems to have eliminated most of the distortion.

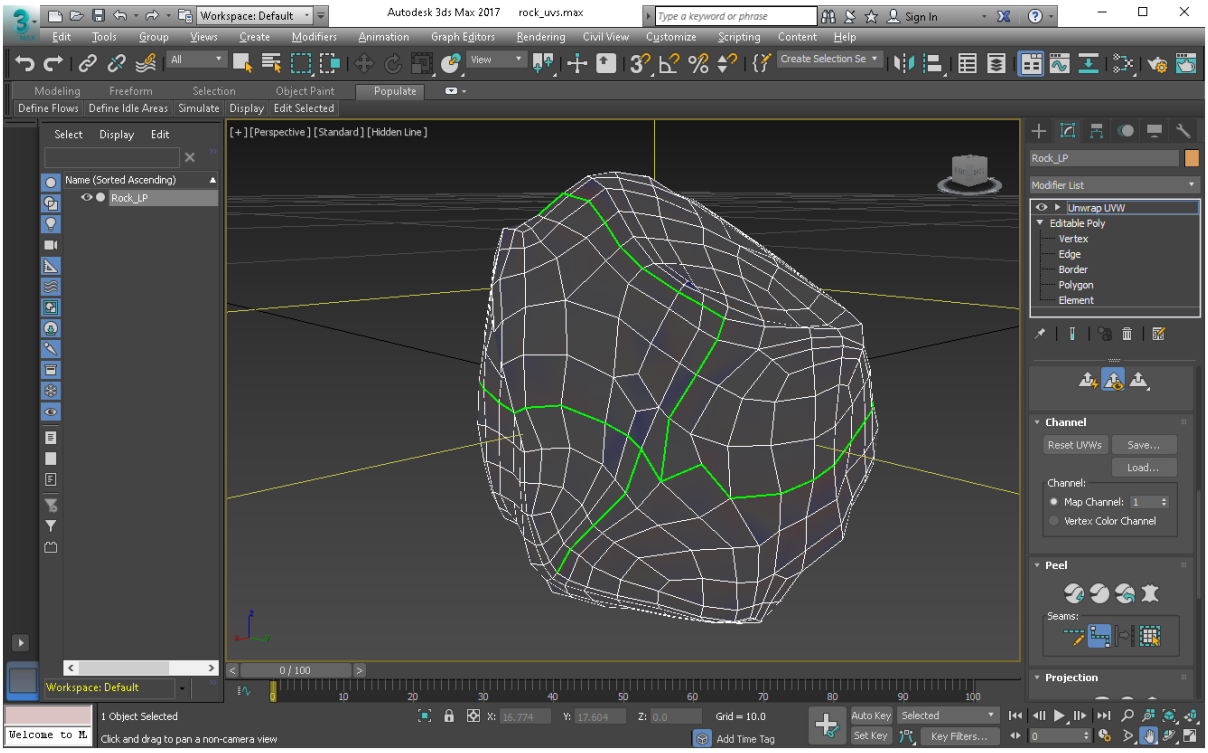


Figure 1129 - Added extra seams.

After running Relax again, the UVWs now look like this:

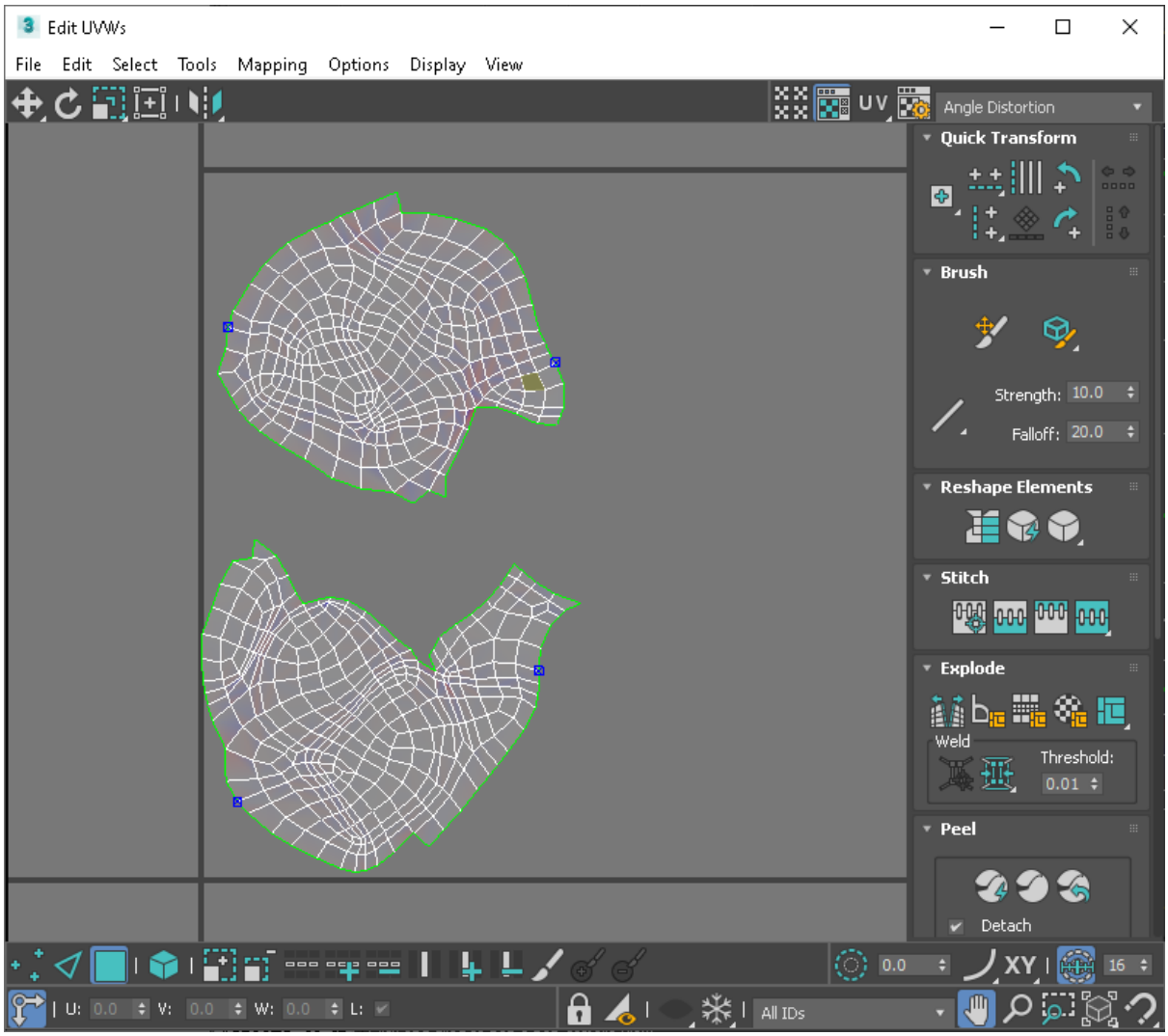


Figure 1130 - UVWs with distortion mostly fixed.

Setting the drop-down to Area Distortion shows us some mild distortion around the sharper edges of the rock and along the seam.

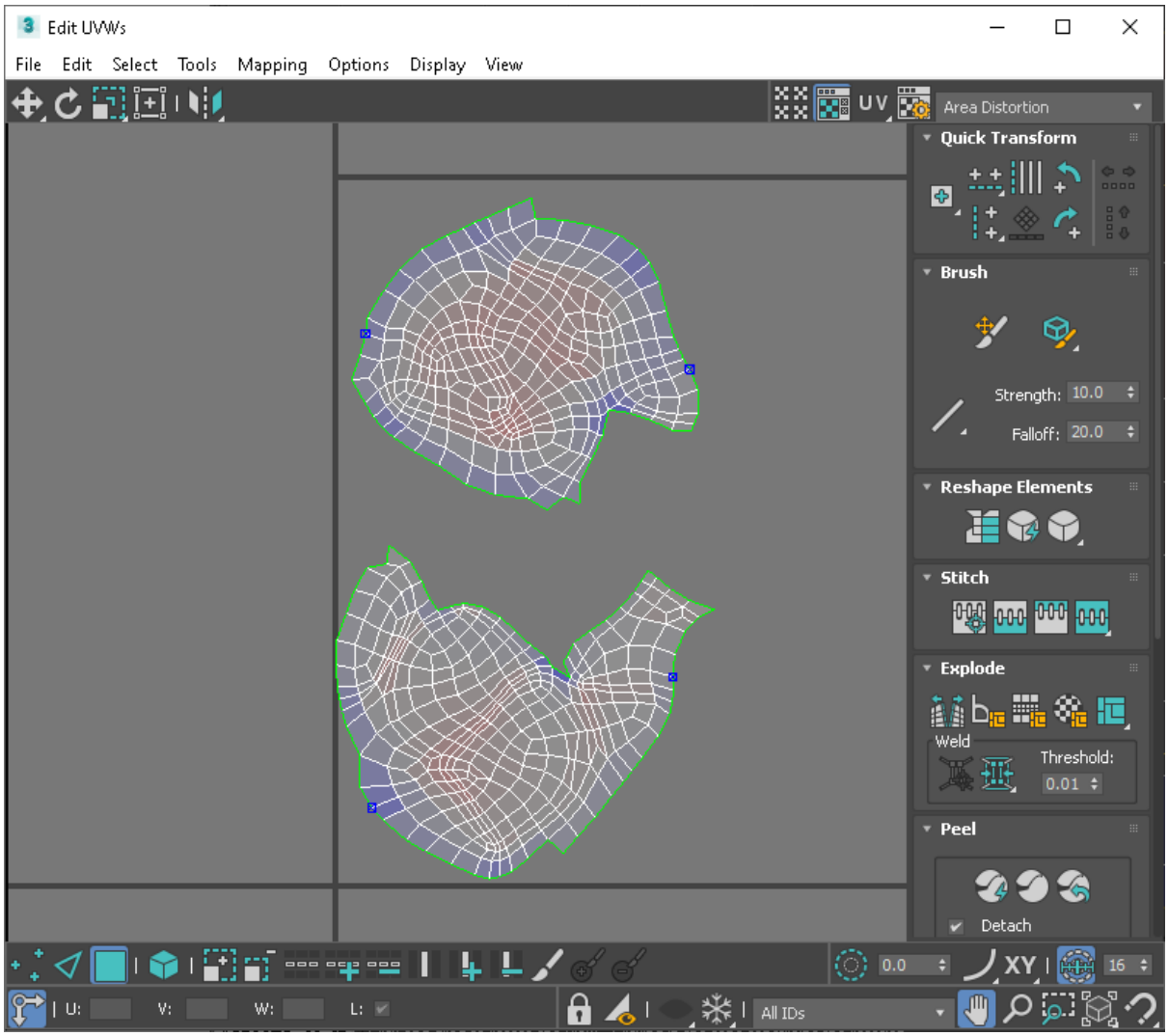


Figure 1131 - Area Distortion.

Once you're done tweaking the UVW map, scroll down to Arrange Elements.

Ensure Rescale and Rotate are ticked then click on the Pack Normalize button.

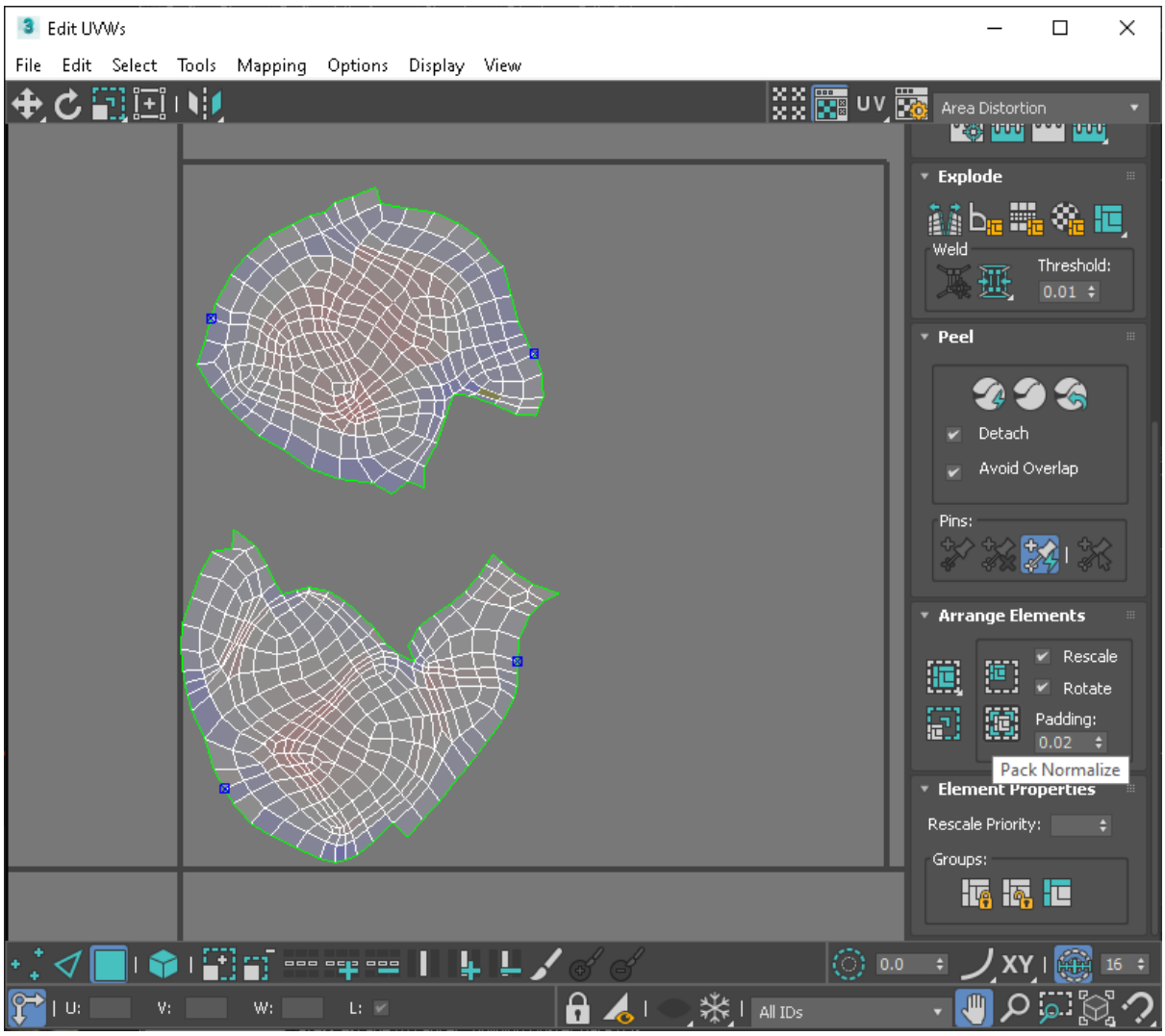


Figure 1132 - Pack Normalize.

This will attempt to make scale up the UVWs and rotate them so they use up as much of the available space as possible. It's not perfect but you can manually adjust the scale and rotation after the fact.

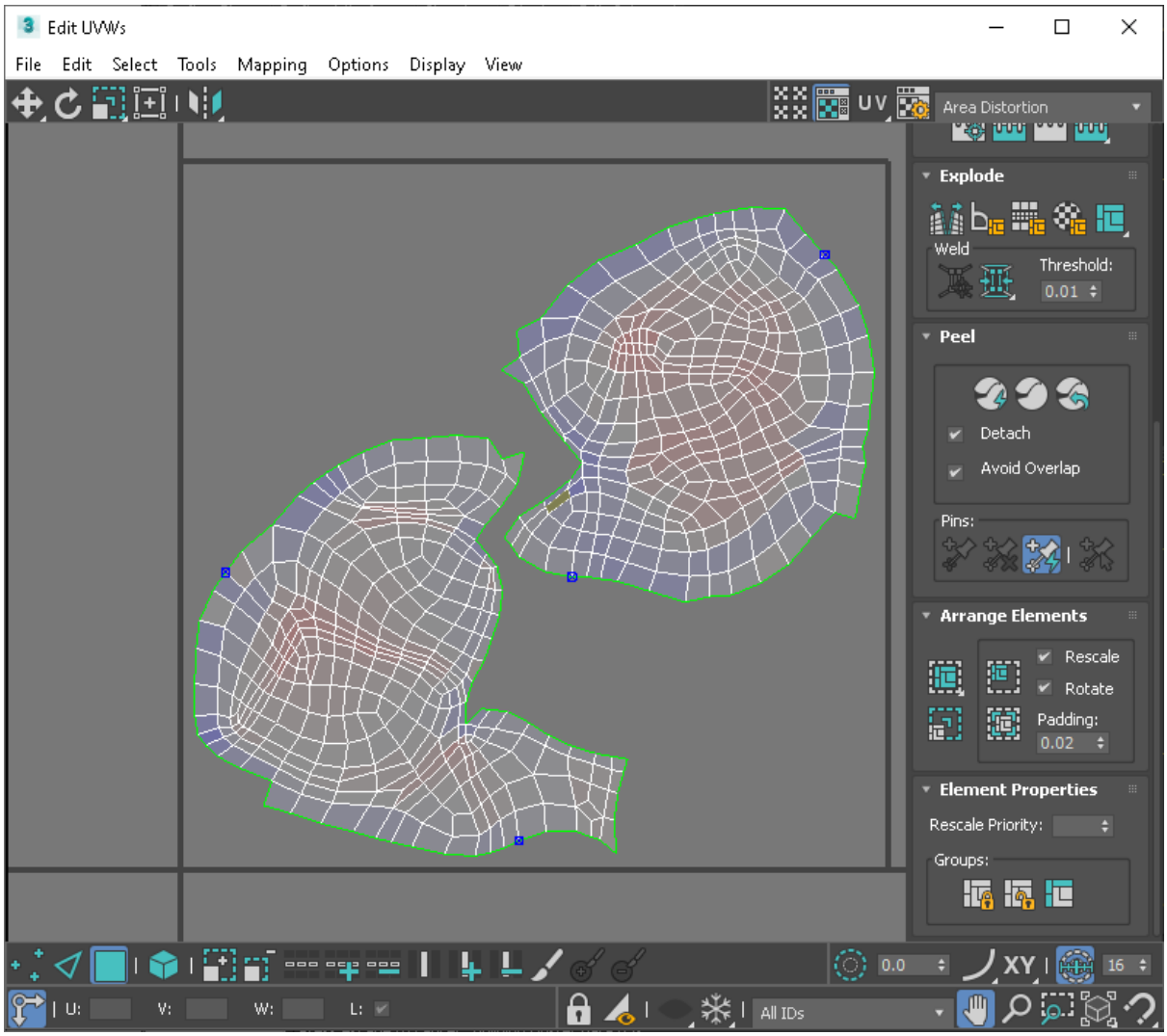


Figure 1133 - UVWs repacked.

Close Edit UVW, go back to the Modifier stack, right-click on Unwrap UVW and select Collapse All.

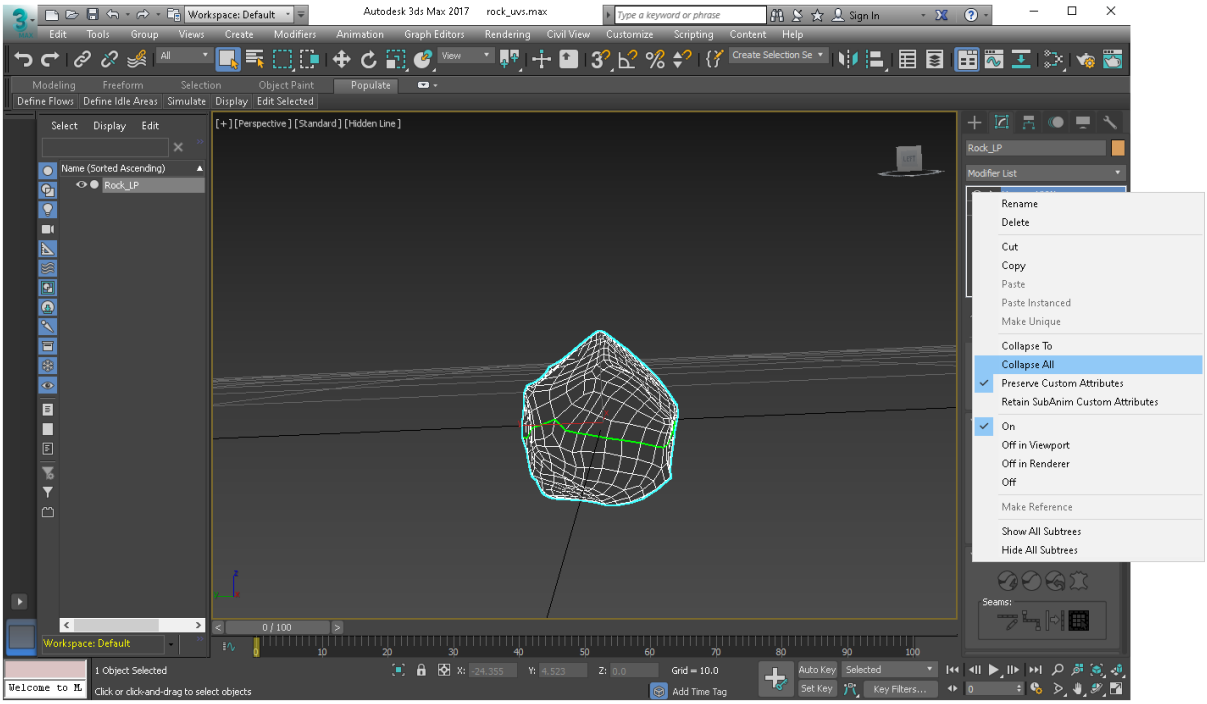


Figure 1134 - Collapse All.

This will collapse the Modifier stack down to Editable Poly.

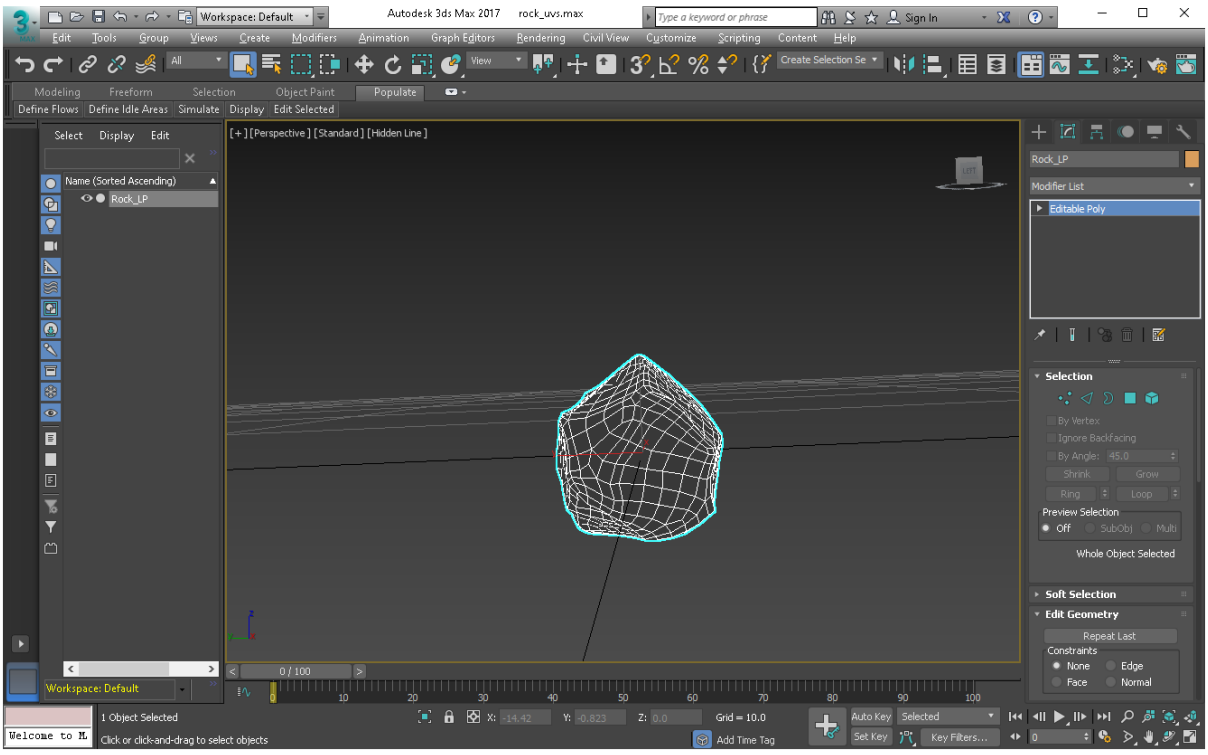


Figure 1135 - UVs embedded to Editable Poly.

Now we can export the .obj back out.

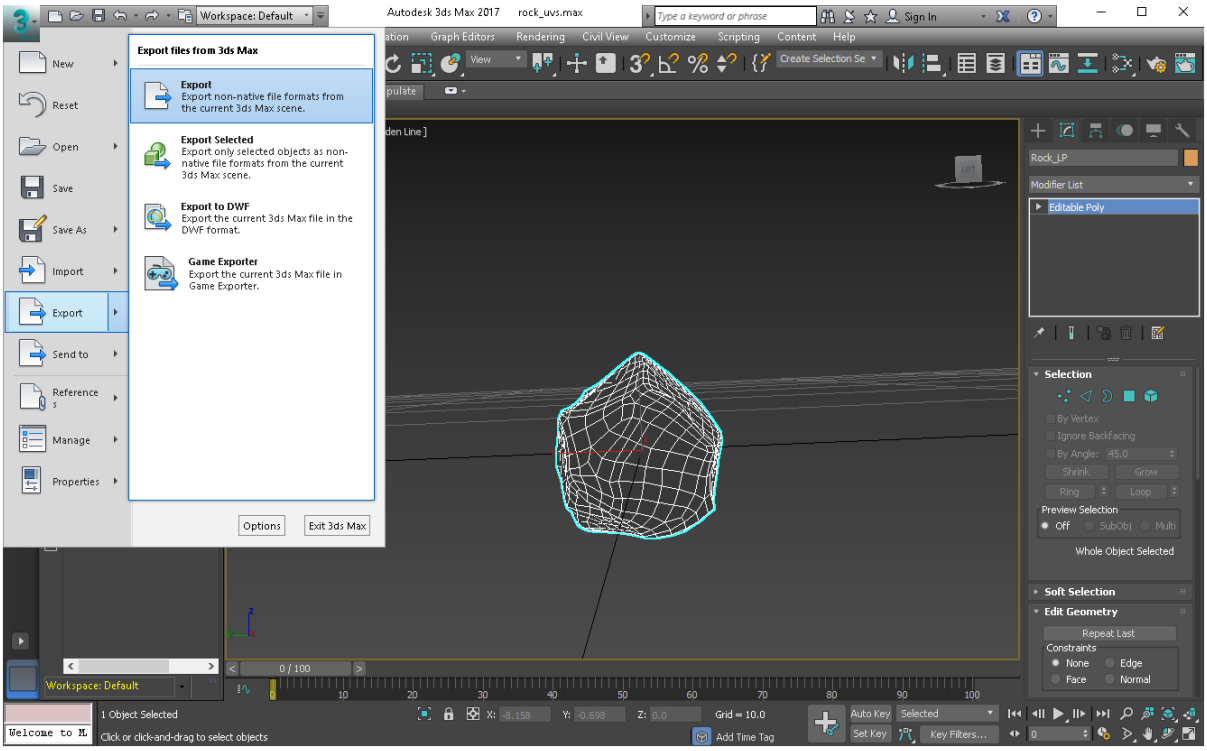


Figure 1136 - Export.

Select the folder to export to, enter in a file name and click Save.

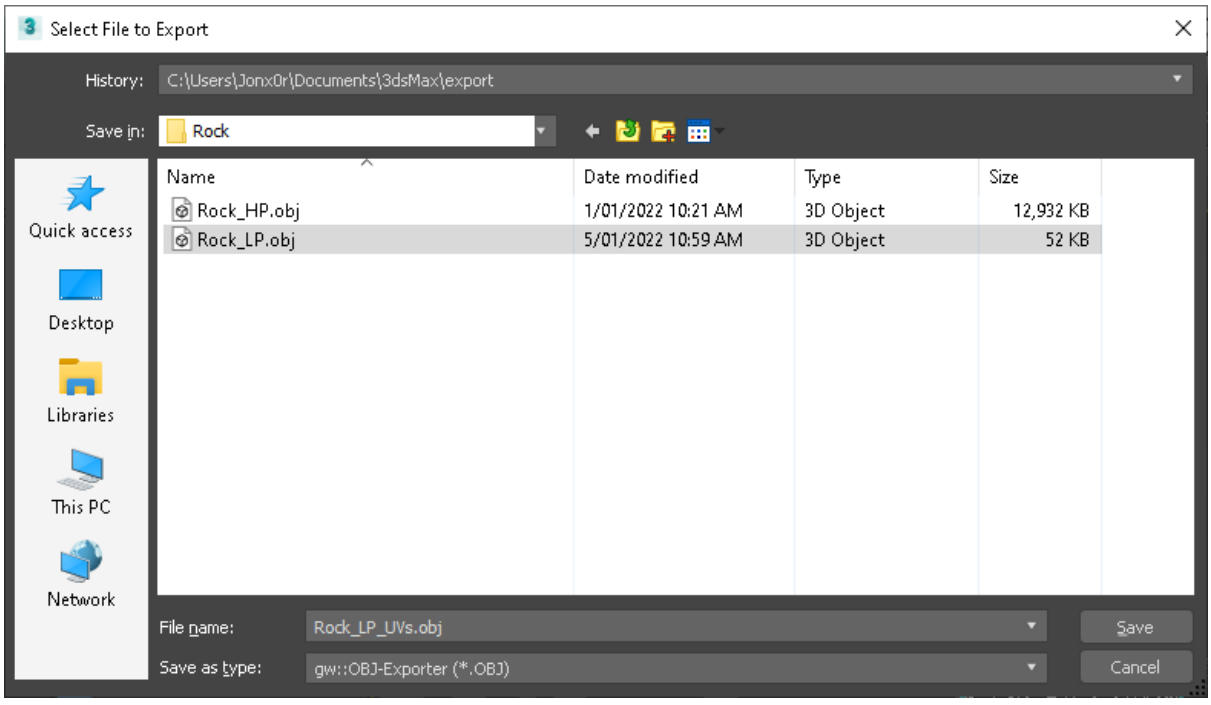


Figure 1137 - .obj export directory and file name.

Under Geometry, ensure 'Flip YZ-axis (Poser-like)', 'Texture coordinates', 'Normals' and 'Smoothing groups' and ticked then click Export.



Figure 1138 - OBJ Export Options.

Next, let's head into Substance Painter to try and texture the boulder.

Create a new project by going to File > New.

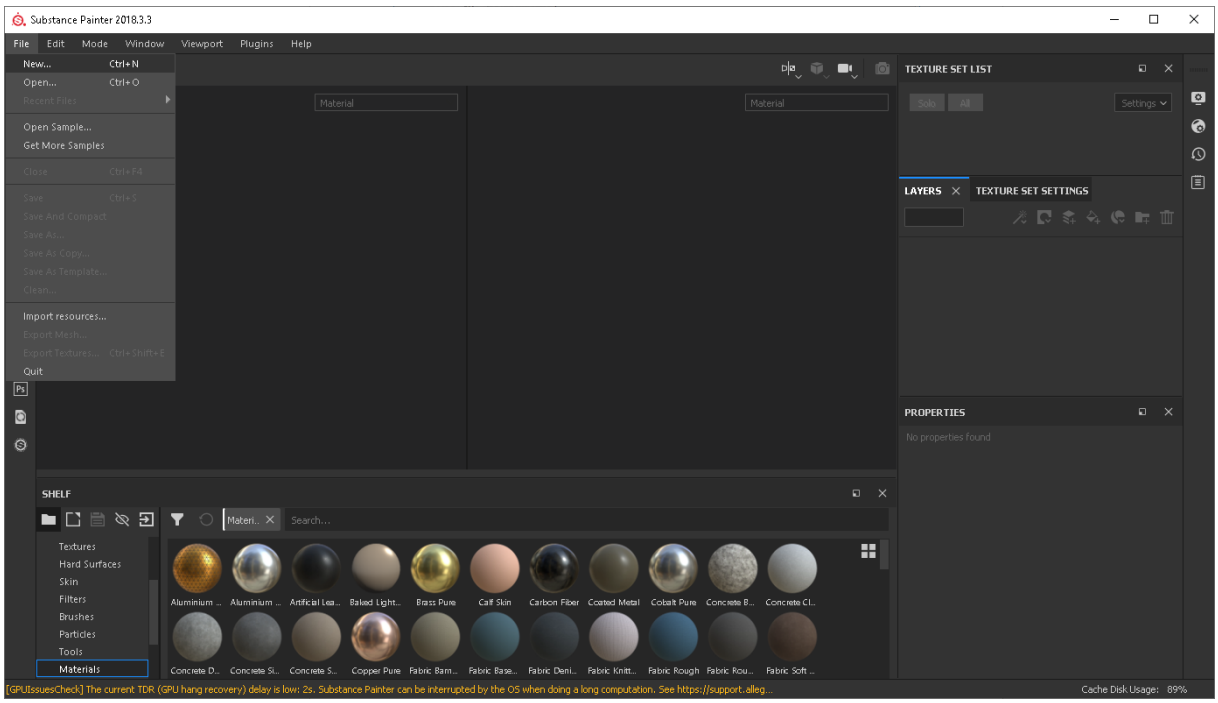


Figure 1139 - Substance Painter.

Set the Template to 'Non-PBR - Specular Glossiness (allegorithmic)'.

Click on the Select button next to File.

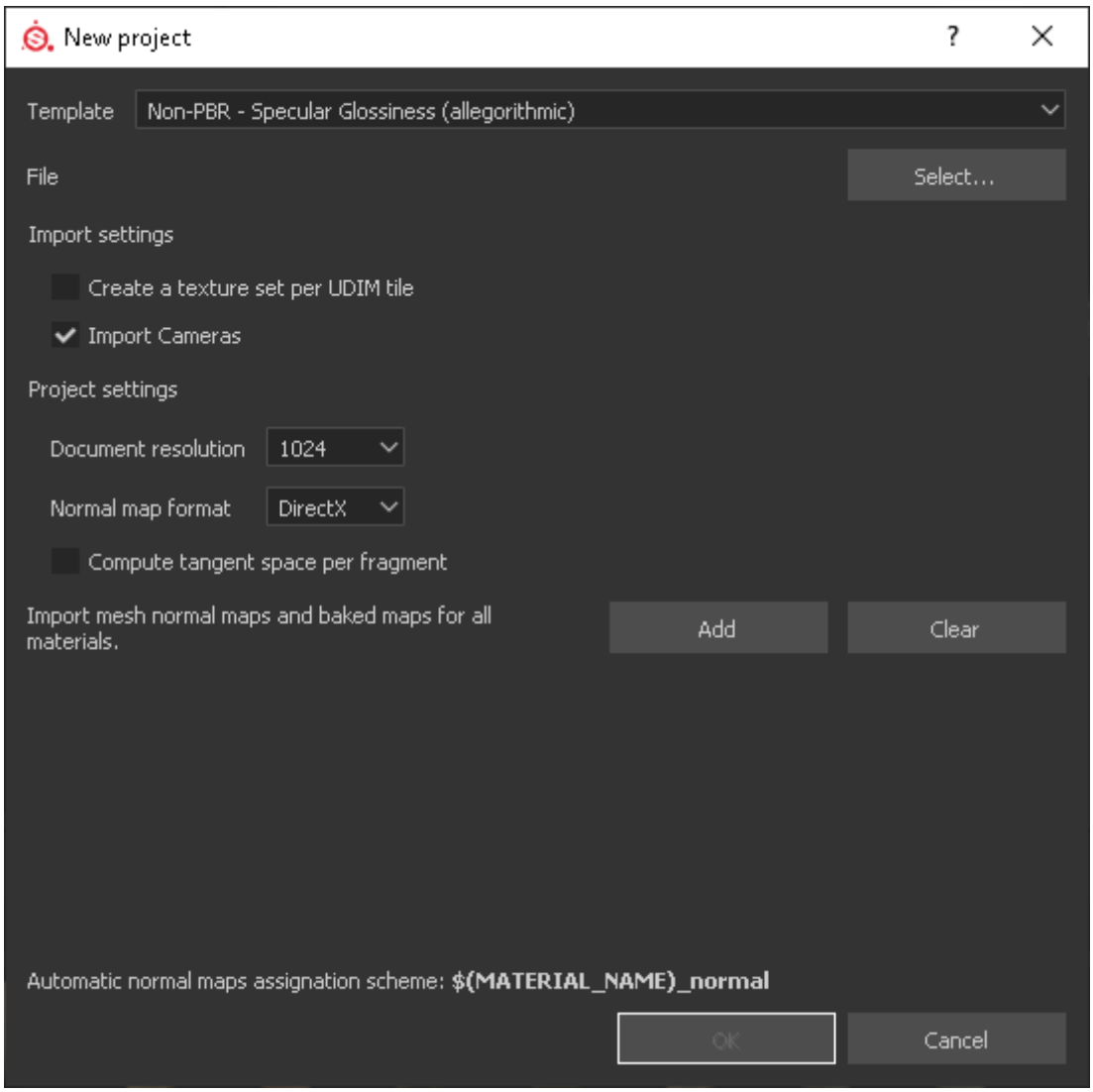


Figure 1140 - Selecting the file to load in Substance Painter.

Select the low poly version of your asset and click Open. Make sure you select the version with the fixed UVs.

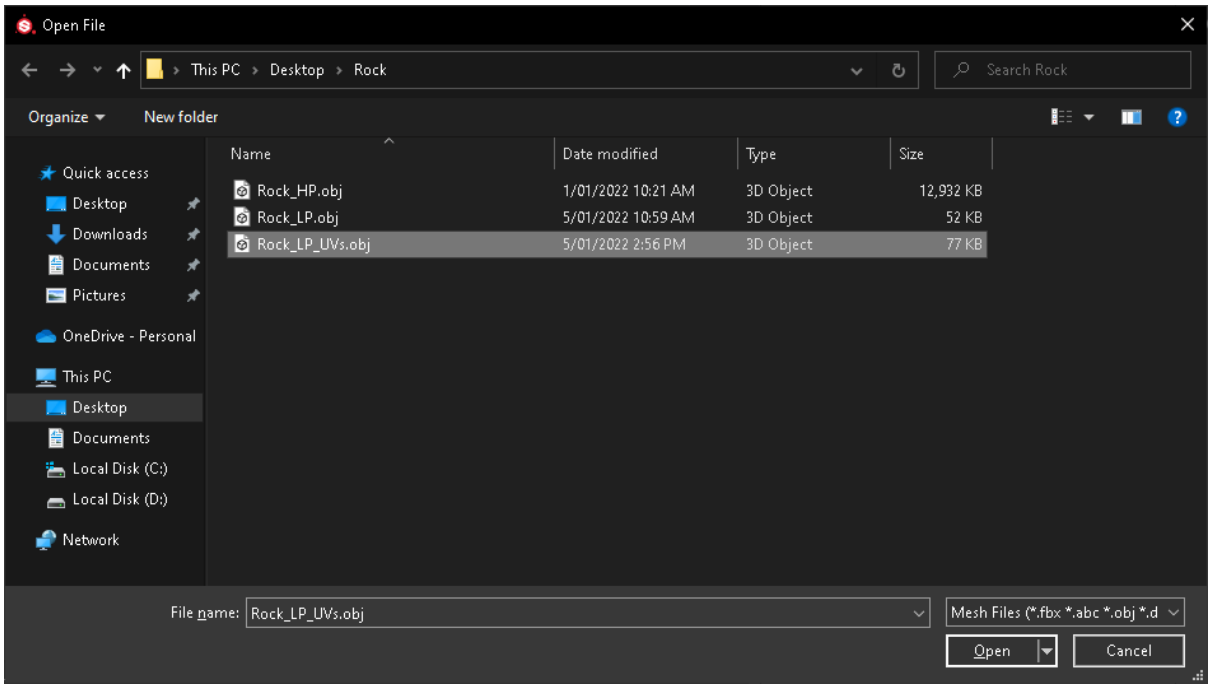


Figure 1141 - Selecting the low poly version of the asset.

The default resolution of Skyrim texture maps is 1024 so I'm going to leave the Document Resolution set to that.

Ensure the Normal map format is set to DirectX.

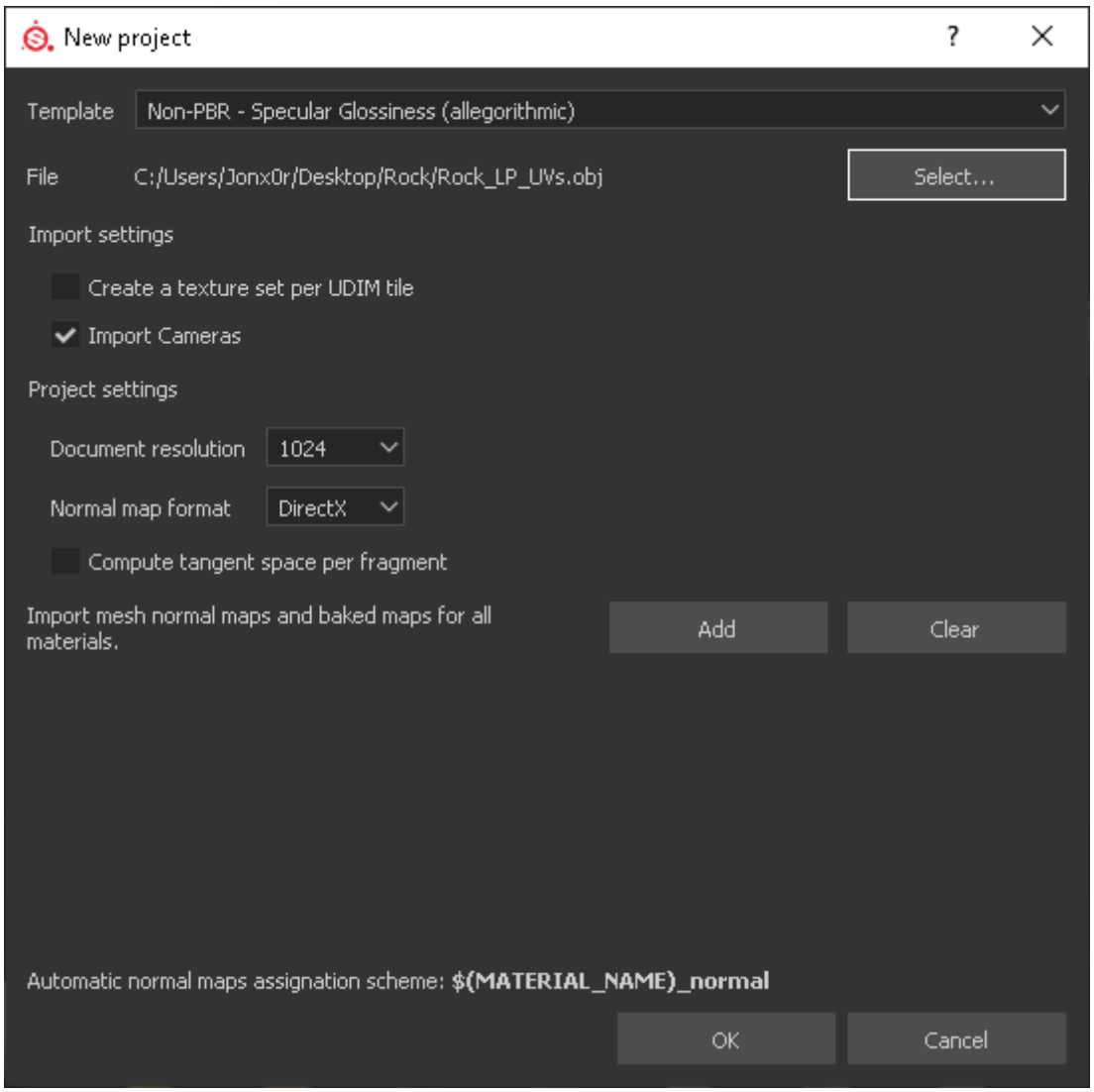


Figure 1142 - Project settings.

Click OK to load the asset into Substance Painter.

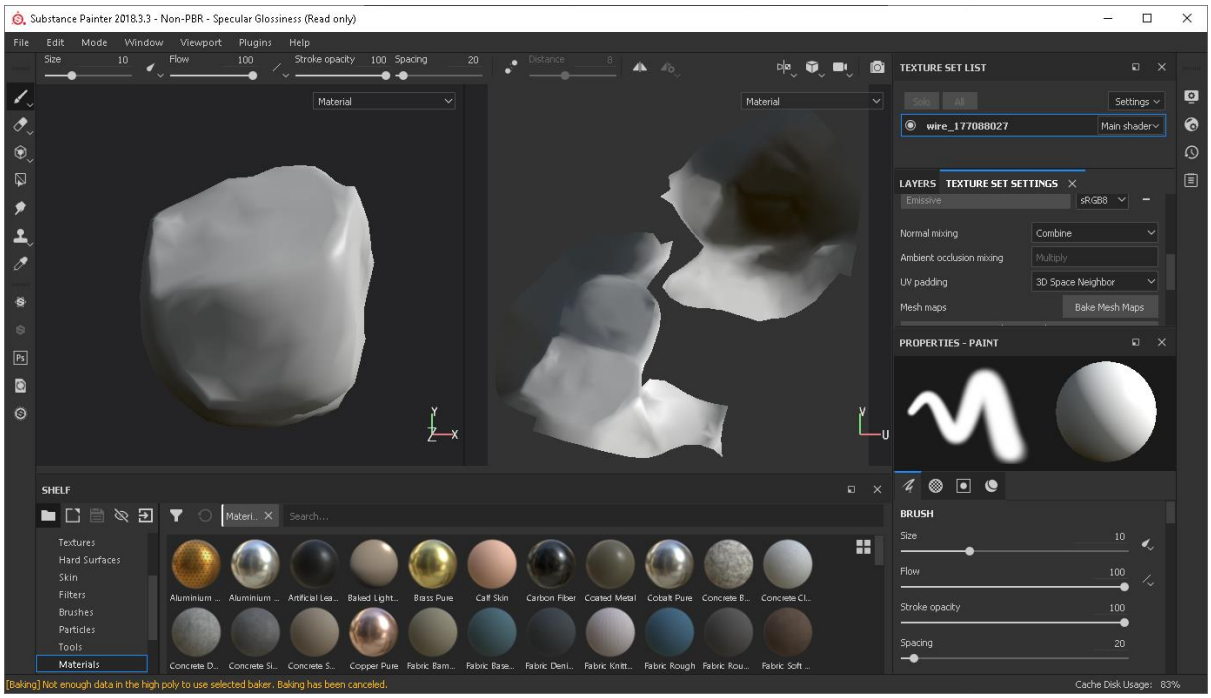


Figure 1143 - Rock loaded into Substance Painter.

The first thing we'll need to do in Substance Painter is bake in the details from our high poly asset.

Under the Texture Set Settings tab, click on the Bake Mesh Maps button.

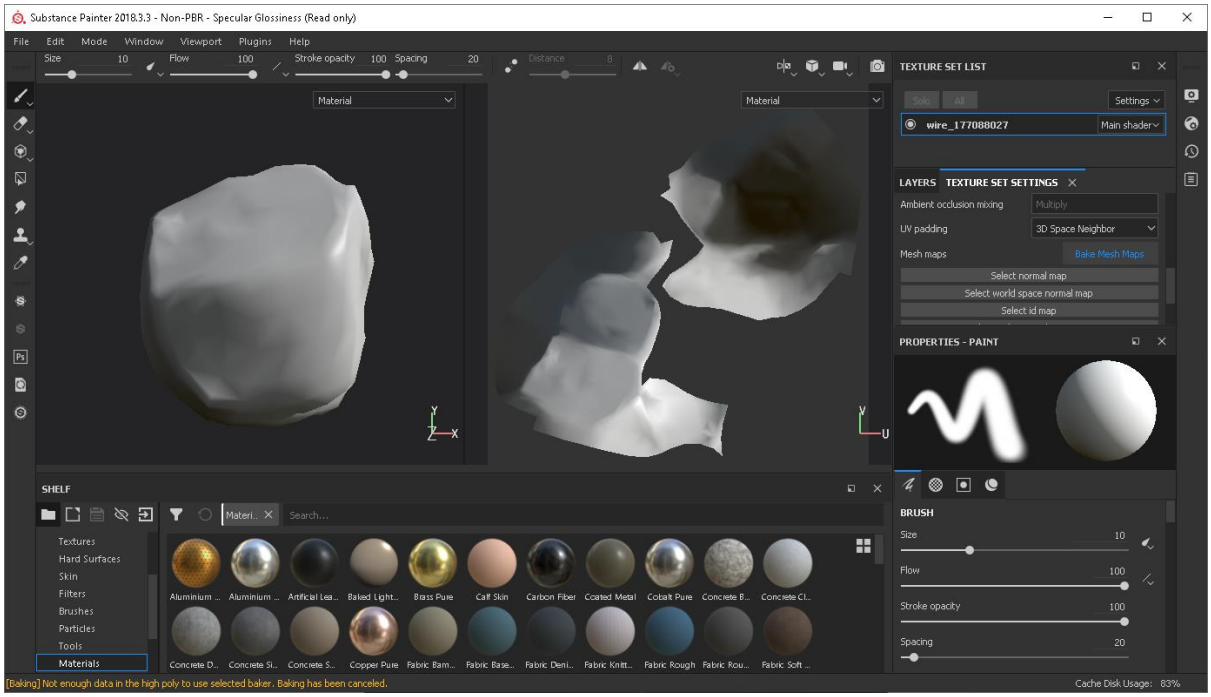


Figure 1144 - Bake Mesh Maps.

Click on the File icon next to High Definition Meshes to select the high poly version of the asset.

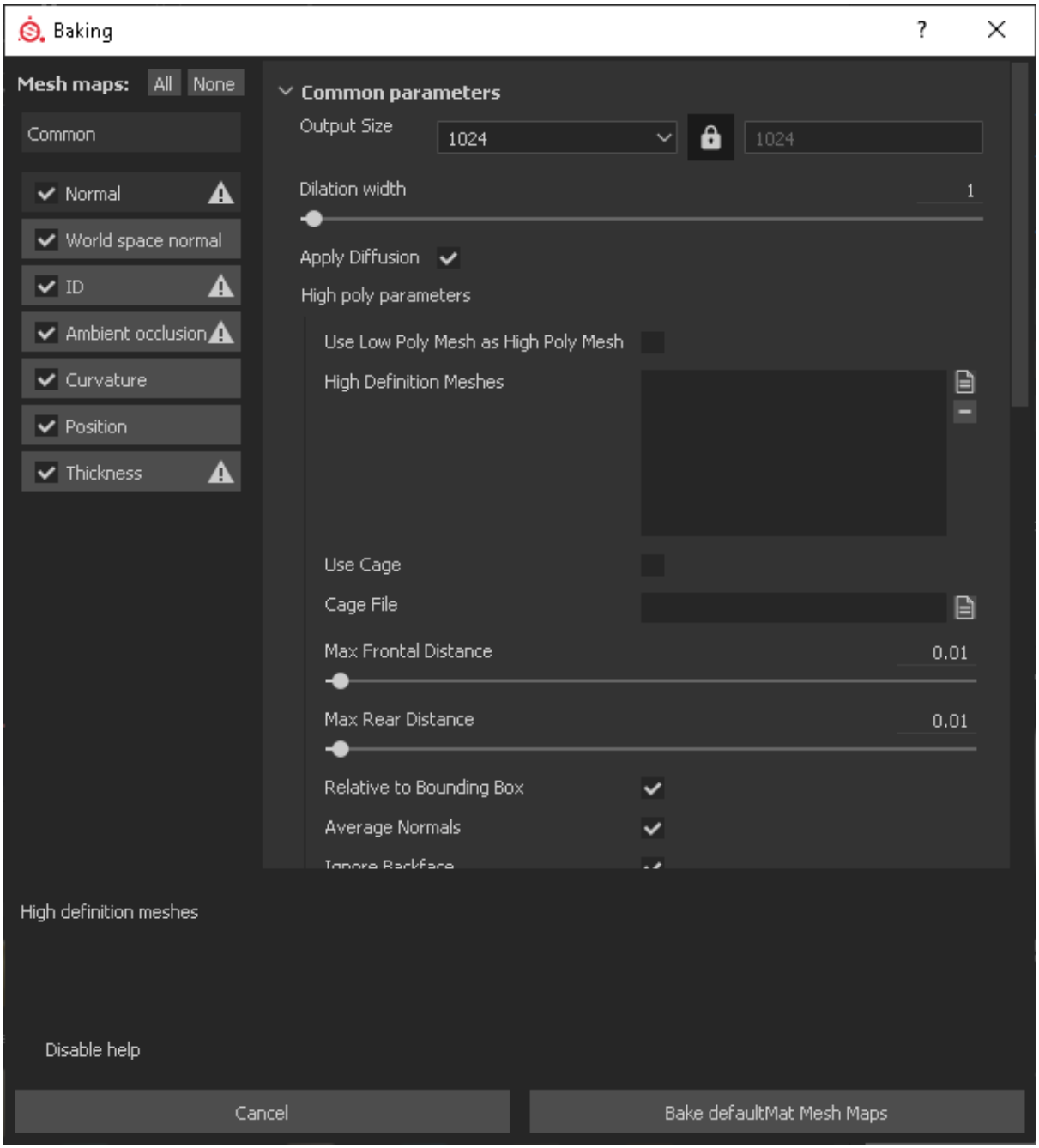


Figure 1145 - Selecting the High Definition Mesh.

Select the high poly version then click Open.

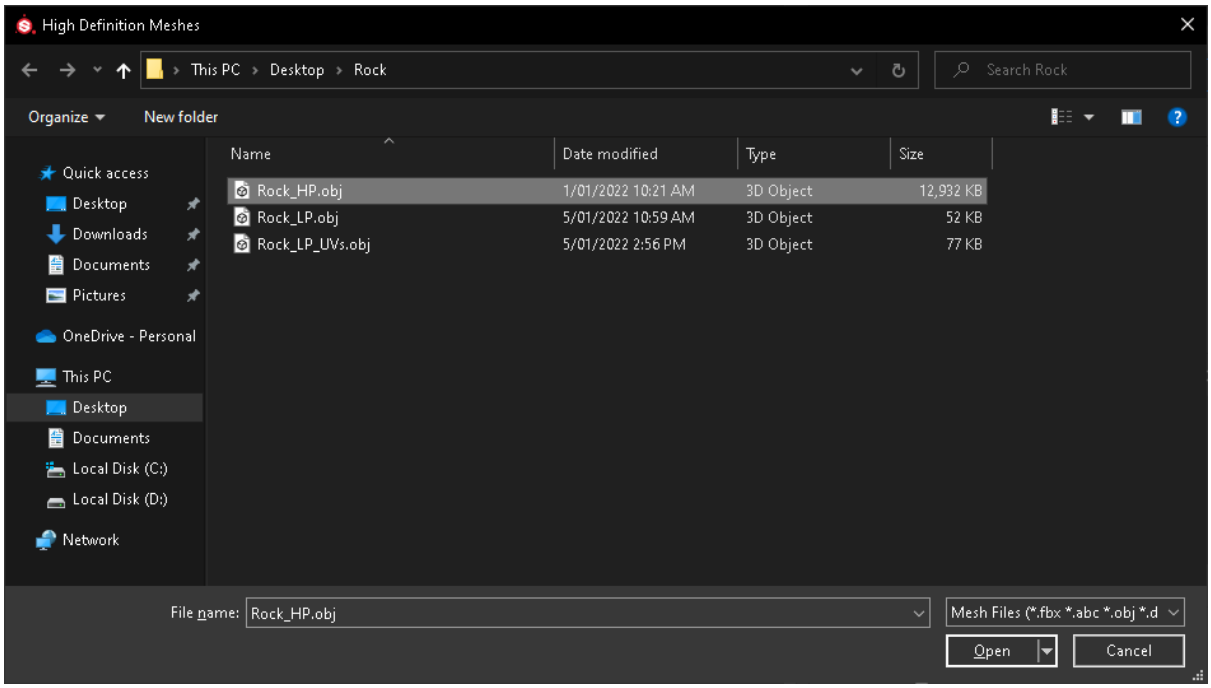


Figure 1146 - Selecting the high poly asset.

Some modellers also bake an ambient occlusion pass to add some fake shadows onto the actual texture. For Skyrim assets I would recommend using vertex colours instead.

Click on the Bake button to begin.

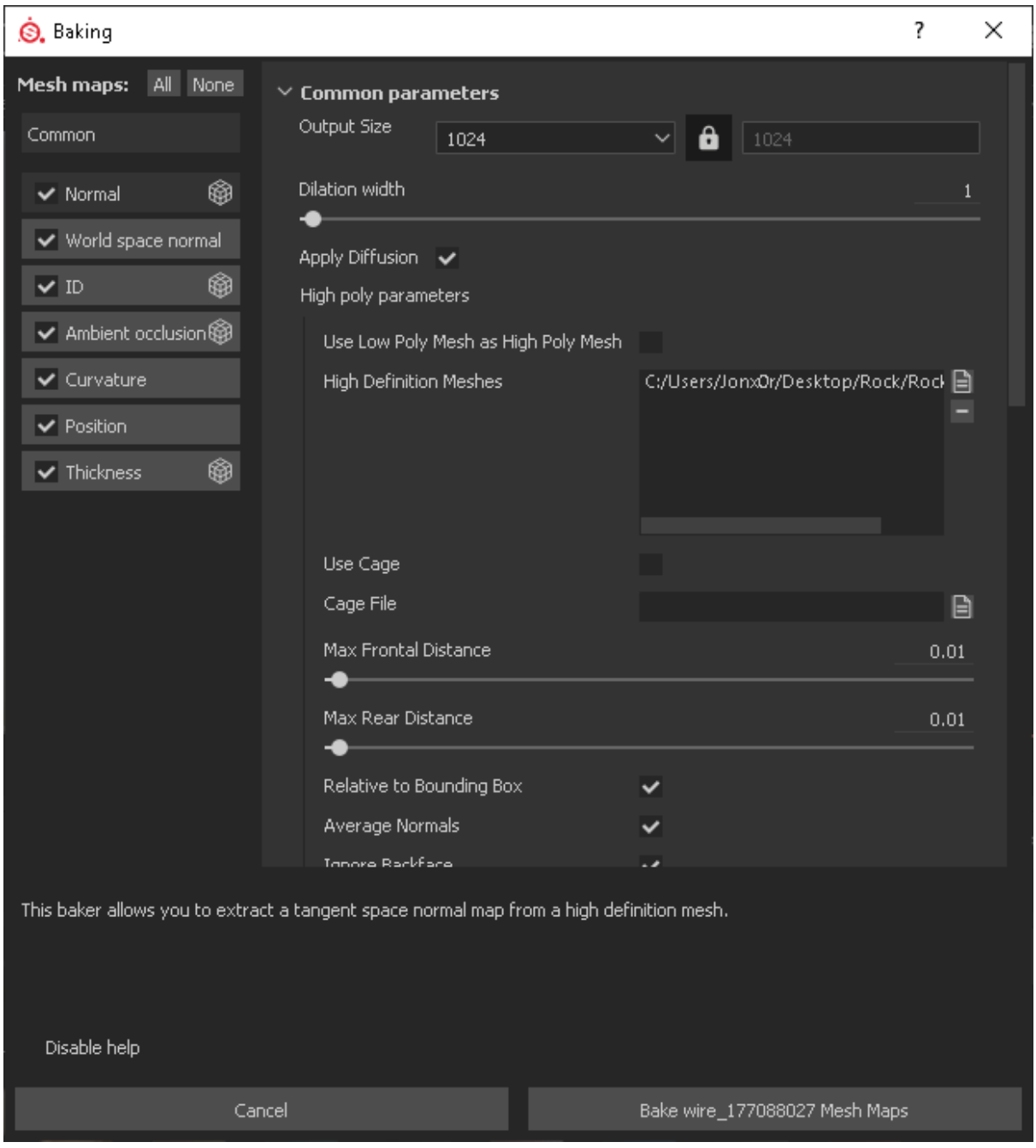


Figure 1147 - High definition mesh added.

You should now see details from the high poly version of the asset appear on the low poly mesh in the preview section.

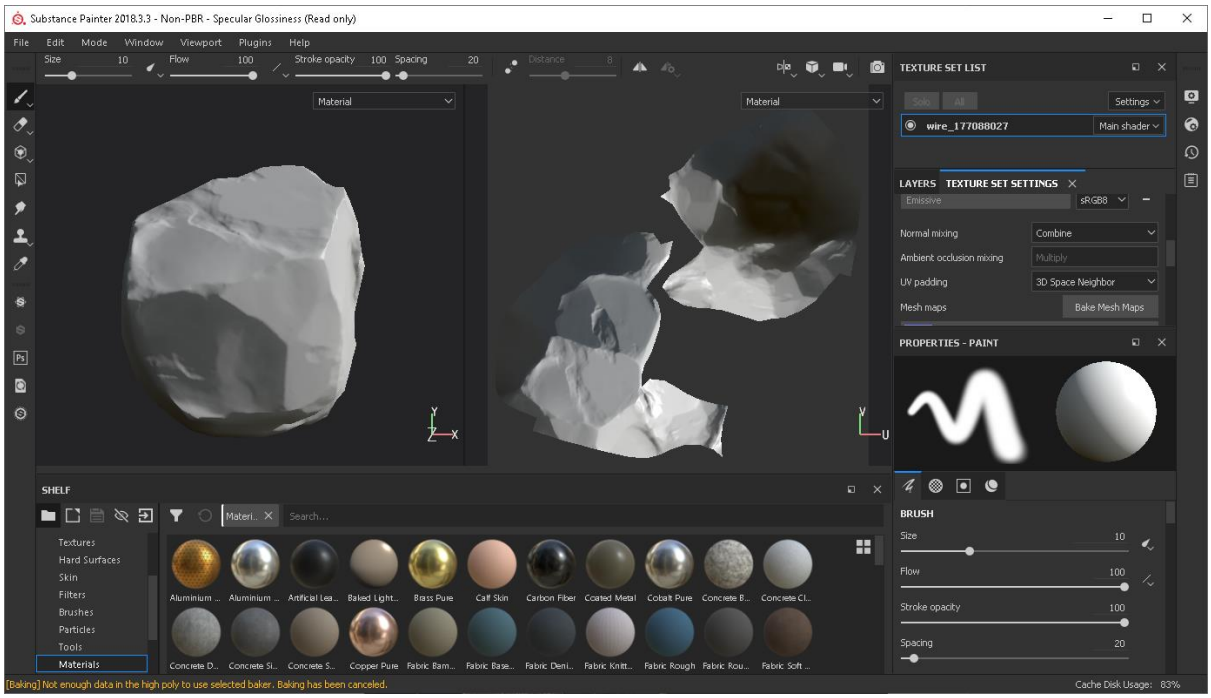


Figure 1148 - High poly detail baked onto low poly mesh.

For this example I just used Dirt, Concrete Bare with an Edges Scratched smart mask, and Mortar Wall with a few alterations to base colour to create the texture for the rock.

If you notice UV seams, try changing the Projection drop-down to Tri-Planar Projection.

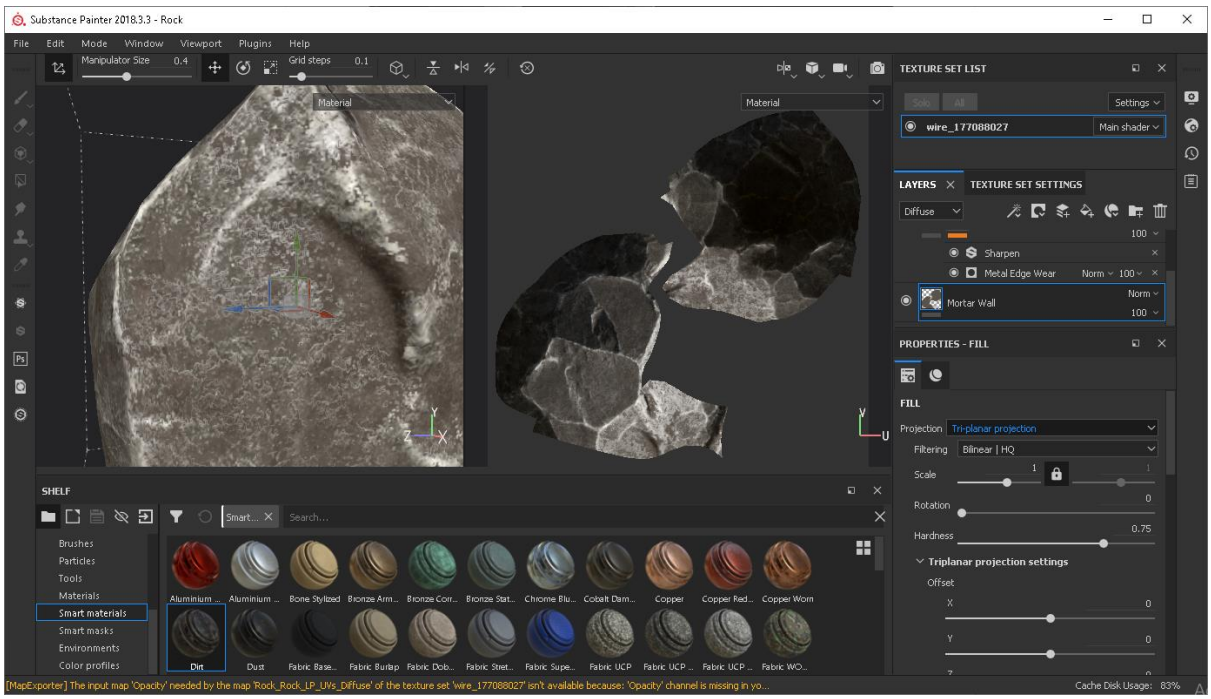


Figure 1149 - Tri-Planar Projection.

The last thing we need to do is export so we can bring the finished asset into 3ds Max.

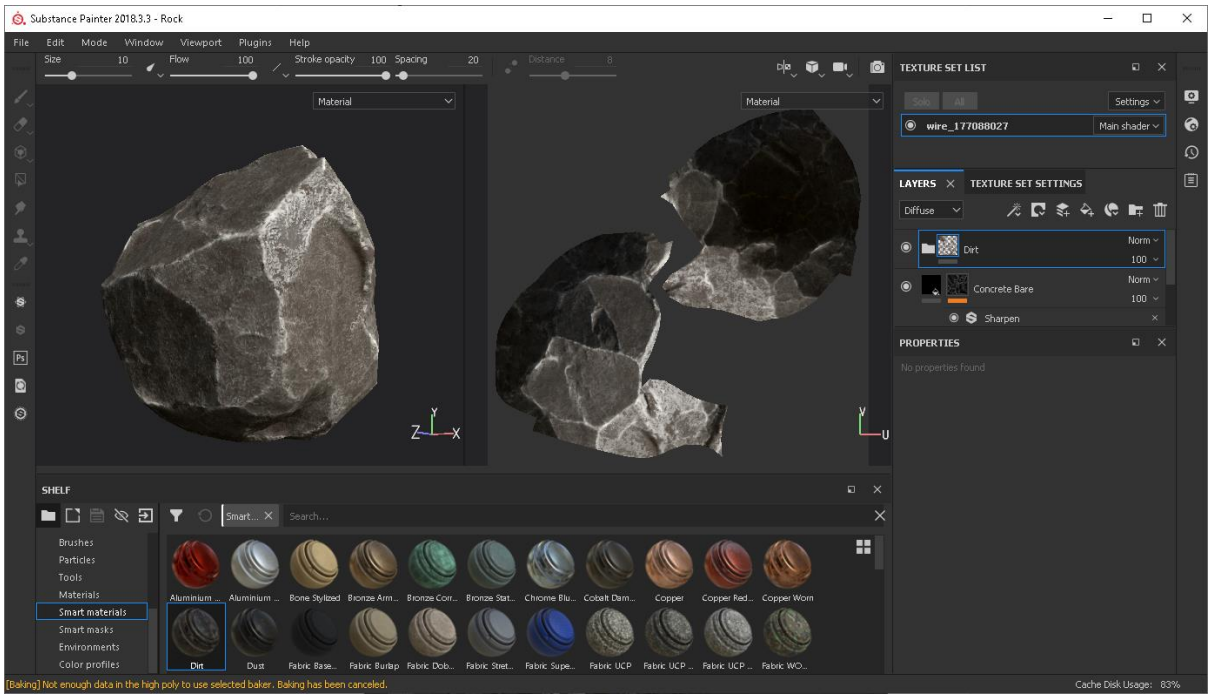


Figure 1150 - Sample rock textured.

Go to File > Export Textures.

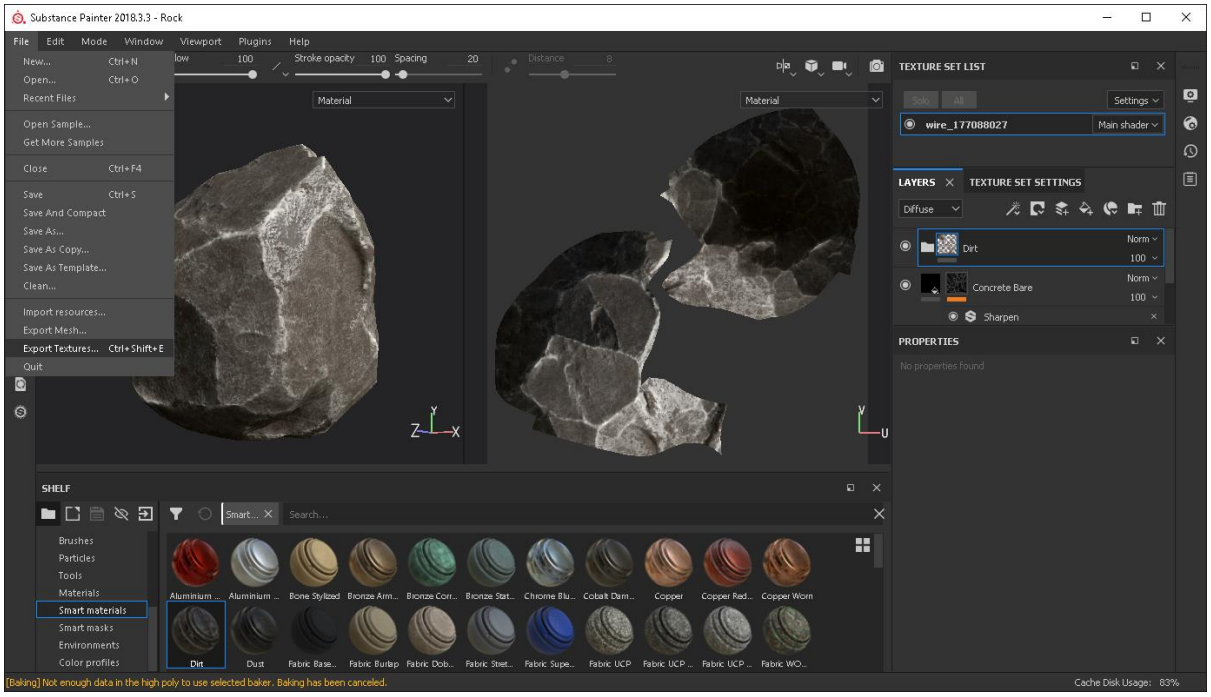


Figure 1151 - Export Textures.

We'll need to export the textures in the .png format then convert them to .dds in Photoshop later, so set the export format to 'png' and the bit depth to '8 bits'.

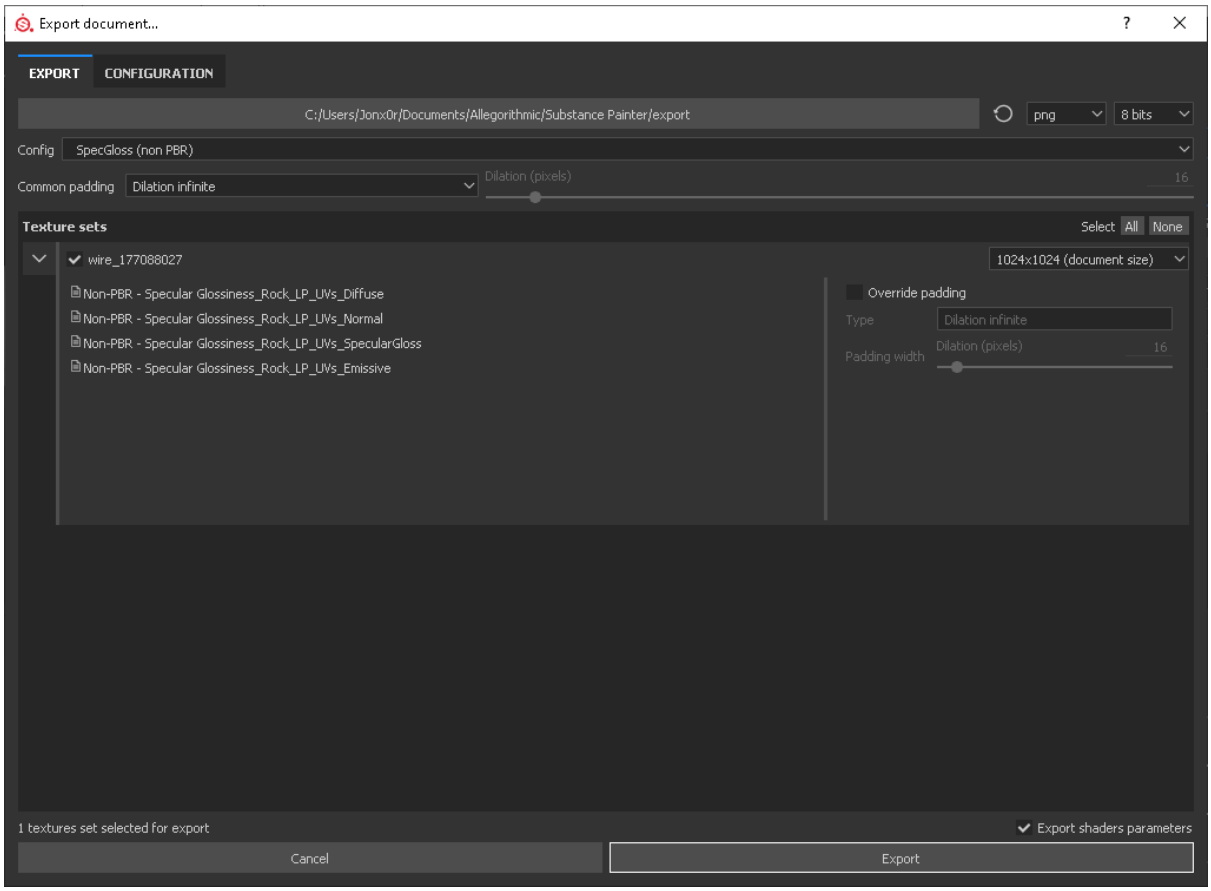


Figure 1152 - Export tab.

Set the export location by clicking on the folder path.

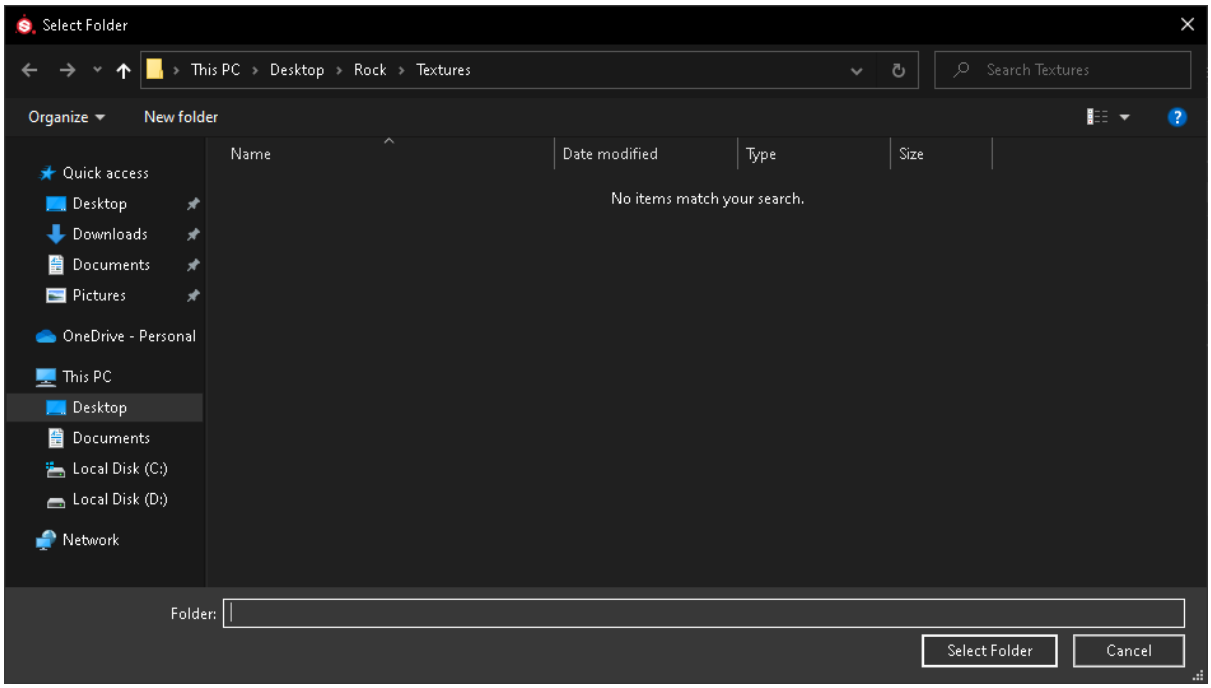


Figure 1153 - Setting the texture export path.

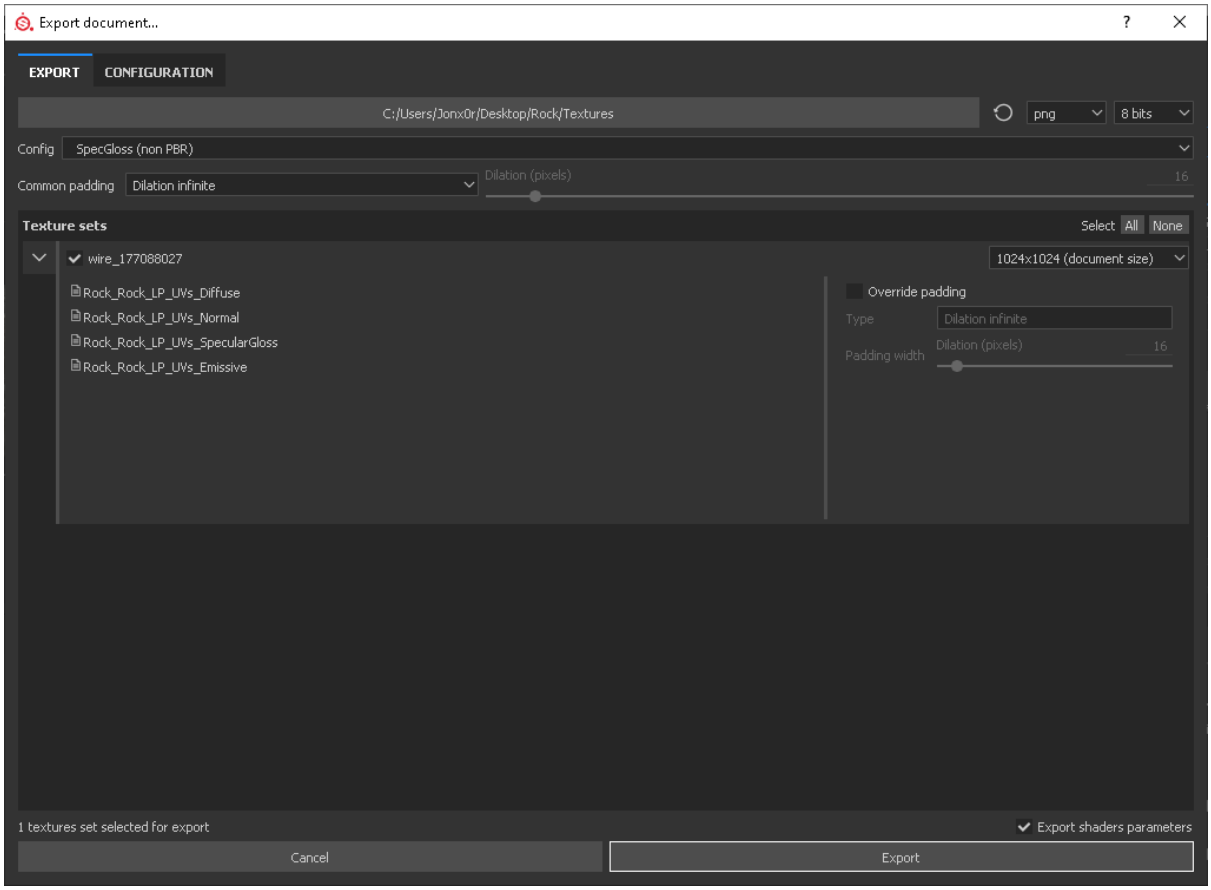


Figure 1154 - Texture export path set.

Go to the Configuration tab and click on the 'SpecGloss (non-PBR)' preset.

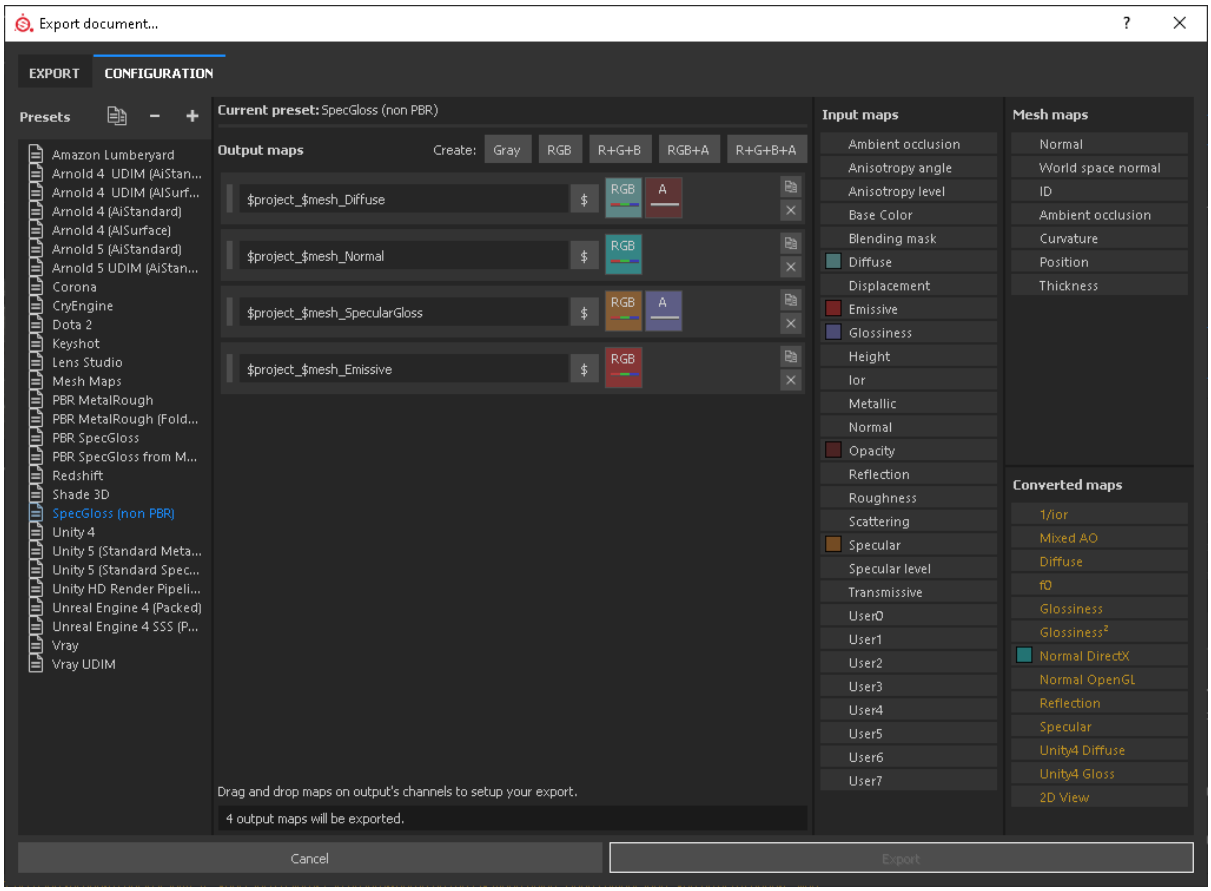


Figure 1155 - SpecGloss (non PBR) preset.

This will export a diffuse map with an alpha channel, a normal map, a specular map and an emissive map. All we really need are the diffuse and normal maps for this example.

Go back to the Export tab, untick 'Export shaders parameters'.

Click on the Export button to begin the process.

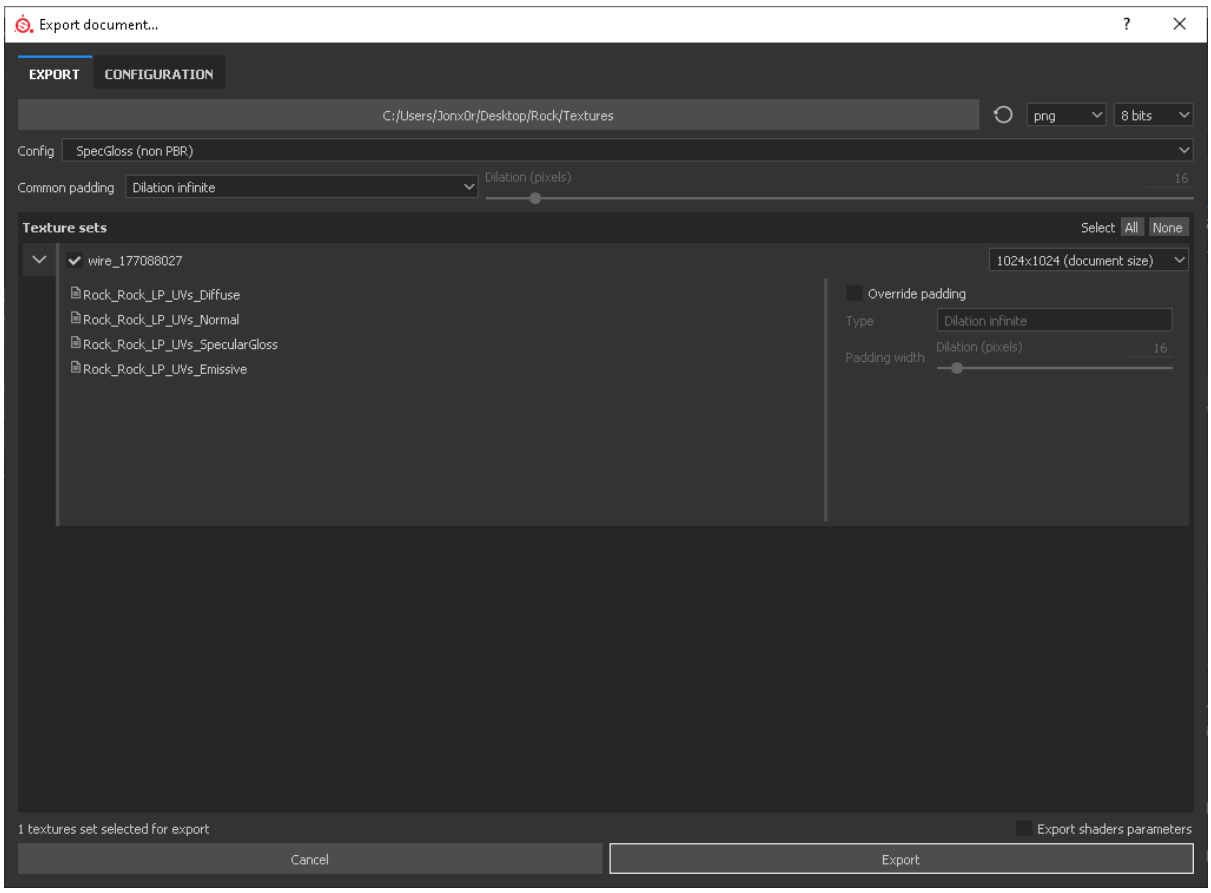


Figure 1156 - Exporting the texture maps.

Click OK to the pop-up once the process is complete.

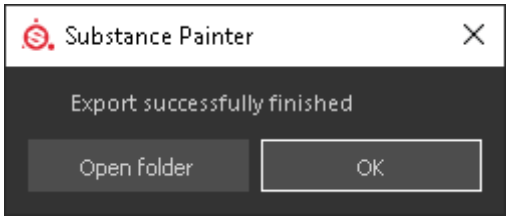


Figure 1157 - Export successfully finished.

Go to your export folder and confirm the texture maps have been successfully created.

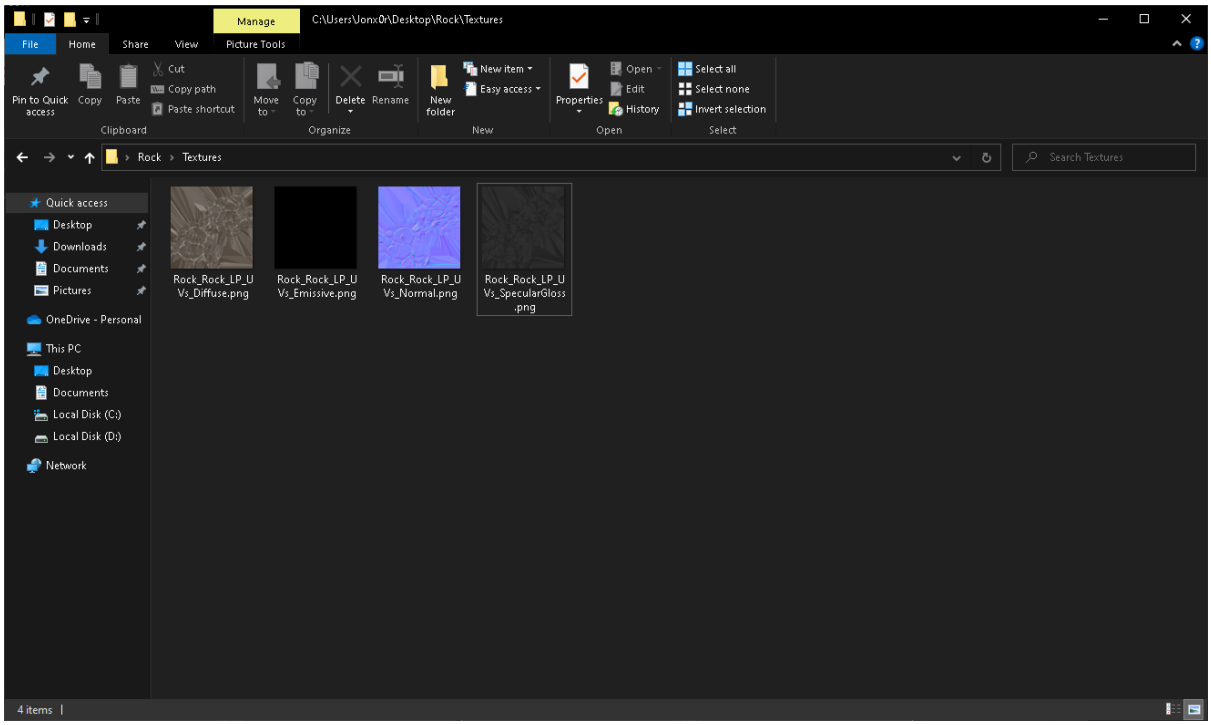


Figure 1158 - Texture maps created.

These maps can be converted to the .dds format in Photoshop using the NVIDIA Texture Tools plugin.

Open a texture map in Photoshop.

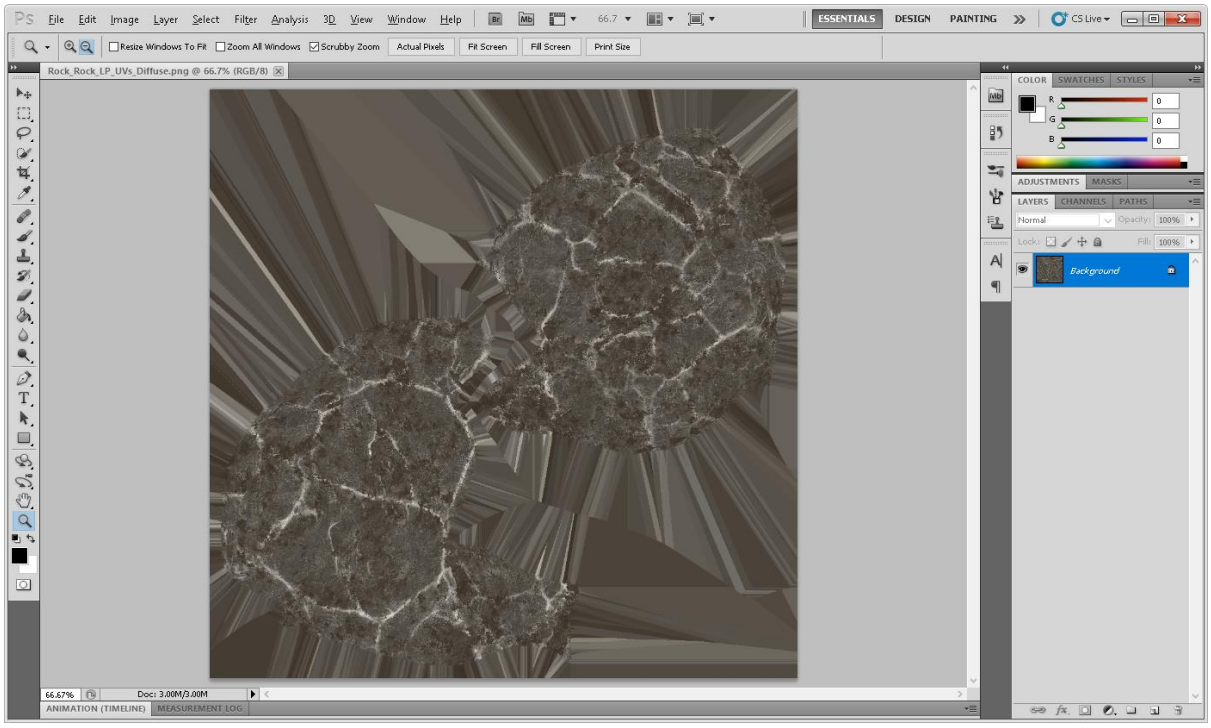


Figure 1159 - Diffuse map in Photoshop.

Go to File > Save As.

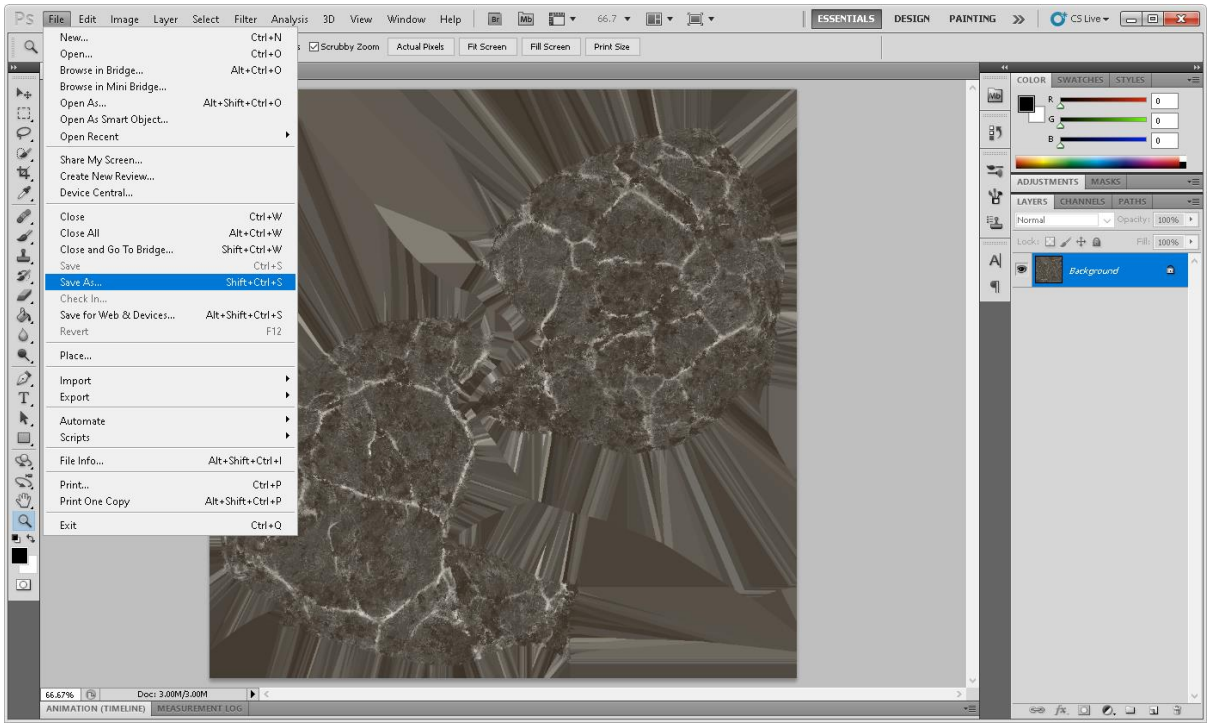


Figure 1160 - Saving a file in Photoshop.

Set the Format to .dds then click Save.

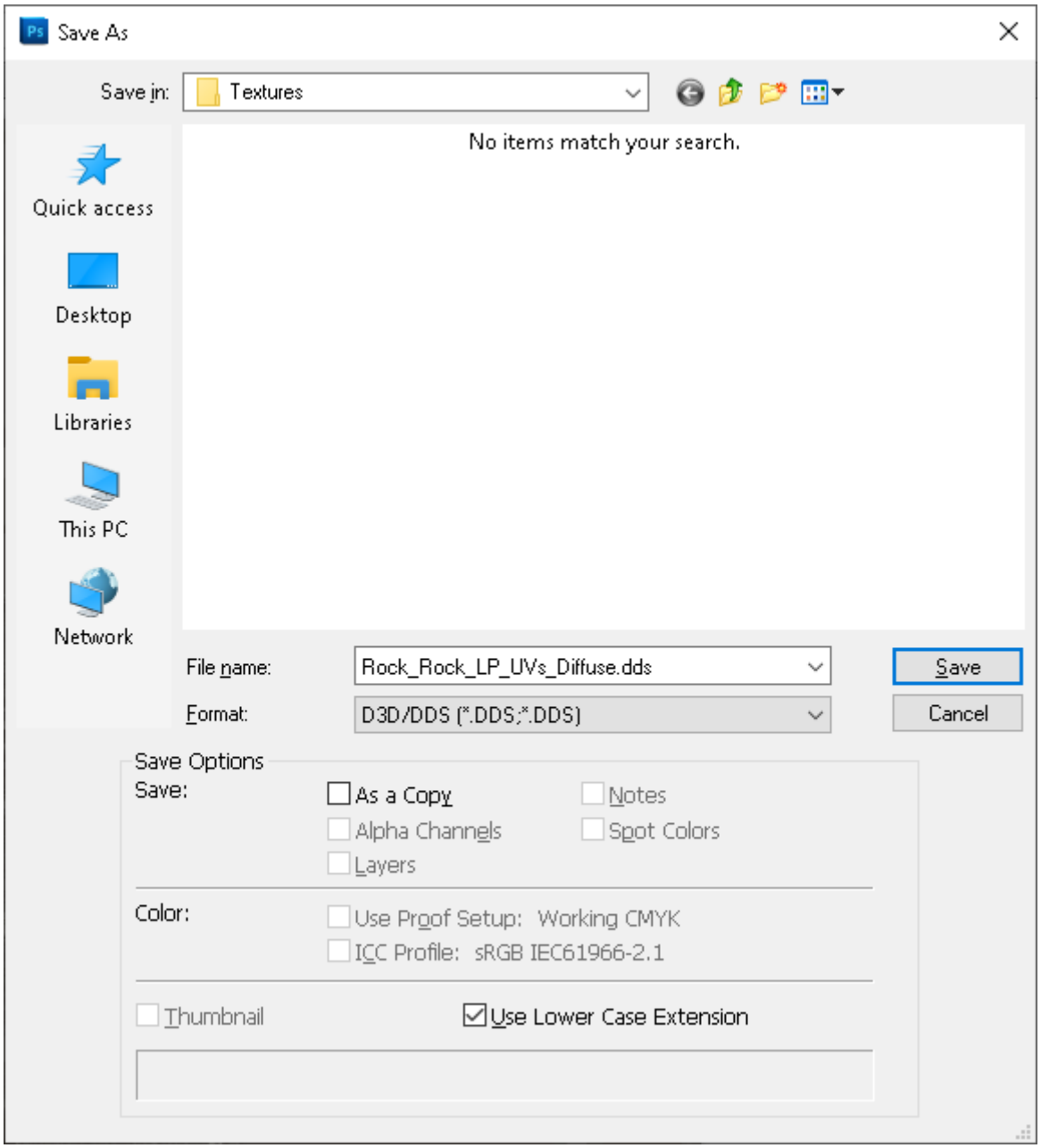


Figure 1161 - Saving a .dds file in Photoshop.

Set the format to DXT5, make sure 'Generate MIP maps' is selected then click Save.

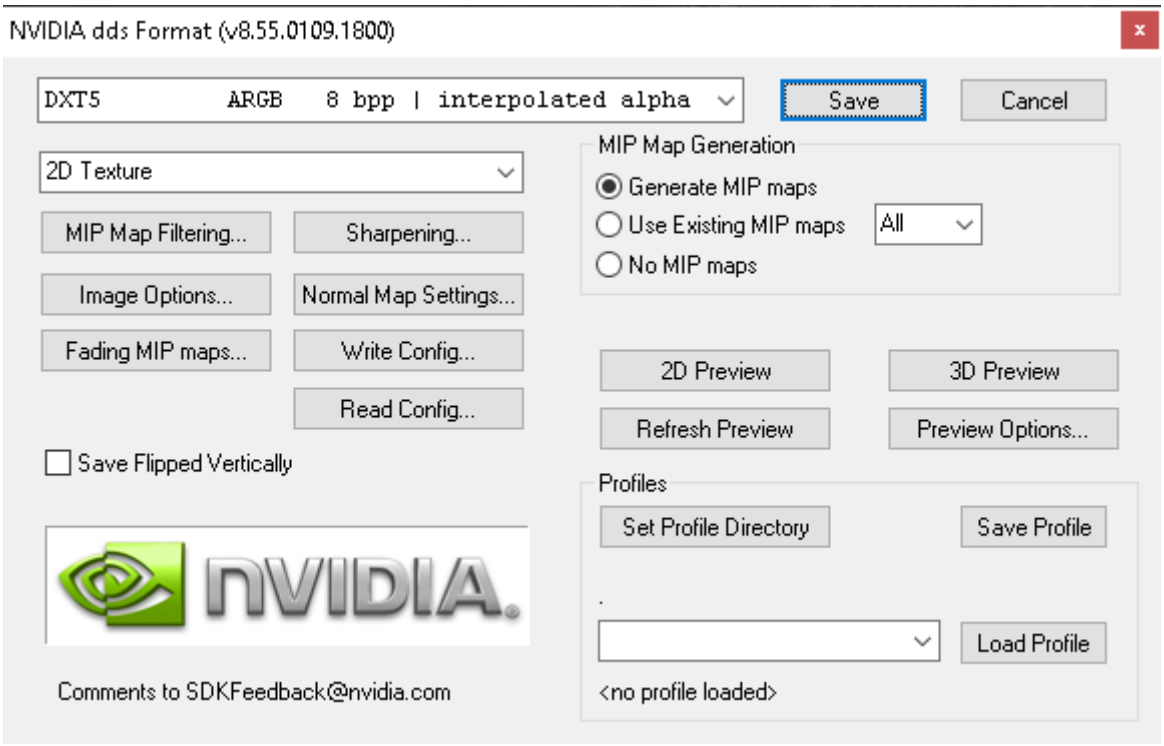


Figure 1162 - NVIDIA Texture Tools plugin for Photoshop.

Repeat these steps for the other texture maps.

Alright let's go back to 3ds Max.

To make setting up a new material a little bit easier and to help with scale, I'm going to import rockl01 from Skyrim. You can find it under meshes\landscape\rocks in Skyrim - Meshes.bsa. See the section titled Importing assets from Skyrim in 3ds Max for steps on how to do this.

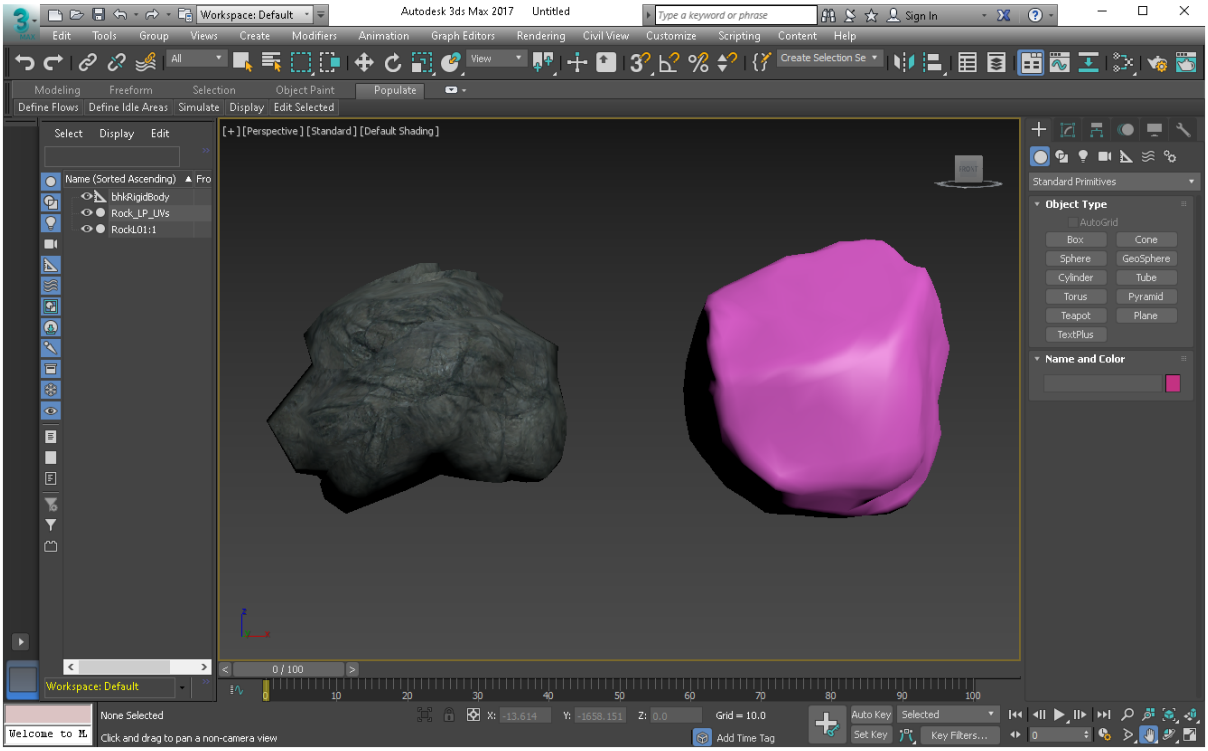


Figure 1163 - A Skyrim rock and our custom asset in 3ds Max.

Open the material editor. If you don't already have a view, right-click in the Views heading area and select Create New View.

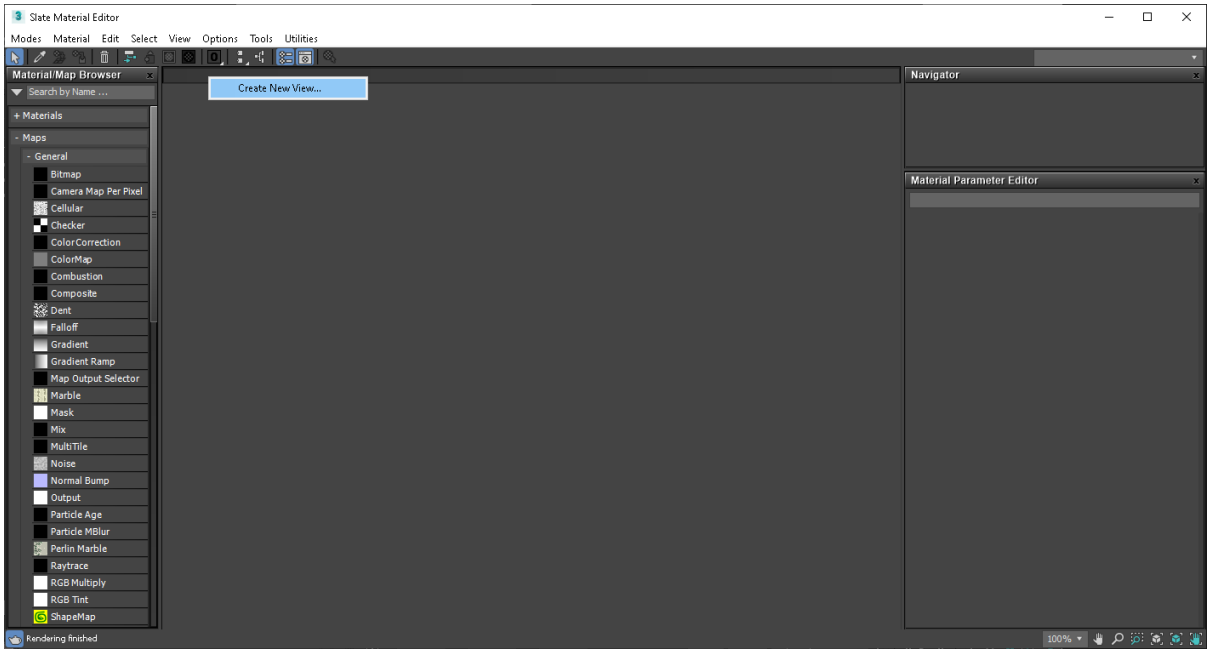


Figure 1164 - Material editor.

I'm just going to call this tab Skyrim Rock for this example.

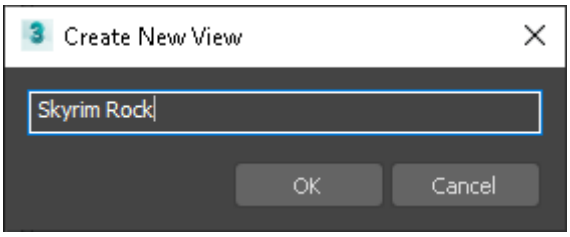


Figure 1165 - Naming the first tab.

Click on the eye-dropper in the material editor toolbar, then click on the Skyrim asset to get its material.

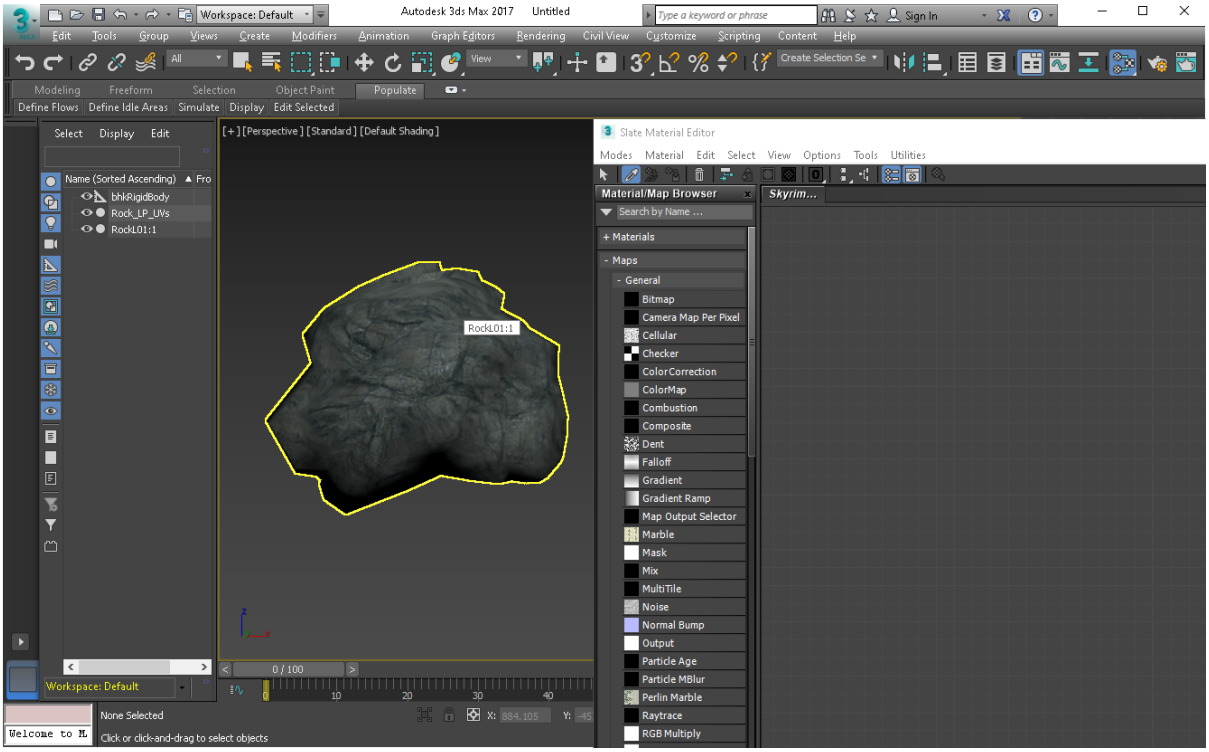


Figure 1166 - Loading the Skyrim rock material into the material editor.

The Skyrim asset's material setup should look like this:

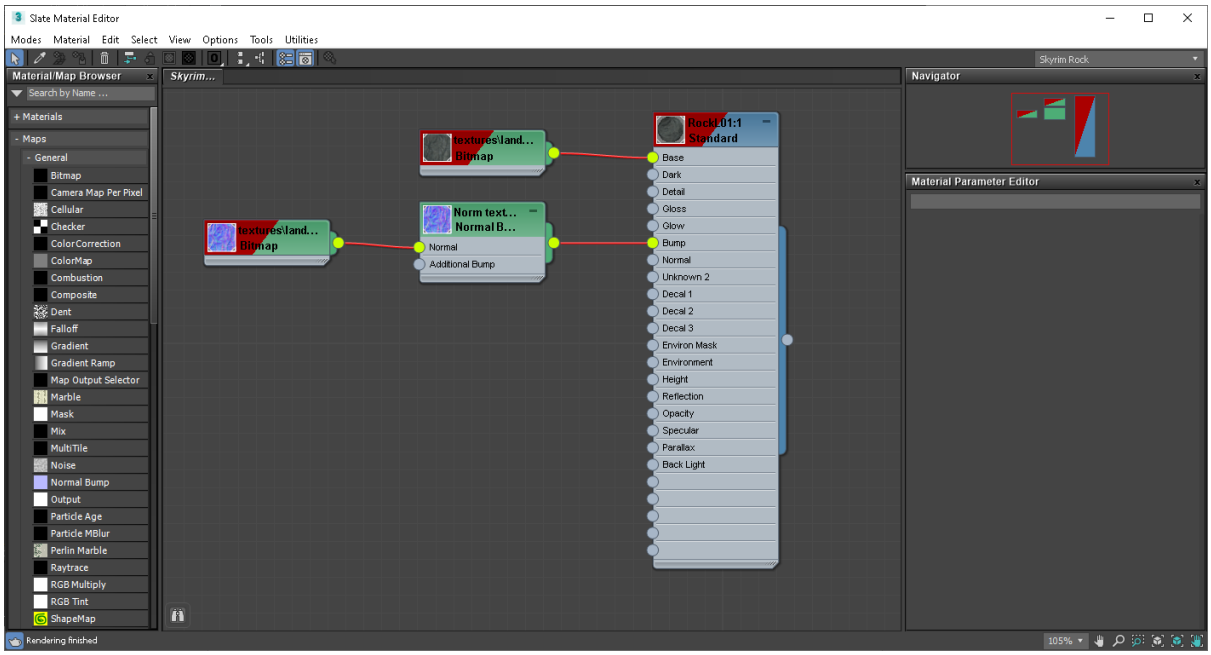


Figure 1167 - Skyrim rock material.

Double-click on the Bitmap node plugged into Base in the Standard node to open its properties.

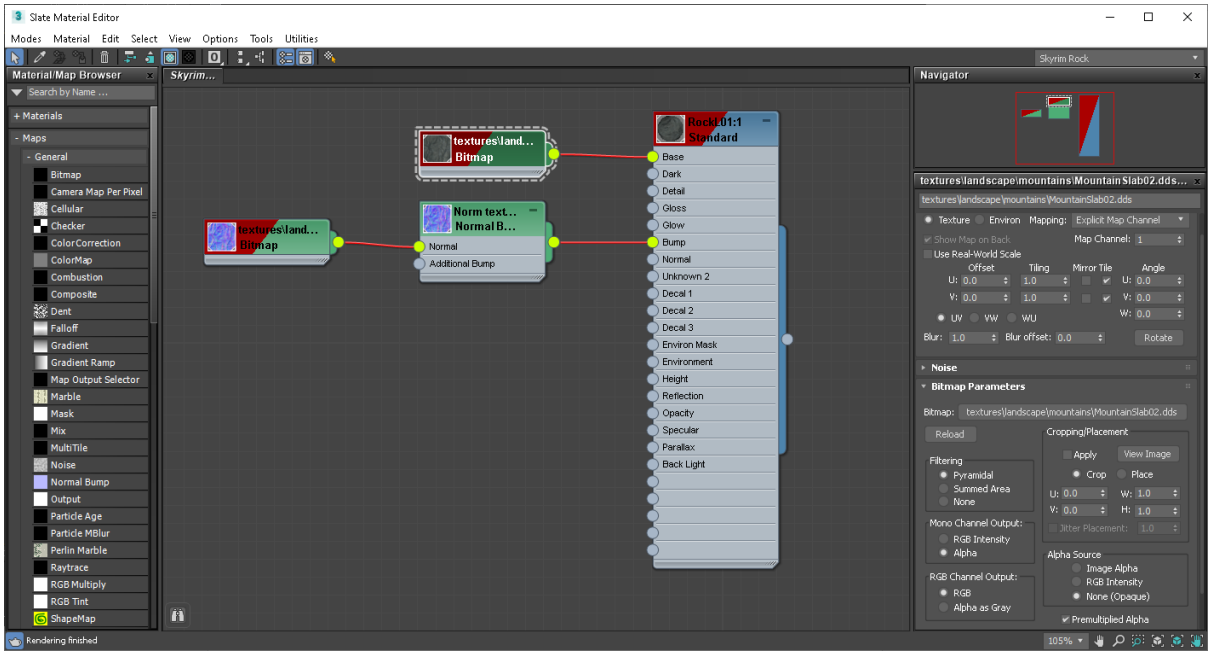


Figure 1168 - Diffuse map properties.

Go to the Bitmap Parameters section and click on the bitmap path. It should currently say ‘textures\landscape\mountains\MountainSlab02.dds’.

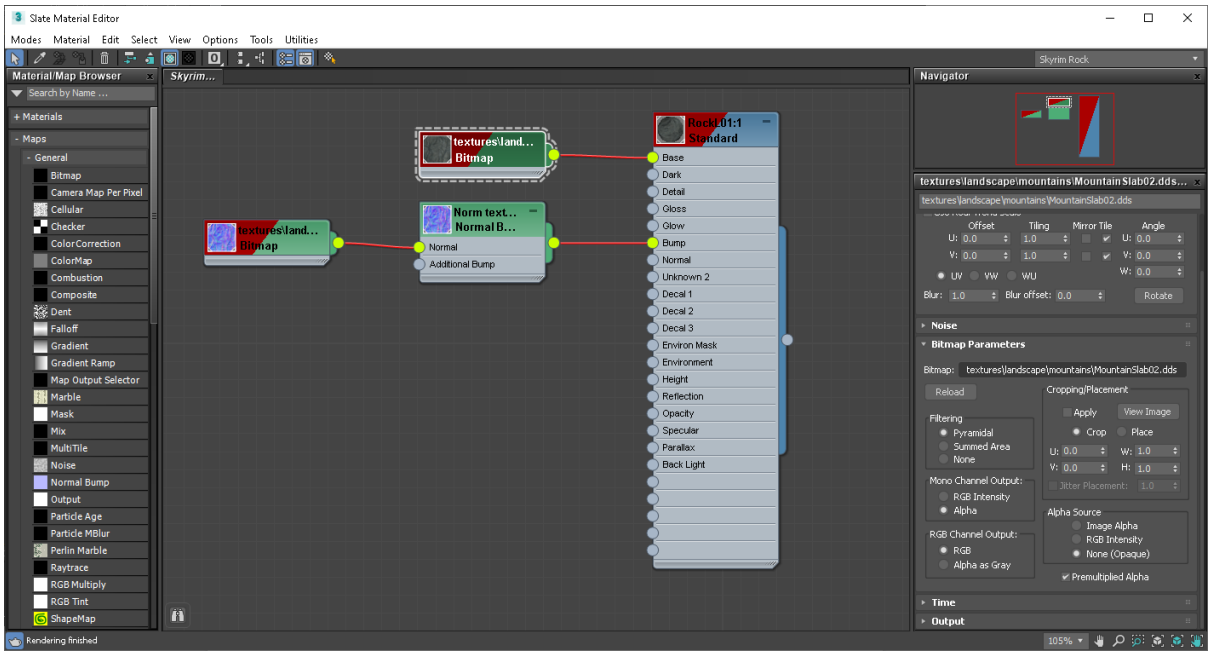


Figure 1169 - Bitmap Parameters.

Select the diffuse map exported from Substance Painter then click Open.

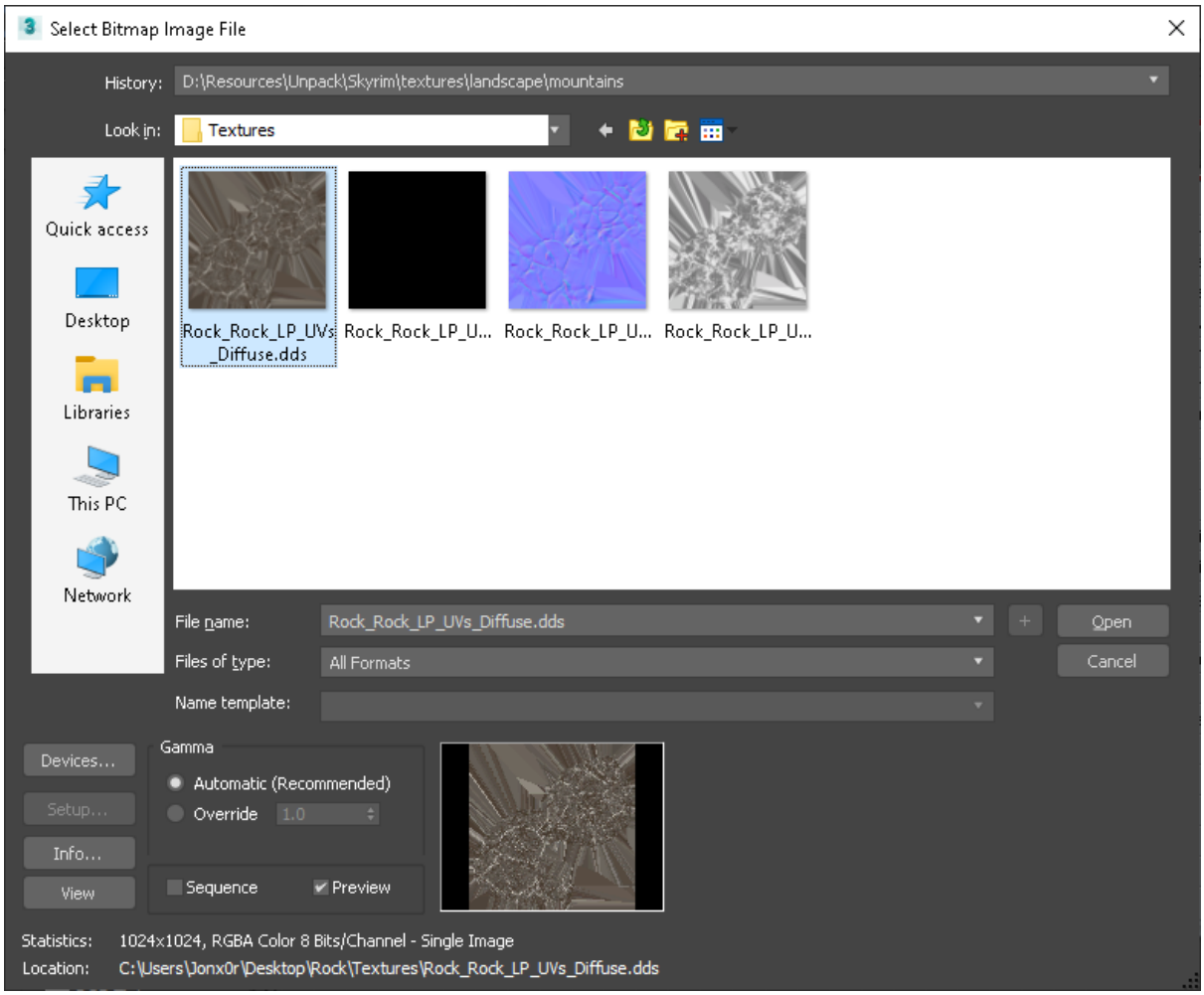


Figure 1170 - Changing the diffuse map.

Next, double-click on the Bitmap node used for the Normal/Bump map.

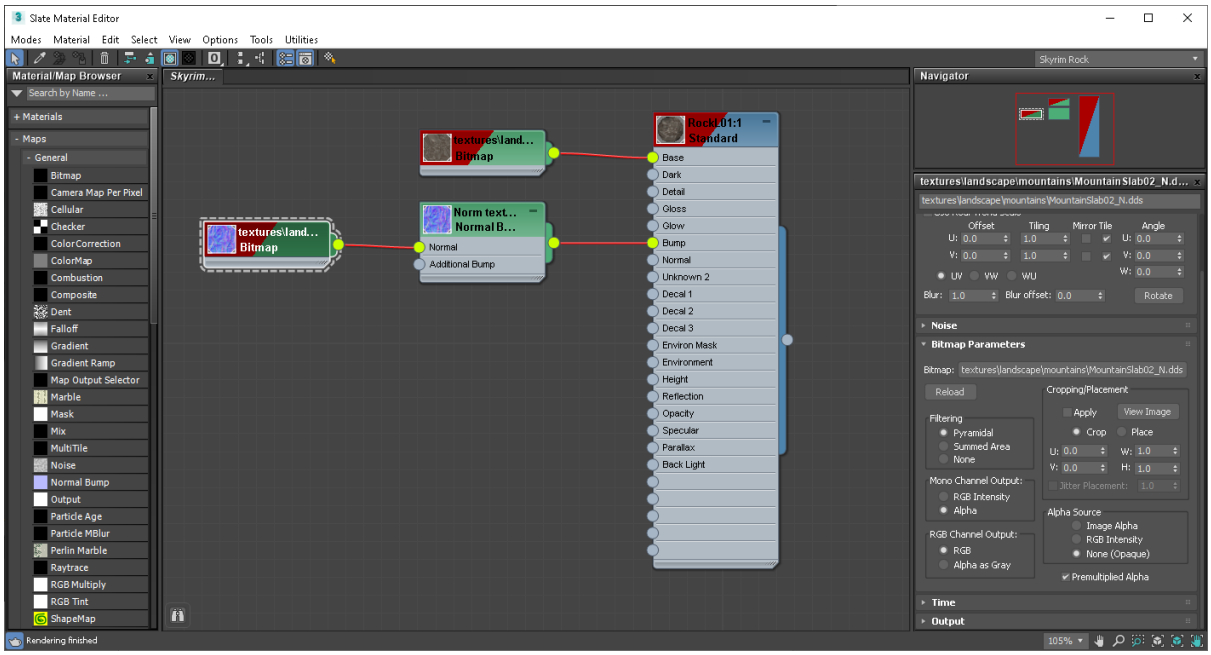


Figure 1171 - Changing the normal map.

Under Bitmap Parameters, click on the Bitmap path and change it to the normal map exported from Substance Painter then click Open.

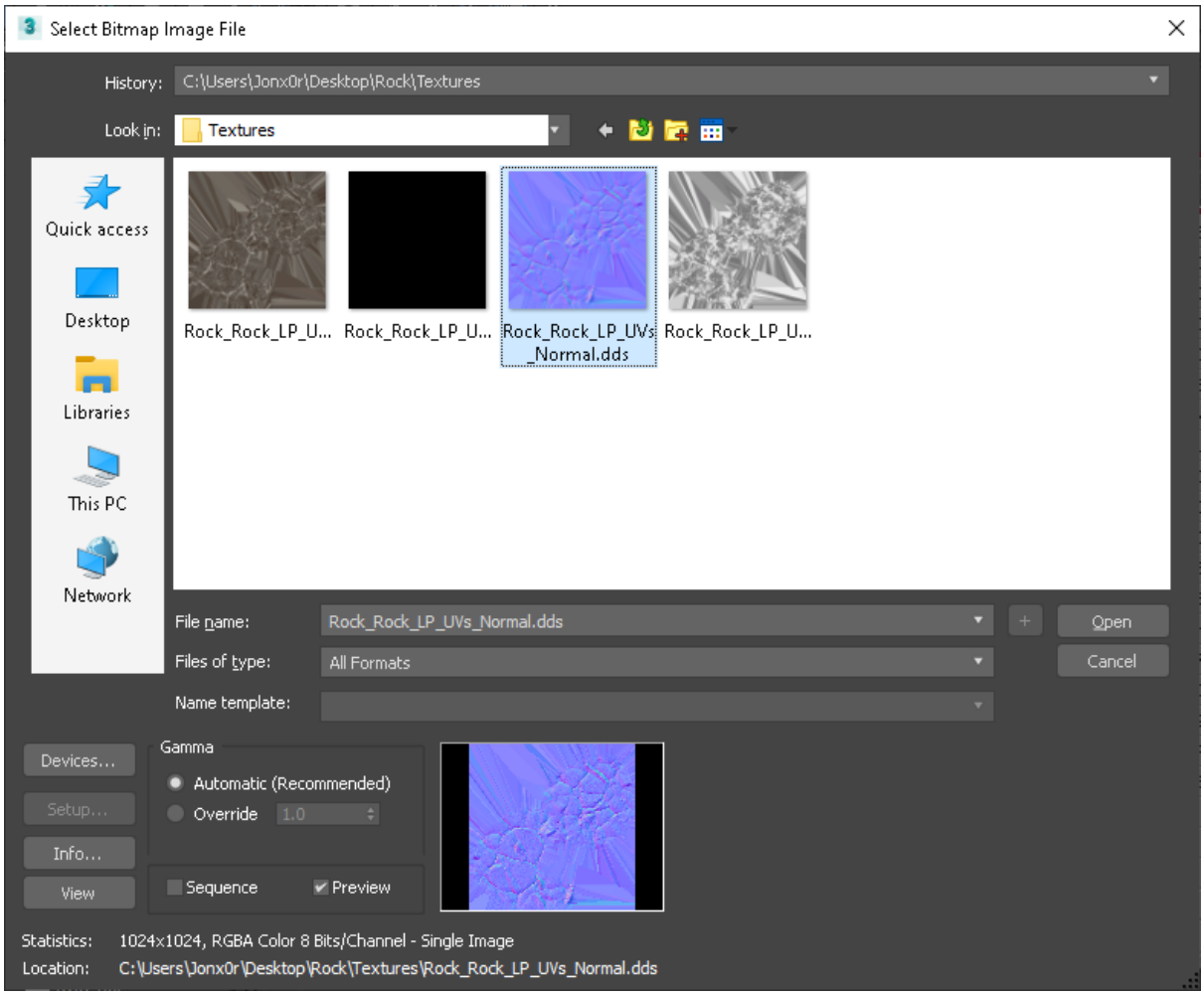


Figure 1172 - Changing the normal map.

The material layout should now look like so:

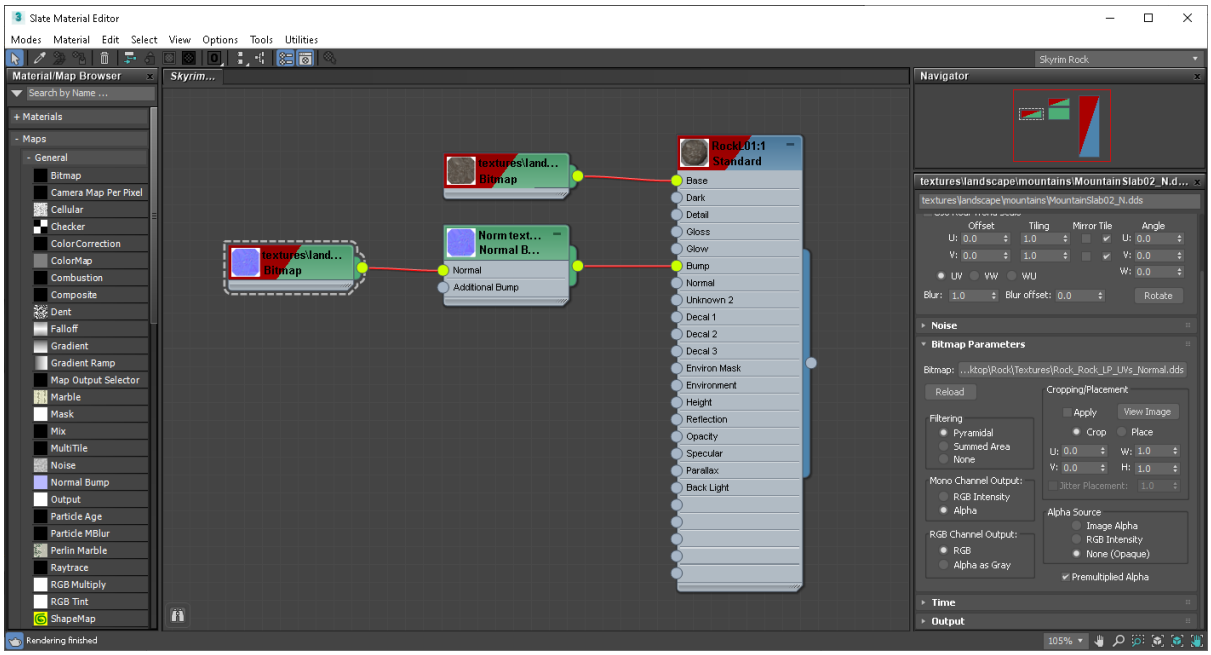


Figure 1173 - Updated diffuse and normal maps.

To apply the material to our custom asset, select it in the viewport then click on the 'Assign Material to Selection' button in the material editor.

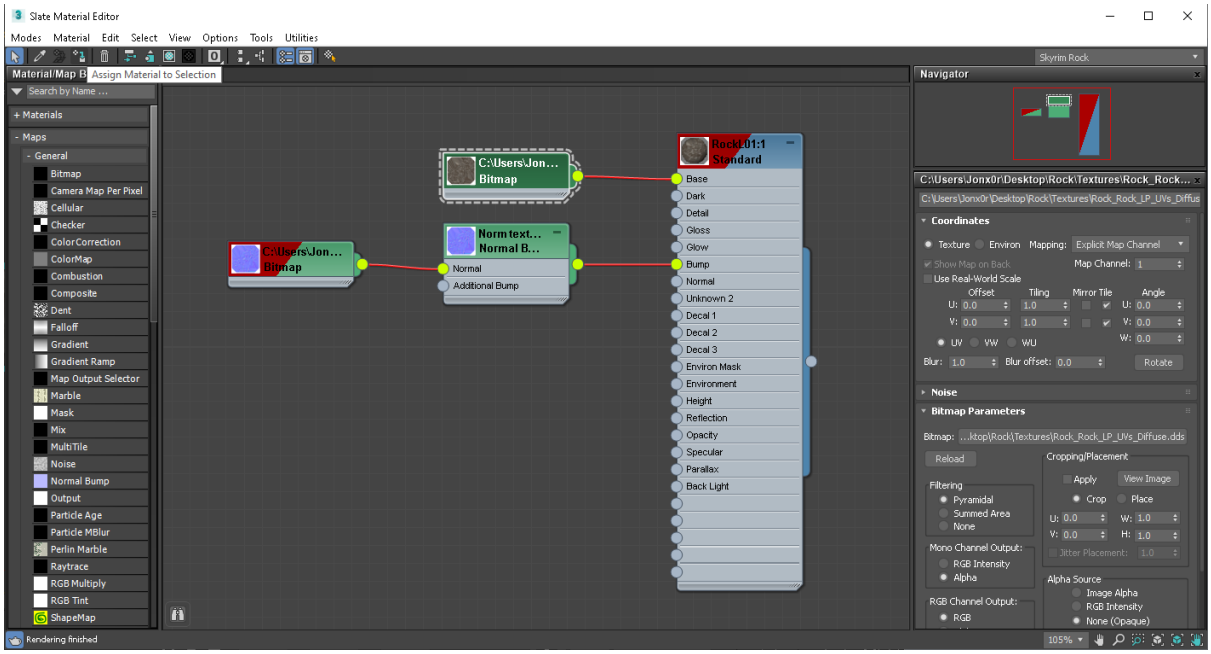


Figure 1174 - Assigning the material to our custom asset.

The specular map can also be added in the same way, but for this specific asset with a rough matte surface it isn't really necessary.

If any parts of the model needed to glow, we would connect the emissive map to Glow.

The viewport might still show the old material, so click on Standard and go to Materials > Shaded Materials with Maps to refresh it.

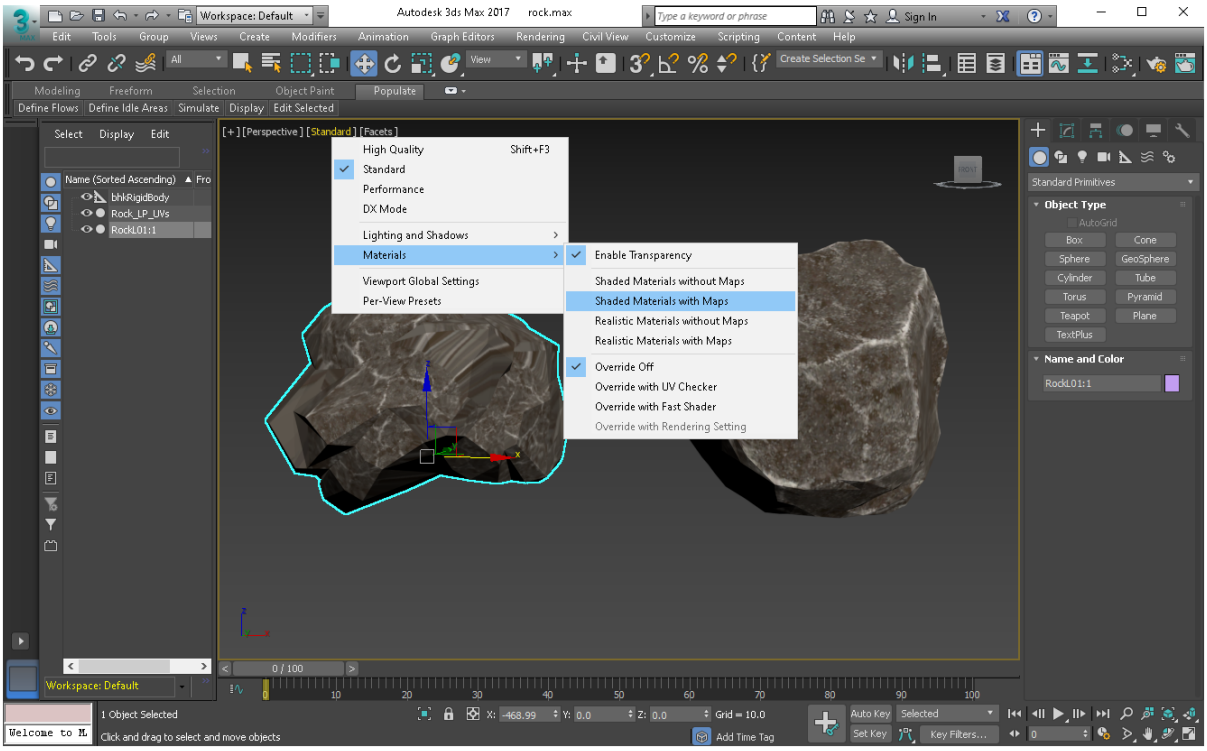


Figure 1175 - Refreshing the material view in viewport.

That's basically it. We no longer need the Skyrim asset at this stage so it can be deleted.

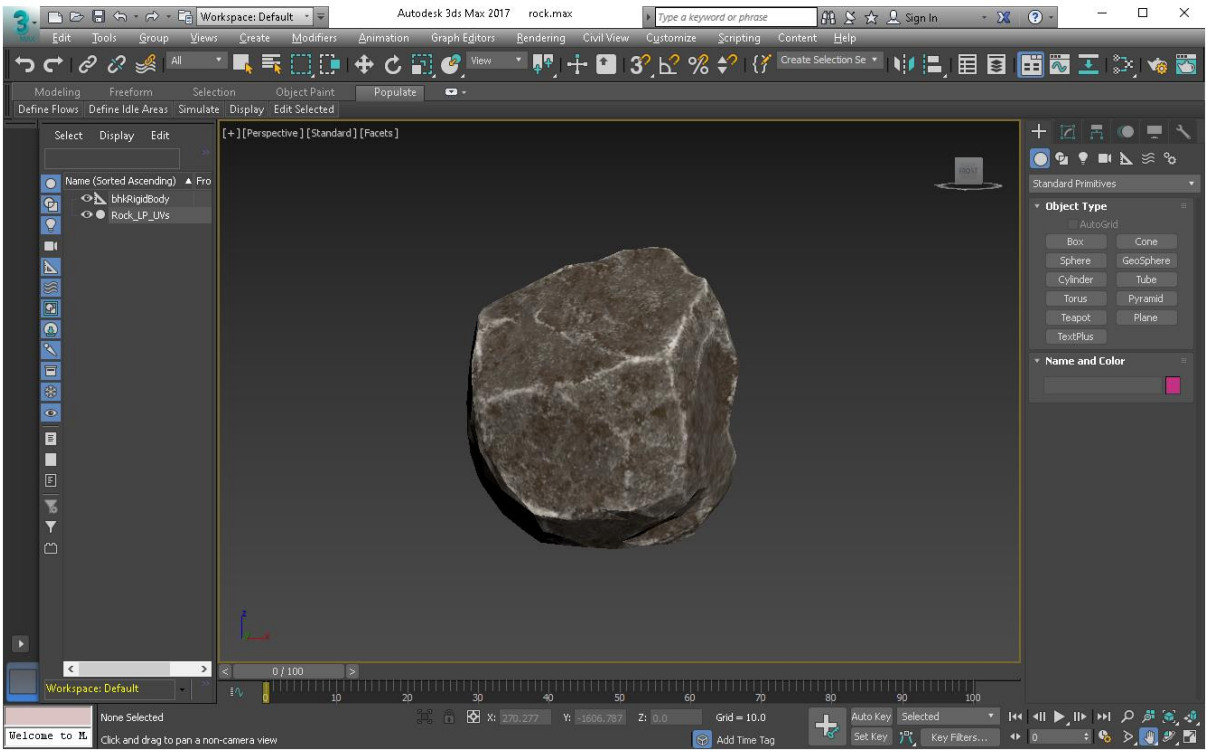


Figure 1176 - Custom asset textured in 3ds Max.

EXPORTING ASSETS FROM 3DS MAX TO SKYRIM

This section will cover the process of exporting a model from 3ds Max as a .nif and setting it up as a new object in the Creation Kit.

For this section I'll be working with 3ds Max 2017. The NIF Export/Import Plugin for 3ds Max 2017 is available on Nexusmods. For other versions of 3ds Max, try NIF Plugin for 3ds Max 2015-2018 available on Nexusmods.

In this example, I'm going to be exporting a custom wine bottle and will be setting it up in the Creation Kit as a new consumable item.

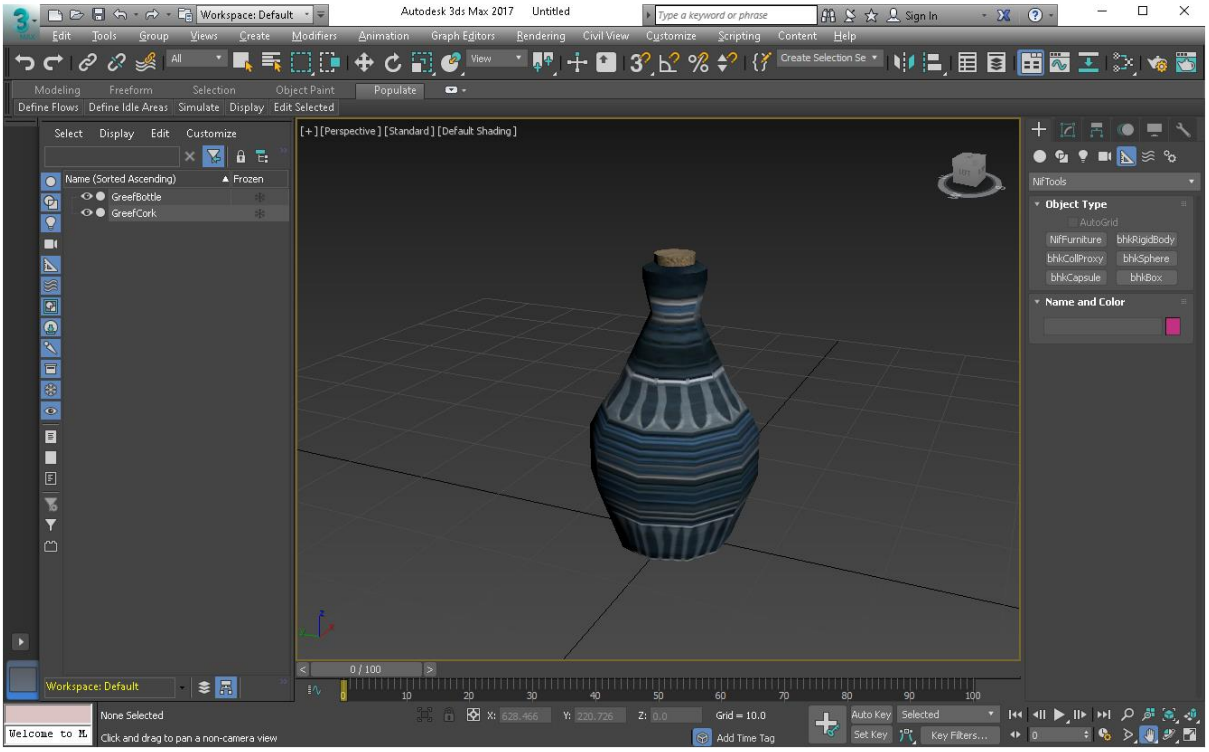


Figure 1177 - A model in 3ds Max.

The first thing we need to do is add collision.

Let's start by adding a bhkRigidBody object.

Under the Create tab, go to Helpers and click on bhkRigidBody.

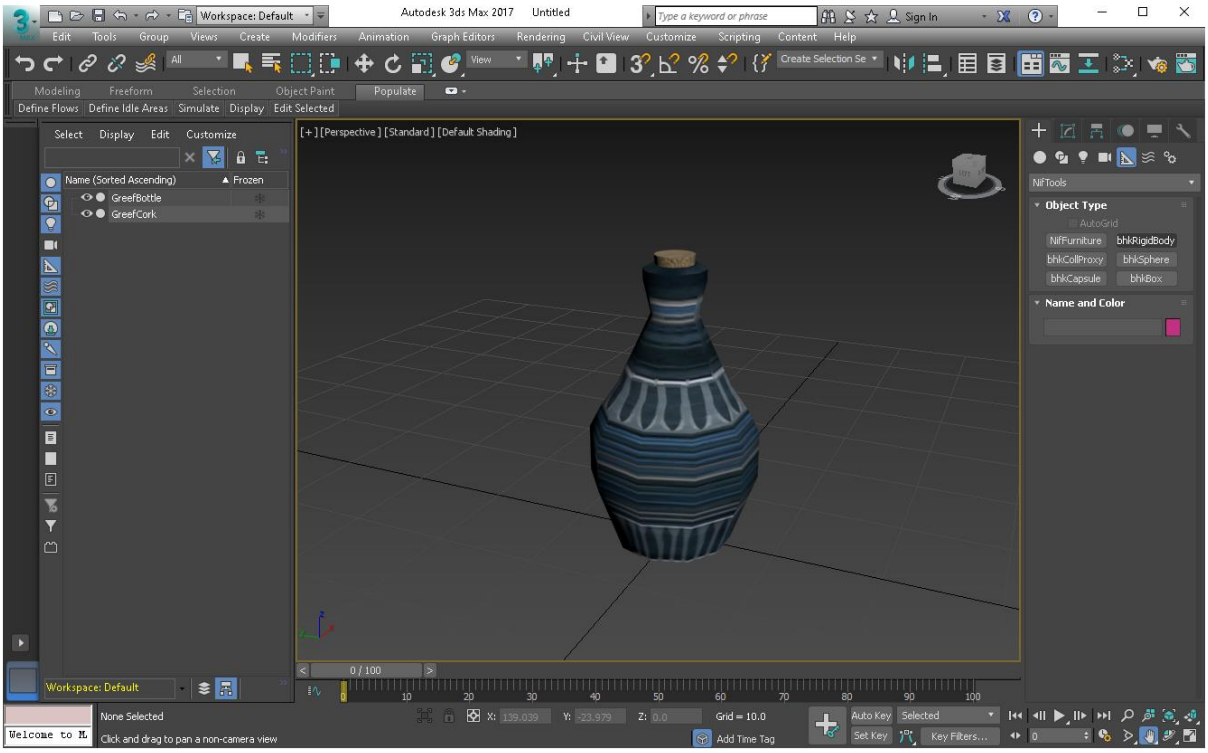


Figure 1178 - Adding a bhkRigidBody object.

Click in your scene to add it.

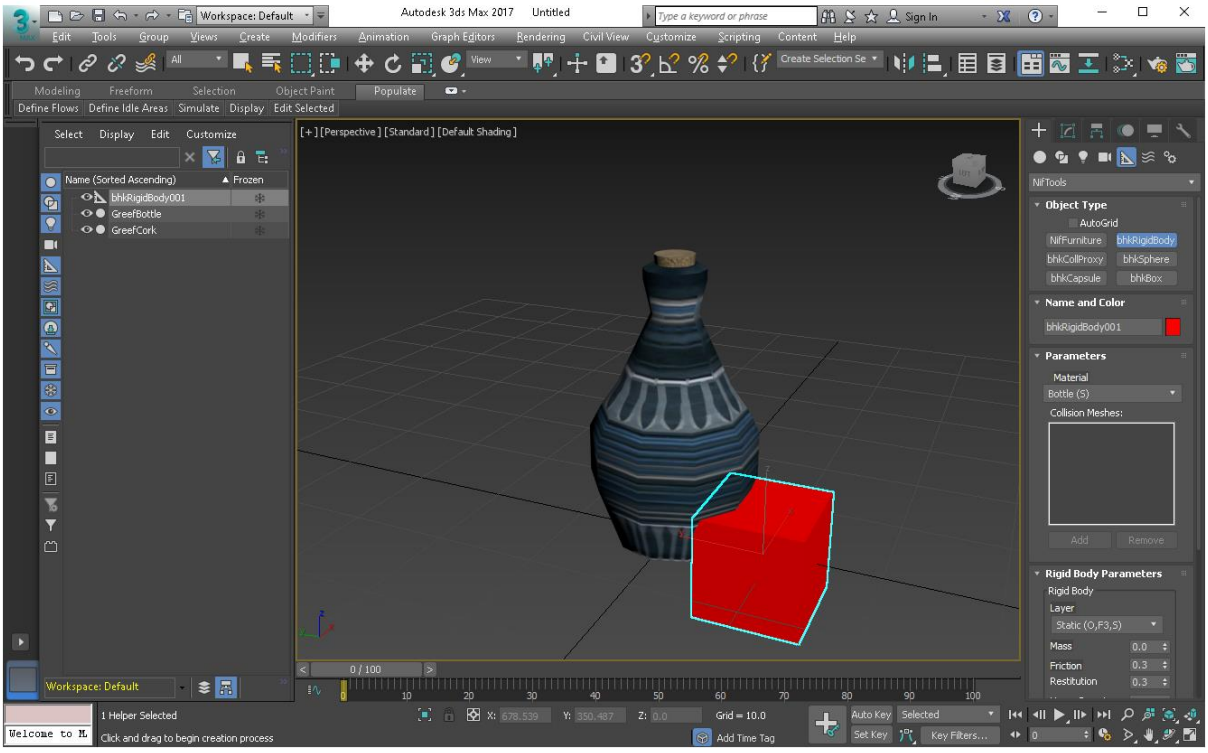


Figure 1179 - bhkRigidBody object added to scene.

Right-click on the Move tool and center the bhkRigidBody object.

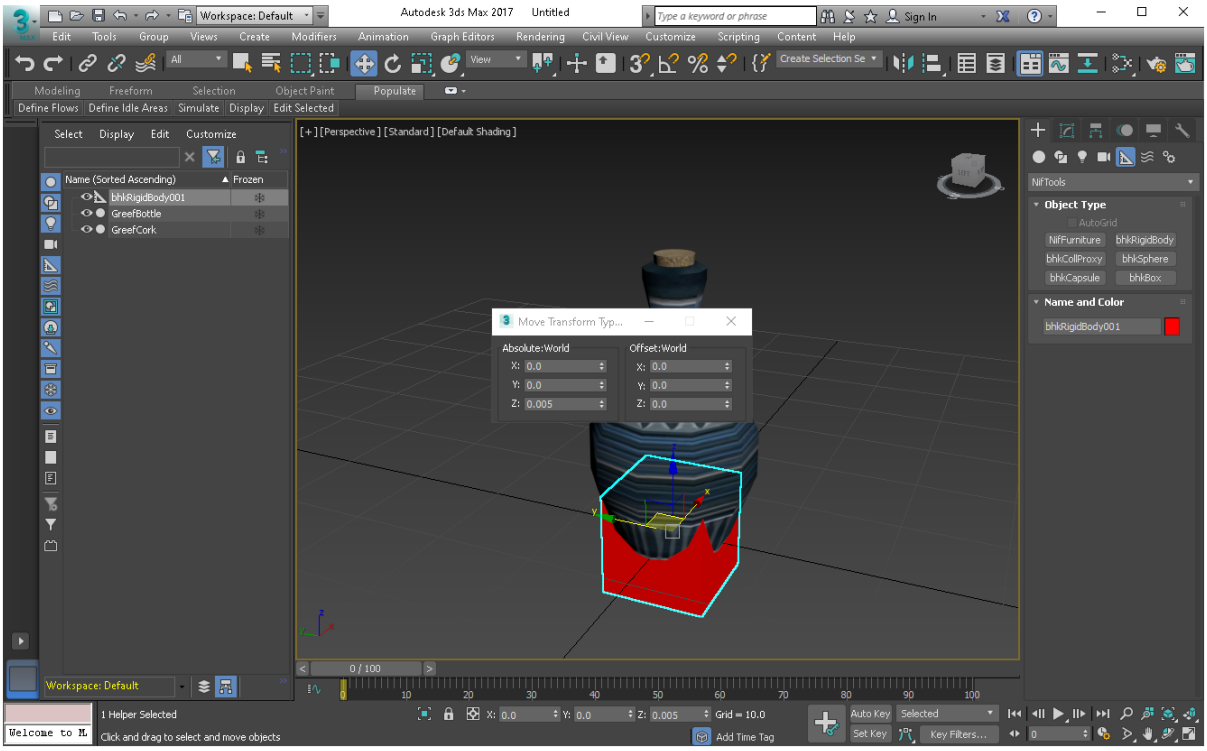


Figure 1180 - bhkRigidBody object centered.

Rename the object 'bhkRigidBody'.

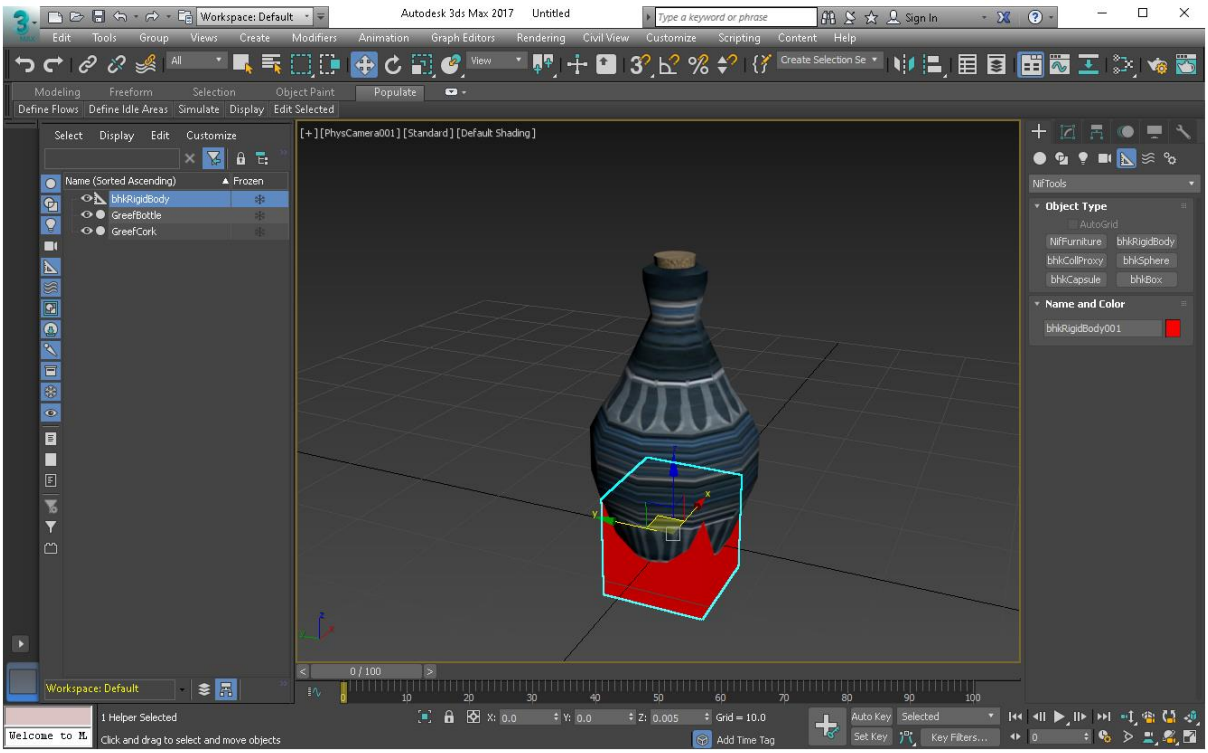


Figure 1181 - Renamed bhkRigidBody.

For this example, I'm just going to duplicate the GreefBottle mesh and use the duplicate as the object's collision. Here I'm using CTRL + V with the GreefBottle selected to duplicate it. I set the duplicate as a Copy of the original object and named it GreefBottle_collision.

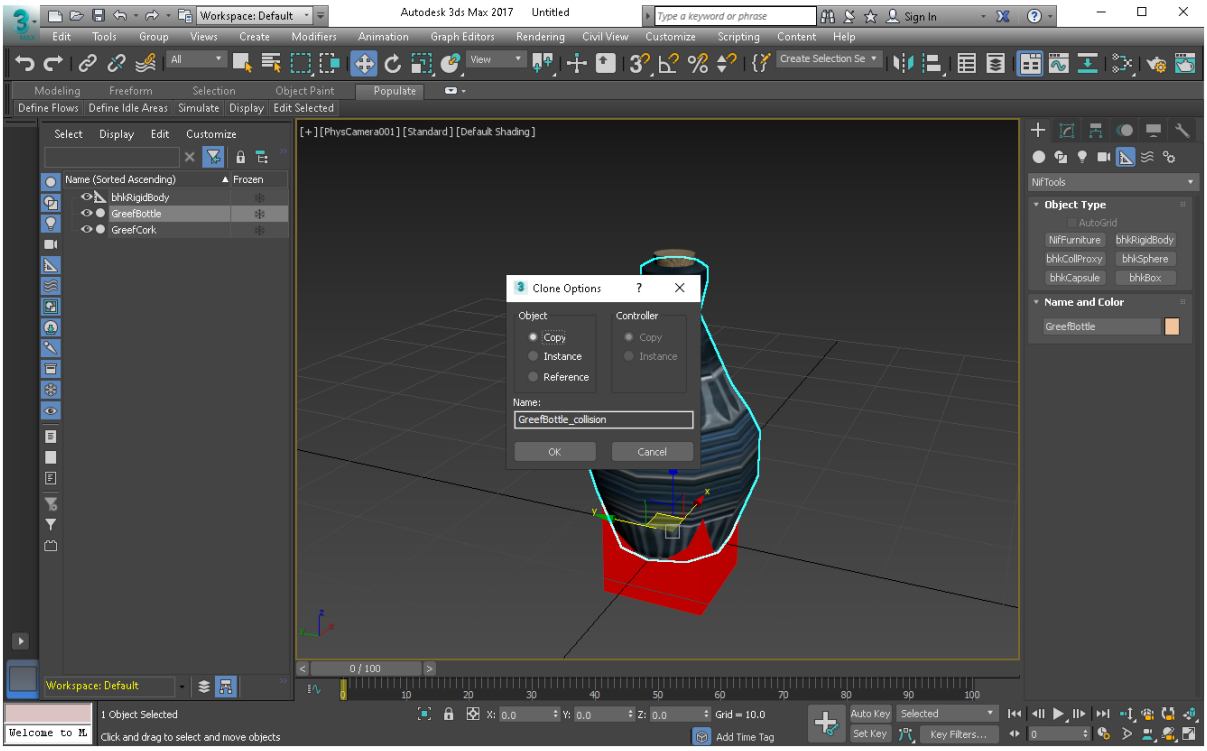


Figure 1182 - Duplicating the mesh to use as collision.

With the collision mesh selected, go to Modifiers and add bhkRigidBodyModifier.

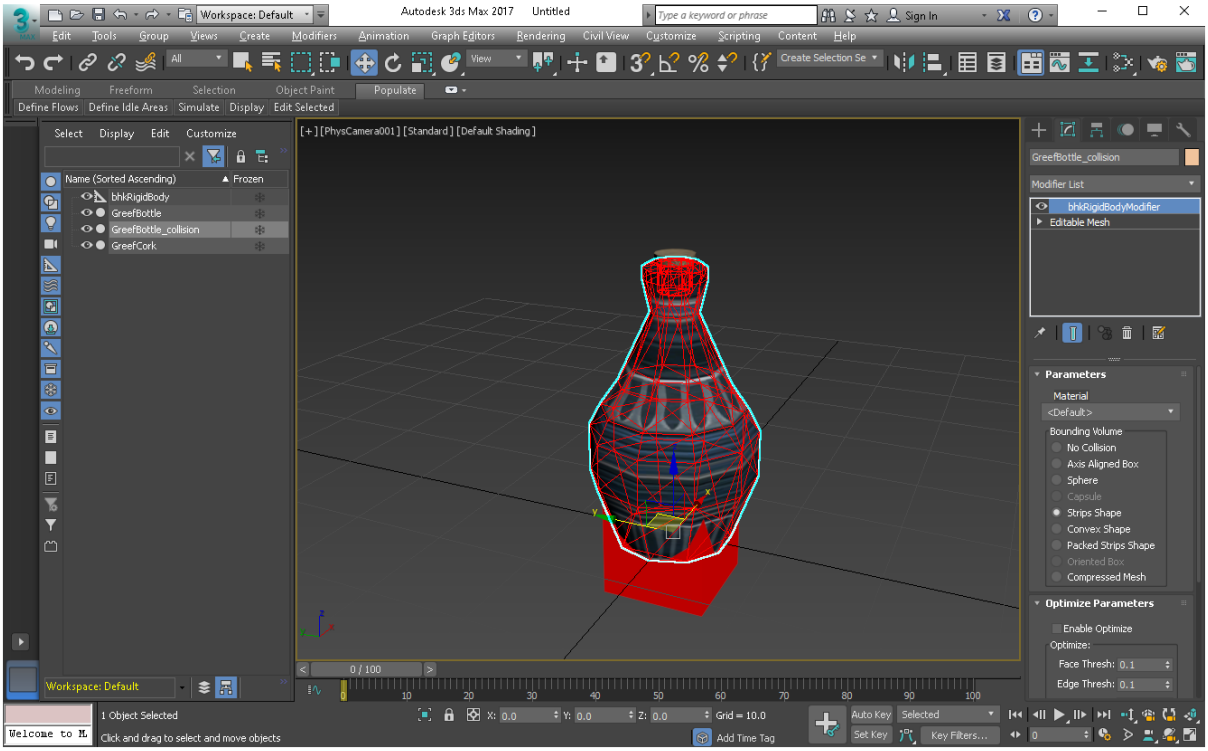


Figure 1183 - Adding bhkRigidBody modifier to the collision object.

For this object, set the Material to 'Bottle (S)' and the Bounding Volume to Convex Shape.

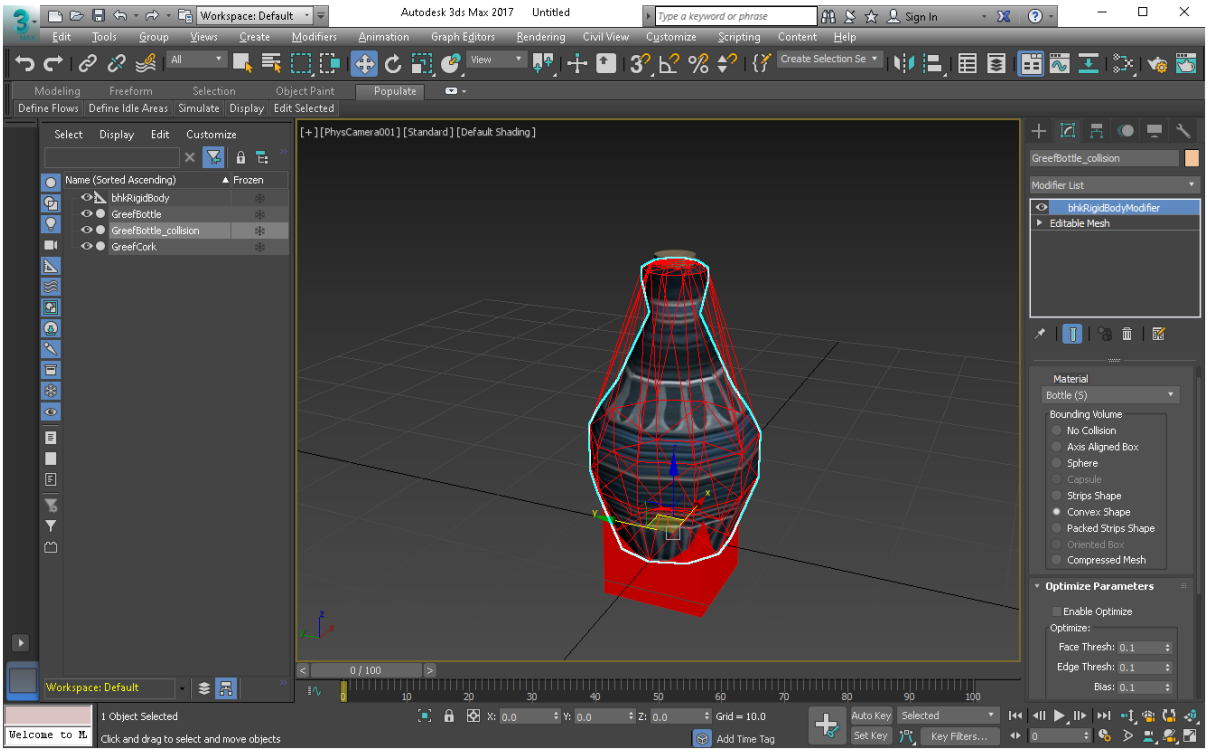


Figure 1184 - Setting the bounding volume.

If you were working with a static mesh select Compressed Mesh instead.

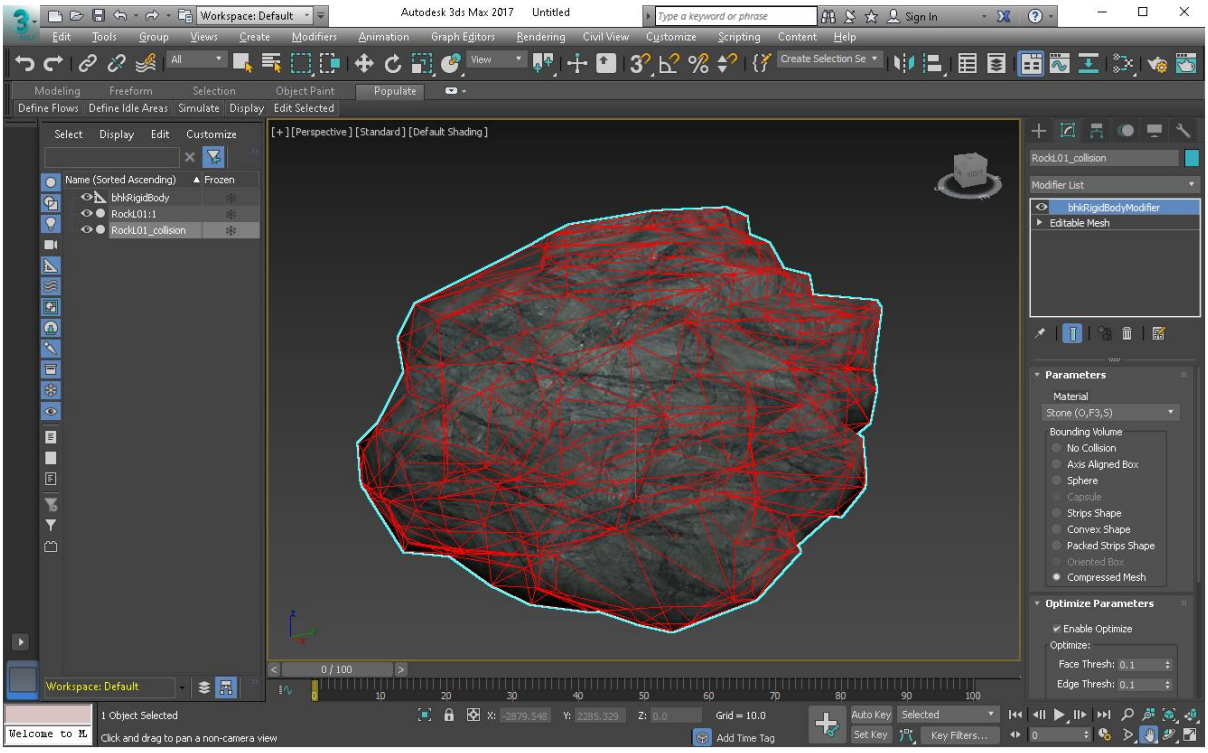


Figure 1185 - RockL01 collision.

Scroll down and tick 'Enable Optimize'. This should render the geometry as quads where possible rather than in triangles.

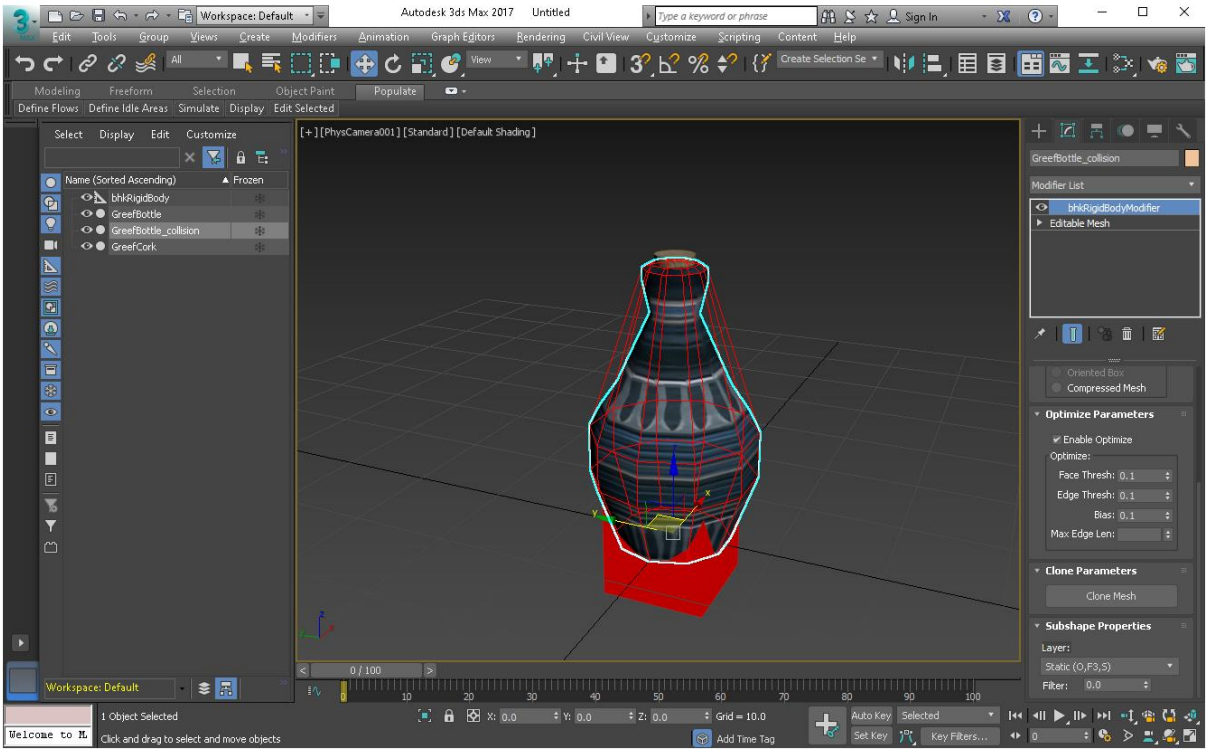


Figure 1186 - Collision geometry optimized.

Go back to the bhkRigidBody object. Under Collision Meshes click Add.

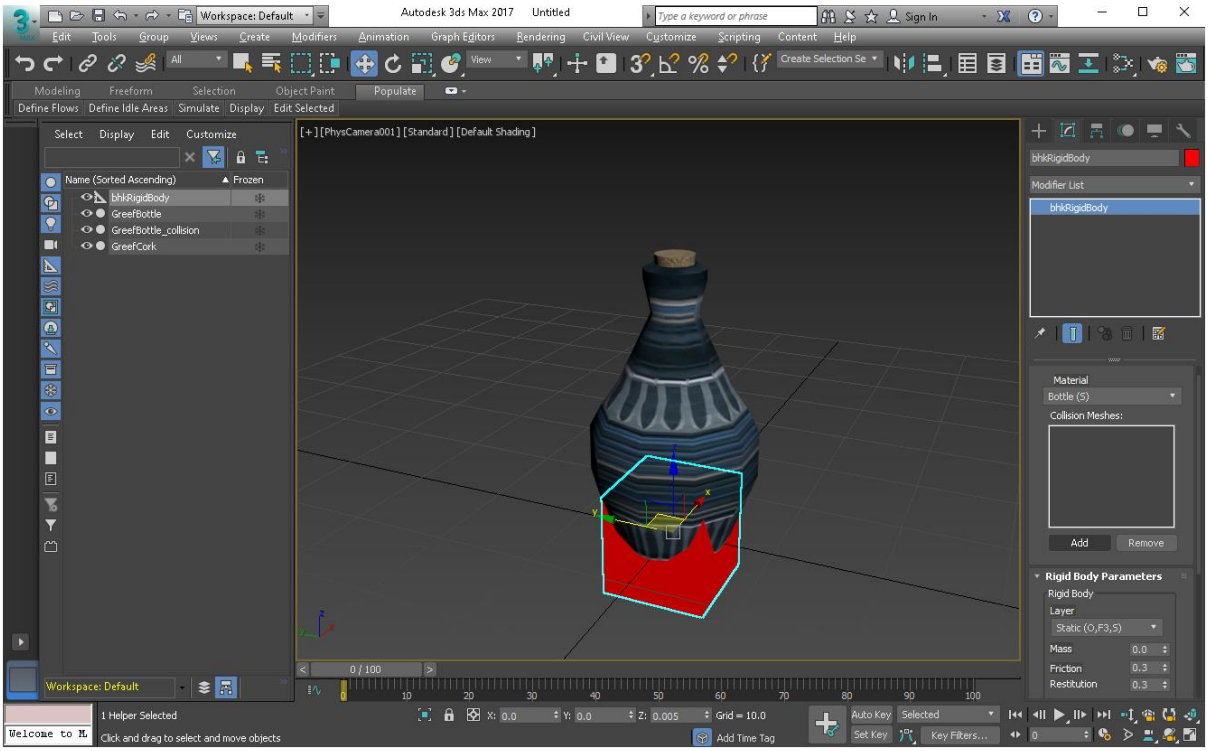


Figure 1187 - Adding a collision mesh to bhkRigidBody.

Add the collision mesh by clicking on it in the scene explorer.

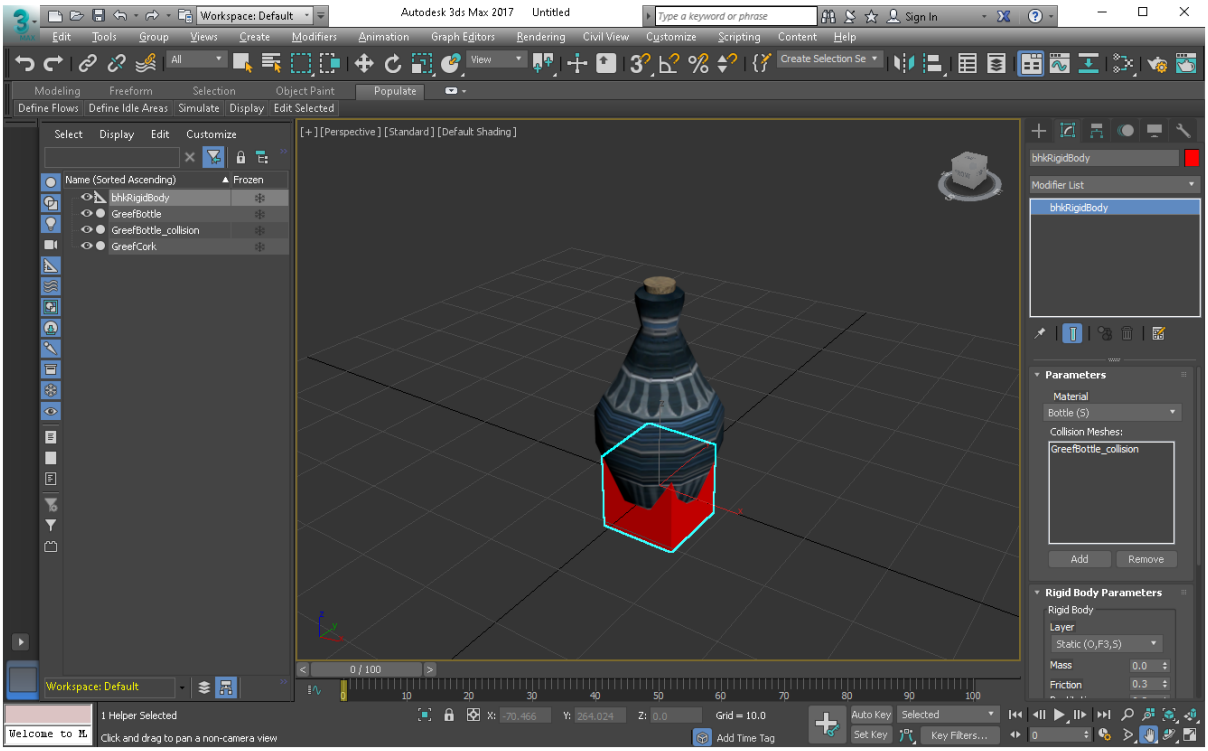


Figure 1188 - Adding the collision mesh to bhkRigidBody.

Now we can export the model from 3ds Max. To accomplish this, go to Export.

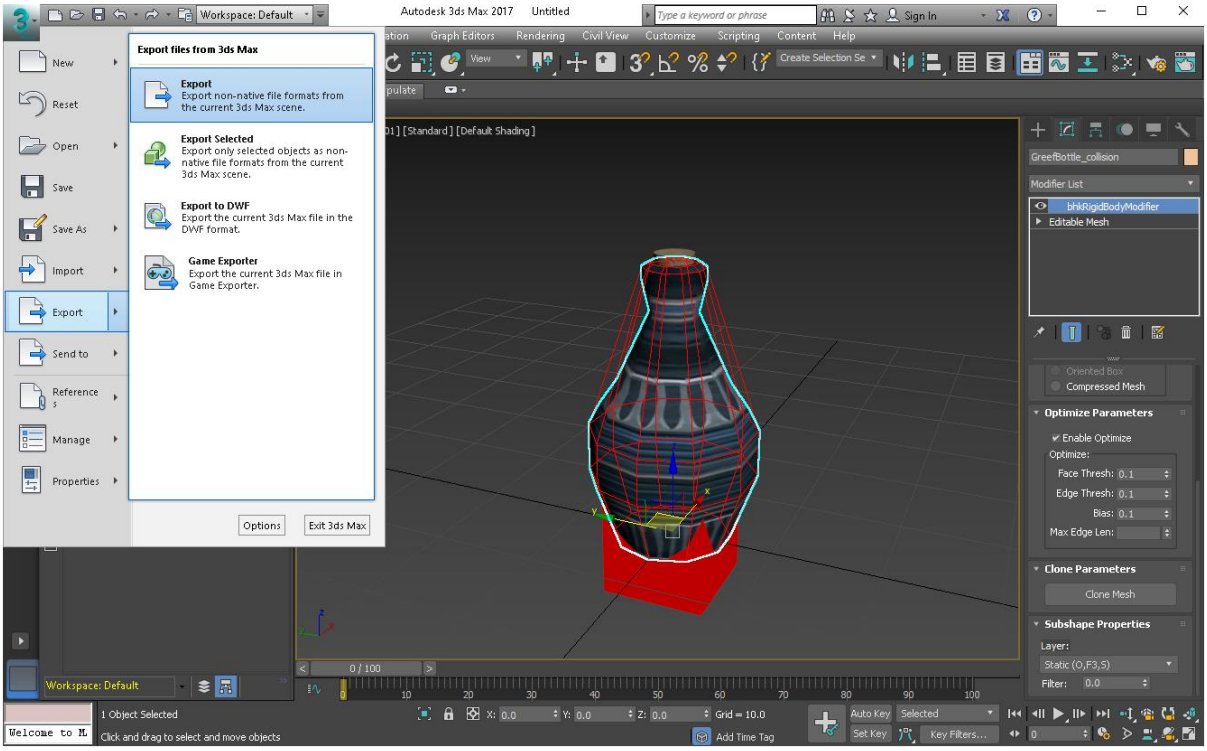


Figure 1189 - Exporting a model.

Set 'Save as type' to 'NetImmerse/Gamebryo (*KF,*.NIF)'. Set the location to export the model to and enter in a file name. In my example I'm saving to the Skyrim\Data\Meshes\clutter\wine folder, and the model is going to be called 'wtgreef.nif'.

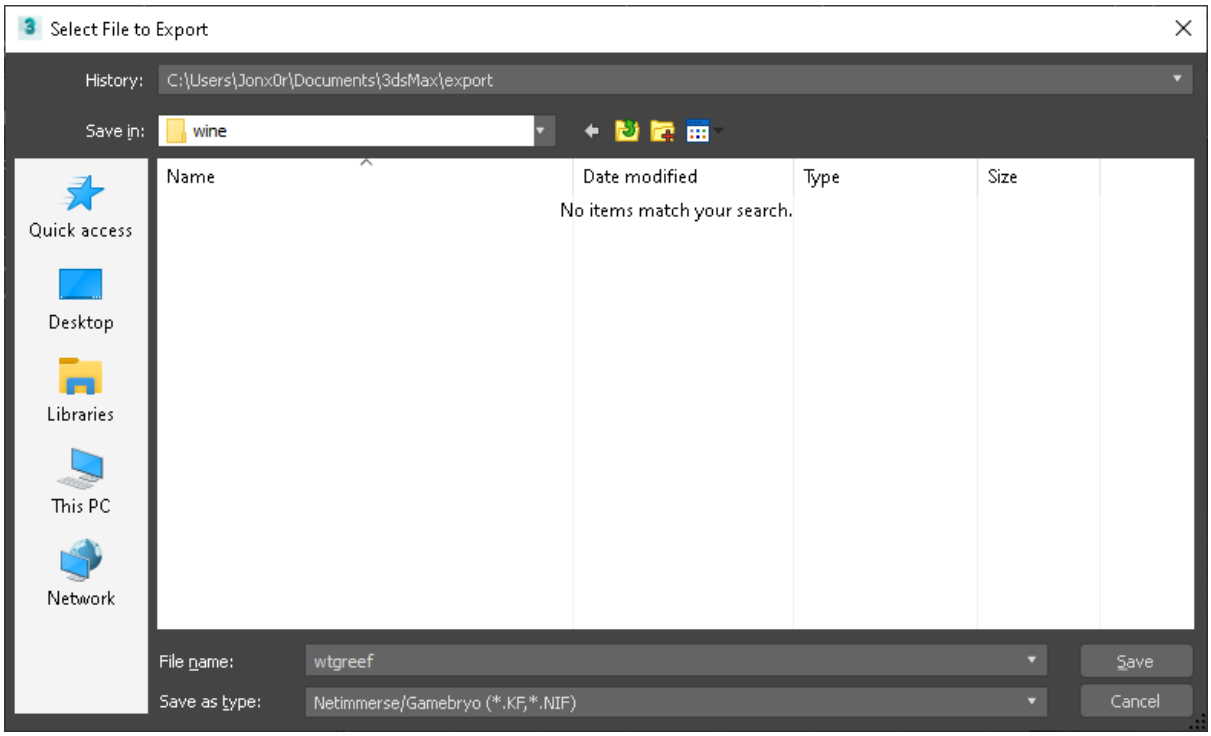


Figure 1190 - Set 'Save as type' to 'NetImmerse/Gamebryo (.KF,*.NIF)'.*

Set the Game to Skyrim.

Under the Export section I ticked Collision. If your model includes lights, cameras and hidden nodes, tick those tick boxes as well.

Under Mesh, tick Flatten Hierarchy, Update Tangent Space, Collapse Transforms and Zero Transforms.

For my example, I unticked the options under Skin Modifier as it doesn't apply to the model I'm exporting.

Under Animation, I left the drop-down set to 'NIF w/o Animation' and left 'Transforms' ticked.

Under Scene, I set the Root Node Type to BSFadeNode and under Miscellaneous I just left 'Sort Nodes' ticked.

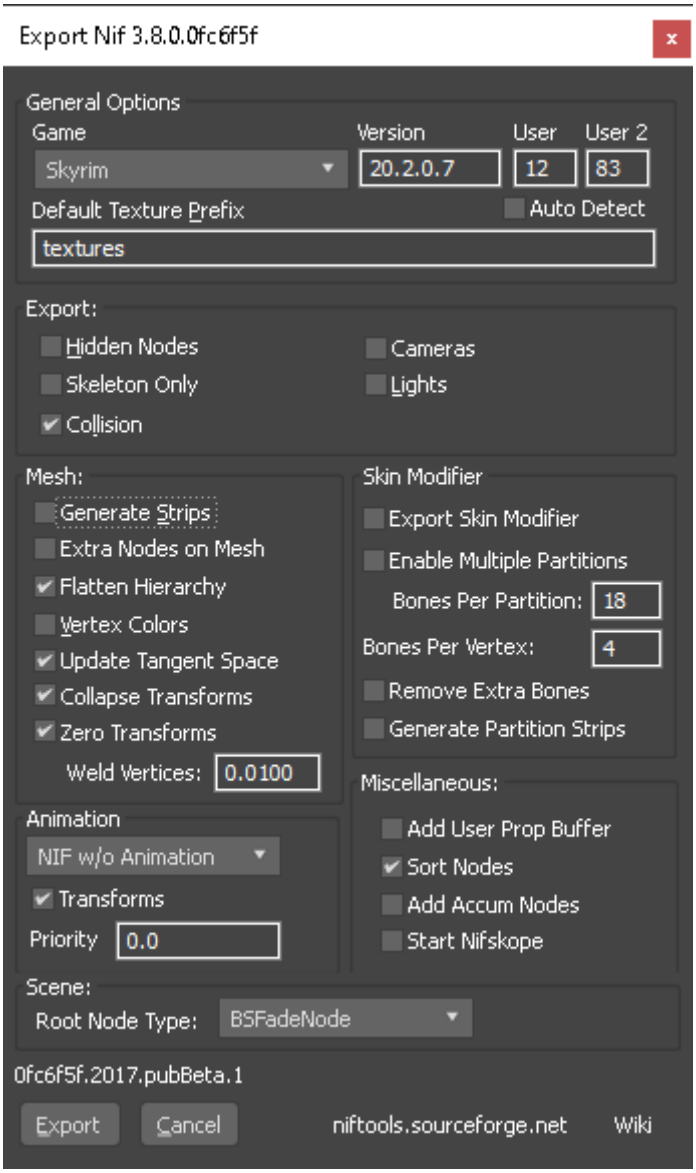


Figure 1191 - Export options.

Click Export to create the .nif.

For more information on what each export option does, see the 3ds Max Export article on the Beyond Skyrim wiki.

Important: If we add this .nif to a mod right now and launched Skyrim, Skyrim will crash as soon as it tries to render it.

There are several ways to work around this issue. Typically what I do is take a similar .nif exported from the .bsa archives and **turn it into the new .nif** by copying across the NiTriShapeData and bhkConvexVerticesShape blocks and the texture paths from the .nif we exported from 3ds Max.

Open the original exported .nif in NifSkope.

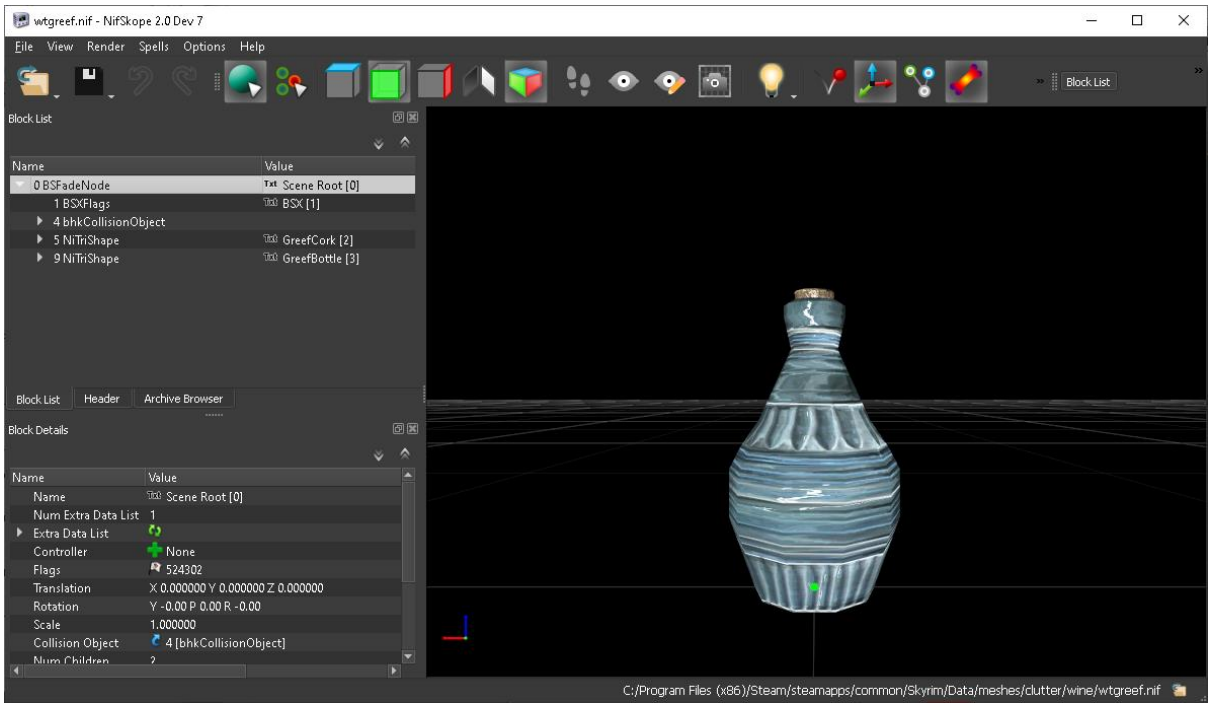


Figure 1192 - The .nif in NifSkope.

Open a second instance of NifSkope and open a similar .nif. For this example, I'm going to be working with winebottle01a.nif.

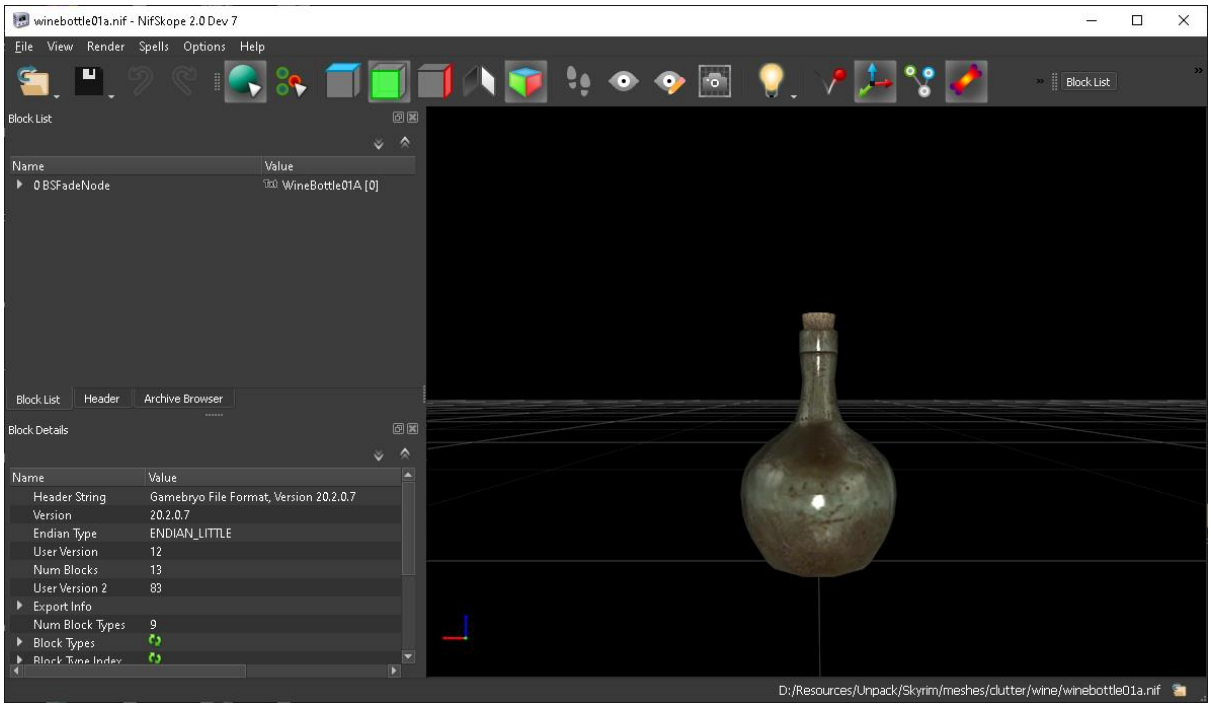


Figure 1193 - winebottle01a opened in a second instance of NifSkope.

In the source .nif, right-click on the NiTriShapeData block and select Block > Copy.

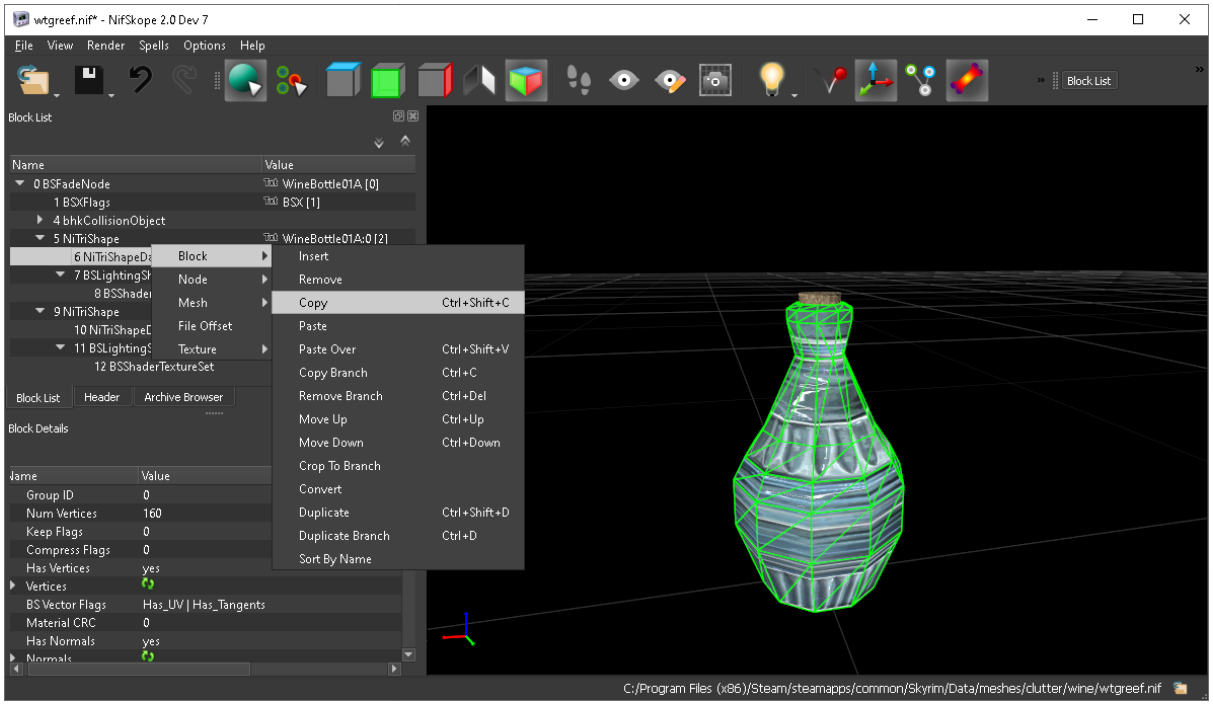


Figure 1194 - Copying the NiTriShapeData block from our source .nif.

Go back to the destination .nif, right-click on its NiTriShapeData block and select Block > Paste Over. It's important that you select Paste Over and not Paste. We need to replace the existing blocks.

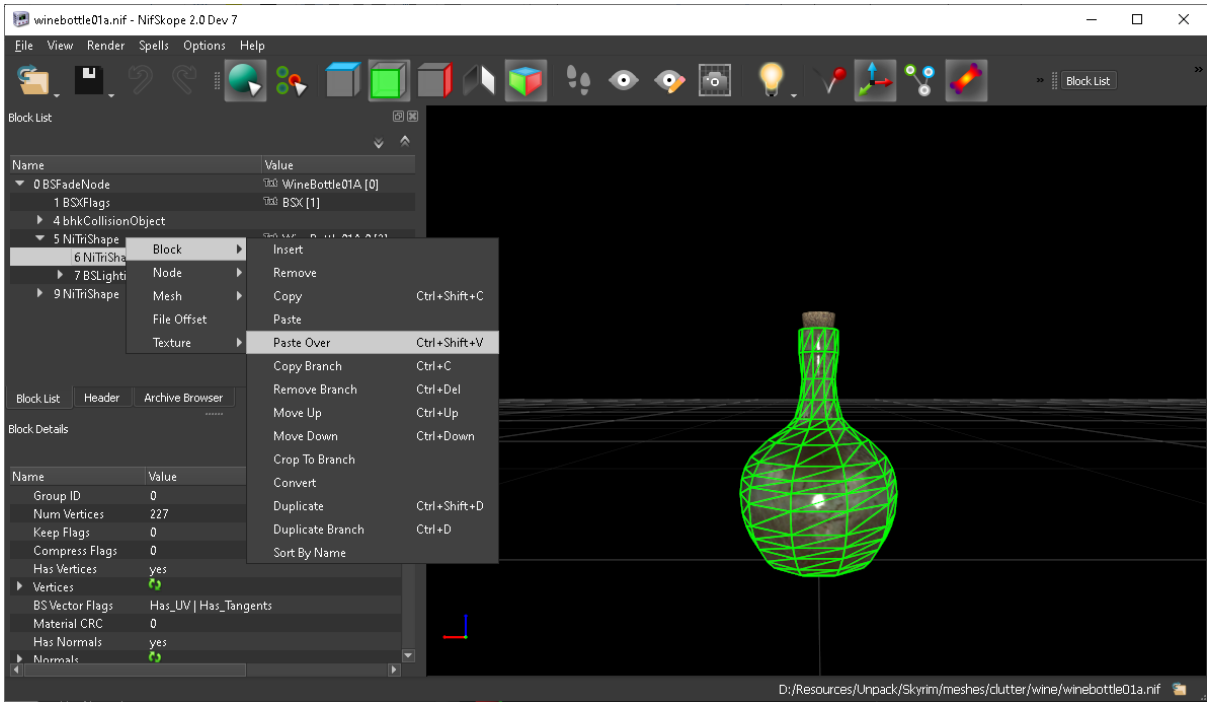


Figure 1195 - Pasting over an existing NiTriShapeData block.

Ok. It has the model, but now we need to transfer the textures.

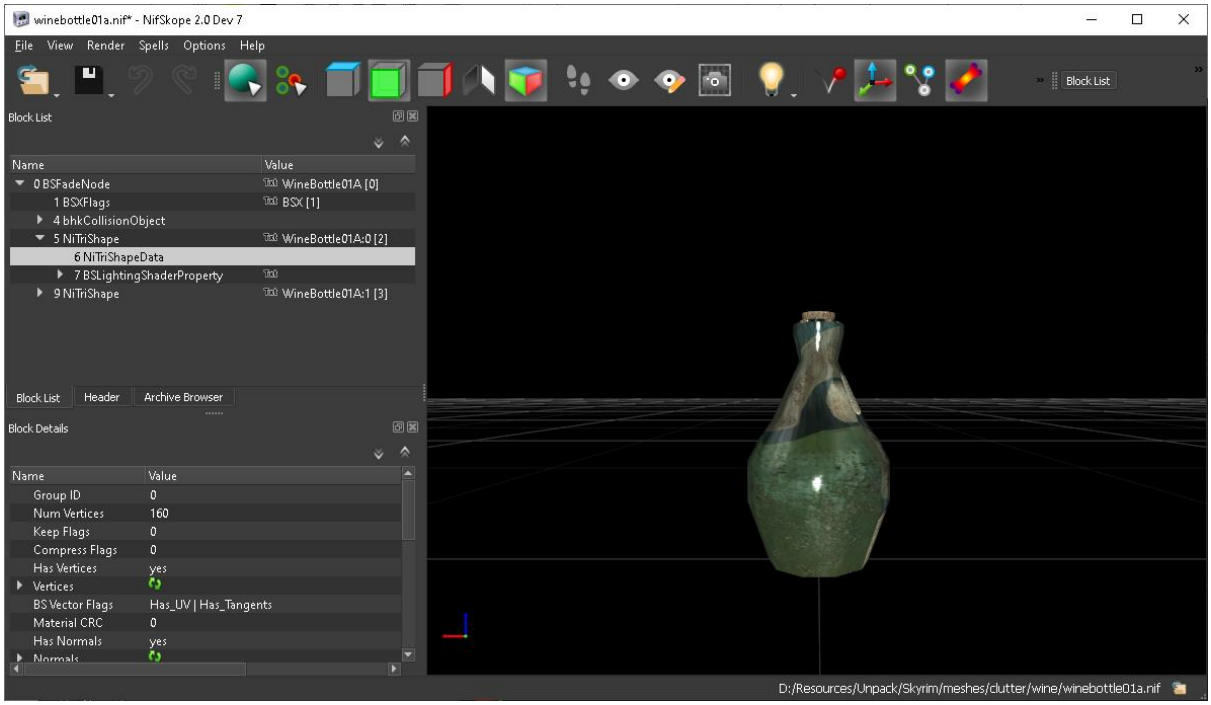


Figure 1196 - NiTriShapeData block replaced.

Copy the texture paths under the BSLightingShaderProperty > BSShaderTextureSet nodes in the source .nif.

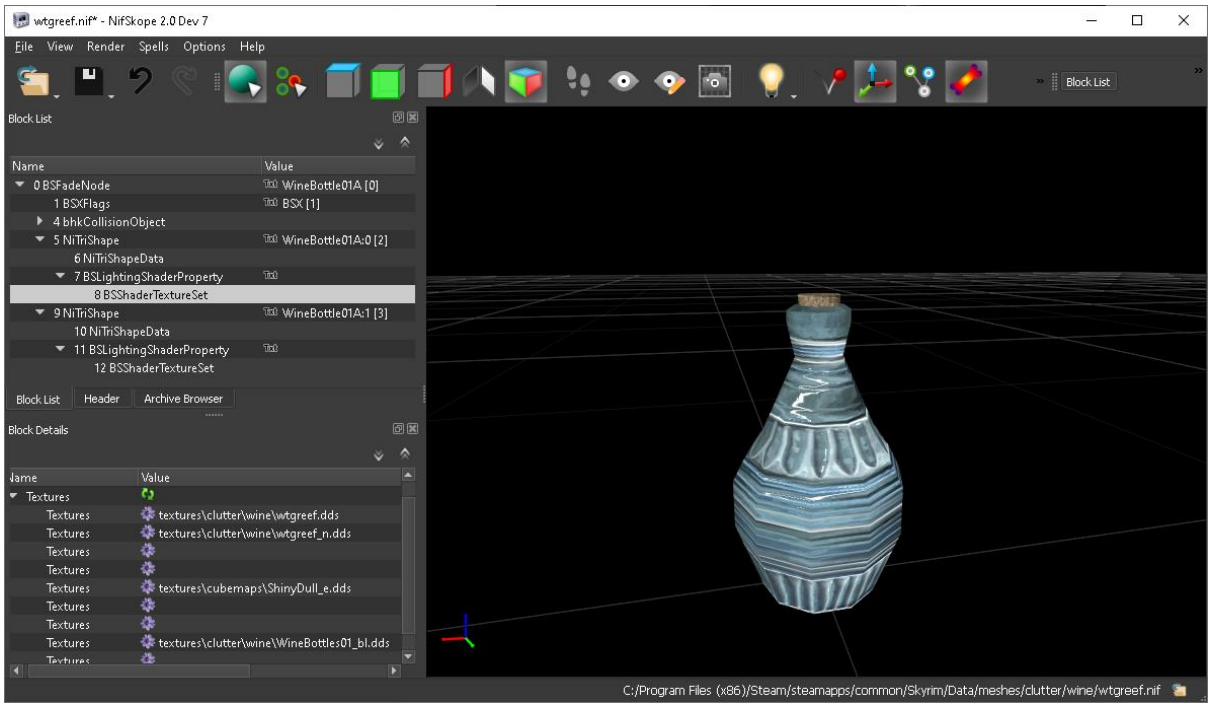


Figure 1197 - Copying texture paths from source .nif.

Replace the texture paths on the destination .nif.

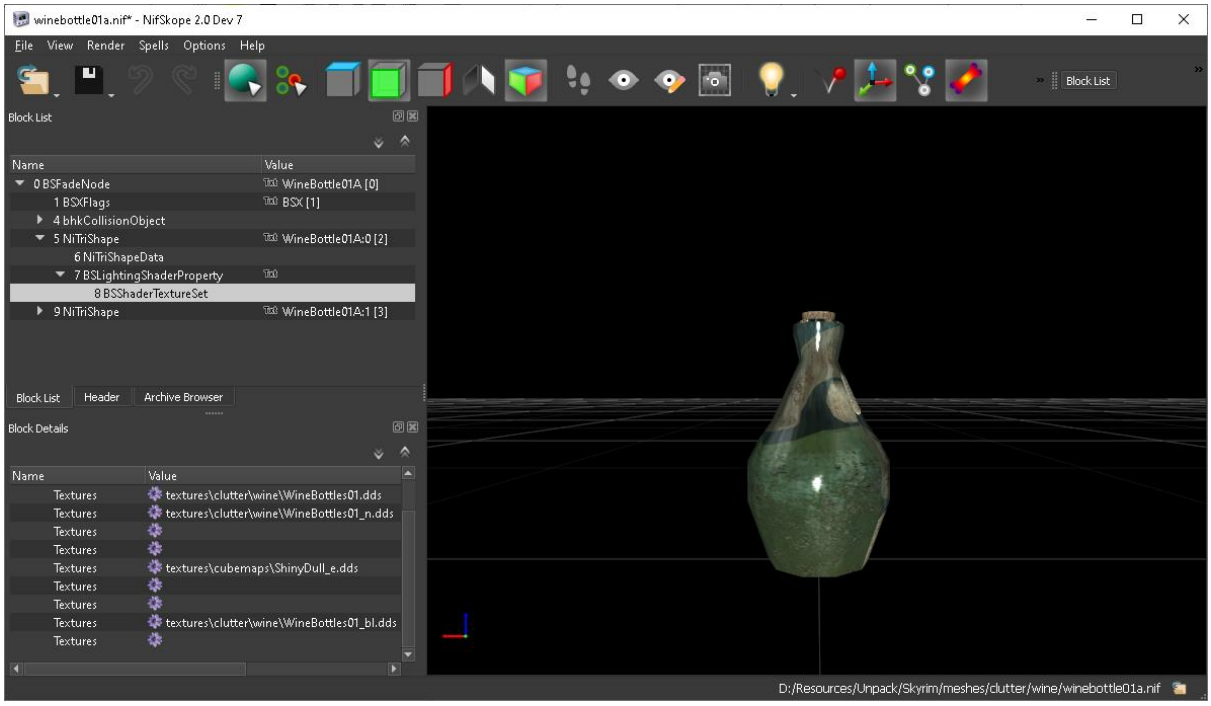


Figure 1198 - Texture paths on destination .nif.

Here's the destination .nif with the mesh and textures of the source .nif.

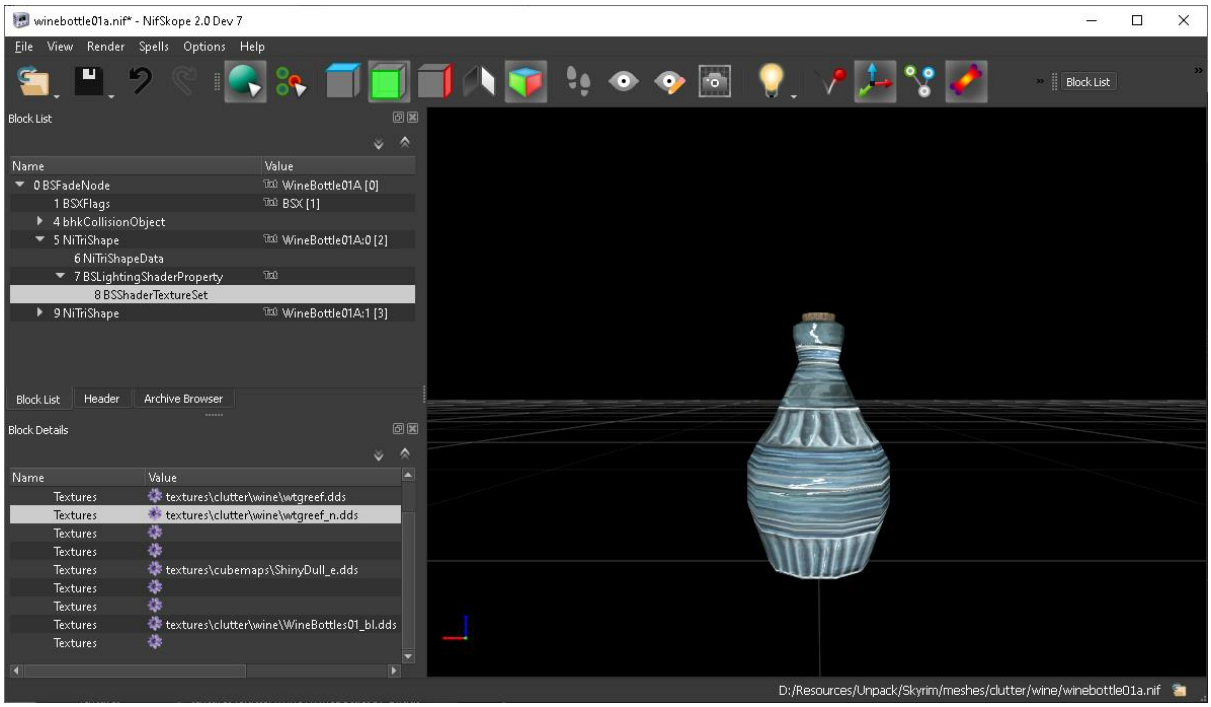


Figure 1199 - Destination .nif with mesh and textures from source .nif.

Repeat these steps for any other separate NiTriShapeData nodes in the model. For this example I also had to replace the cork object too.

Now, to replace the collision. On the source .nif, right-click on the bhkConvexVerticesShape node and select Block > Copy.

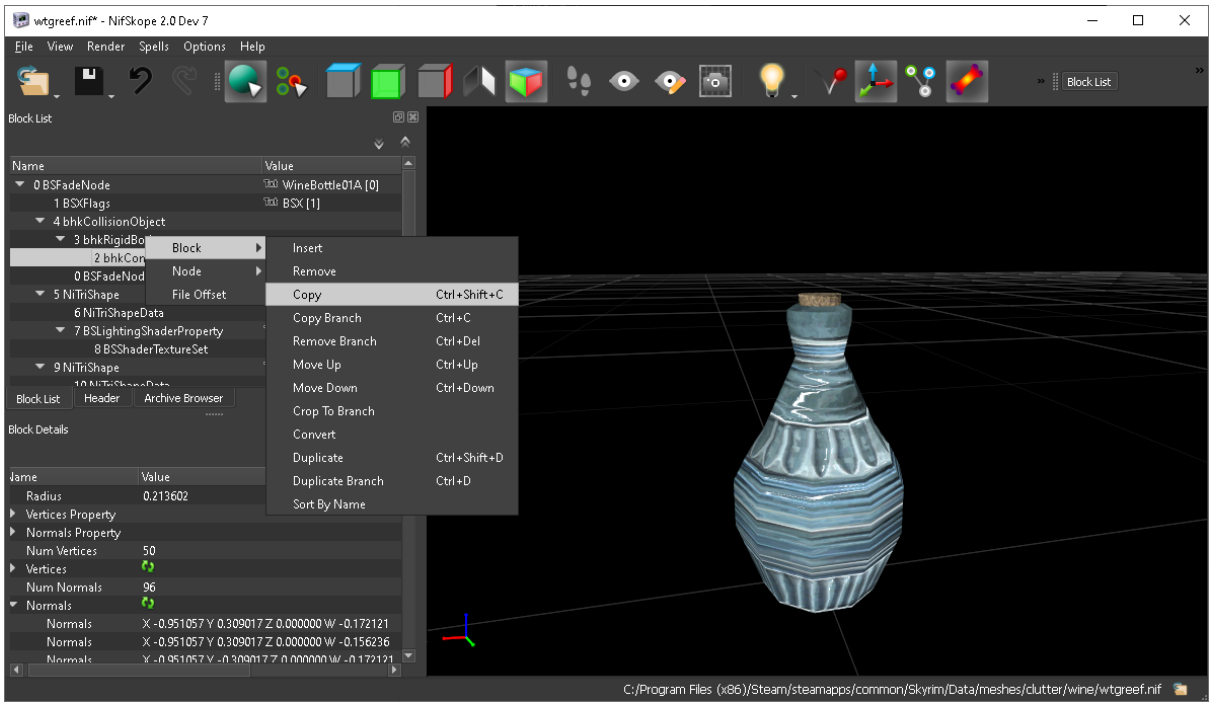


Figure 1200 - Copying the bhkConvexVerticesShape node from the source .nif.

On the destination .nif, right-click on its bhkConvexVerticesShape node and select Block > Paste Over.

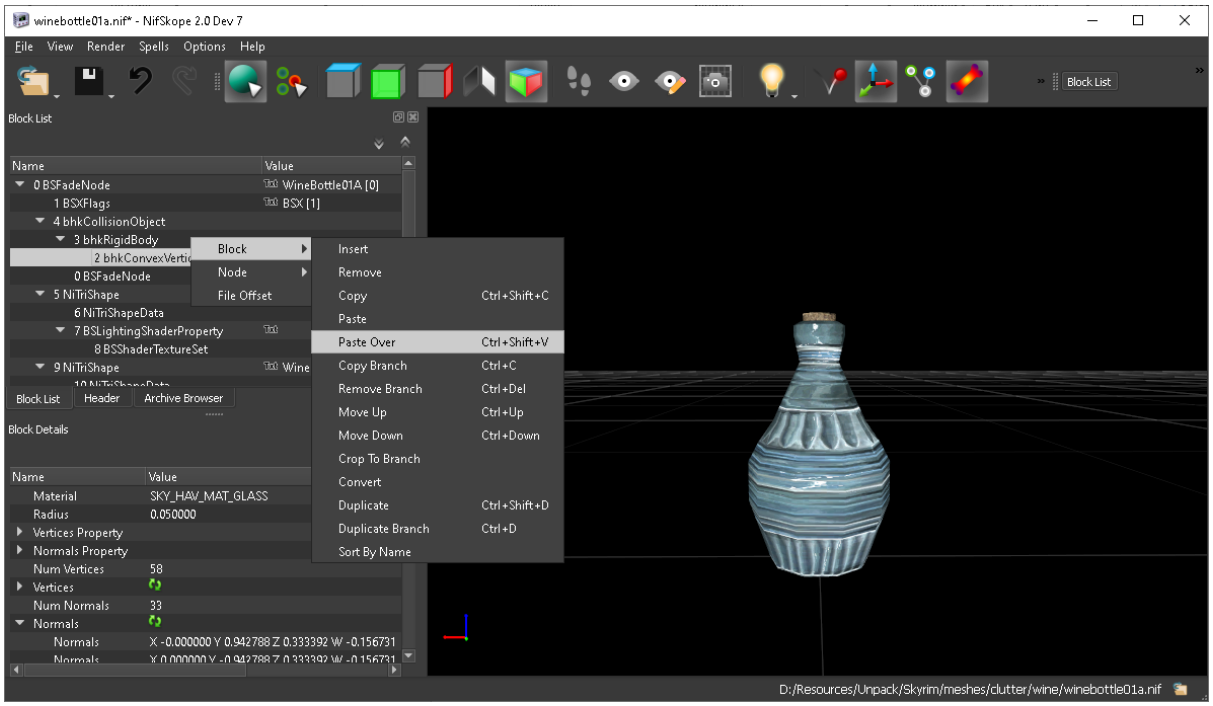


Figure 1201 - Replacing the existing bhkConvexVerticesShape node on the destination .nif.

Alright, time to save our .nif. Go to File > Save As.

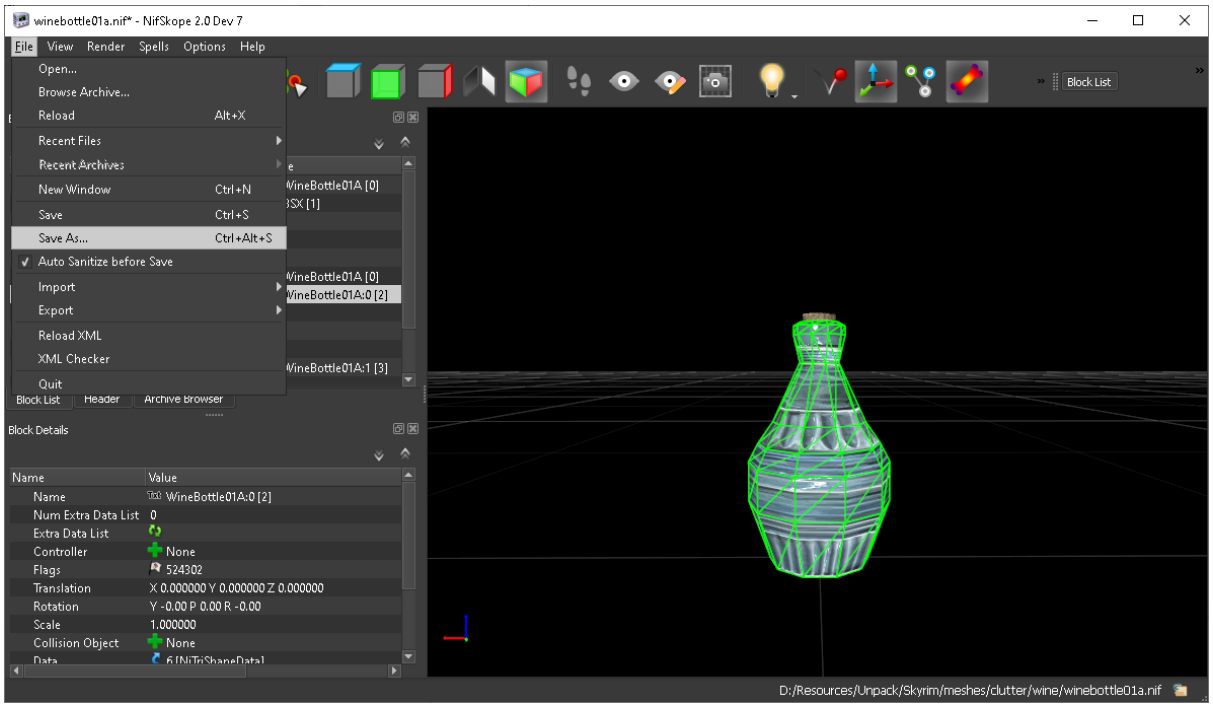


Figure 1202 - Saving the destination .nif.

Overwrite the source .nif that we exported from 3ds Max earlier.

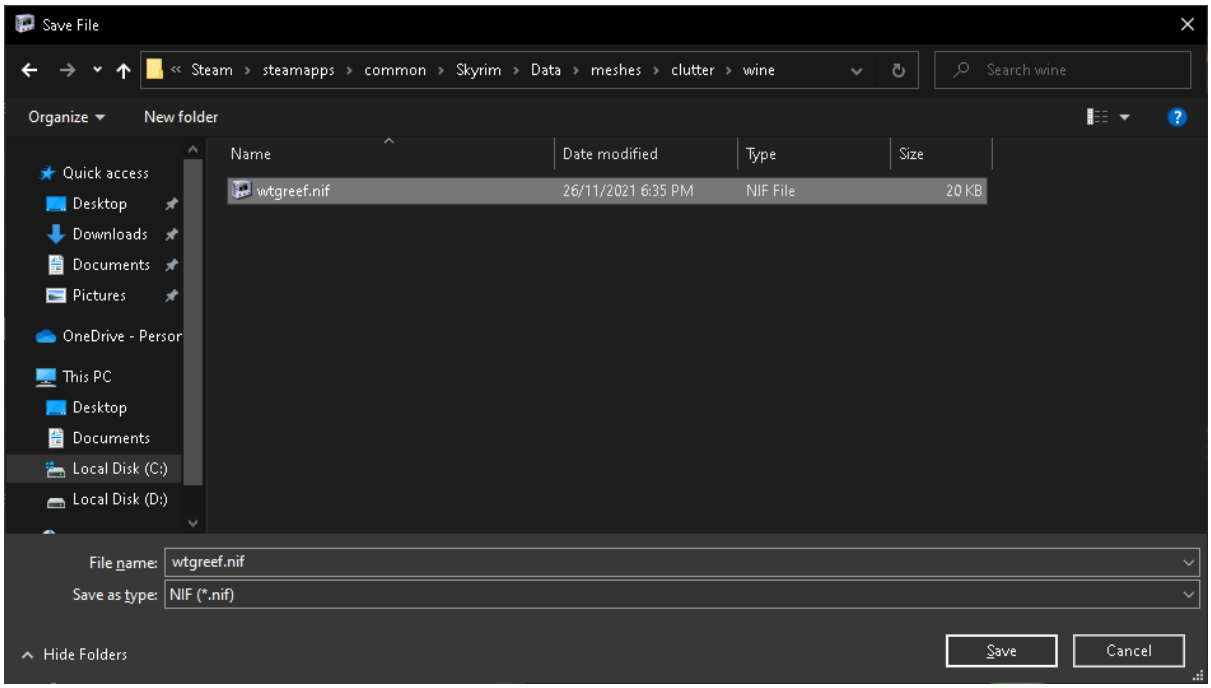


Figure 1203 - Replacing the source .nif with the destination .nif.

Good news. If we launch the game and go to a cell containing the object, the game doesn't crash. The bad news is that its collision radius is too large and is causing our object to float above the surface of the table.



Figure 1204 - New model in-game.

So let's fix that by reducing the collision radius in the .nif.

In the bhkConvexVerticesShape, set the Radius field to 0.000000.

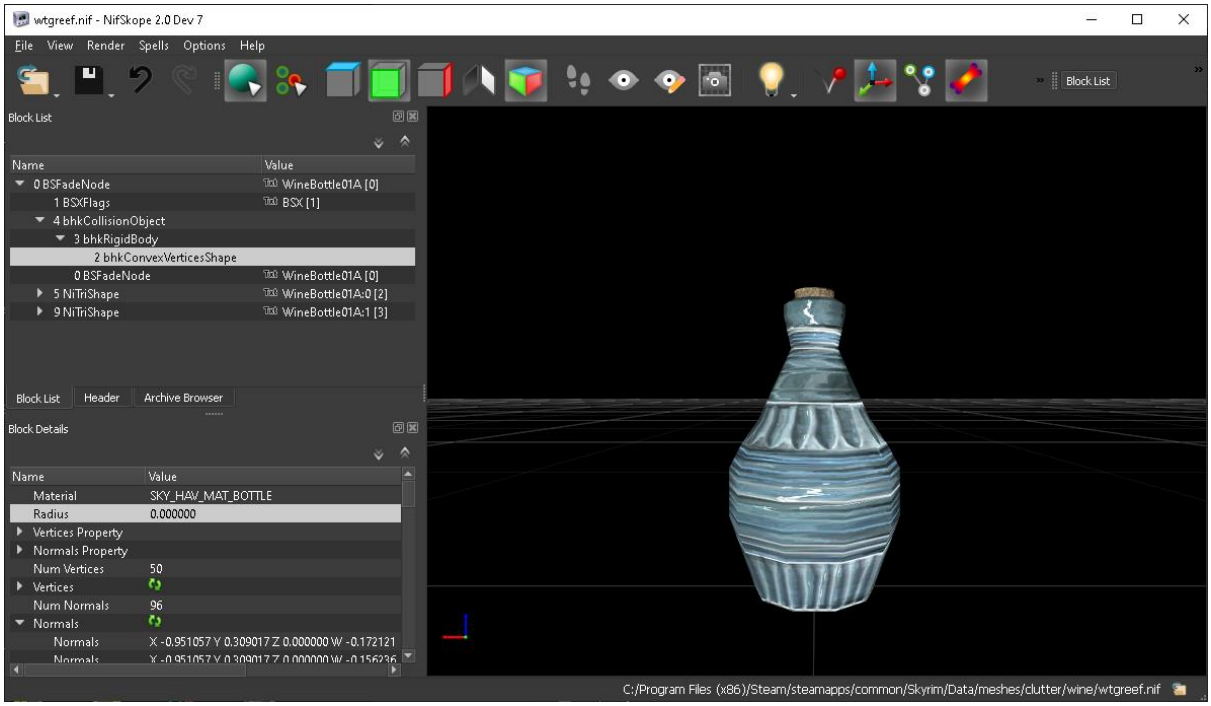


Figure 1205 - Setting the collision Radius to 0.000000.

Save the .nif. And that looks like it fixed the issue. The model is now sitting flush with the surface of the table instead of floating above it.



Figure 1206 - Collision radius fixed.

CHAPTER 9: MOTION-CAPTURE ANIMATION

SETTING UP THE CHARACTER SKELETON IN 3DS MAX

Important: You'll need to use BAE to extract the contents of the Skyrim - Meshes.bsa and Skyrim - Animations.bsa archives. Instruction on how to do so can be found in the section Unpacking the BSA Archives.

For these next few sections, we'll need a copy of Havok Content Tools. Sadly it doesn't seem like there's an official download source anymore following Microsoft's acquisition of Havok. But I can't stop you from Googling around…

I would recommend using 3ds Max 2012 or earlier. Some modders have reported that the 3ds Max 2013/2014-compatible versions of Havok Content Tools have issues exporting animations properly for Skyrim. As far as I'm aware, there are no 3ds Max 2015+ compatible versions of Havok Content Tools.

For 3ds Max 2012 I'd also recommend using figment's branch of the Nif Importer/Exporter plugin as the current version on Nexusmods may raise a ParamBlockDesc2 definition error when 3ds Max launches.

There are many body type frameworks for Skyrim with differing bone setups. In the following examples I'm just going to be working with the base-game skeleton.

Now with all of that out of the way, let's get started.

Firstly, we'll need a copy of the Vanilla Skeleton which you can download from Loverslab. You'll need to be logged in to download this file.

Warning: There's some NSFW content on Loverslab. *Some* might be an understatement…

Open 3ds Max 2010_Skyrim_Vanilla Skeleton.max.

If you see a Gamma & LUT Settings Mismatch warning, select 'Adopt the File's Gamma and LUT Settings' then click OK.

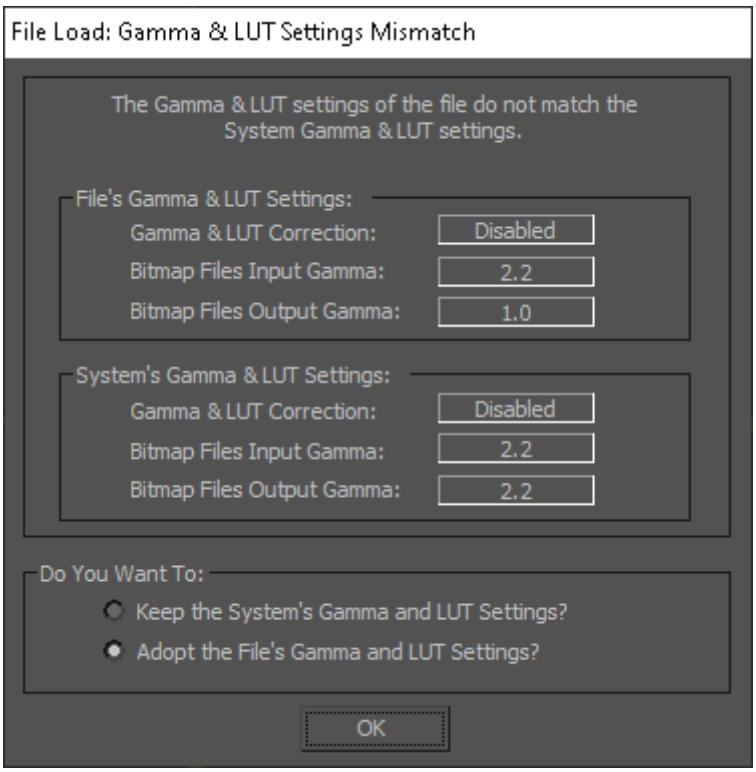


Figure 1207 - Gamma & LUT Settings Mismatch.

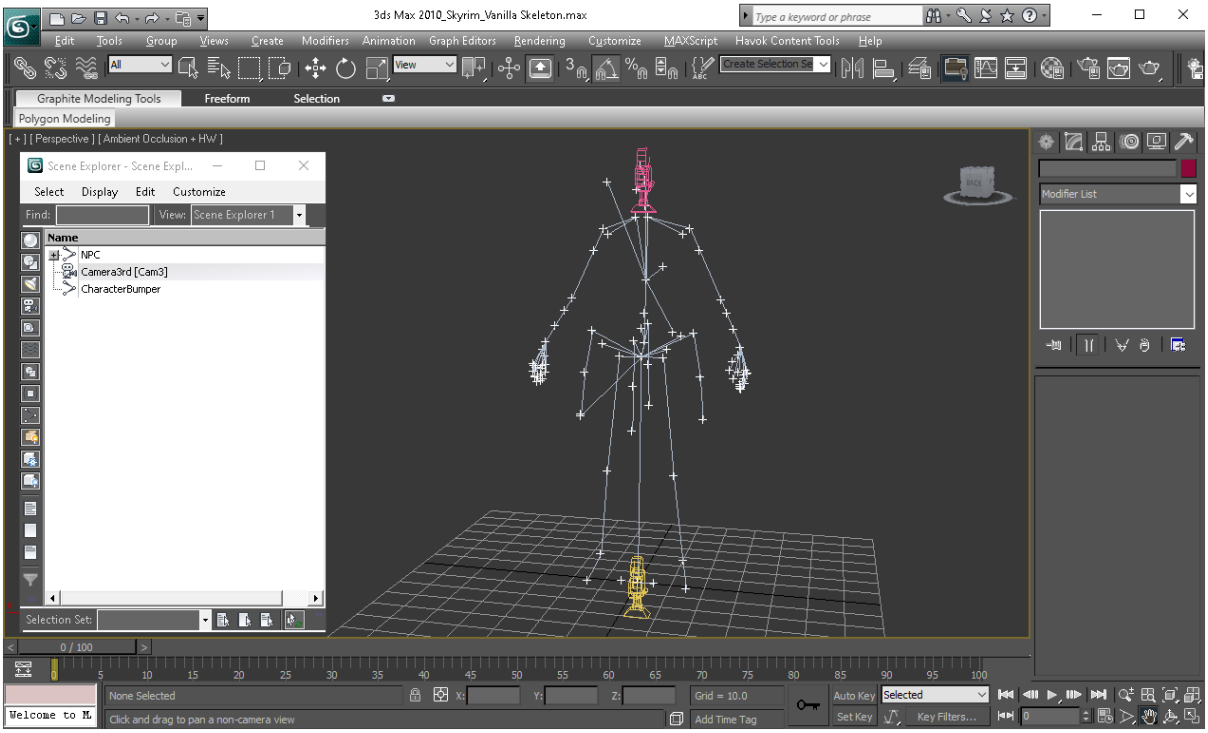


Figure 1208 - Vanilla skeleton loaded into 3ds Max.

Next, let's import a body mesh.

Go back to Import.

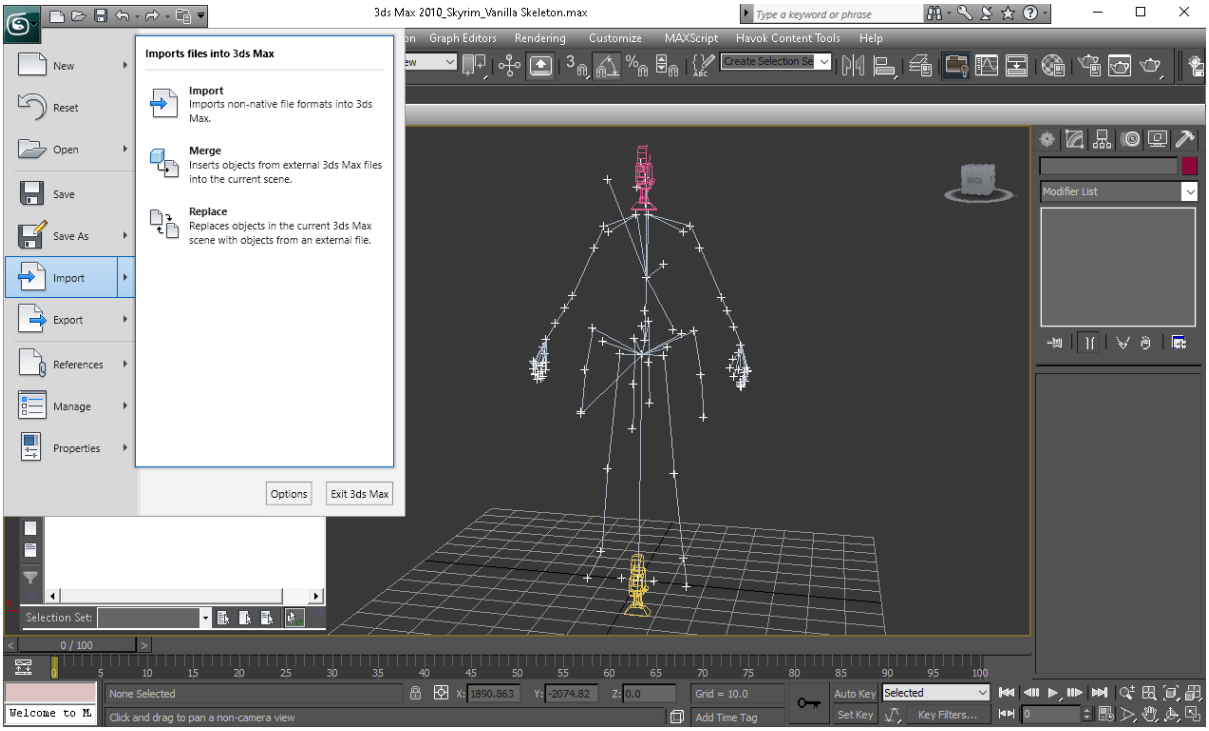


Figure 1209 - Importing a new asset.

This time, select malebody_1.nif then click Open.

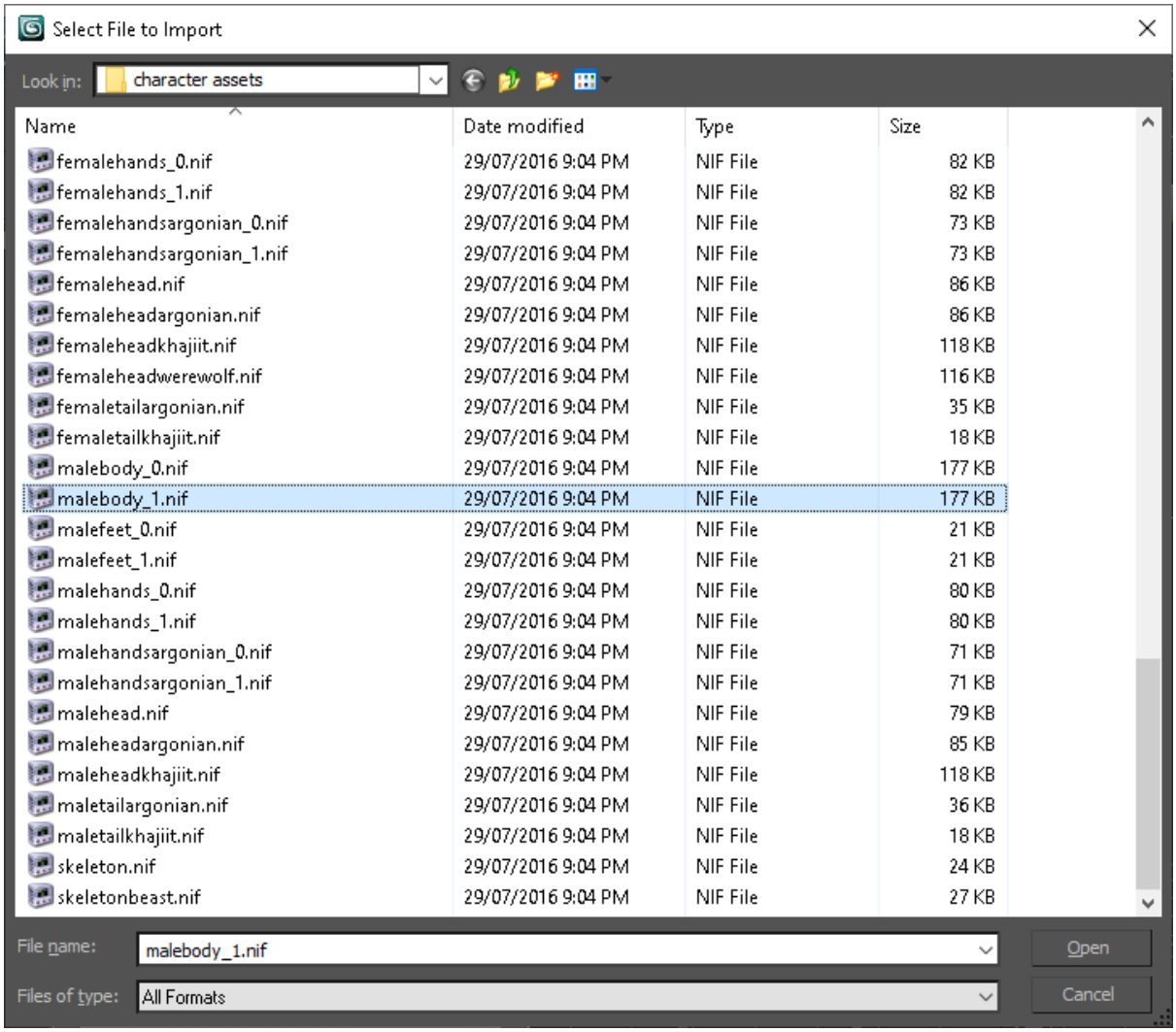


Figure 1210 - Importing malebody_1.nif.

Tick Vertex Colours, Skin Modifier, Auto Smooth Mesh, Flip UV, Render Textures in View, Ignore Root Node and Use Niftools Shader are ticked then click Import.

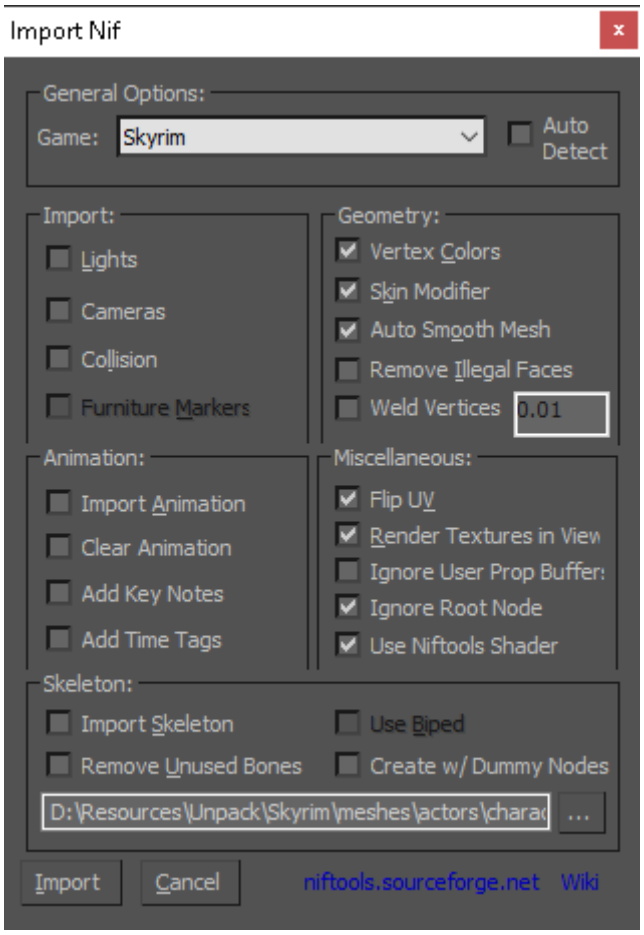


Figure 1211 - Import settings.

You should now see the torso and arms and legs covering the skeleton.

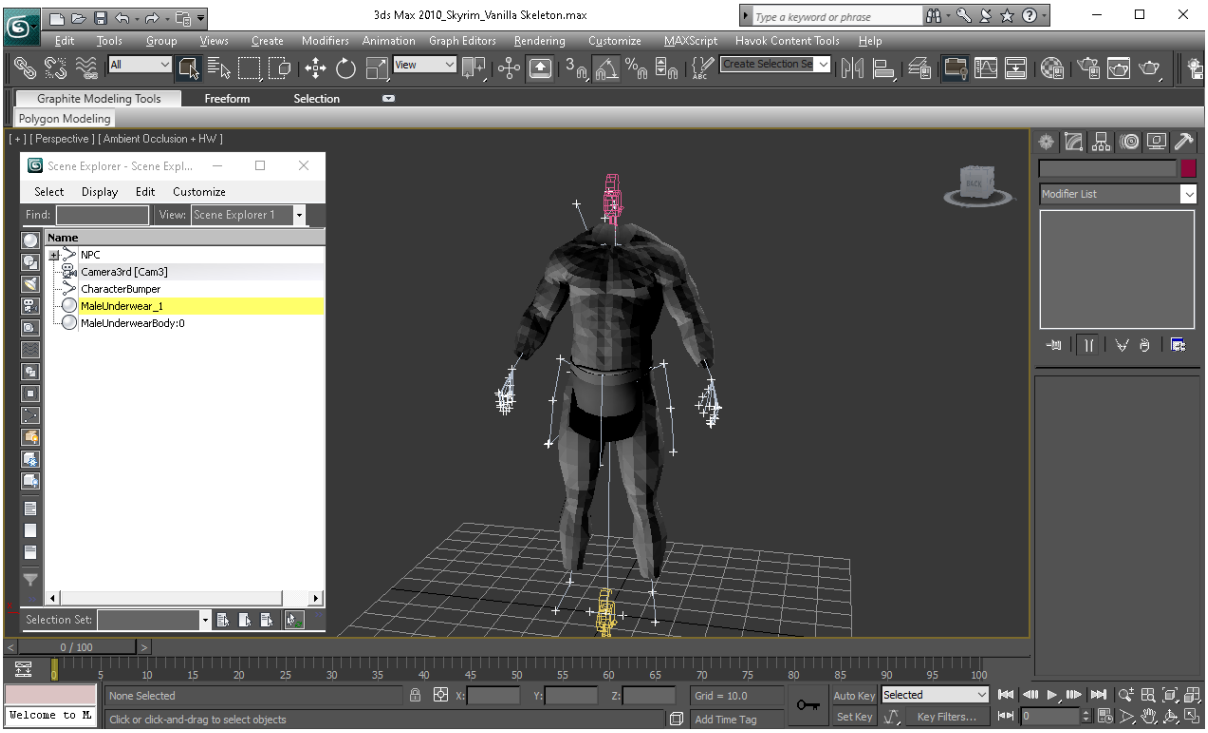


Figure 1212 - Imported malebody_1.nif.

Repeat these steps to import malefeet_1.nif, malehands_1.nif, and malehead.nif.

We should now have a full body as per the screenshot below:

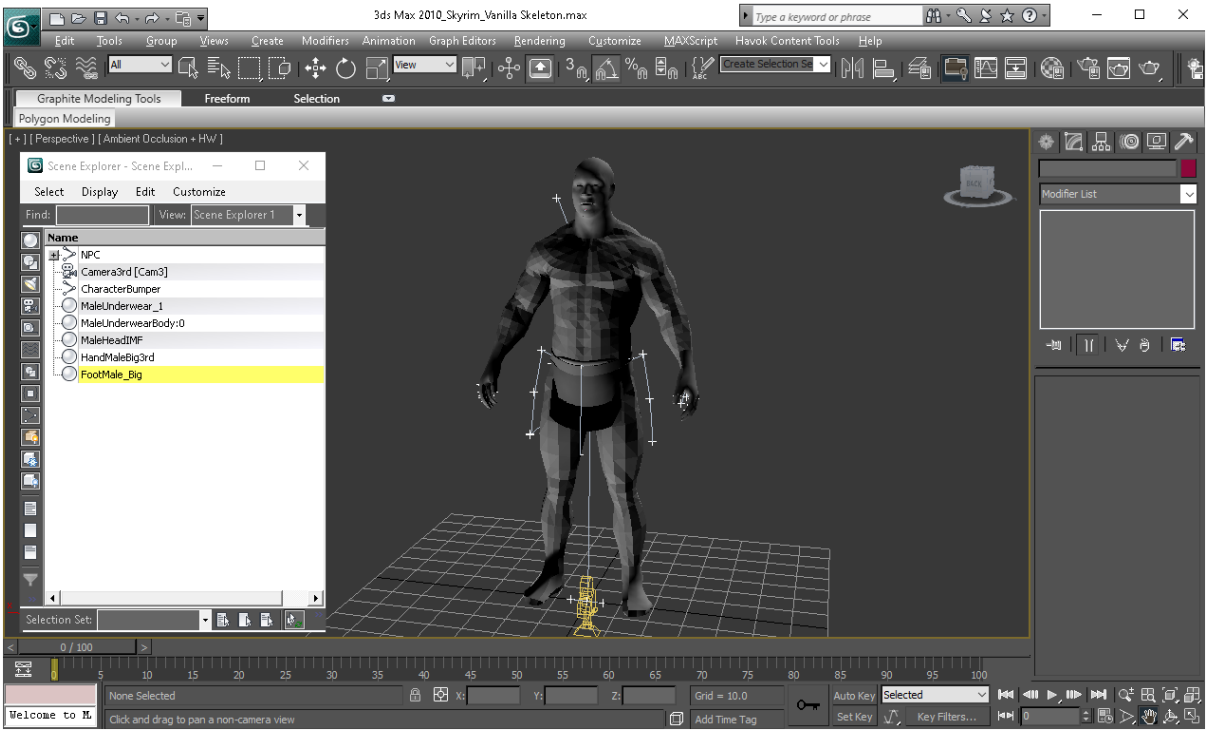


Figure 1213 - The rest of the body parts imported.

Next, let's test importing an animation to ensure everything works as it should.

In order to do this we'll need a copy of hkxcmd which can be downloaded from Nexusmods.

Extract hkxcmd to a new folder.

Browse to that folder and create two new subfolders, one called Input and the other called Output.

Copy the skeleton.hkx file from your skyrim\meshes\actors\character\character assets directory.

Note: You'll need to extract the assets from the Skyrim - Animations.bsa archive first.

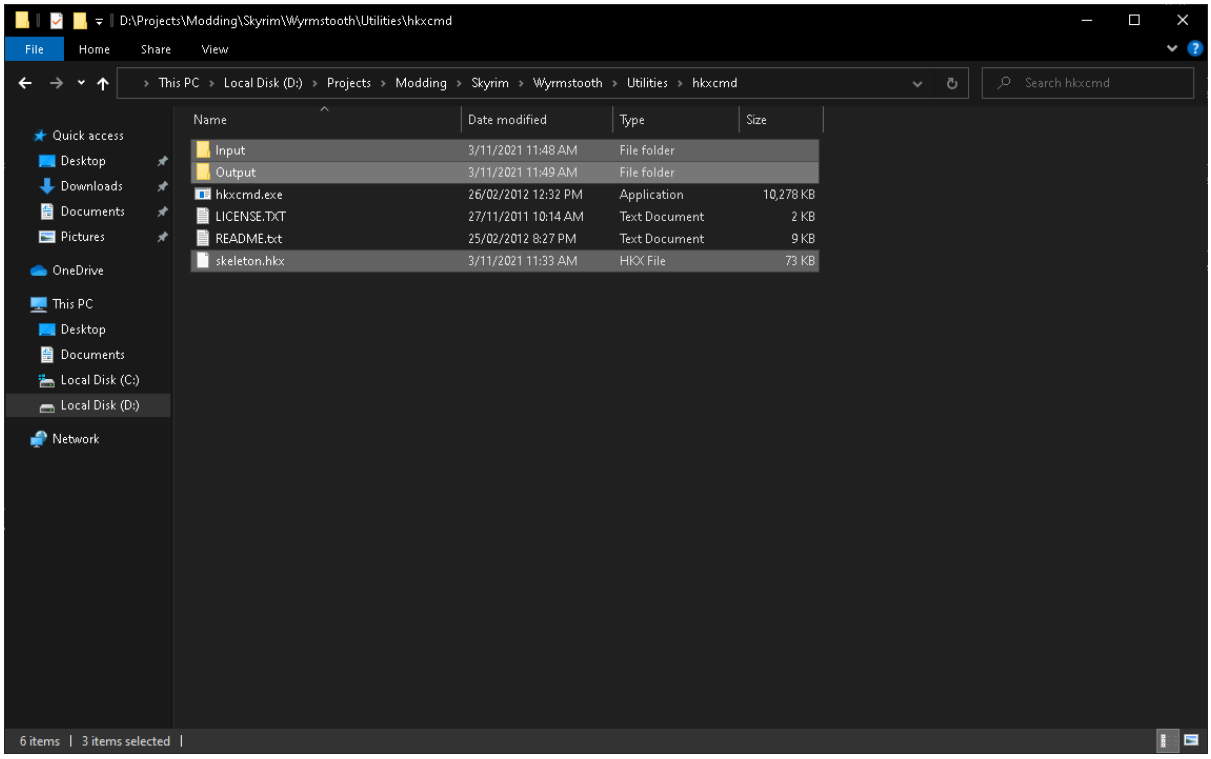


Figure 1214 - Setting up the hkxcmd folder.

Next, browse to skyrim\meshes\actors\character\animations.

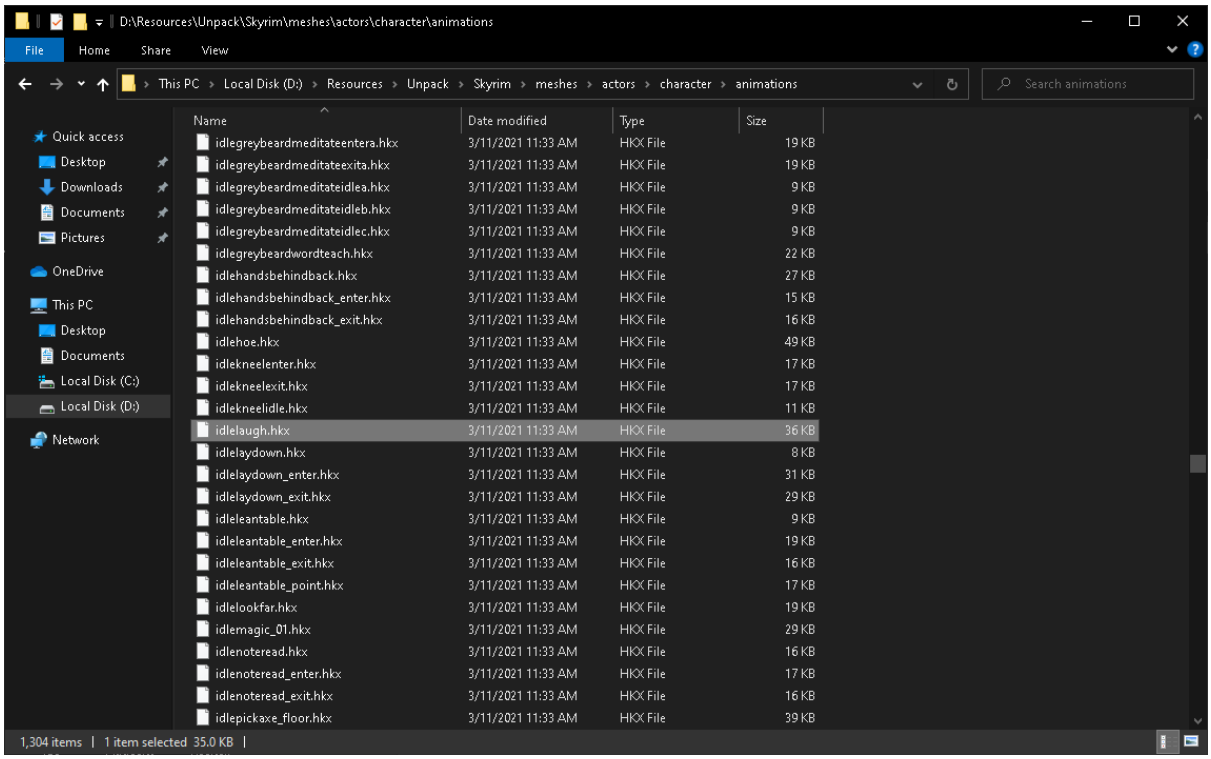


Figure 1215 - List of idle animations.

Copy one of the .hkx files to your hkxcmd\Input folder.

For this example, I just copied idlelaugh.hkx.

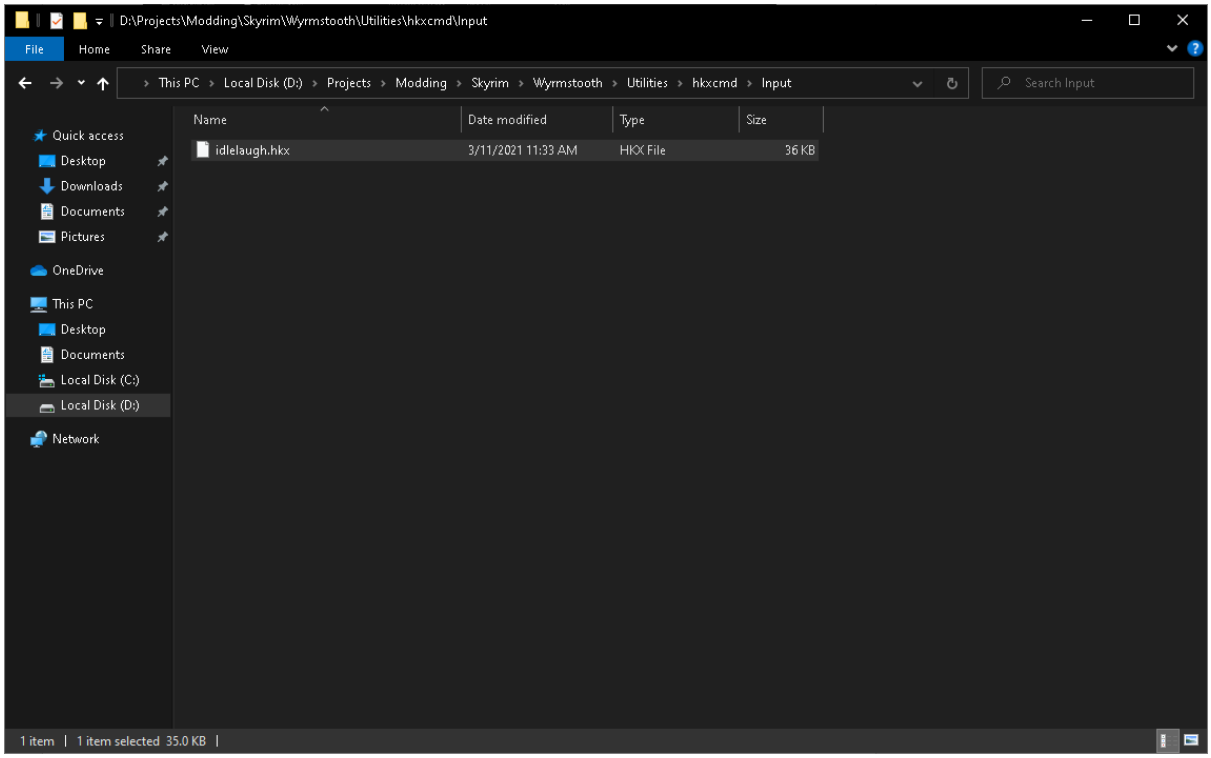


Figure 1216 - idlelaugh.hkx copied to the input folder.

Now open a command prompt and navigate it to your hkxcmd folder.

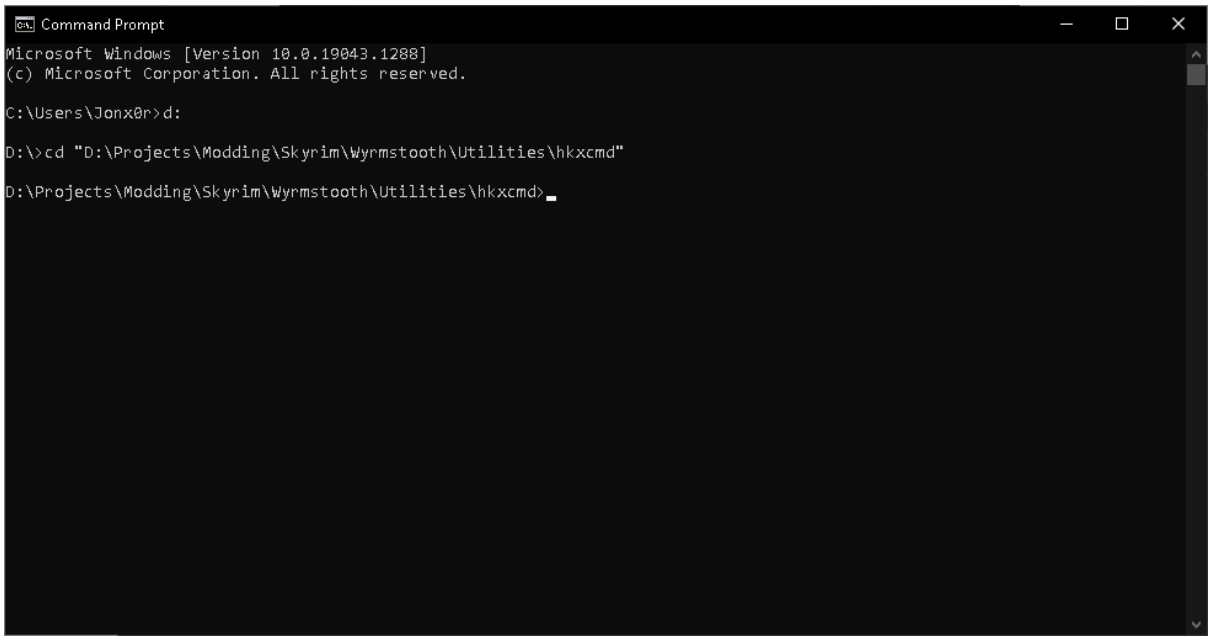


Figure 1217 - Command prompt pointed to the hkxcmd folder.

Run the following command:

```
hkxcmd exportkf "skeleton.hkx" "Input" "Output"
```

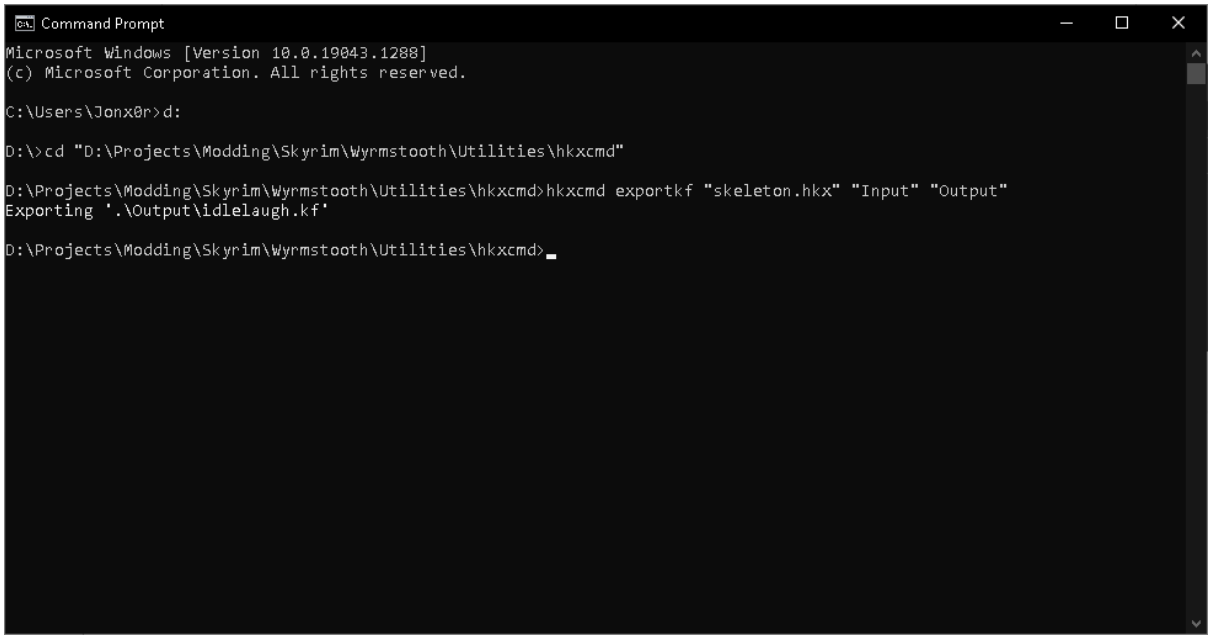


Figure 1218 - Converted the .hkx idle animation file to .kf.

This utility will convert all .hkx files in the Input folder to .kf files which are placed in the Output folder.

Confirm the animation file(s) were successfully converted to .kf.

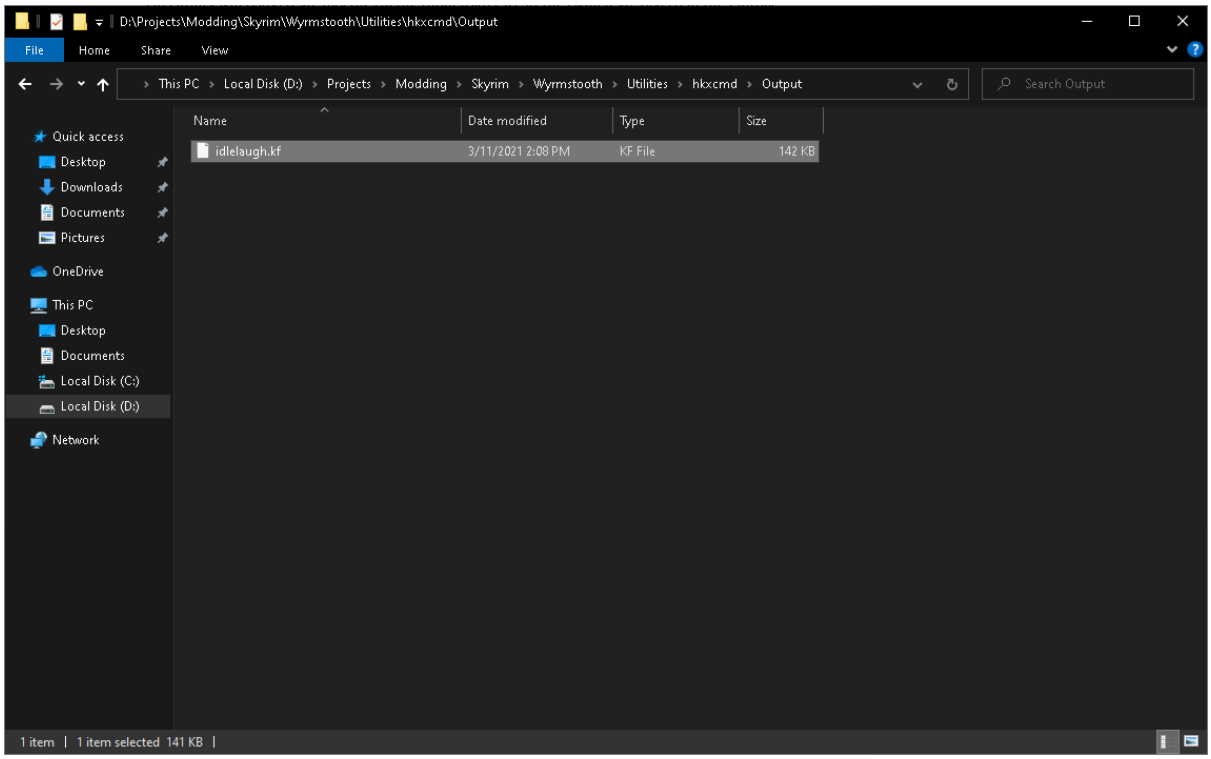


Figure 1219 - idlelaugh converted to .kf format.

Back in 3ds Max, go to Import.

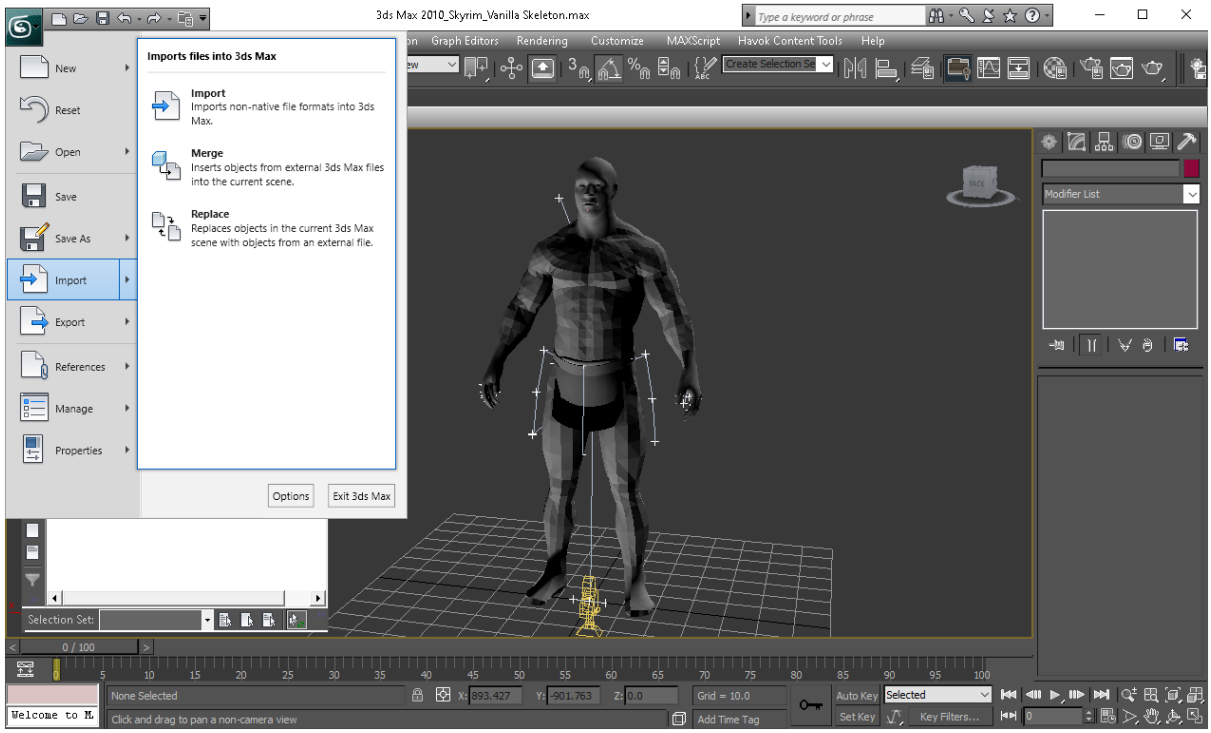


Figure 1220 - Importing a new asset.

Browse to the hkxcmd\Output folder, select the .kf file you converted, then click Open.

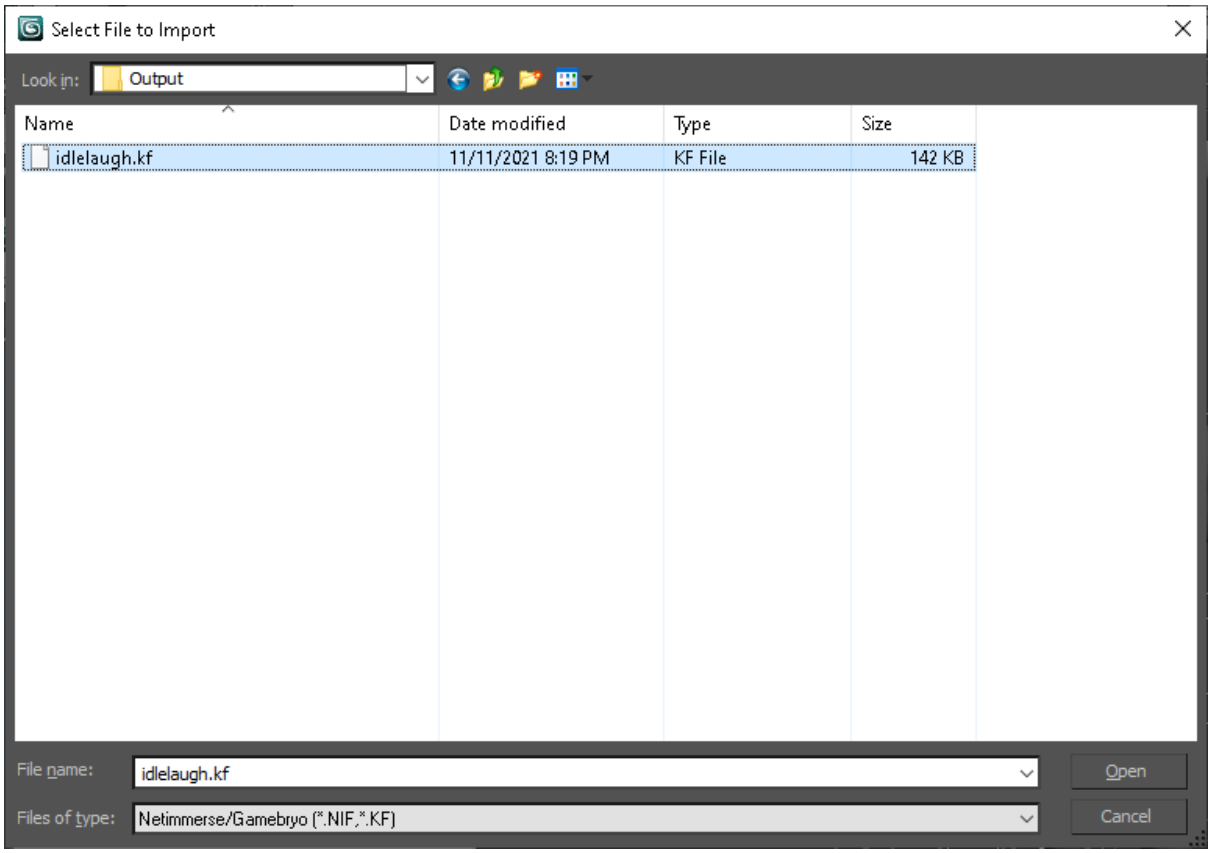


Figure 1221 - Importing idlelaugh.kf.

No need to tick anything, just click Import.

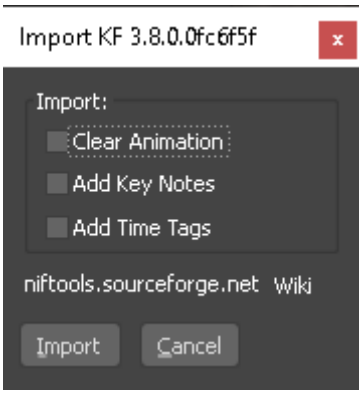


Figure 1222 - .kf import pop-up.

Scrub the timeline and confirm the animation plays as expected.

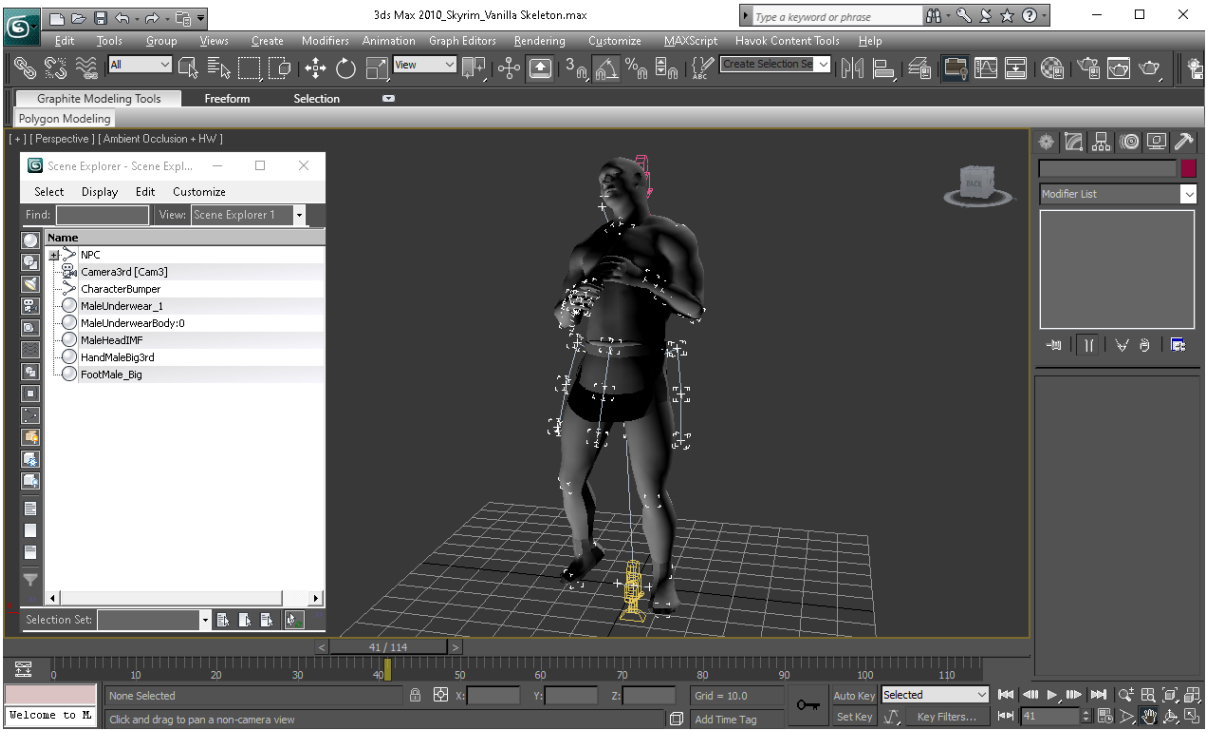


Figure 1223 - Animation plays as expected.

Both the bones and the character mesh should move as you scrub the timeline.

CONSTRAINING THE SKELETON TO A BIPED AND IMPORTING MOCAP DATA

Now that we have a skeleton imported into 3ds Max, let's try and import some mocap data.

These next few sections can be a bit complicated, so I've tried to keep each step as granular as possible.

For this example, I'm going to use the mocap data recorded by Carnegie Mellon University which is available on their website.

In order to use this data in 3ds Max to animate our skeleton, we'll first need to create a biped object and constrain the bones of our skeleton to the biped object. So basically we'll be importing the mocap animation onto the biped object and as it moves, it'll also move our skeleton.

First, let's make the character mesh transparent. Hold down CTRL and left click on the head, torso, hands, loincloth and feet to select them all then press ALT + X to make them transparent.

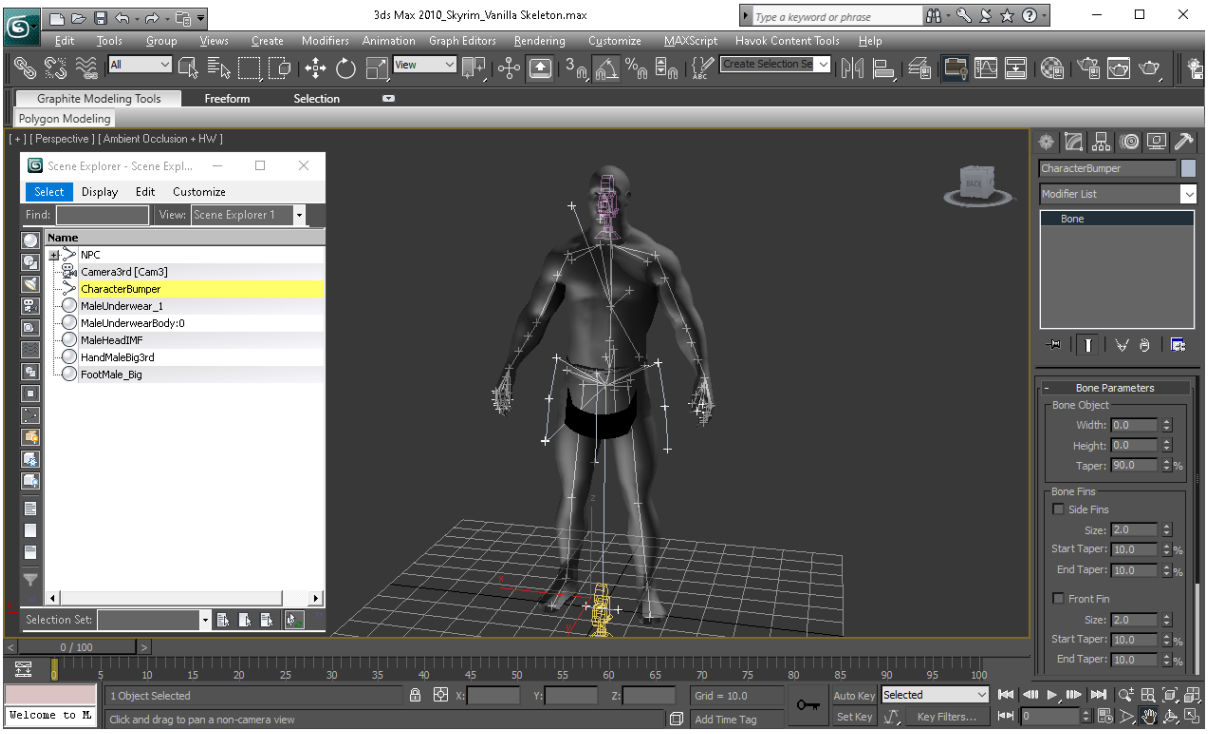


Figure 1224 - Making the character mesh transparent.

With the character mesh still selected, right click and select Freeze Selection.

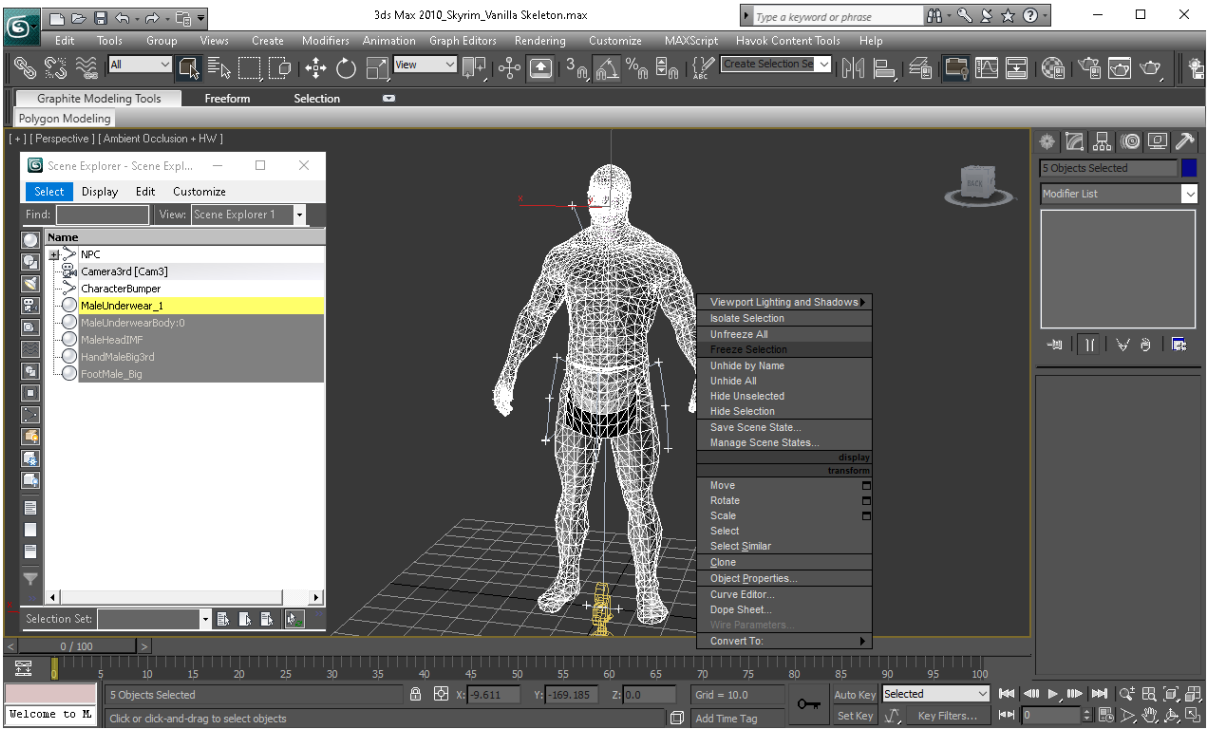


Figure 1225 - Freezing the character mesh.

This will prevent us from accidentally selecting the character mesh when working with the bones in the upcoming steps.

Go to the Create tab and go to Systems.

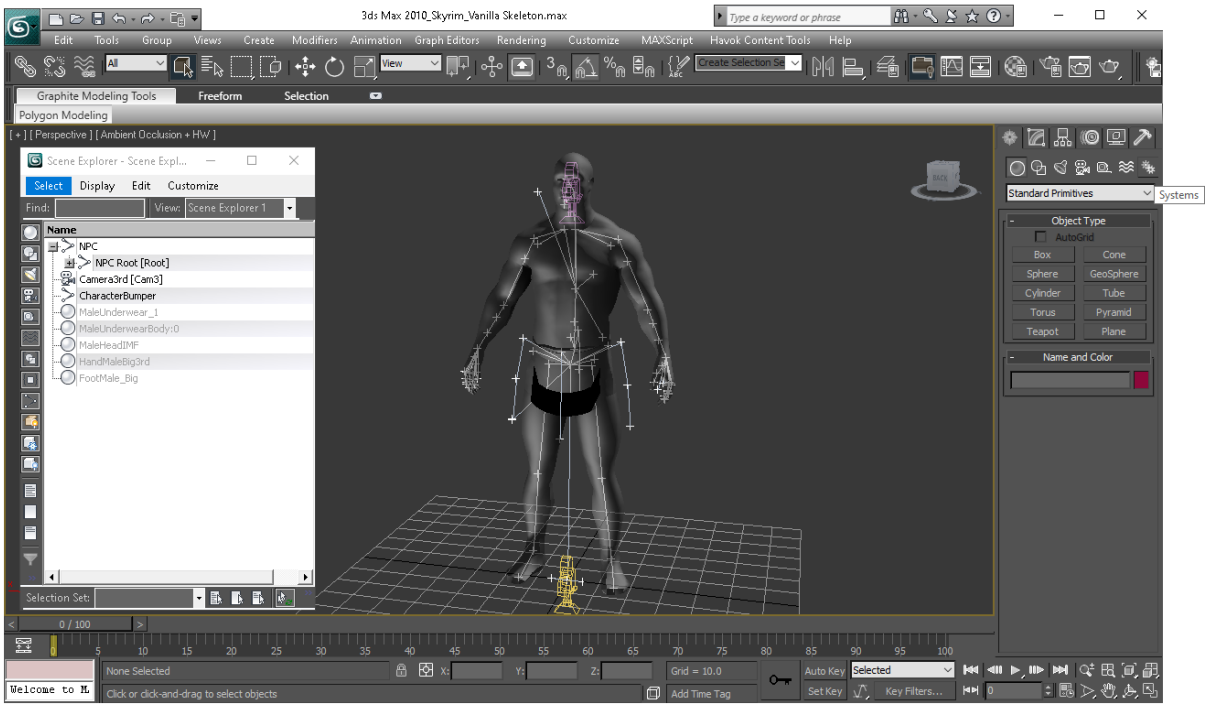


Figure 1226 - The Create tab.

Press F to go to the front view then click on Biped.

Set Fingers to 5 and set Finger Links to 3. Ensure Spine Links is set to 4.

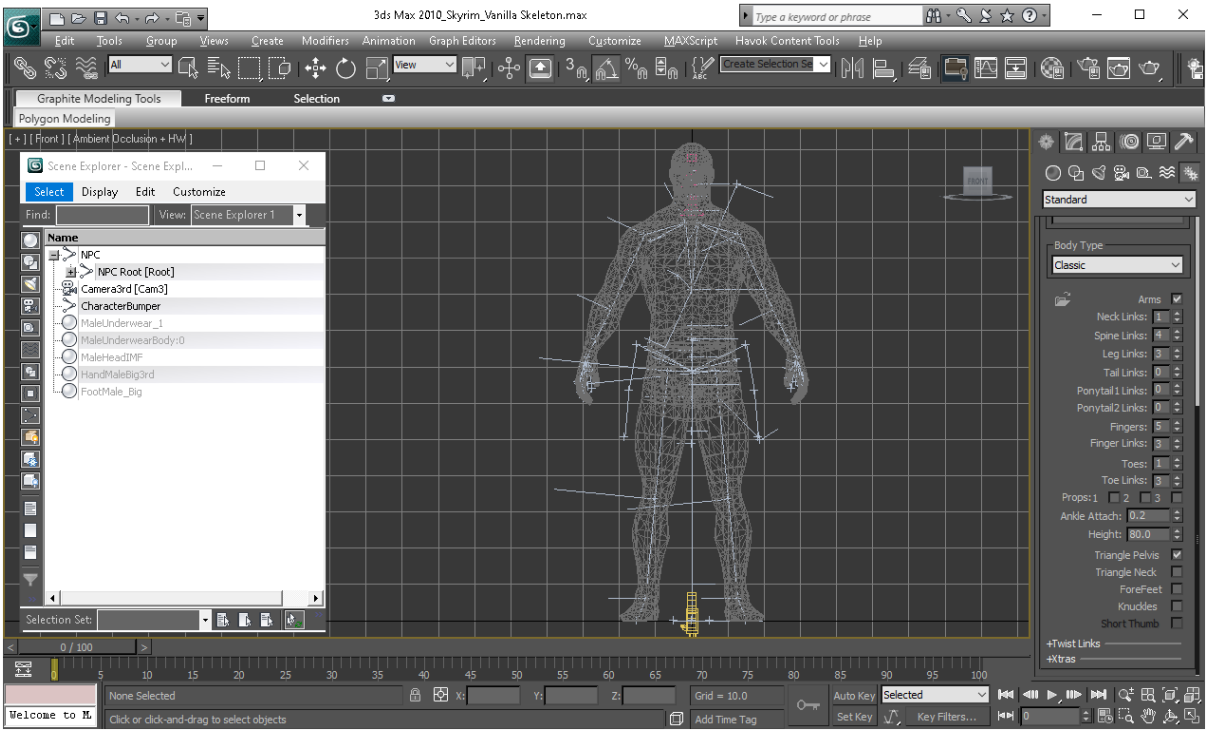


Figure 1227 - Biped settings.

Click and drag in the viewport starting from the bottom. Make it about the same height as the skeleton. This new biped should be named Bip01 by default.

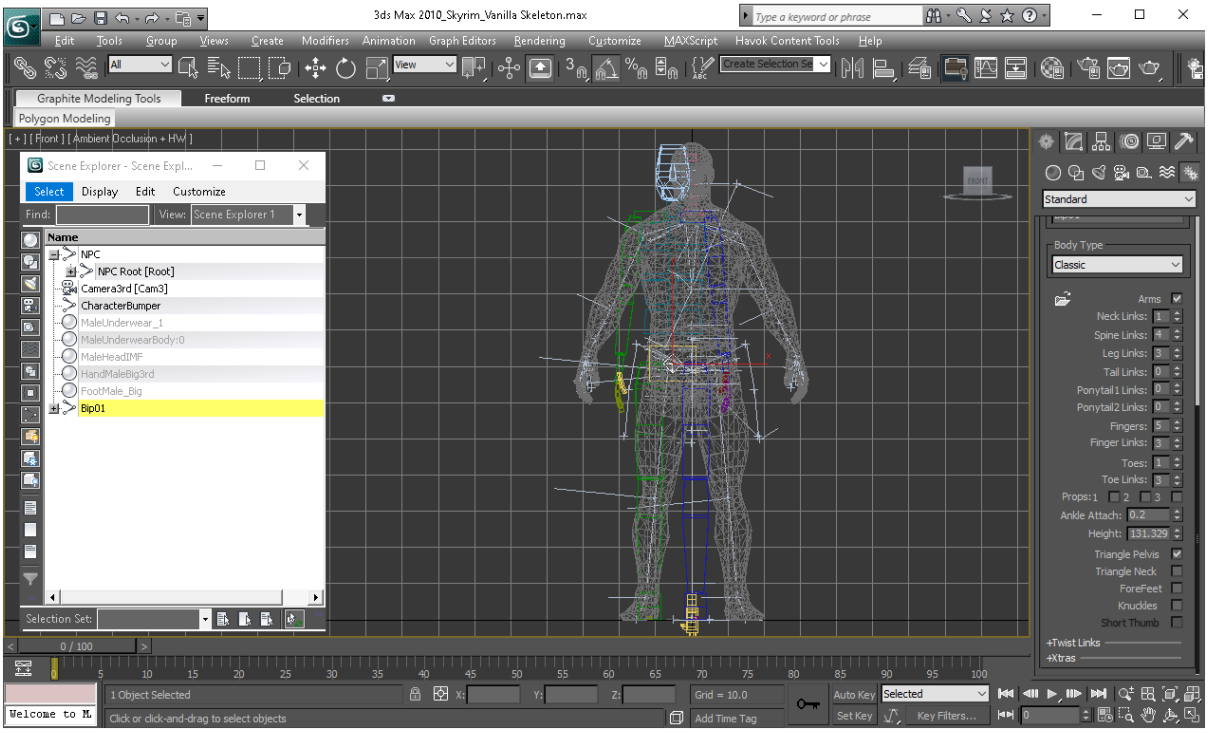


Figure 1228 - Adding a biped.

With Bip01 selected, go to the Motion tab. Scroll down to the Biped section and click on the Figure Mode button.

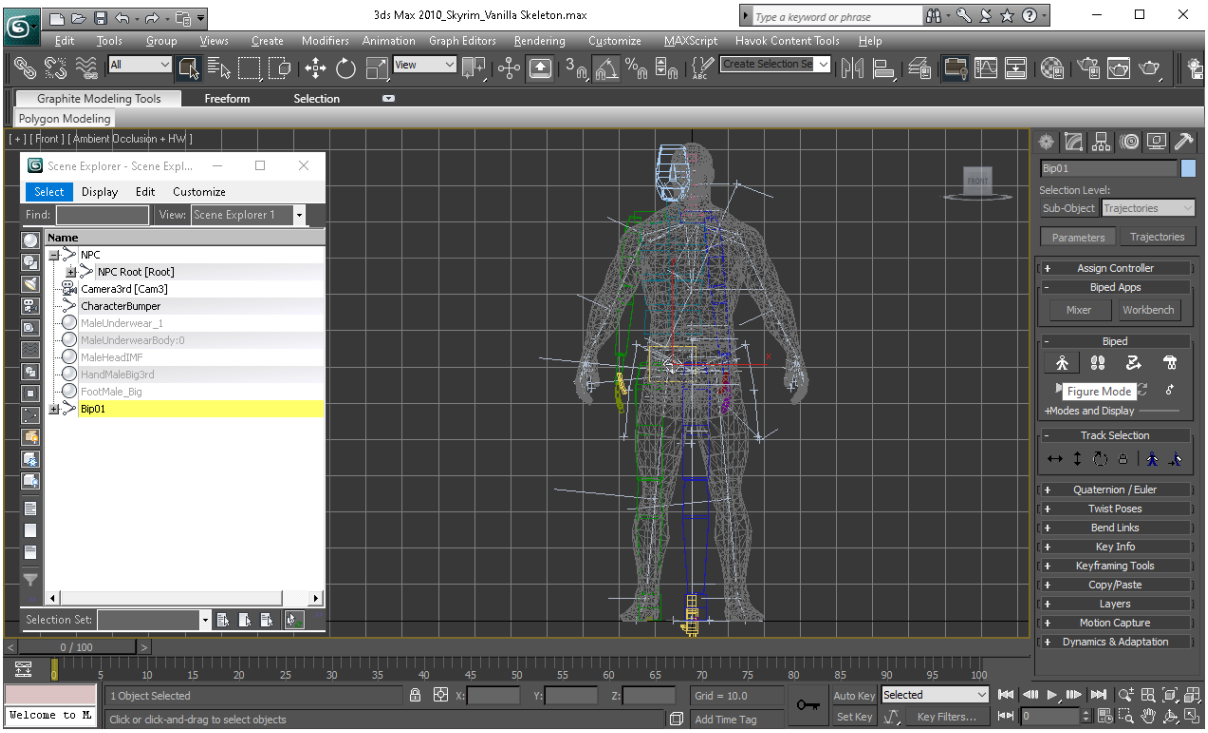


Figure 1229 - Figure mode.

In the object list, click on just Bip01 then right-click on the Move tool in the toolbar.

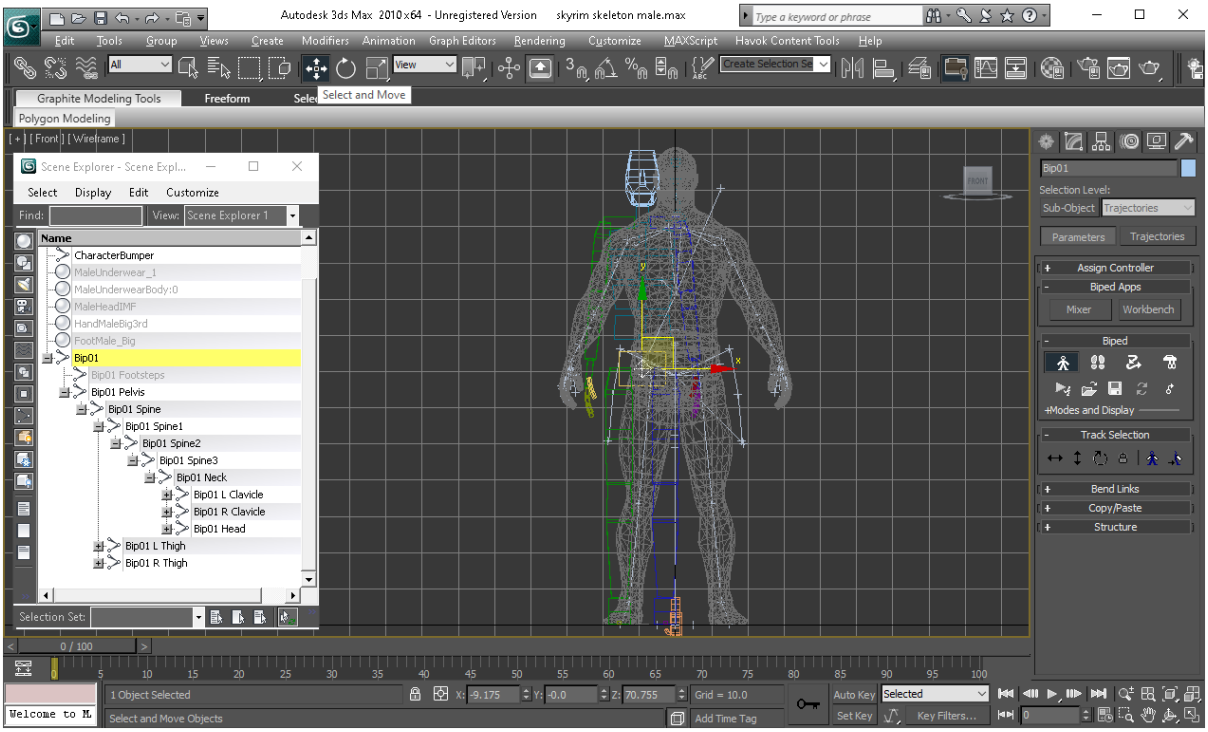


Figure 1230 - Selecting the Bip01 parent object.

Set X and Y to 0.0 then close the move tool properties.

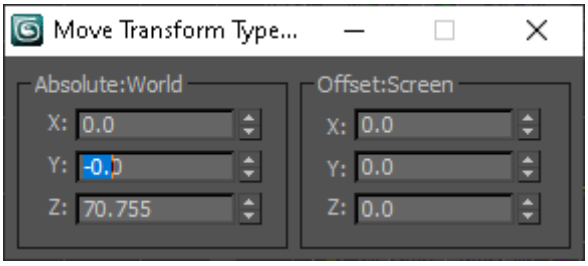


Figure 1231 - Centering Bip01 to the world.

If you press P to switch the viewport back to perspective, you may notice that the biped is facing the wrong way.

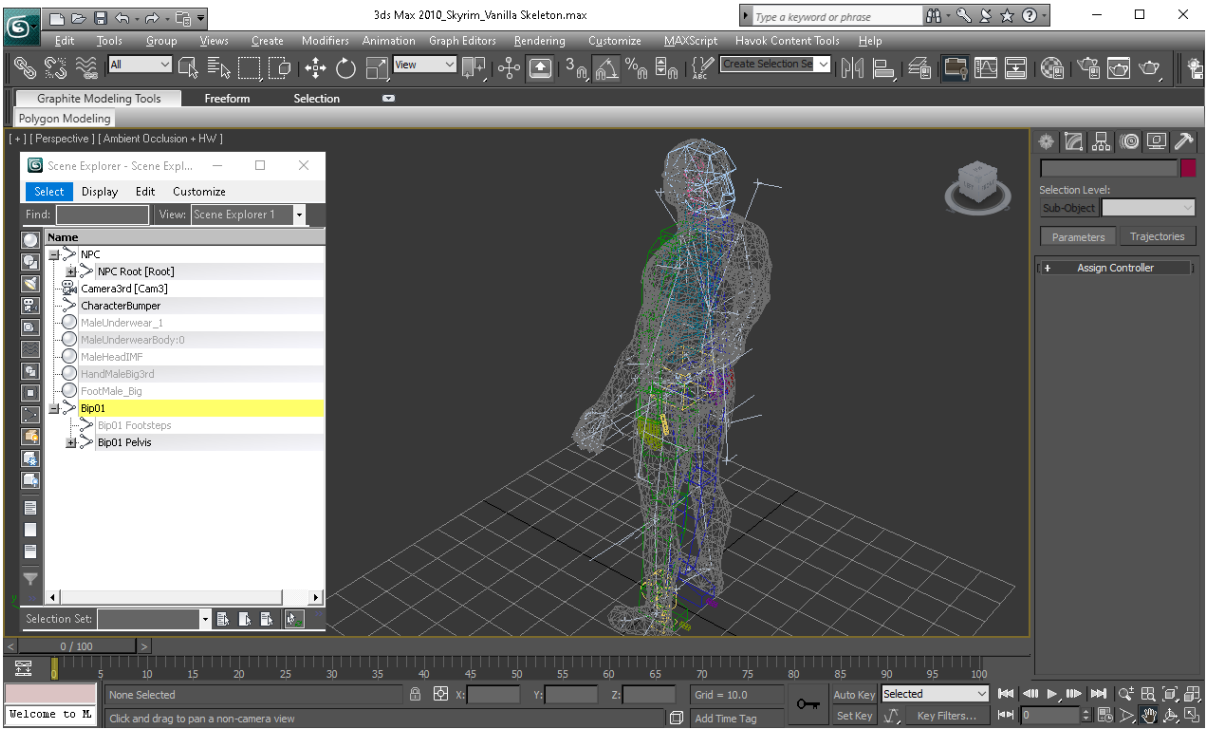


Figure 1232 - Bip01 facing the wrong direction.

To fix this, with the Bip01 parent object still selected, right-click on the Select and Rotate tool.

Set Z to 90 then close the Select and Rotate tool.

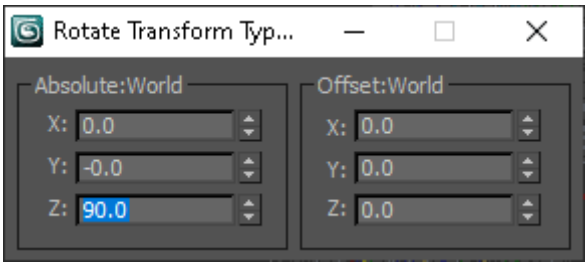


Figure 1233 - Rotating Bip01 to face the right way.

Select the bottom spine segment and move it down until the clavicle is roughly in line with the clavicle of our custom skeleton.

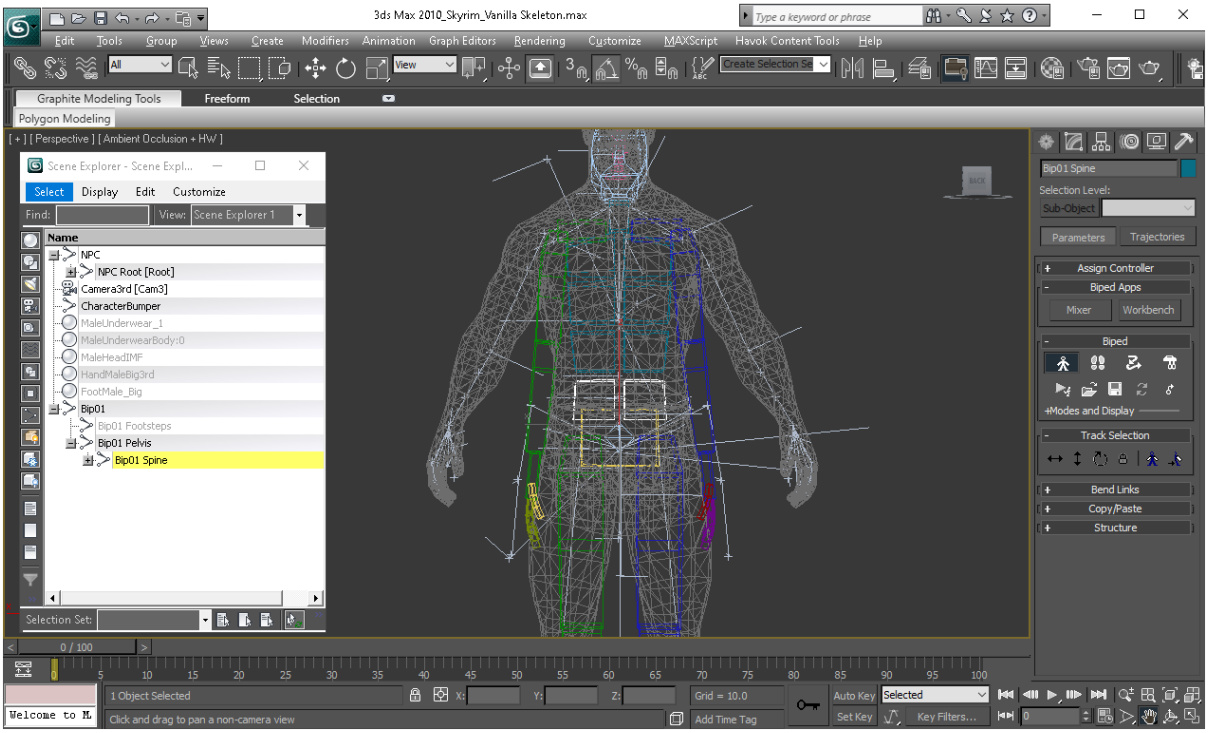


Figure 1234 - Lowering the bottom spine segment.

Click on Angle Snap Toggle to turn it on if it isn't already.

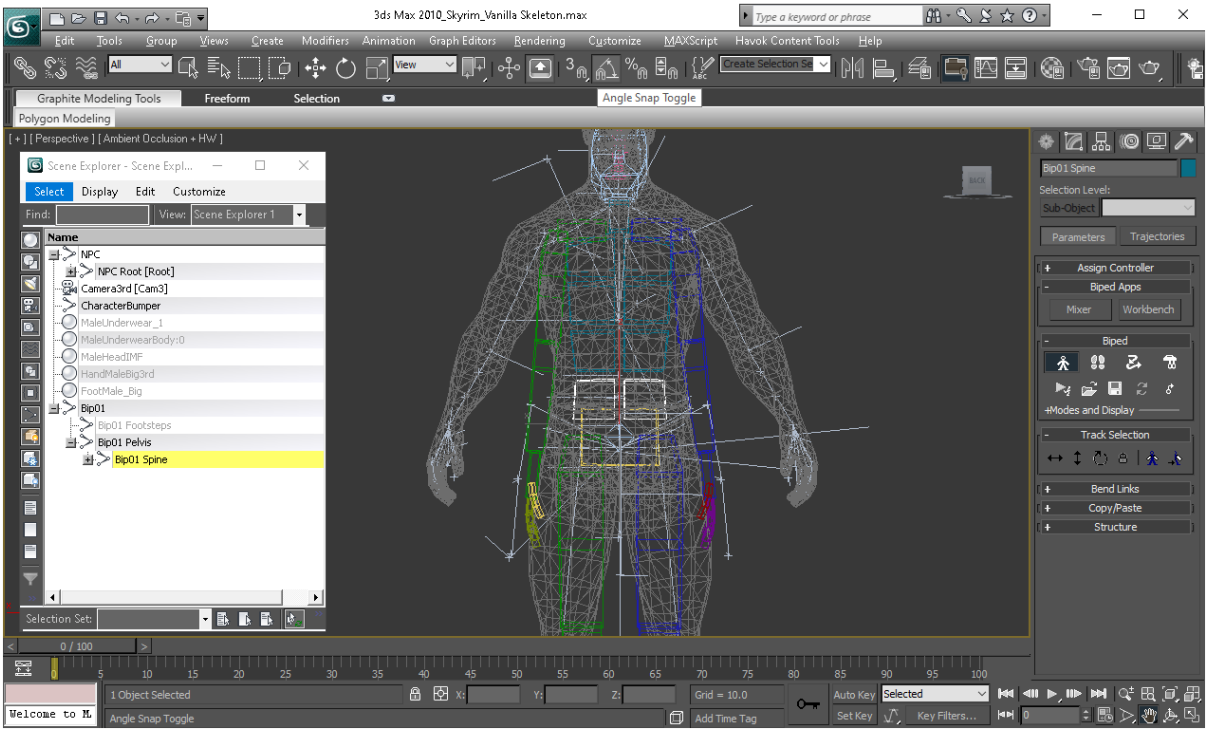


Figure 1235 - Selecting the upper arms on Bip01.

Click on the Select and Rotate tool and rotate Bip01's right upper arm 20 degrees.

It shouldn't matter if it doesn't perfectly match the bones in our skeleton.

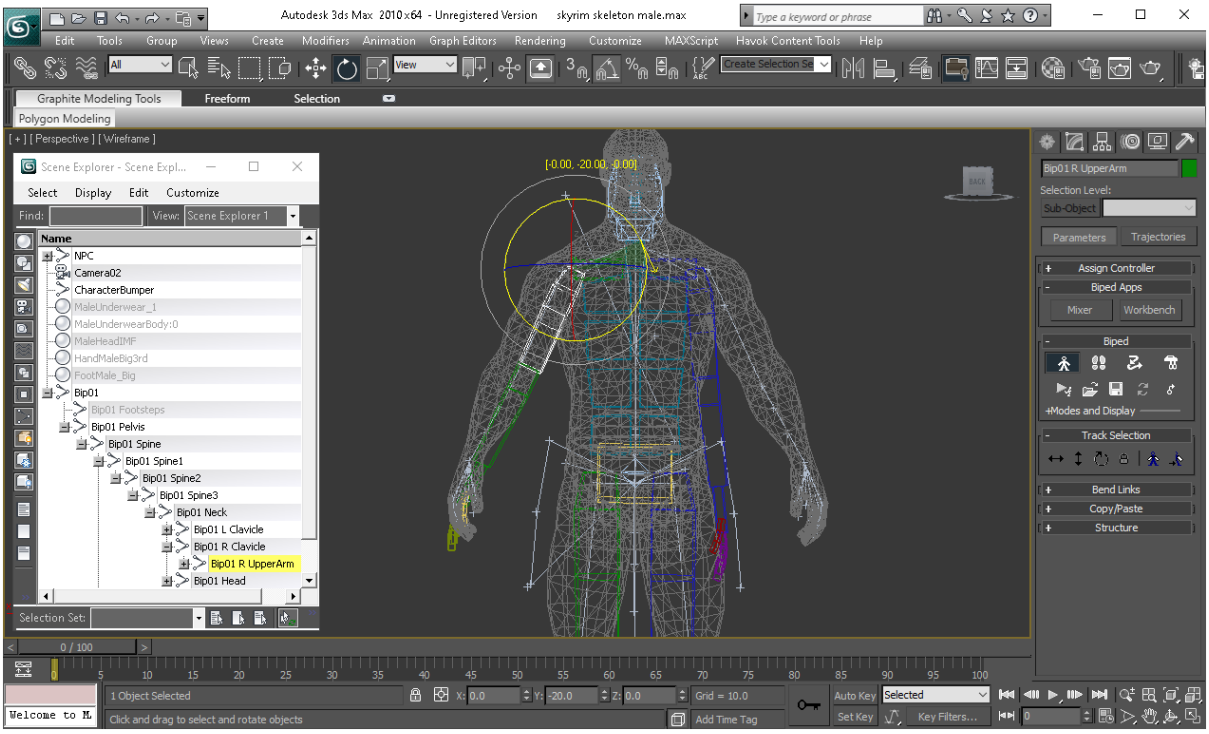


Figure 1236 - Moving the right-arm into place.

Do the same for Bip01's left arm.

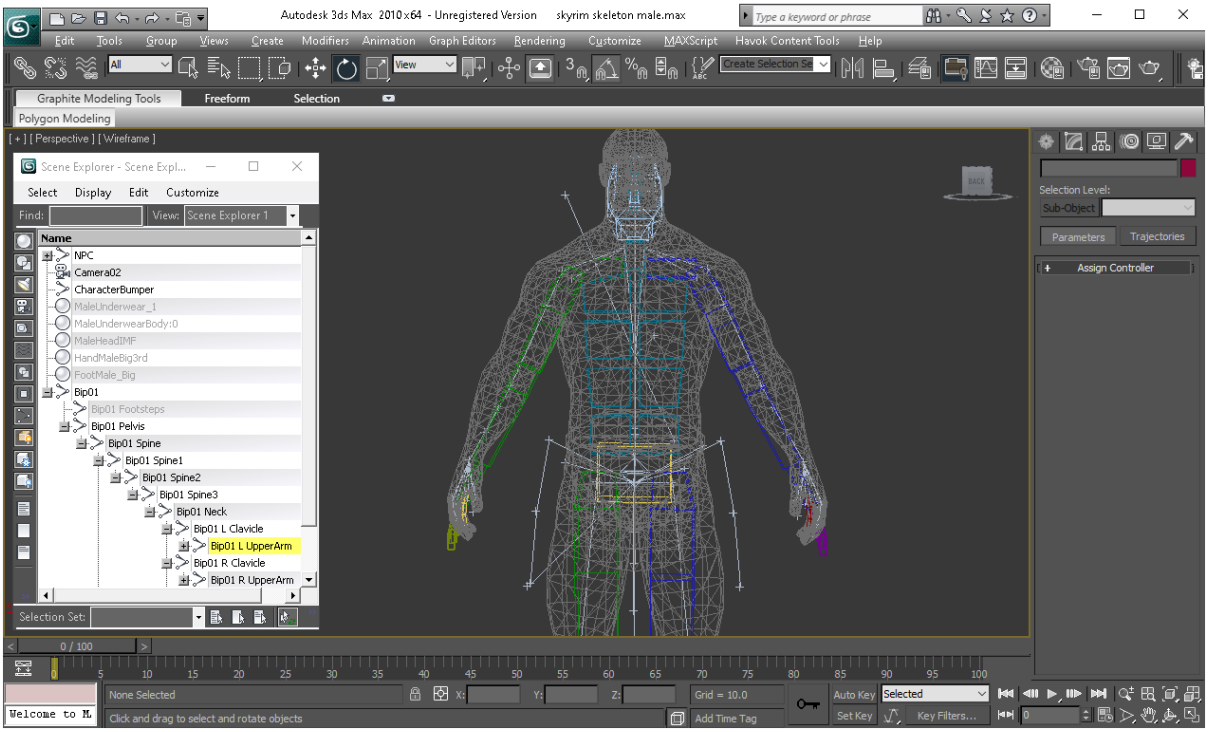


Figure 1237 - Both arms rotated into place.

Select both hands on Bip01 and move them into roughly the same spot as the hands on our skeleton.

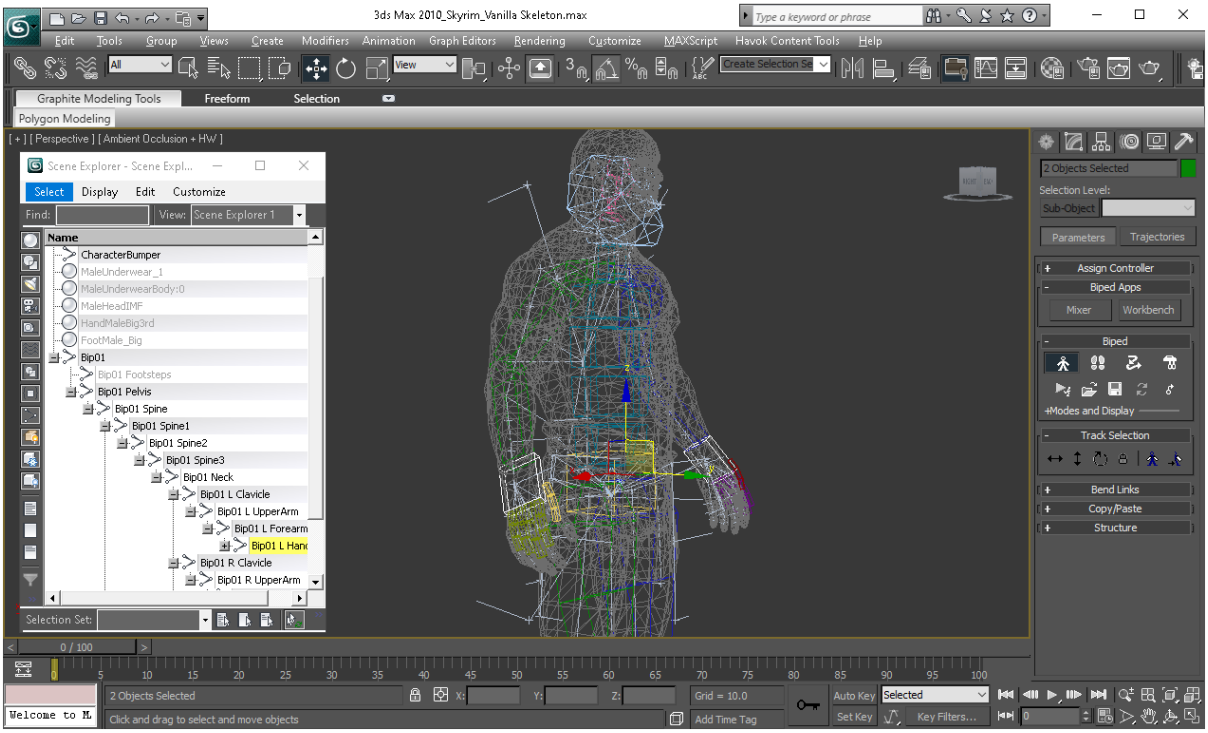


Figure 1238 - Moving both hands into position.

Move Bip01's right leg 5 degrees.

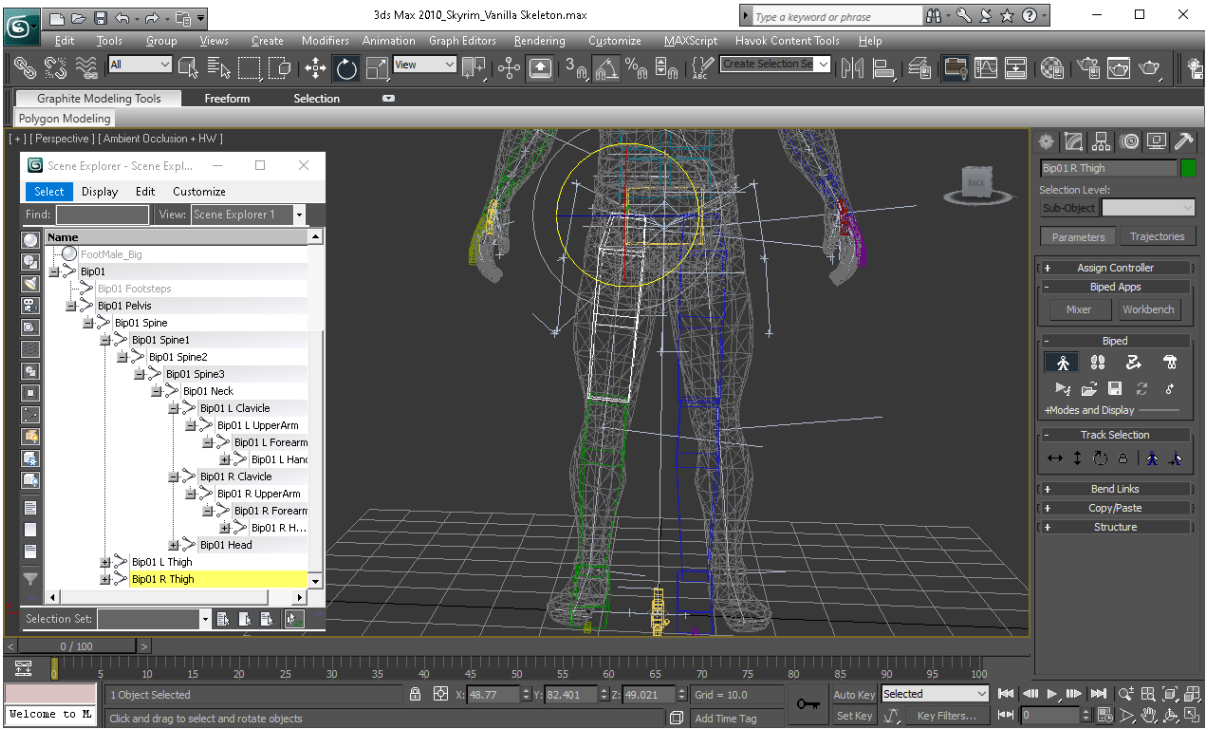


Figure 1239 - Moving the right leg into position.

Do the same for the left leg.

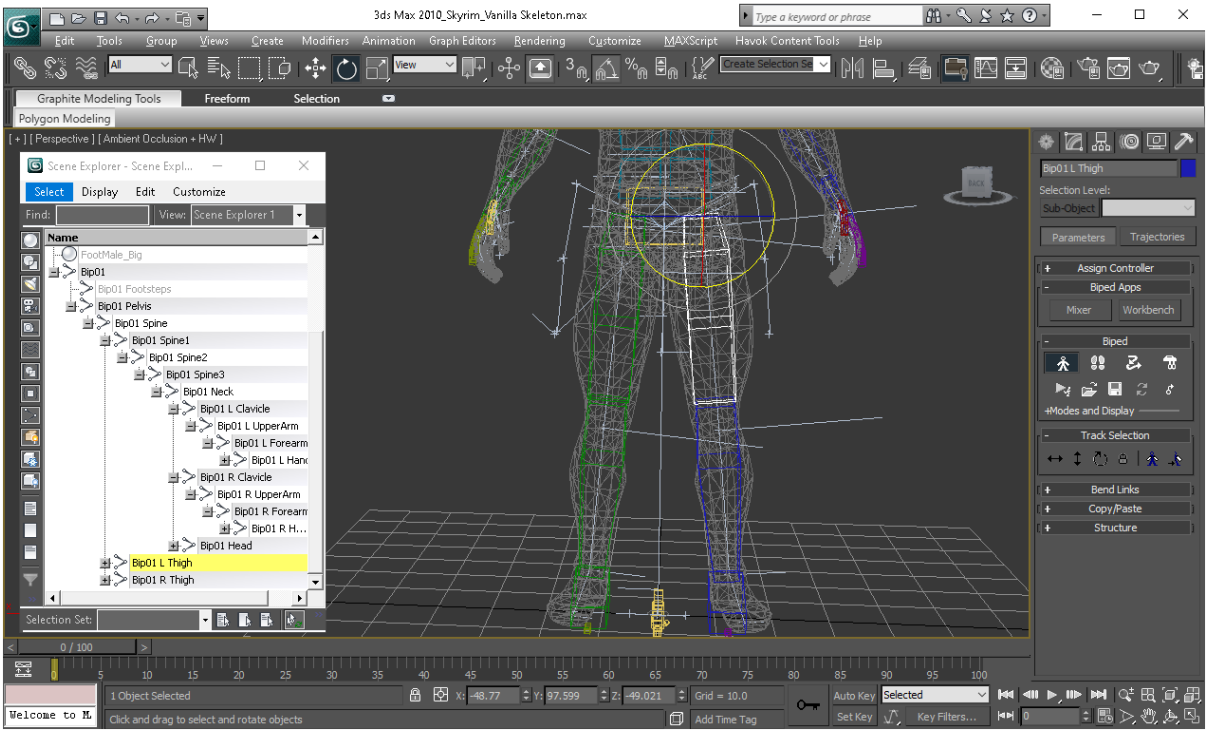


Figure 1240 - Right-leg moved into place.

Move both feet back so that they're roughly in the same place as the feet in our custom skeleton.

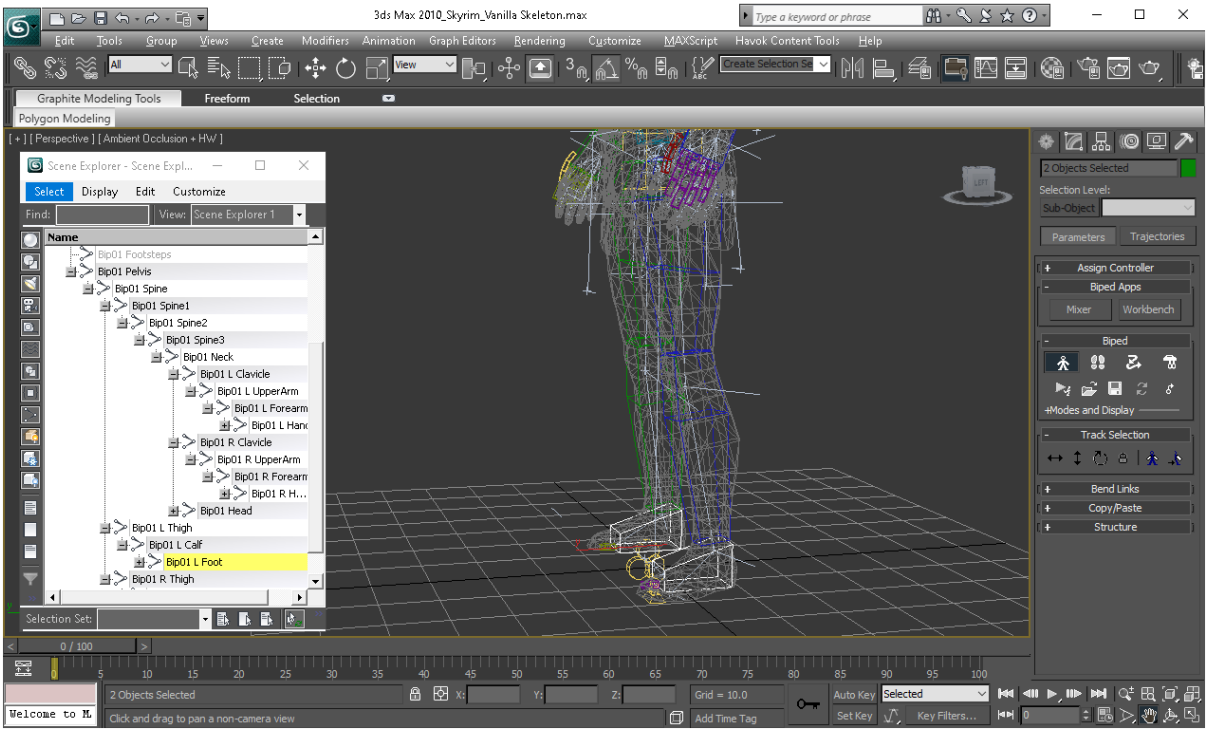


Figure 1241 - Moving the feet into place.

Rotate the feet by about 15 degrees so they match the feet of our custom skeleton.

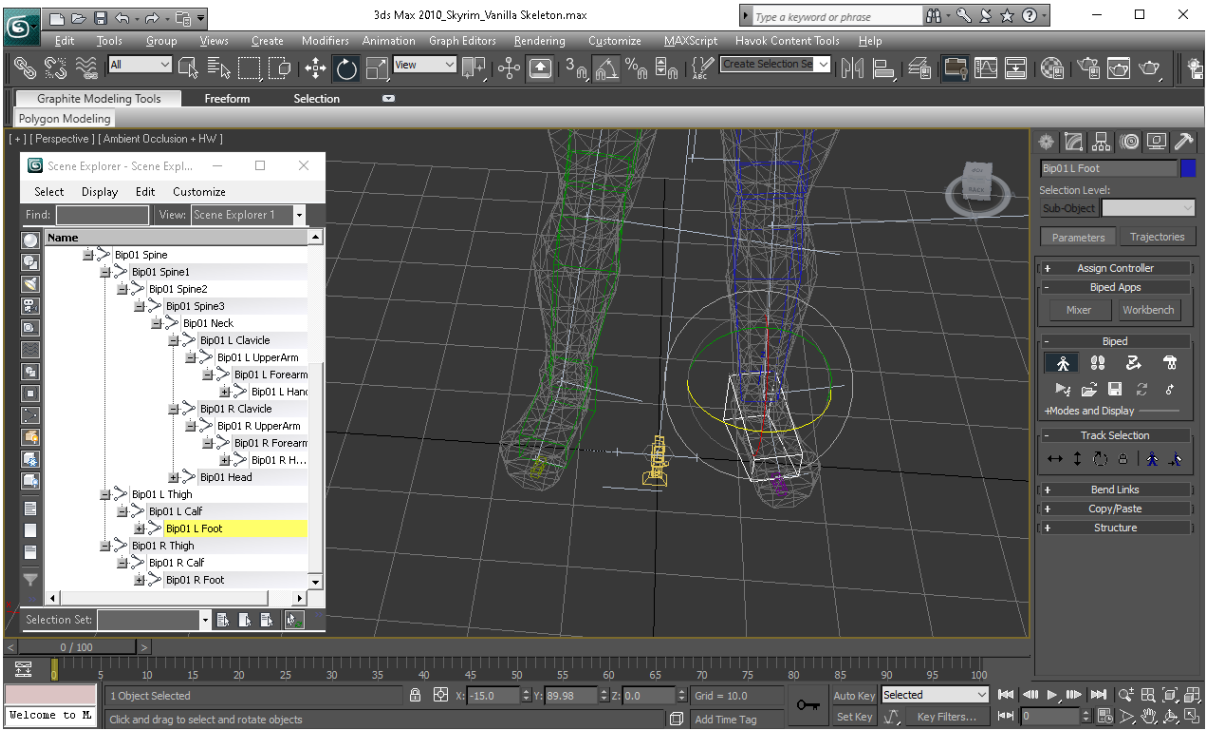


Figure 1242 - Rotating the feet on Bip01to match the feet on the skeleton.

Rotate the hands about 10 degrees to match the skeleton.

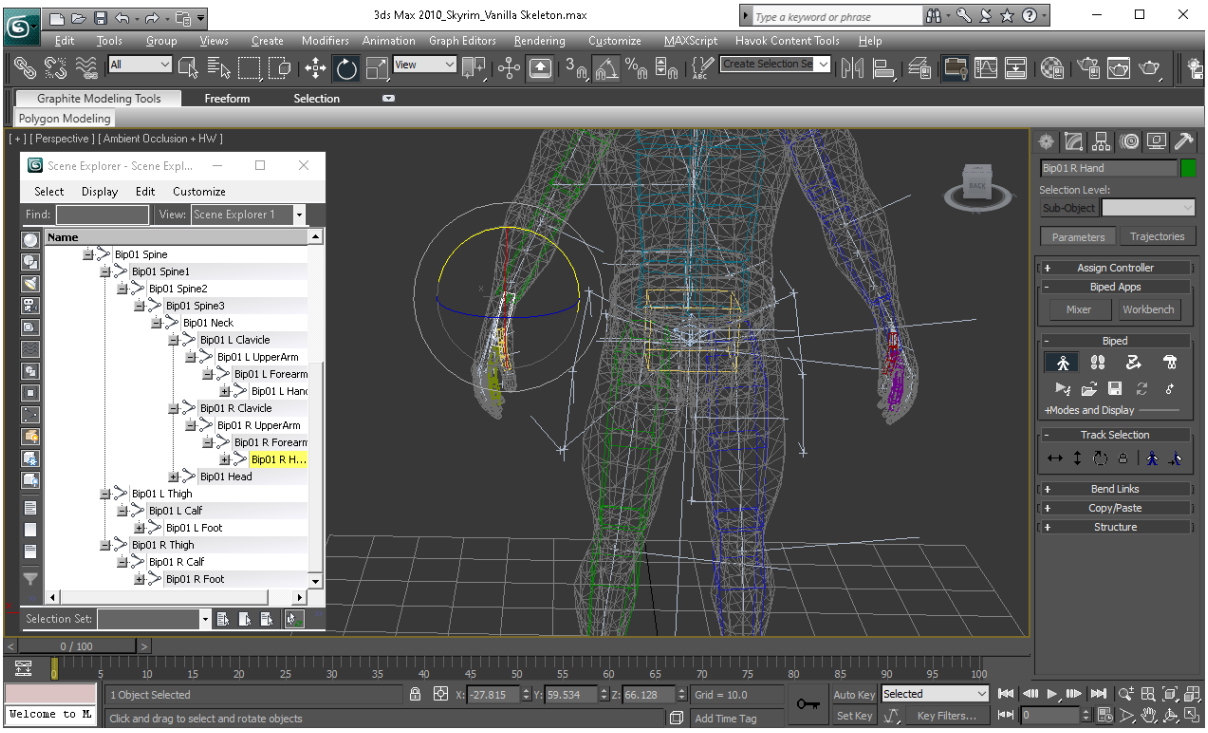


Figure 1243 - Rotating the hands on Bip01 slightly.

Now to set up constraints. For this example, I'm not going to constrain individual toes and fingers to the biped as the motion capture data I'll be working with doesn't capture that sort of information.

Let's start with the NPC L Toe0 [LToe] bone. Expand the NPC object in the hierarchy list and click on this bone to select it.

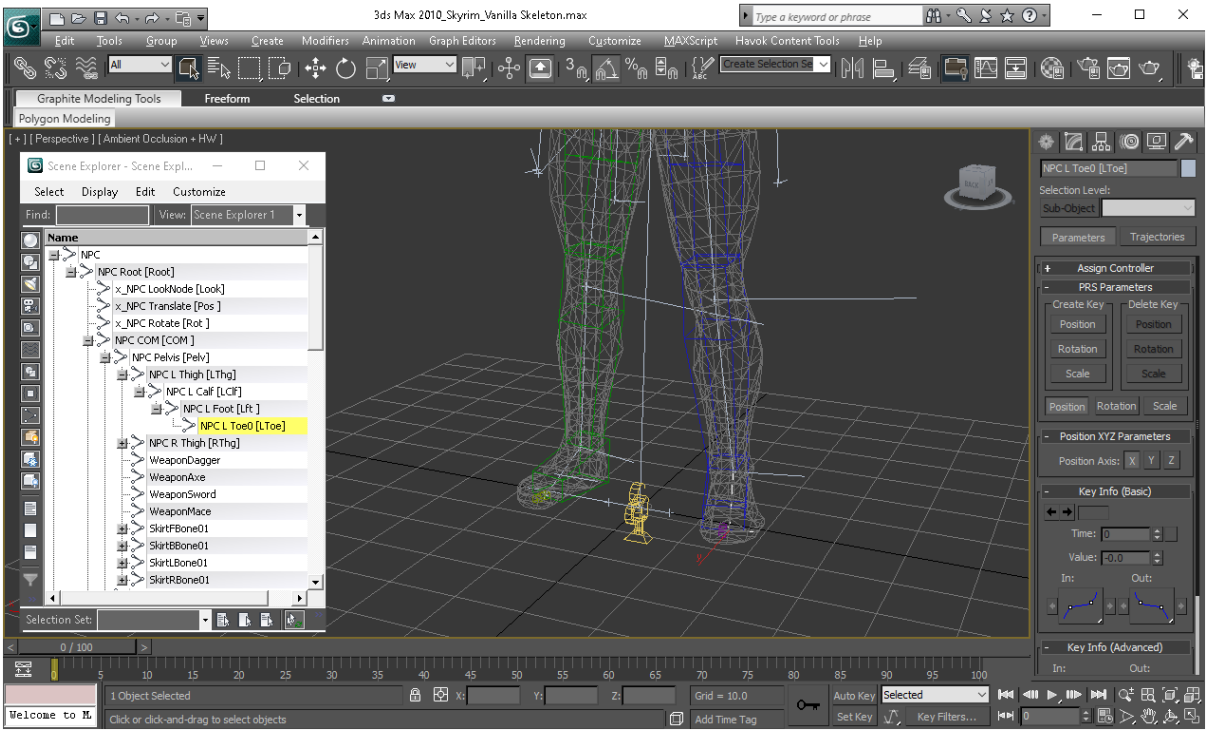


Figure 1244 - Selecting NPC L Toe0 [LToe].

Go to Animation > Constraints > Orientation Constraint.

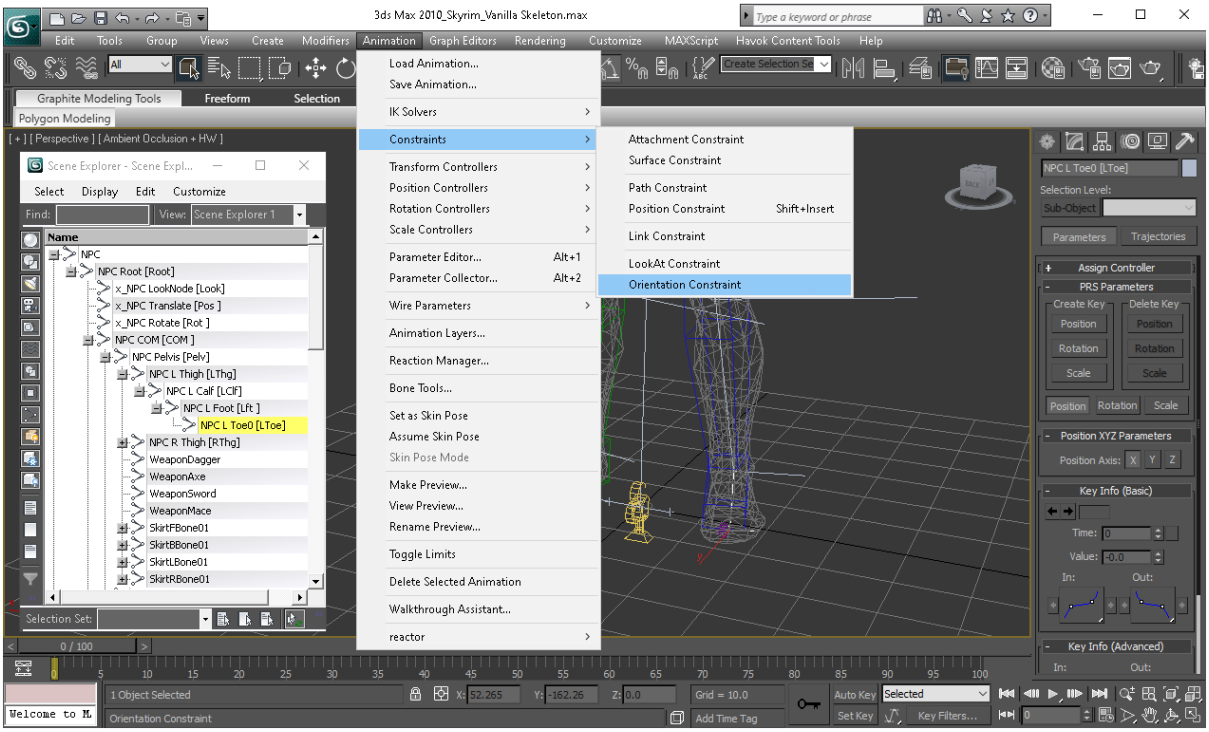


Figure 1245 - Setting up an Orientation Constraint.

Link it to the toe bone closest to Toe0 on Bip01's left foot.

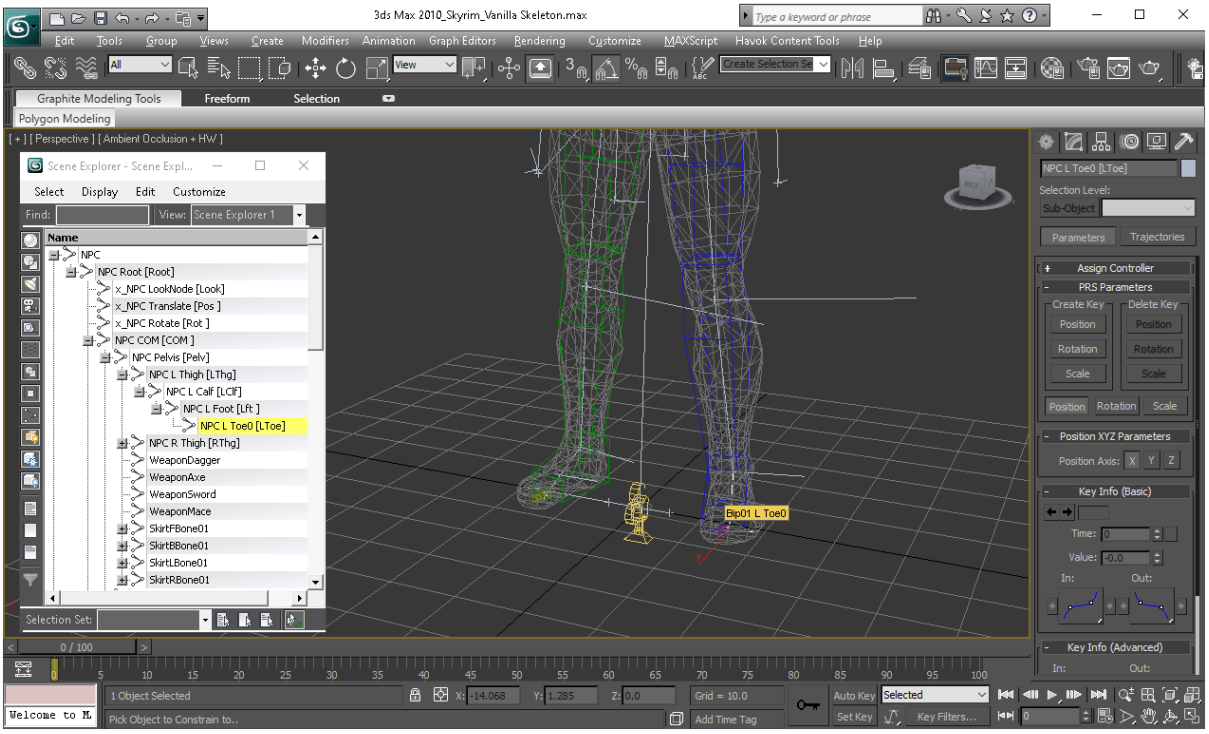


Figure 1246 - Linking the left toe bone.

On the right hand side, under Orientation Constraint, tick 'Keep Initial Offset'.

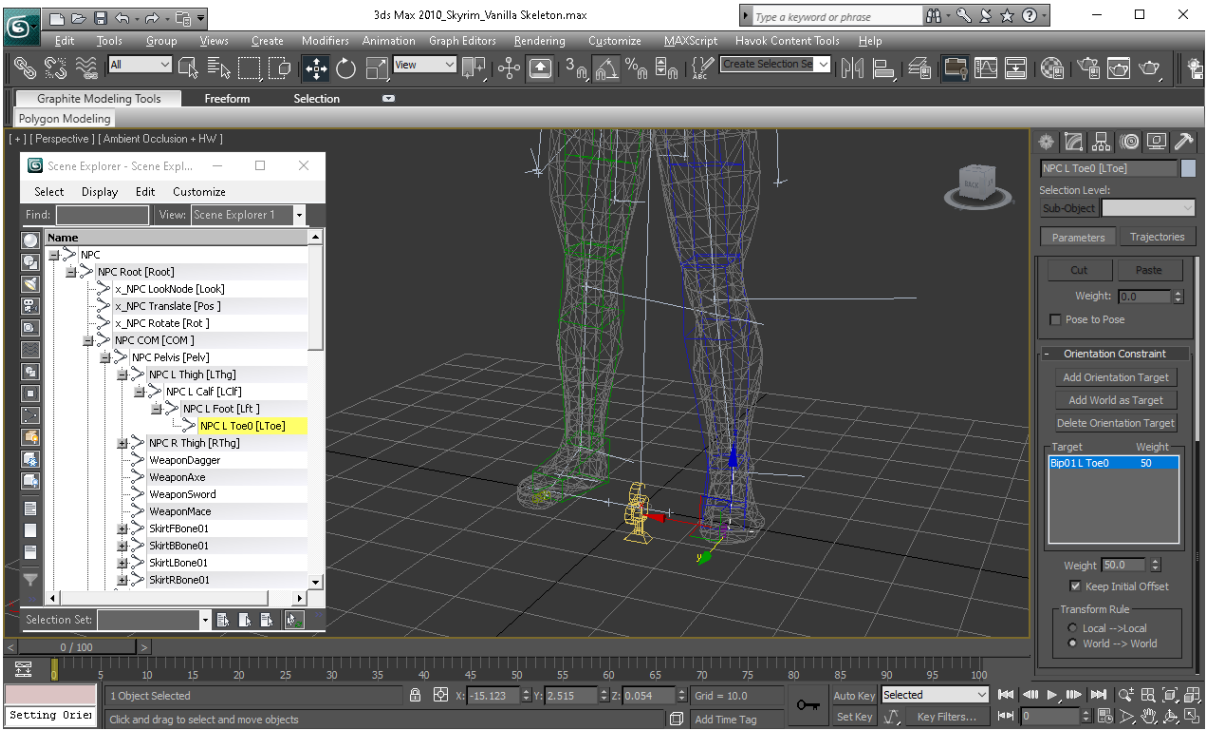


Figure 1247 - Ticking Keep Initial Offset.

Now move up to NPC L Foot [Lft]. Link it to Bip01's left foot. Remember to tick Keep Initial Offset for each bone we link.

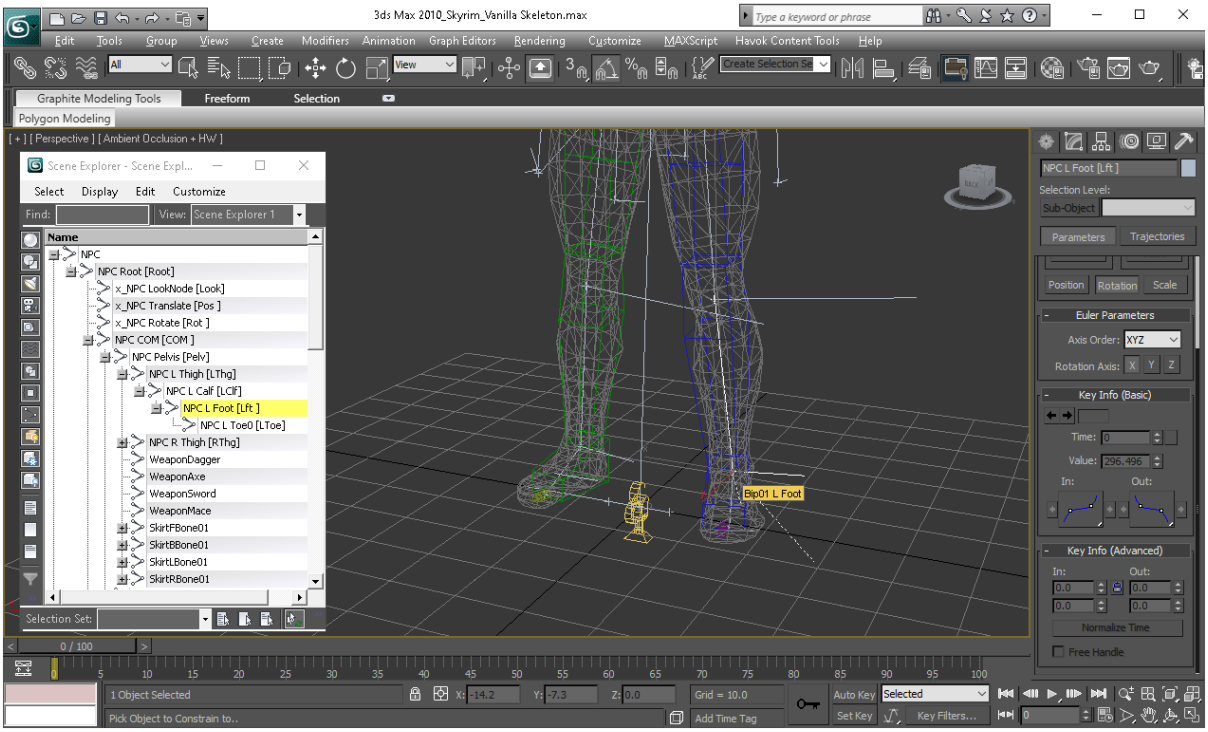


Figure 1248 - Linking the left foot bone.

Next, link NPC L Calf [LClf].

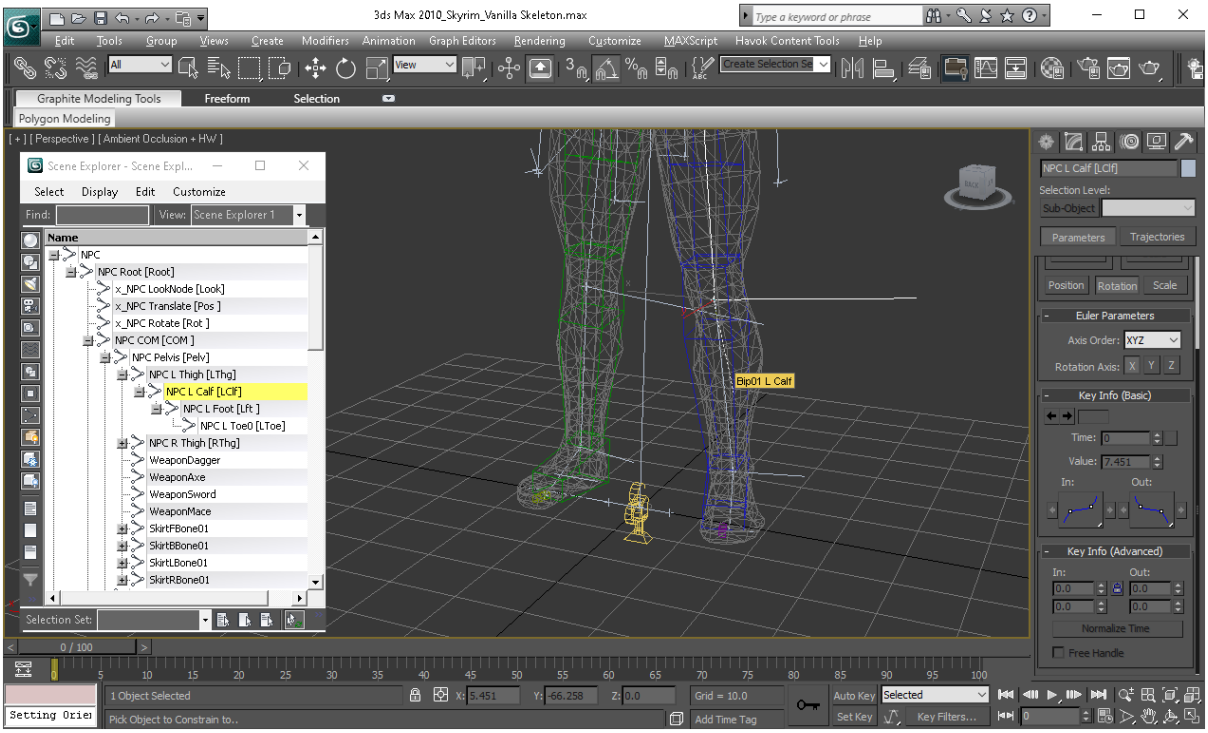


Figure 1249 - Linking the left calf bone.

Then NPC L Thigh [LThg].

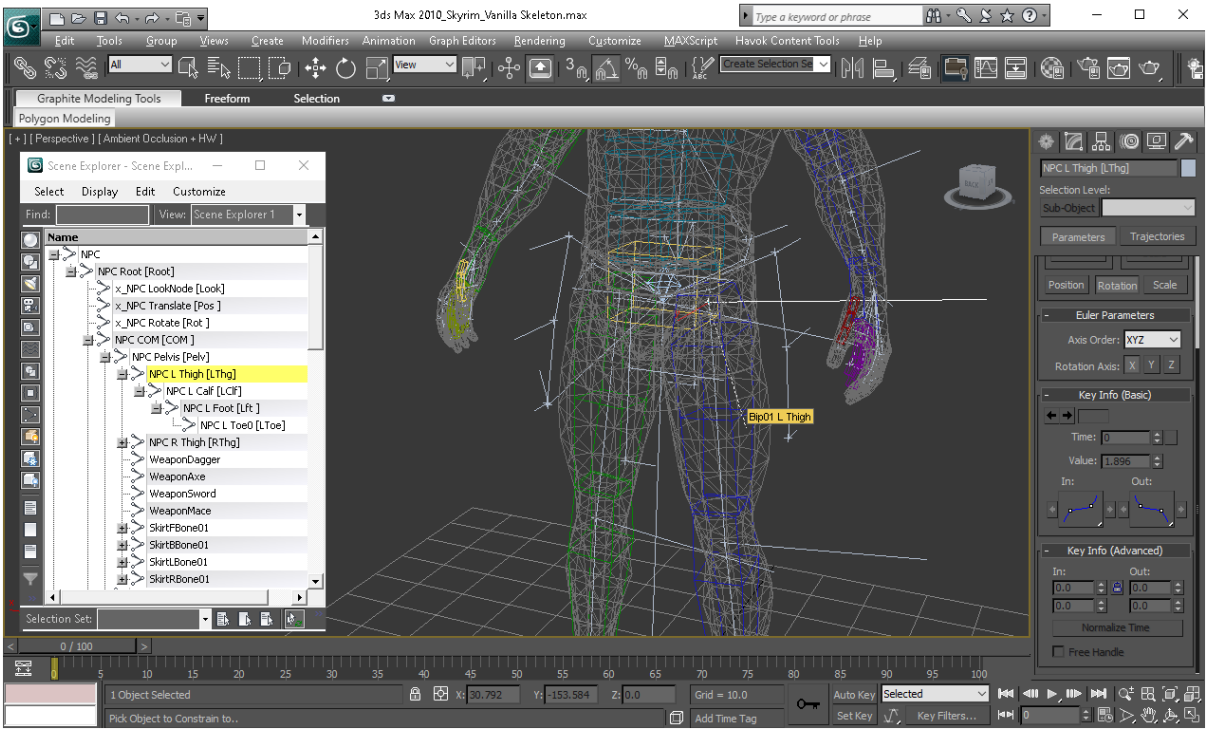


Figure 1250 - Linking the left thigh bone.

Now repeat these steps for the right leg, starting with NPC R Toe0 [RToe].

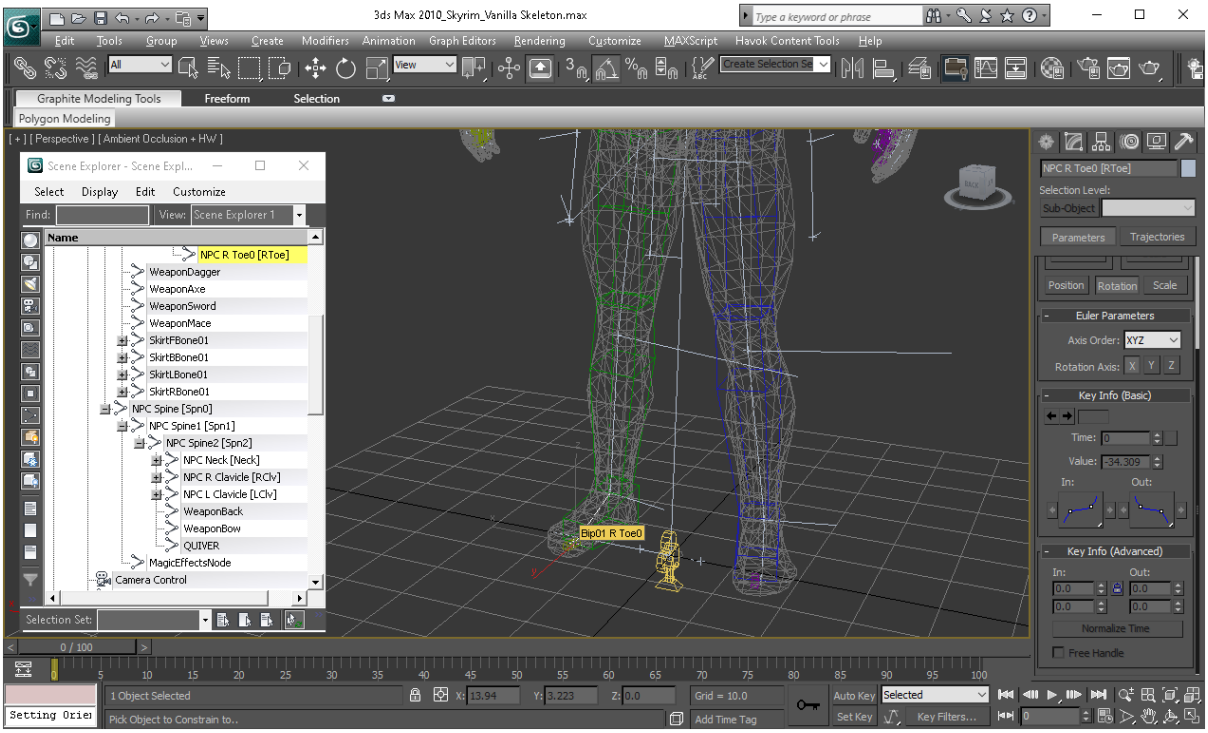


Figure 1251 - Linking bones on the right leg.

Next, the arms.

Start with NPC L Hand [LHnd] and link it to the left hand on Bip01.

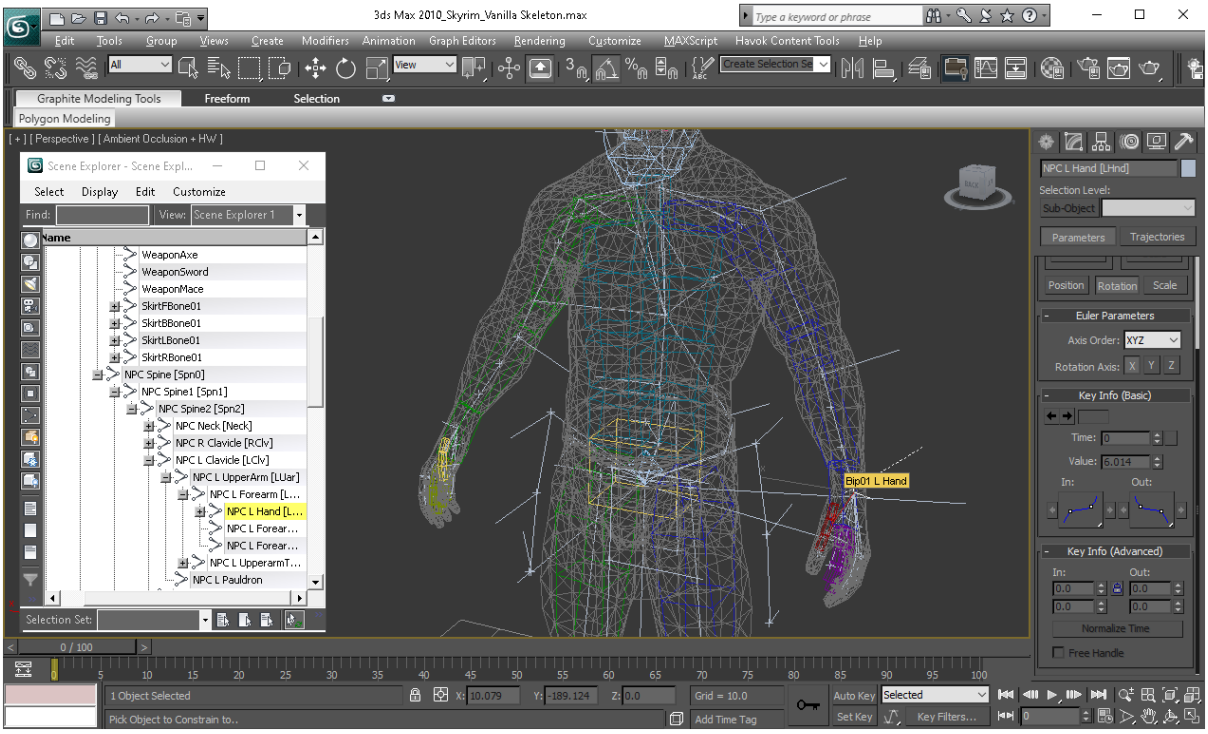


Figure 1252 - Linking the left hand bones.

NPC L Forearm [LLar] is next.

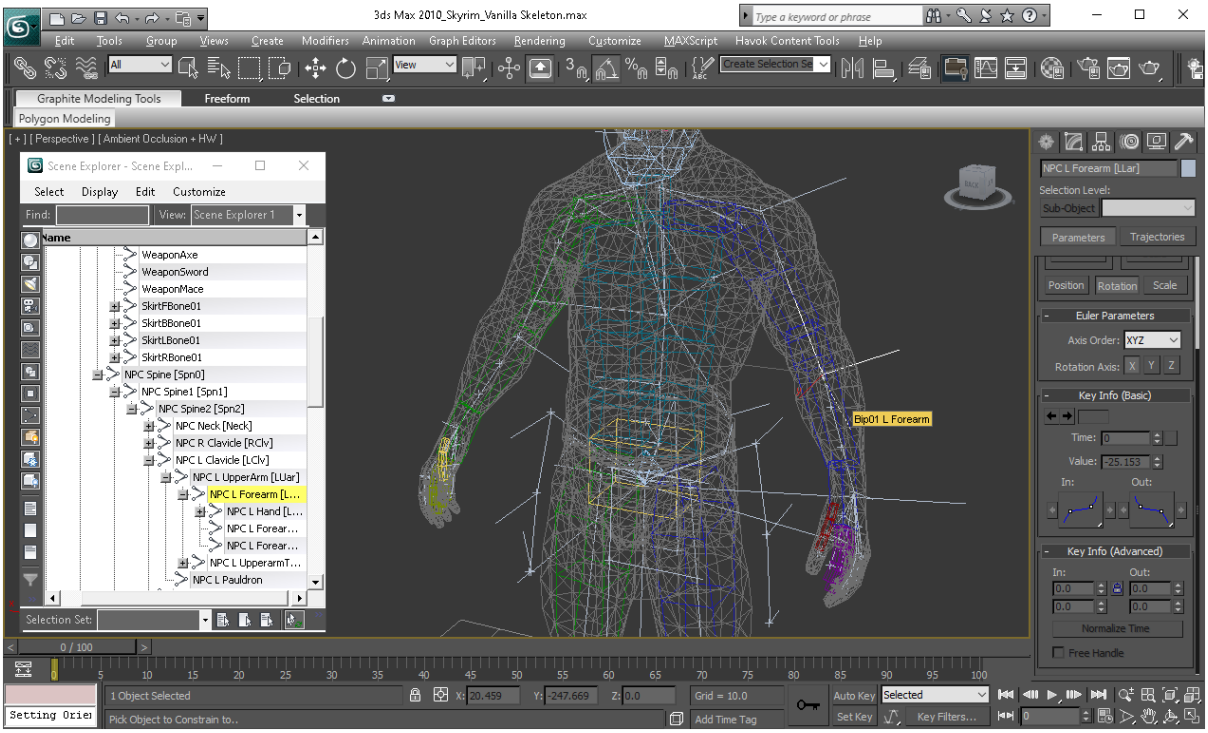


Figure 1253 - Linking the forearm bones.

Then NPC L UpperArm [LUar].

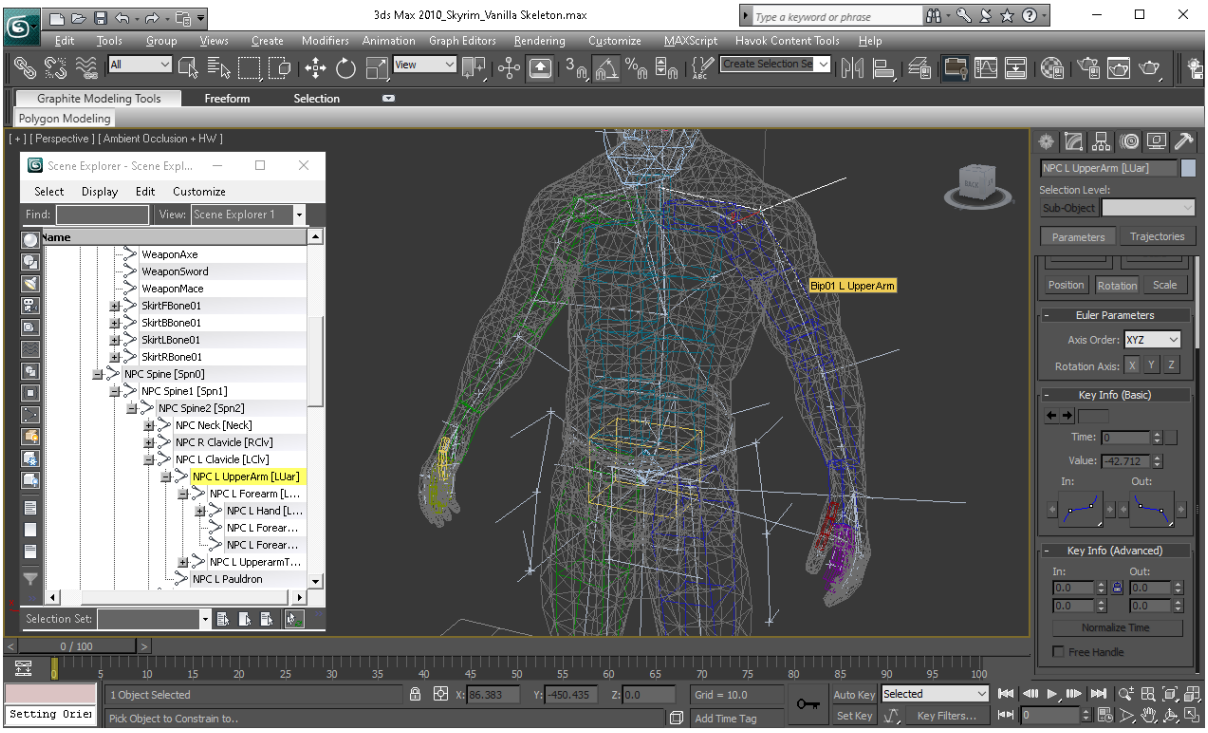


Figure 1254 - Linking the upper arm bones.

Finally NPC L Clavicle [LClv].

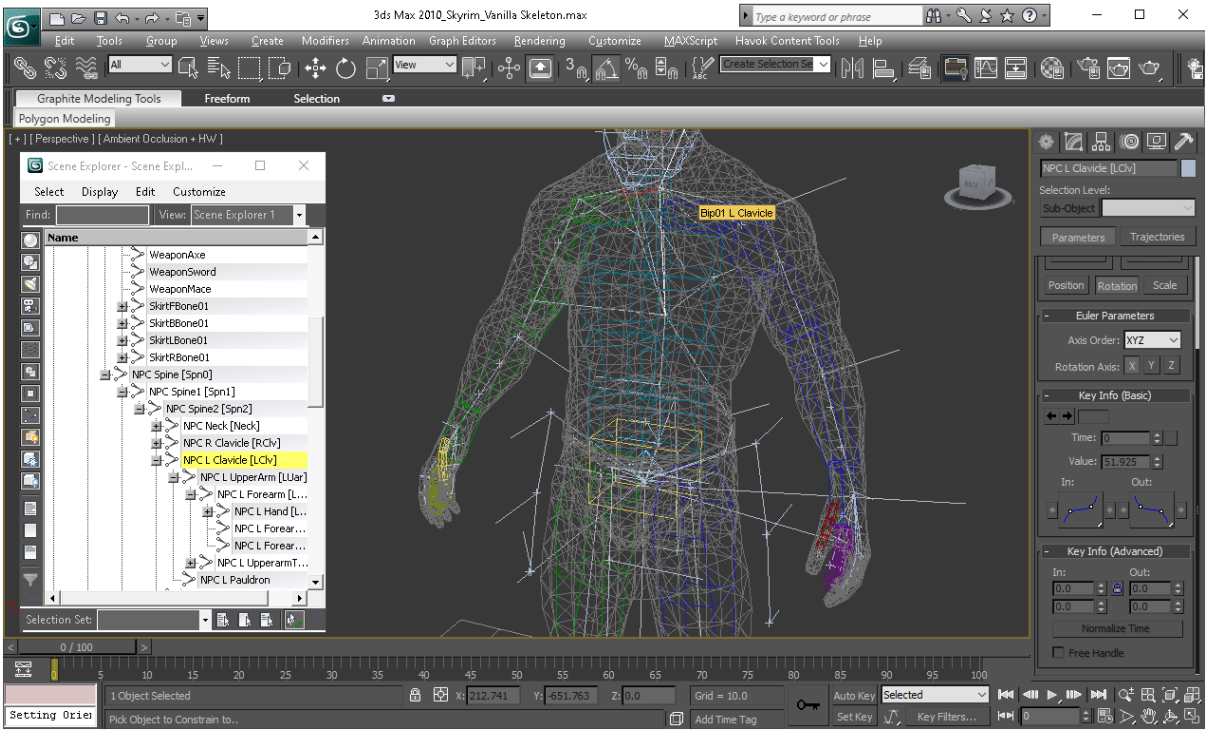


Figure 1255 - Linking the clavicle.

Now repeat these steps for the right arm.

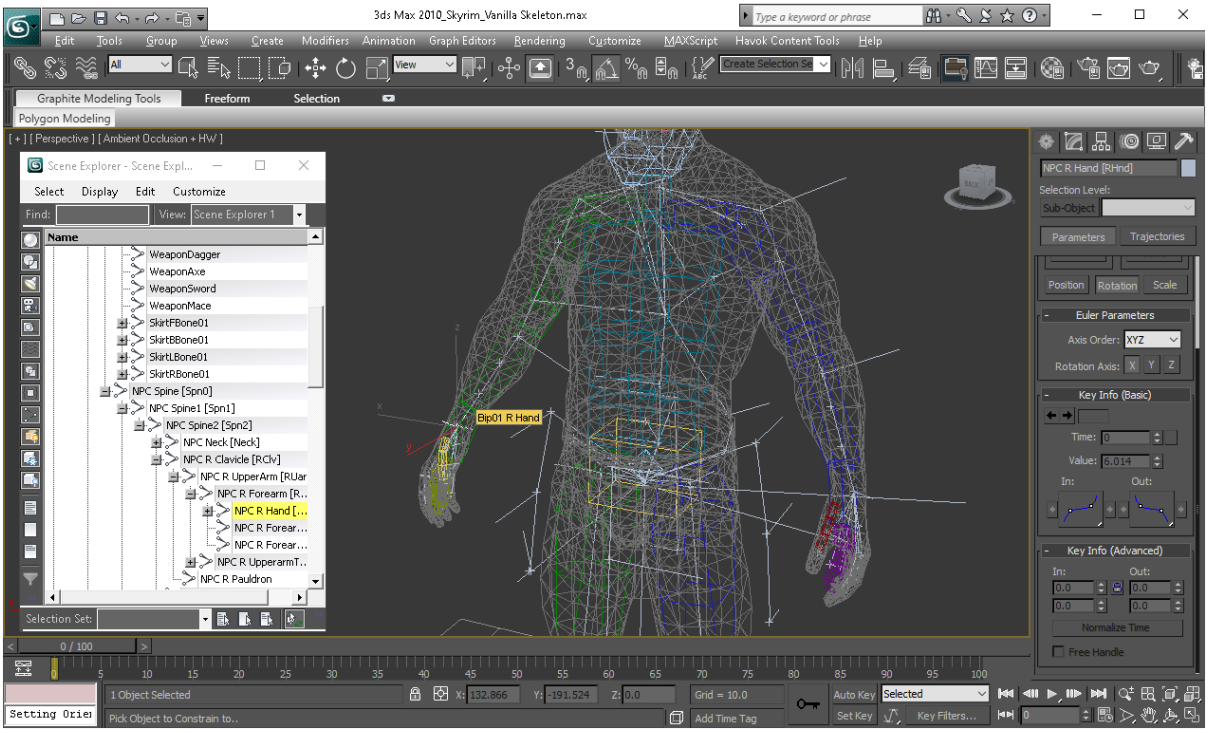


Figure 1256 - Linking the bones in the right arm.

Link NPC Neck [Neck] to Bip01's neck bone.

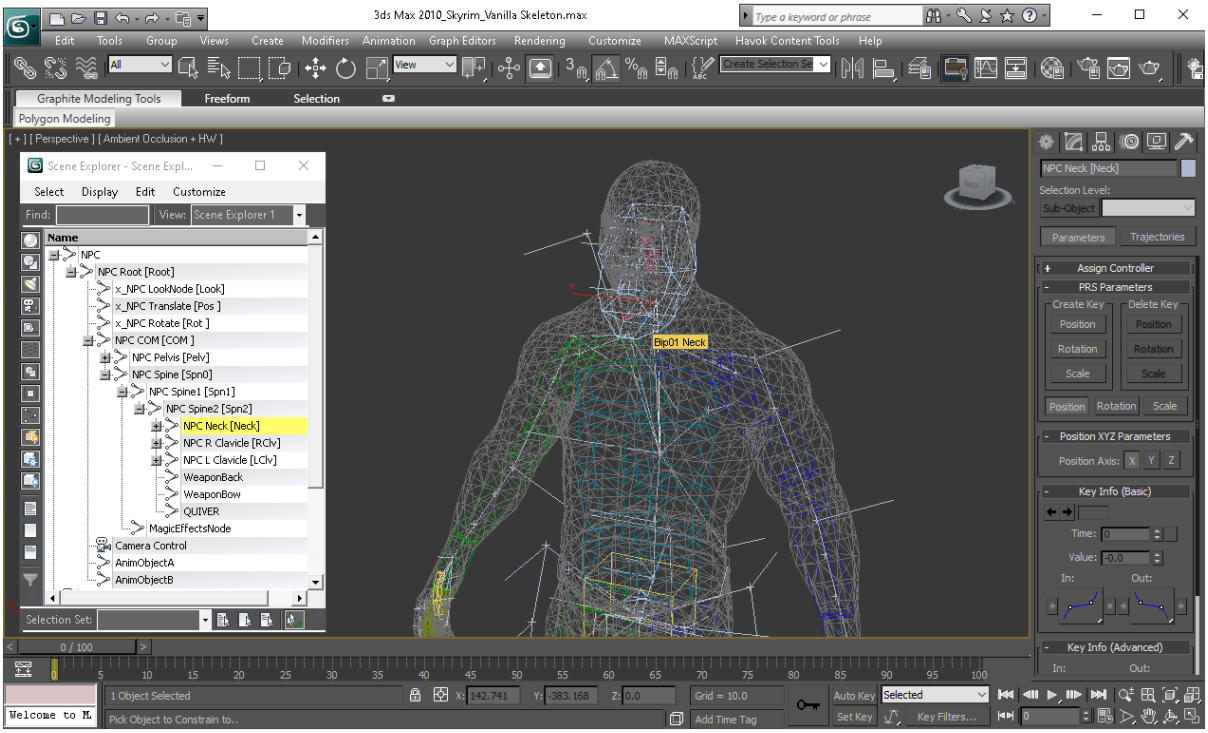


Figure 1257 - Linking the neck bone.

Then link NPC Head [Head] to Bip01's head bone.

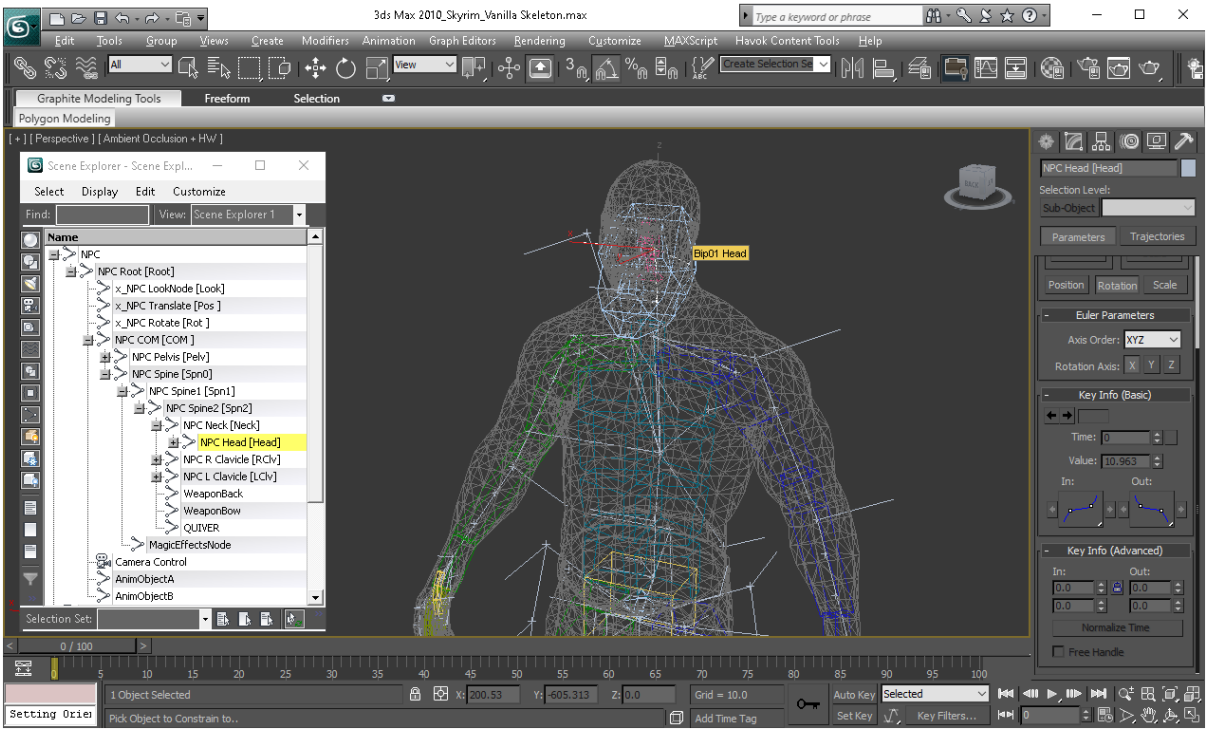


Figure 1258 - Linking the head bone.

Link NPC Pelvis [Pelv] to Bip01's pelvis bone.

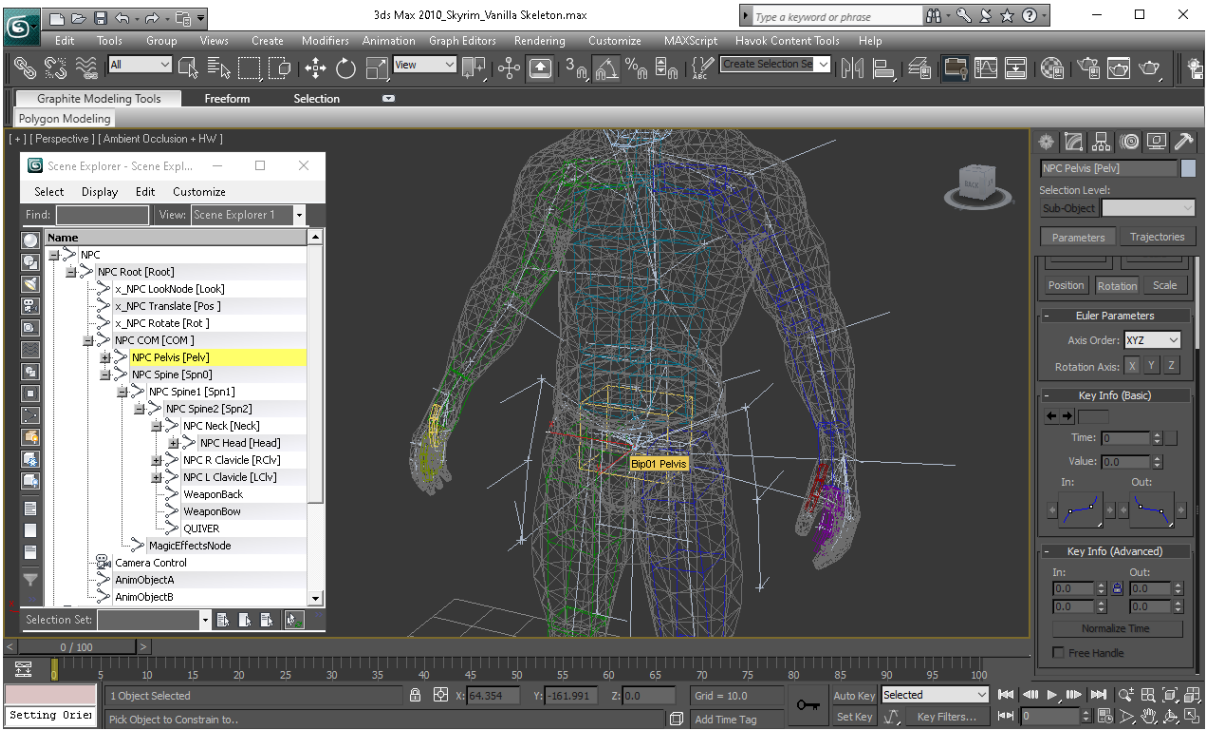


Figure 1259 - Linking the pelvis bone.

Link NPC Spine [Spn0] to the bottom spine bone.

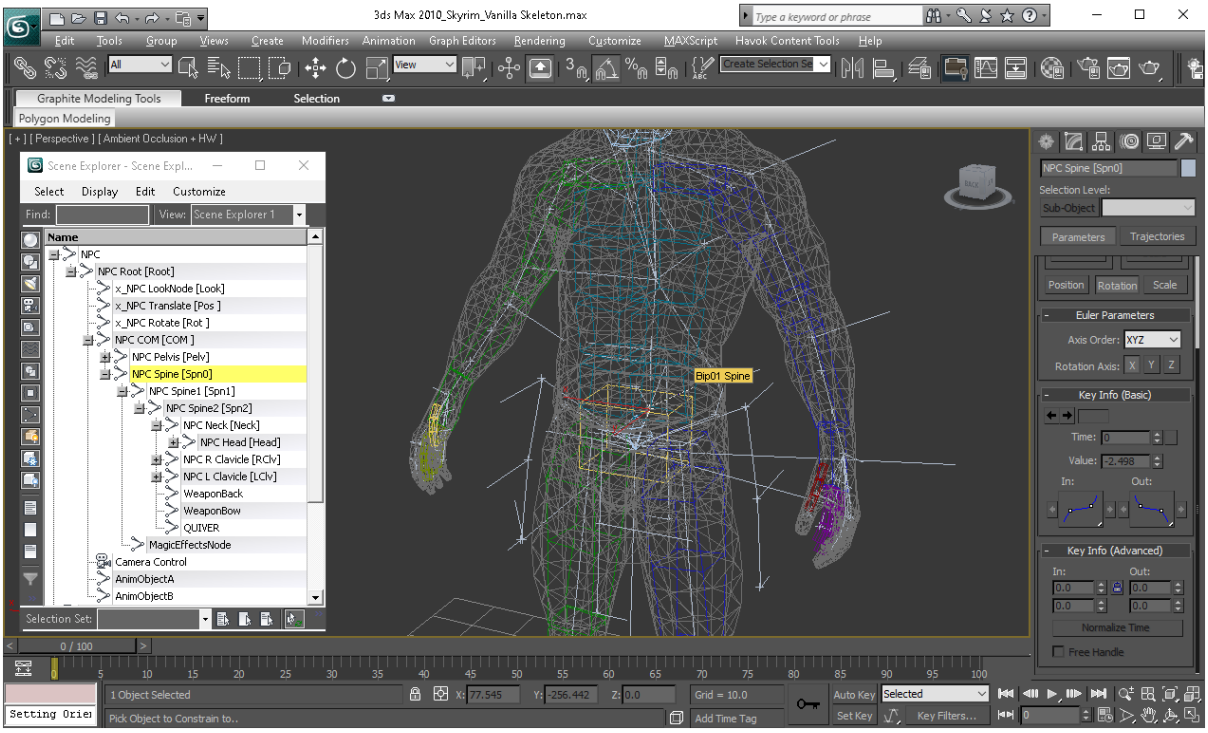


Figure 1260 - Linking the first spine bone.

Link NPC Spine1 [Spn1] to the spine bone above the one we previously linked.

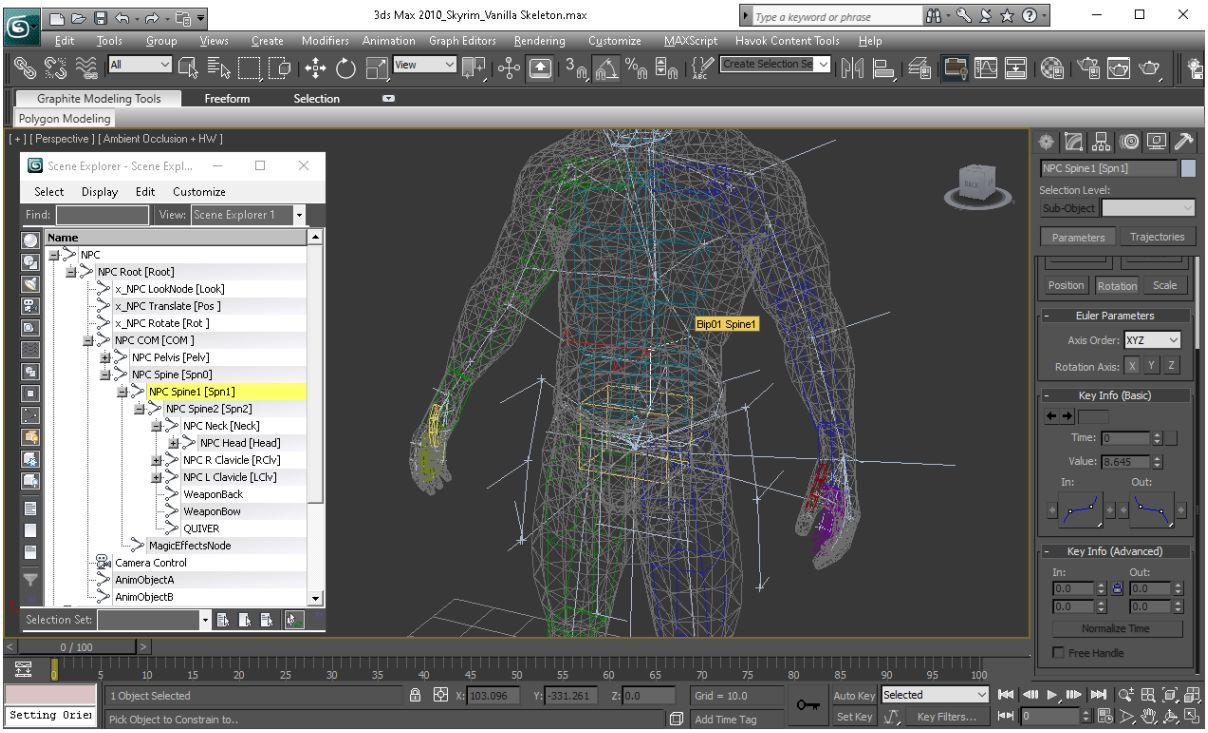


Figure 1261 - Linking the second spine bone.

And link NPC Spine2 [Spn2] to the next spine bone above. The spine bone above that one on Bip01 shouldn't be linked to anything.

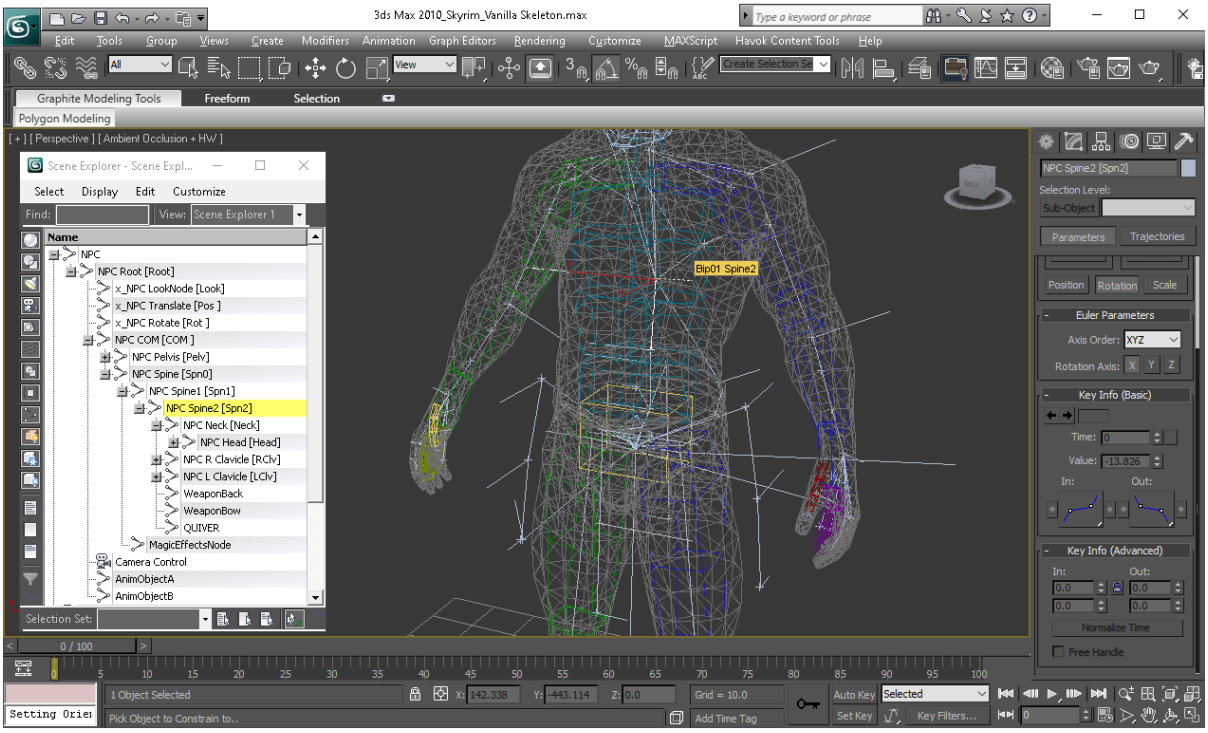


Figure 1262 - Linking the third spine bone.

Next, we need to set up a position constraint.

In order to do that, select NPC COM [COM], then go to Animation > Constraints > Position Constraint.

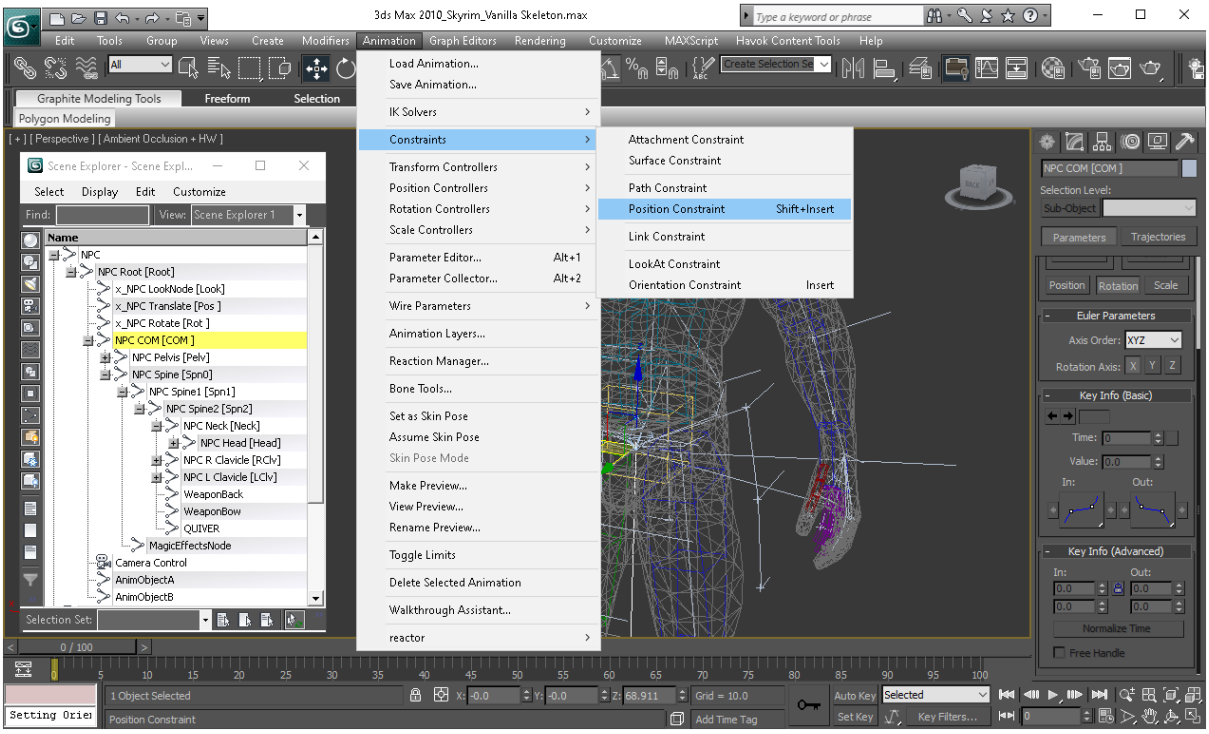


Figure 1263 - Setting up a position constraint.

Link NPC COM [COM] to the pelvis bone on Bip01. Don't forget to tick Keep Initial Offset.

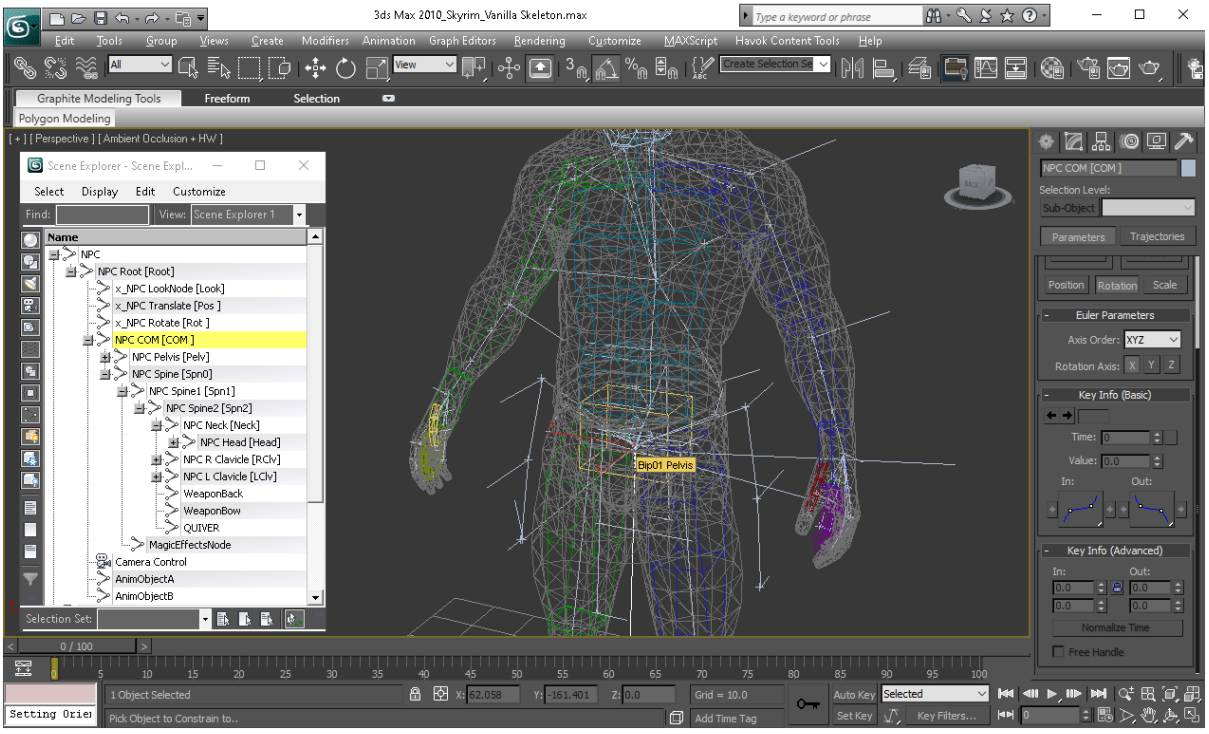


Figure 1264 - Linking NPC COM [COM] to the pelvis bone on Bip01.

We'll also need to add an Orientation Constraint on NPC COM [COM] linking it to the pelvis bone on Bip01 to ensure the spine doesn't get bent out of place when importing mocap data.

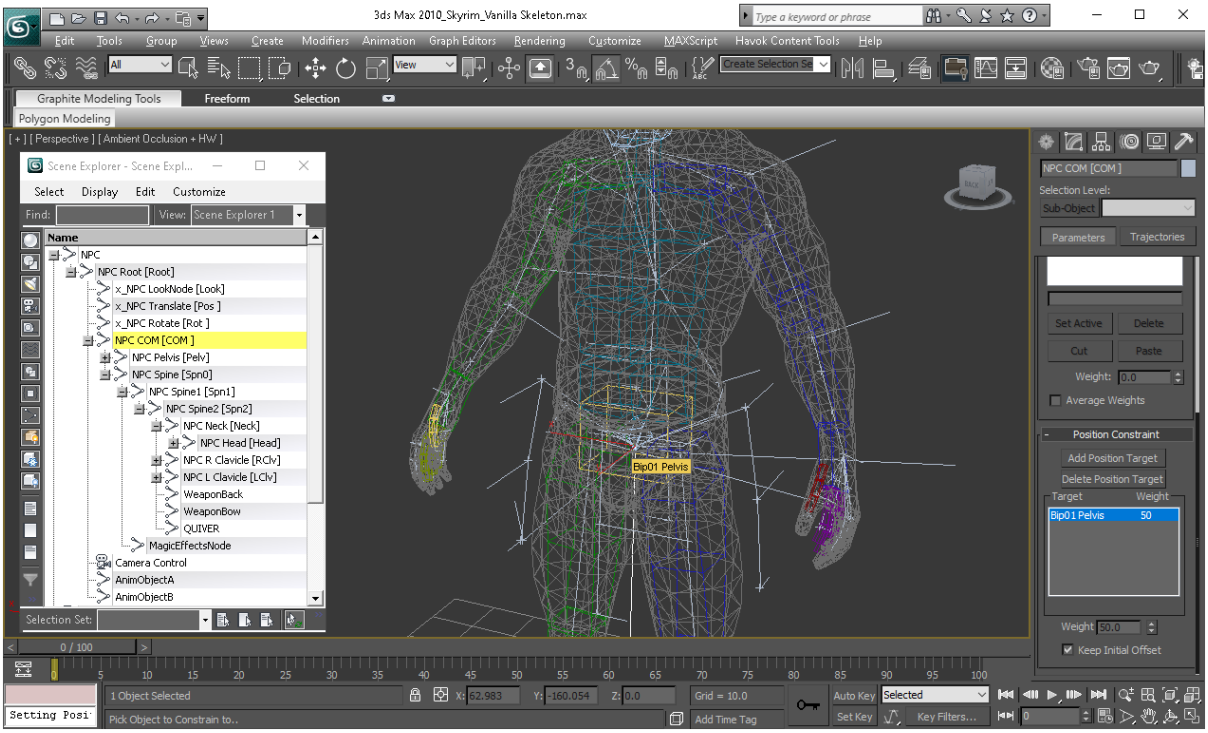


Figure 1265 - Adding an orientation constraint to NPC COM [COM].

Click on Bip01 in the hierarchy list. On the right hand side, go down to the Biped section and click on Figure Mode again to exit out of it.

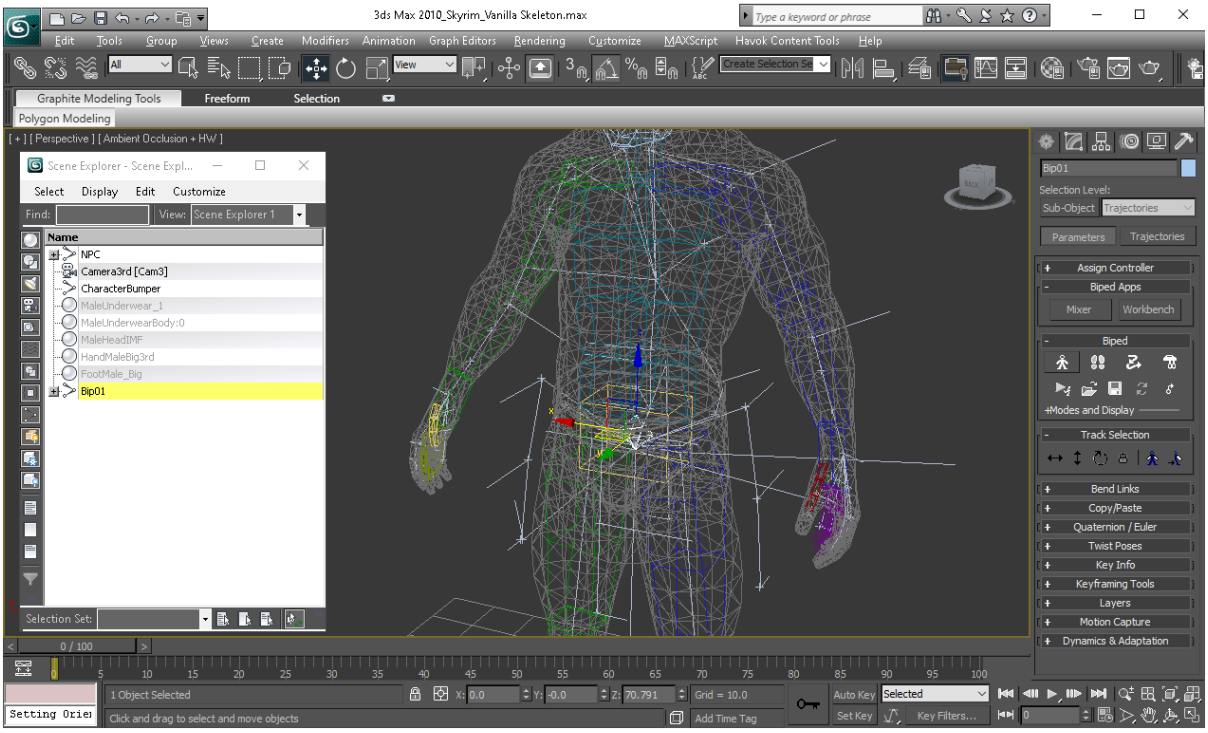


Figure 1266 - Exiting out of figure mode.

Now to test importing some motion capture data.

With Bip01 still selected, go down to the Motion Capture section and click on Load Motion Capture File.

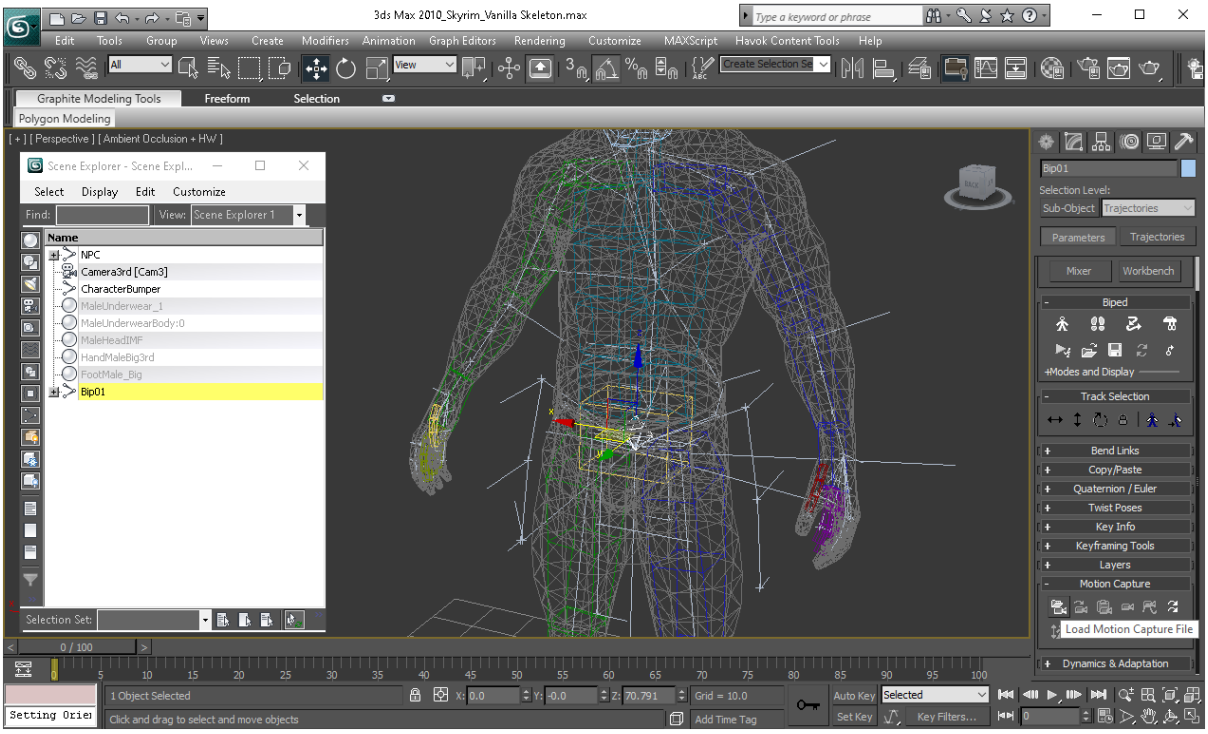


Figure 1267 - Loading a motion capture file.

For this example, I'm going to load a .bvh file provided by Carnegie Mellon University.

Set the 'Files of type' drop down to '(*.BVH)' and select a .bvh file to import.

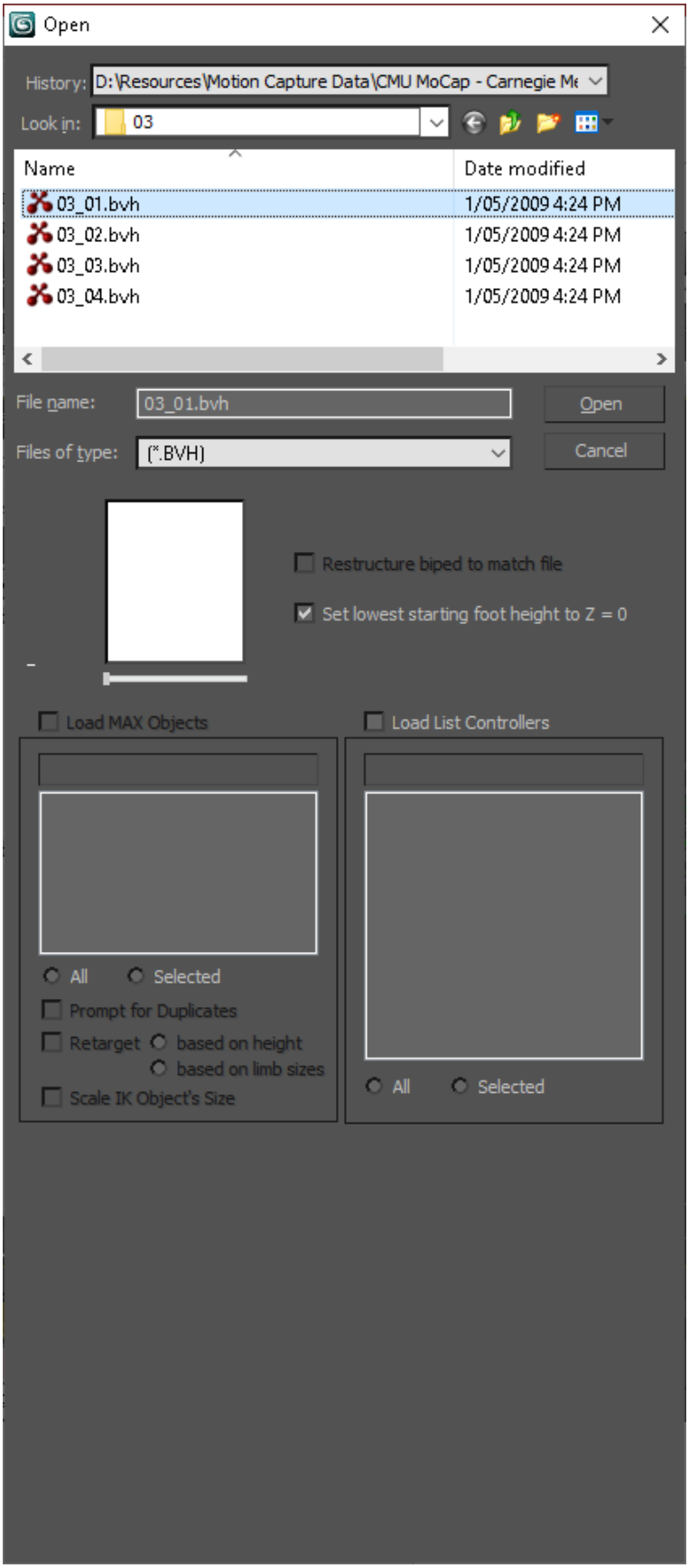


Figure 1268 - Selecting a .bvh file to load.

Click OK to dismiss this warning.

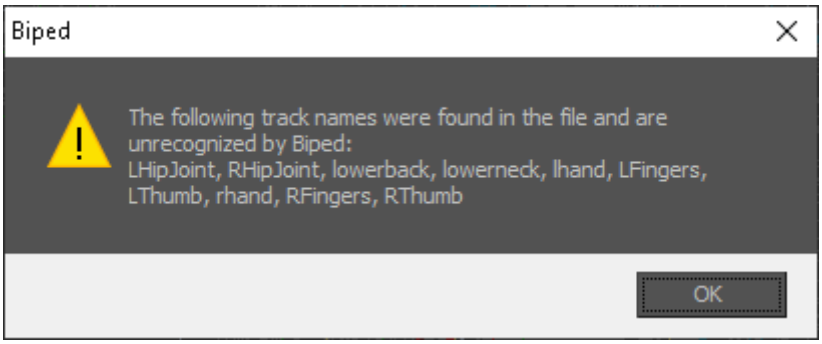


Figure 1269 - Unrecognised bones.

For mocap data from Carnegie Mellon University, a Scale Factor of 4.5 seems to fit best.

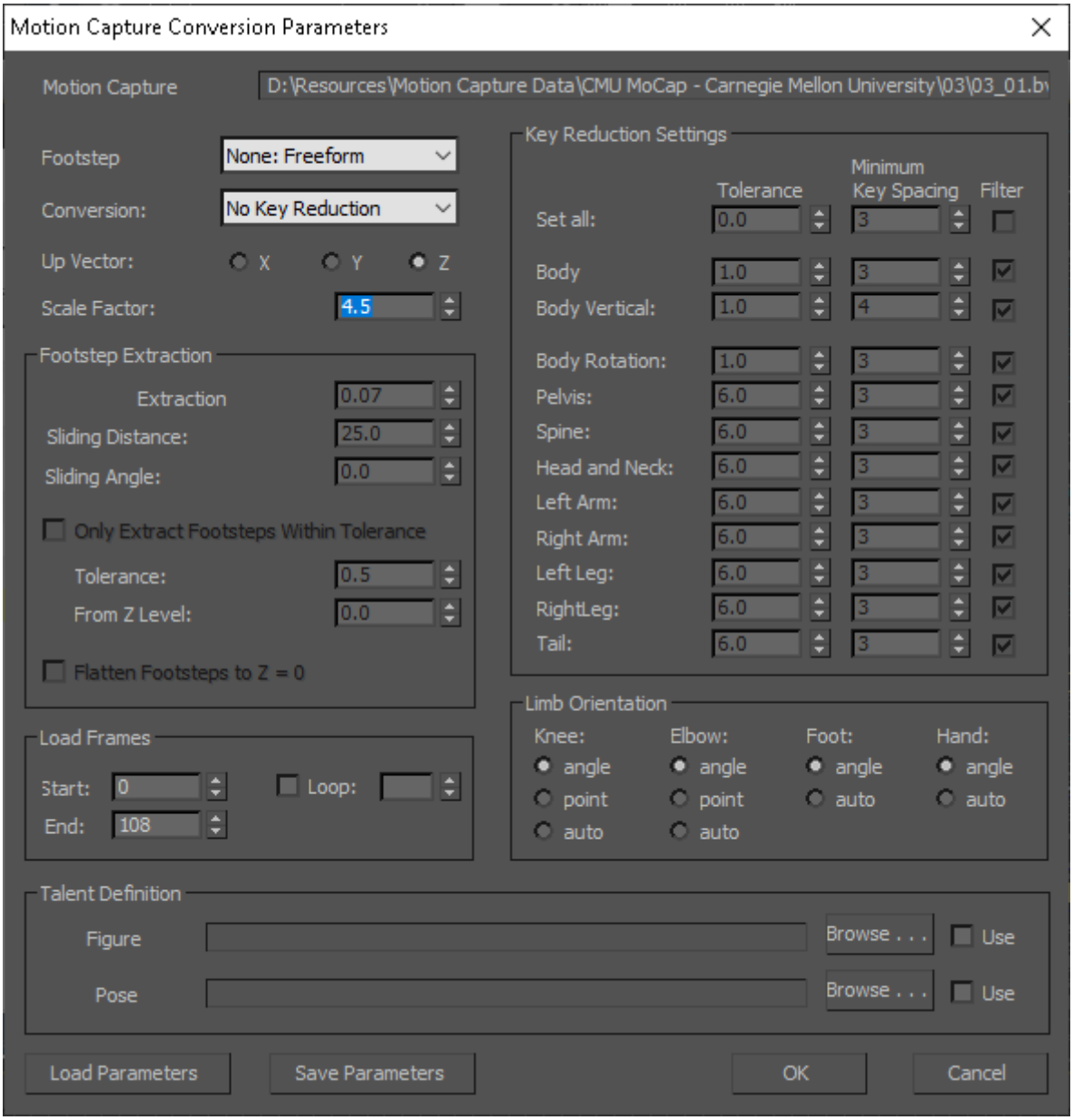


Figure 1270 - Motion Capture Conversion Parameters.

Set Knee and Elbow to 'angle' then click OK.

Scrub the timeline and confirm that the movement of our skeleton matches the movement of the biped.

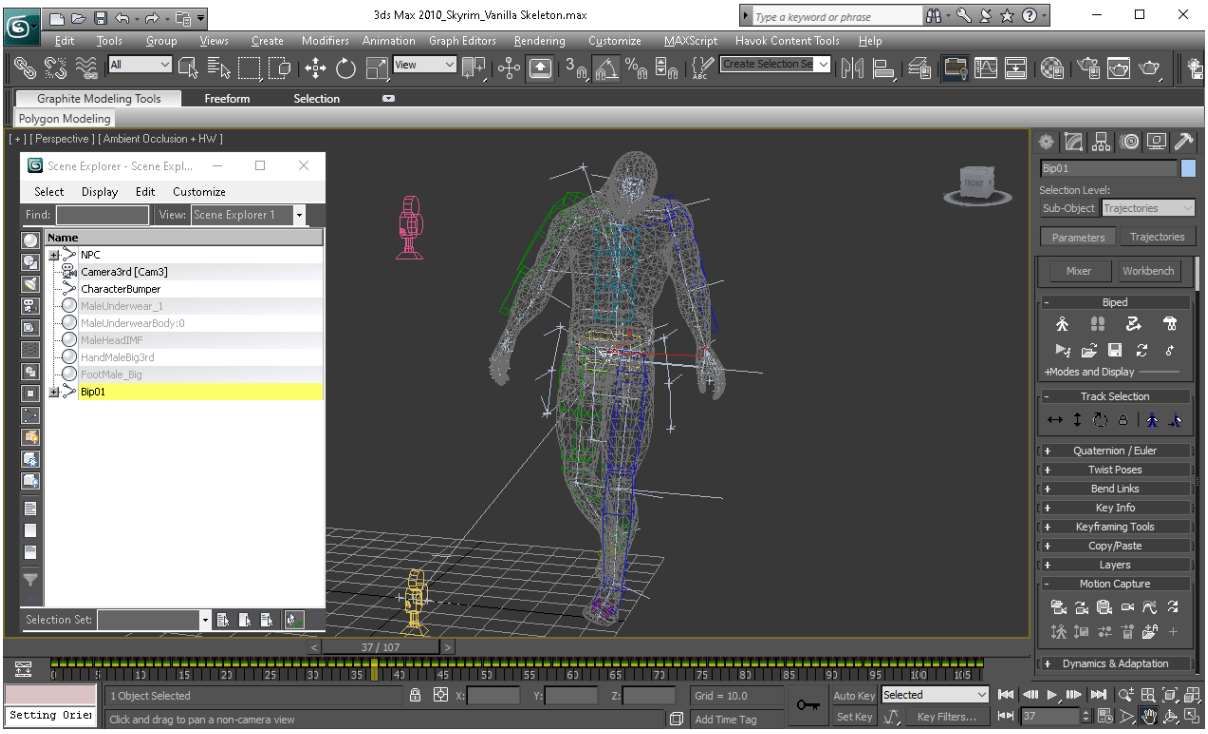


Figure 1271 - The movement of our skeleton constrained to the movement of the biped.

You'll notice that the size of the biped has changed a bit. You will likely need to play around with the Scale Factor when importing motion capture data until you match the size of the skeleton. This is to ensure that the position constraint moves the skeleton around accurately. For example, mocap data from Eyes, JAPAN seems to work better with a scale factor of about 1.8 instead.

EXPORTING ANIMATION FROM 3DS MAX WITH HAVOK CONTENT TOOLS

This section will cover the steps necessary for exporting animation from 3ds Max into Skyrim.

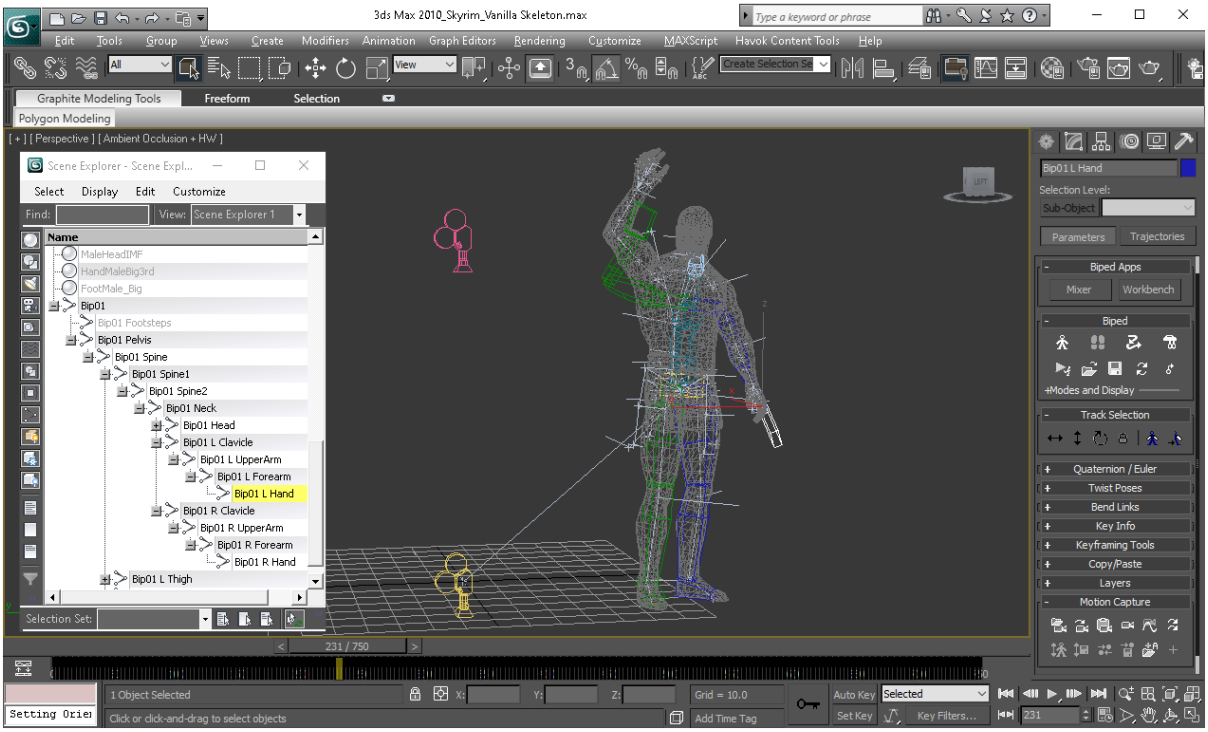


Figure 1272 - Waving motion capture in 3ds Max.

Select the skeleton and all child objects.

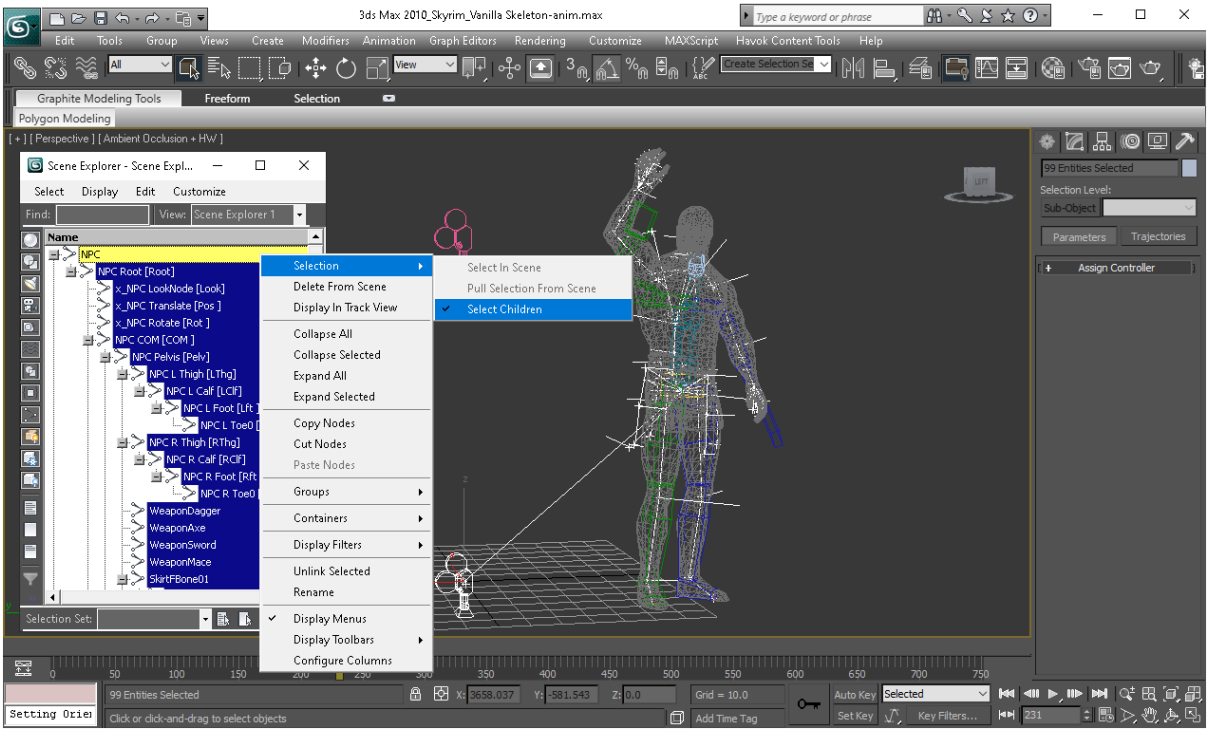


Figure 1273 - Selecting all bones.

Under the Motion tab on the right hand side, click on Trajectories.

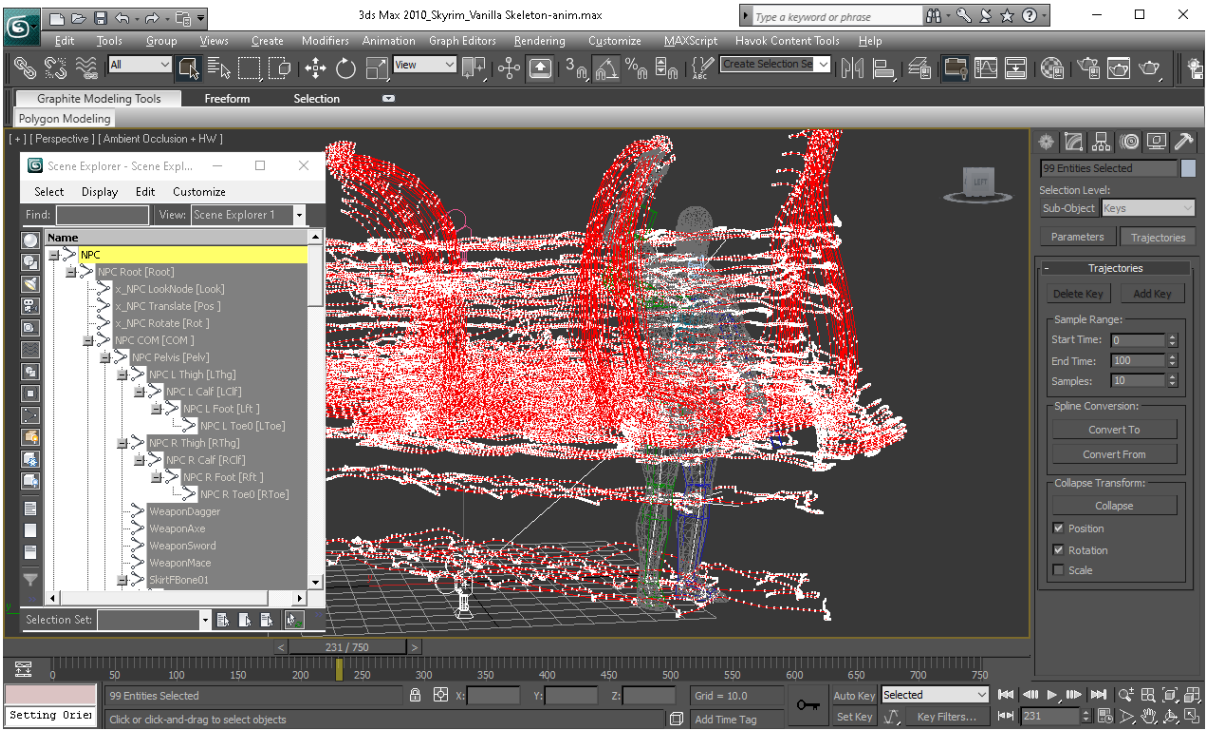


Figure 1274 - Showing trajectories.

Set Start Time to 0, End Time to the last frame (in this example that's 750), increase Samples to 500, and then click Collapse under Collapse Transform.

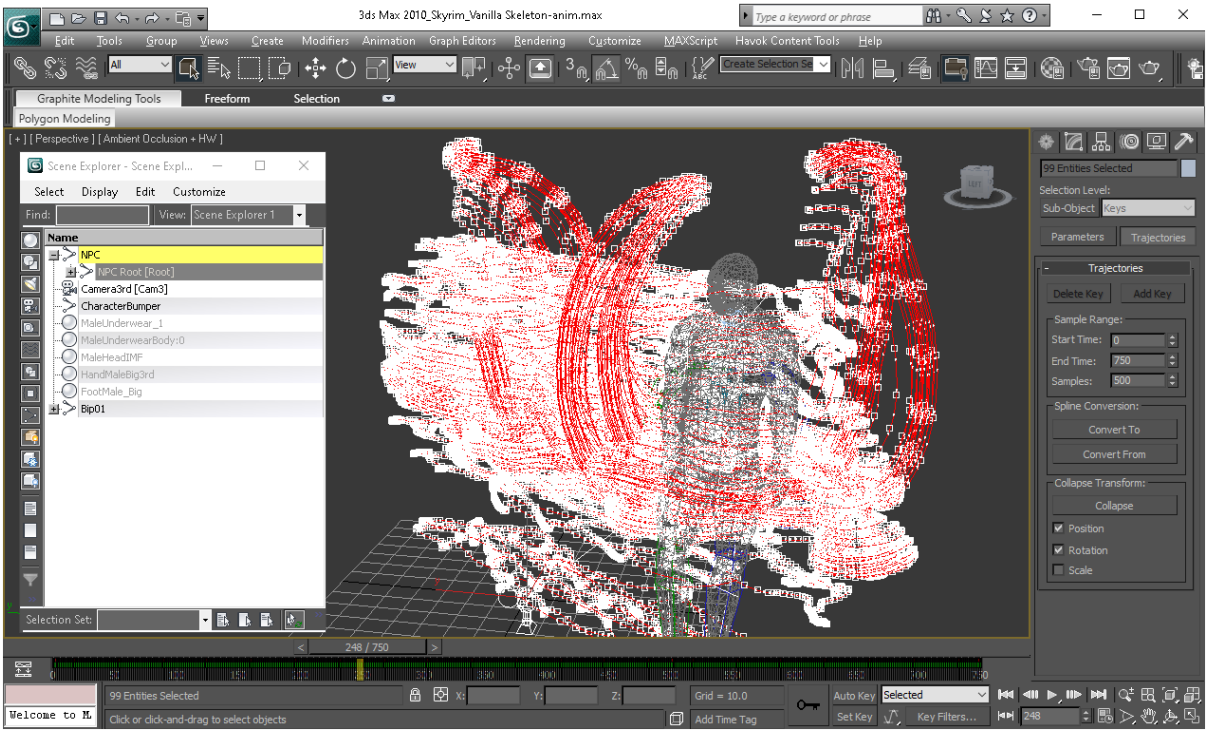


Figure 1275 - Copying transforms from the biped over to the skeleton.

You should see keyframes added to our skeleton's bones in the timeline below the viewport.

At this stage we won't need the biped anymore. Double-click on Bip01 to select all child objects then delete it.

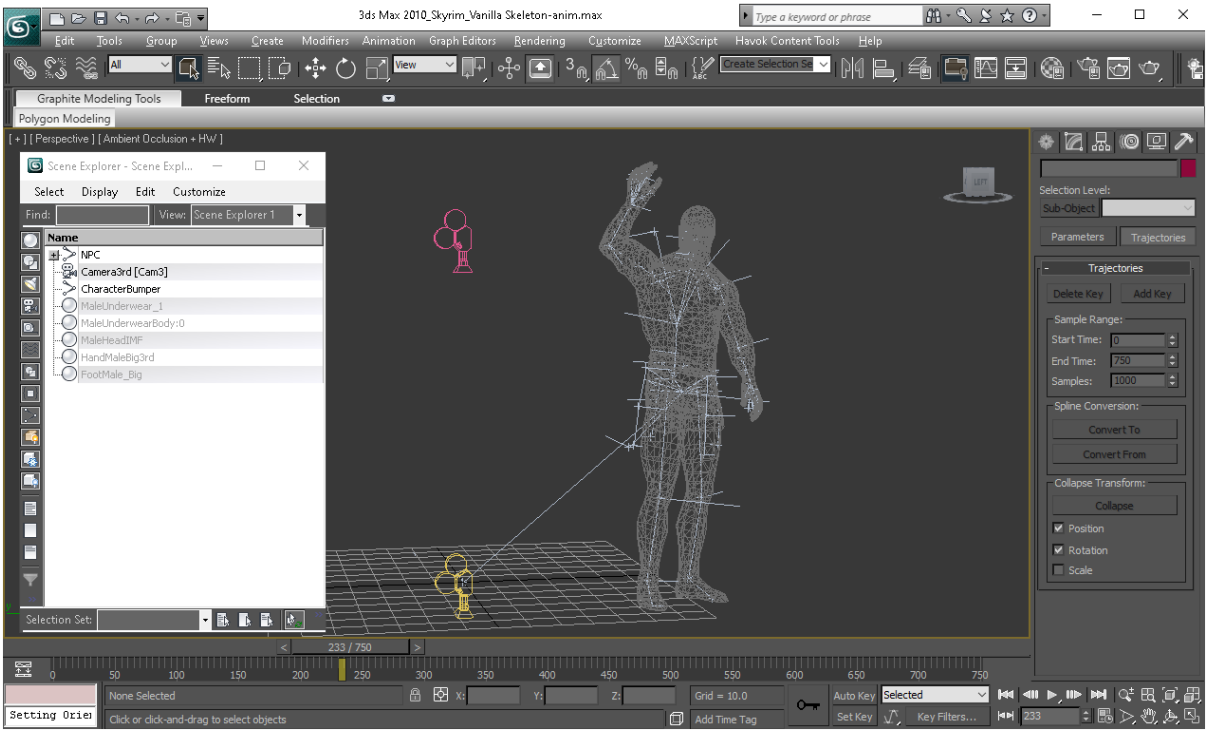


Figure 1276 - Deleting Bip01.

Scrub the timeline and confirm that the skeleton still moves without the biped.

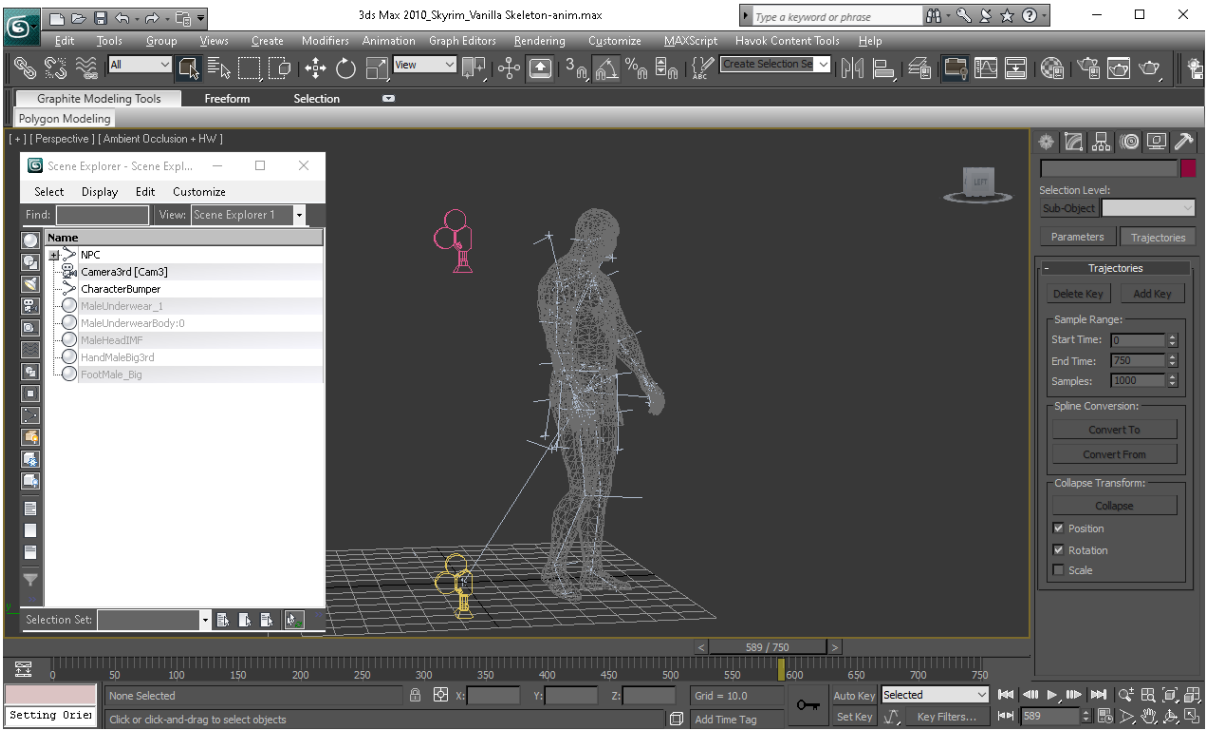


Figure 1277 - Skeleton still moves when scrubbing the timeline.

Select the skeleton and all child objects then right-click in the viewport and select Dope Sheet.

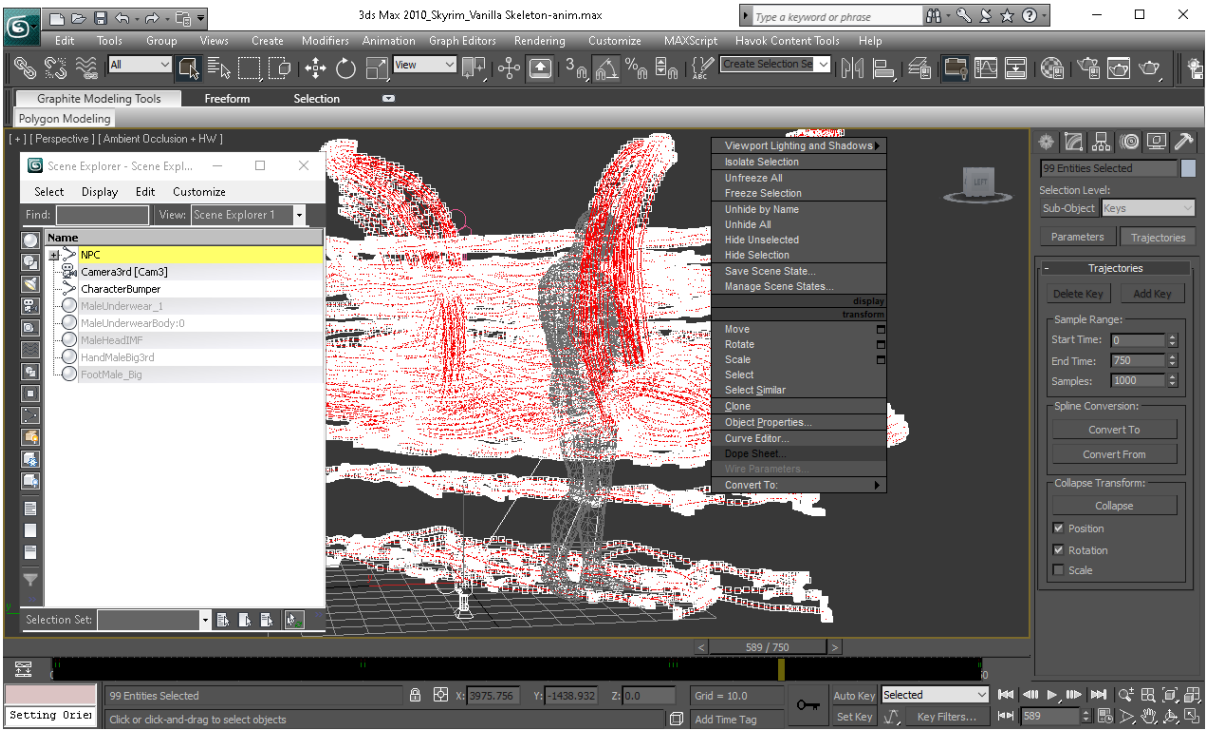


Figure 1278 - Opening the dope sheet.

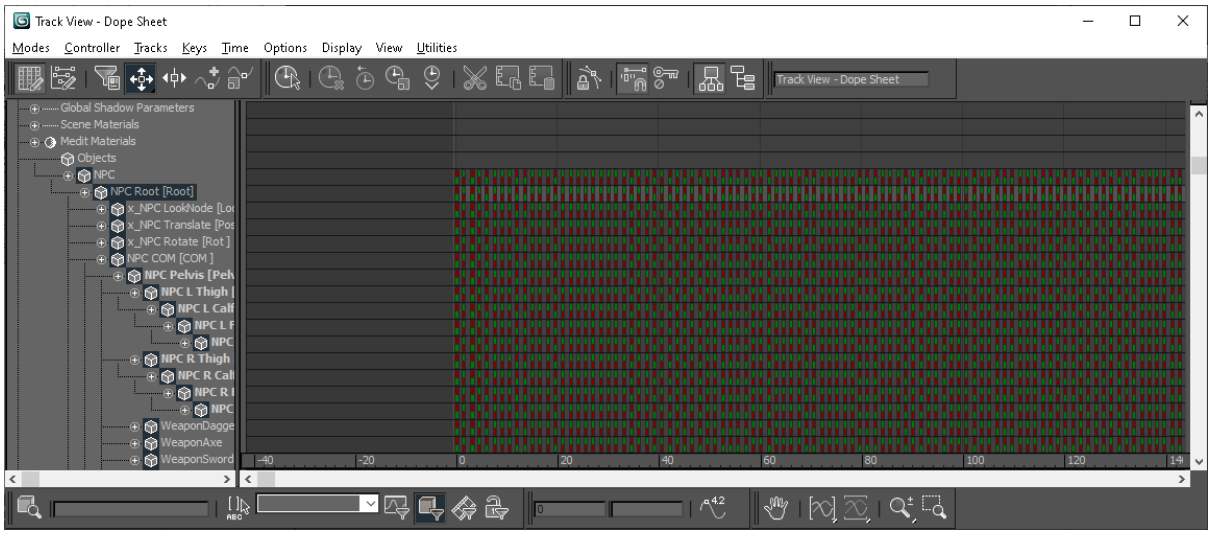


Figure 1279 - Keyframes added to the skeleton.

For exporting the animation, we'll first need to add a note track in the dope sheet on the first and last frames.

Firstly, click on NPC Root [Root] to select it.

Scrub the timeline back to the first frame.

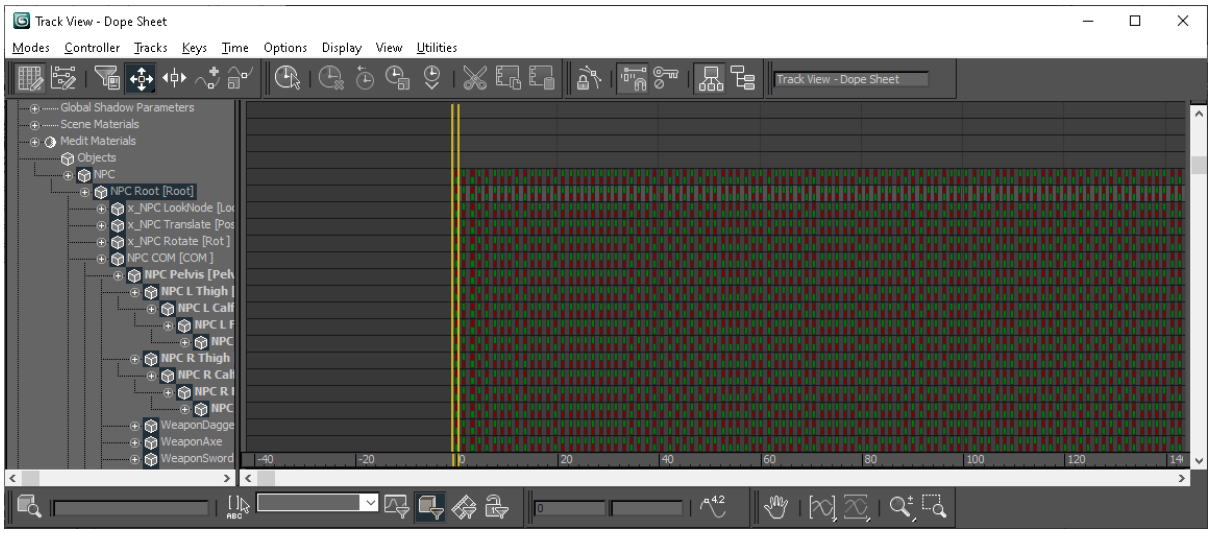


Figure 1280 - Selecting NPC Root [Root] in the dope sheet.

Go to Tracks > Note Track > Add.

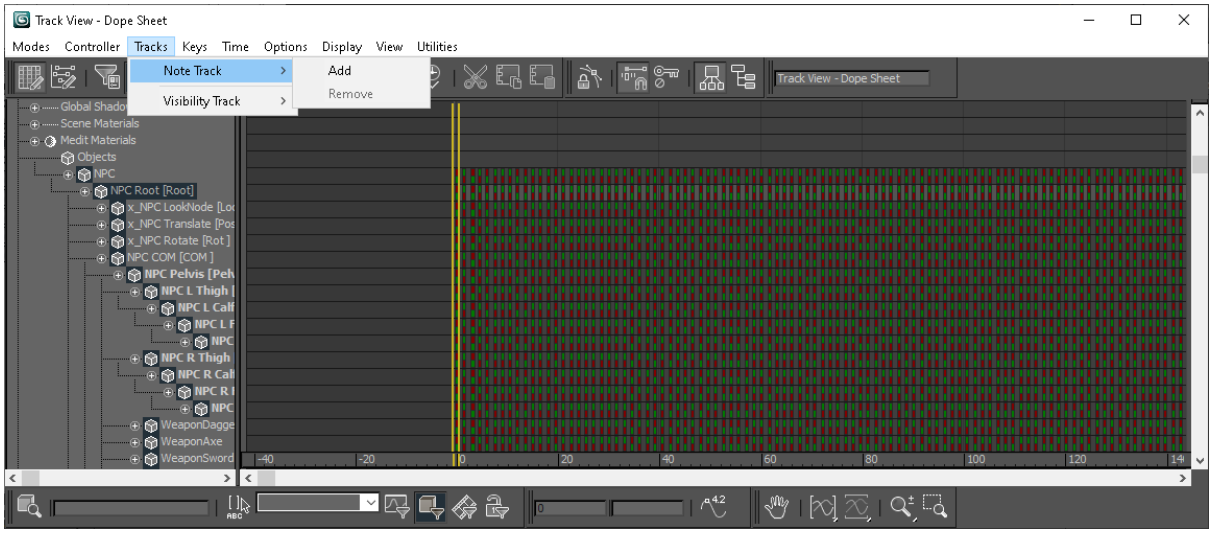


Figure 1281 - Adding a note track.

Go to Keys > AddKeys.

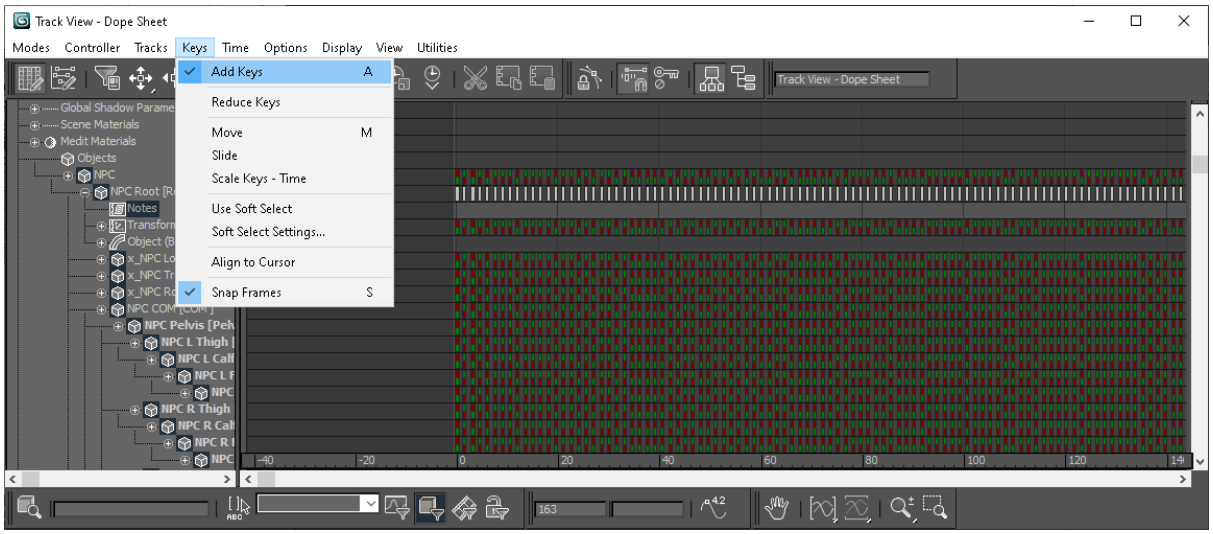


Figure 1282 - Adding a new key to the notes track.

Add a key to the first frame of the notes track and the last frame of the notes track. For this example, that's going to be frame 750.

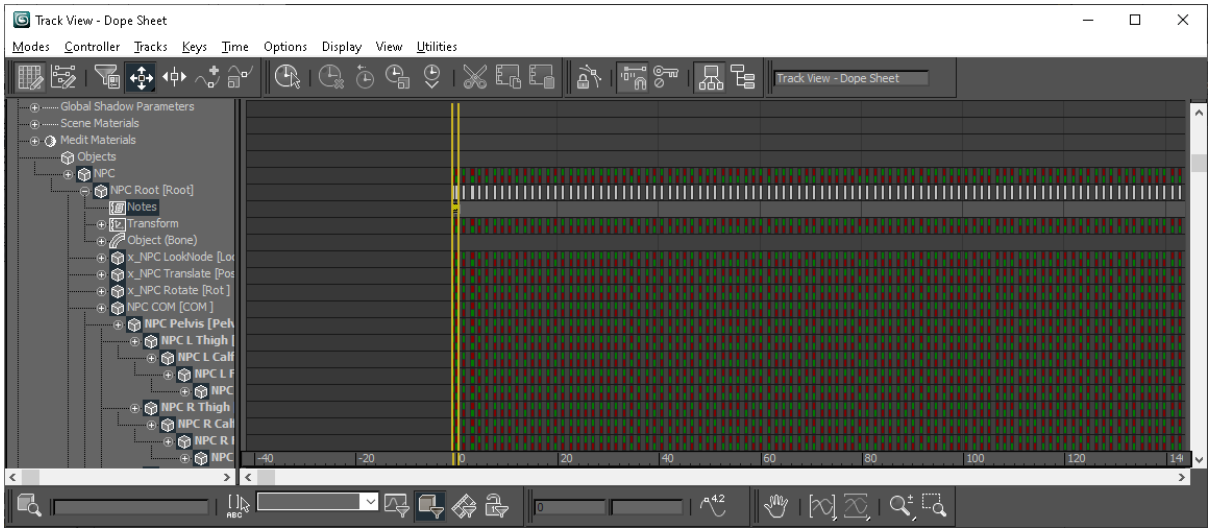


Figure 1283 - New key added to notes track on frame 0.

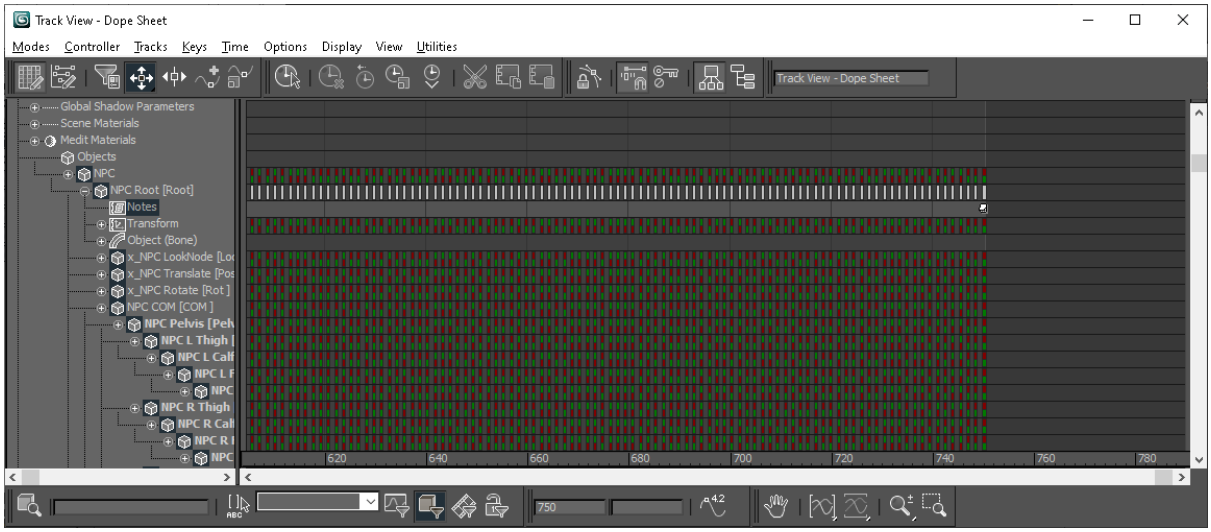


Figure 1284 - New key added to notes track on frame 750.

Right-click on the note on the first frame to open it.

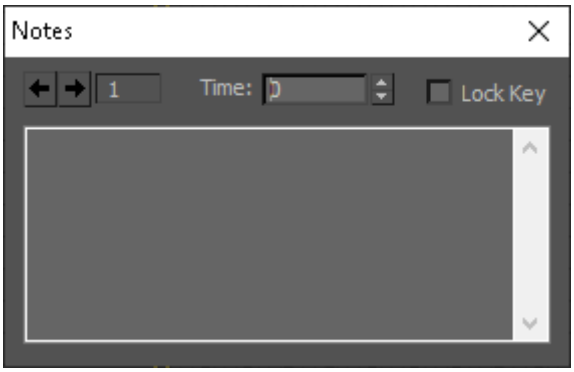


Figure 1285 - Editing the note on the first frame.

We need to add a specific command here:

```
start -name <NameOfYourAnimation> -at y
```

For this example, I'm just going to call the animation 'walkingaroundwaving'.

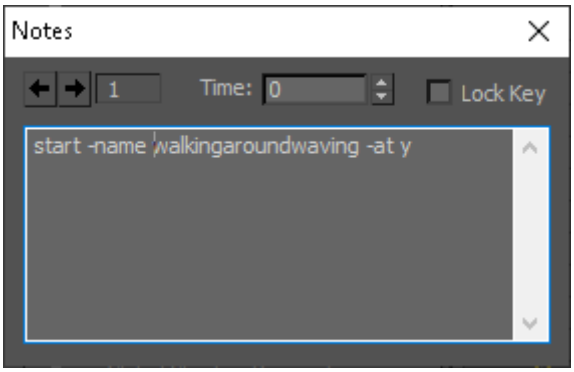


Figure 1286 - Command added to first note.

Add `end` in the last note like so:

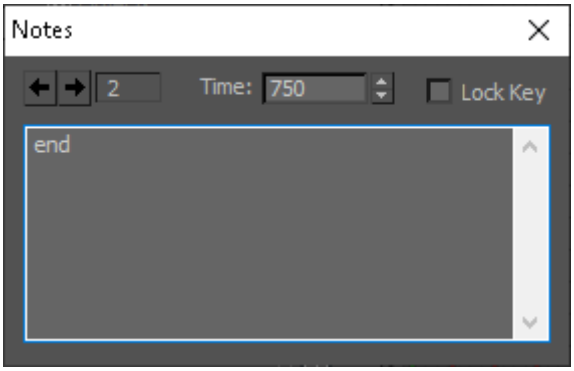


Figure 1287 - Command added to last note.

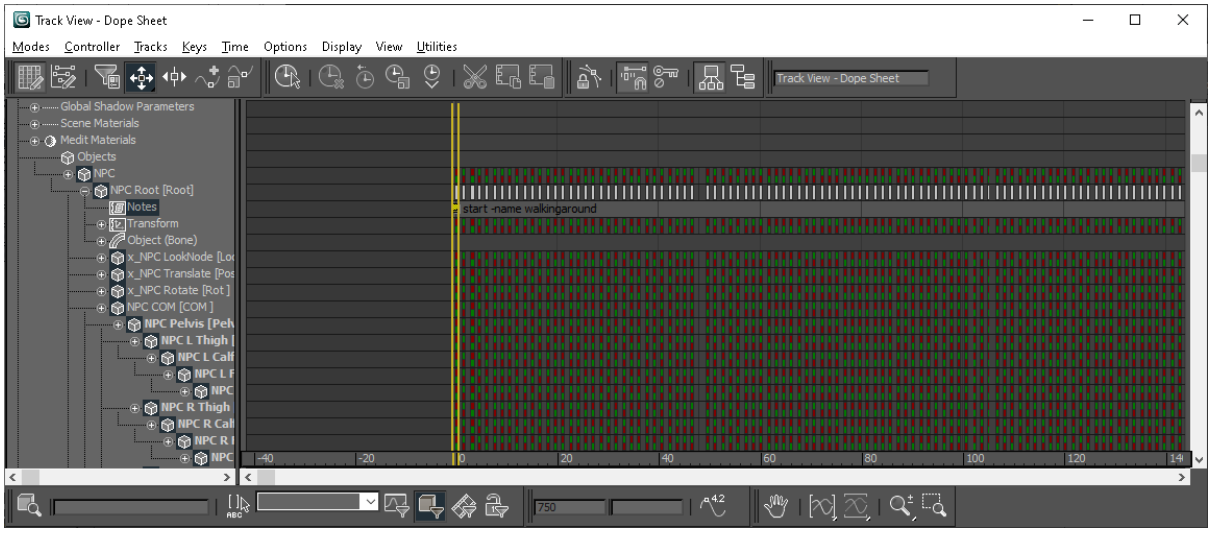


Figure 1288 - Notes added to dope sheet.

Close out of the dope sheet.

Go to Havok Content Tools > Export.

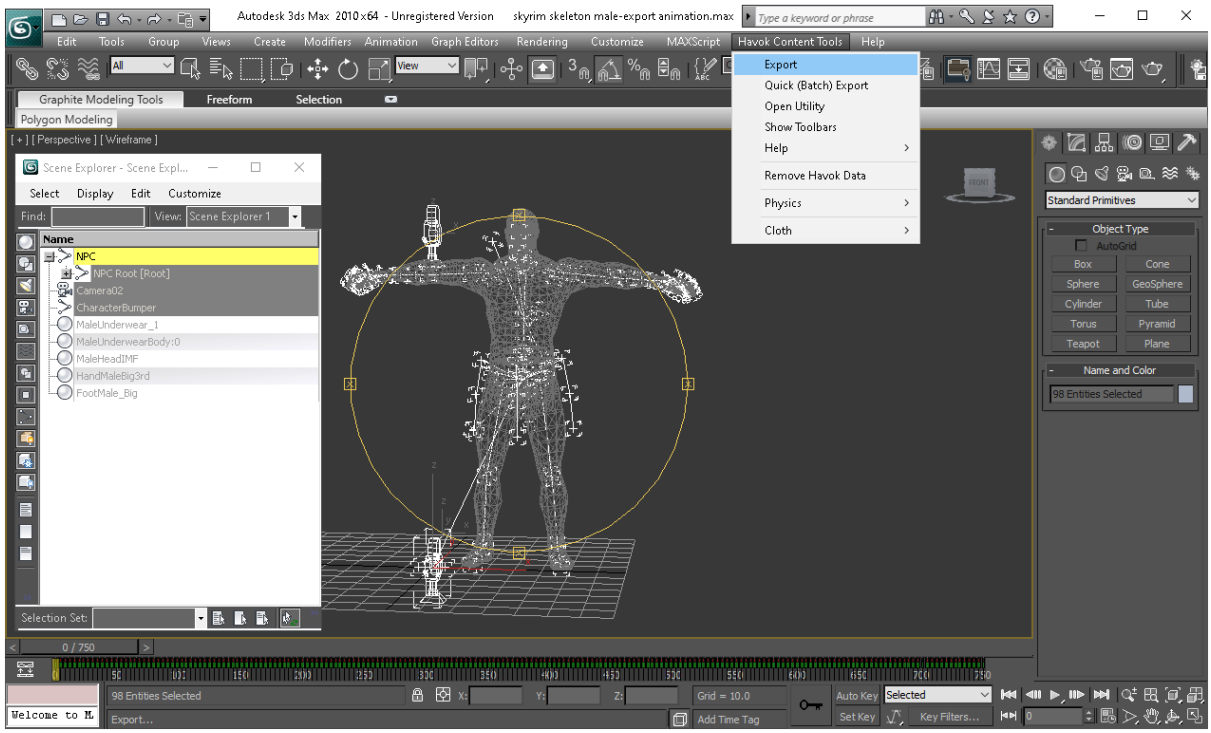


Figure 1289 - Opening the Havok Content Tools exporter.

Go to the Animation tab, click on Create Skeletons then click Add to add it to the current configuration set. Add Create Animations, Extract Motion, Spline Compression, Footstep Analysis and Loop Animations in that order.

Then go to the Core tab and add Prune Types and Write to Platform.

Set the name of the configuration set to Animations.

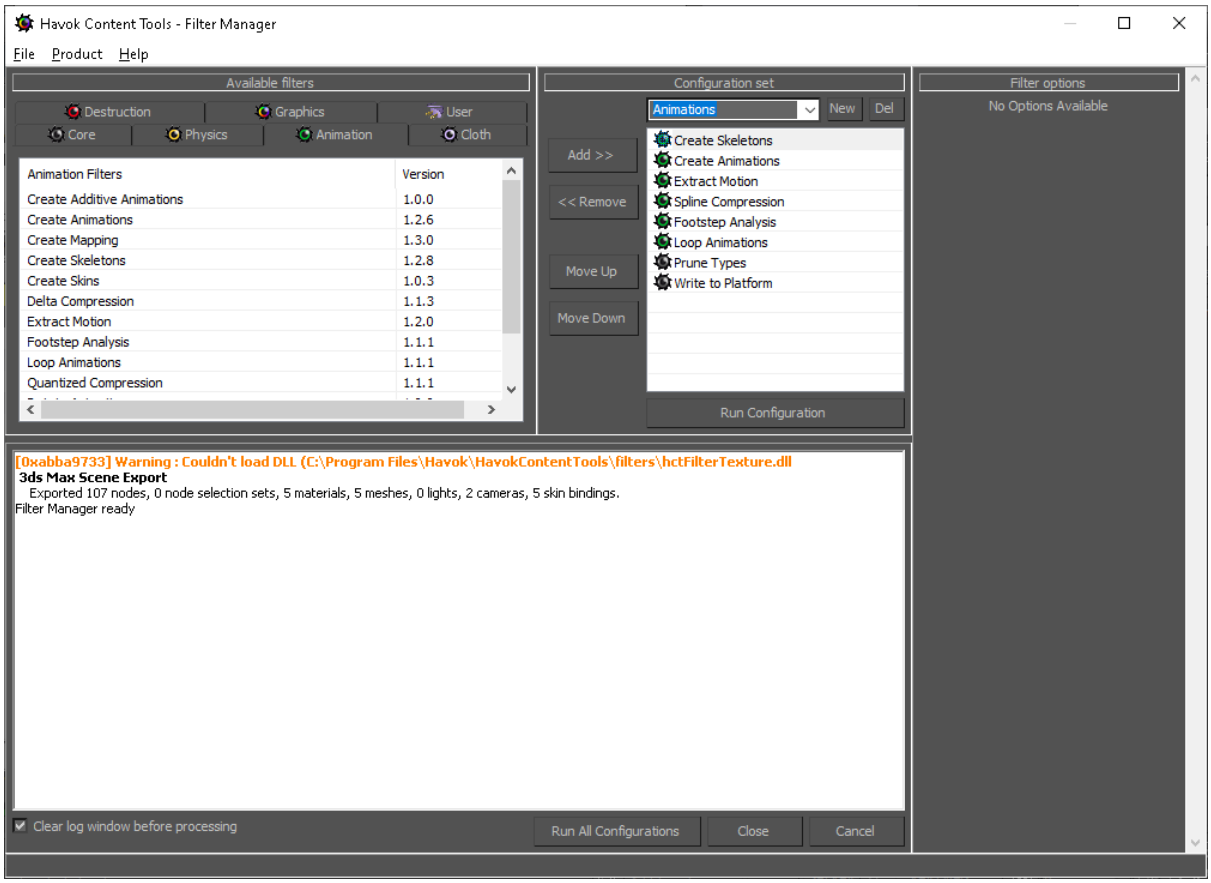


Figure 1290 - Adding tasks to the current configuration set.

First let's start with Create Skeletons.

Under Build Rig, set it to 'From File' and tick 'Use File Order'. Set the file to Skyrim3rdPersonRig.txt. This file came with 3ds Max 2010_Skyrim_Vanilla Skeleton.max.

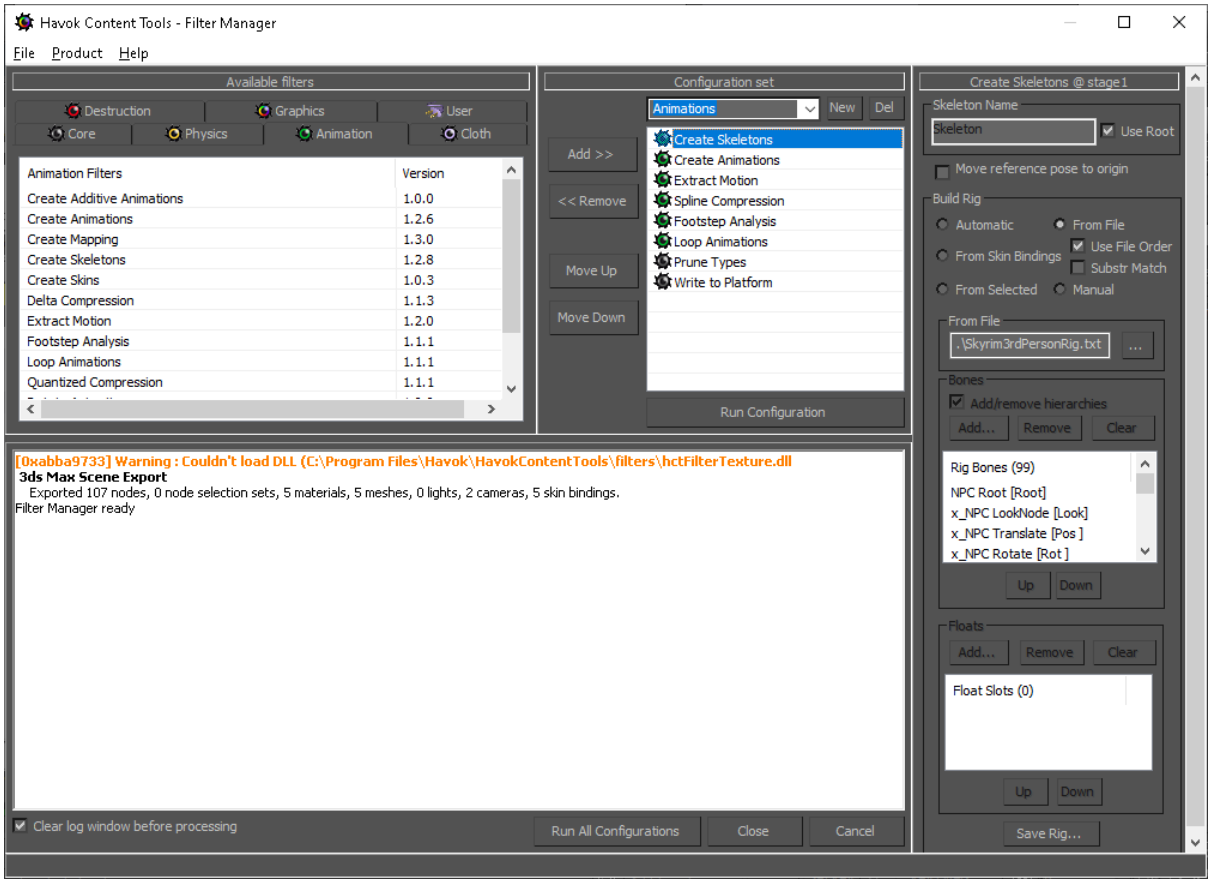


Figure 1291 - Create Skeletons.

Next, under Create Animations, untick X, Y and Z next to Move To Origin. Ensure the first frame is set to 0 and the last frame is set correctly. In my example, the last frame is 750.

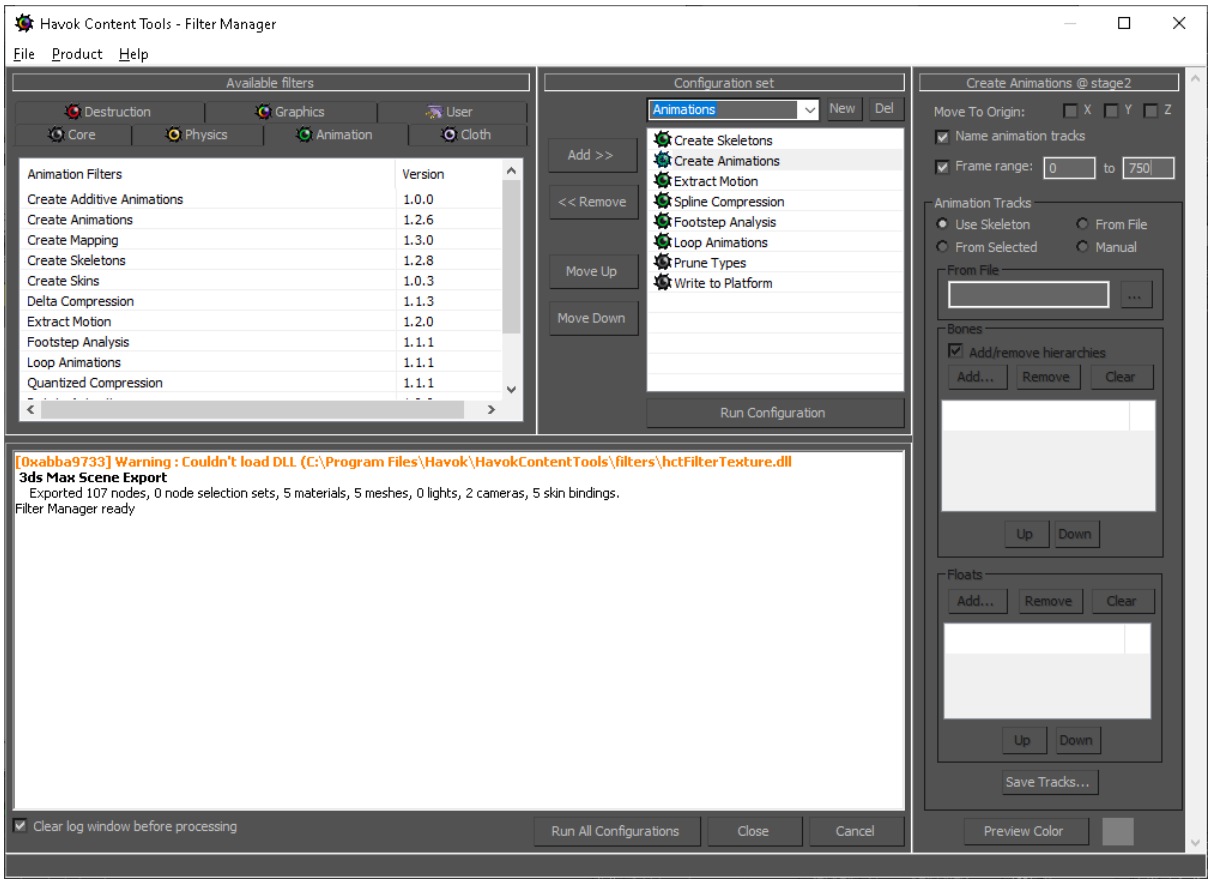


Figure 1292 - Create Animations.

Ensure Extract Motion is set up as per the screenshot below. We shouldn't need to change anything.

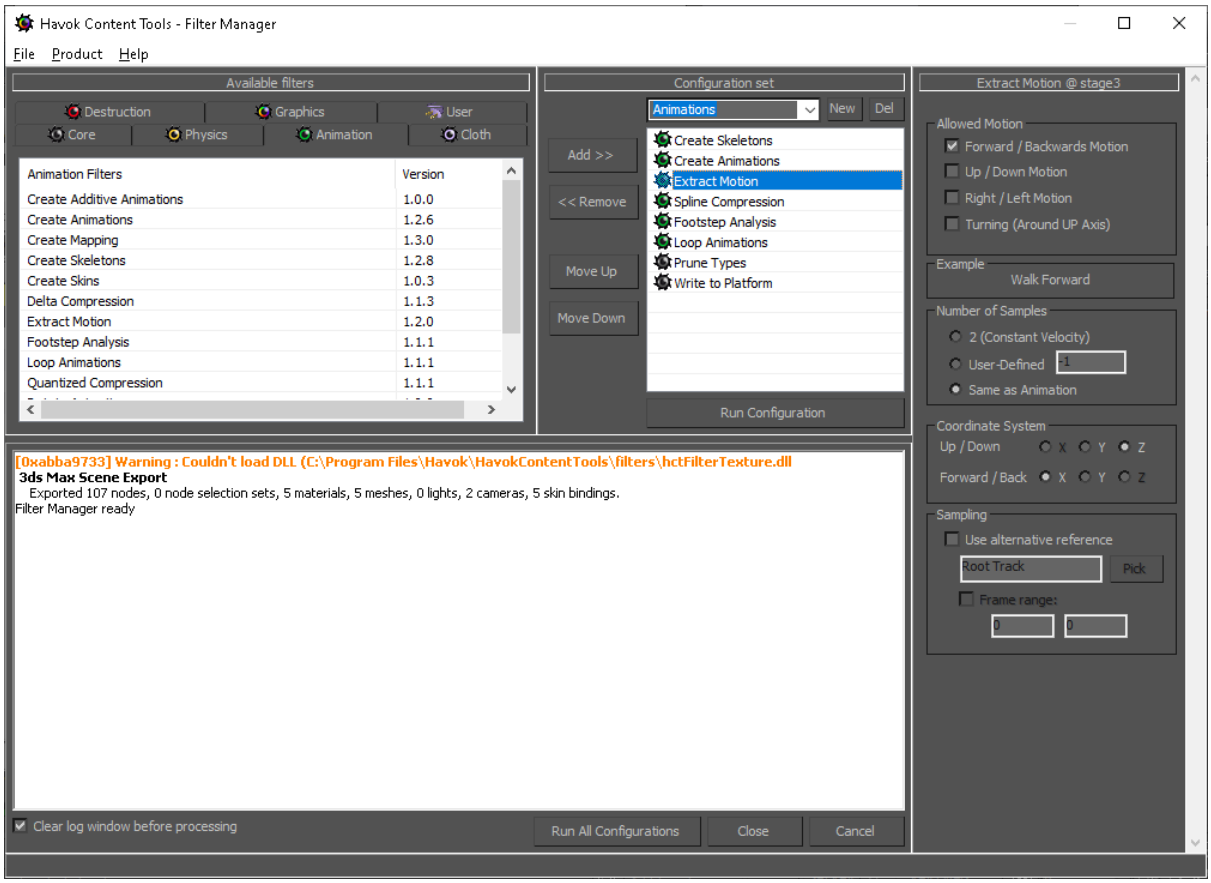


Figure 1293 - Extract Motion.

Spline Compression is next. Make sure the Quality slider is set to High [Lossless].

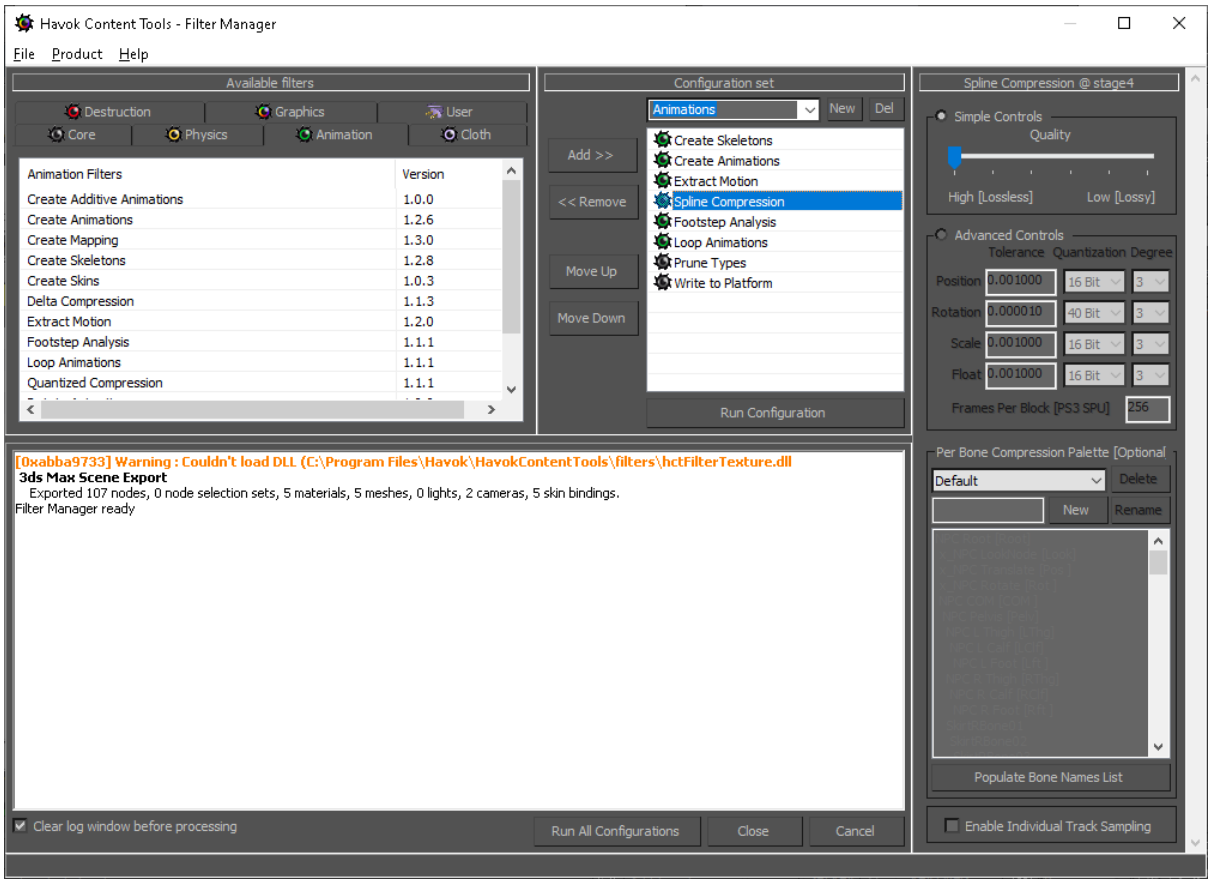


Figure 1294 - Spline Compression.

We shouldn't need to change anything under Footstep Analysis.

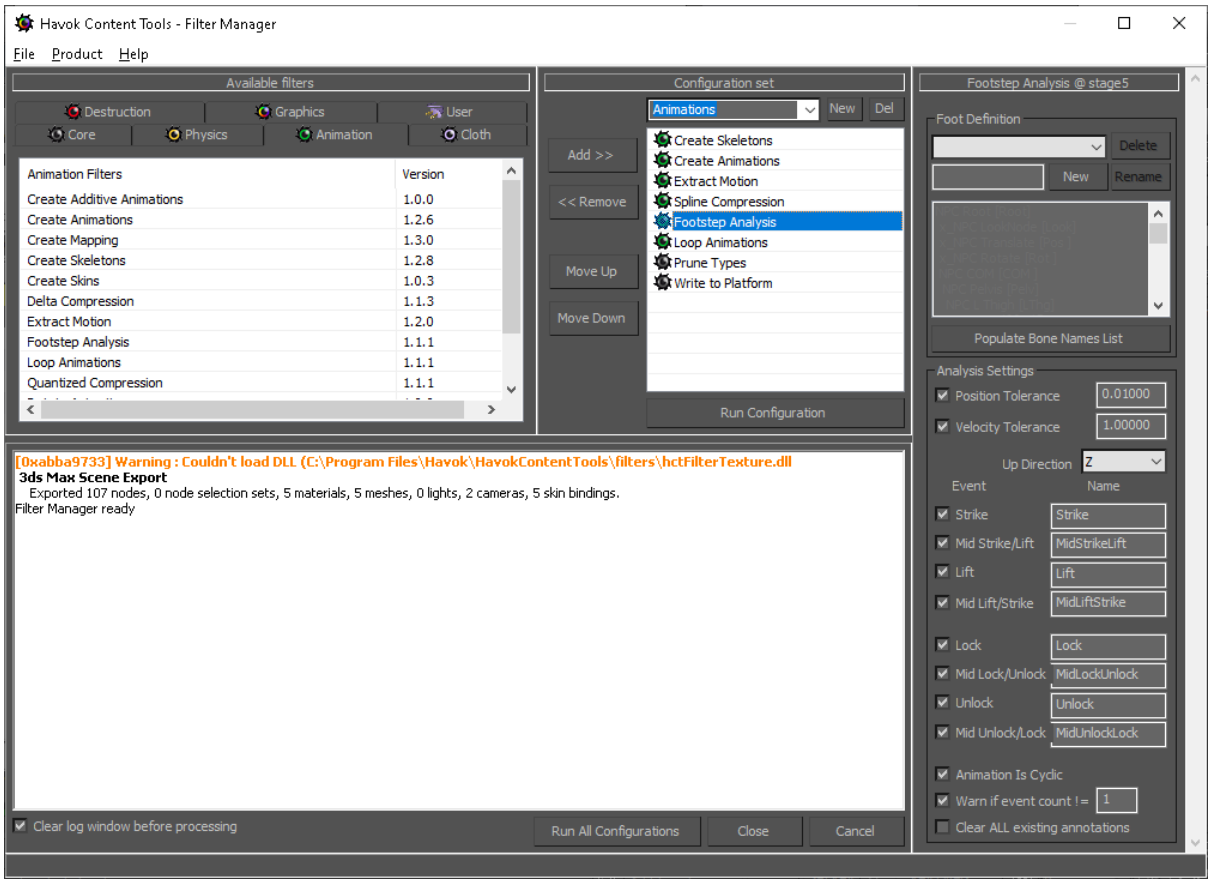


Figure 1295 - Footstep Analysis.

For this example, set the Number of Frames field to one frame after the last frame of your animation. E.g.: 751.

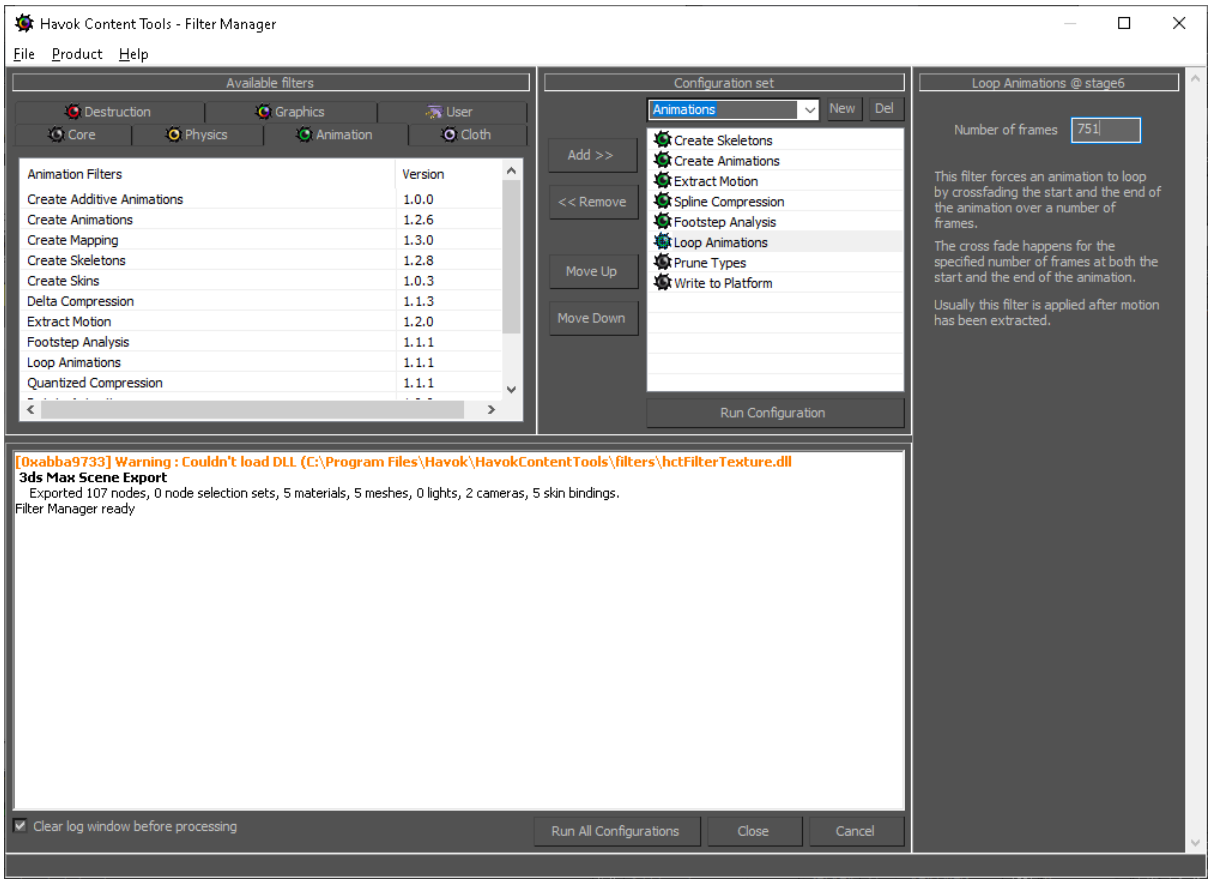


Figure 1296 - Loop Animations.

Go to Prune Types. Under Scene, tick 'All Scene Data'. Under Animation, tick Skeletons, Bone Attachments, Identity Binding Indices, Mesh Bindings and Ragdoll and Mappers. Under Environment, tick Environment Data and under Resource Data tick All Resource Data.

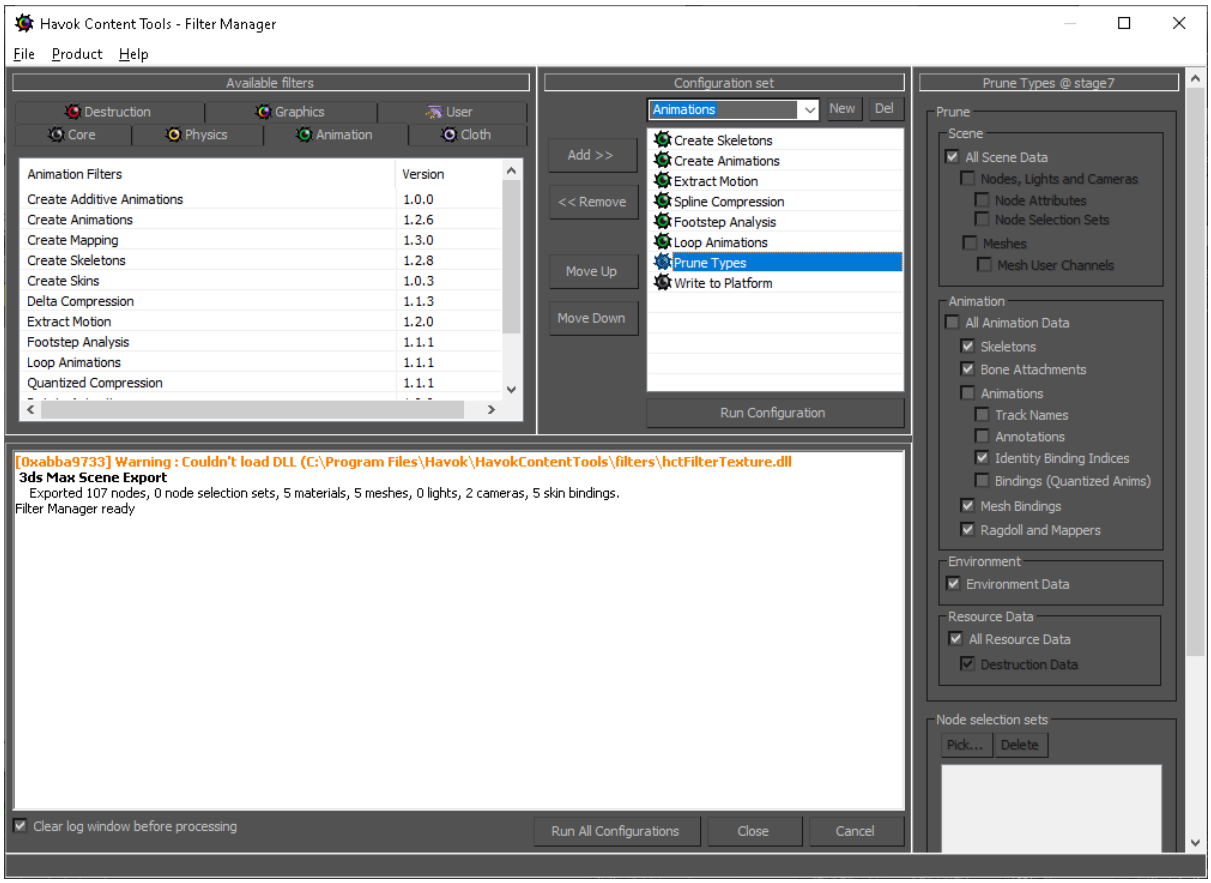


Figure 1297 - Prune Types.

Finally, under Write to Platform, set the Format to Packfile, tick Remove Metadata and click on the '…' button to set the file export location.

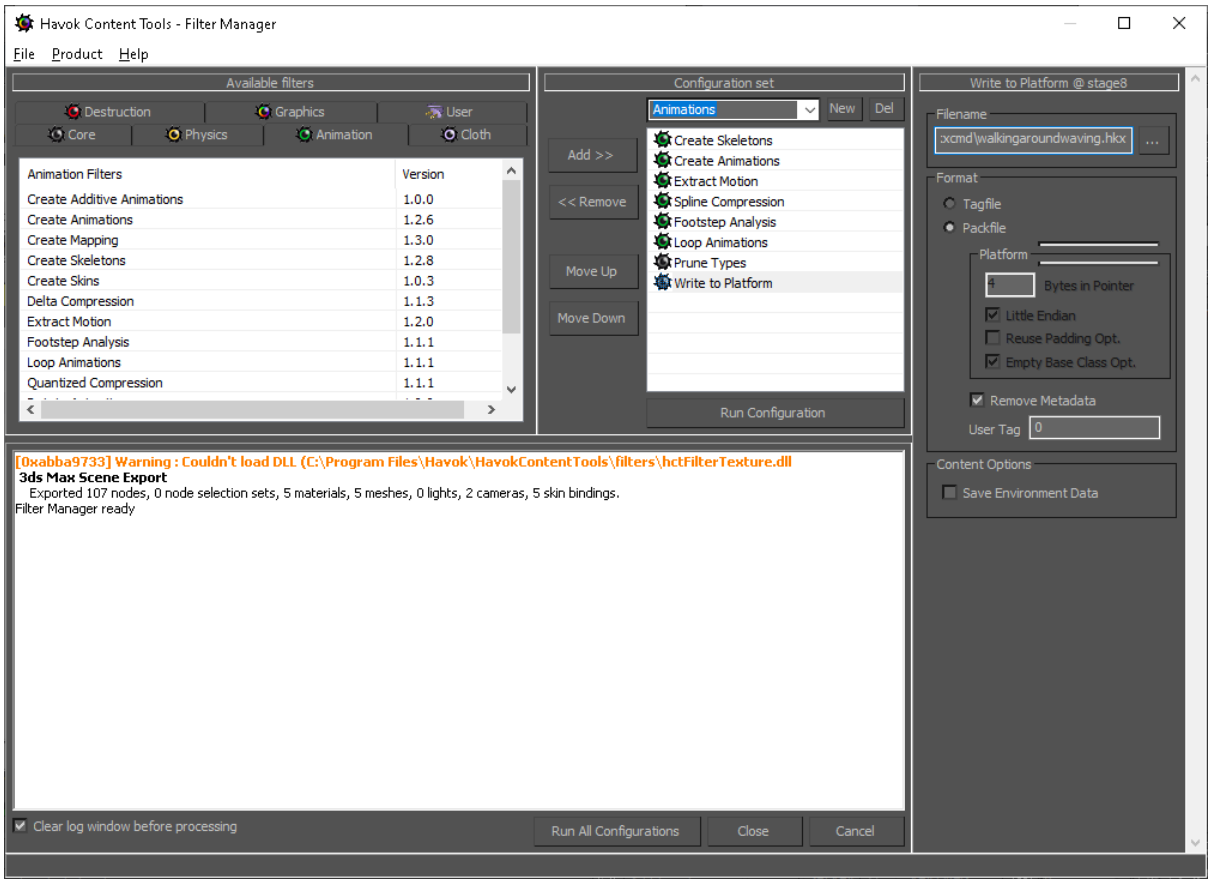


Figure 1298 - Write to Platform.

Ensure you're saving the file as Binary Packfile (*.hkx).

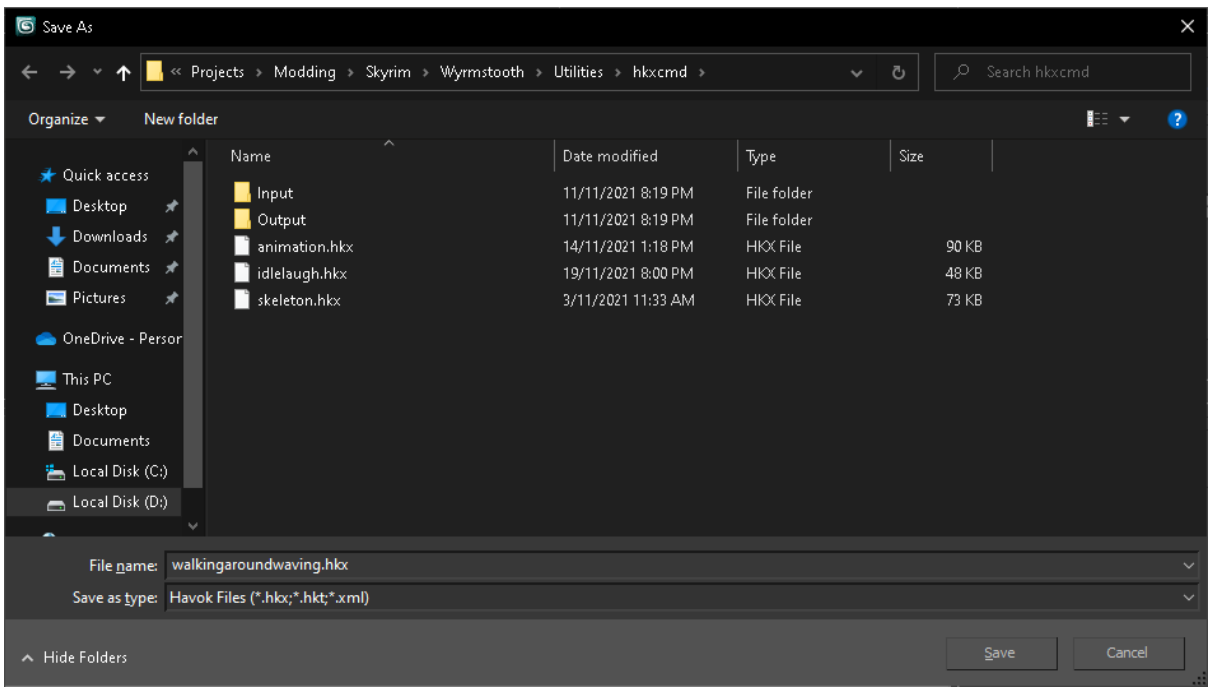


Figure 1299 - File export location.

To begin the export process, click Run Configuration.

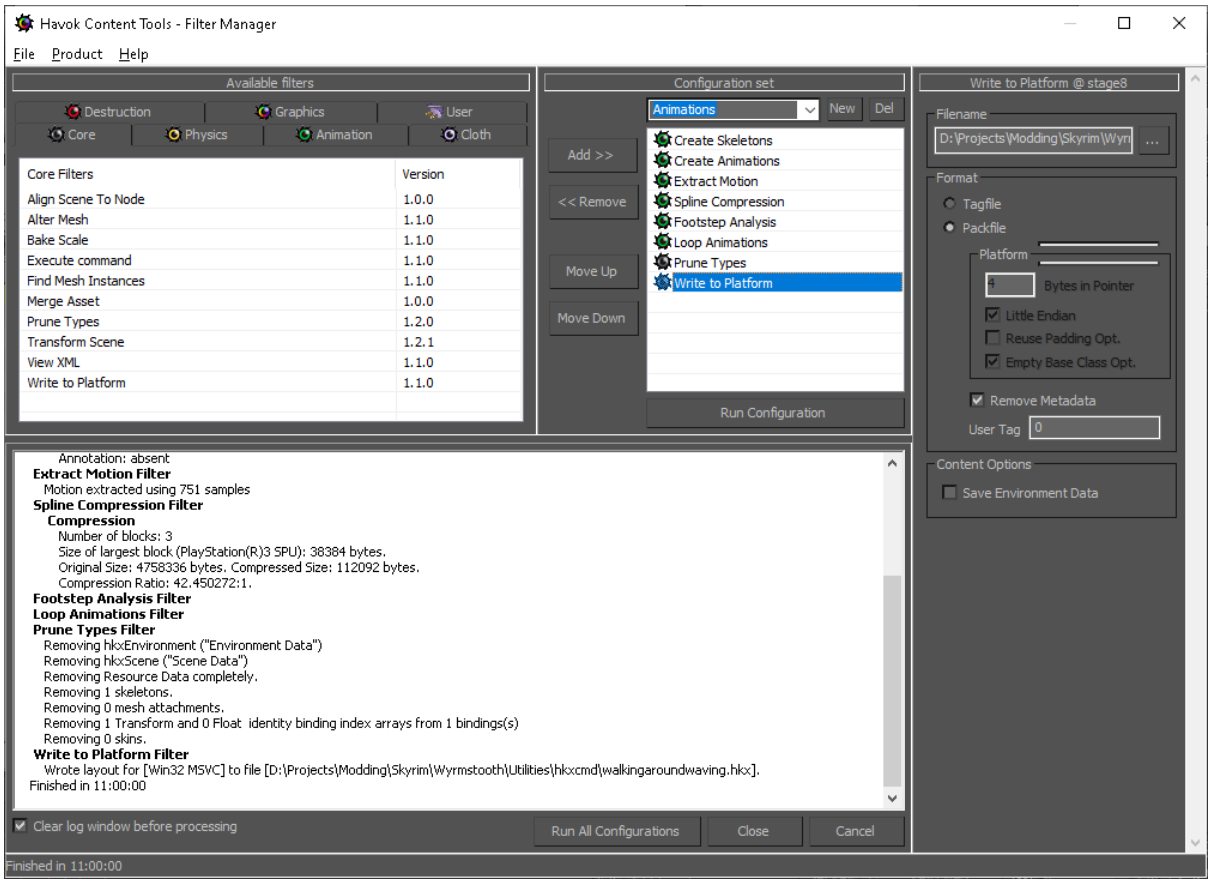


Figure 1300 - Run Configuration.

Browse to the folder you saved the .hkx file to and confirm it has been created successfully.

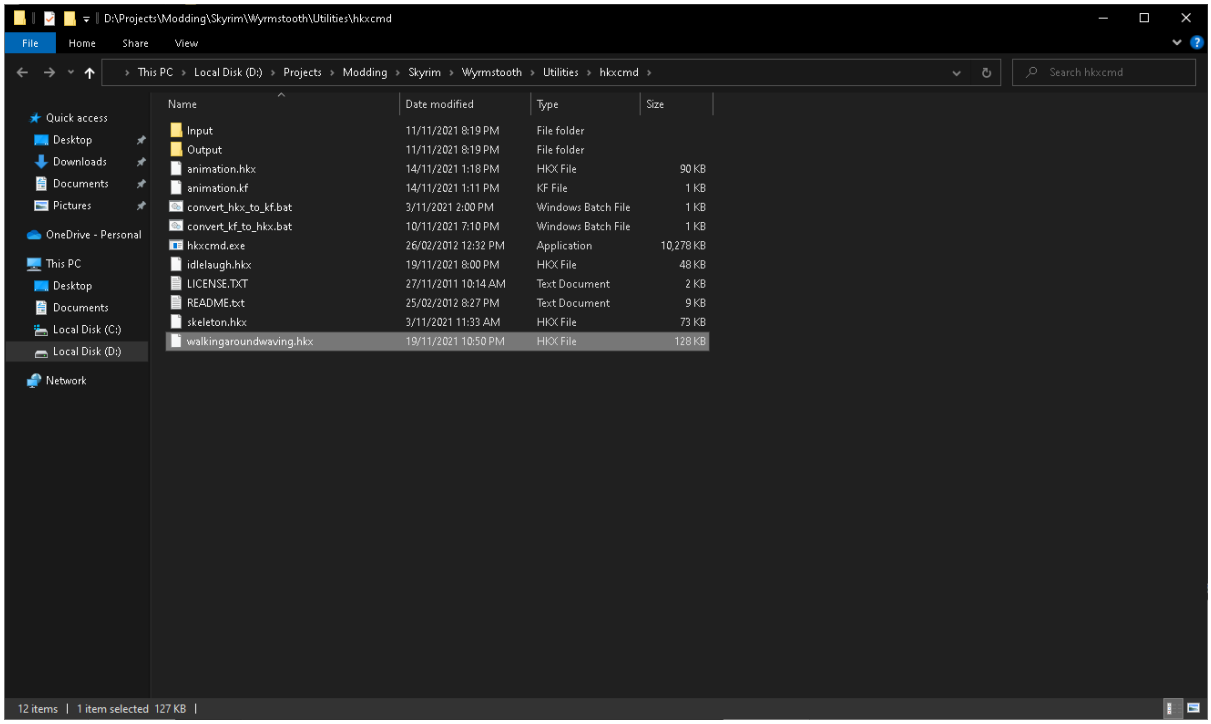


Figure 1301 - Confirming the .hkx file has been exported.

Copy the file to your Skyrim\Data\meshes\actors\character\animations or Skyrim Special Edition\Data\meshes\actors\character\animations folder.

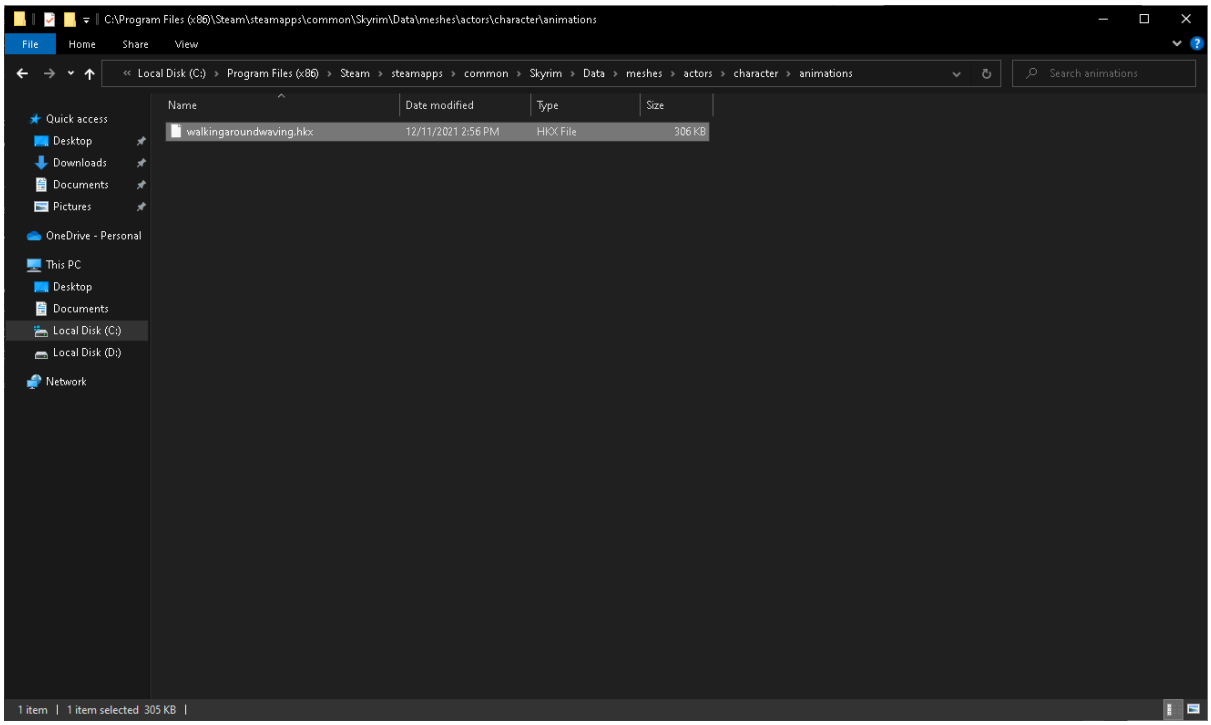


Figure 1302 - .hkx file copied to the Skyrim data folder.

For now, I'm just going to test out this animation in-game by replacing an existing animation temporarily.

To replace an existing animation, simply name it after an existing .hkx file. In this example, I'm going to rename walkingaroundwaving.hkx to idlelaugh.hkx.

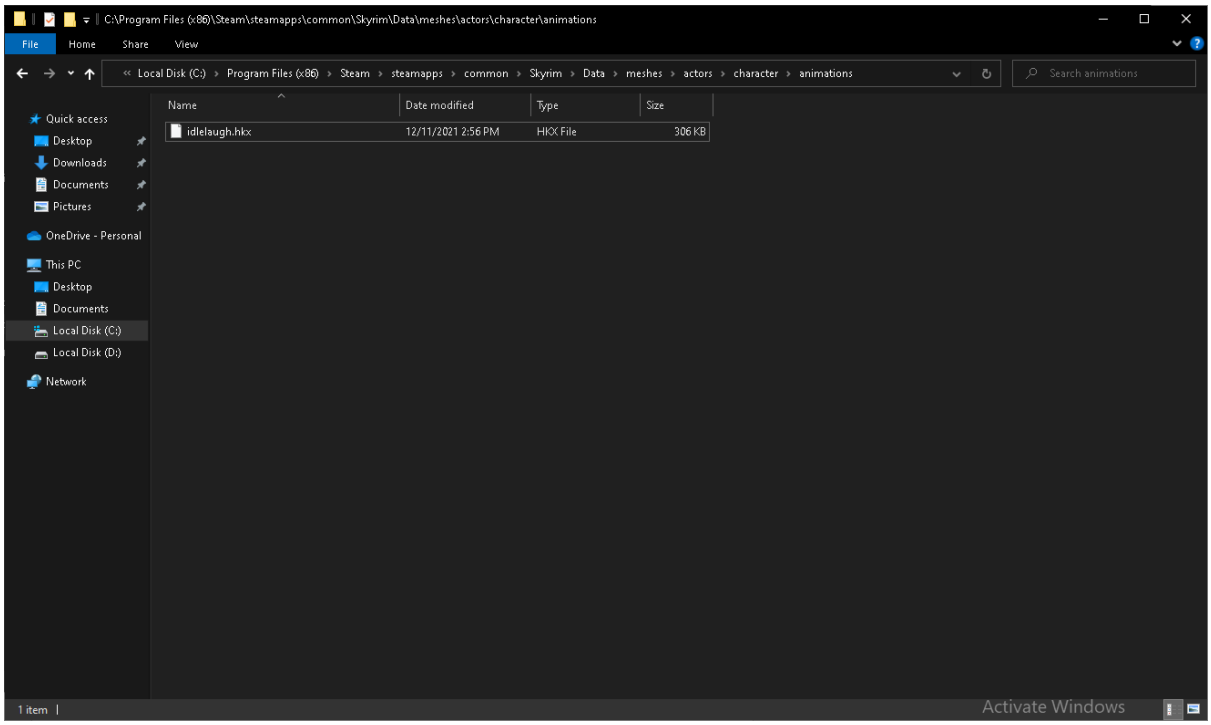


Figure 1303 - .hkx file renamed.

In the Creation Kit, go to Miscellaneous > IdleMarker in the Object Window. Right-click on one of the existing idle animations and select New.

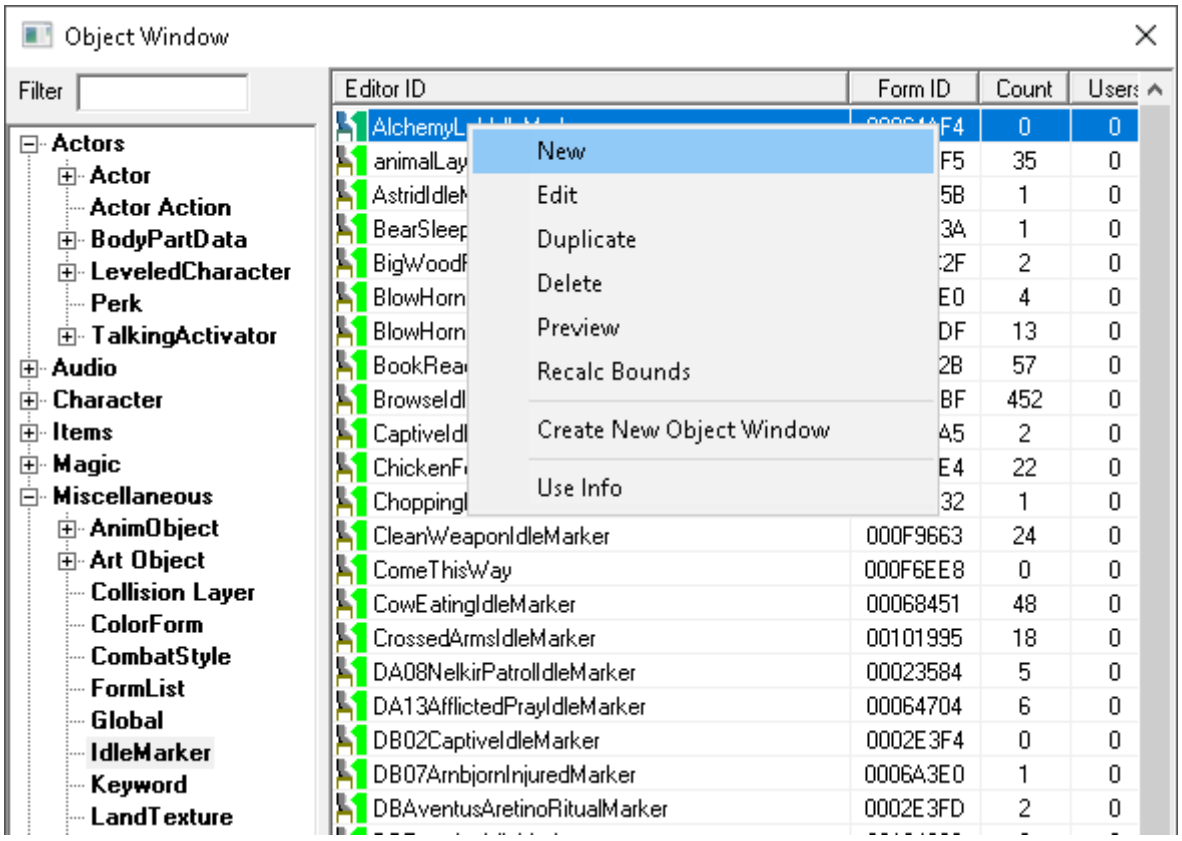


Figure 1304 - Creating a new idle marker.

Set an ID in the ID field. For this quick example I'm just going to set it to WTIdleLaugh.

Since we replaced IdleLaugh I added it to the list of idle animations.

For more guidance on setting up idle markers, see Adding Idle Markers.

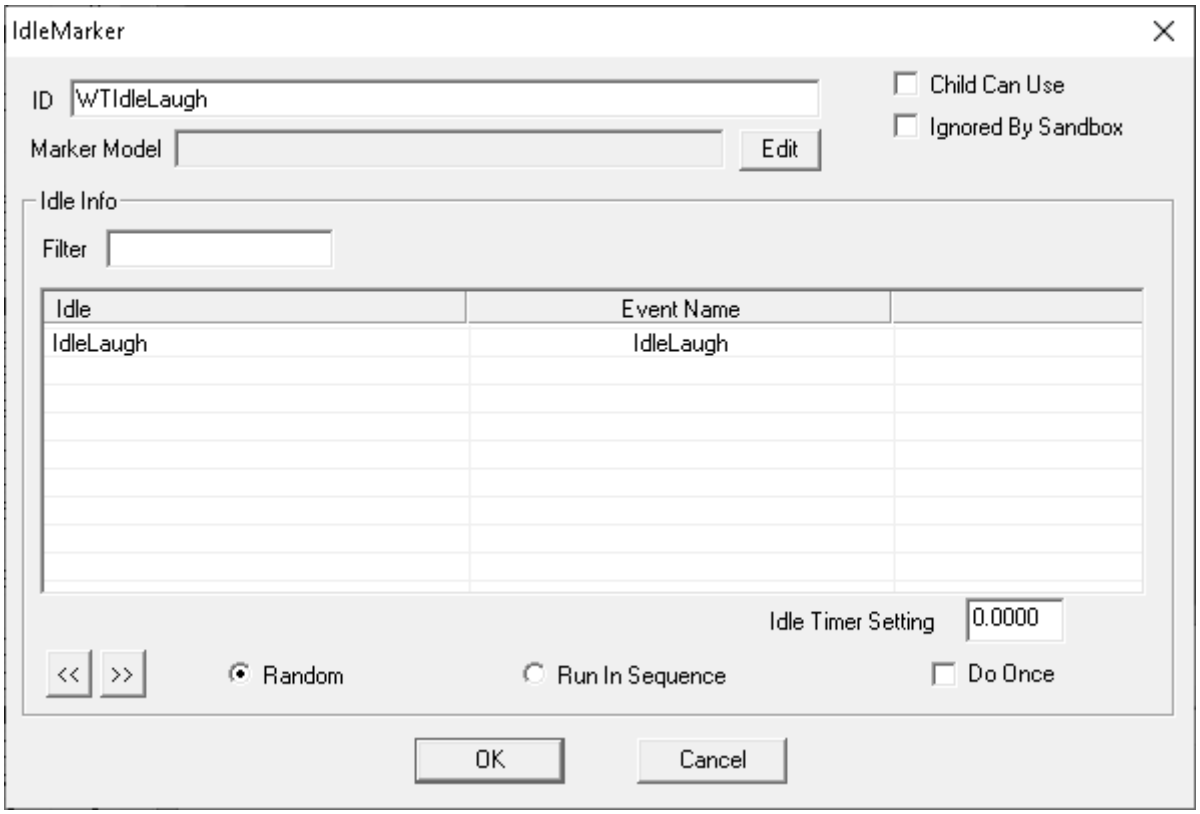


Figure 1305 - IdleMarker setup.

To test things out, I just added an NPC linked to the WTIdleLaugh idle marker and added a COCMarkerHeading marker so when I use console commands to teleport into the cell, I'll be able to see the animation as soon as it starts.

I also added a simple navmesh so the NPC can walk over to the marker.



Figure 1306 - An NPC linked to the WTIdleLaugh marker.

And there we have it, the animation plays successfully in-game.



Figure 1307 - Motion capture animation playing in Skyrim.

CHAPTER 10: SPEECH SYNTHESIS

INTRODUCTION TO SPEECH SYNTHESIS

Speech synthesis can be used to add additional lines of dialogue to NPCs that use base game voice types.

In Wyrmstooth, I use speech synthesis to give background NPCs, such as the hunters at Hunter's Shack or the marauder leader at Cragwater Camp, unique things to say.

In this section I'll be showing you how to train a custom Tacotron and WaveGlow model on the Google Colab platform using a dataset based on a voice type from The Elder Scrolls V: Skyrim.

Tacotron is a generative text-to-speech synthesis program. For this tutorial I'll be using the NVIDIA Tacotron 2 repository available on GitHub: https://github.com/NVIDIA/tacotron2.

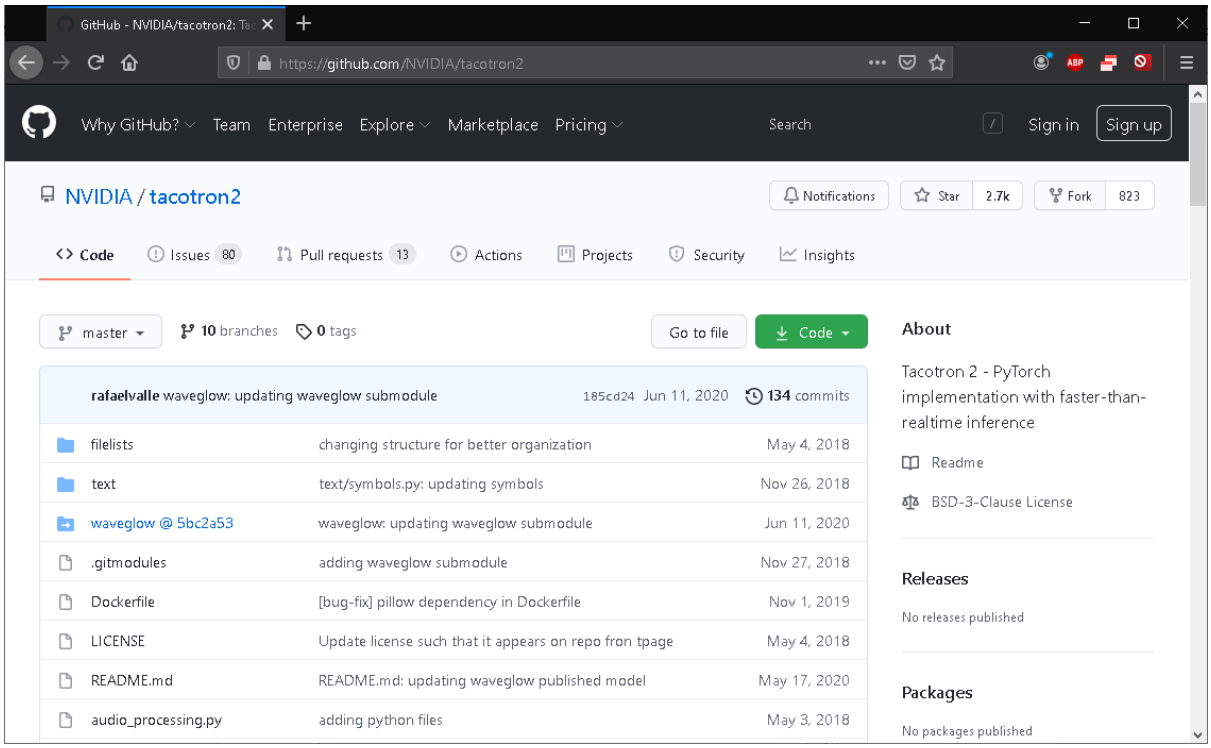


Figure 1308 - NVIDIA Tacotron 2 on GitHub.

WaveGlow is a flow-based generative speech-synthesis program. A custom WaveGlow model will significantly improve the quality of synthesized speech. For this tutorial I'll be using the NVIDIA WaveGlow repository available on GitHub: https://github.com/NVIDIA/waveglow.

Google Colab is a cloud platform that I'll be using in this tutorial to run Tacotron and WaveGlow. It can be used to run python code and other commands in a pre-built python environment via your browser. We can link our Google Drive storage to a Colab session so we can access our dataset and save checkpoints back to it as we train.

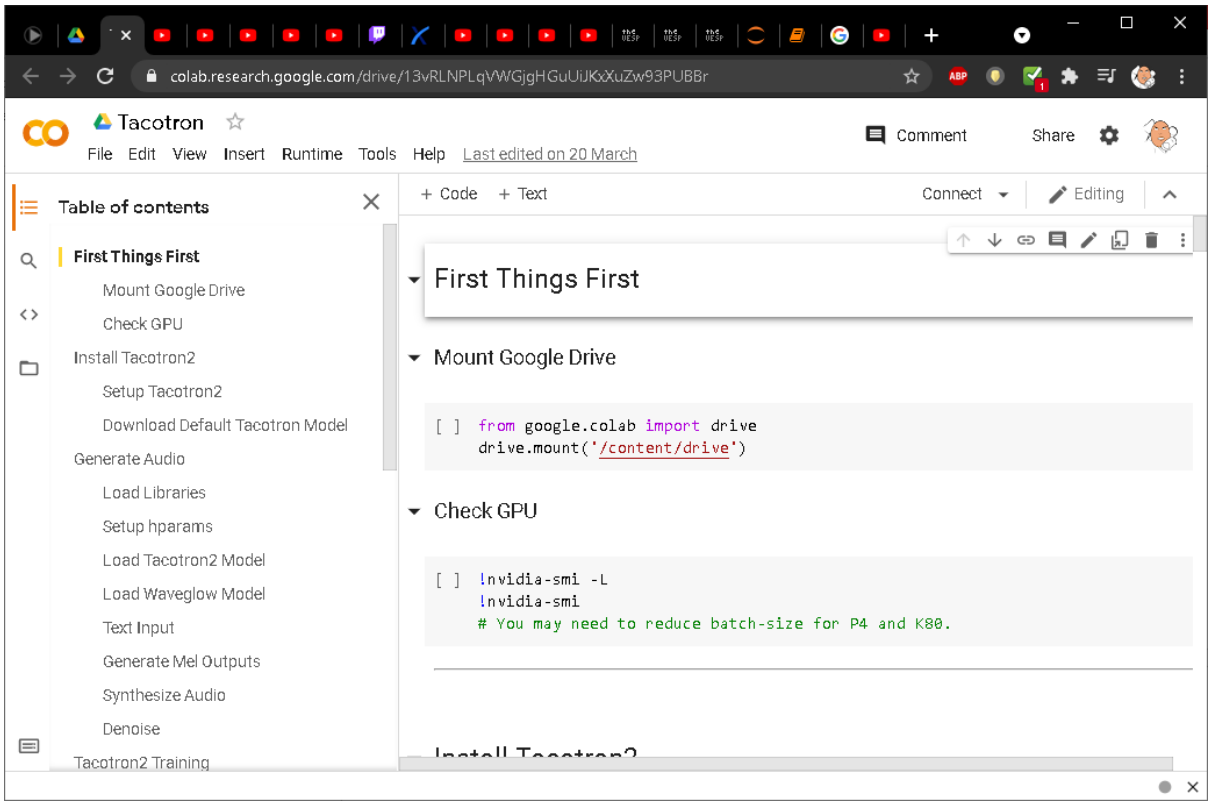


Figure 1309 - A Colab Notebook.

I've already set up a Colab Notebook with all the commands you'll need to run. You can access it here:

https://colab.research.google.com/drive/13vRLNPLqVWGjgHGuUiJKxXuZw93PUBBr?usp=sharing.

The main advantage of using Google Colab is that we don't need to set up our own python environment locally.

Here is some sample output from a few different models I've trained:

tacotron_sample_femalenord1.wav

tacotron_sample_femalenord2.wav

tacotron_sample_femalenord3.wav

tacotron_sample_joerogan1.wav

tacotron_sample_joerogan2.wav

tacotron_sample_joerogan3.wav

tacotron_sample_femalecommander1.wav

tacotron_sample_femalecommander2.wav

tacotron_sample_femalecommander3.wav

tacotron_sample_malecommander1.wav

tacotron_sample_malecommander2.wav

tacotron_sample_malecommander3.wav

tacotron_sample_maleslycynical1.wav

tacotron_sample_maleslycynical2.wav

tacotron_sample_maleslycynical3.wav

As you can hear, Tacotron is able to pick up on various vocal nuances like the rolling r's of the femalenord voice actor's accent.

PREPARING A DATASET USING VOICE ACTING FROM THE ELDER SCROLLS V: SKYRIM

In this section I'll be showing you how to prepare a new dataset from which we'll be training a new Tacotron and WaveGlow model.

A dataset consists of voice clips from a single speaker and their corresponding subtitles. Generally speaking, the more voice acting we have the better our results should be. Models based on datasets consisting of less than 3 hours of audio are going to struggle with articulation.

The first thing we need to do is extract the subtitle files from The Elder Scrolls V: Skyrim.

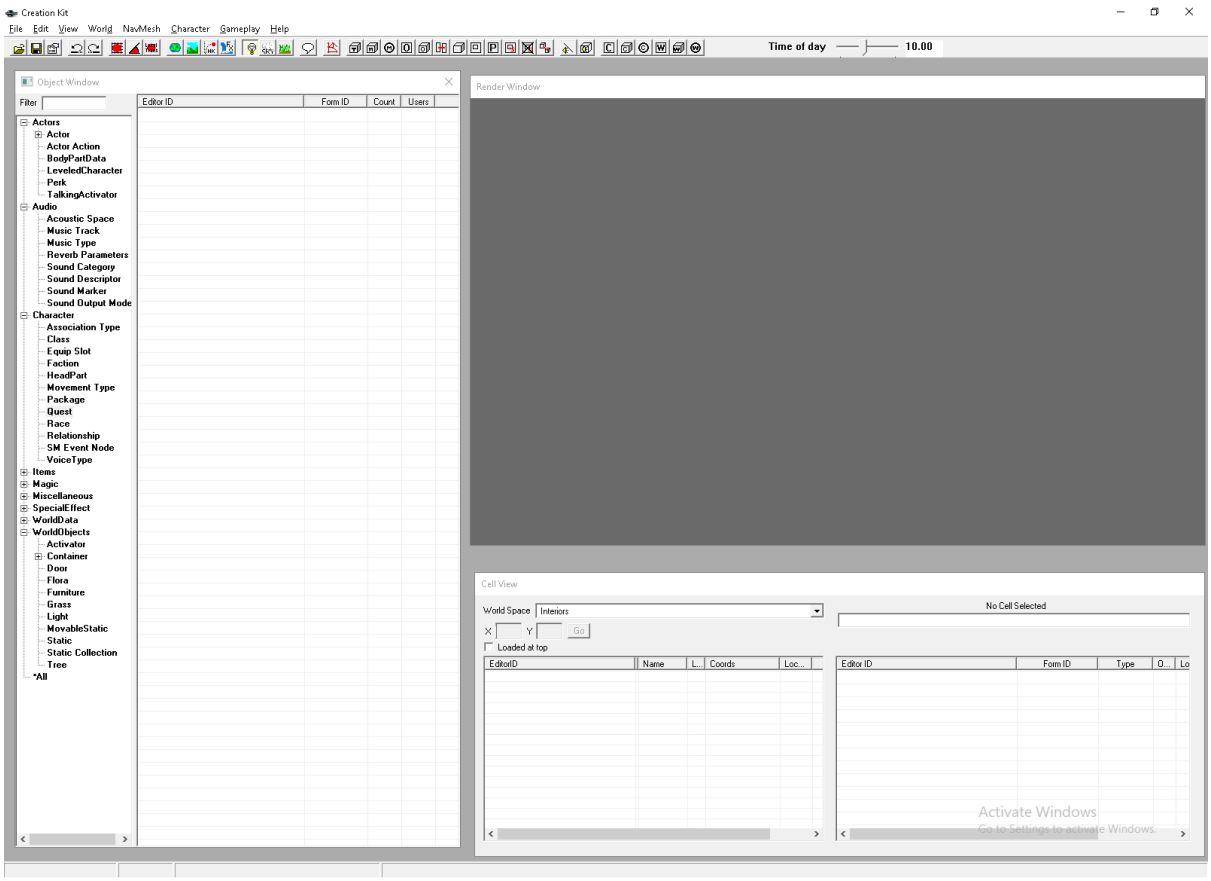


Figure 1310 - Creation Kit.

Once the Creation Kit loads, go to File > Data. Double-click on 'Skyrim.esm' then click OK and wait for it to load.

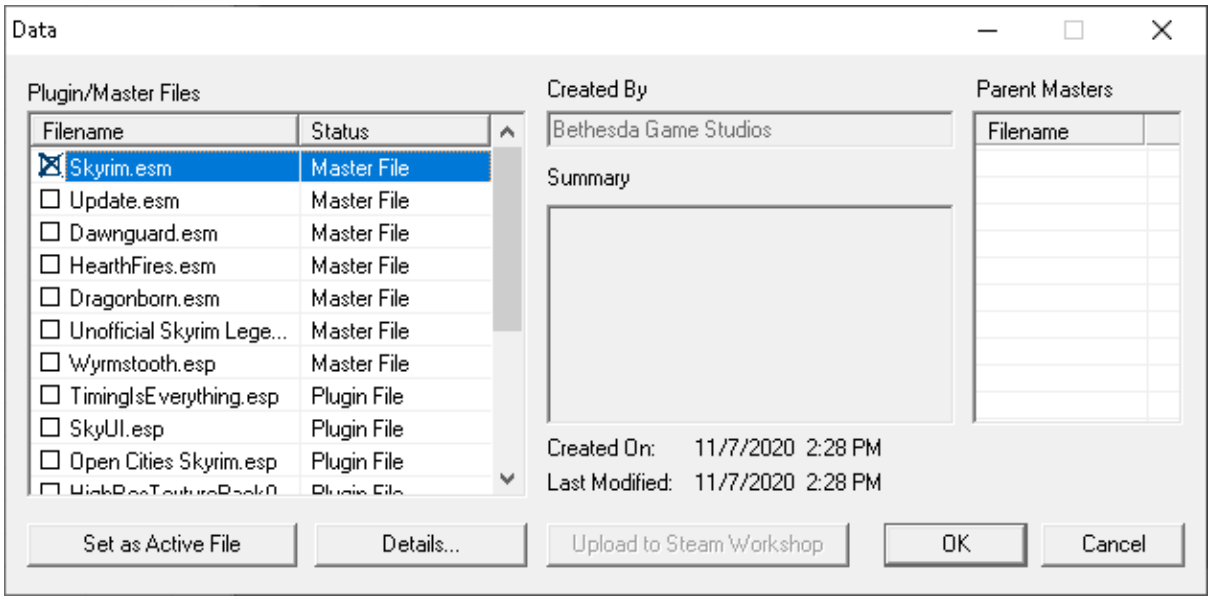


Figure 1311 - Loading Skyrim.esm.

Click 'Yes to All' to any warnings that pop up.

Go to Character > Export Dialogue and click OK to the Export Dialogue popup window.

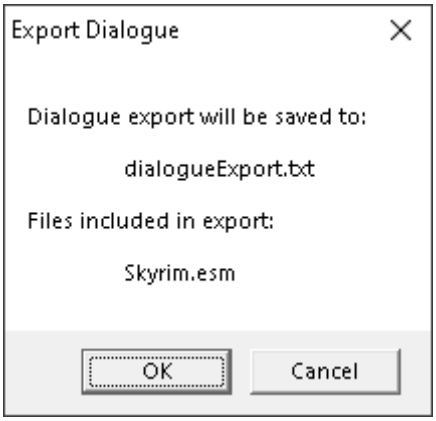


Figure 1312 - Exporting dialogue.

This will create a dialogueExport.txt file in your Skyrim installation folder.

Again, click 'Yes to All' to any warnings that pop up. Close the Creation Kit once the export has finished.

For this next part we'll need **Notepad++** and **Microsoft Excel**, or a similar spreadsheet application. You can download Notepad++ for free from its website: https://notepad-plus-plus.org/downloads/.

Open dialogueExport.txt in Notepad++. It should look like this:

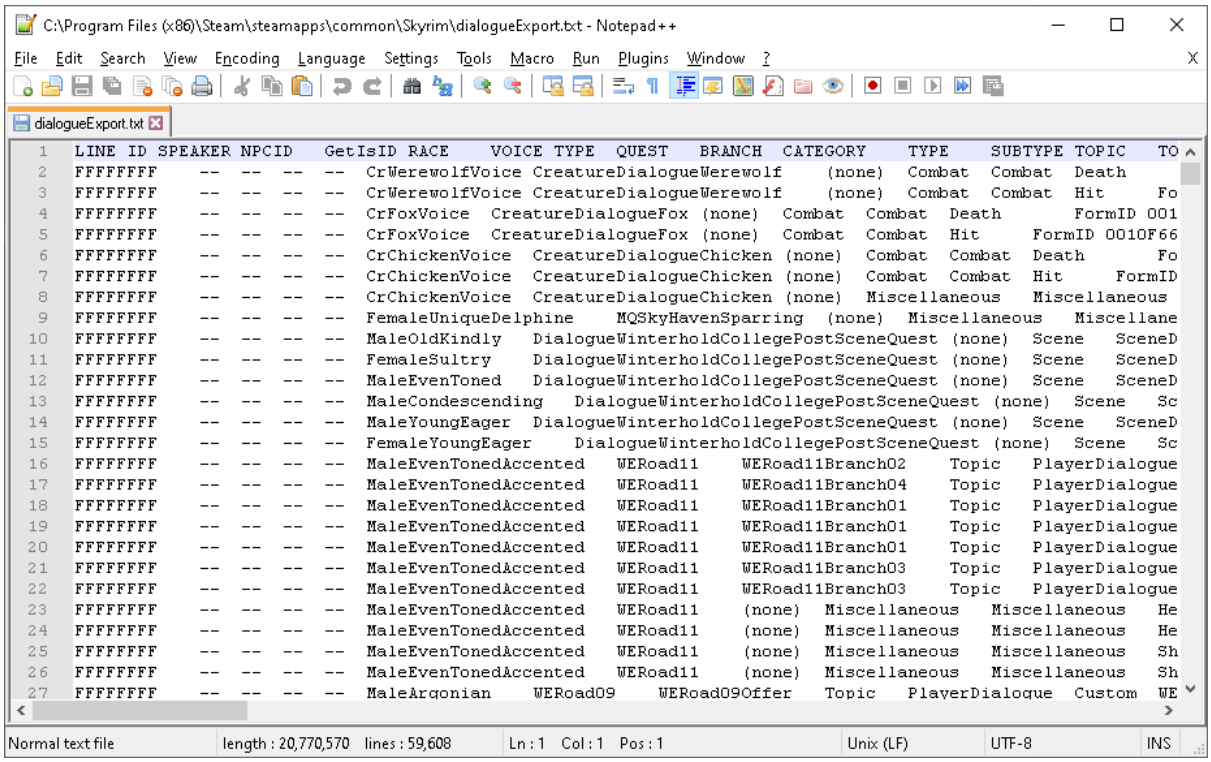


Figure 1313 - dialogueExport.txt.

We'll need to remove unnecessary columns first. Go to Search > Replace. In the 'Find what' field enter a comma ',' and in the 'Replace with' field enter an at symbol '@' then click Replace All.

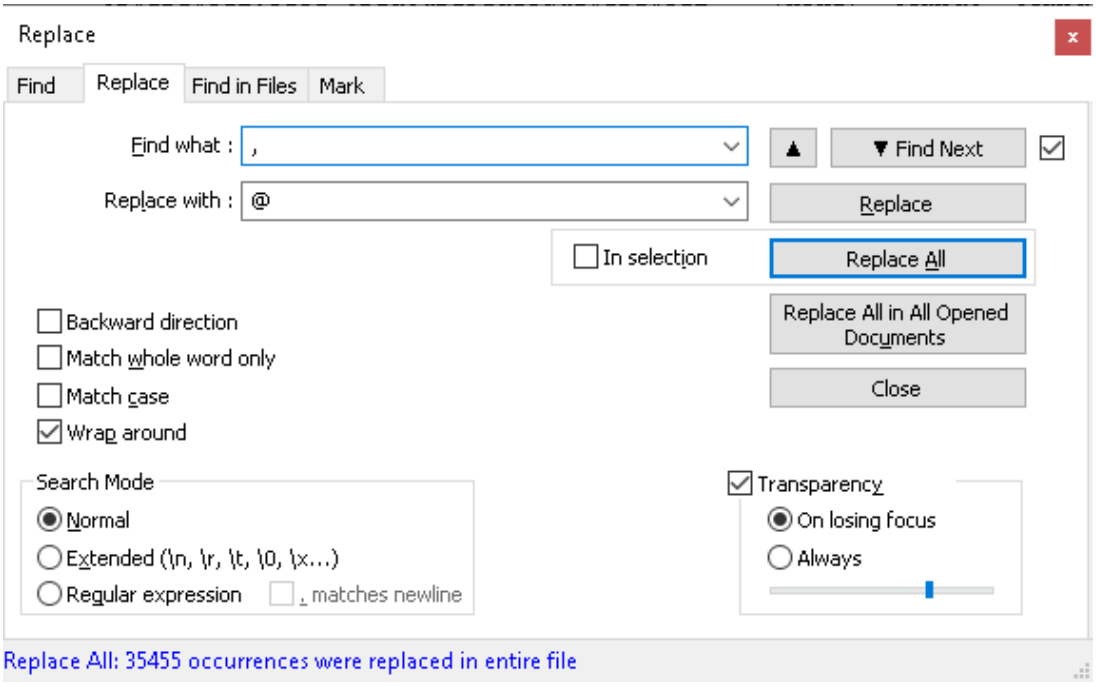


Figure 1314 - Replacing commas with @ symbols.

Don't worry, we'll be reversing that change once we're done in Excel.

Keep Replace open. This time, in the 'Find what' field enter a tab character ' ', and in the 'Replace with' field enter a comma ',' then click Replace All.

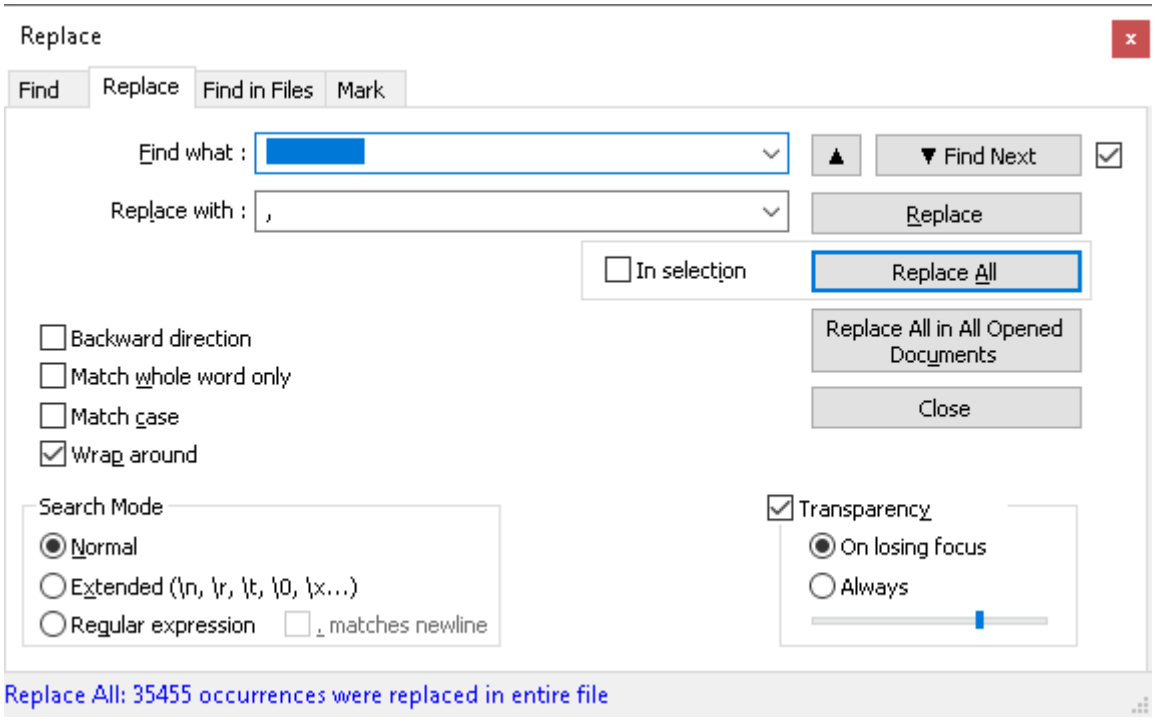


Figure 1315 - Replacing tabs with commas.

Our dialogueExport.txt file should now look like this:

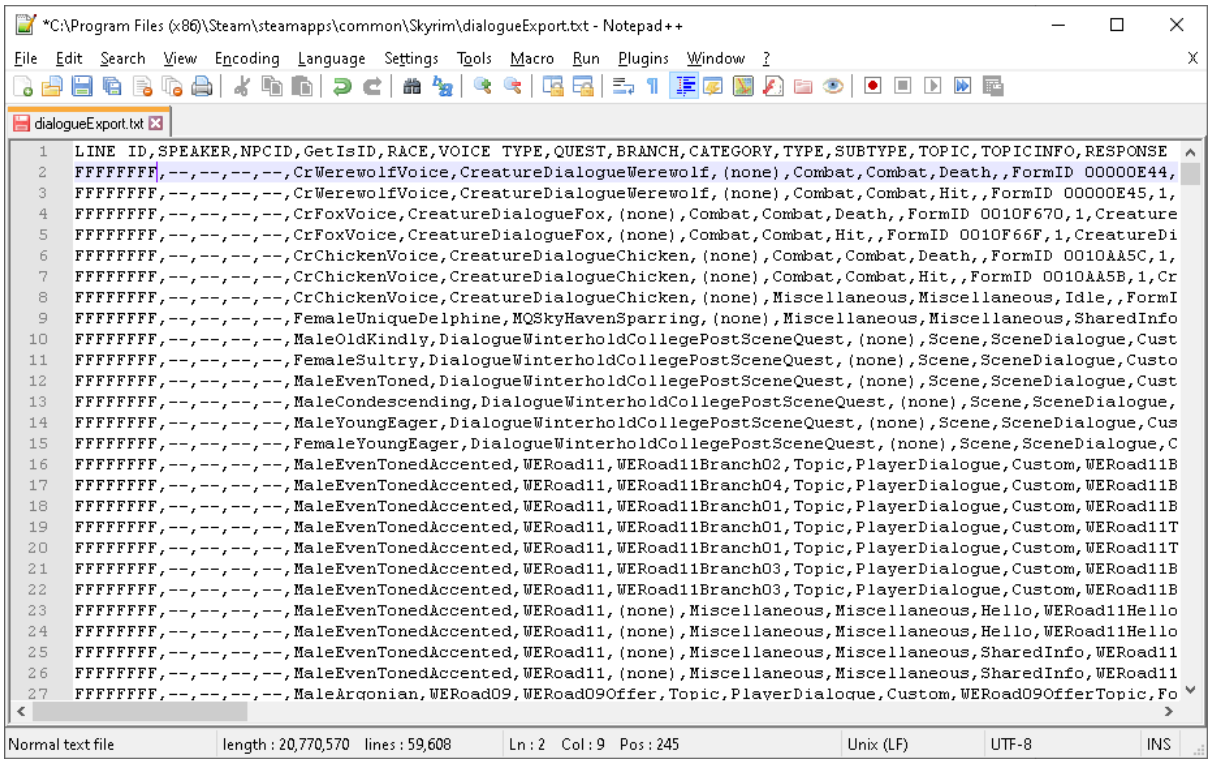


Figure 1316 - dialogueExport.txt modified.

Close Notepad++. Rename dialogueExport.txt to dialogueExport.csv and open it in Excel.

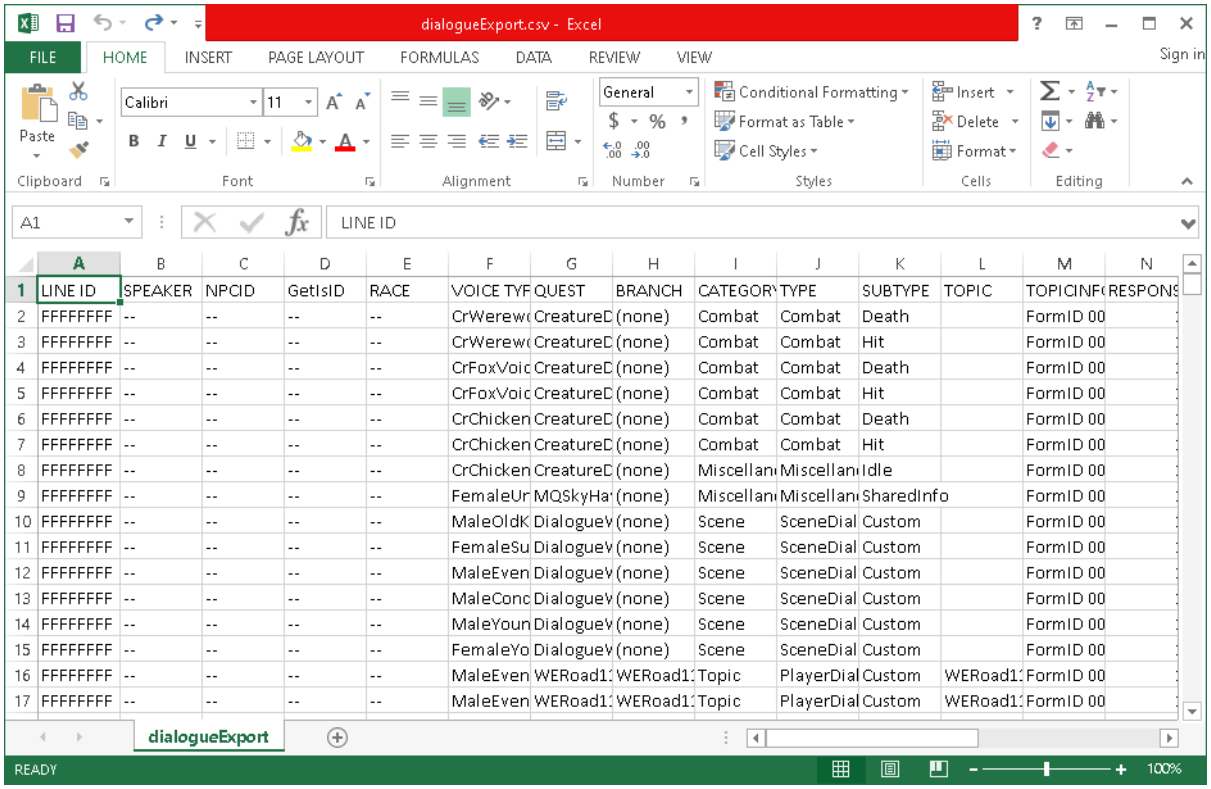


Figure 1317 - dialogueExport.csv.

Delete all columns except 'FULL PATH' and 'RESPONSE TEXT'.

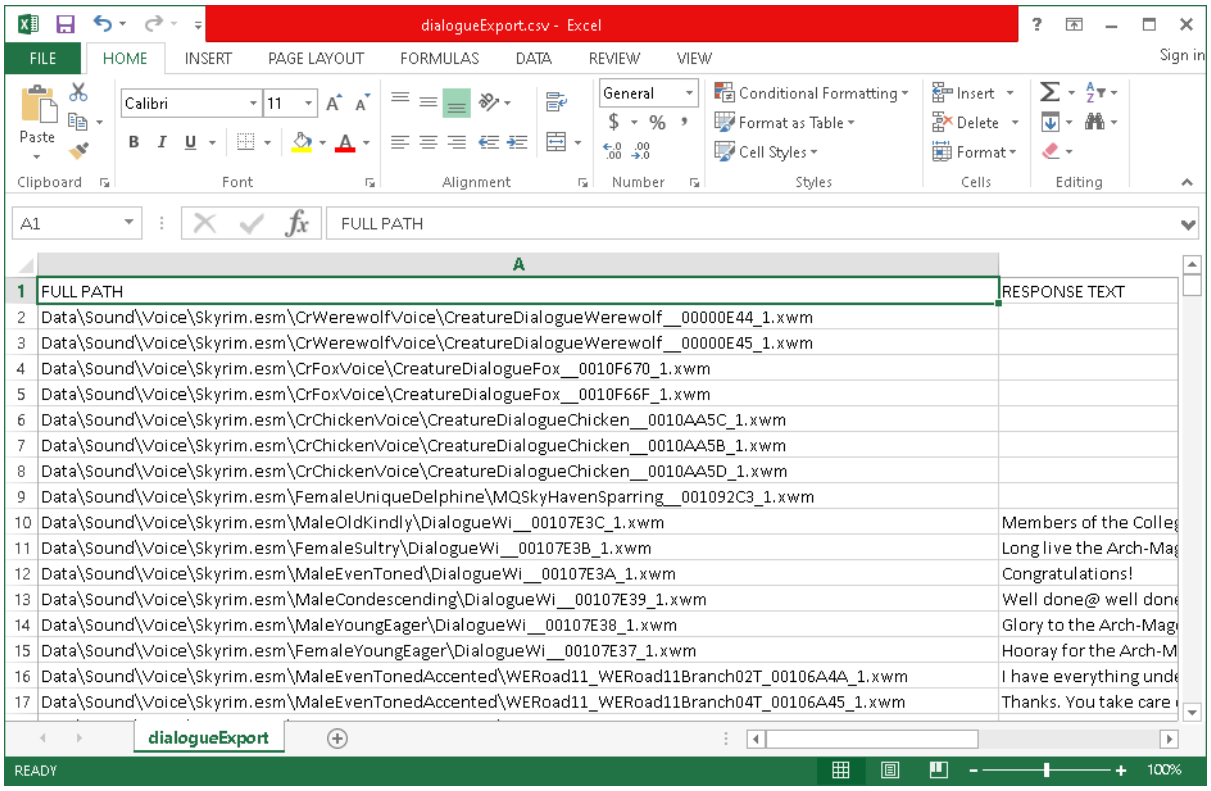


Figure 1318 - Removed unnecessary columns.

Remove the column headers. Right click on '1' on the left hand side to select row 1, then select Delete.

Next we need to clear empty lines. Click on 'B' to select the entirety of column B, then click on the Sort and Filter button and select 'Sort A to Z'.

When the Sort Warning pops up, select 'Expand the selection' then click Sort.

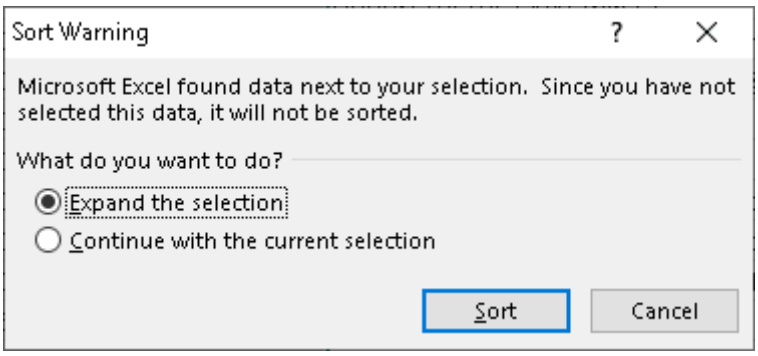


Figure 1319 - Expand the Selection.

Delete the rows where cell B is either blank or has non-dialogue, such as (Deep breath), *cough* or <Laughter>.

Now we need to sort the rows by file name. Click on 'A' to select the entirety of column A, then click on the Sort and Filter button and select 'Sort A to Z'.

Again, when the Sort Warning pops up, select 'Expand the selection' then click Sort.

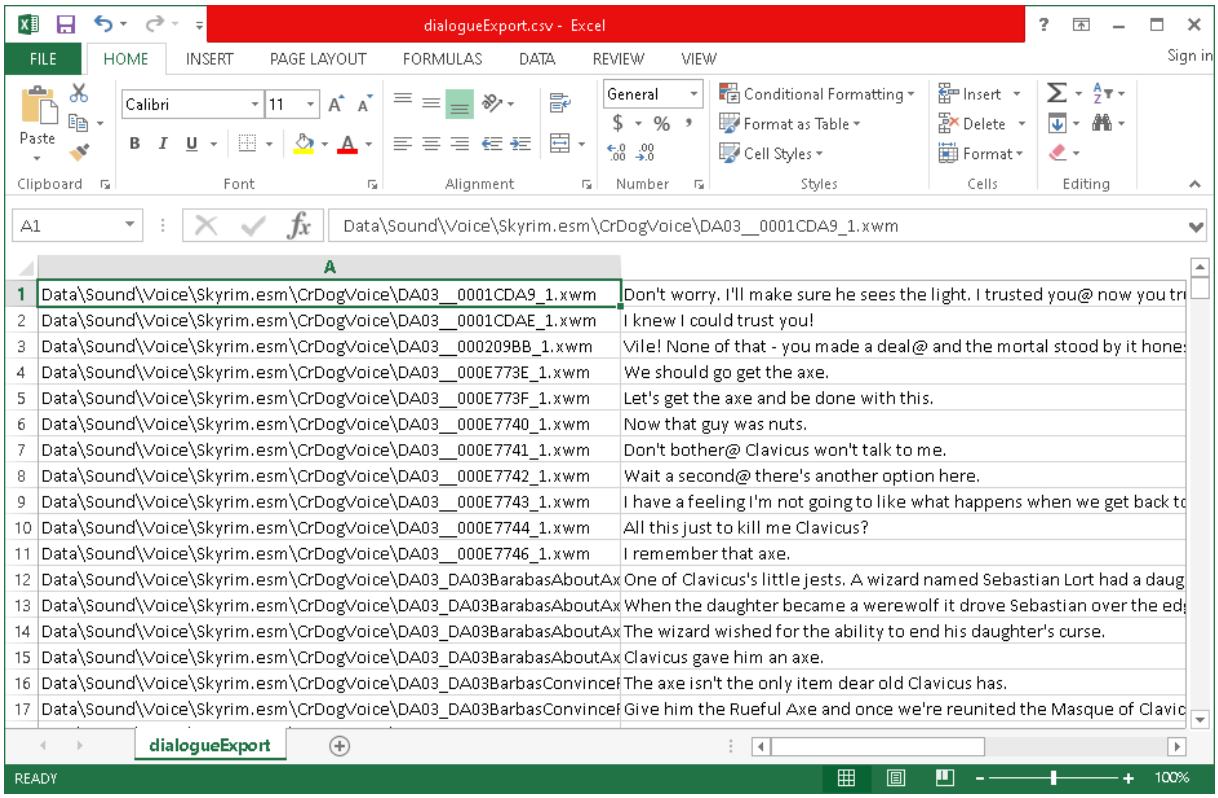


Figure 1320 - dialogueExport.csv properly sorted.

Save the document then close Excel. Open dialogueExport.csv again in Notepad++.

In Notepad++, go to Search > Replace. In the 'Find what' field enter an at symbol '@', and in the 'Replace with' field enter a comma ',' then click Replace All.

Next, we need to change the file extensions. In the 'Find what' field enter '.xwm,' and in the 'Replace with' field enter '.wav|' then click Replace All.

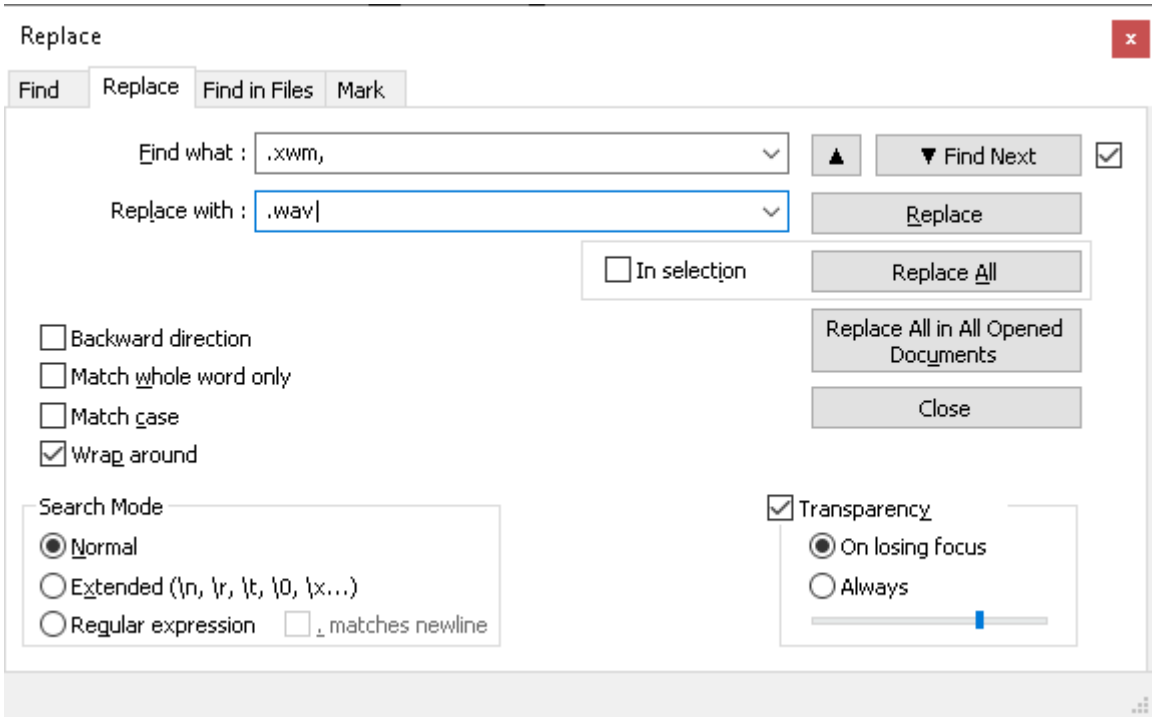


Figure 1321 - Replacing the file extensions and separator character.

We need to make everything lower case. To do this, go to Edit > Select All then go to Edit > Convert Case To > lowercase.

The file paths need to be modified for Google Drive. In the 'Find what' field enter 'data\sound\voice\skyrim.esm\' and in the 'Replace with' field enter '/content/drive/MyDrive/' then click Replace All.

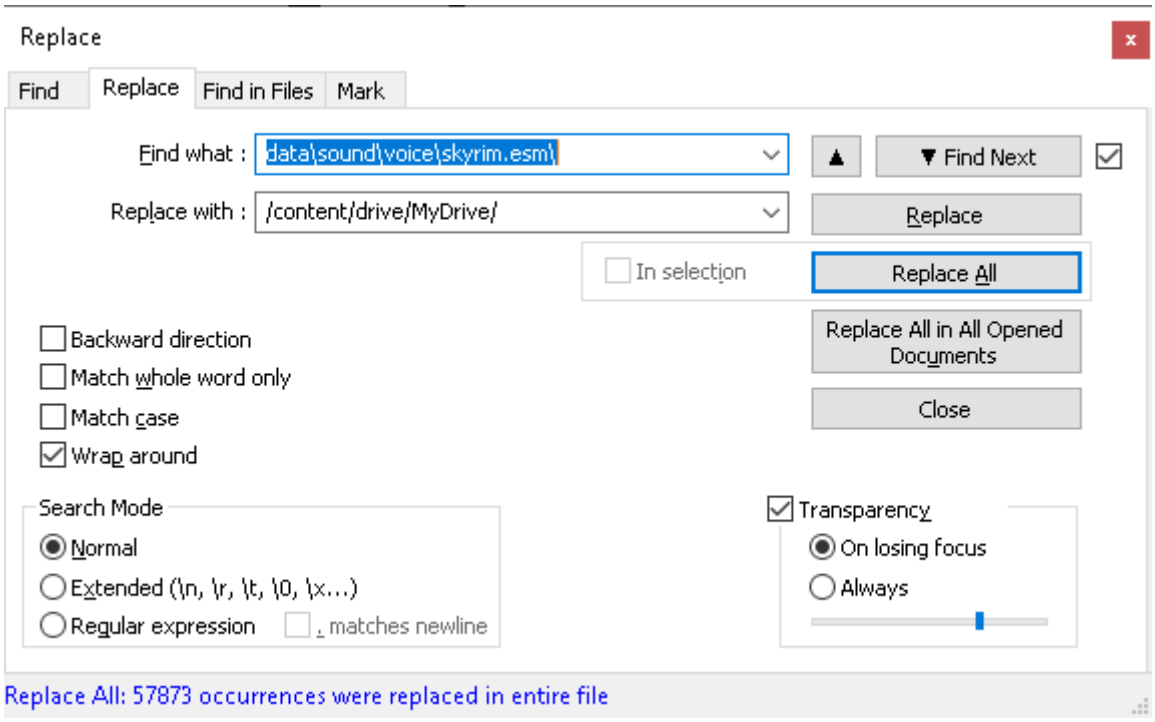


Figure 1322 - Changing the file paths.

Lastly, in the 'Find what' field enter '\' and in the 'Replace with' field enter '/wavs/' then click Replace All.

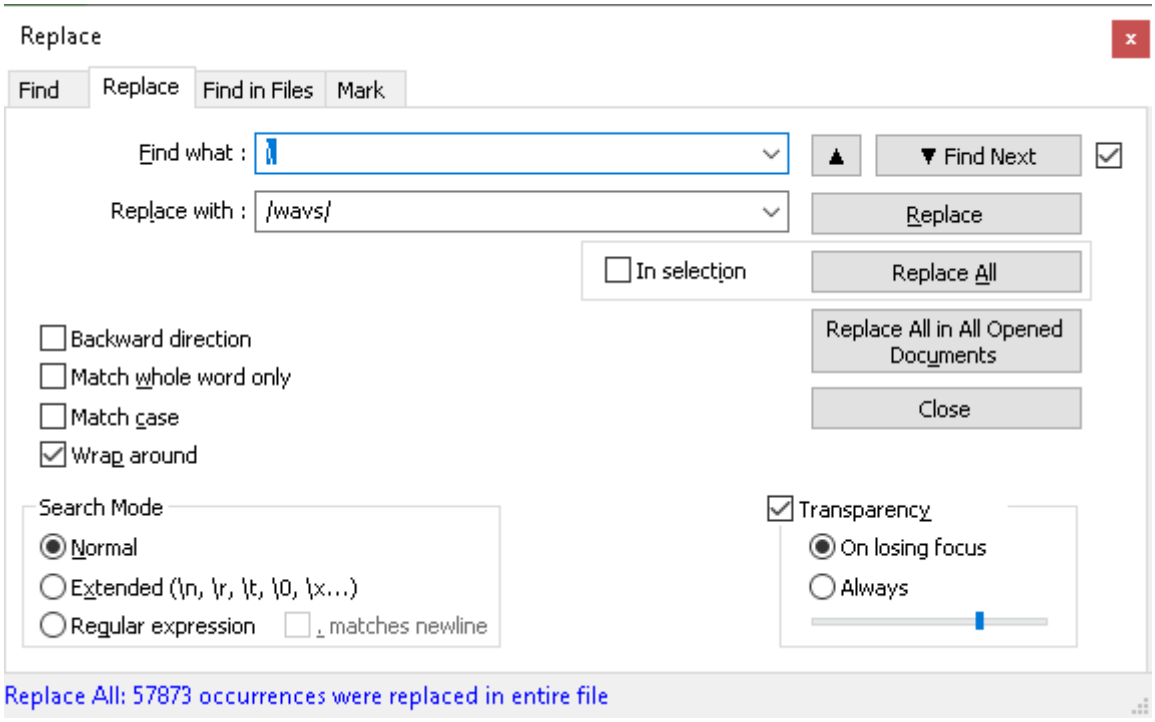


Figure 1323 - Adding wavs folder to file path.

We now have a list of subtitles grouped by order of voice type from which we can easily make new training and validation files.

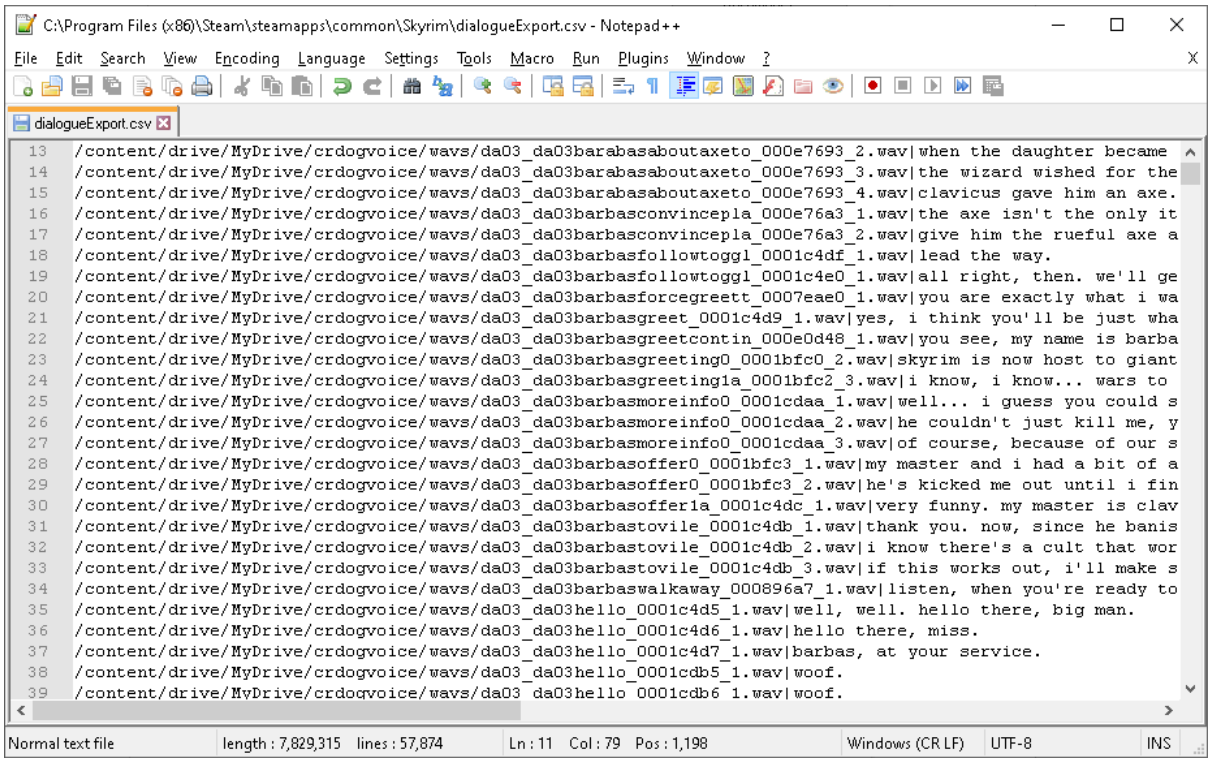


Figure 1324 - dialogueExport.csv formatted for Google Colab.

I've uploaded a couple pre-sanitized dialogueExport files from different Bethesda Game Studio titles to make it easier for you to set up new training and validation files.

Skyrim:
https://drive.google.com/file/d/1lqhCZWOoOsWwK6VDIvakPnUmyMS66As1/view?usp=sharing

Oblivion:
https://drive.google.com/file/d/1uxbN0NwZcWfeqSYzRoFBLQt4iXZNv5cJ/view?usp=sharing

Next we need to create a training text file. Select all the voice lines for the specific voice type you plan on training.

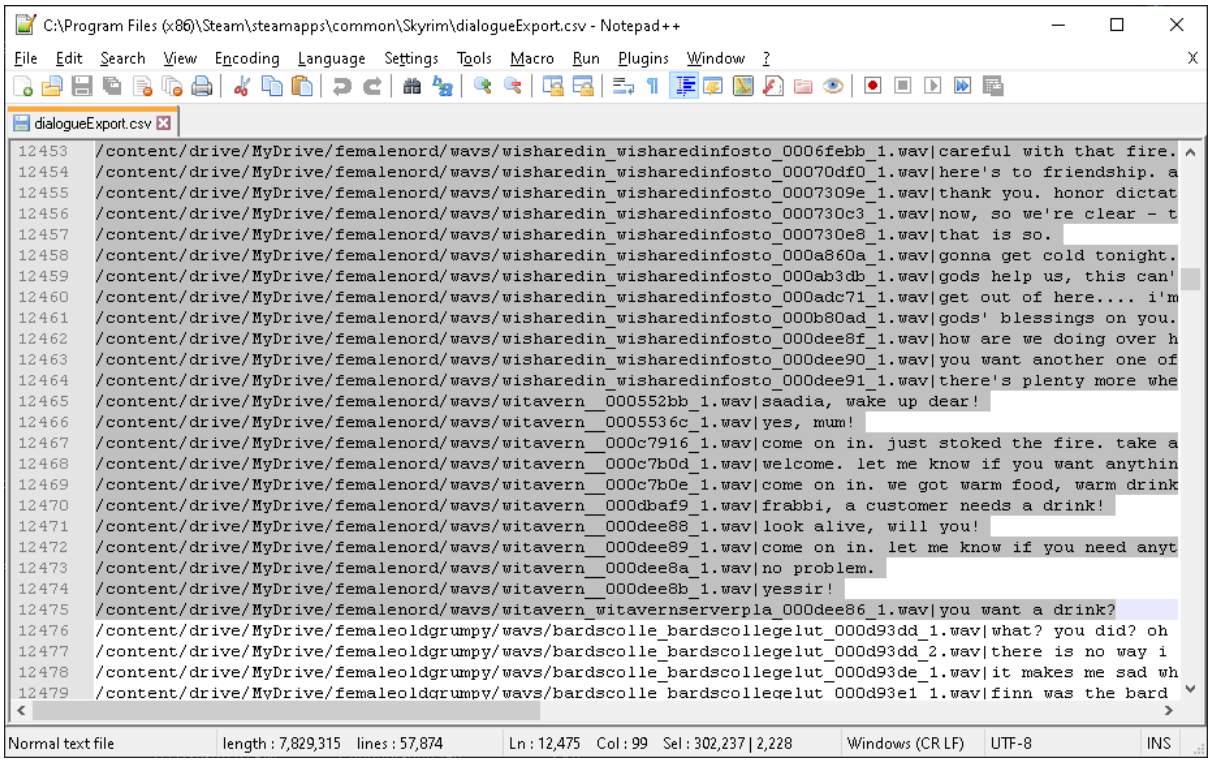


Figure 1325 - Select all lines of dialogue for a specific voice type.

Create a new text file and name it '#_training.txt' where '#' is the name of the voice type you're training, i.e.: femalenord. In my example that's going to be 'femalenord_training.txt' because I'll be making a dataset based on the femalenord voice type.

Copy the lines you highlighted from dialogueExport.csv and paste them into your #_training.txt file.

Now highlight at least 50 lines from your training text file.

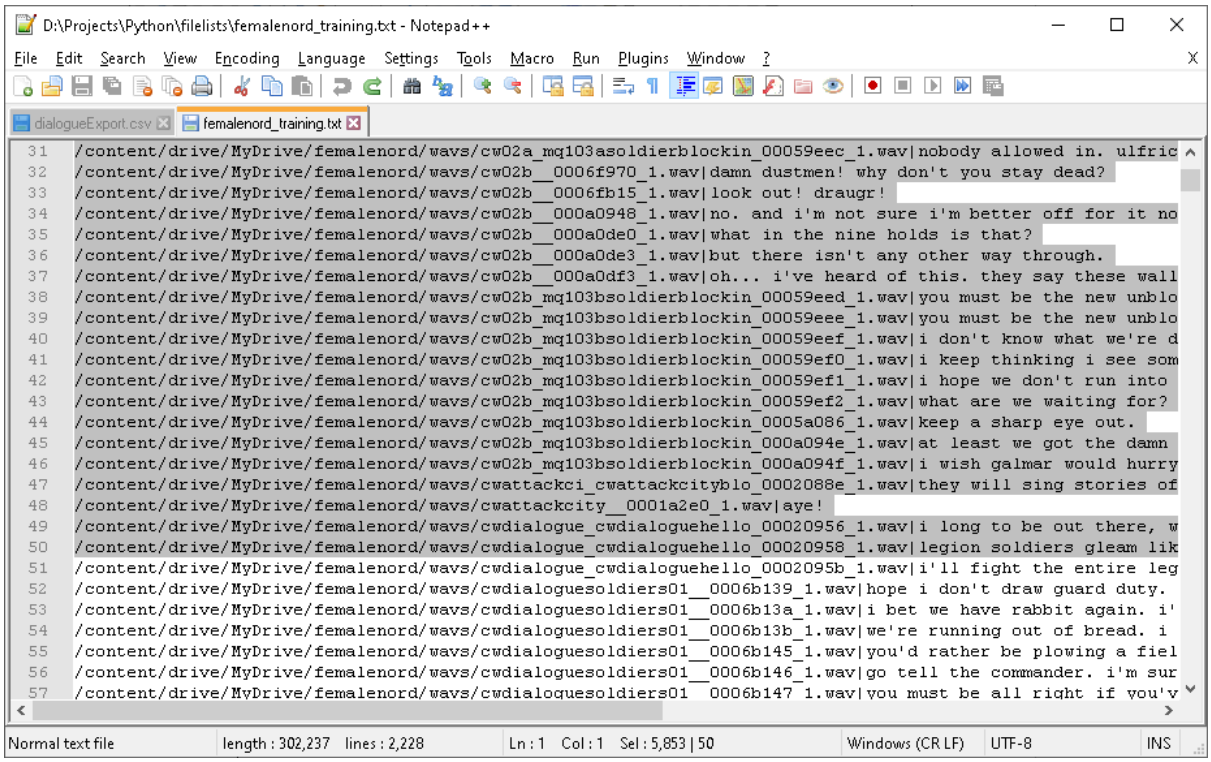


Figure 1326 - 50 lines highlighted.

Create another text file and name it '#_validation.txt' where '#' is the name of the voice type you're training, i.e.: femalenord. In my example that's going to be 'femalenord_validation.txt'.

Cut the highlighted lines out of the training text file and paste them into the validation text file.

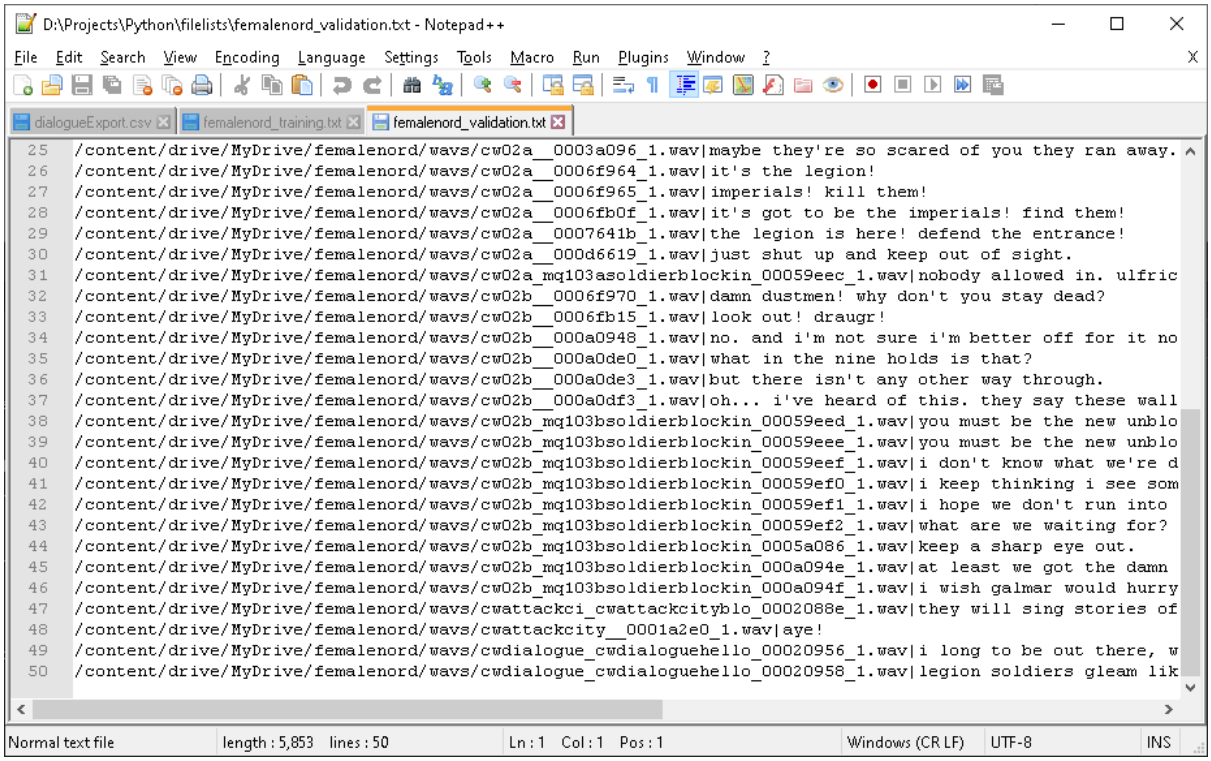


Figure 1327 - Validation text file.

Save both your training and validation text files.

Now that we've prepared the training and validation files, we can move on to preparing the audio.

Extract the voice acting from the 'Skyrim - Voices.bsa' archive. To do this we'll need a copy of BAE, the Bethesda Archive Extractor, which you can download from Nexusmods.

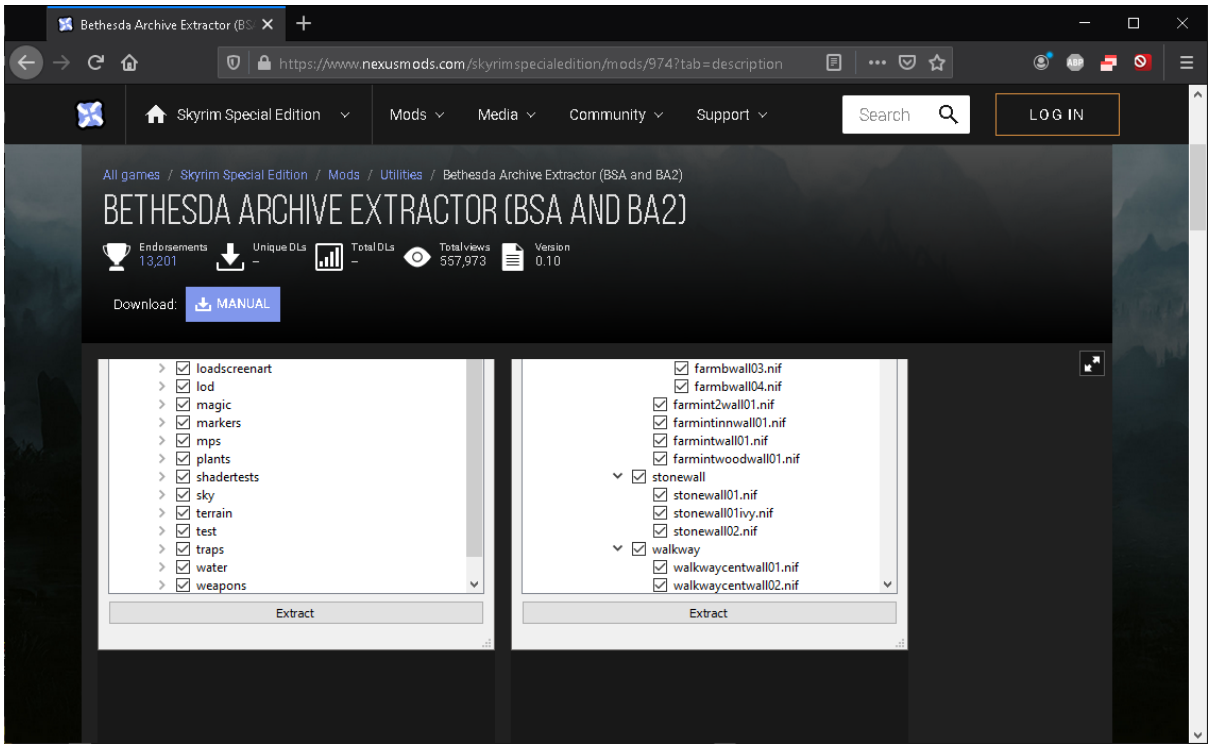


Figure 1328 - BAE on Nexusmods.

Extract the archive to a folder and run bae.exe.

Drag and drop the 'Skyrim - Voices.bsa' file into BAE to open it.

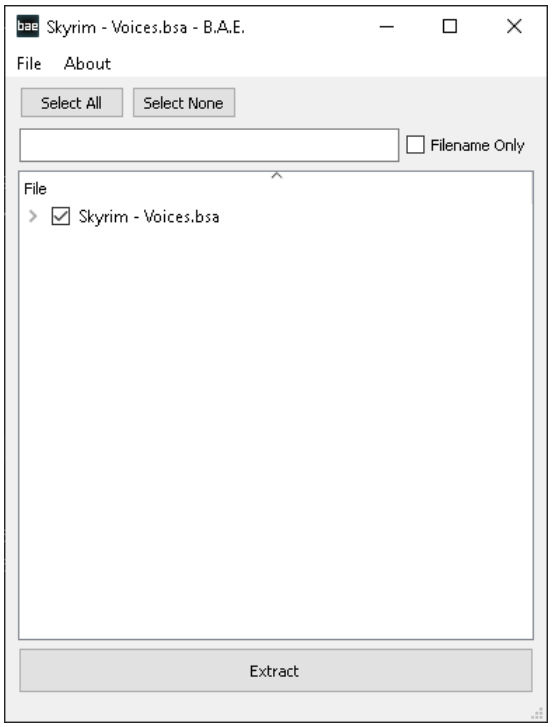


Figure 1329 - Skyrim - Voices.bsa opened in BAE.

Click 'Select All' to ensure everything is selected, then click Extract.

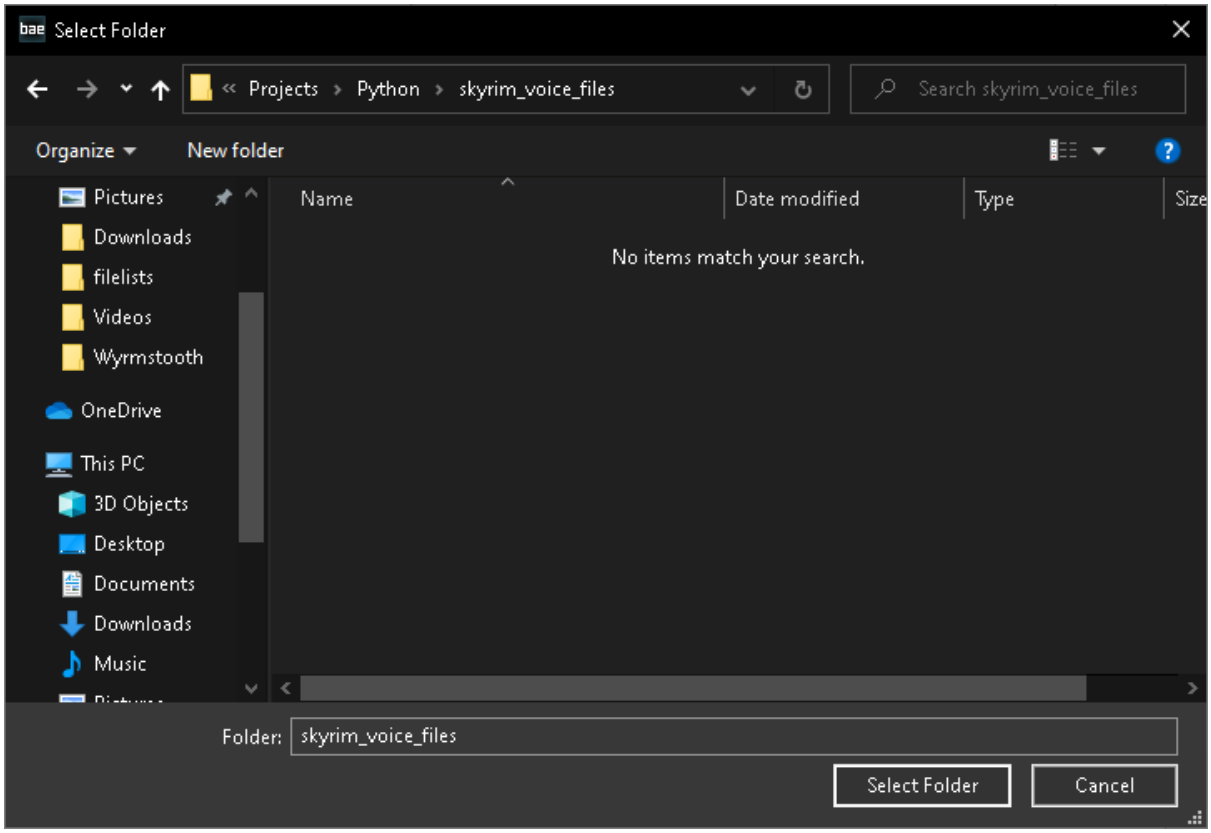


Figure 1330 - Selecting a folder to extract to.

Select a folder to extract the voice files to, then click on Select Folder.

Once BAE has finished extracting everything from the 'Skyrim - Voices.bsa' archive, navigate to the folder you selected, then go to sound > voice > skyrim.esm. You should see a folder for each voice type and these folders should contain .fuz files.

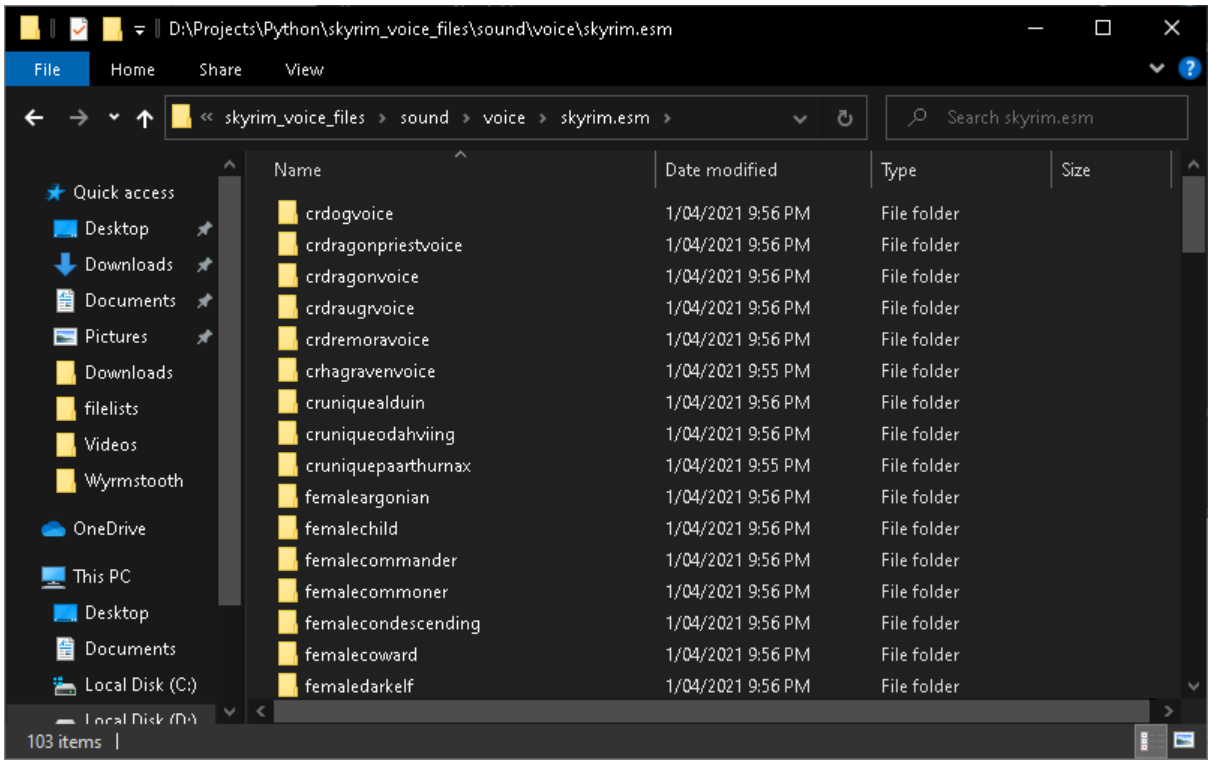


Figure 1331 - Extracted voice type folders.

We'll need to convert those .fuz files to .wav next.

Go ahead and download Unfuzer from Nexusmods.

Extract the archive to a folder and run unfuzer.exe.

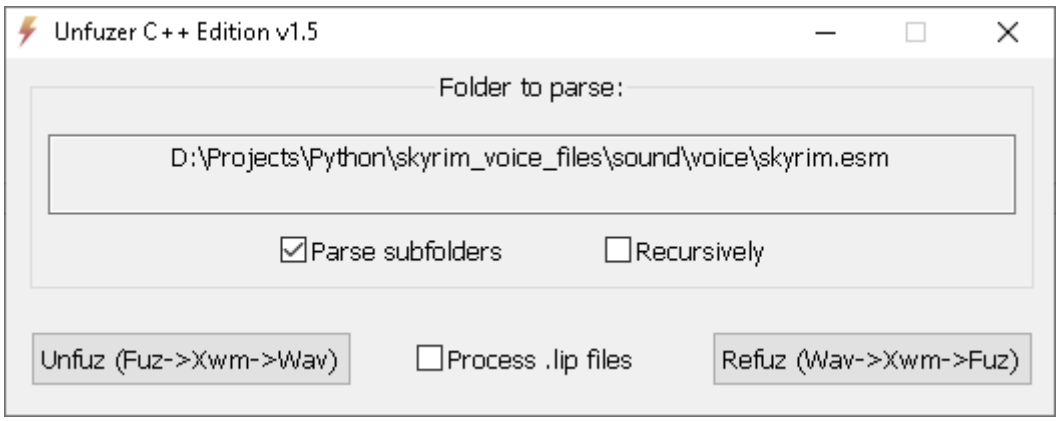


Figure 1332 - Unfuzer.

Click on the file path beneath 'Folder to parse'. Navigate to the folder where you extracted the voice files to, go to sound > voice and select the 'skyrim.esm' folder.

Tick 'Parse subfolders', make sure 'Process .lip Files' isn't ticked, then click Unfuz (Fuz->Xwm->Wav). This will convert every .fuz file in the skyrim.esm subfolders to .wav.

Note: Converting all the .fuz files to .wav will take some time.

In the 'skyrim.esm' folder create a new text file. Rename that file 'delete_fuz.bat'

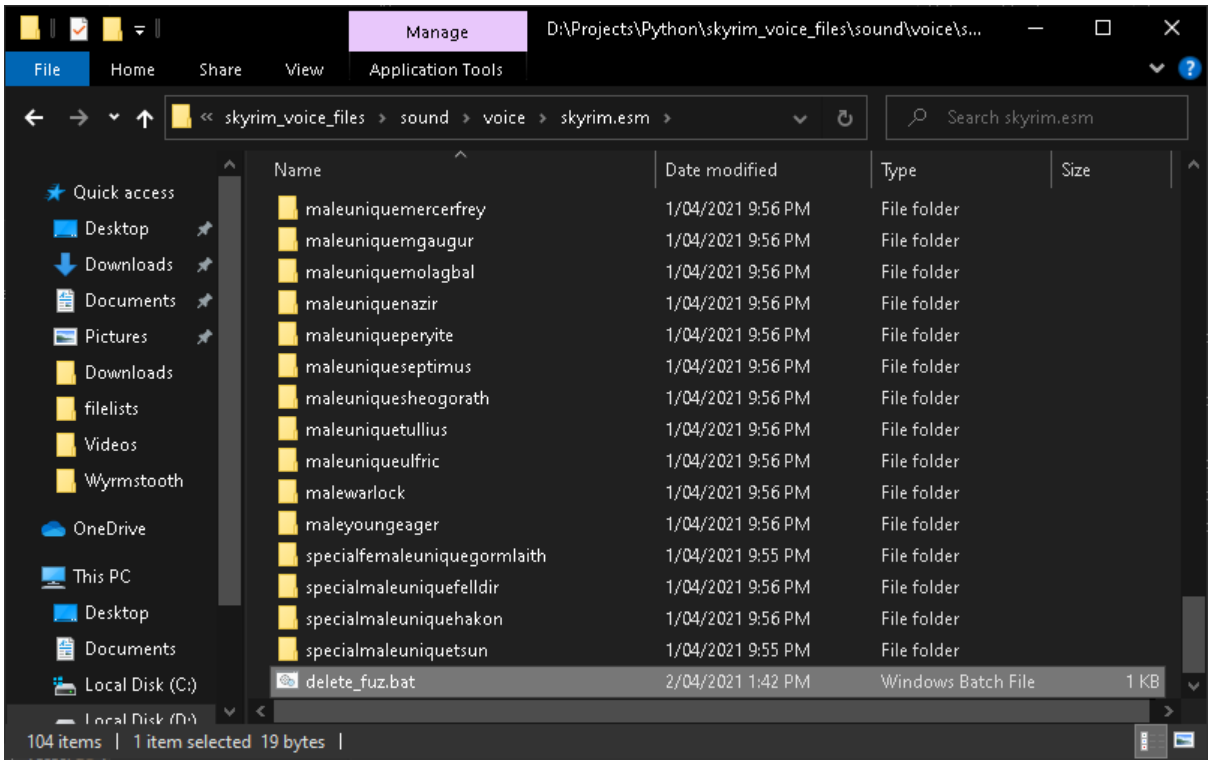


Figure 1333 - Created delete_fuz.bat.

Open the batch file in Notepad and enter the following commands:

```
del /S *.fuz
pause
```

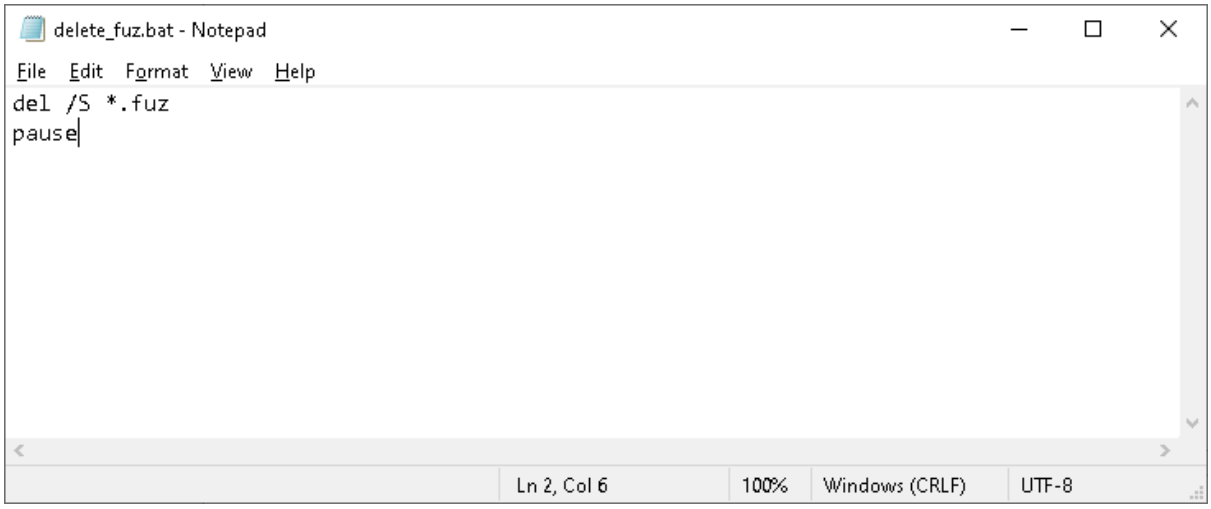


Figure 1334 - delete_fuz.bat contents.

Save and close Notepad then run the batch file. This will delete all .fuz files from the subfolders leaving only the .wav files behind.

We'll need to create a project directory for this next part. I just made a folder called Datasets but you can name it whatever you like.

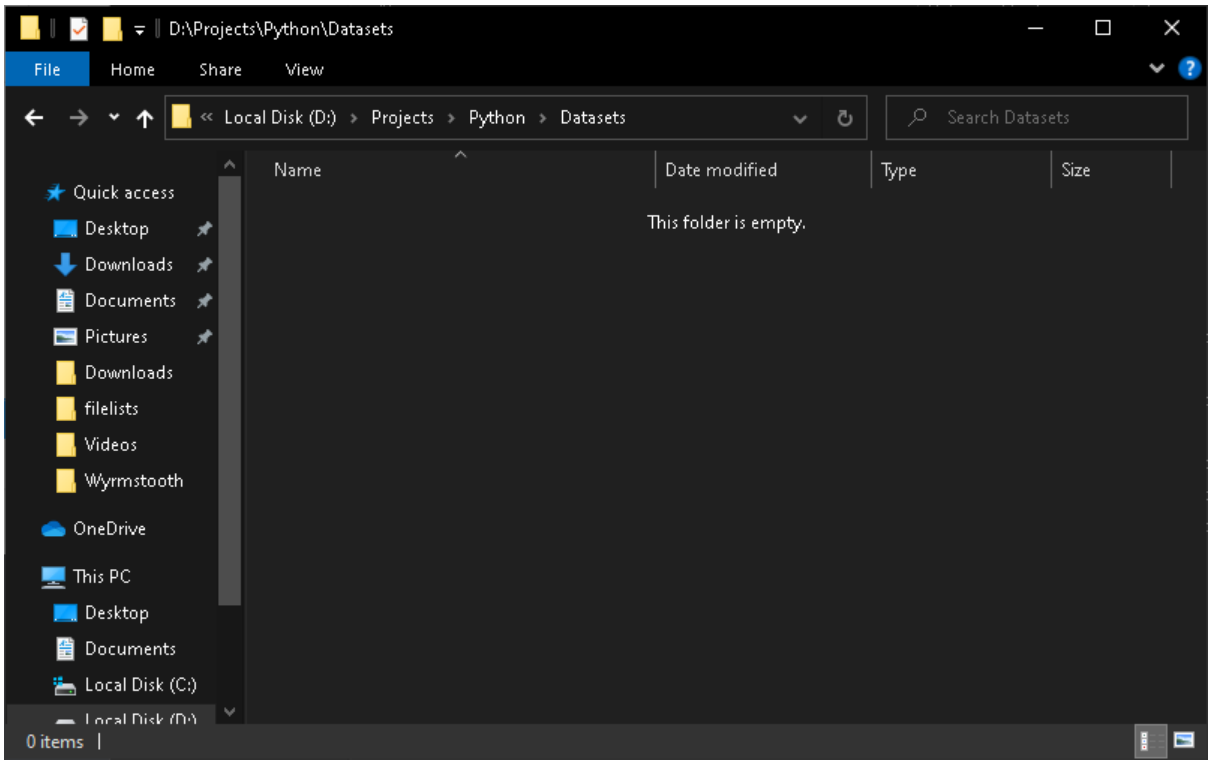


Figure 1335 - New project directory for setting up our dataset.

Copy the folder containing the .wav files for your chosen voice type over to your project folder. In my example I'll be copying the femalenord folder from the 'skyrim.esm' folder over to the Datasets folder I just created.

Once you've copied it to the project folder, append '_base' to the folder name.

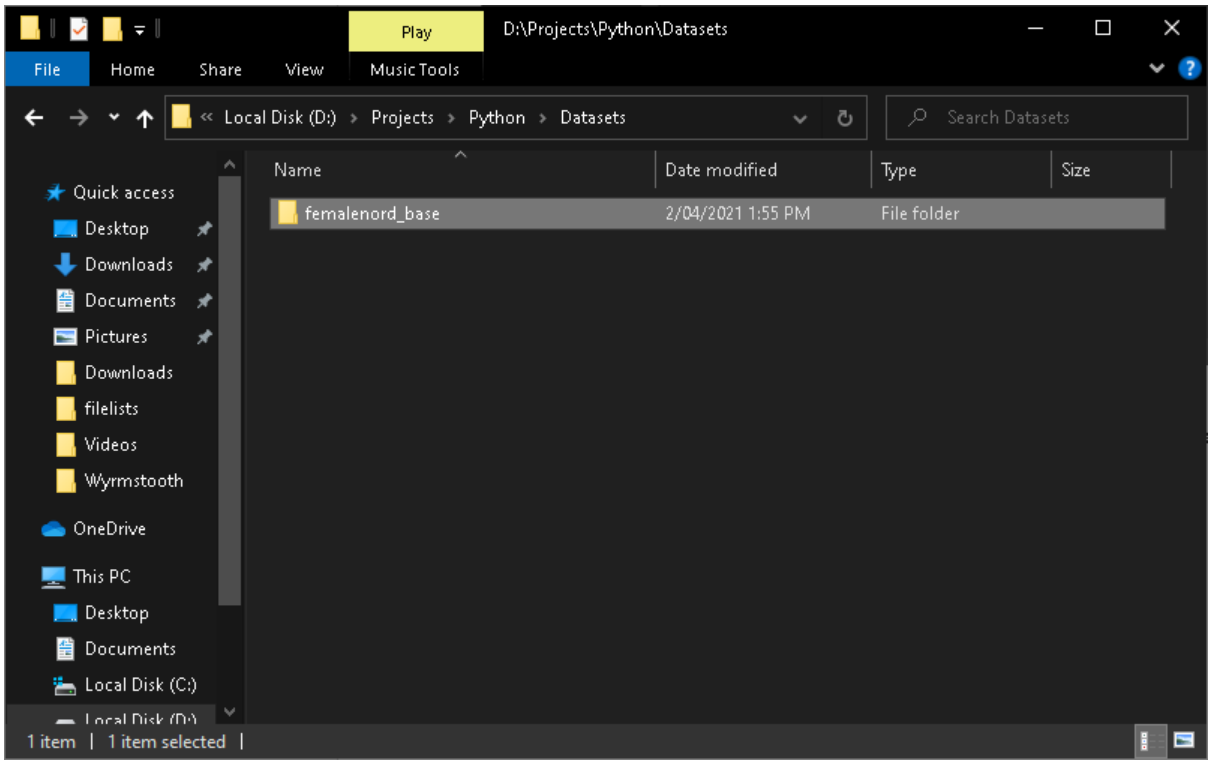


Figure 1336 - Copied the unprocessed audio files to the project folder.

Tacotron and WaveGlow will require audio files to be formatted in a specific way, so in this section I'll be showing you how to batch process the audio files for your dataset. In order to do so we'll need to download two utilities: SoX and ffmpeg.

SoX can be downloaded from its page on Sourceforge: http://sox.sourceforge.net/.

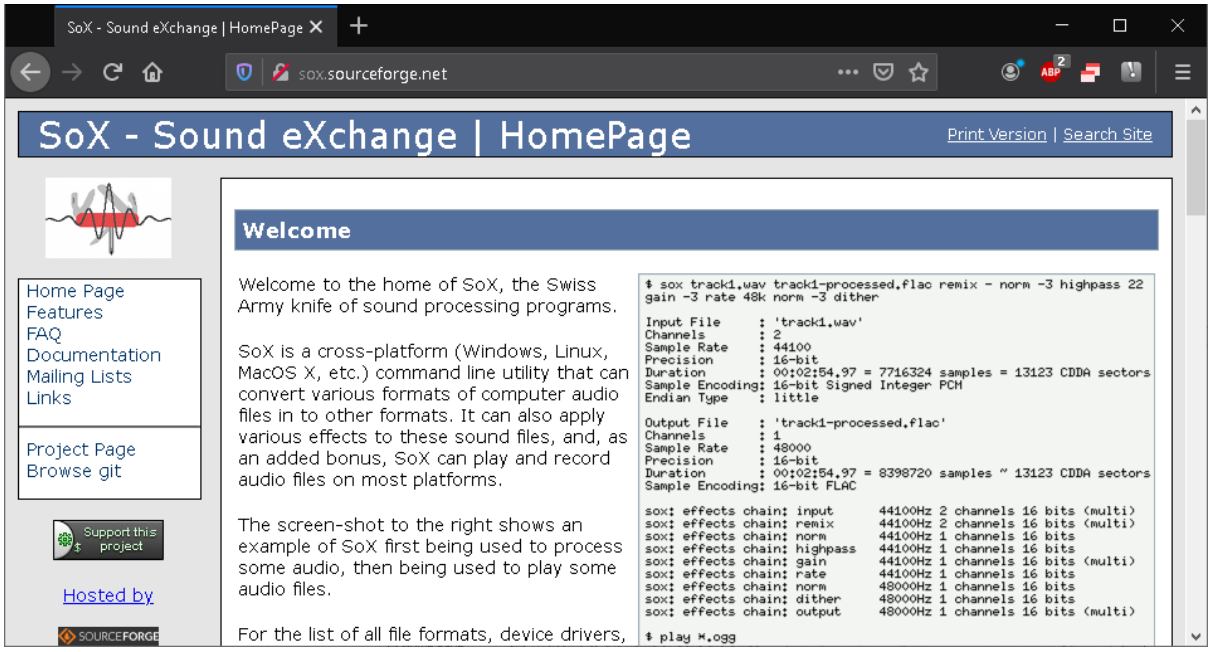


Figure 1337 - SoX website.

Once downloaded, extract the archive to an easily accessible location. On my end I just extracted SoX to a new folder in Program Files (x86).

ffmpeg can be downloaded from its website. Again, once you've downloaded it, extract the files to an easily accessible location. On my end I just extracted ffmpeg to a new folder in Program Files (x86).

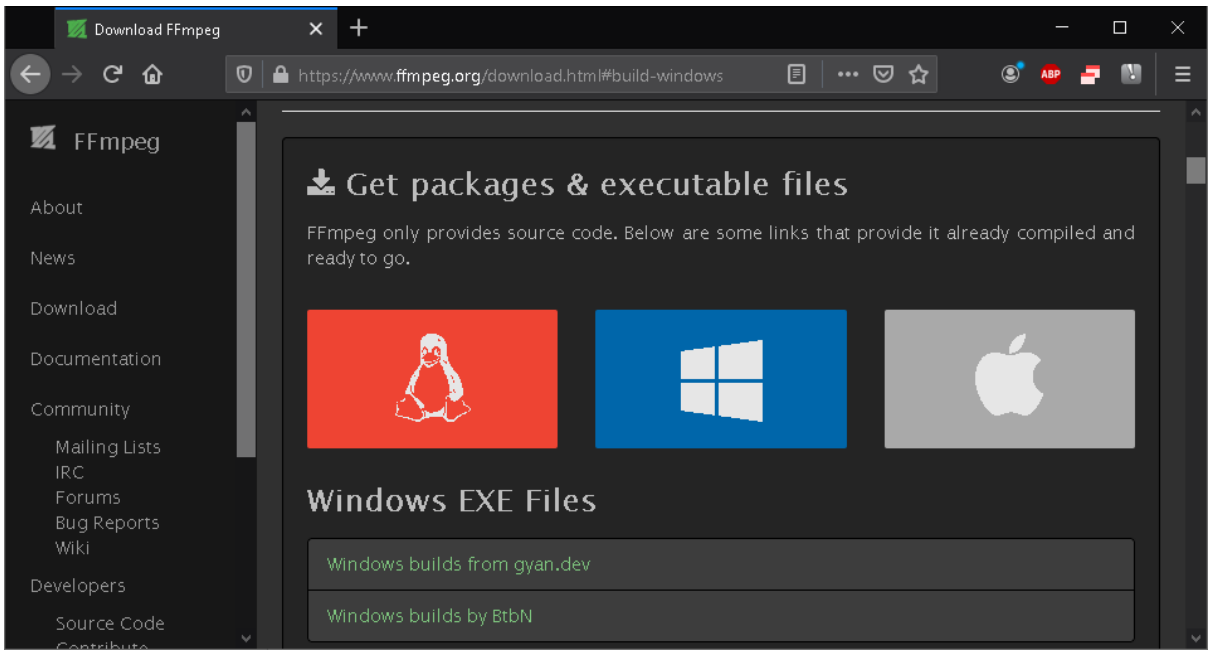


Figure 1338 - ffmpeg website.

Firstly, if you're copying audio files from Oblivion, or from a game where the audio files aren't in .wav format, we'll need to convert them to .wav first.

Create a new batch file and name it '0_mp3_to_wav.bat'. Open it in Notepad and enter in the following commands:

```
cd "D:\Projects\Python\Datasets\femalenord_base\"
for %%a in (*.mp3) do "C:\Program Files (x86)\ffmpeg\bin\ffmpeg.exe"
-i "%%a" "%%~na.wav"
pause
```

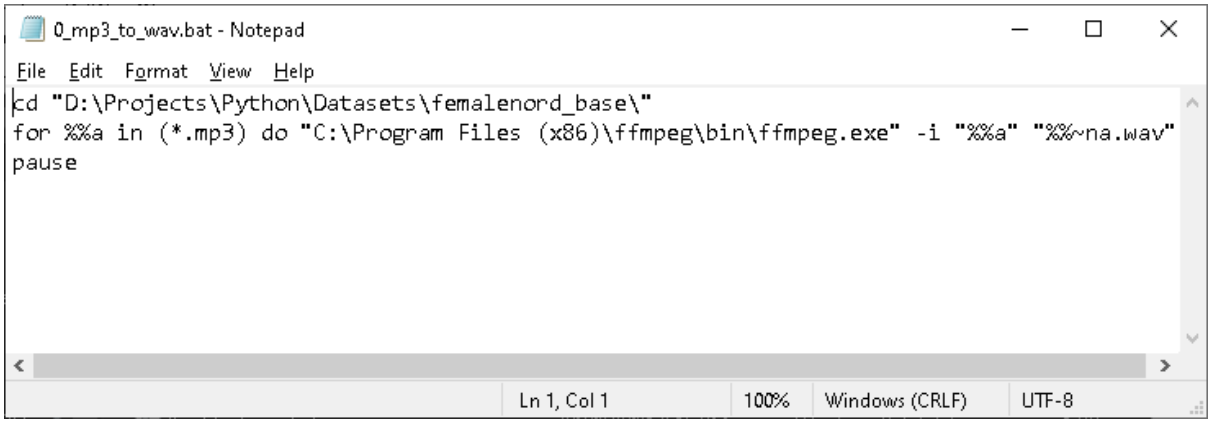


Figure 1339 - 0_mp3_to_wav.bat.

Note: You'll need to change the folder paths to match your environment. In this example I'm converting .mp3 files to .wav. If your files are already in .wav format you won't need to do this.

Save and close Notepad then run '0_mp3_to_wav.bat' and wait for it to complete. The .wav files will be outputted back to the folder containing the original files.

Now we need to trim silence from our .wav files. To do this, create a new batch file and name it '1_trim_silence.bat'. Open it in Notepad and enter in the following commands:

```
if not exist "D:\Projects\Python\Datasets\output_1" mkdir
D:\Projects\Python\Datasets\output_1
cd "D:\Projects\Python\Datasets\femalenord_base"
FOR %%F IN (*.wav) DO "C:\Program Files (x86)\sox-14-4-2\sox.exe"
%%F "D:\Projects\Python\Datasets\output_1\%%~nxF"^
 silence 1 0.1 1% reverse silence 1 0.1 1% reverse silence 1 0.1 1%
reverse
pause
```

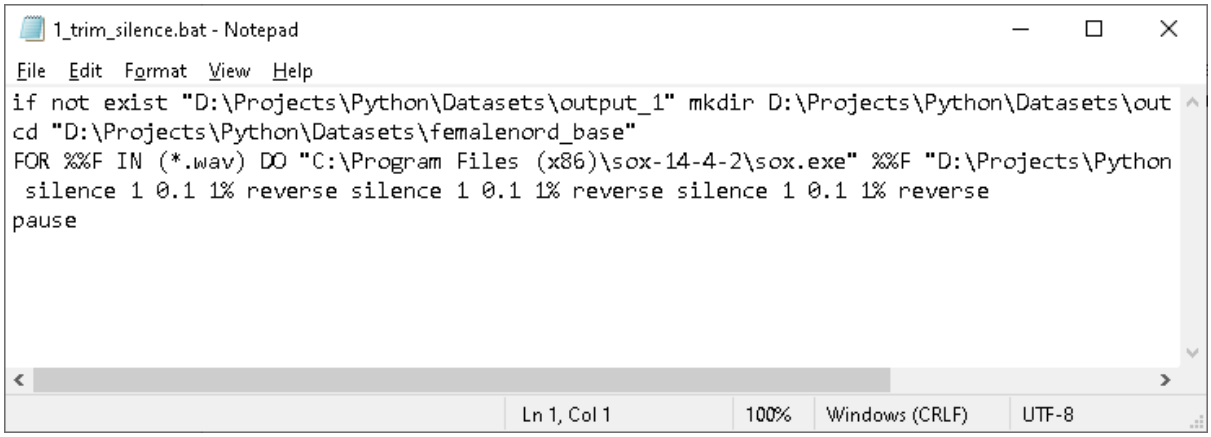


Figure 1340 - 1_trim_silence.bat.

Note: You'll need to change the folder paths to match your environment. We'll be reading .wav files from the #_base folder and outputting to a folder called output_1.

Save and close Notepad then run '1_trim_silence.bat' and wait for it to complete.

Next we'll need to add some silence back to the end of each .wav file. In order to do this, create a new batch file and name it '2_add_silence.bat'. Open it in Notepad and enter in the following commands:

```
if not exist "D:\Projects\Python\Datasets\output_2" mkdir
D:\Projects\Python\Datasets\output_2
cd "D:\Projects\Python\Datasets\output_1\"
FOR %%F IN (*.wav) DO "C:\Program Files (x86)\sox-14-4-2\sox.exe"
%%F "D:\Projects\Python\Datasets\output_2\%%~nxF" pad 0 0.1
pause
```

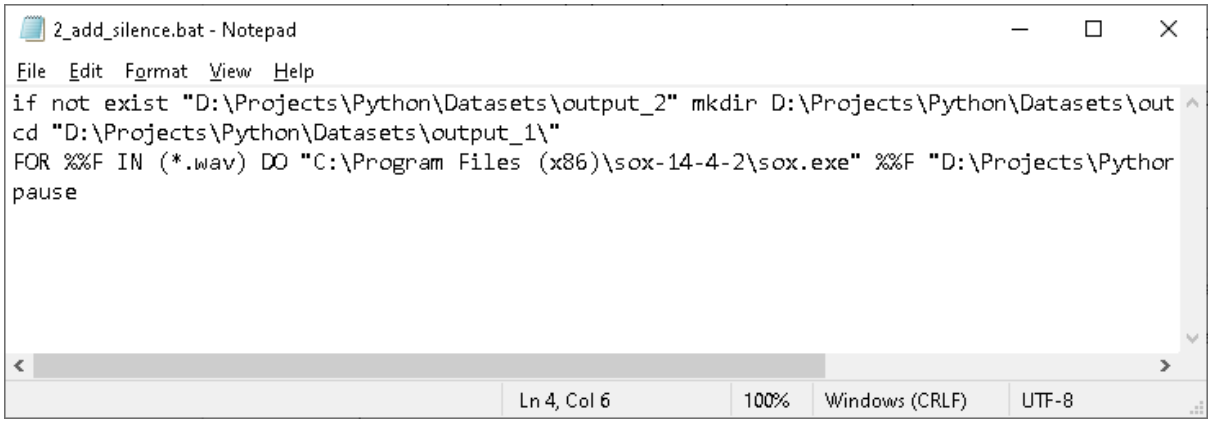


Figure 1341 - 2_add_silence.bat.

Note: You'll need to change the folder paths to match your environment. We'll be reading .wav files from the output_1 folder and outputting to a folder called output_2.

Save and close Notepad then run '2_add_silence.bat' and wait for it to complete.

We'll need to downsample the audio to 22050Hz. Voice acting in most Bethesda Game Studios titles is recorded at 44100Hz. Create a new batch file and name it '3_downsample.bat'. Open it in Notepad and enter in the following commands:

```
if not exist "D:\Projects\Python\Datasets\output_3" mkdir
D:\Projects\Python\Datasets\output_3
cd "D:\Projects\Python\Datasets\output_2\"
FOR %%F IN (*.wav) DO "C:\Program Files (x86)\sox-14-4-2\sox.exe"
%%F "D:\Projects\Python\Datasets\output_3\%%~nxF" rate -v 22050
pause
```

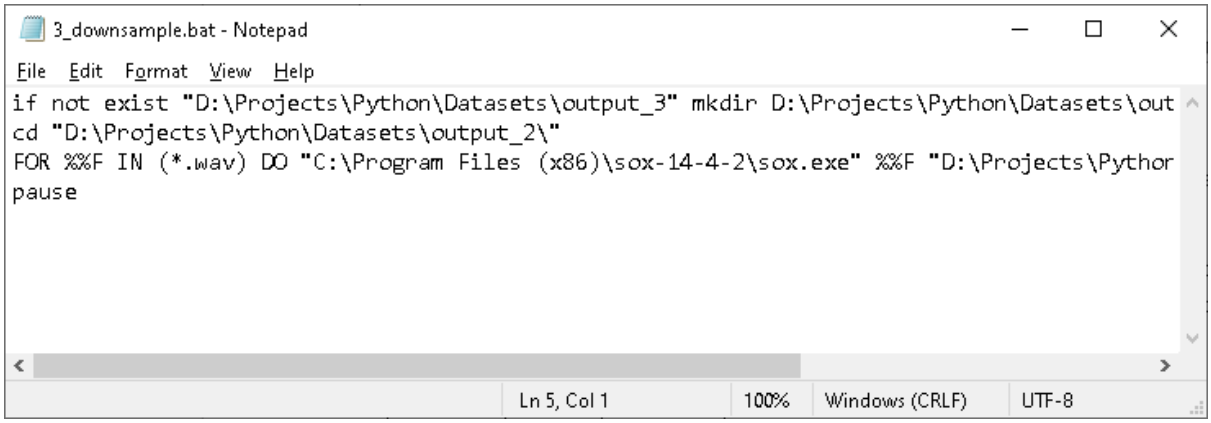


Figure 1342 - 3_downsample.bat.

Note: You'll need to change the folder paths to match your environment. We'll be reading .wav files from the output_2 folder and outputting to a folder called output_3.

Save and close Notepad then run '3_downsample.bat' and wait for it to complete.

Lastly we need to make sure all our .wav files are in mono. Create a new batch file and name it '4_mono.bat'. Open it in Notepad and enter in the following commands:

```
if not exist "D:\Projects\Python\Datasets\output_4" mkdir
D:\Projects\Python\Datasets\output_4
cd "D:\Projects\Python\Datasets\output_3\"
for %%S in (*.wav) do "C:\Program Files (x86)\ffmpeg\bin\ffmpeg.exe"
-i "%%S" -ac 1 "D:\Projects\Python\Datasets\output_4\%%S"
pause
```

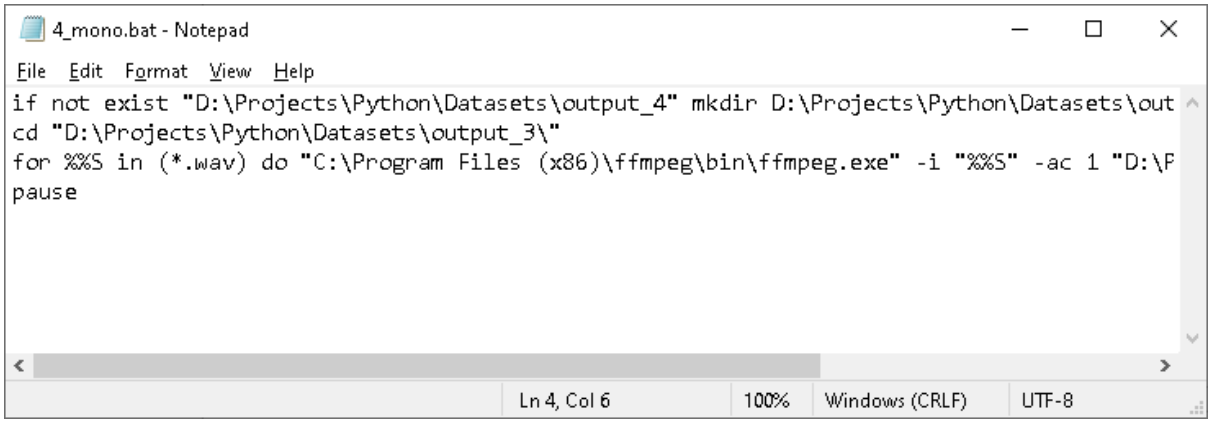


Figure 1343 - 4_mono.bat.

Note: You'll need to change the folder paths to match your environment. We'll be reading .wav files from the output_3 folder and outputting to a folder called output_4.

Save and close Notepad then run '4_mono.bat' and wait for it to complete.

Rename the folder output_4 to match the name of the voice type you're setting up a dataset for. In my example, that will be femalenord.

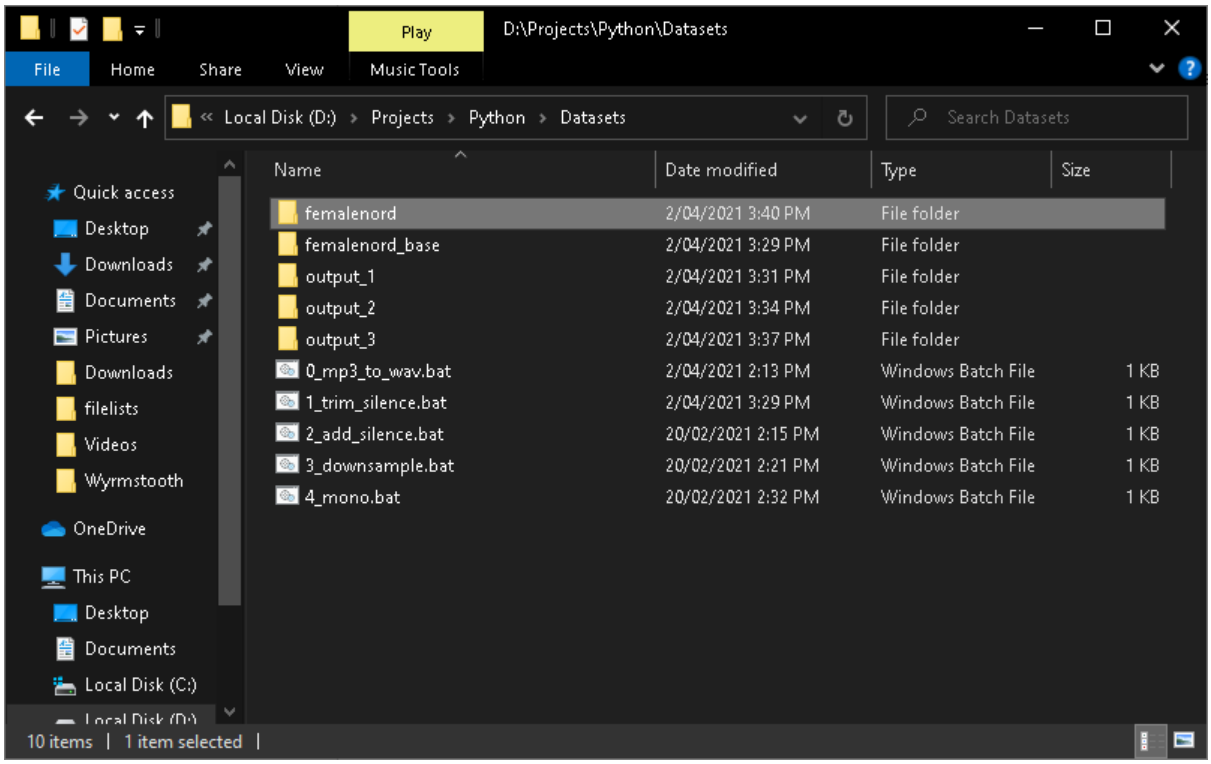


Figure 1344 - Renamed the output_4 folder.

Create a new folder named 'wavs' within it, then move all the .wav files into the 'wavs' folder.

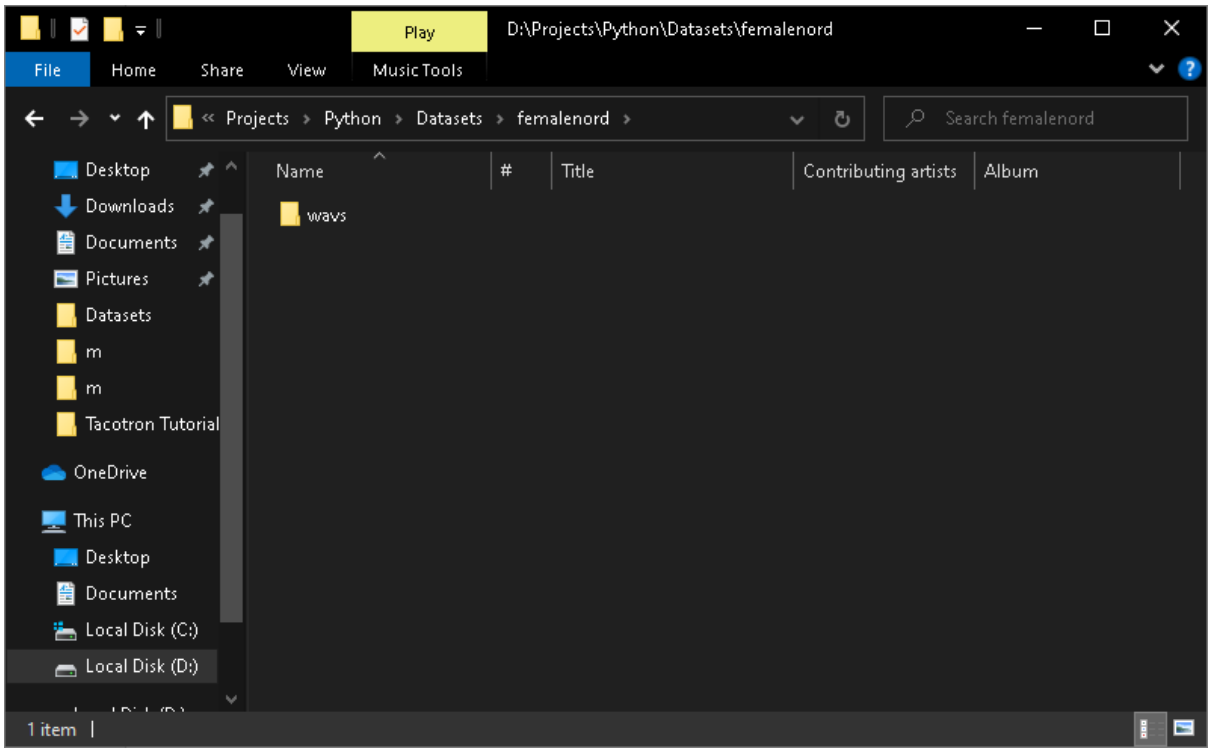


Figure 1345 - wavs folder.

Go to your Google Drive: https://drive.google.com/.

Create a new folder in Google Drive called 'filelists', then drag-and-drop the training and validation text files into your browser to upload them. In my example that's going to be the femalenord_training.txt and femalenord_validation.txt files.

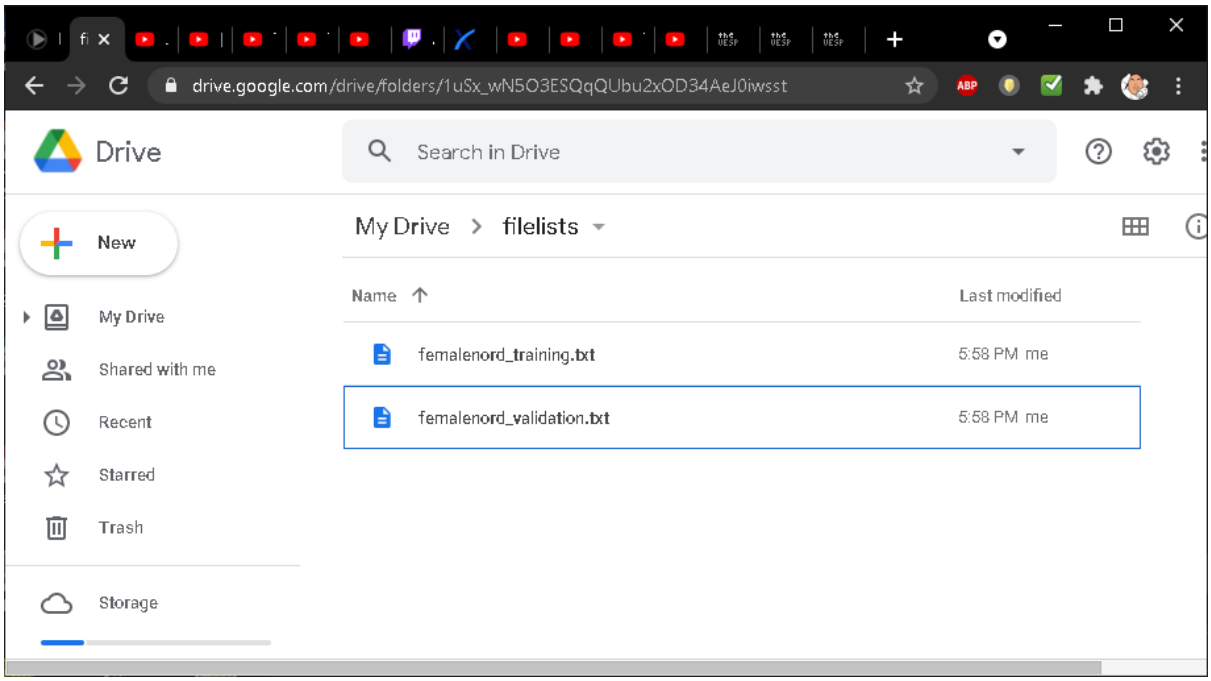


Figure 1346 - Uploading the training and validation files.

Drag-and-drop the folder containing the processed .wav files into your browser to the root of your Google Drive directory to upload them. In my example that's going to be the femalenord folder.

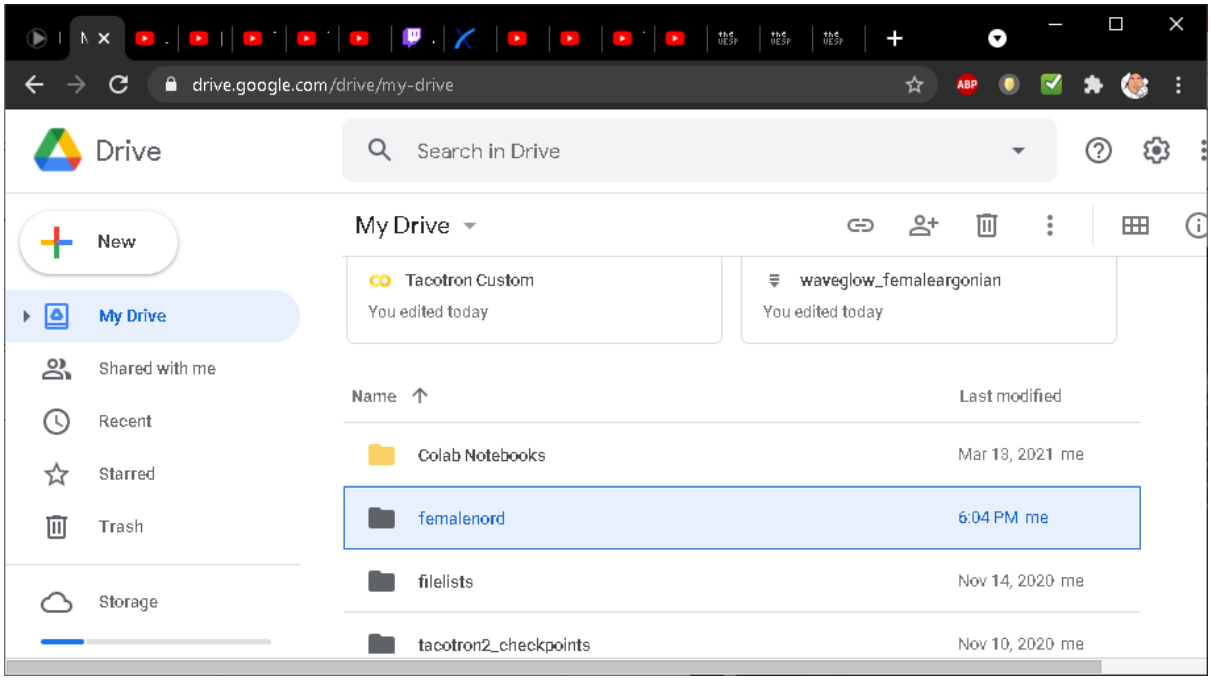


Figure 1347 - Uploading the processed audio files.

MOUNTING YOUR GOOGLE DRIVE STORAGE IN GOOGLE COLAB

As mentioned earlier, I've already set up a Colab Notebook with all the commands you'll need to run. You can access it here: https://colab.research.google.com/drive/13vRLNPLqVWGjgHGuUiJKxXuZw93PUBBr?usp=sharing.

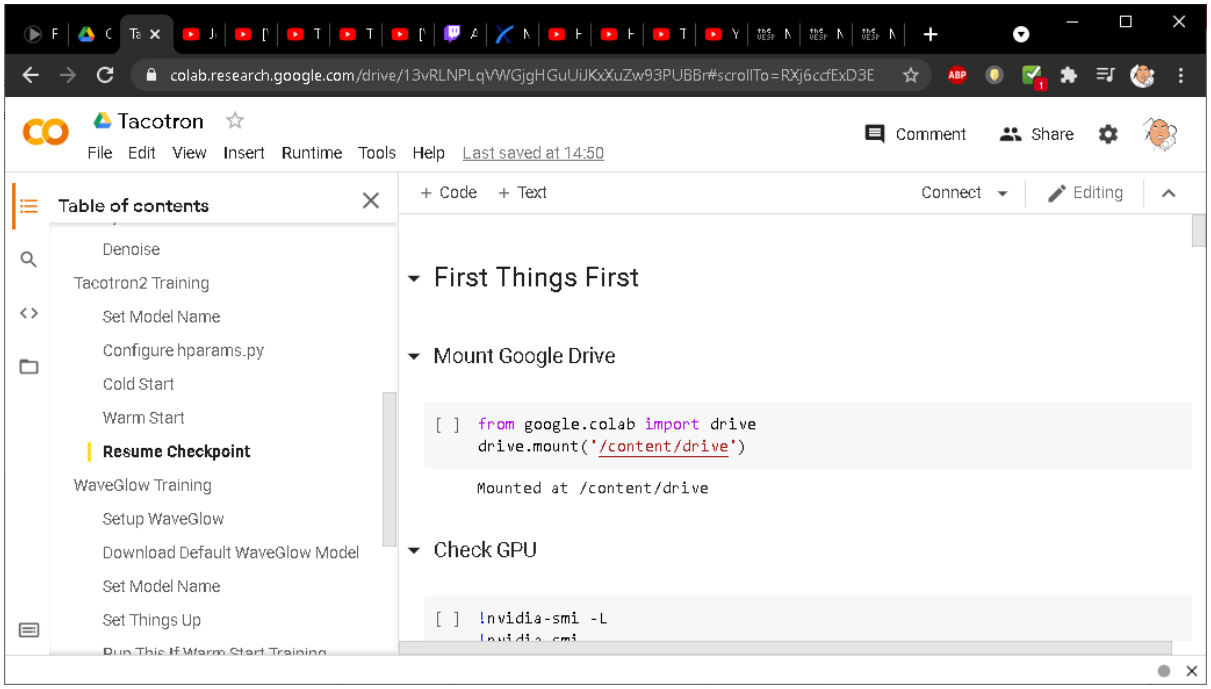


Figure 1348 - Colab Notebook.

The first thing we need to do is go up to Runtime > Change Runtime Type.

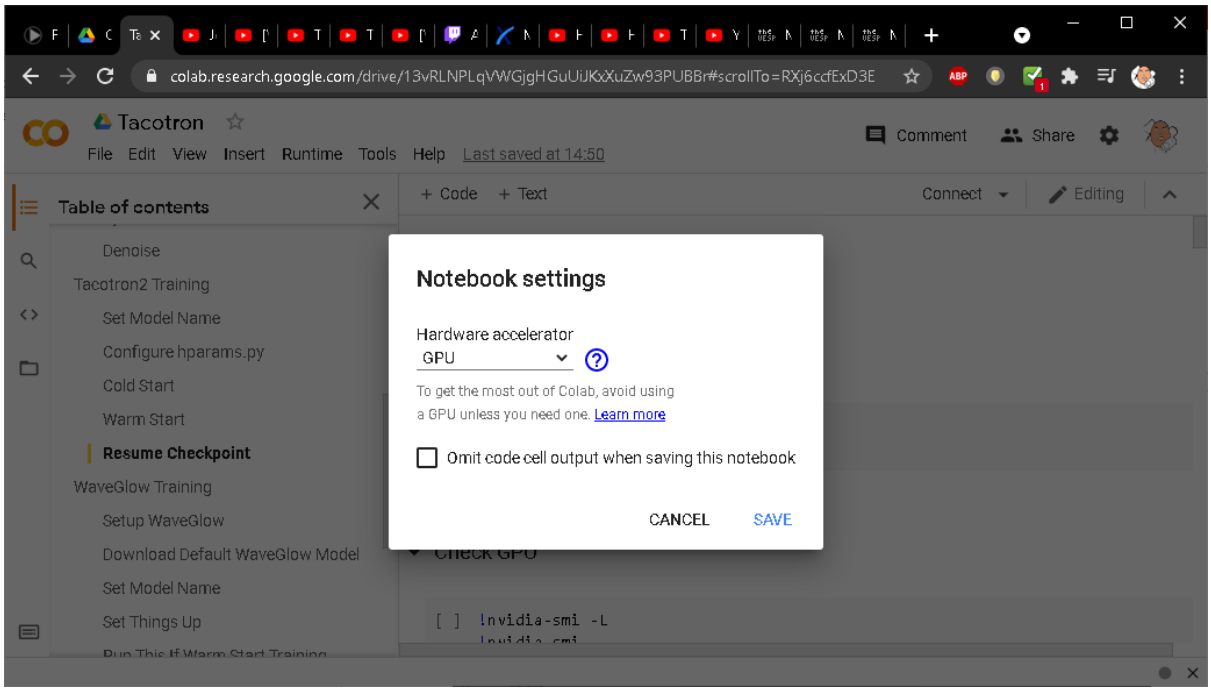


Figure 1349 - Notebook settings.

Make sure 'Hardware accelerator' is set to GPU then click Save.

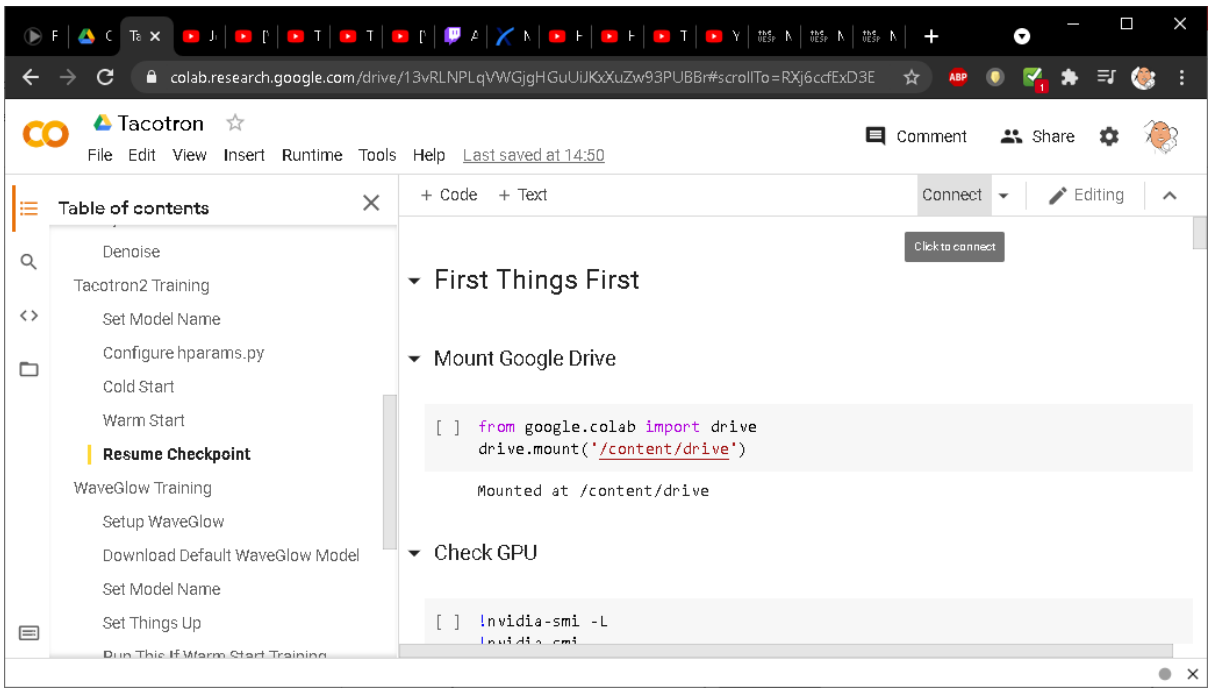


Figure 1350 - Clicking Connect.

Click Connect near the top-right to start a new Colab session.

It's important to note that sessions are temporary. Anything we install or save to our session will be lost once the session is terminated. We can terminate sessions manually by clicking on the down arrow next to the Connect button, selecting Manage Sessions, then click Terminate next to our active session.

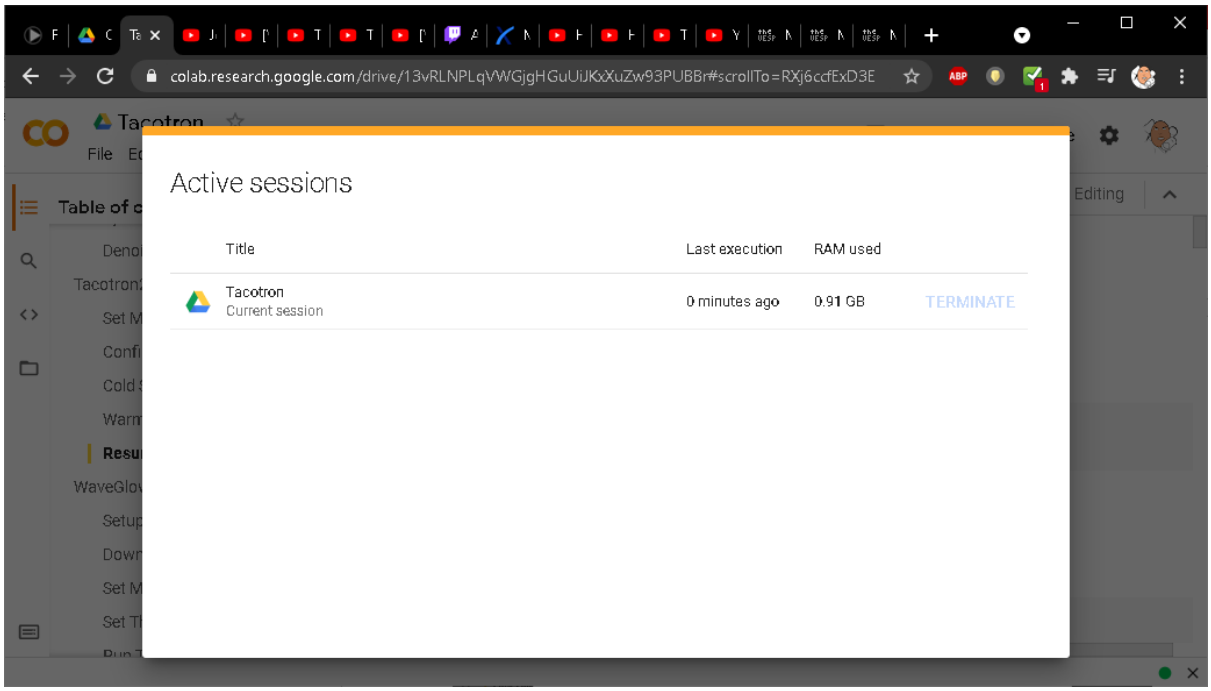


Figure 1351 - Active session.

If Terminate is greyed out, just refresh the Colab tab in your browser then try again.

Important: We're provided free access to a physical GPU. If a GPU session is left idle it will terminate automatically and you will need to reconnect.

It's also important to note that sessions will only persist for a few hours, depending on the overall system usage of the Google Colab service.

Because sessions are temporary we'll need to connect our Colab session to our Google Drive storage so we can both access our dataset and save the checkpoints and log files from Tacotron and WaveGlow back to it.

First, a bit about Colab Notebooks. Colab Notebooks are made up of two types of cells. Text Cells and Code Cells.

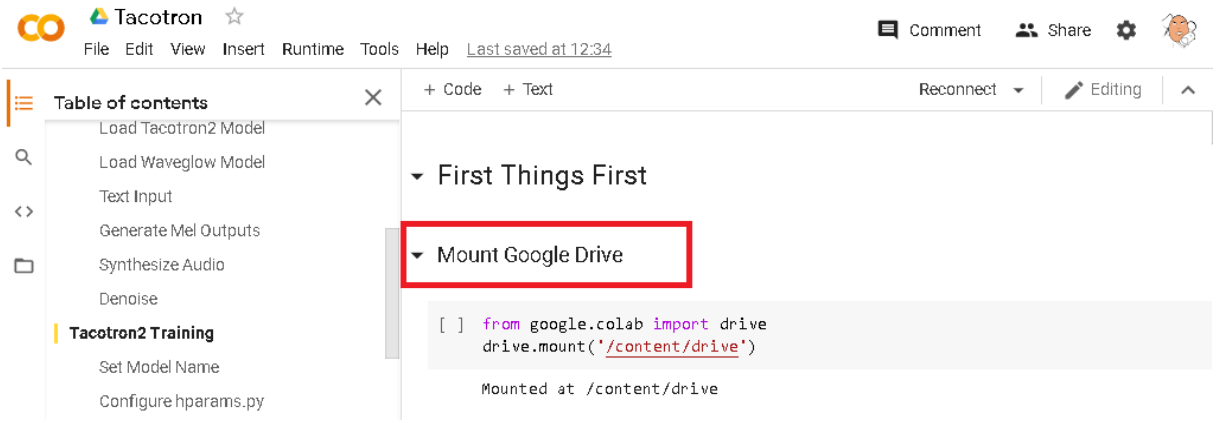


Figure 1352 - A text cell.

Text Cells are used to add labels.

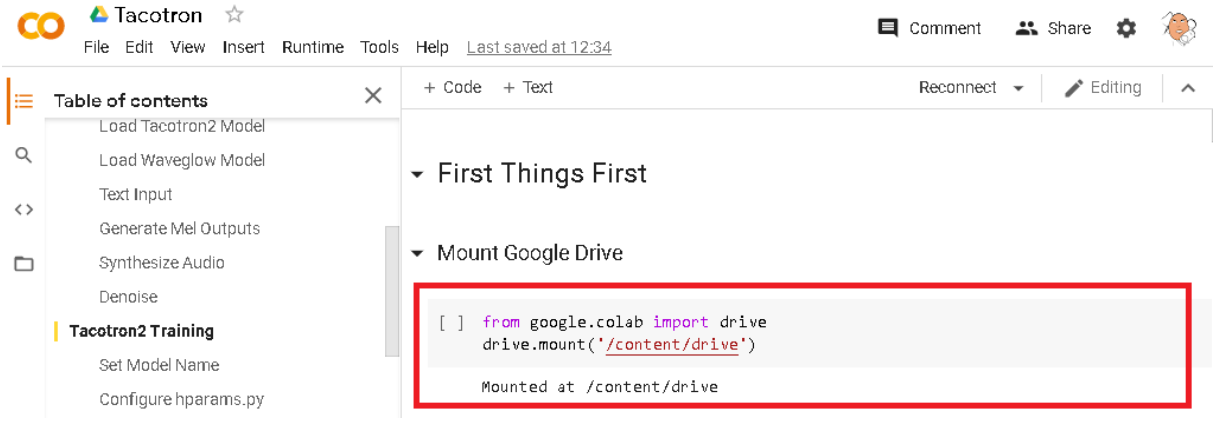


Figure 1353 - A code cell.

Code Cells are used to run commands or python code.

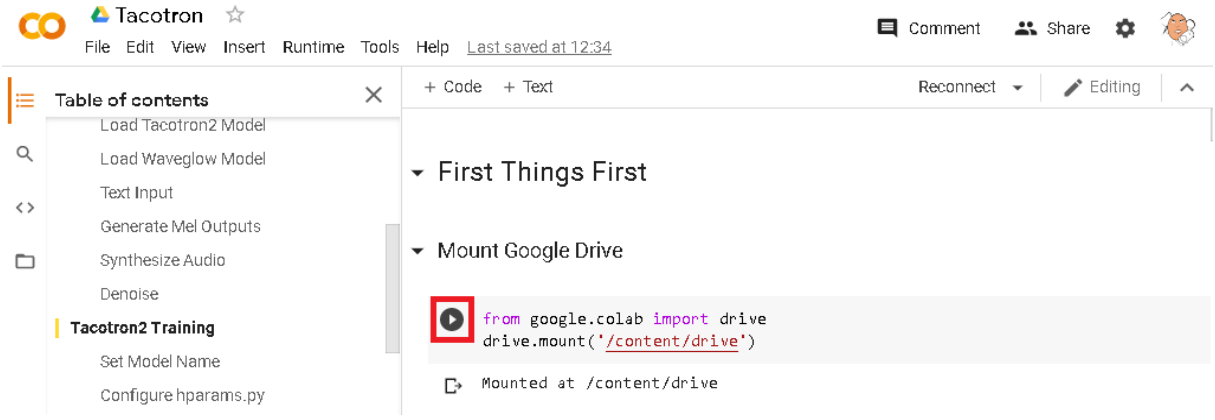


Figure 1354 - Run cell button.

To execute a code cell, click on the **Run Cell** button.

Run the code cell under **Mount Google Drive**.

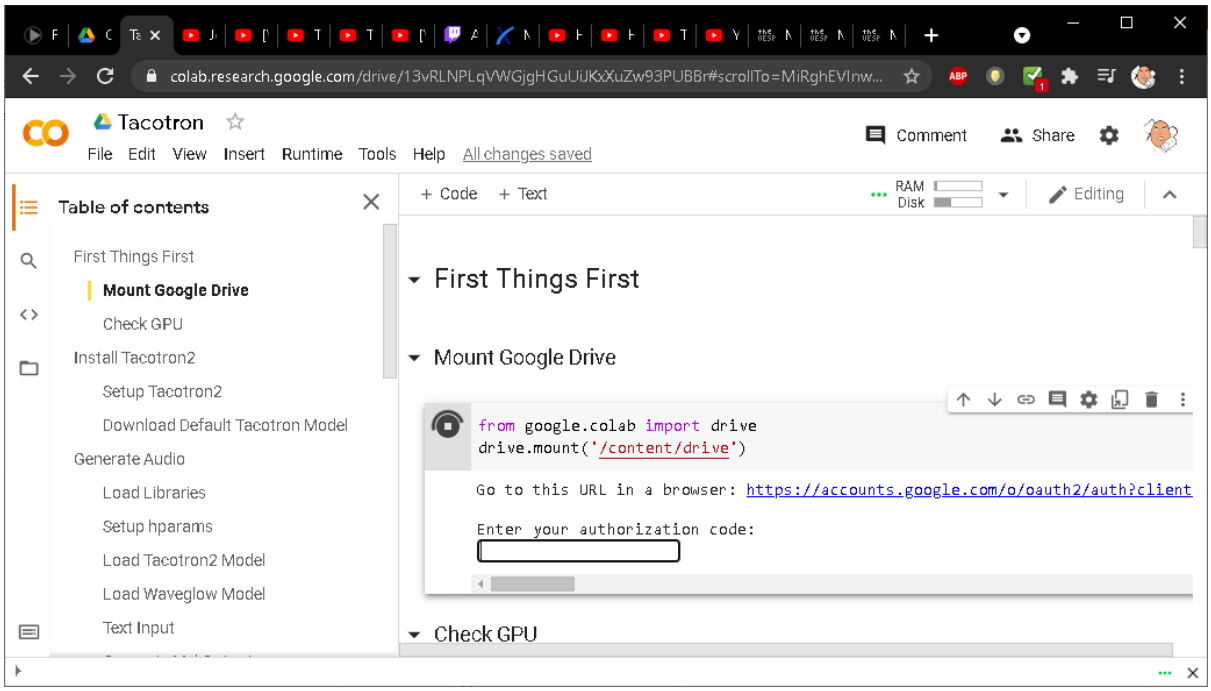


Figure 1355 - Mount Google Drive.

We'll need to verify our connection before we can link our Google Drive storage to our Colab session. Click on the link produced in the cell.

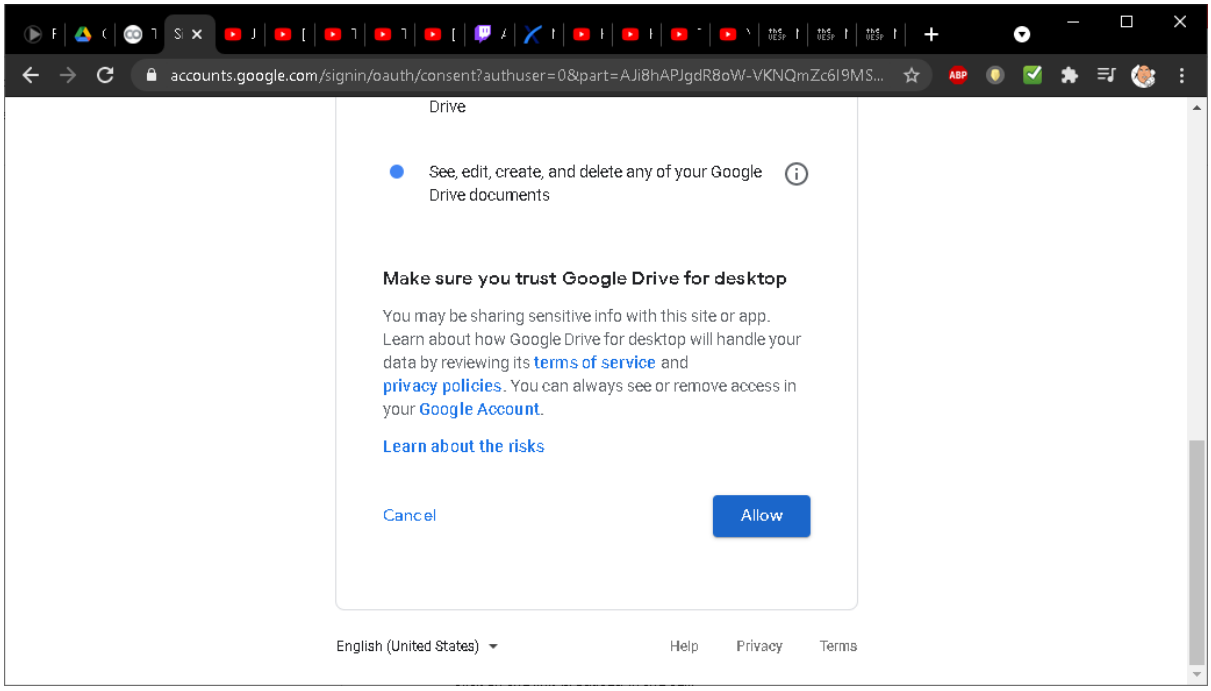


Figure 1356 - Google Drive verification.

Click Allow.

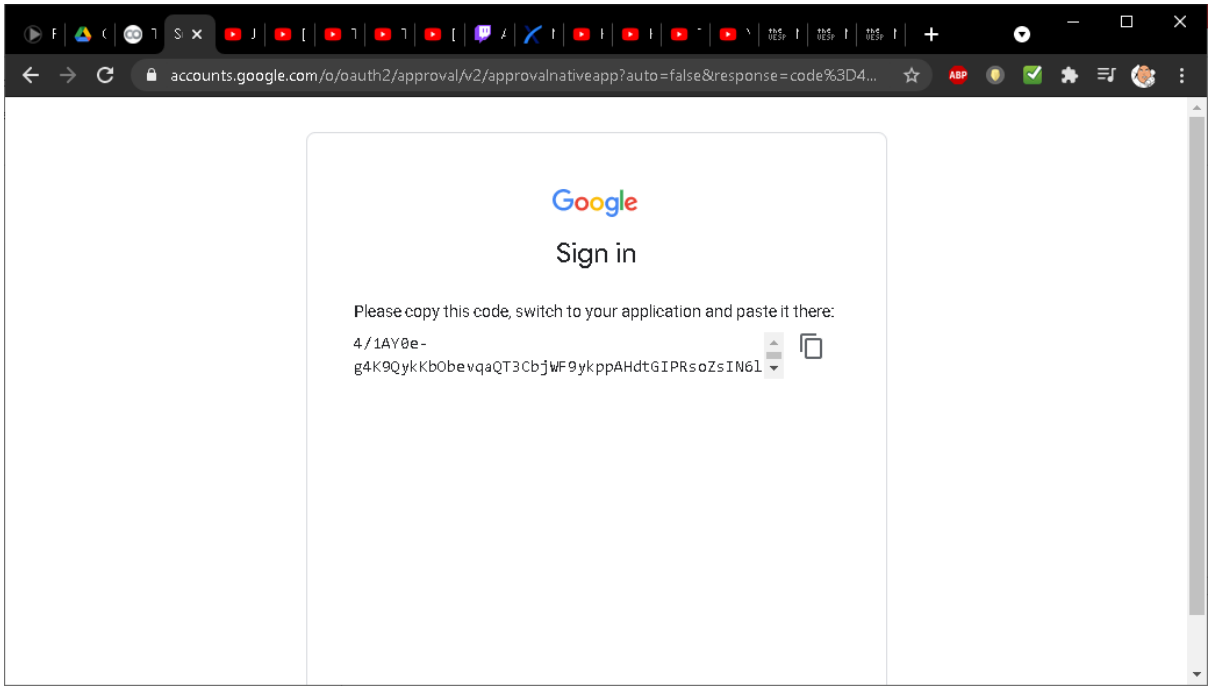


Figure 1357 - Authentication code.

Copy the authentication code and paste it into the text field back in the code cell like so, then press Enter.

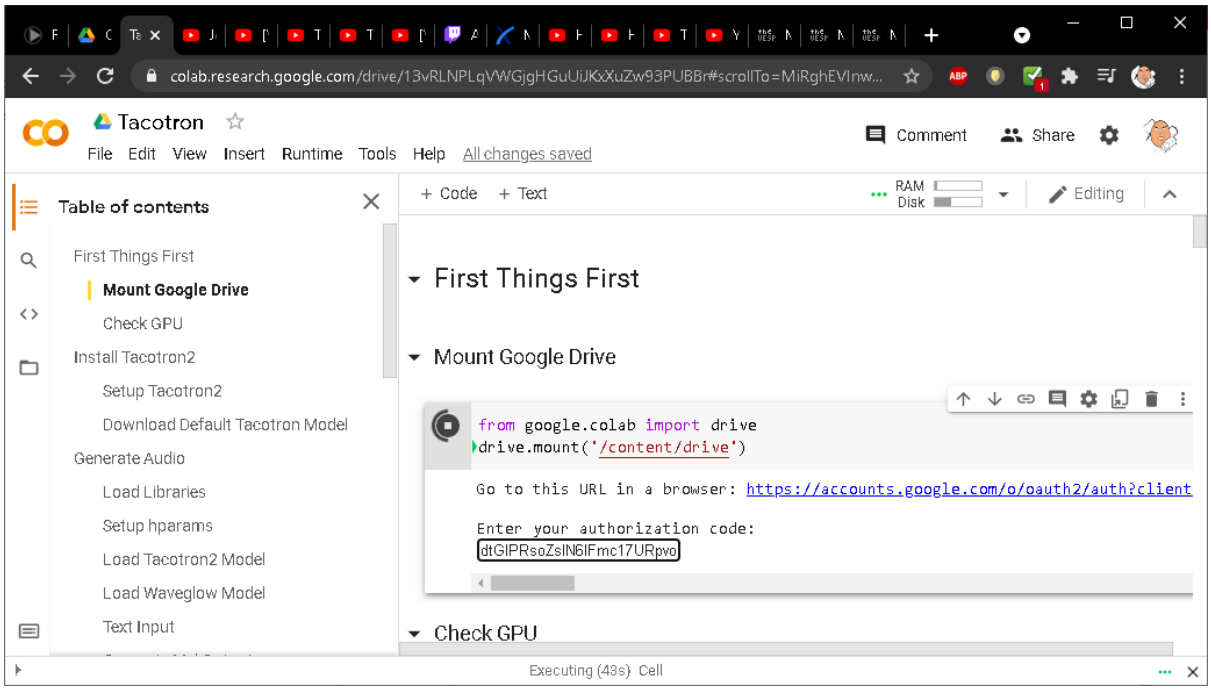


Figure 1358 - Entered the authentication code.

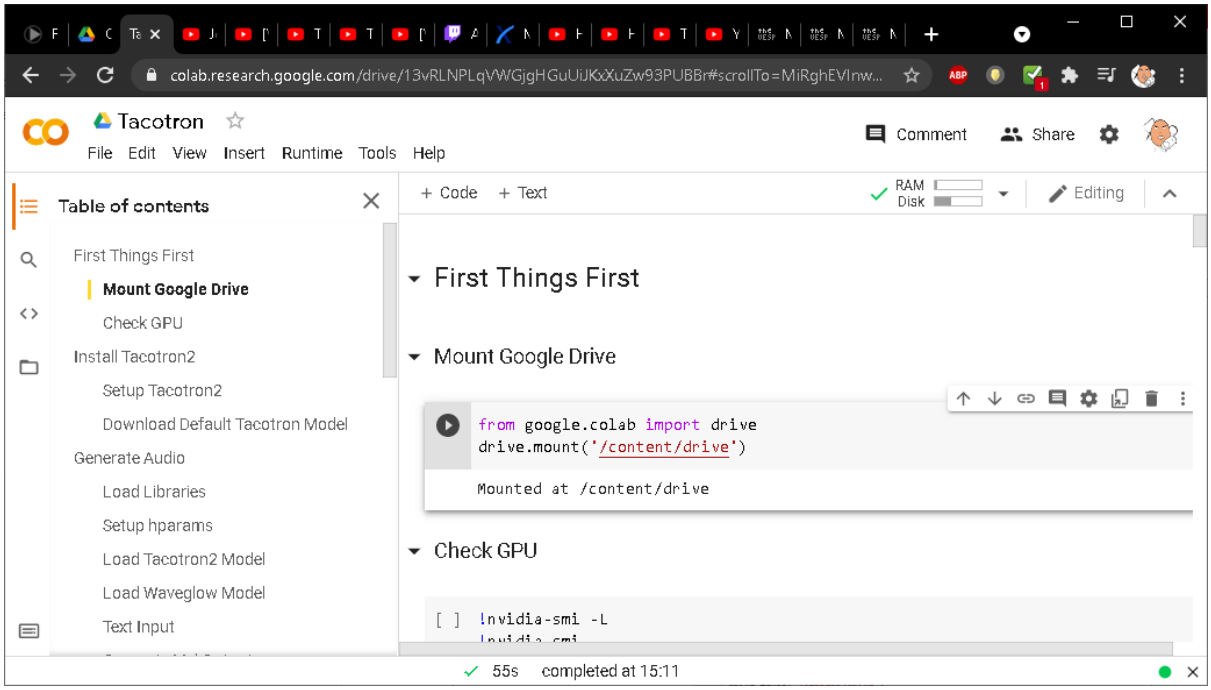


Figure 1359 - Successfully mounted Google Drive storage.

Once it's mounted successfully, the cell will report 'Mounted at /content/drive'.

Run the code cell under **Check GPU**.

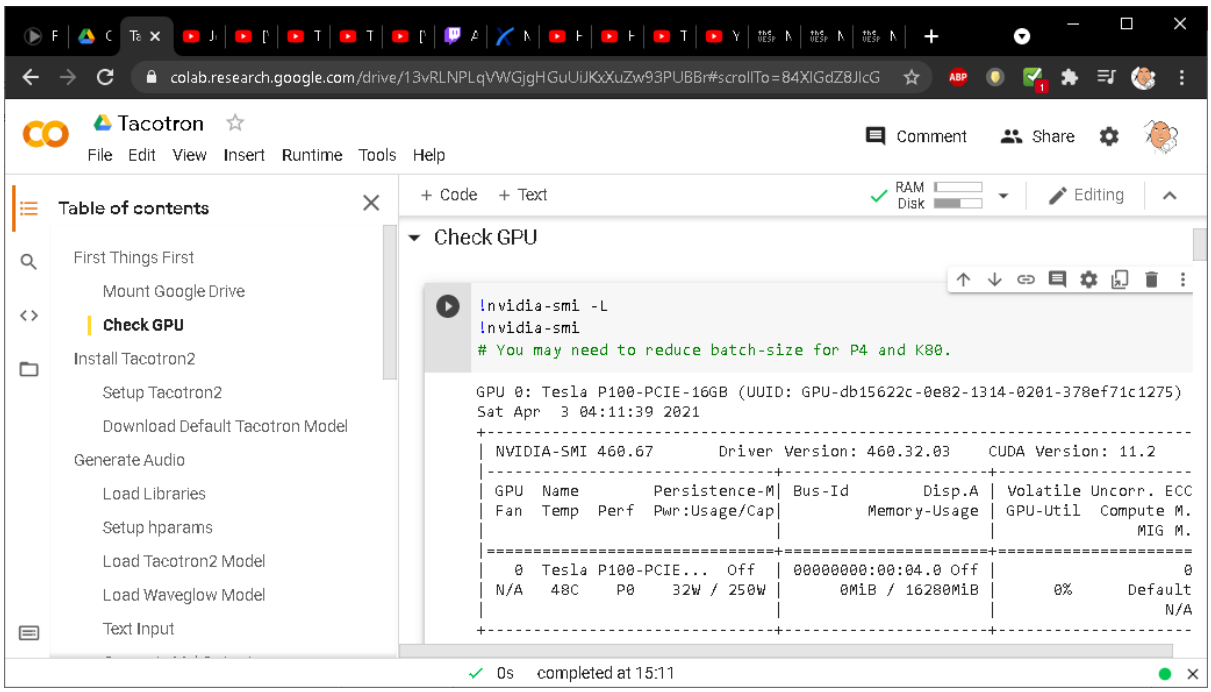


Figure 1360 - Check GPU.

This will tell us what kind of GPU has been assigned to our session. For P4 and K80 GPUs you may need to lower the batch_size settings later on if you run into an 'Out of Memory' error when you start training.

TRAINING A TACOTRON MODEL IN GOOGLE COLAB

Scroll down to the section labelled **Install Tacotron 2**.

Run the code cell under **Setup Tacotron 2**. This will install and Tacotron 2 and its dependencies to your Colab session.

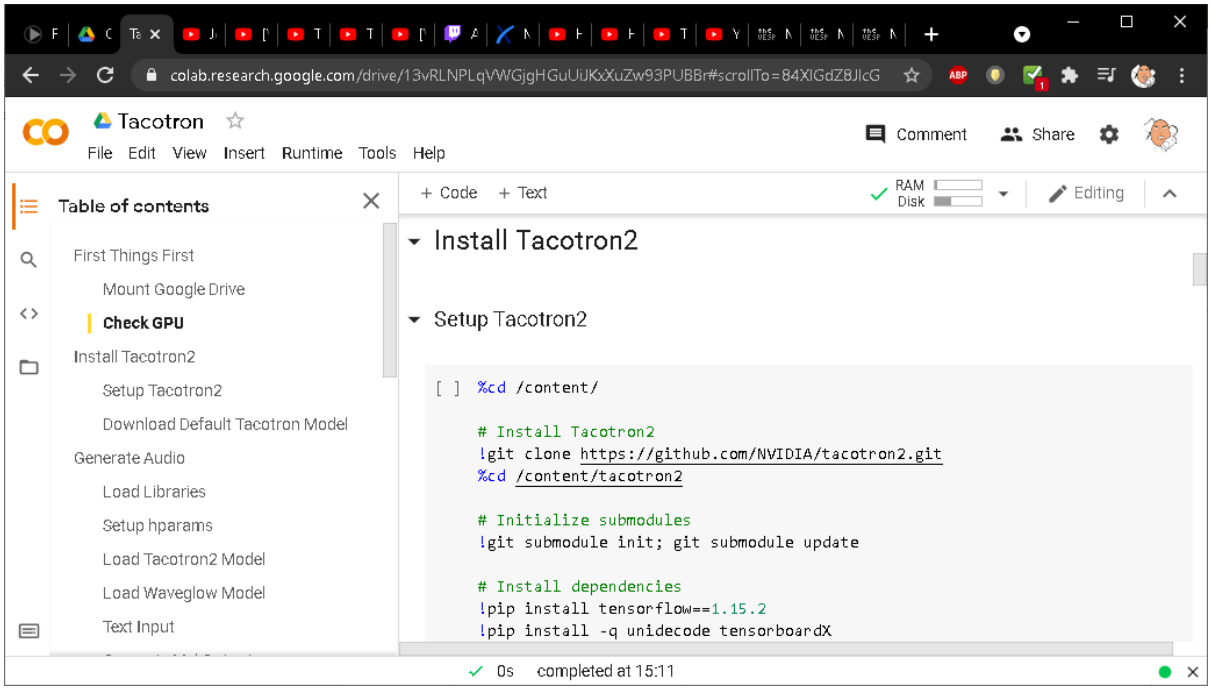


Figure 1361 - Installing Tacotron 2.

Run the code cell under **Download Default Tacotron Model**. This will download the default Tacotron model that we'll be using to warm-start our own model from.

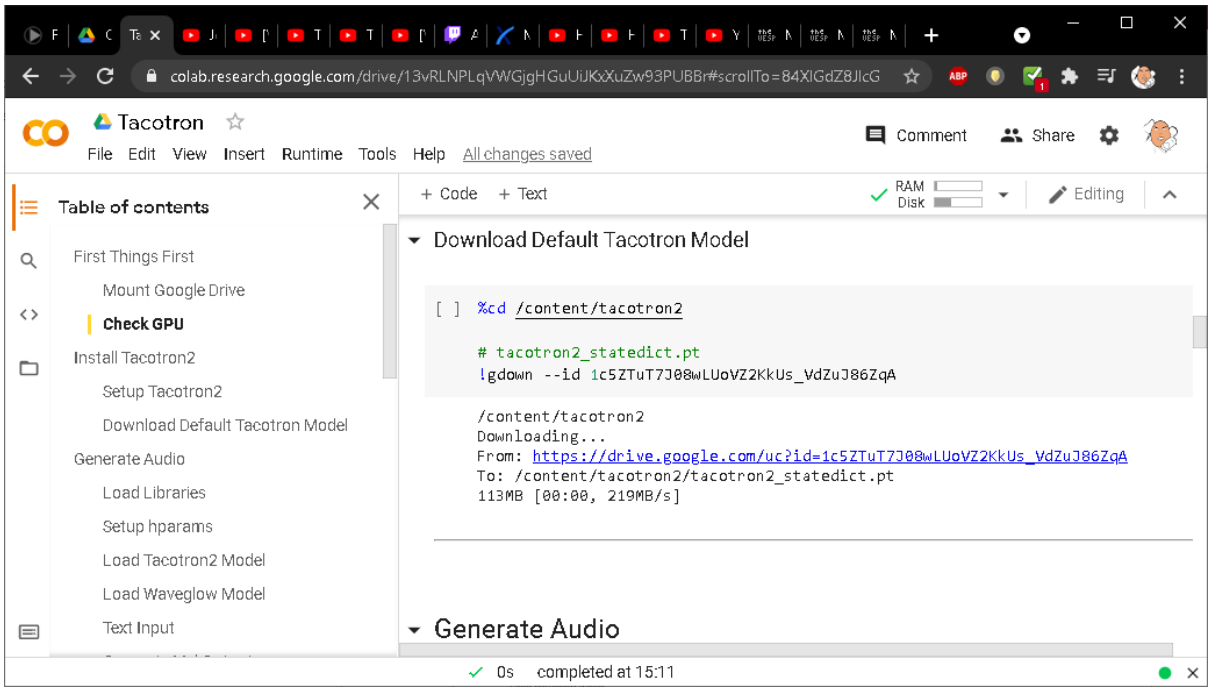


Figure 1362 - Download default Tacotron model.

Scroll down to **Tacotron 2 Training**.

In the code cell beneath **Set Model Name**, change 'tacotron_femalenord' to match the voice type of the dataset you've prepared.

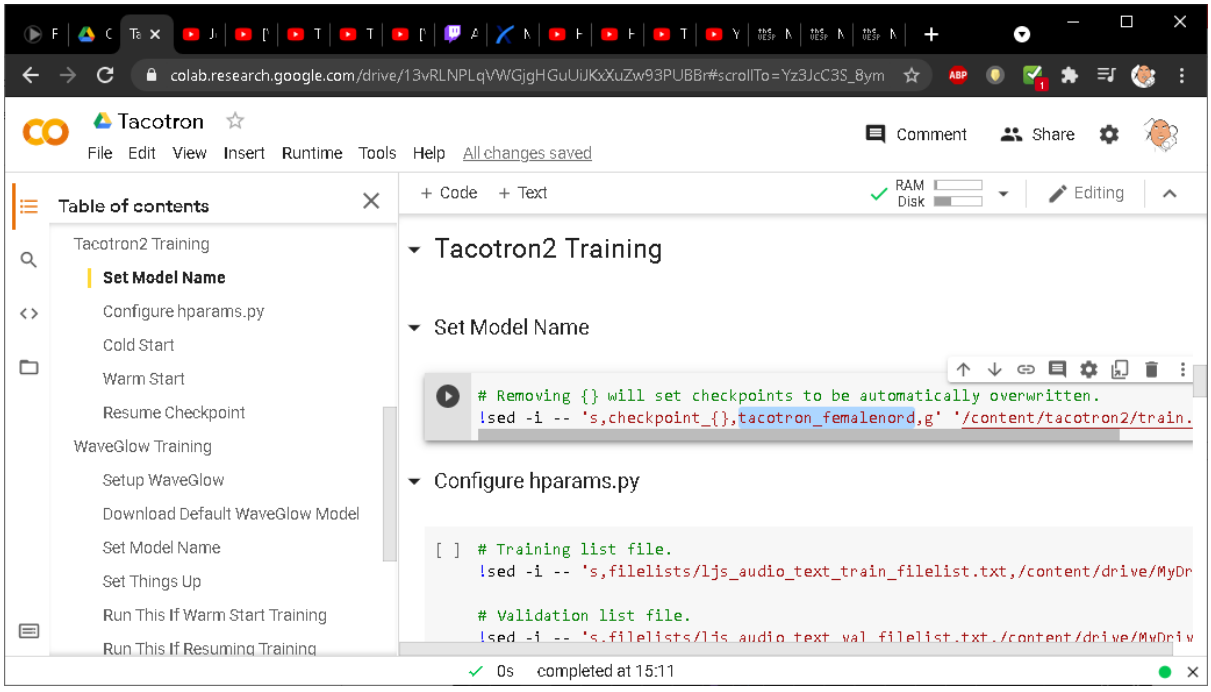


Figure 1363 - Set model name.

By default Tacotron saves individual checkpoints as it trains as 'checkpoint_{}'. Removing '{}' from the model name tells Tacotron to just keep one checkpoint and overwrite it when it needs to save a new one.

Google Drive is limited to 15GB of storage space and this can be used up pretty quickly when training a Tacotron or WaveGlow model if we save checkpoints individually.

Click Run Cell once you've changed the model name.

Scroll down to **Configure hparams.py**.

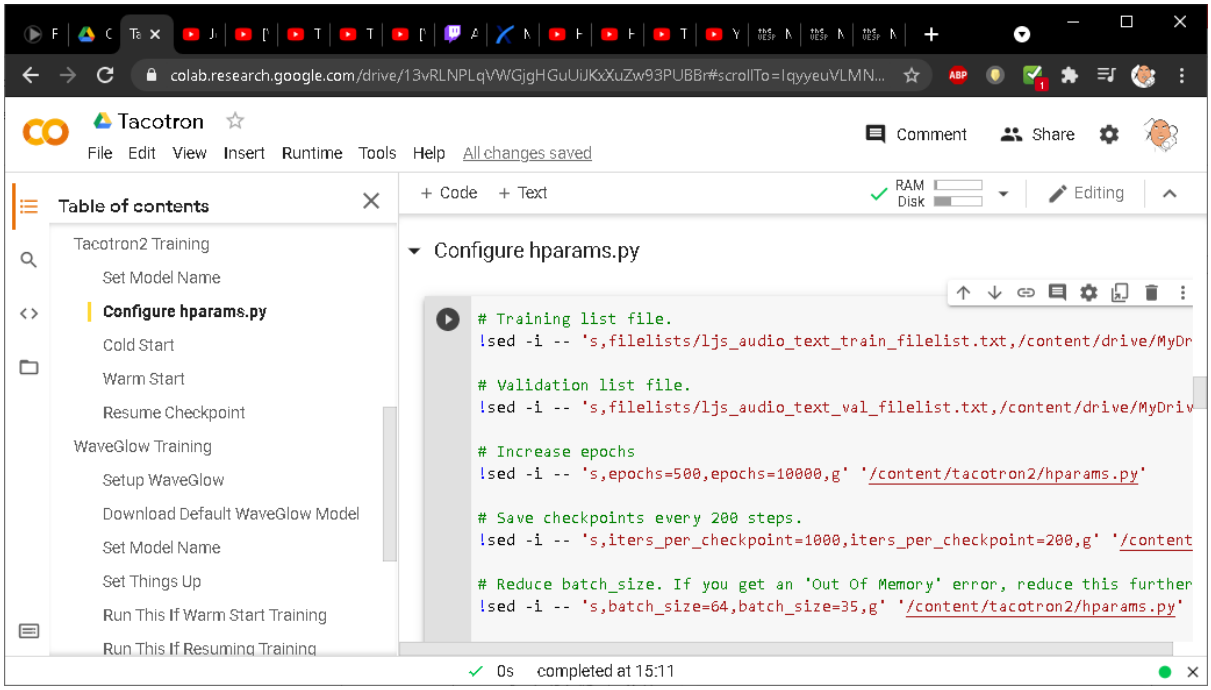


Figure 1364 - Configure hparams.py.

Set the path to your training and validation text files. By default, I have the training file path set to '/content/drive/MyDrive/filelists/femalenord_training.txt' and the validation file path set to '/content/drive/MyDrive/filelists/femalenord_validation.txt'.

The other settings can be left as is for now.

Click Run Cell once you've changed the file names for the training and validation files.

Scroll down to **Warm Start**.

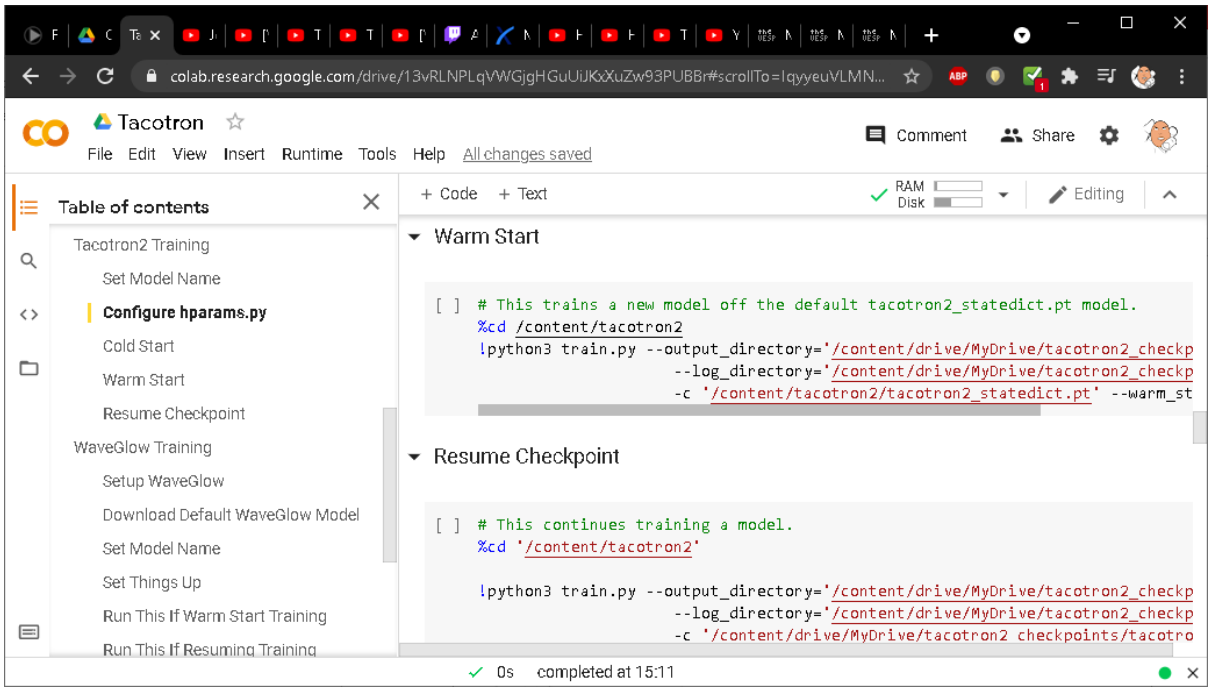


Figure 1365 - Warm Start.

Warm Start allows us to begin training a new model off the default Tacotron model. This is useful for small datasets like the ones we make from Skyrim or Oblivion.

Models will be saved to a new folder in Google Drive called 'tacotron2_checkpoints'. The log directory will be saved there as well.

Click Run Cell to begin training a new model.

Assuming you don't run into any errors, you should see a new model in the 'tacotron2_checkpoints' folder after 200 iterations. Every 200 iterations your progress will be saved, overwriting the existing checkpoint.

To resume training an existing model, scroll down to **Resume Checkpoint**.

Note: You'll need to make sure you've run **Setup Tacotron2**, **Set Model Name** and **Configure hparams.py** first before resuming.

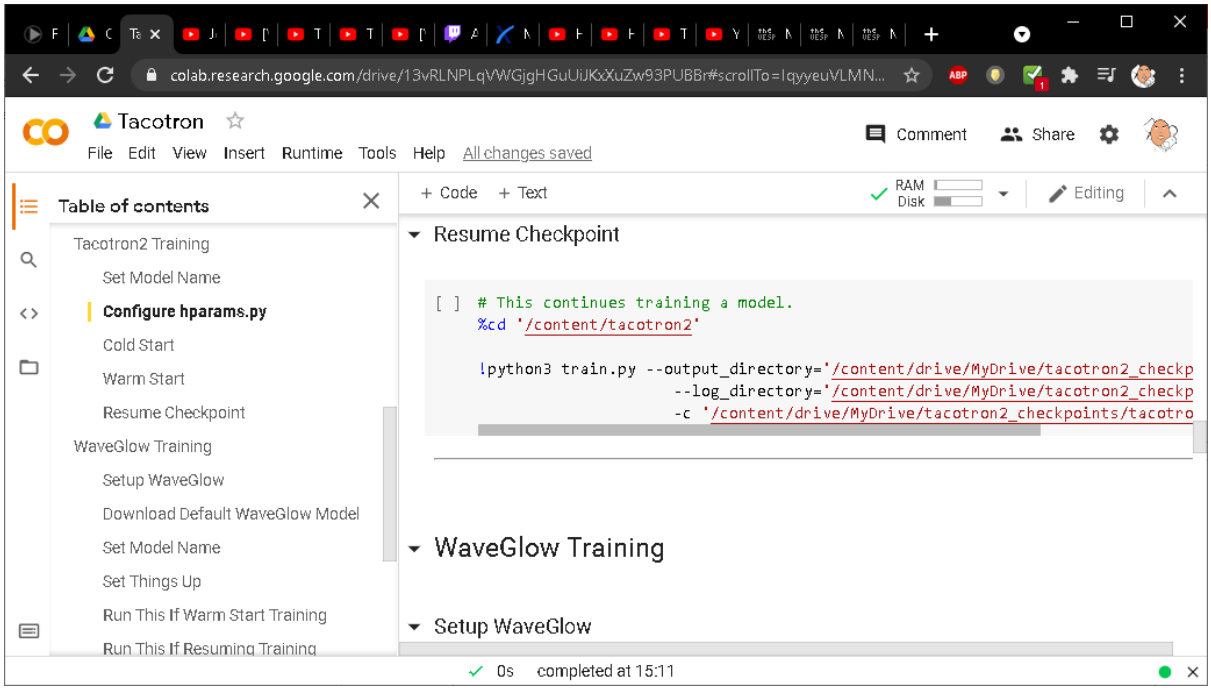


Figure 1366 - Resume Checkpoint.

For the -c switch, change the name of the model and ensure the path to it is correct. By default I have this switch set to '/content/drive/MyDrive/tacotron2_checkpoints/tacotron_femalenord'.

Click Run Cell to resume training a model.

TRAINING A WAVEGLOW MODEL IN GOOGLE COLAB

Scroll down to the section labelled **Install WaveGlow**.

Run the code cell under **Setup WaveGlow**.

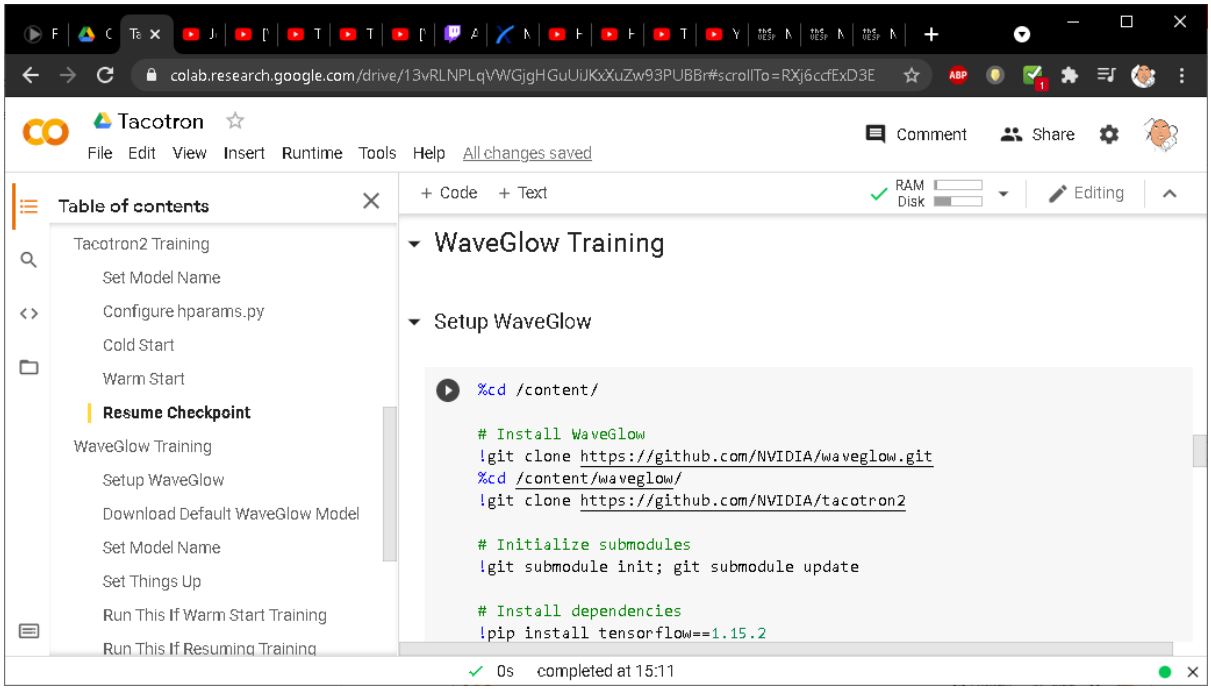


Figure 1367 - Setup WaveGlow.

Run the code cell under **Download Default WaveGlow Model**. This will download the default WaveGlow model that we'll be using to warm-start our own model from.

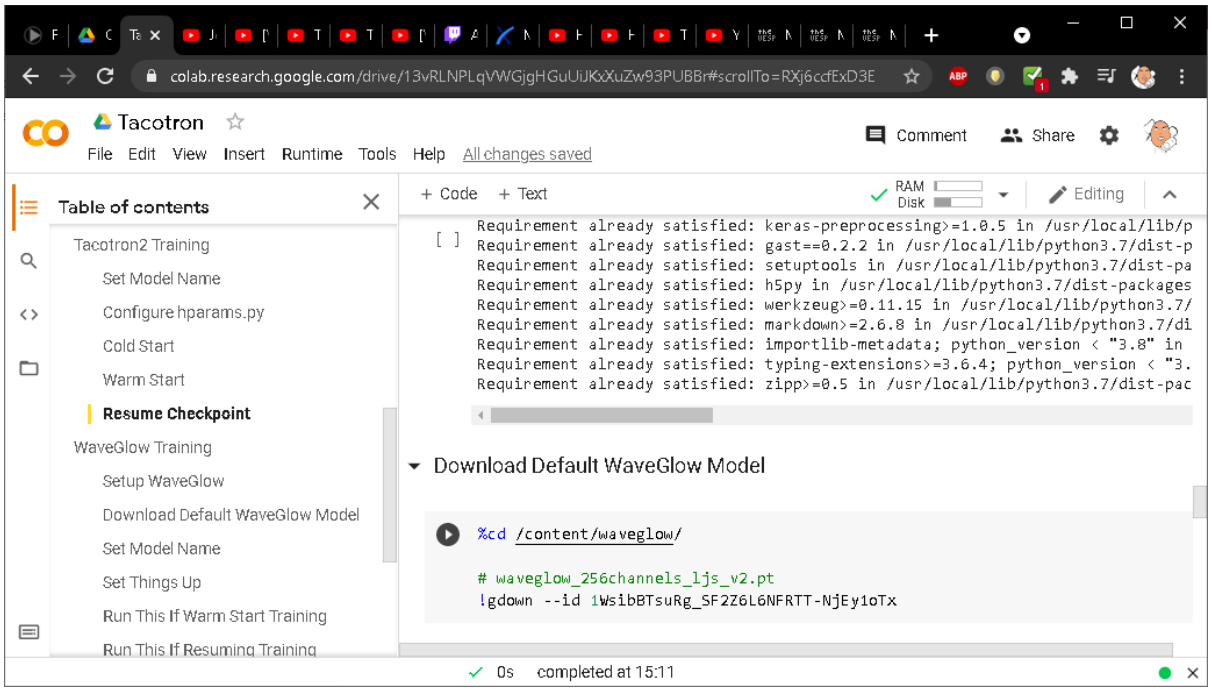


Figure 1368 - Download Default WaveGlowModel.

In the code cell beneath **Set Model Name**, change 'waveglow_femalenord' to match the voice type of the dataset you've prepared.

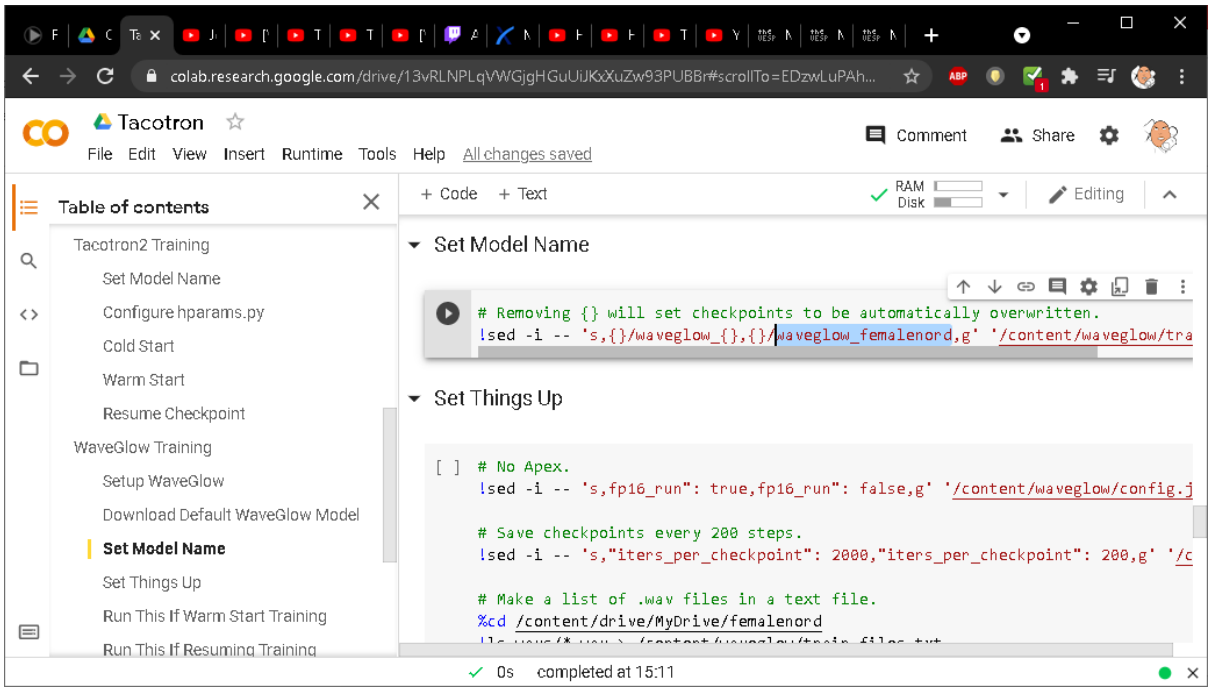


Figure 1369 - Set Model Name.

By default WaveGlow saves individual checkpoints as it trains as 'waveglow_{}'. Removing '{}' from the model name tells WaveGlow to just keep one checkpoint and overwrite it when it needs to save a new one.

Note: Google Drive is limited to 15 GB of storage space and this can be used up pretty quickly when training a Tacotron or WaveGlow model if we save checkpoints individually.

Click Run Cell once you've changed the model name.

Scroll down to **Set Things Up**.

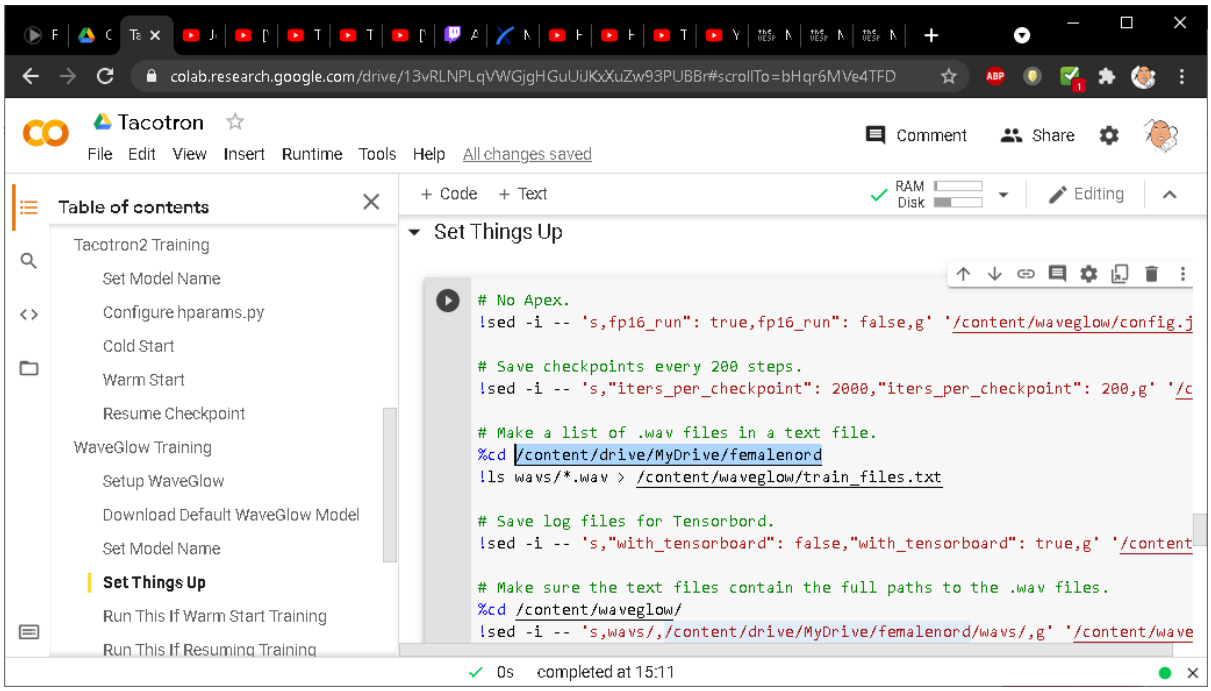


Figure 1370 - Set Things Up.

Set the path to the folder containing your .wav files in the areas highlighted in the screenshot above.

Click Run Cell once you've changed those paths.

We won't need the training and validation files we made earlier for Tacotron. This cell will create its own train_files.txt file for WaveGlow to use. It will also tell WaveGlow to save checkpoints to the 'waveglow_checkpoints' folder back in Google Drive. The log directory will be saved there as well.

If you're training a new model, run the first two code cells under **Run This If Warm Starting**.

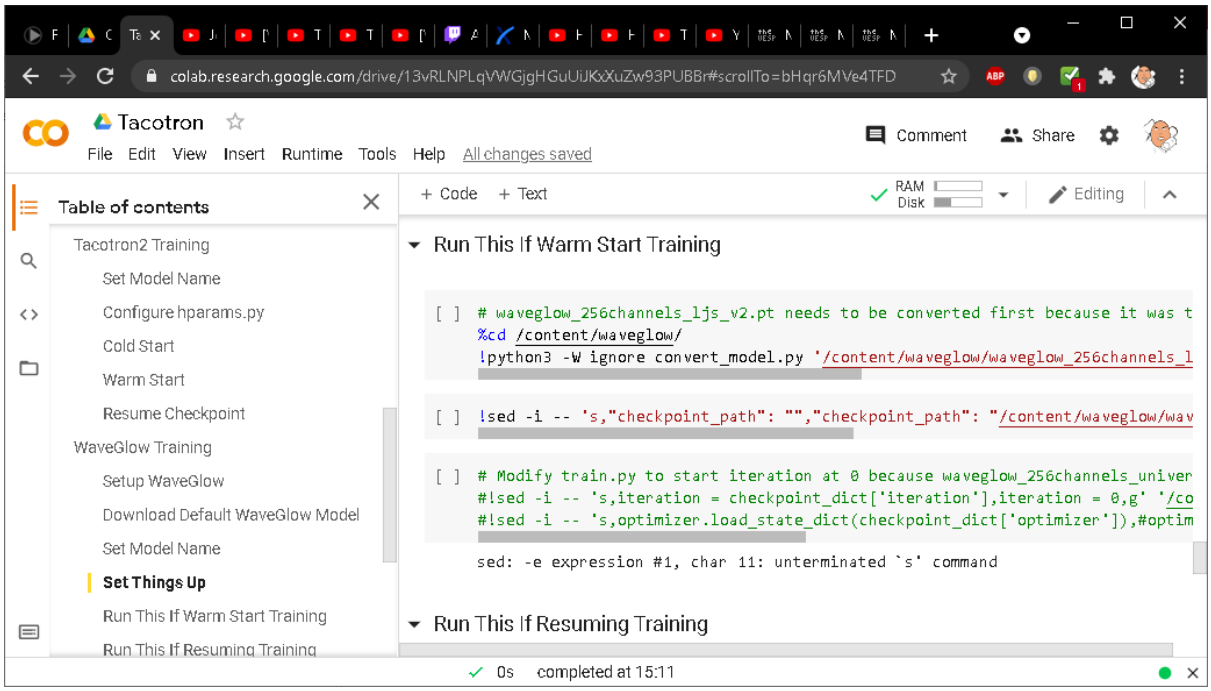


Figure 1371 - Run This If Warm Starting.

This will convert the default WaveGlow model 'waveglow_256channels_ljs_v2.pt' to allow us to warm start from it.

Click on Browse on the left hand side. Navigate to content > waveglow and open train.py.

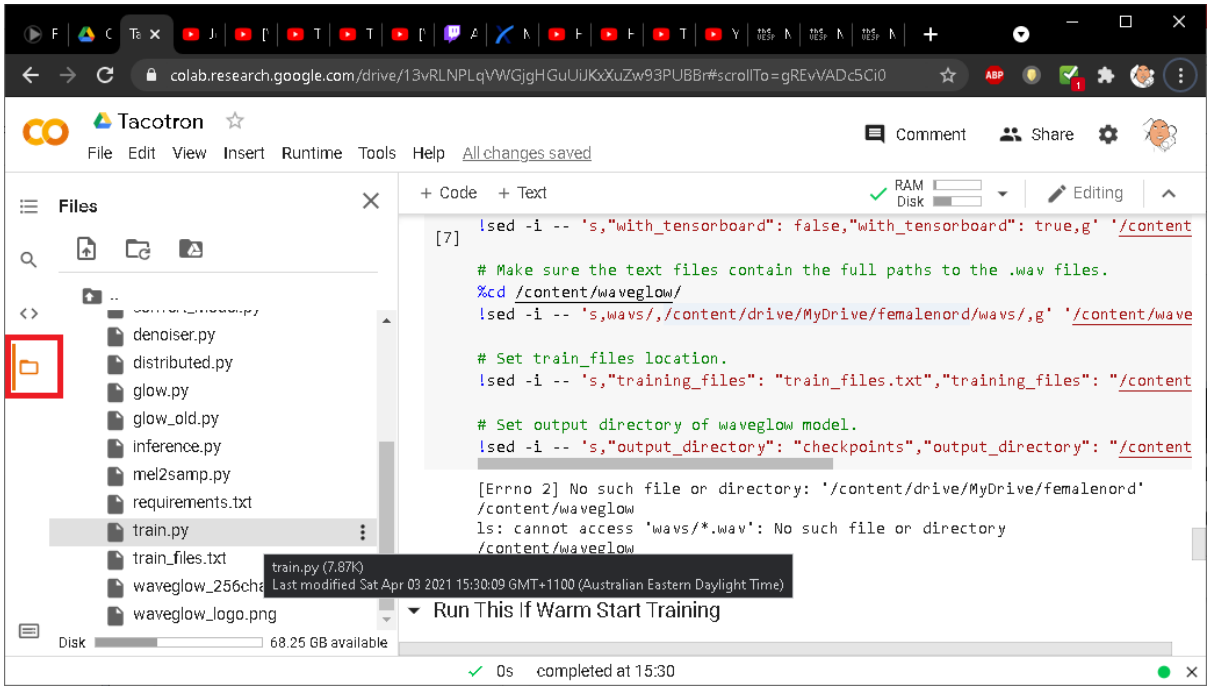


Figure 1372 - Opening train.py.

Scroll down to line 44 in train.py. Change

```
iteration = checkpoint_dict['iteration']
```

to

```
iteration = 0
```

Comment out line 45 by adding a '#' in front.

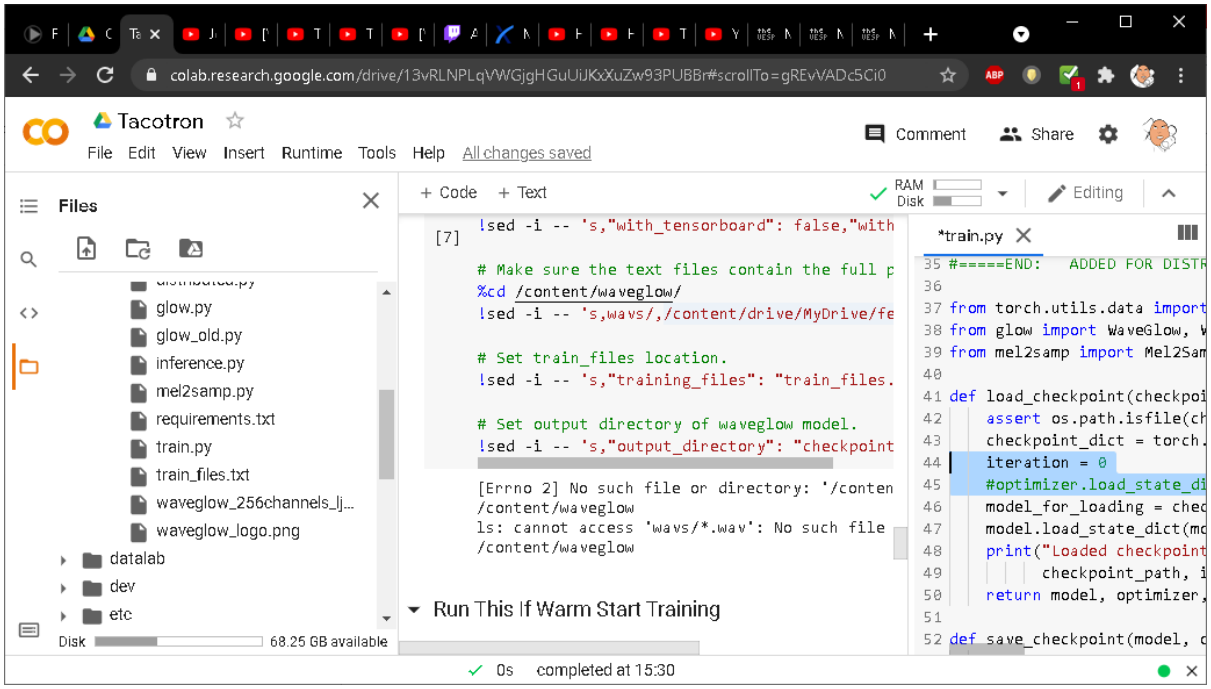


Figure 1373 - train.py modified.

Press CTRL+S to save the document, then close train.py.

This will allow us to use the default WaveGlow model we just converted.

Scroll down to **Start Training**.

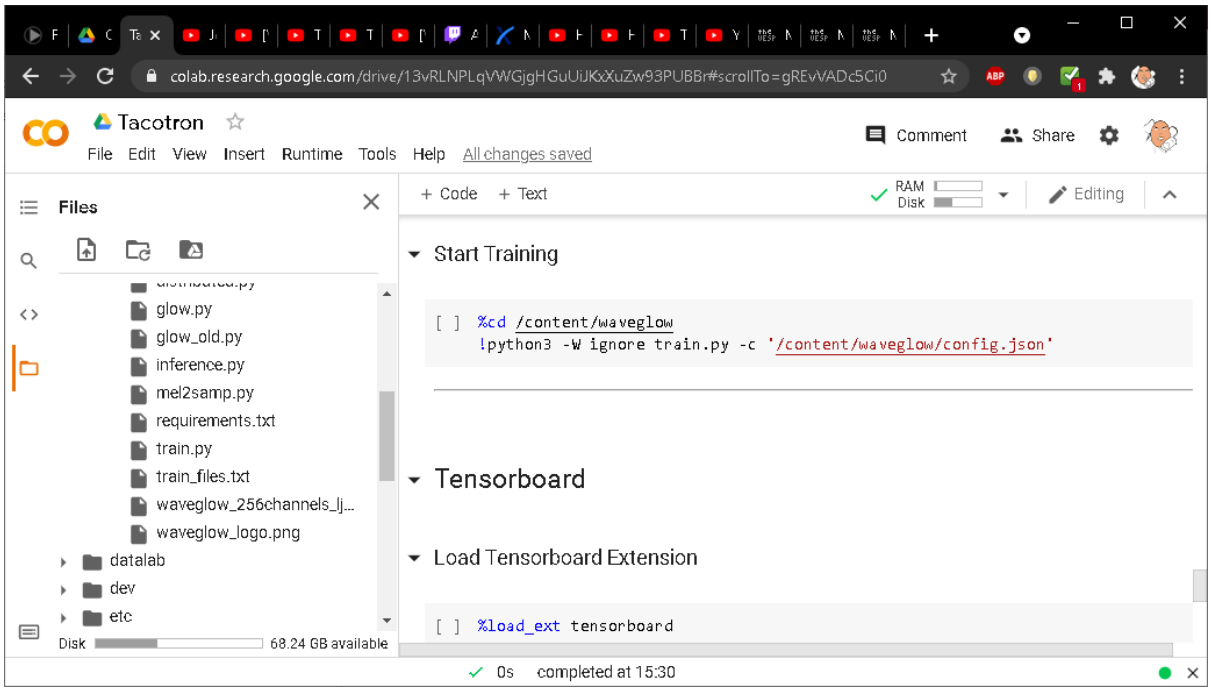


Figure 1374 - Start Training.

Run the cell below it to begin training a new WaveGlow model.

Assuming you don't run into any errors, you should see a new model in the 'waveglow_checkpoints' folder after 200 iterations. Every 200 iterations your progress will be saved, overwriting the existing checkpoint.

To resume training an existing model, scroll down to **Run This If Resuming Training**.

Note: you'll need to make sure you've run **Setup WaveGlow**, **Set Model Name**, and **Set Things Up** first.

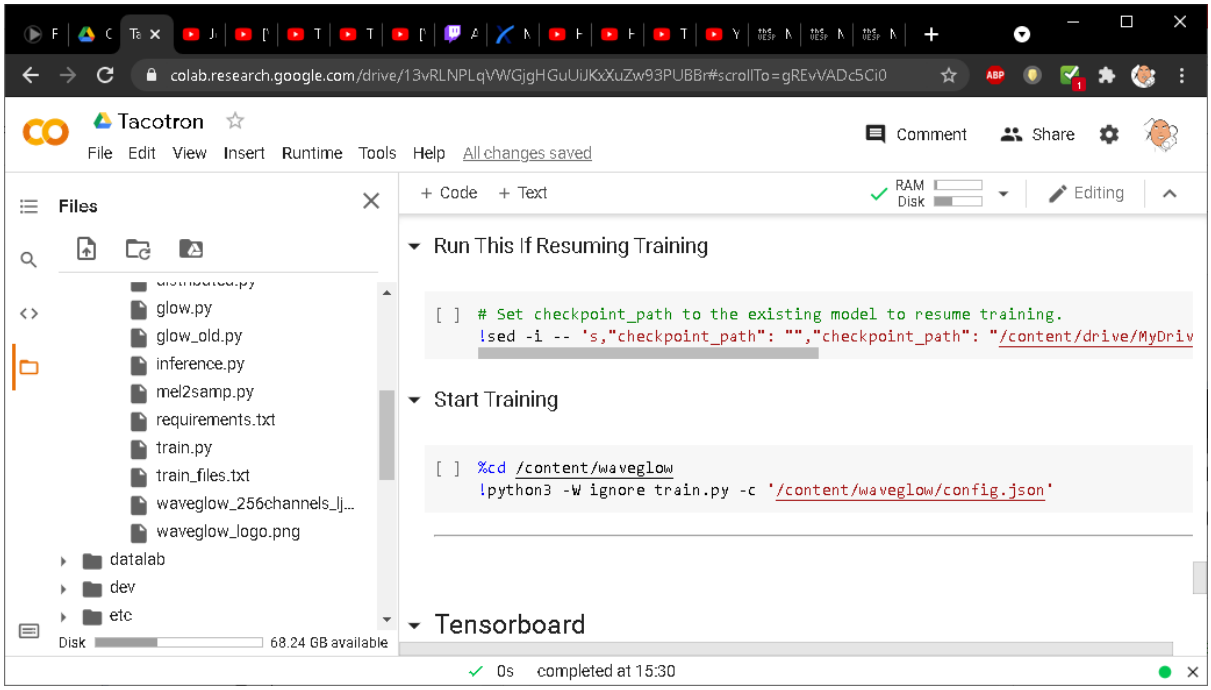


Figure 1375 - Run This If Resuming Training.

By default this cell is set to resume training the waveglow_femalenord model, so change this to match the file name of the model you're training, then click Run Cell.

Now scroll down to **Start Training** and click Run Cell to resume training a model.

Training will continue from the last saved checkpoint.

CHECKING PROGRESS IN TENSORBOARD

We can check the progress of our Tacotron and WaveGlow model in Tensorboard.

Tensorboard will read the log files from the logdir folder that was saved to our checkpoint directories. In our example we saved our tacotron checkpoints to 'tacotron_checkpoints'. For WaveGlow, that folder is 'waveglow_checkpoints'.

I would recommend testing your models after every 10,000 iterations just to check that things are on the right track. See the next section titled **Synthesizing audio using the models we've trained** for the steps on how to generate audio to do this.

Generally speaking, training is considered 'done' once we're satisfied with the output.

Scroll down to **Tensorboard** and run the code cell under **Load Tensorboard Extension**.

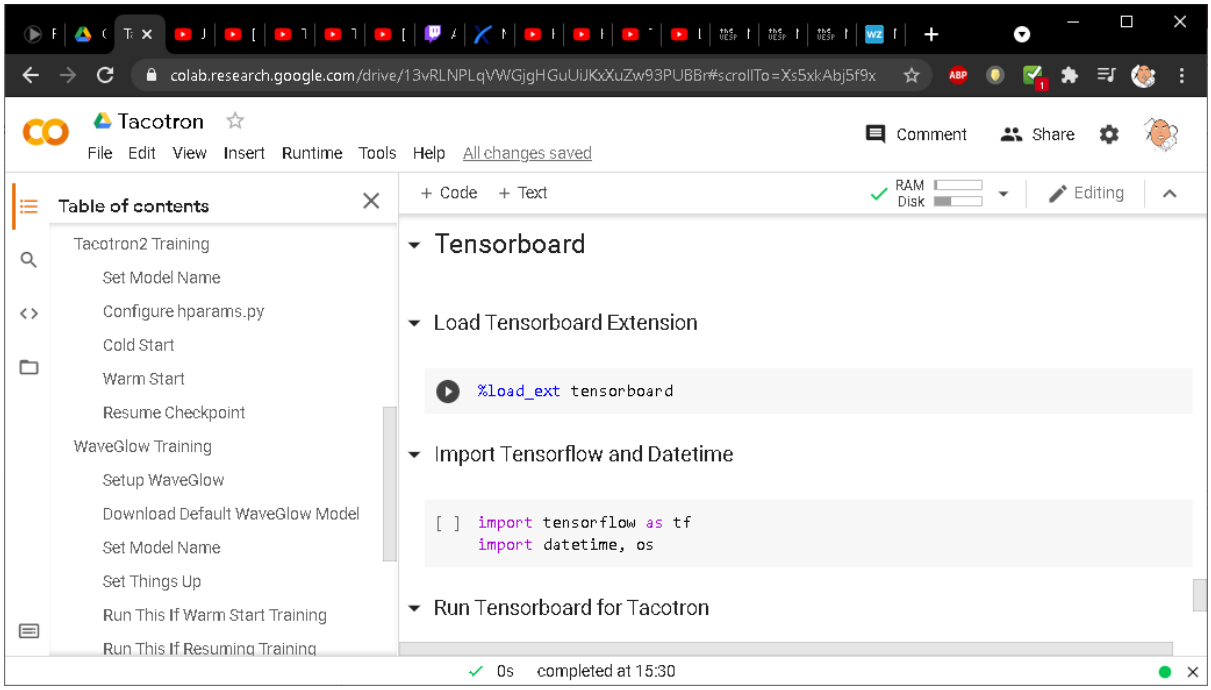


Figure 1376 - Load Tensorboard Extension.

Next, run the code cell under **Import Tensorflow and Datetime**.

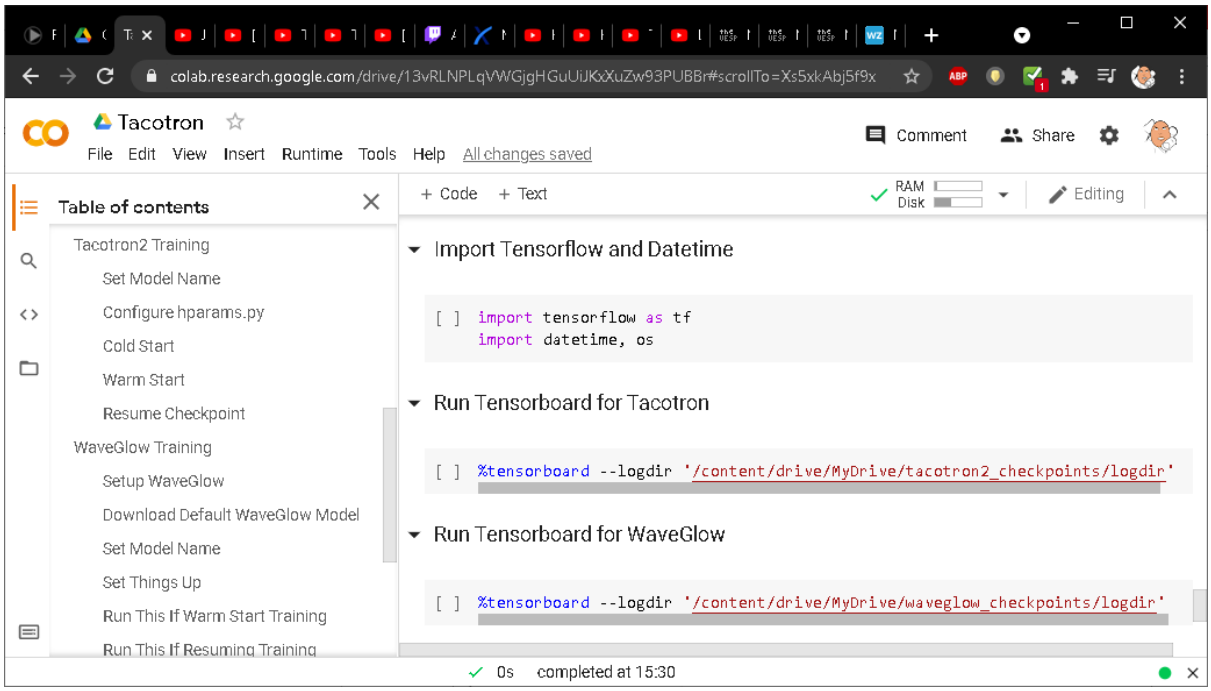


Figure 1377 - Import Tensorflow and Datetime.

Run either the code cell under **Run Tensorboard for Tacotron** or **Run Tensorboard for WaveGlow** depending on which one you want to check.

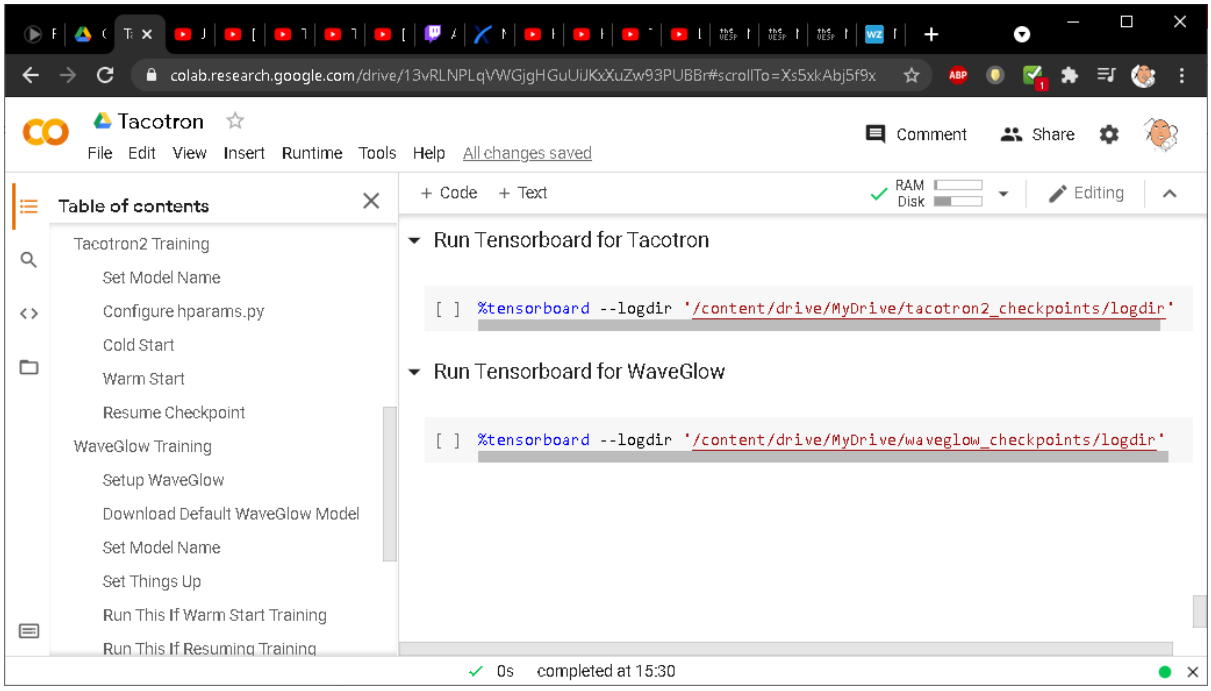


Figure 1378 - Run Tensorboard.

In the Scalars section, the graphs we need to pay attention to are the training validation loss graphs.

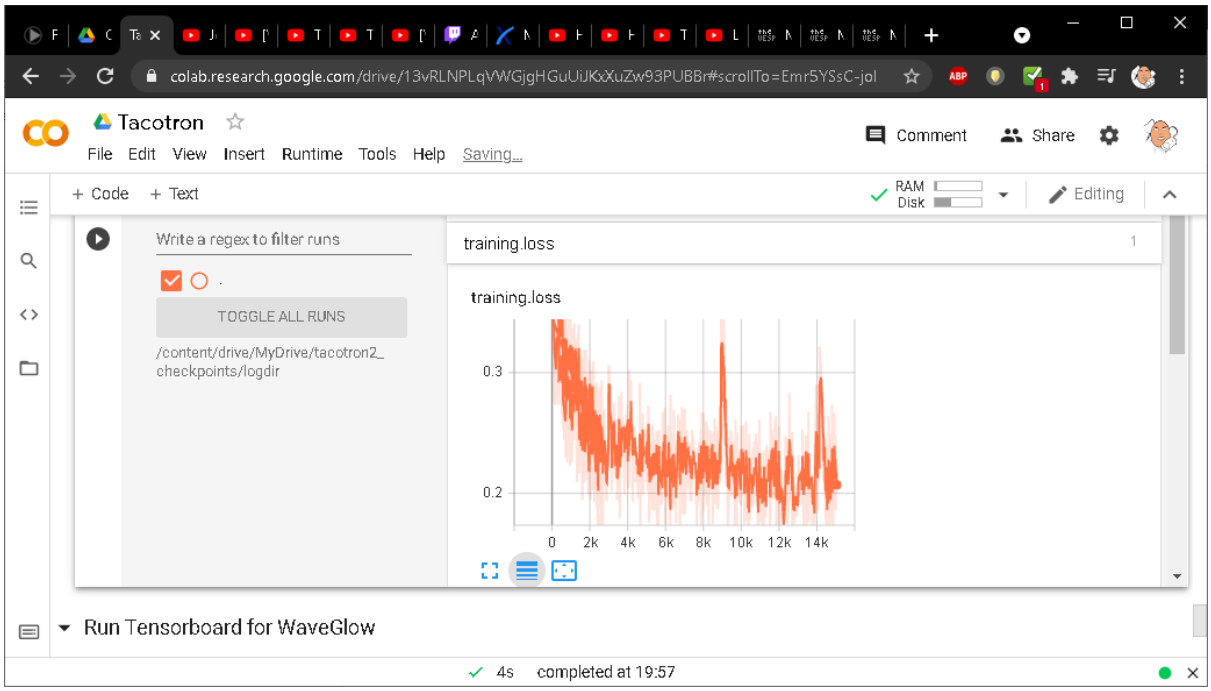


Figure 1379 - Tensorboard Scalars.

For Tacotron, once the training and validation loss curves flatten out, you should reduce the learning rate.

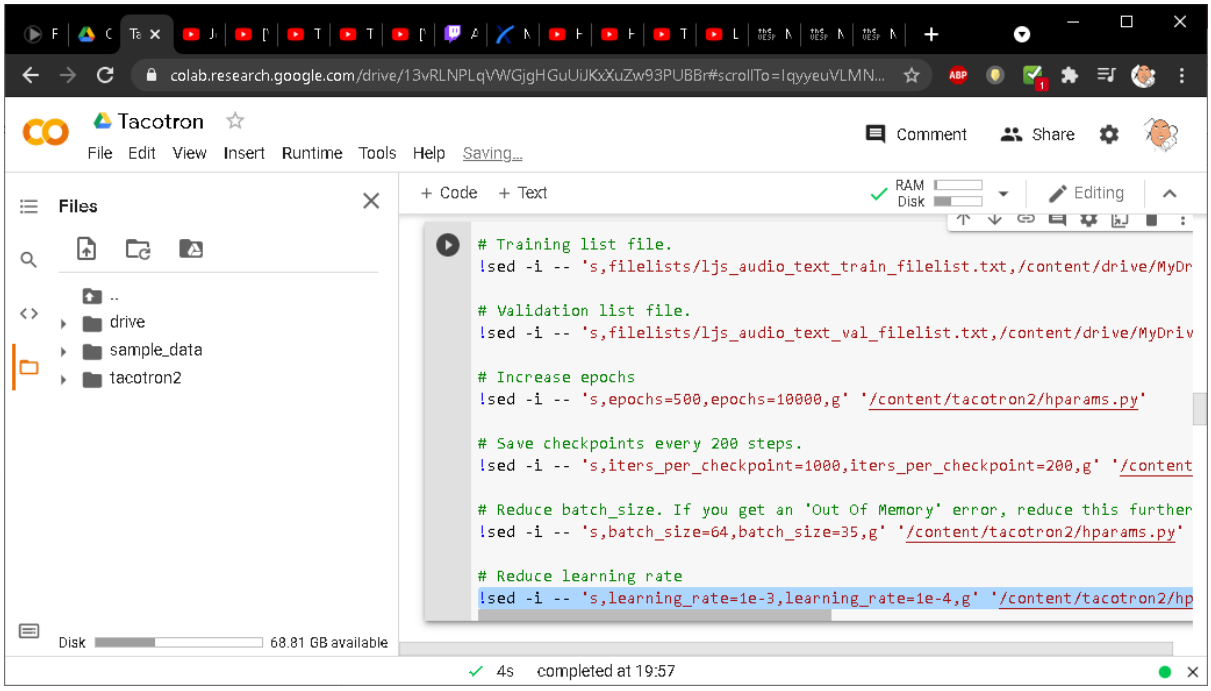


Figure 1380 - Reducing learning rate.

You can do this back in the **Configure hparams.py** section by uncommenting the last time then running that code cell again.

Generally, for datasets based on Skyrim voice acting, I train for 20,000 iterations before lowering the learning rate then train for another 5,000 to 10,000 iterations. Beyond that I don't notice much improvement.

You can check the alignment in the Images section.

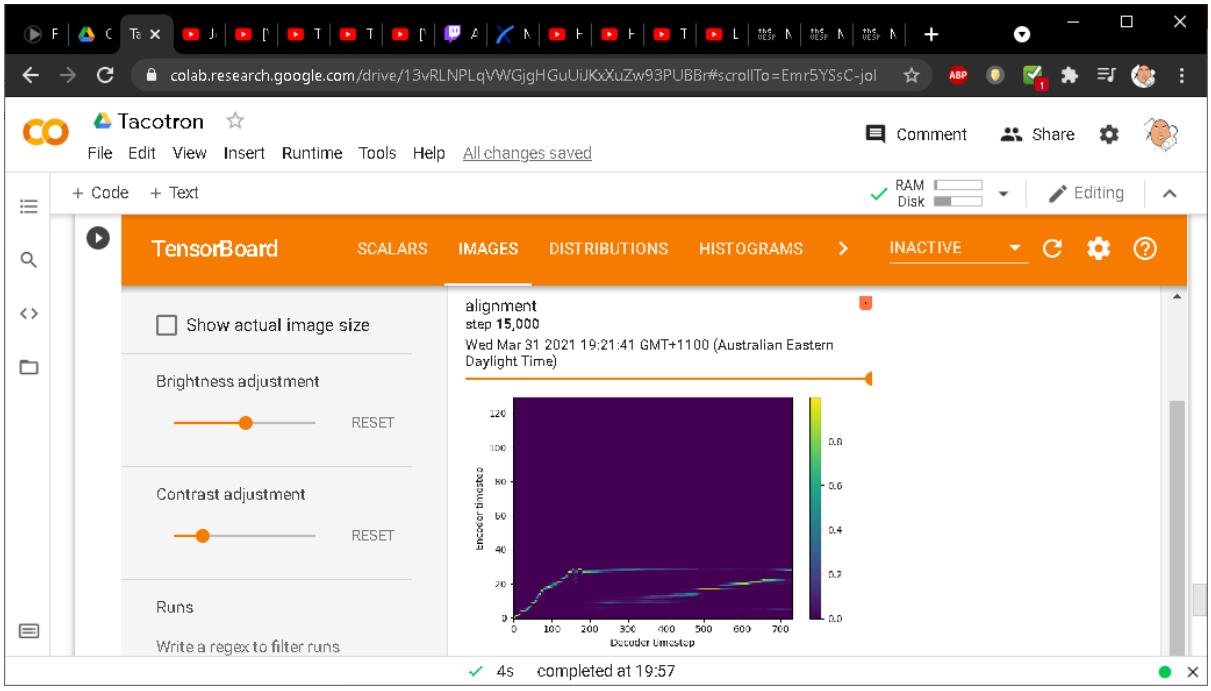


Figure 1381 - Alignment.

Alignment indicates how well a sound generated by the decoder matches a character read by the encoder.

Good alignment will look more like a horizontal line from the bottom left to the top right.

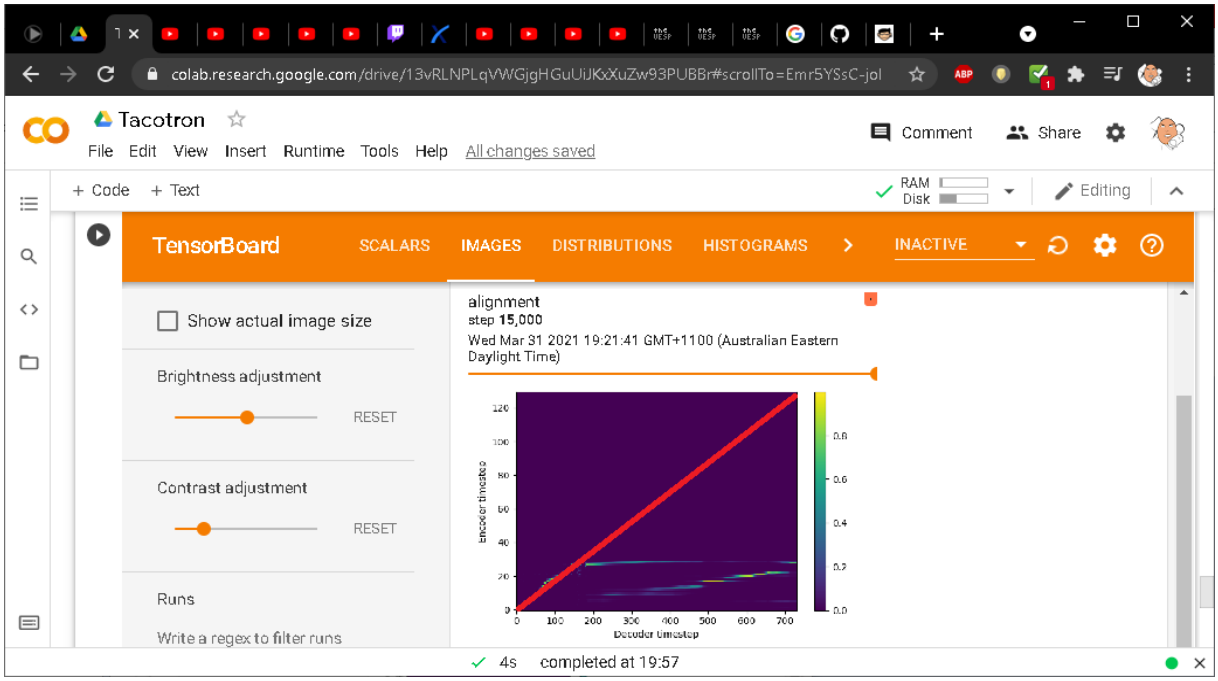


Figure 1382 - What good alignment should look like.

SYNTHESIZING AUDIO FROM THE MODELS WE'VE TRAINED

Scroll down to the section labelled **Install Tacotron 2**.

Run the code cell under **Setup Tacotron 2**. This will install and Tacotron 2 and its dependencies to your Colab session.

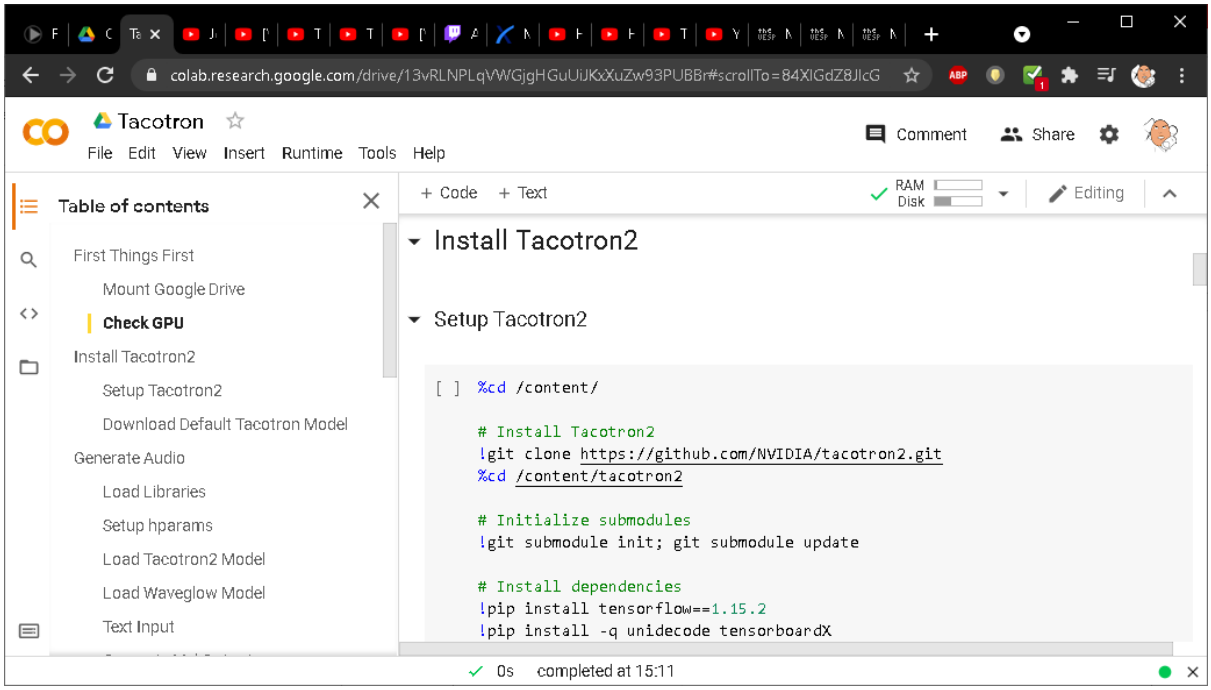


Figure 1383 - Installing Tacotron 2.

Now scroll down to **Generate Audio** and run the two code cells under **Load Libraries**.

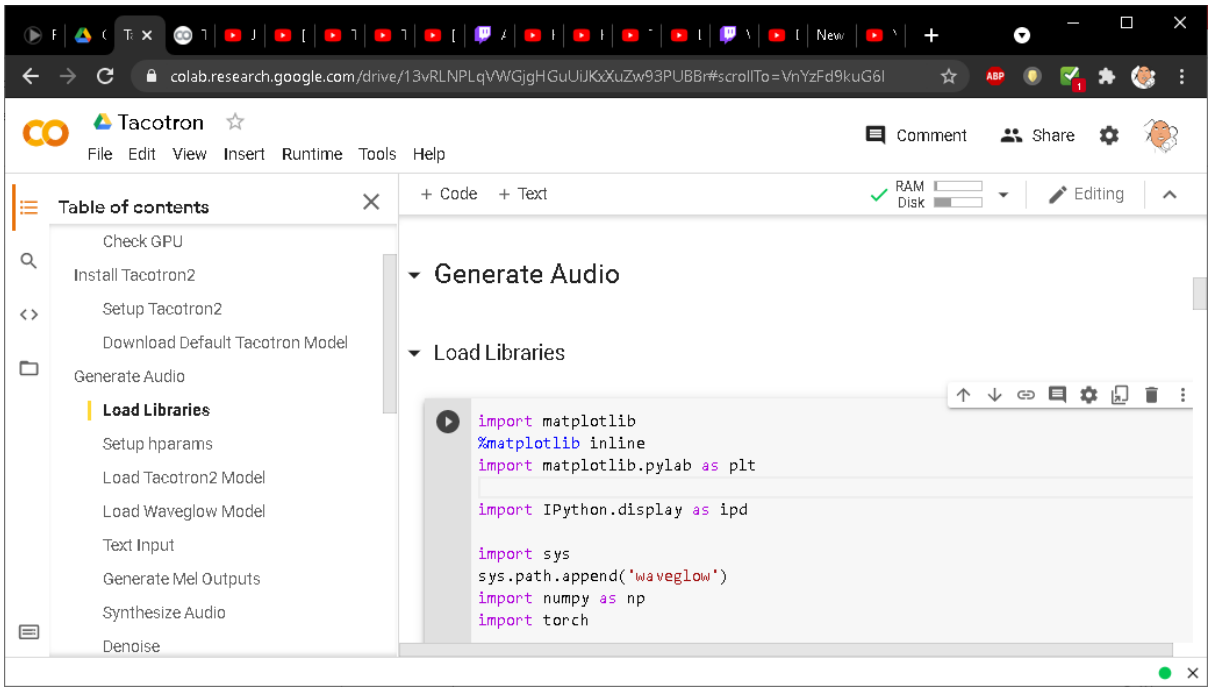


Figure 1384 - Load libraries.

Run the code cell under **Setup hparams**.

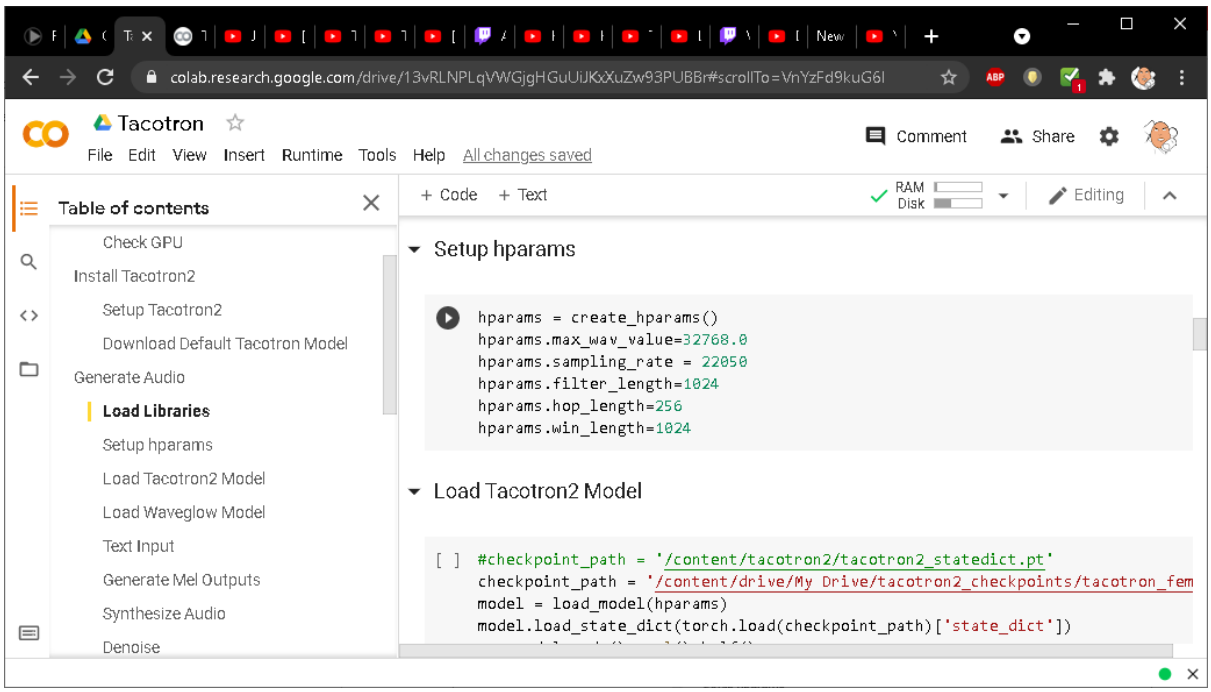


Figure 1385 - Setup hparams.

Under **Load Tacotron2 Model**, set 'checkpoint_path' to point to your Tacotron model before running the code cell. For my femalenord example, that's going to be '/content/drive/My Drive/tacotron2_checkpoints/tacotron_femalenord'.

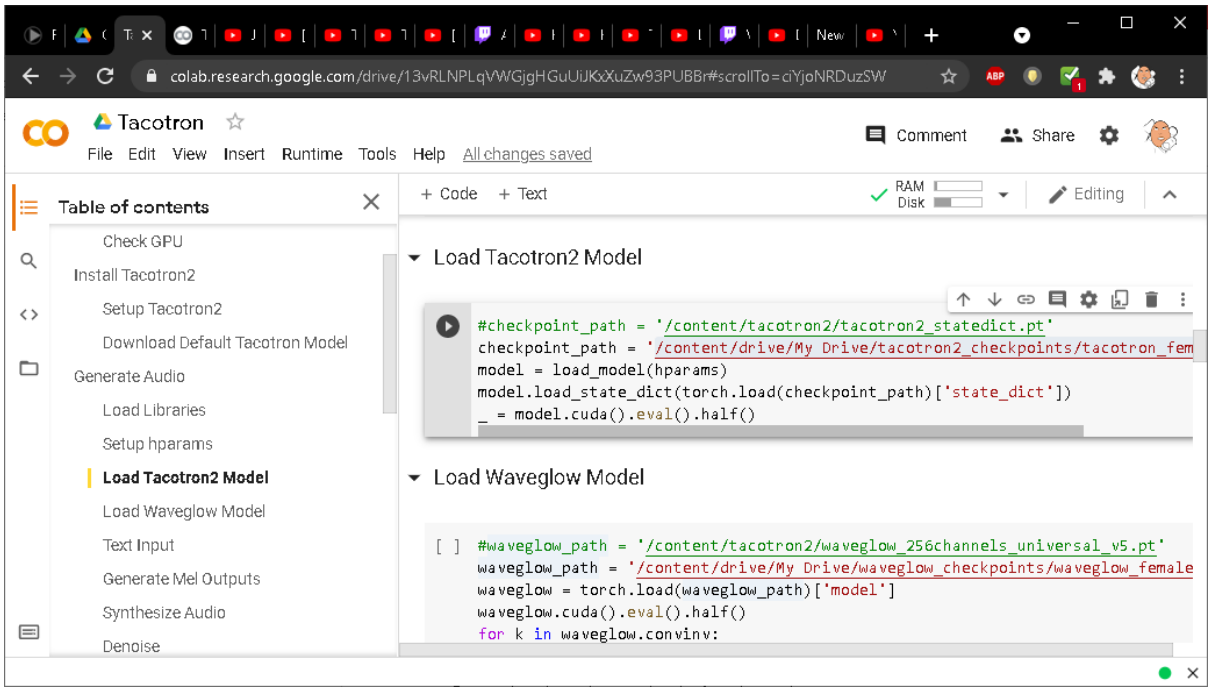


Figure 1386 - Load Tacotron2 Model.

Now under **Load WaveGlow Model**, set 'waveglow_path' to point to your WaveGlow model before running the code cell. For my femalenord example, that's going to be '/content/drive/My Drive/waveglow_checkpoints/waveglow_femalenord'.

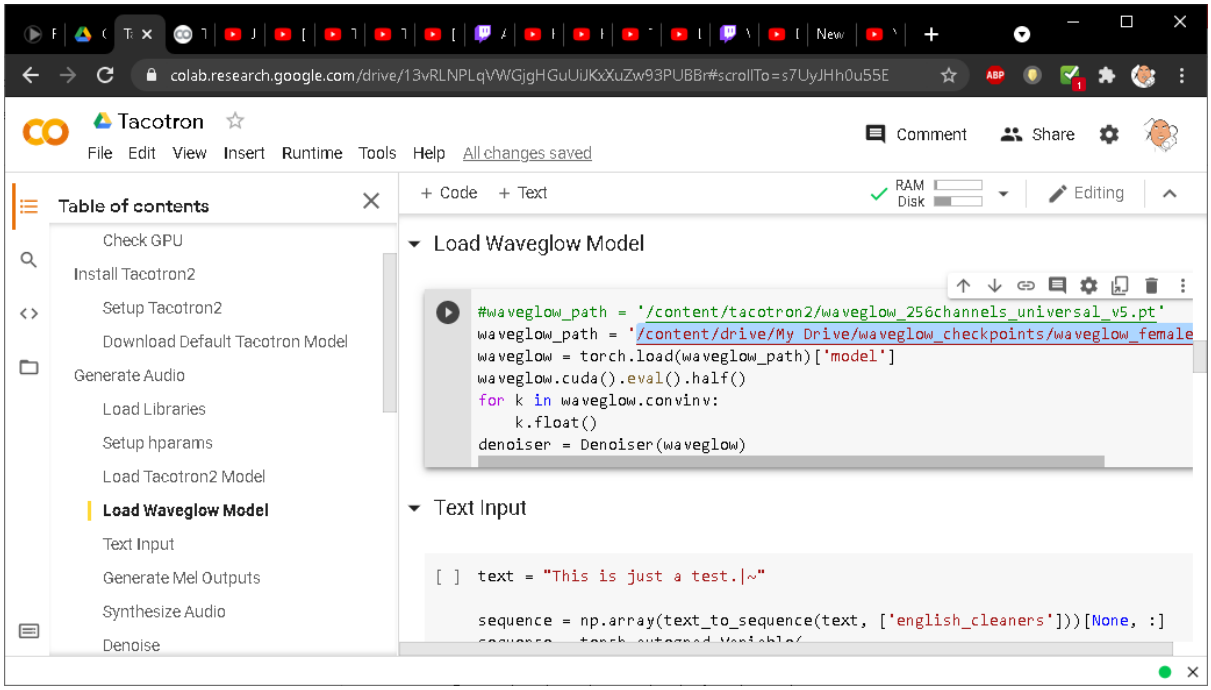


Figure 1387 - Load WaveGlow Model.

Under **Text Input** you specify the line of dialogue you want to generate then click run cell. Try to keep this down to one or two sentences at most. You can always edit the dialogue together in an audio editing program like Audacity later on.

I marked the end of the line with '|~'. Usually this isn't necessary but it helps tell Tacotron to stop once it reaches that character, just in case your model tends to produce gibberish at the end of dialogue.

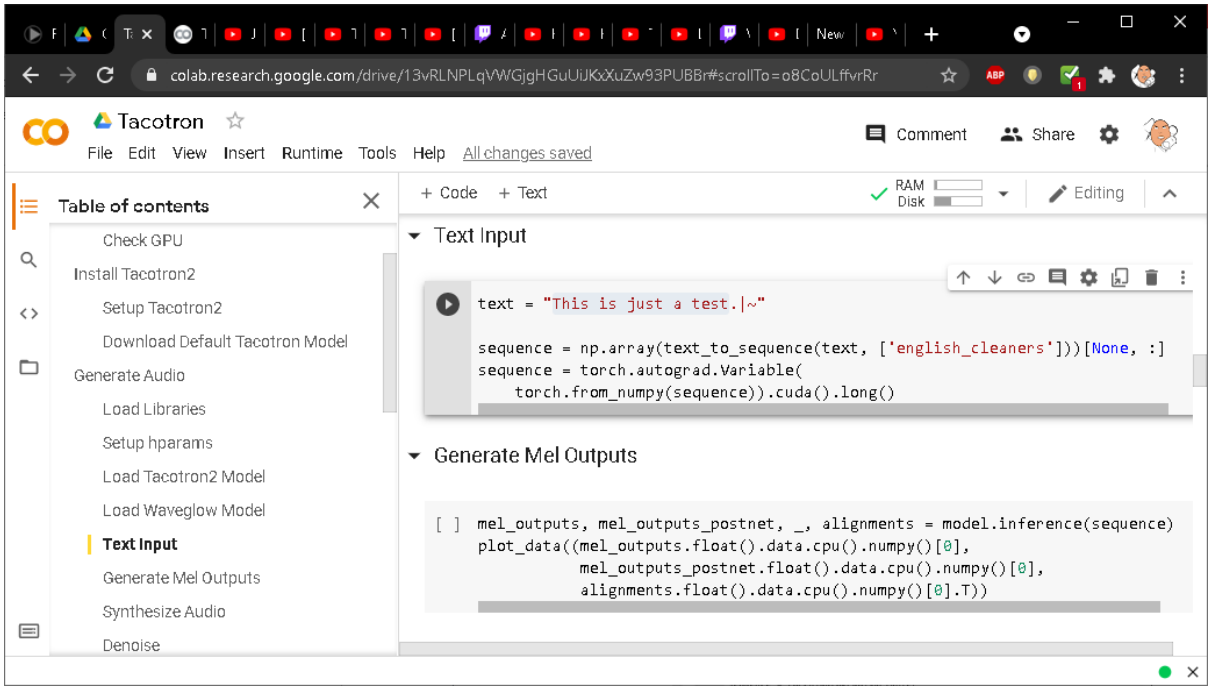


Figure 1388 - Text Input.

Run the code cell under **Generate Mel Outputs**.

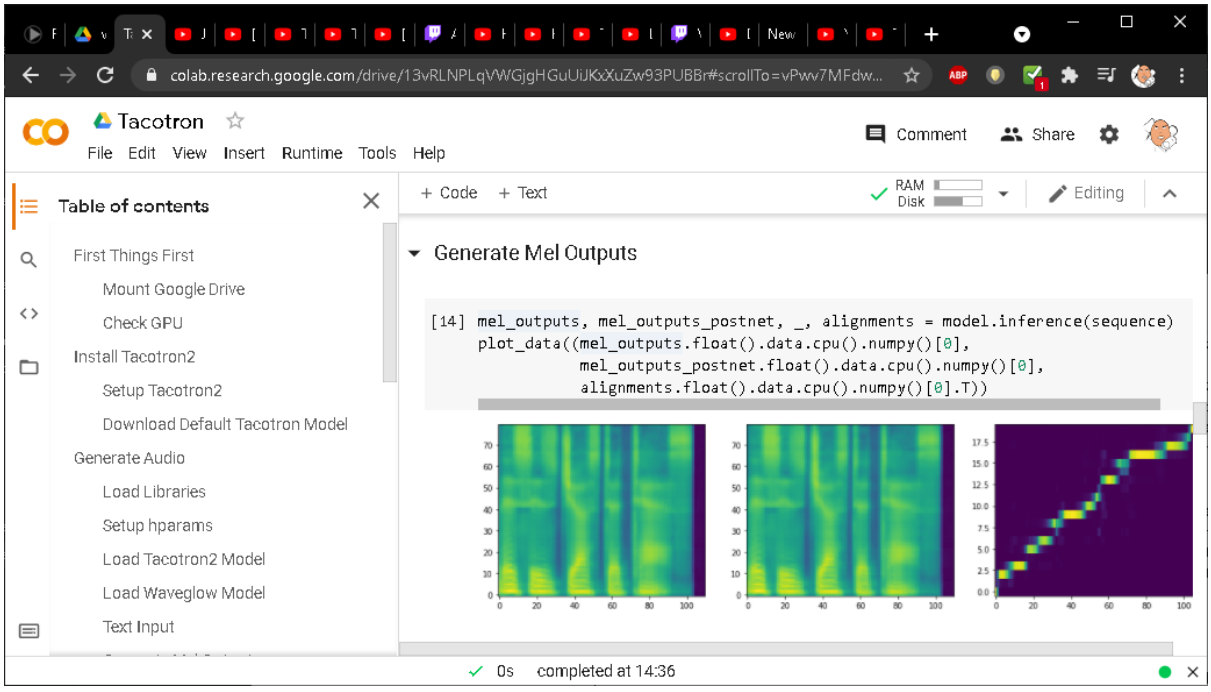


Figure 1389 - Generate Mel Outputs.

Running the code cell under **Synthesize Audio** will generate raw unprocessed output. Press the play button to listen to the audio. You'll most likely notice some high pitched humming. The default sigma value is '1'. Lowering this value can help remove some robotic-like sounds but note that the further you lower it the more muffled the voice will be. I usually keep this set to '0.85', but you might need to lower it to '0.83' depending on how many distortions you hear in the voice.

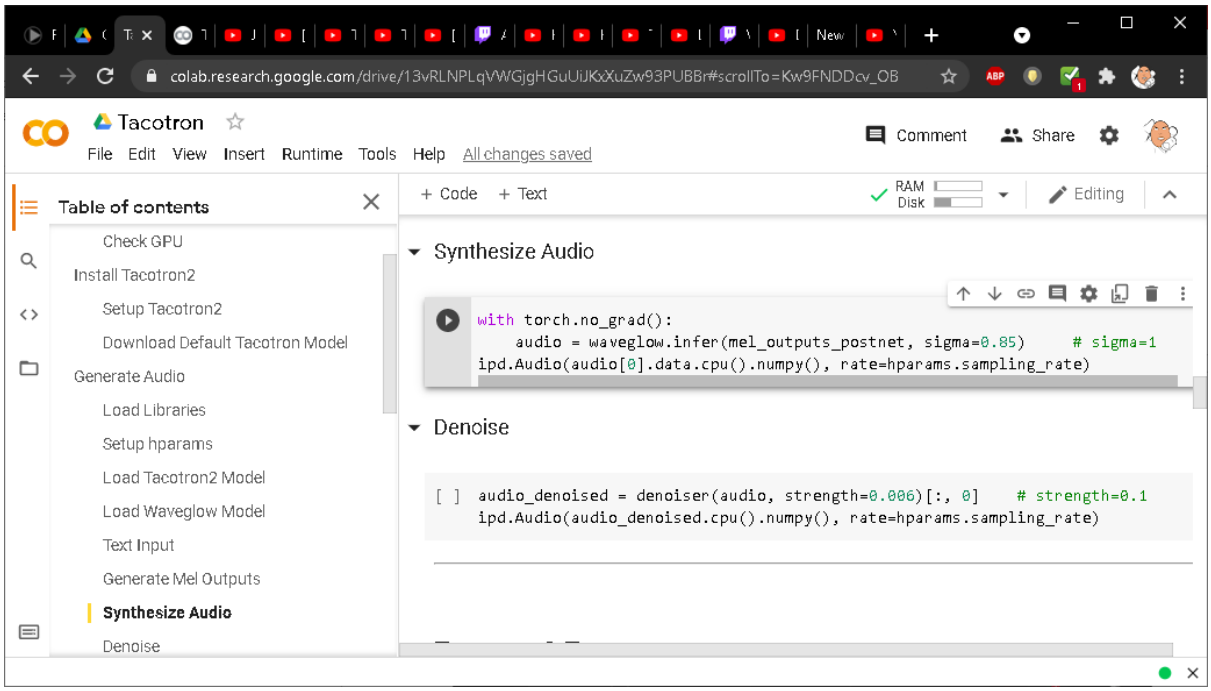


Figure 1390 - Synthesize Audio.

We'll need to run the **Denoise** code cell to clean things up.

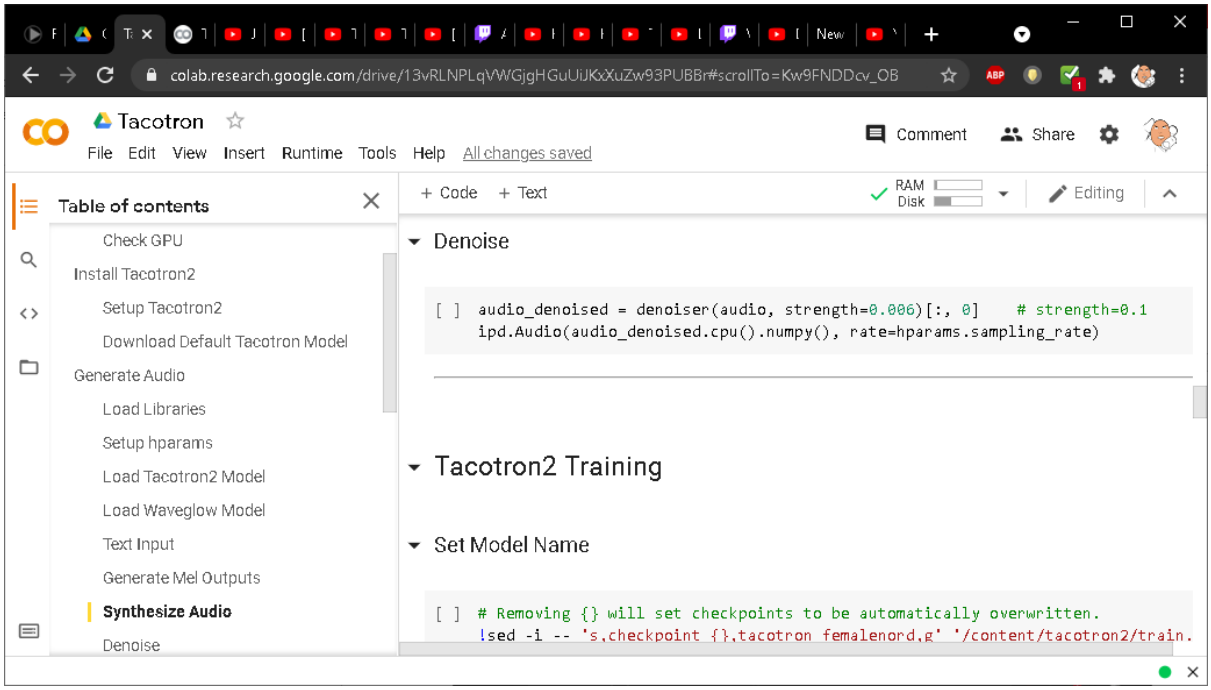


Figure 1391 - Denoise.

The higher the strength of the denoiser the more muffled your audio will sound so try and keep this value just low enough to remove the high pitched buzzing. I usually keep the strength set to '0.006'. The default is '0.1' but I think this is a bit excessive.

You can download the audio as a .wav file by clicking on the three dots and selecting **Download**.

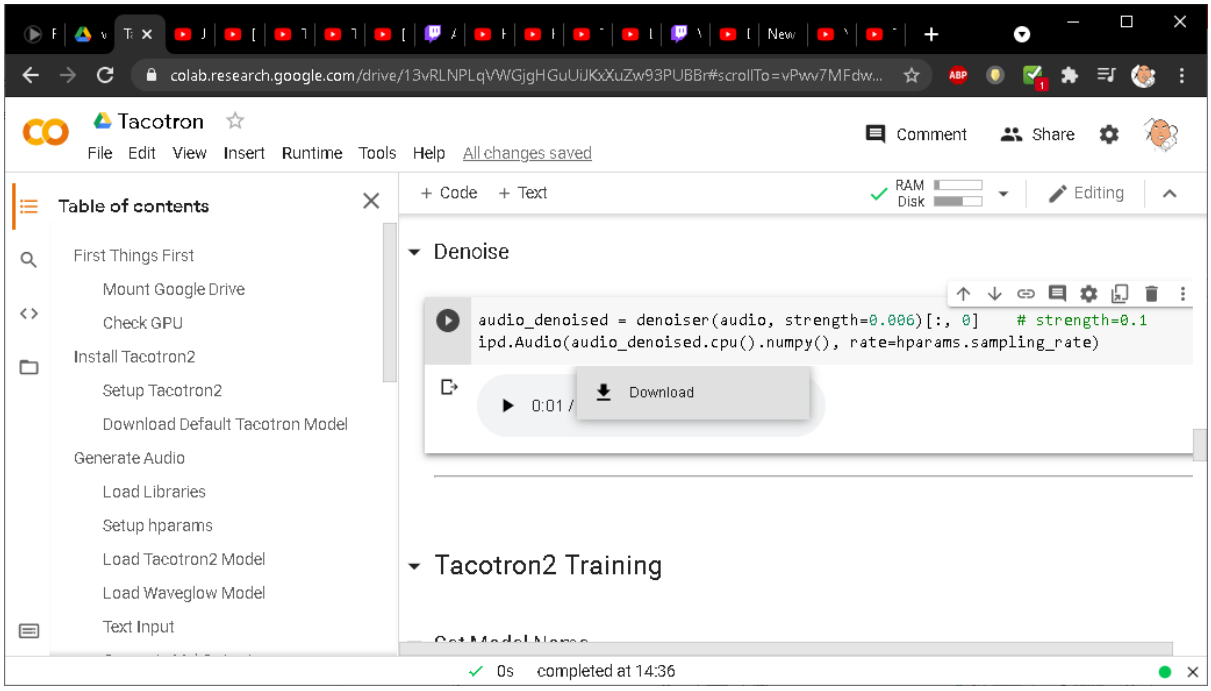


Figure 1392 - Downloading our audio.

To generate a new line of dialogue, just run the cells from **Text Input** onwards.

Now that we've generated some audio samples, it's time to try and improve audio quality in Audacity.

IMPROVING AUDIO QUALITY IN AUDACITY

In this section I'll be using Audacity to try and upsample the audio clips generated by Tacotron. You can download Audacity for free from its website.

Audio output from Tacotron will be at 22050Hz by default. While we can output audio at a higher sample rate, it likely won't sound much better.

Upsampling alone won't do much to improve the quality of our audio. We need to try and boost some of the higher frequencies to add a bit more depth into the voice.

Important: There's no *good* way to do this and there are probably better methods than what I'm about to show you. But this is just the method that I use to upsample vocal recordings.

There are AI-based audio upsampling programs out there, but they tend to introduce more distortions so I'm not going to cover that in this tutorial.

Open the .wav file in Audacity.

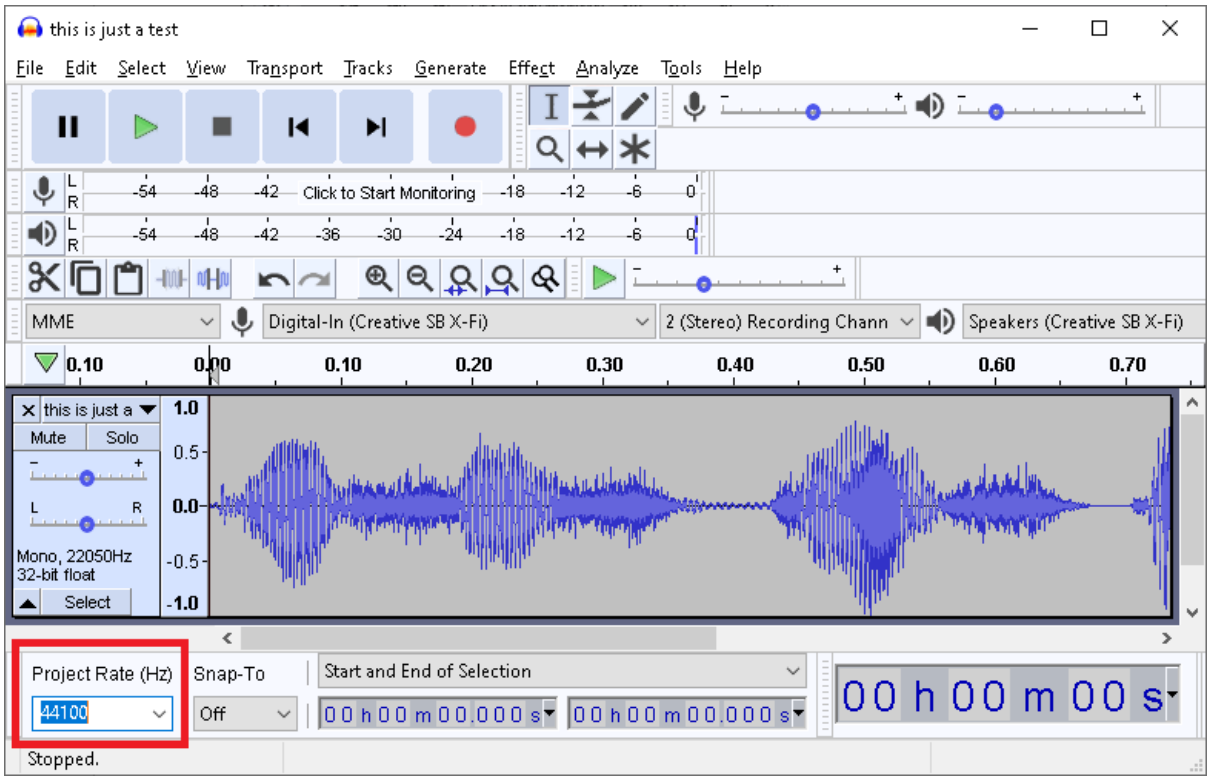


Figure 1393 - Changing the Project Rate.

Set the Project Rate (Hz) in the bottom-left corner to 44100.

Press CTRL+A to select the entire audio track.

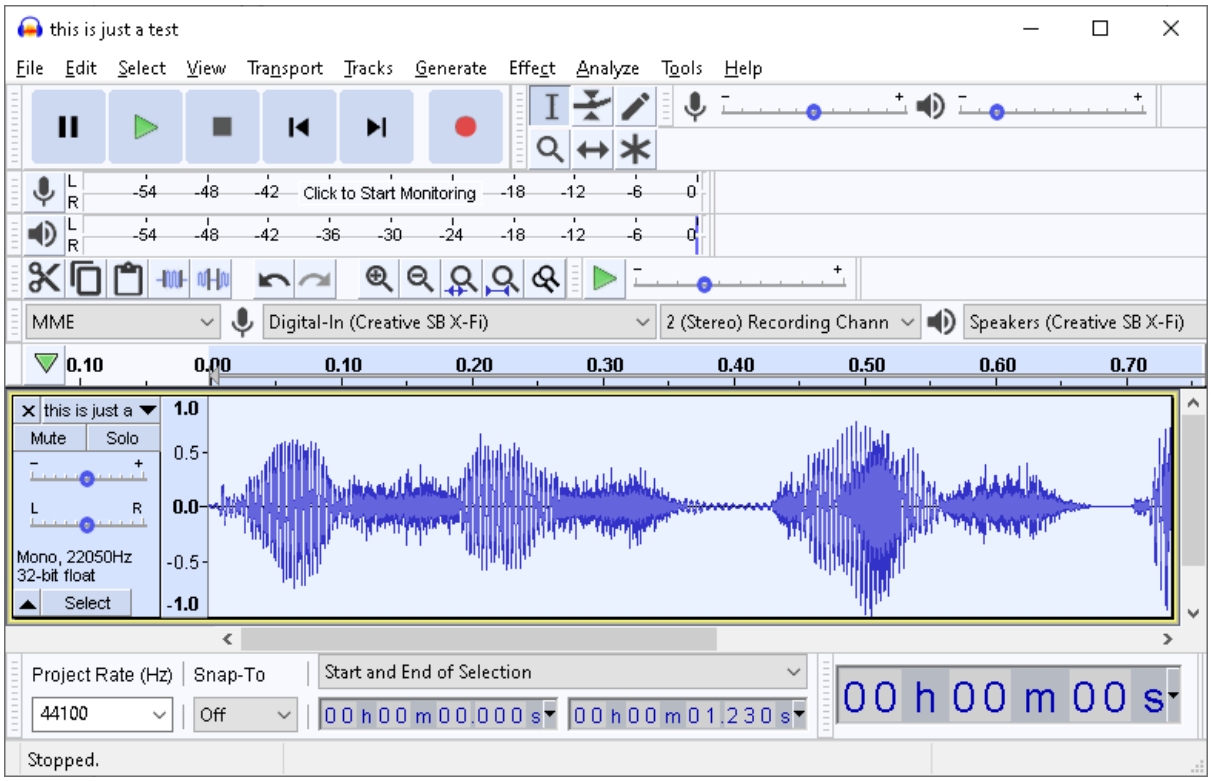


Figure 1394 - Entire audio track selected.

Now go to Tracks > Resample…

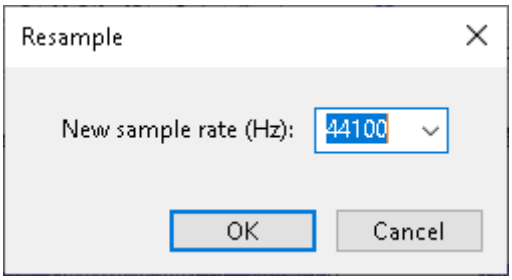


Figure 1395 - Resampling to 44100 Hz.

Make sure the 'New sample rate (Hz)' is set to 44100 then click OK.

Go to Edit > Duplicate to copy the selected audio track.

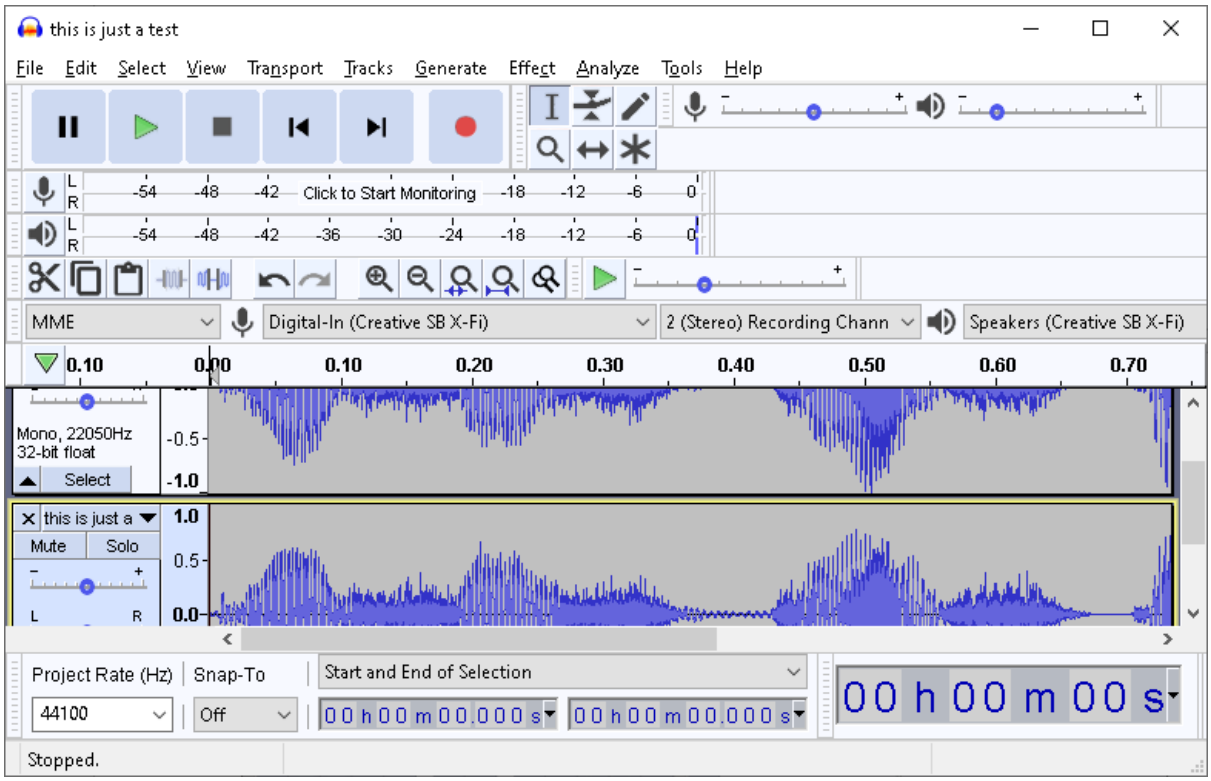


Figure 1396 - Audio track duplicated.

Select the entirety of the duplicate track by double-clicking anywhere within it.

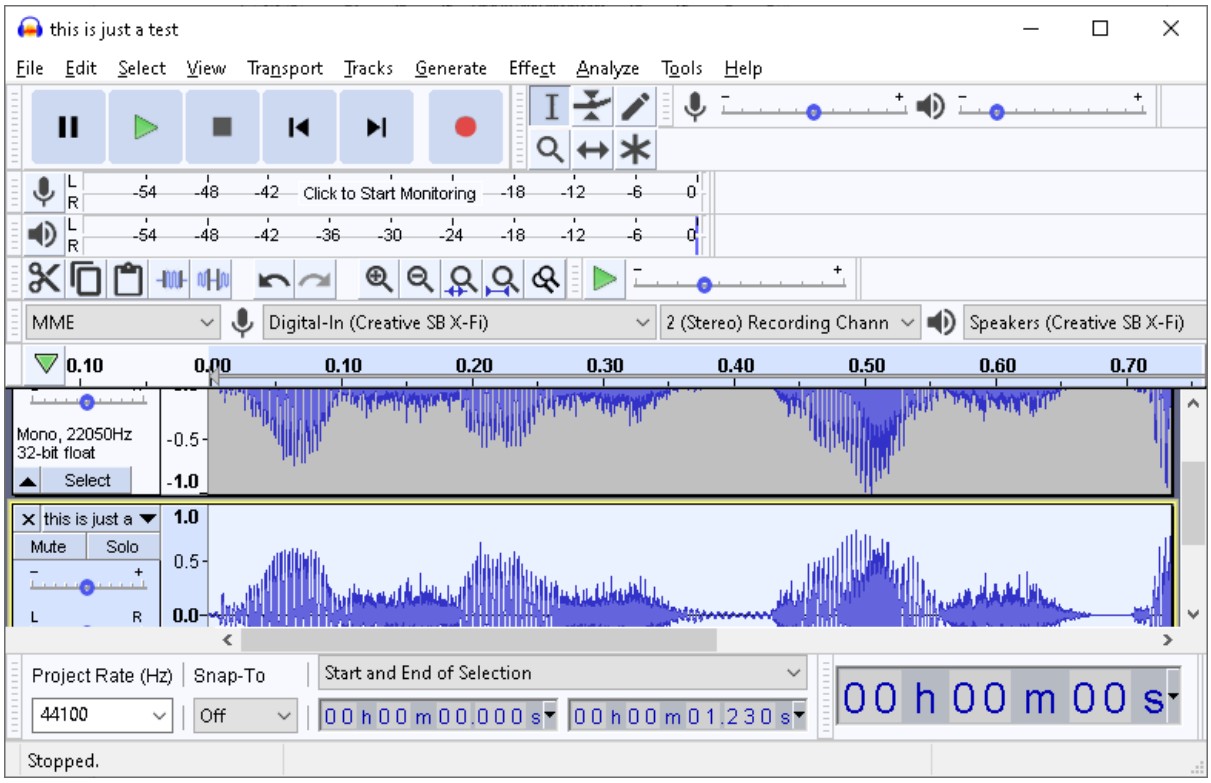


Figure 1397 - Duplicate track selected.

Go to Effect > High Pass Filter…

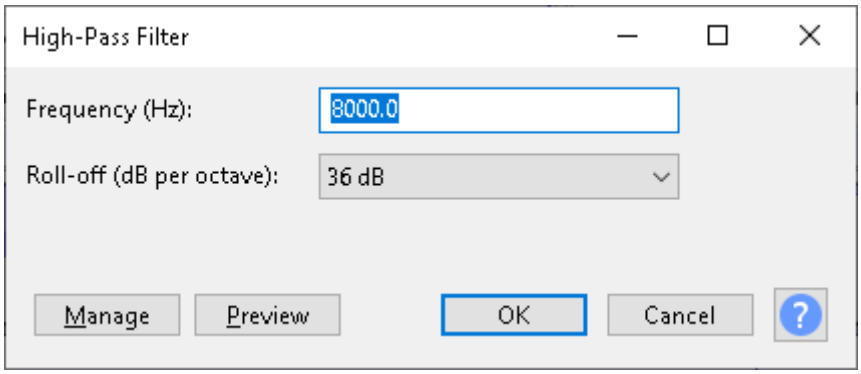


Figure 1398 - High Pass Filter.

Set 'Frequency (Hz)' to 8000.0 and 'Roll-off (dB per octave)' to 36 dB then click OK. This will isolate just the higher frequencies.

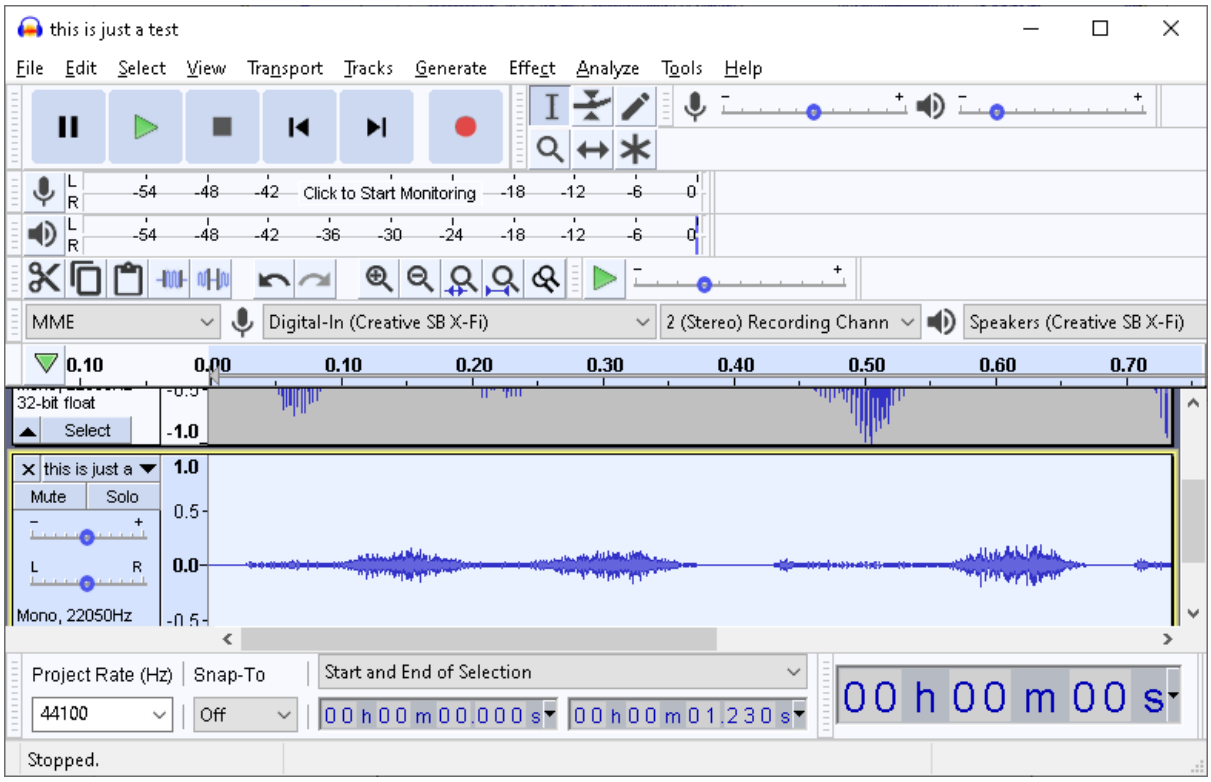


Figure 1399 - High Pass Filter applied to duplicate track.

We need to raise the pitch of these higher frequencies a bit. With the duplicate track still selected, go to Effect > Change Pitch…

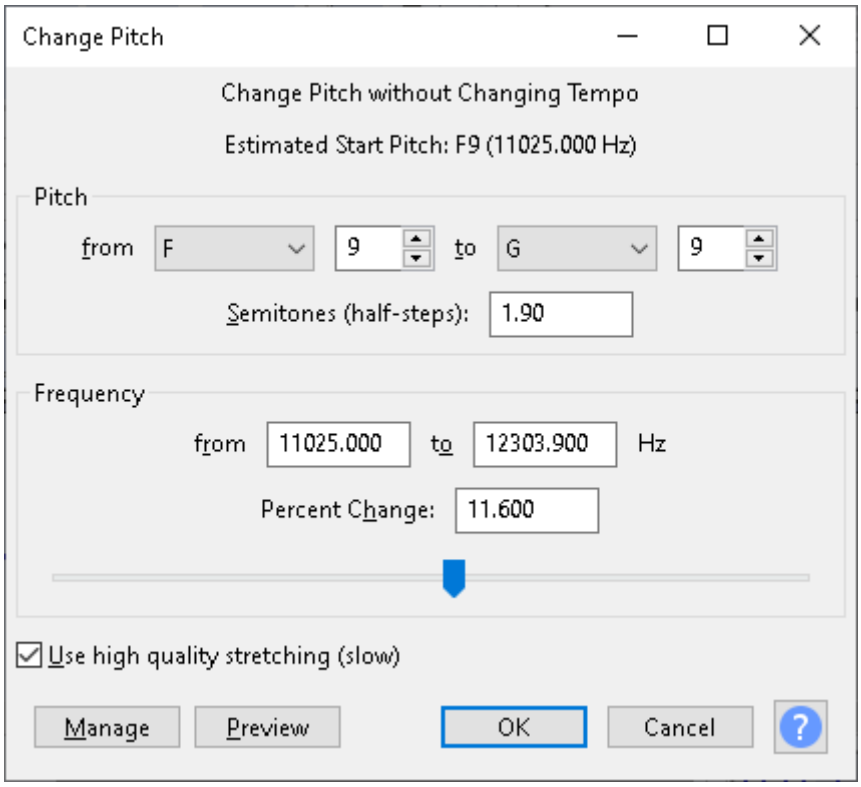


Figure 1400 - Change Pitch.

Set 'Semitones (half-steps)' to 1.90. Ensure 'Use high quality stretching (slow)' is ticked then click OK.

The last thing we need to do is play around with the gain of the duplicate track. For male voices I usually set this to around +3 dB depending on the voice, just to add a little bit more depth. For female voices I set this to -3 dB to lessen high pitched hissing.

You'll need to play this by ear and find out what sounds best yourself.

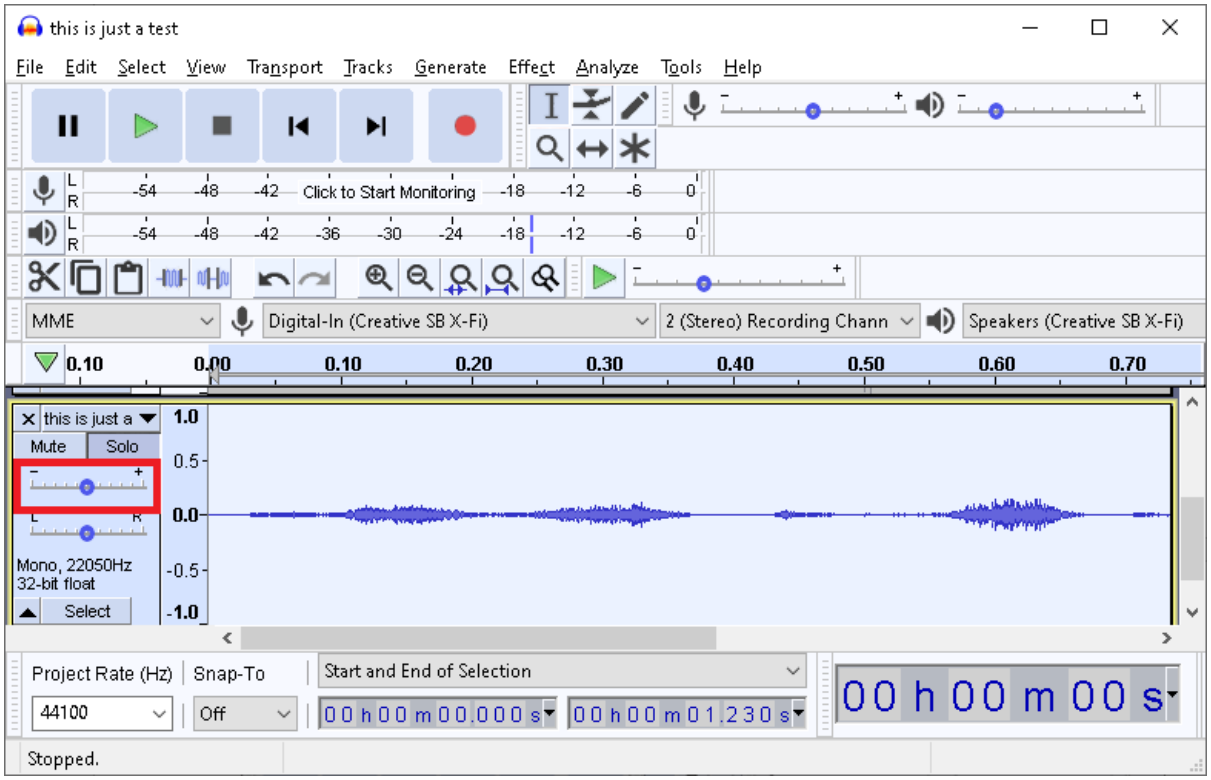


Figure 1401 - Gain controls.

To save a new .wav file, go to File > Export > Export as WAV.

Select the folder you want to save the .wav file to then click Save.

A COUPLE TIPS AND TRICKS

TIP 1: Sometimes part of the last word will get cut off, so what I like to do is add another word after it, like 'end', to help mitigate that.

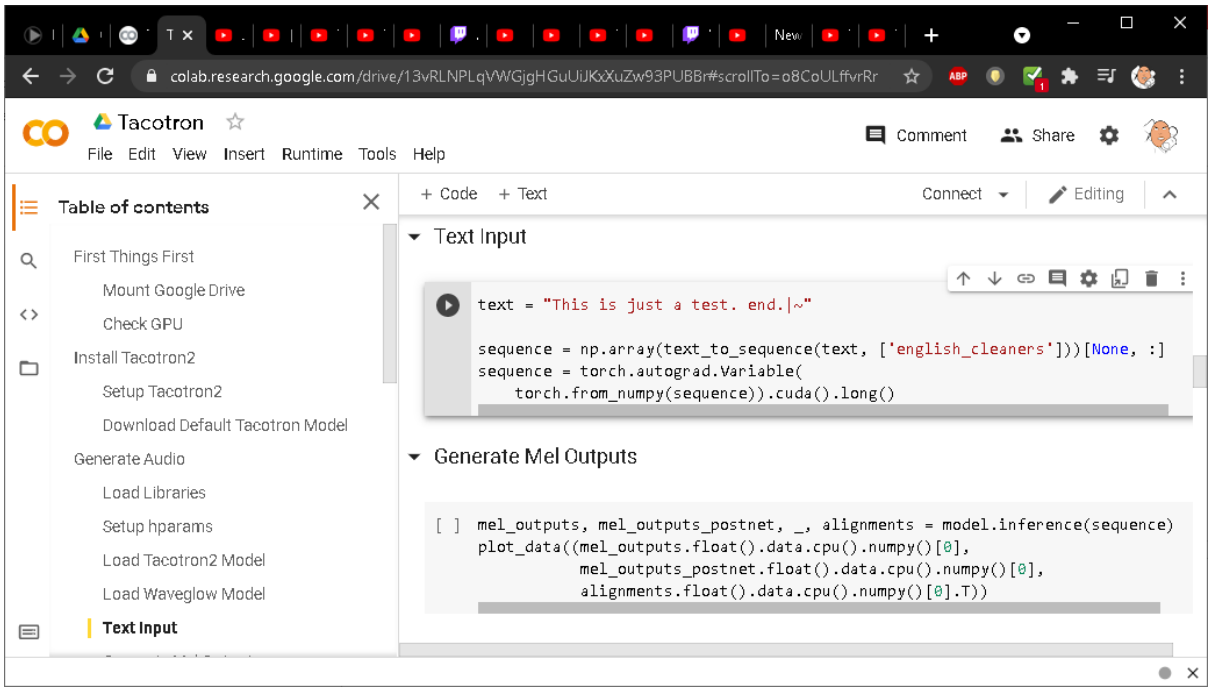


Figure 1402 - Ending word.

TIP 2: The first few words seem to have a large impact on the delivery of the rest of the dialogue.

'this is just a test' may sound a lot different to 'elf scum! this is just a test'. If you copy part of an existing line of dialogue from your dataset, like 'never should've come here!' and prefix it before the sentence you actually want to generate, some of the emotional intensity from the original line of dialogue will transfer across to the new line of dialogue.

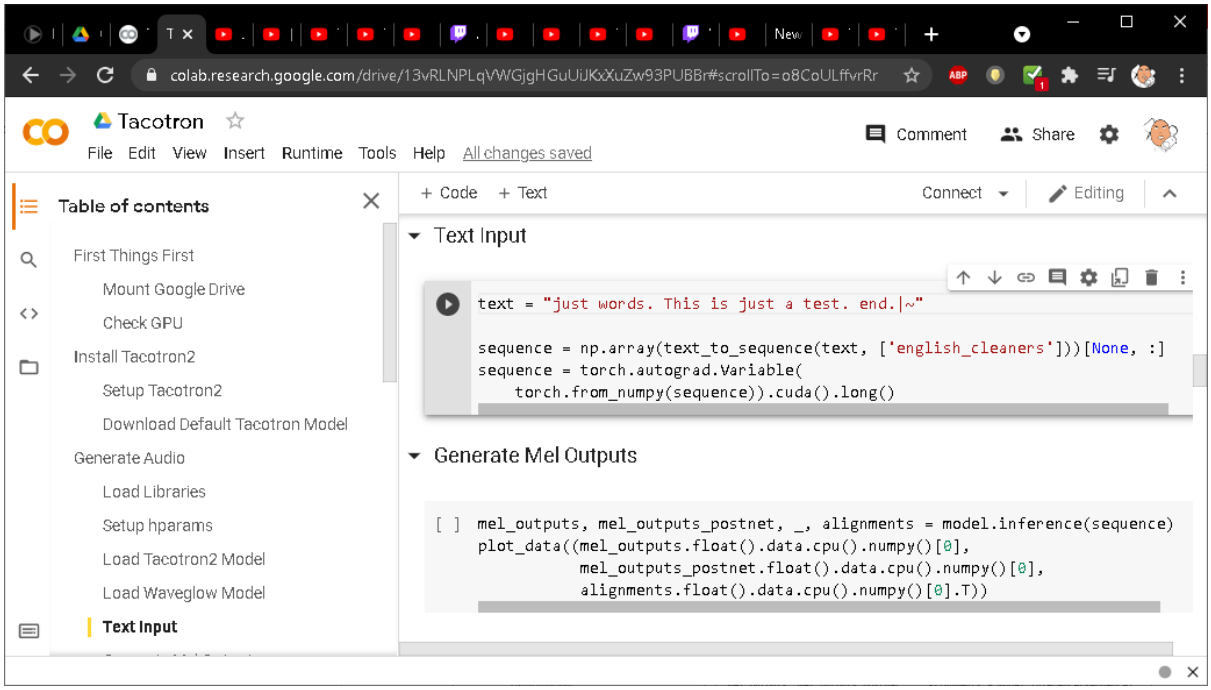


Figure 1403 - Starting words.

If your dialogue is being screamed or shouted and you don't want it to be, try prefixing something benign like 'hello there' and that should lower the emotional intensity of the rest of the dialogue.

Capitalization doesn't really make a difference to the output.

TIP 3: Every time you generate dialogue it'll sound different. If you don't like the way a specific sentence was conveyed, just run the code cells from **Generate Mel Outputs** onwards and check again.

If I hear some robotic-like noises or distortions, I just generate the dialogue over and over until I get a clean read.

TIP 4: Not all words will be pronounced properly. The malenordcommander model I trained has issues with words like 'dialogue'. In those cases I'd have to type in 'die a log' instead. For making malenordcommander pronounce 'python' properly (and not as 'pee-thon') I'd have to type it in as 'piiethon'.

It'll require a bit of trial and error to figure out what works in those cases.

CHAPTER 11: OTHER TASKS

UNPACKING THE BSA ARCHIVES

This section will cover unpacking the .bsa archives.

For this we'll need a copy of BAE, the Bethesda Archive Extractor, which you can download from Nexusmods.

Launch bae.exe.



Figure 1404 - BAE.

To extract a .bsa archive, drag and drop it into the utility.

All items in the archive should be selected by default. If they aren't, press Select All or manually choose the specific items you want to extract.

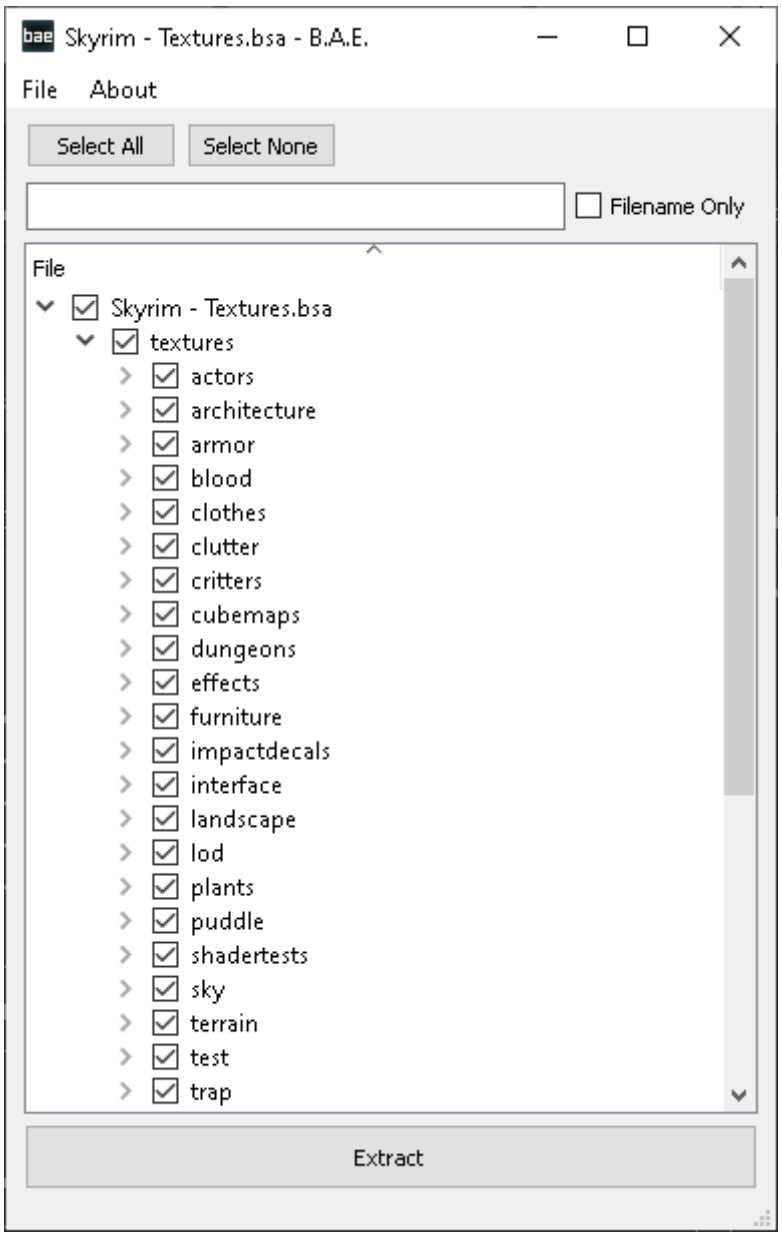


Figure 1405 - The Skyrim - Textures.bsa in BAE.

Click Extract.

Select the folder to extract the assets to and click the Select Folder button.

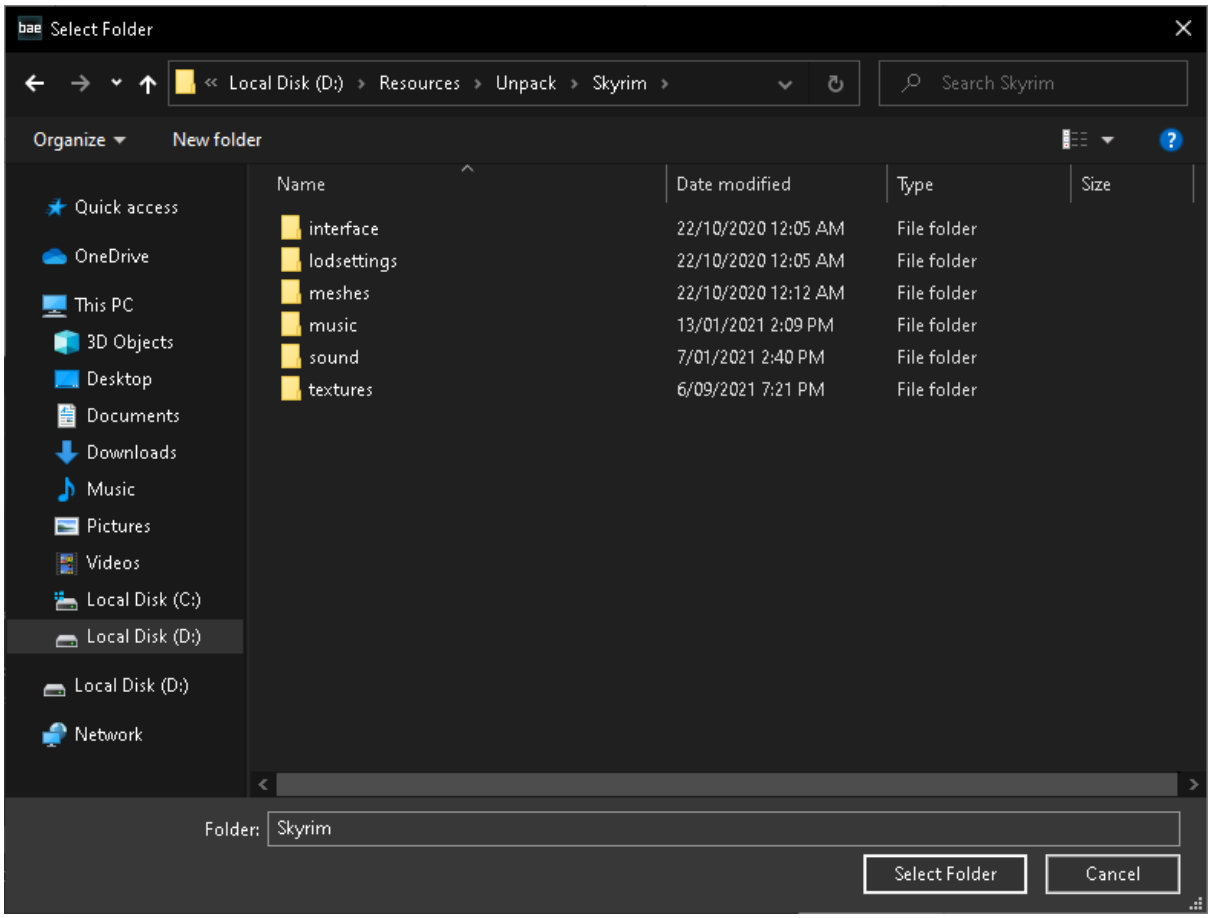


Figure 1406 - Choosing the folder you want to extract to.

Wait until the extraction process is complete.

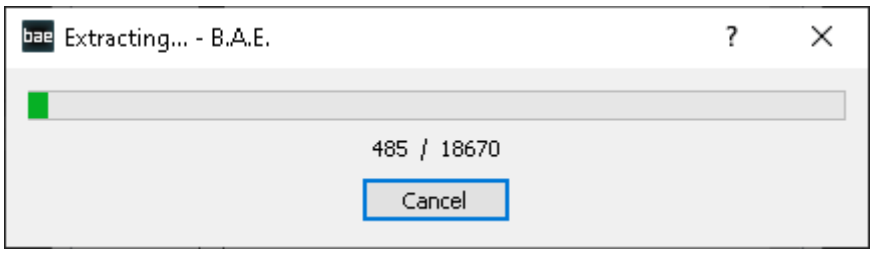


Figure 1407 - Extraction progress.

Once the process is complete, close BAE.

ADDING ADDITIONAL MASTER FILES

If you want to include assets from the DLC or from other master files, you can add additional dependencies to your mod.

These steps assume you haven't used xEdit to set the ESM flag in your mod yet.

Firstly, open your SkyrimEditor.ini file. Go to [General] and add the following:

bAllowMultipleMasterFiles=1
bAllowMultipleMasterLoads=1

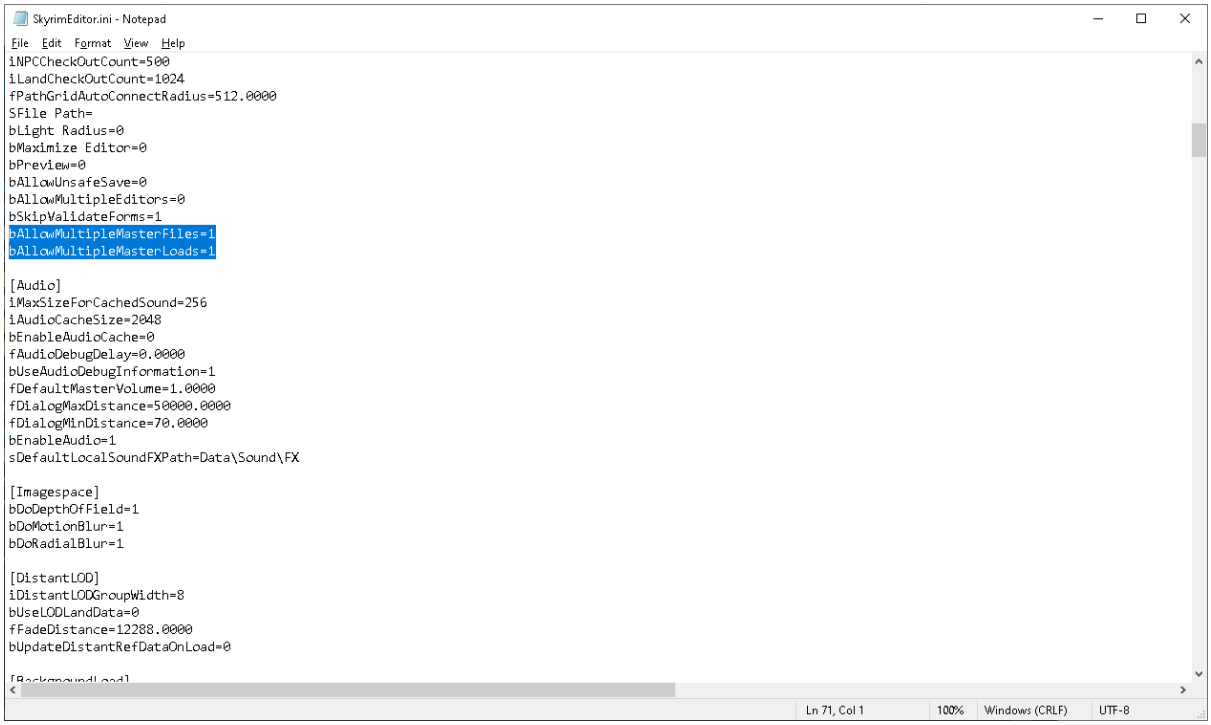


Figure 1408 - SkyrimEditor.ini load multiple master files.

Next, scroll down to [Archive] and add the .bsa file associated with the master to SArchiveList and SResourceArchiveList.

For example, if you want to make your mod dependent on Dawnguard, make sure Dawnguard.bsa is added to these two entries.

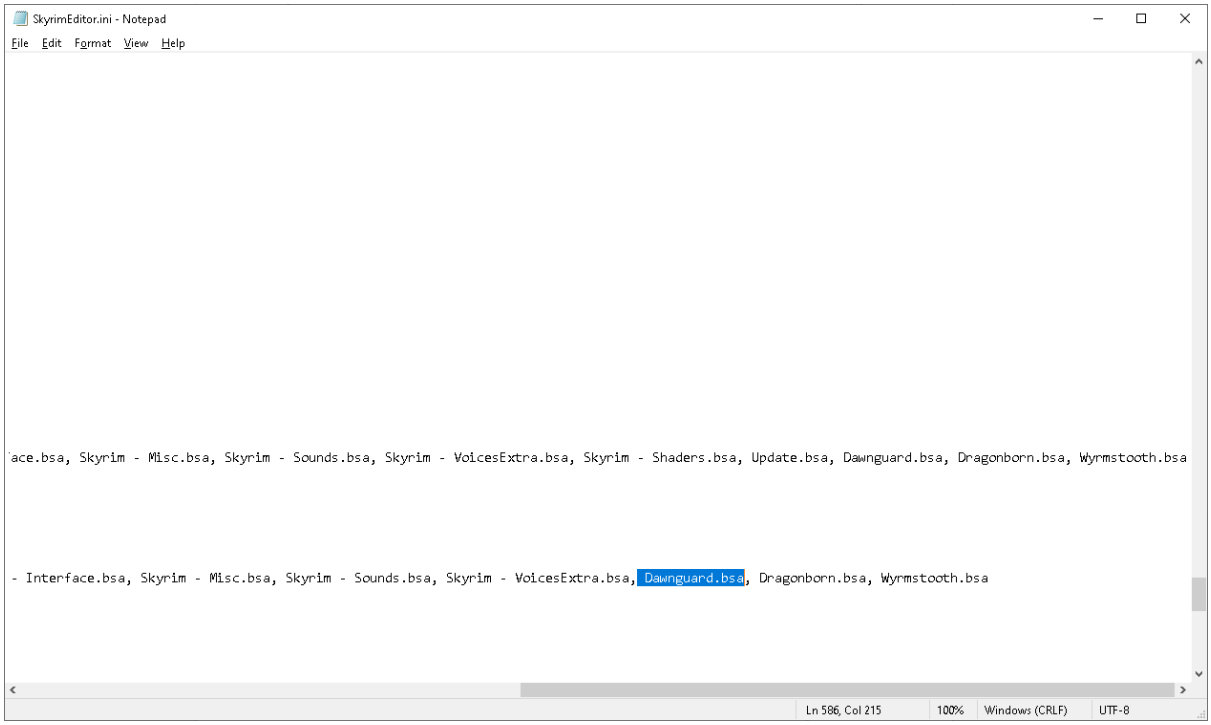


Figure 1409 - List of BSA archives loaded by the Creation Kit.

Launch the Creation Kit. Go to File > Data. Tick your mod and click 'Set as Active File'.

Now tick the master file you want to make your mod dependent on, and click OK to load.

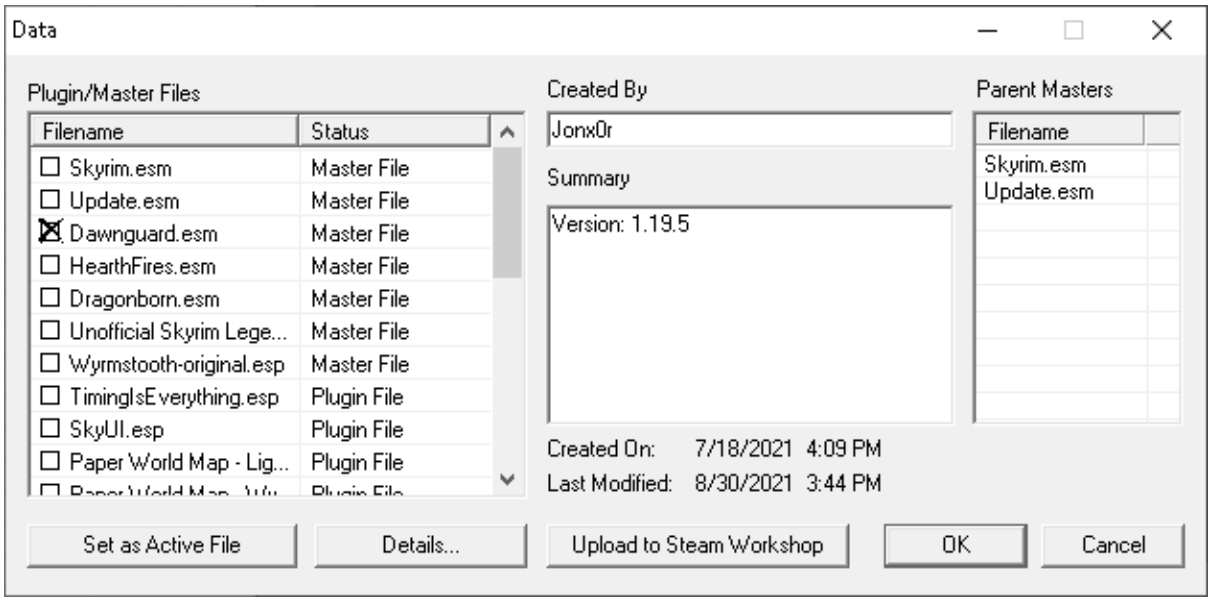


Figure 1410 - Adding Dawnguard.esm as a dependency.

Click Save.

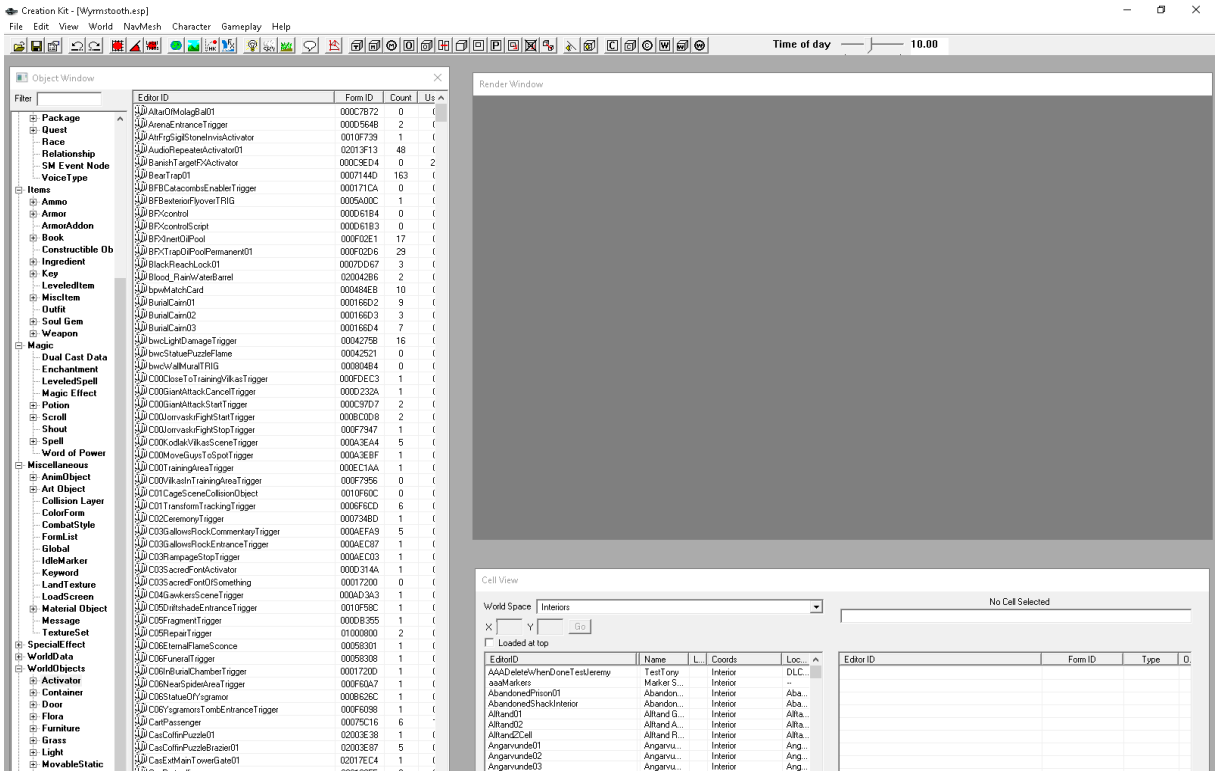


Figure 1411 - Saving the mod in the Creation Kit.

Once saving has completed, go back to File > Data. The master file you added should now appear as a dependency when you highlight your mod.

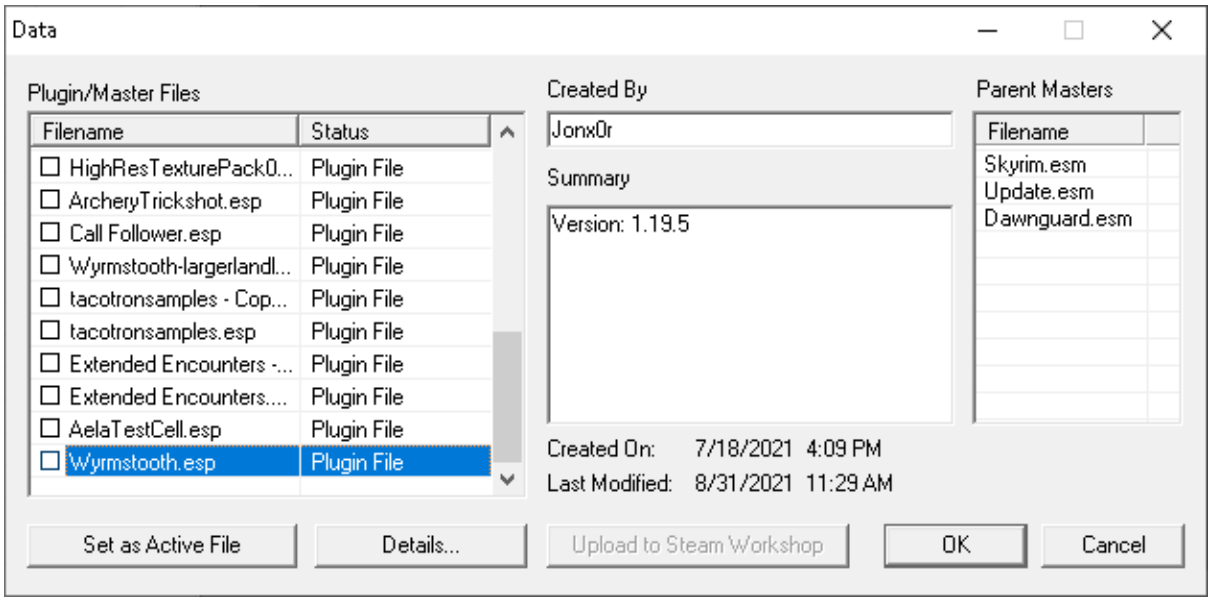


Figure 1412 - New dependency is listed.

REMOVING ACCIDENTAL EDITS

Say, for example, you accidentally moved a barrel in a base-game cell and already saved your mod, but you'd like to remove that change for compatibility purposes.

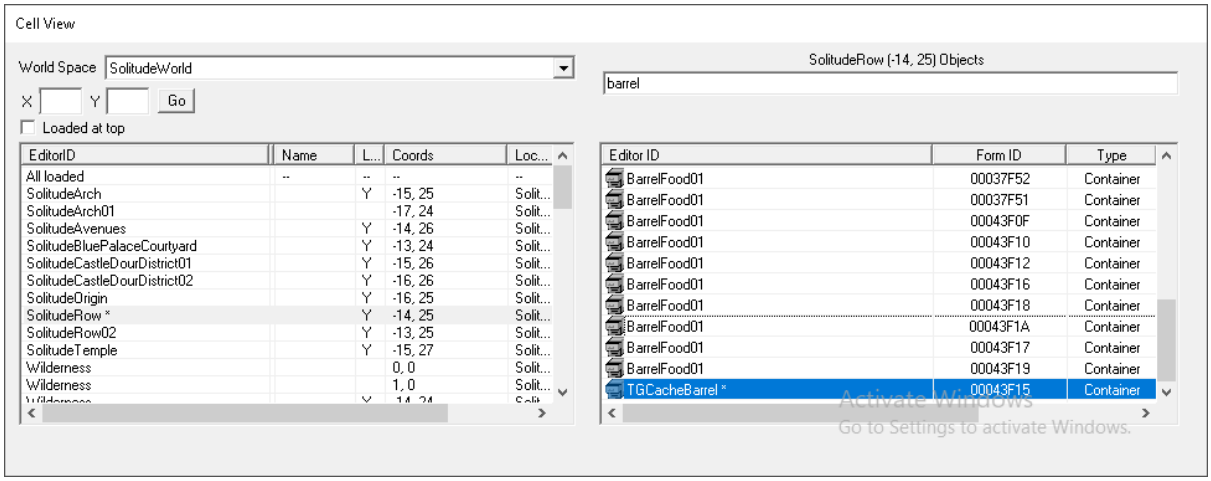


Figure 1413 - An example of an unwanted edit.

Go to File > Data.

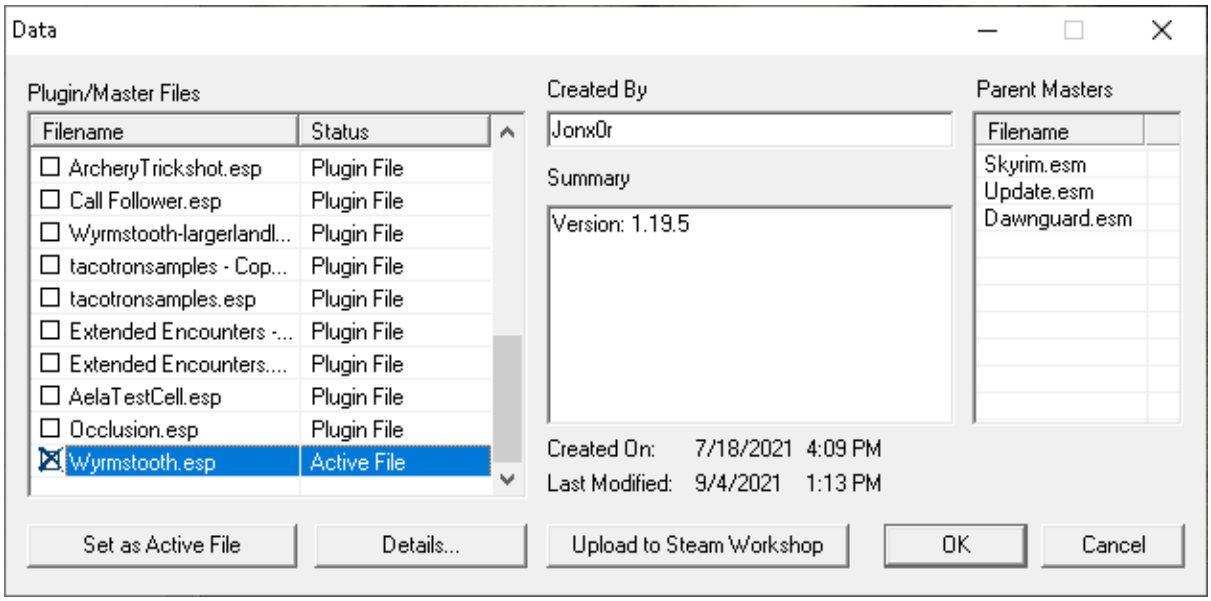


Figure 1414 - List of mods.

Highlight your mod in the Plugin/Master Files list and click on the Details button.

Expand the Form ID column. It'll be to the right of the Editor ID column and will be collapsed by default.

Click on the Form ID field to sort records numerically from lowest to highest.

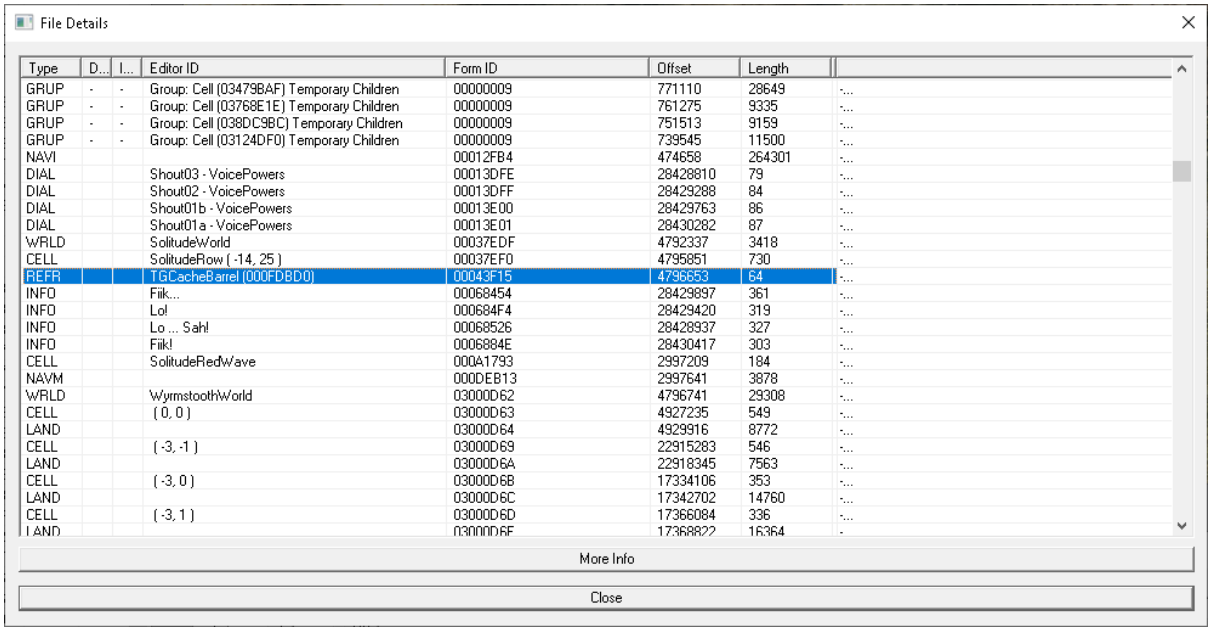


Figure 1415 - Form ID field sorted numerically.

Find the reference you want to remove from your mod. In our example that'll be a TGCacheBarrel.

Click on it to highlight it and press Delete.

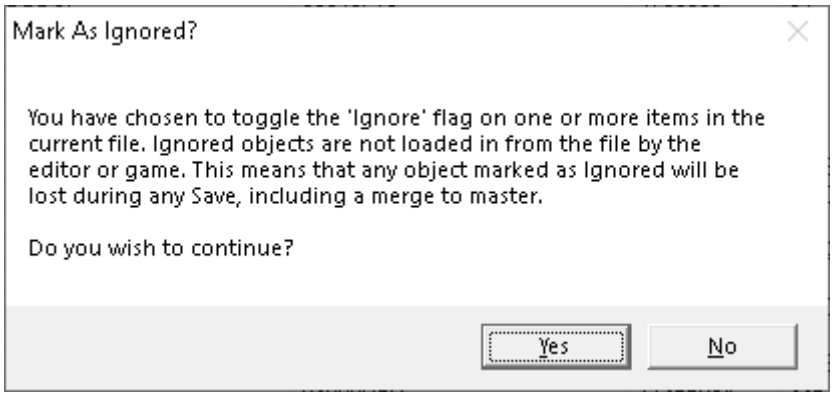


Figure 1416 - Removal confirmation.

Click Yes to the Mark As Ignored pop-up warning.

You should now see an 'I' next to the deleted reference.

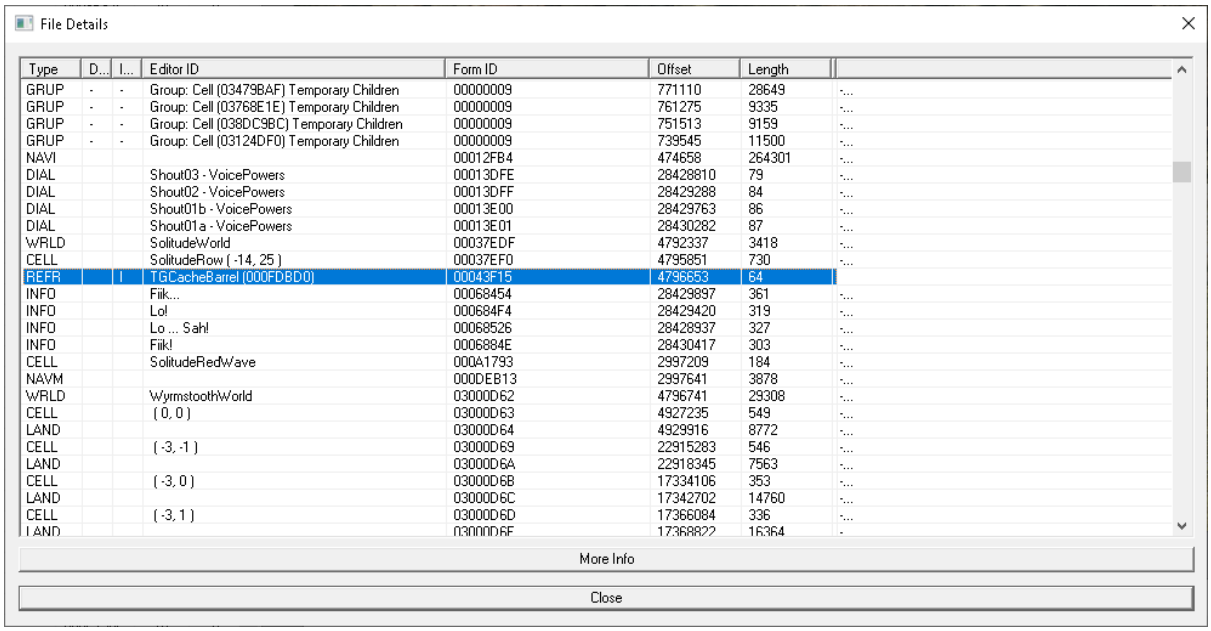


Figure 1417 - The edit to TGCacheBarrel has been flagged as Ignored.

For my example, I'll be removing the CELL and WRLD changes that were also created when the barrel was moved.

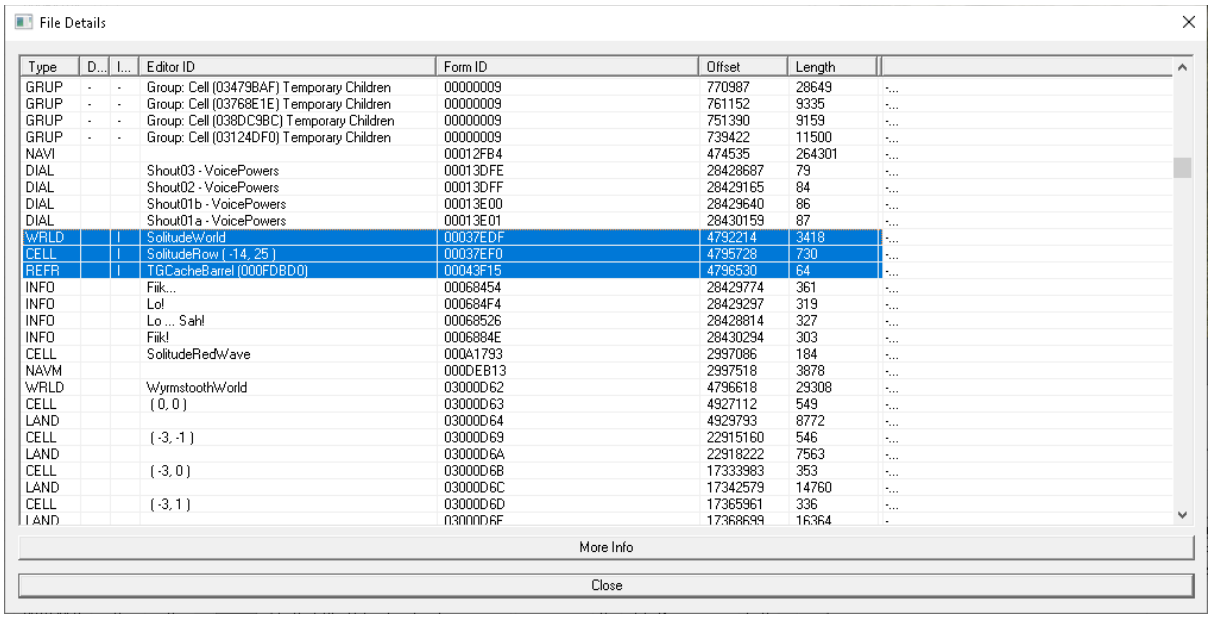


Figure 1418 - Removing the CELL and WRLD changes as well.

Click Close to close File Details.

Click OK to reload your mod.

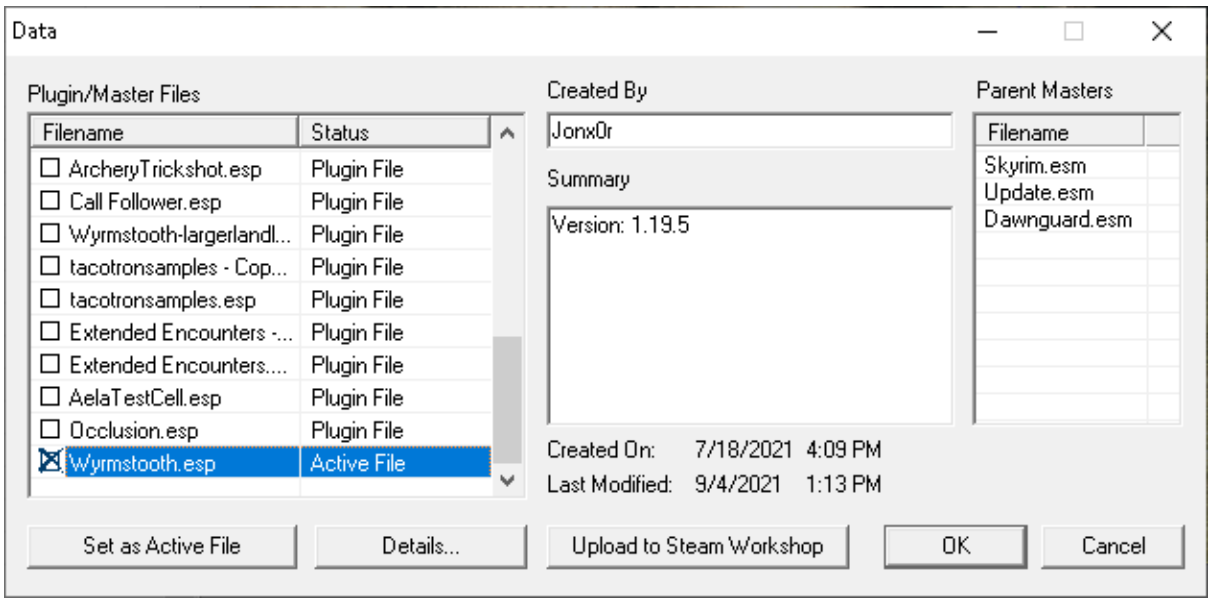


Figure 1419 - Reloading your mod.

Click on the Save button to save your mod.

Now go back to File > Data.

Again, expand the Form ID column and click on it to sort the records by Form ID.

Confirm that the deleted records are no longer listed.

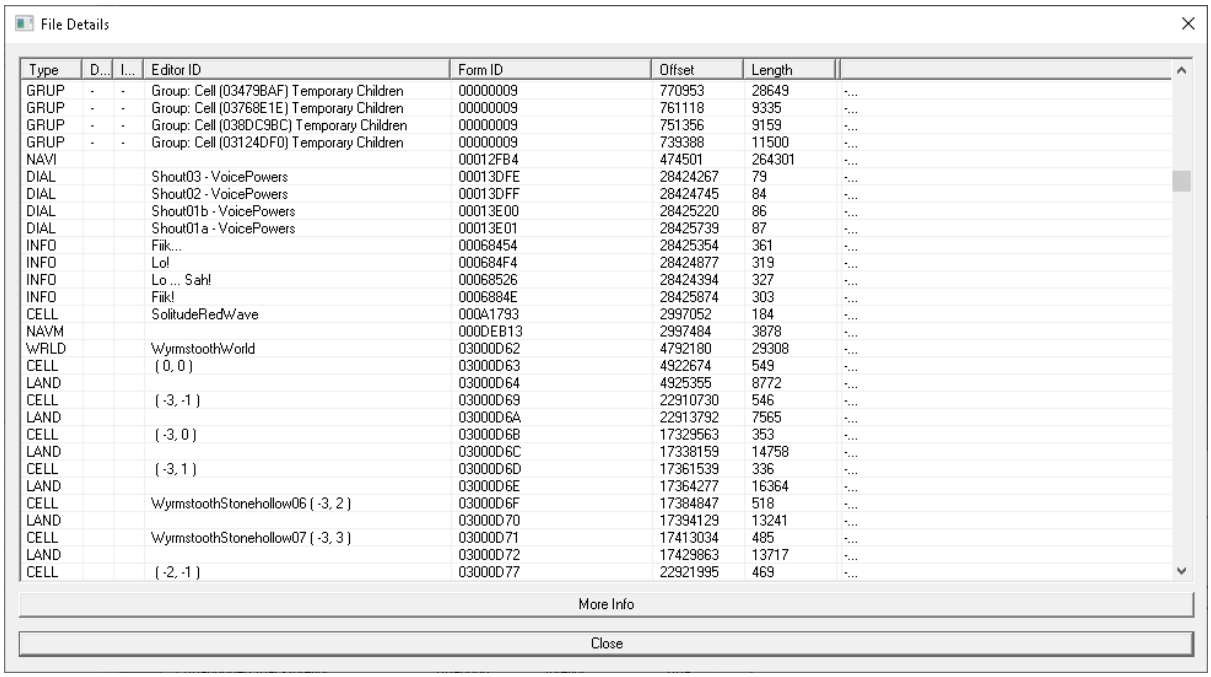


Figure 1420 - Accidental edits are gone.

ADDING TEXTURE PATHS TO 3DS MAX

In order to see textures in the 3ds Max viewports, you'll need to add external file paths pointing to the folders containing the textures extracted from the Skyrim .bsa archives.

Go to Customize > Configure User Paths.

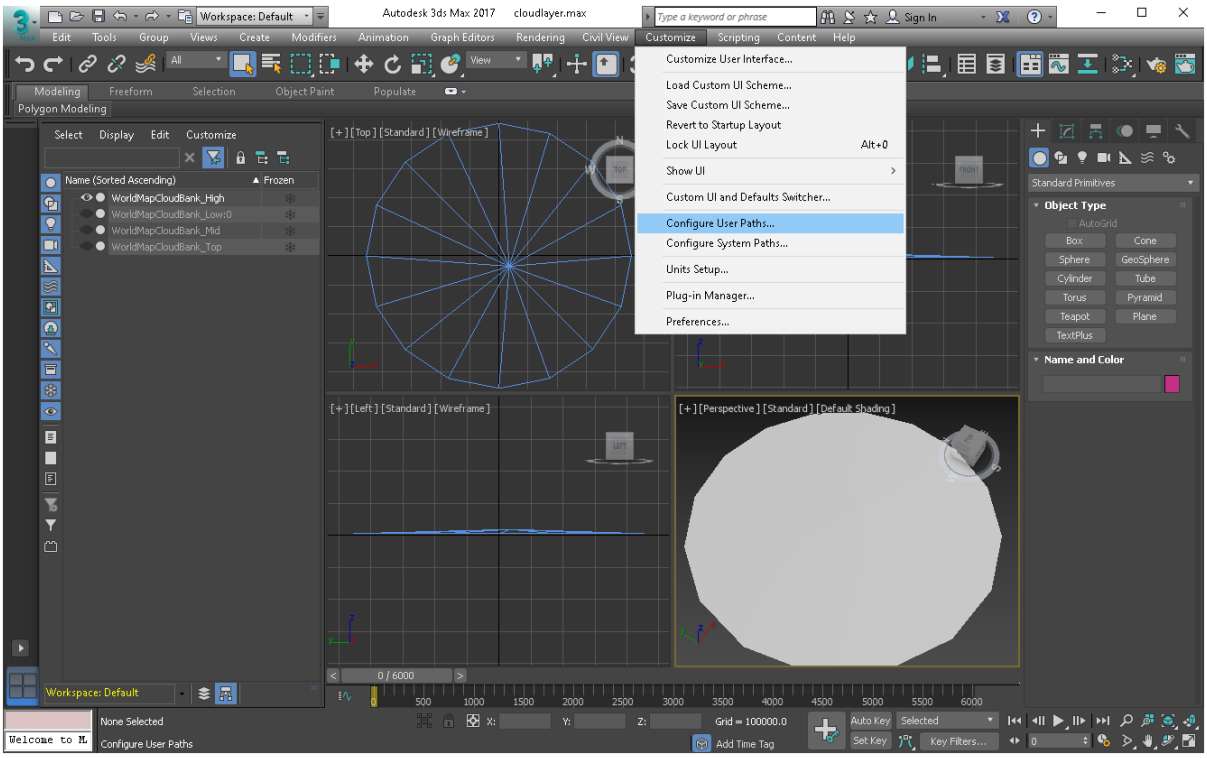


Figure 1421 - Configuring User Paths.

Go to the External Files tab and click on the Add button.

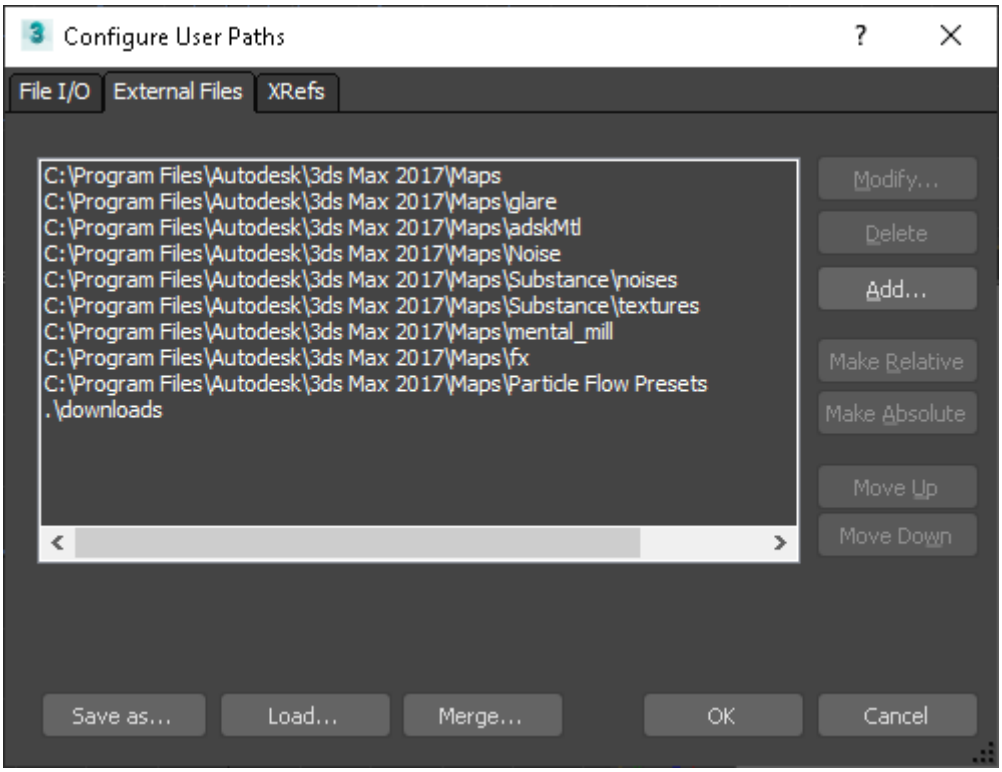


Figure 1422 - Adding a new path for Skyrim textures.

Note: you will need to have exported the textures from the Skyrim .bsa archives first. See Unpacking the BSA Archives for information on how to export textures from this archive.

Select the root path containing the textures and tick 'Add Subpaths'.

Click Use Path.

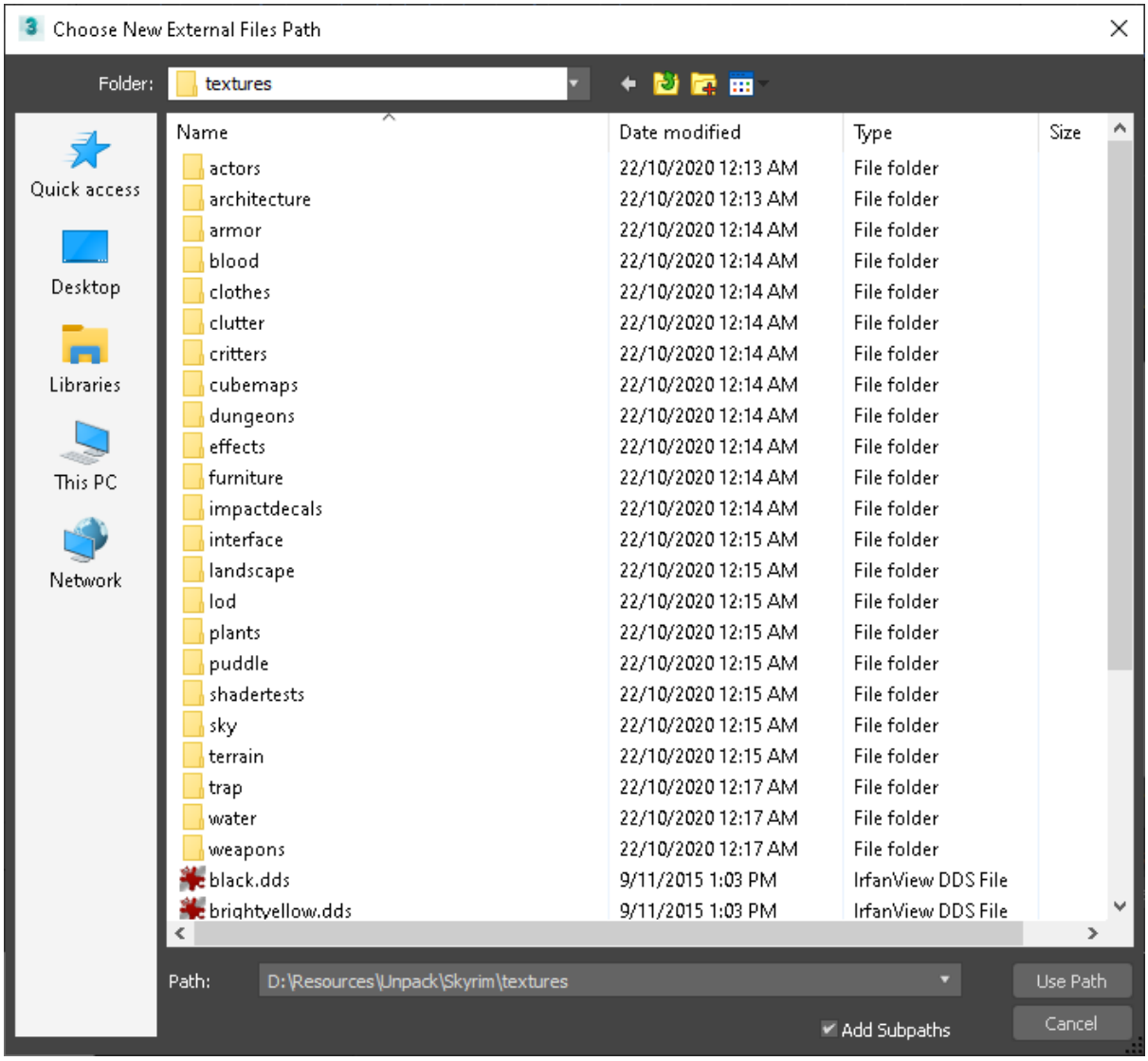


Figure 1423 - Adding all texture subpaths.

Confirm subpaths were automatically added.

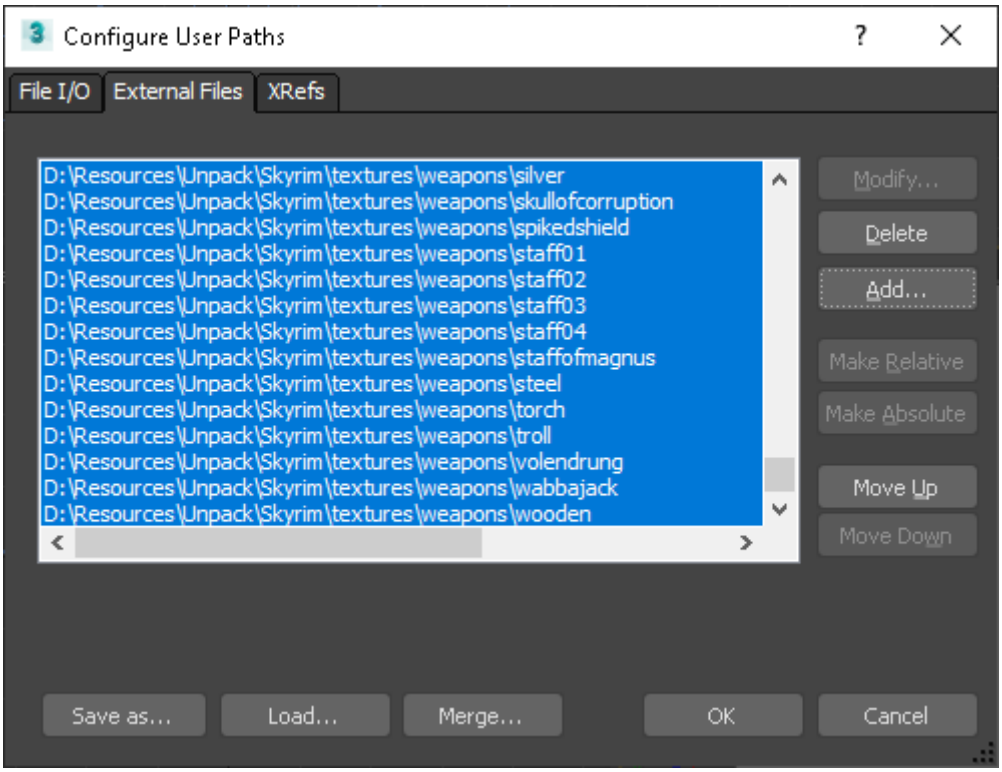


Figure 1424 - All Skyrim texture paths added.

Click OK.

Close 3ds Max and re-open it.

You should now be able to see the cloud texture in the perspective viewport and the problem we need to fix; the textures are now pinching to the central vertex.

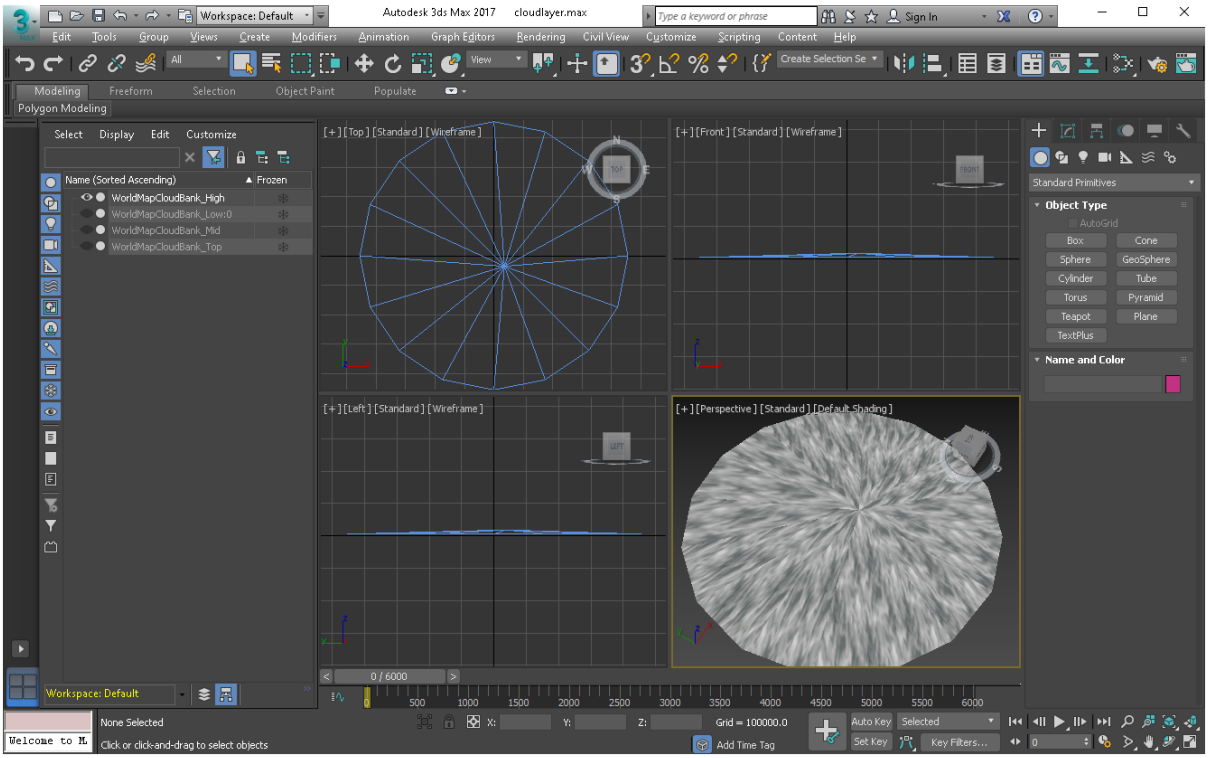


Figure 1425 - Textures are now showing in the perspective viewport.

ADDING TEXTURE PATHS TO NIFSKOPE

Similar to 3ds Max, we need to add texture paths in NifSkope in order to see textures on imported models.

Open NifSkope.

Go to Options > Settings.

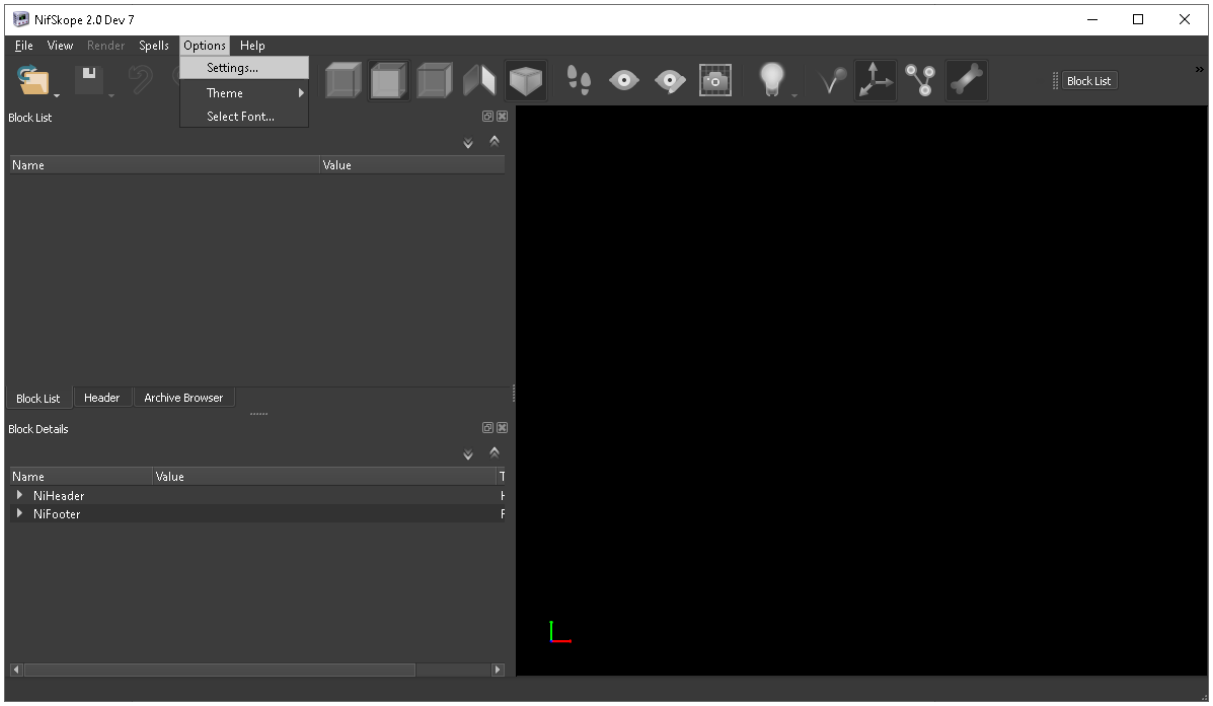


Figure 1426 - Options menu in NifSkope.

Go to the Resources section. Here we can add either a path to loose texture files or link directly to a .bsa archive containing textures.

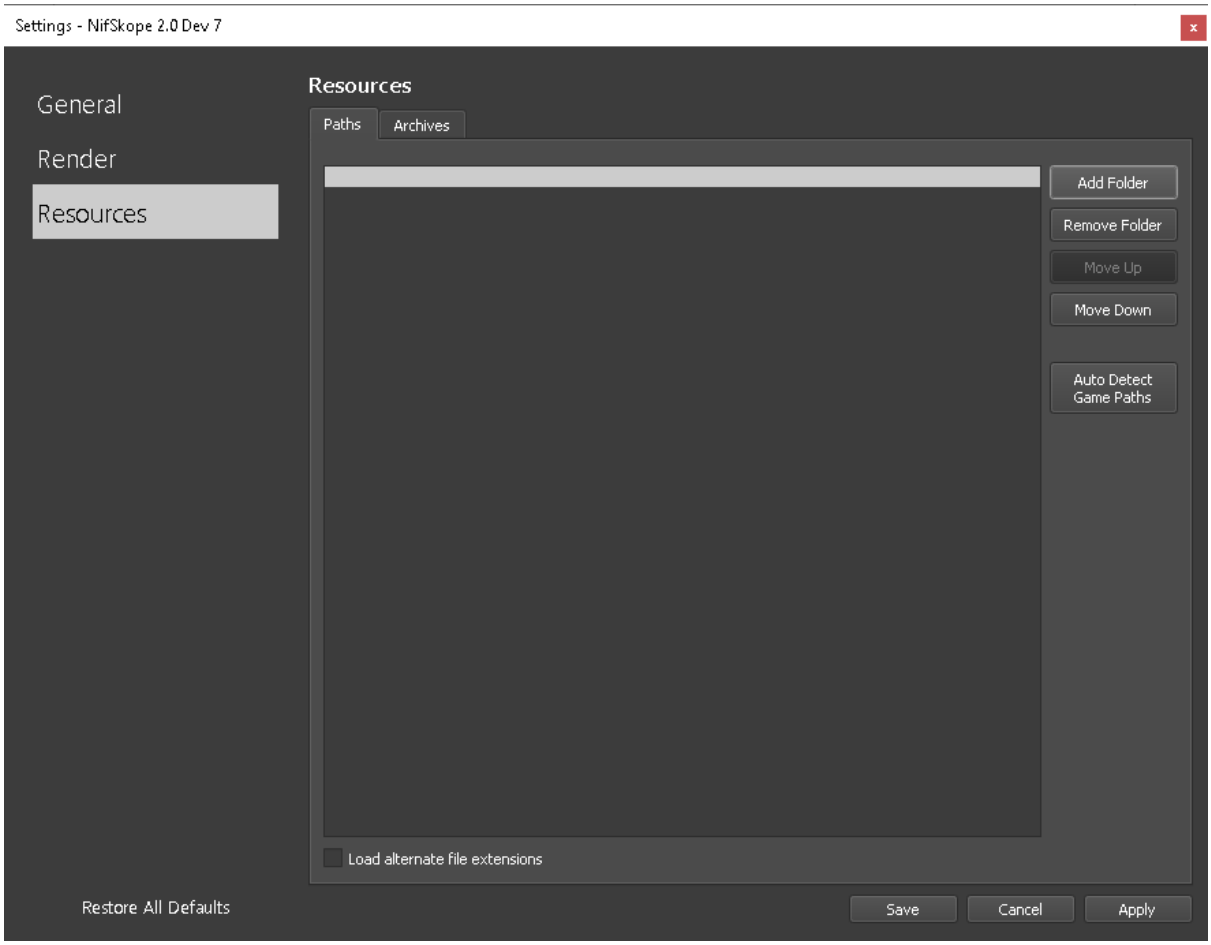


Figure 1427 - Resources.

Let's start by adding a texture path.

On the Paths tab, click Add Folder.

Select the folder path to add to NifSkope and click Select Folder. In the following example I'm going to add the root path containing the texture files for my mod.

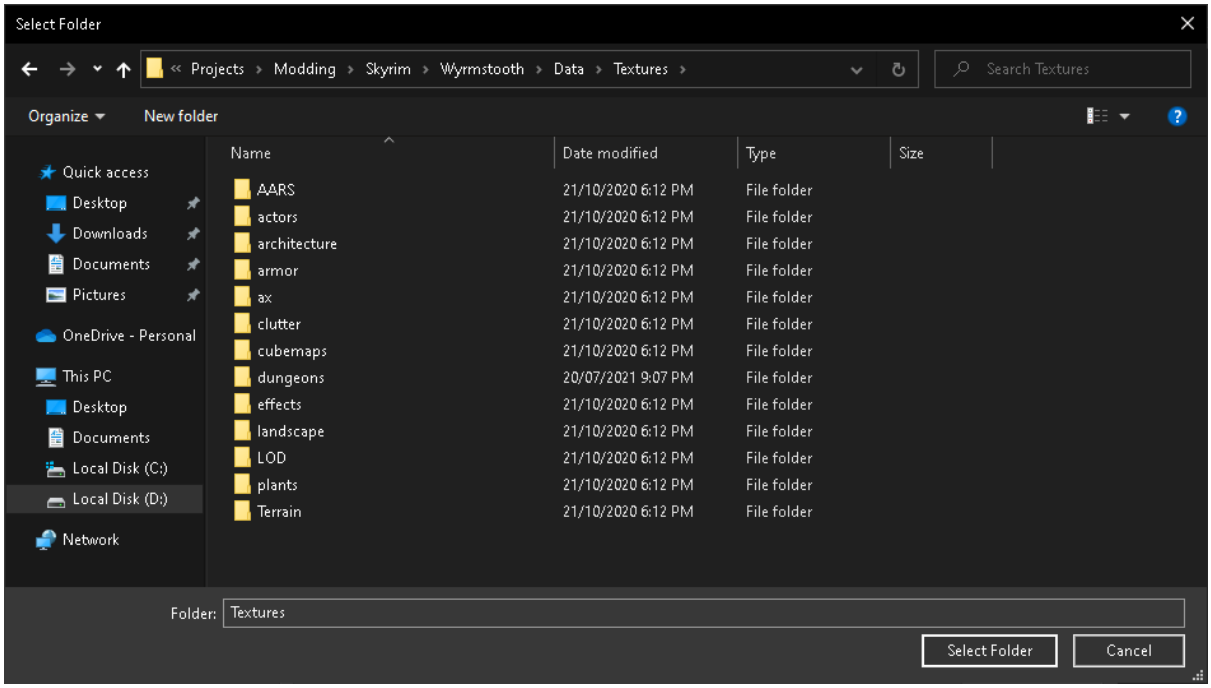


Figure 1428 - Selecting the folder to add to NifSkope.

Confirm that the path has been added to the paths list.

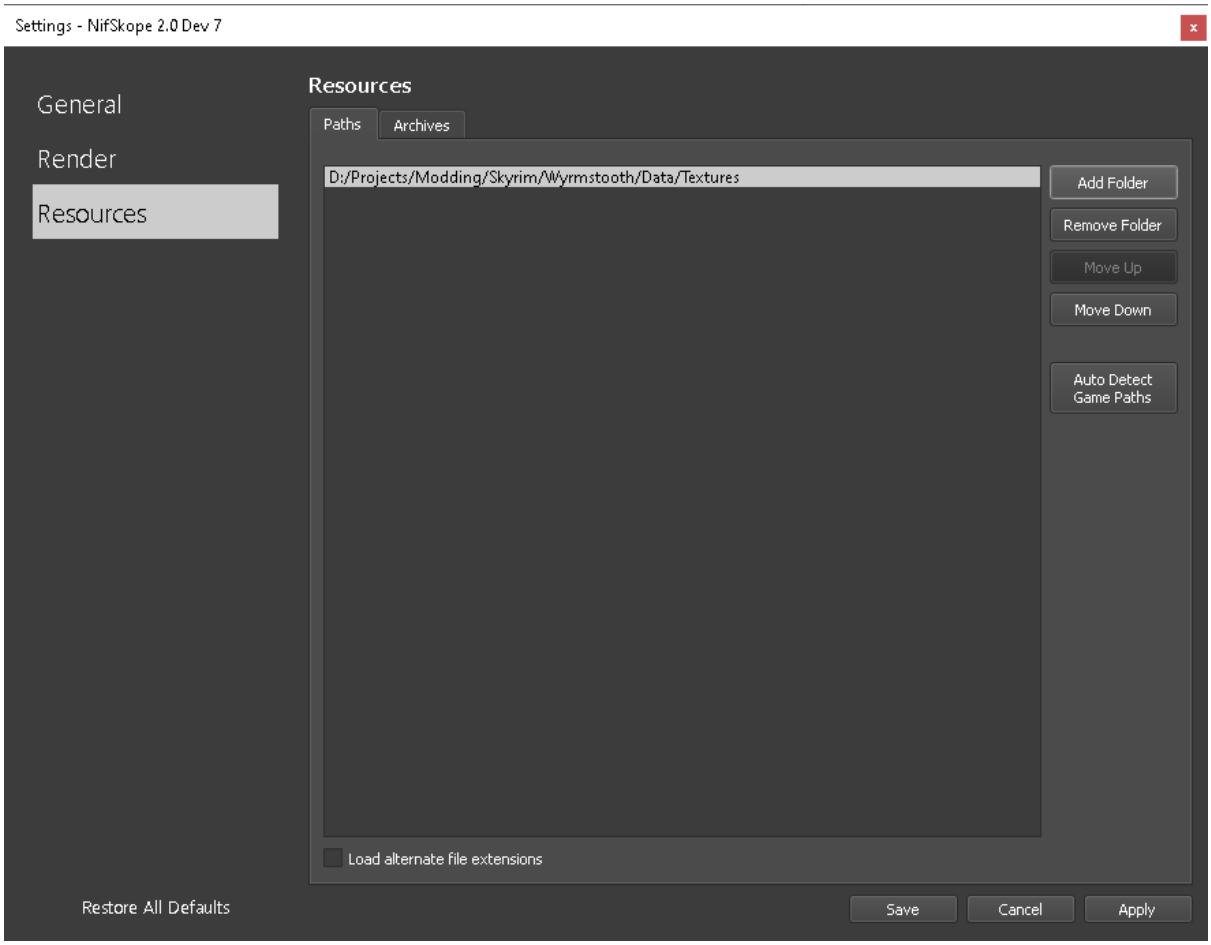


Figure 1429 - Folder path added to NifSkope.

More often than not we're going to want to add a .bsa archive containing the textures.

Go to the Archives tab and click on the Add Archives button.

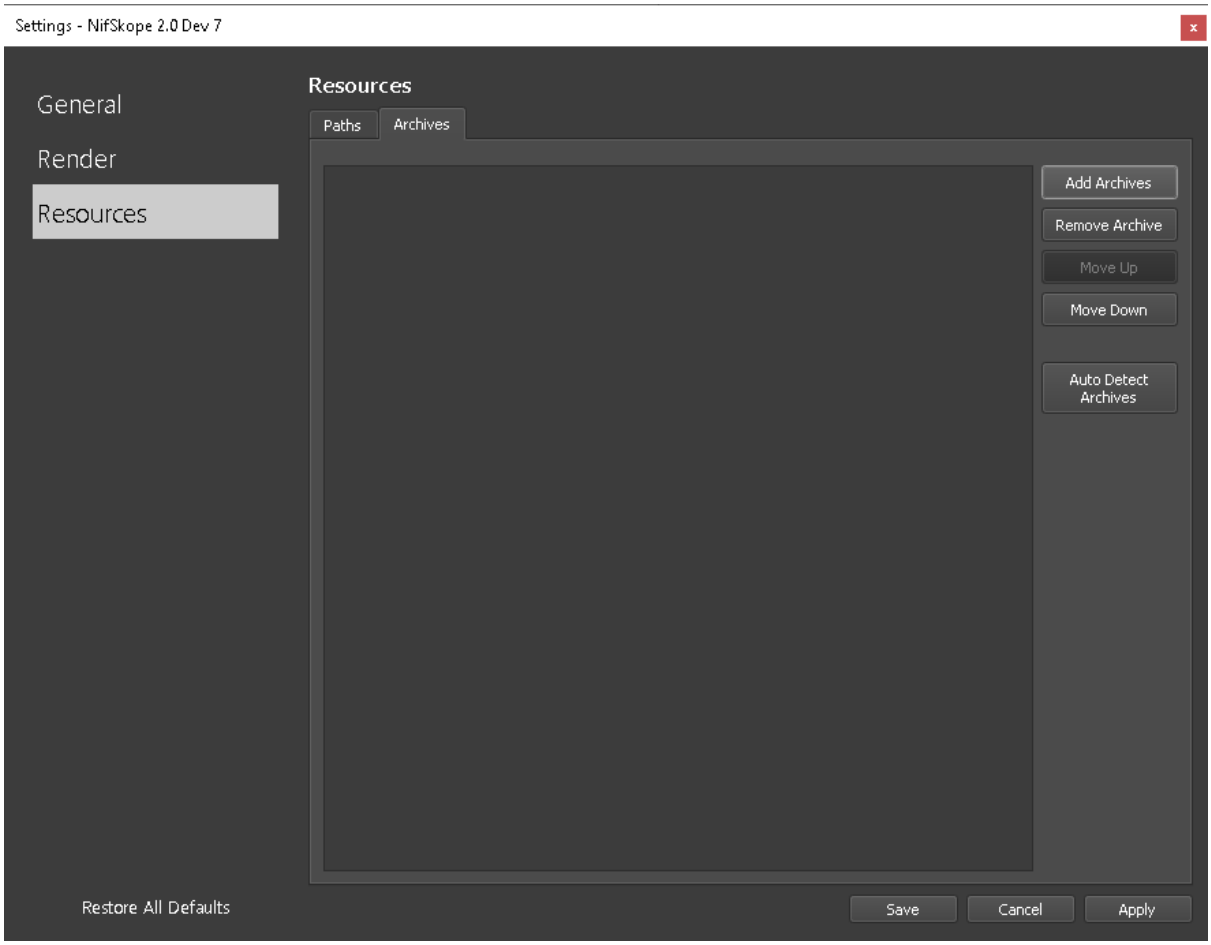


Figure 1430 - Archives tab.

Select a .bsa containing texture files such as Skyrim - Textures.bsa then click Open.

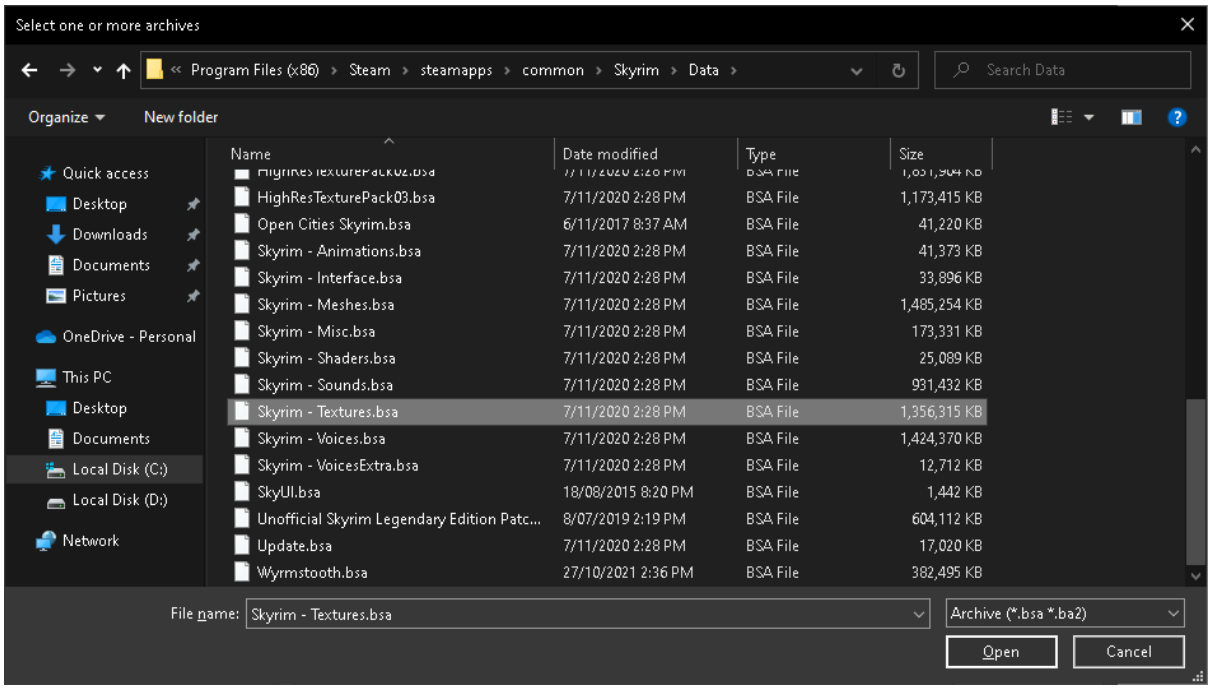


Figure 1431 - Selecting Skyrim - Textures.bsa.

Confirm the .bsa has been added to the archives list.

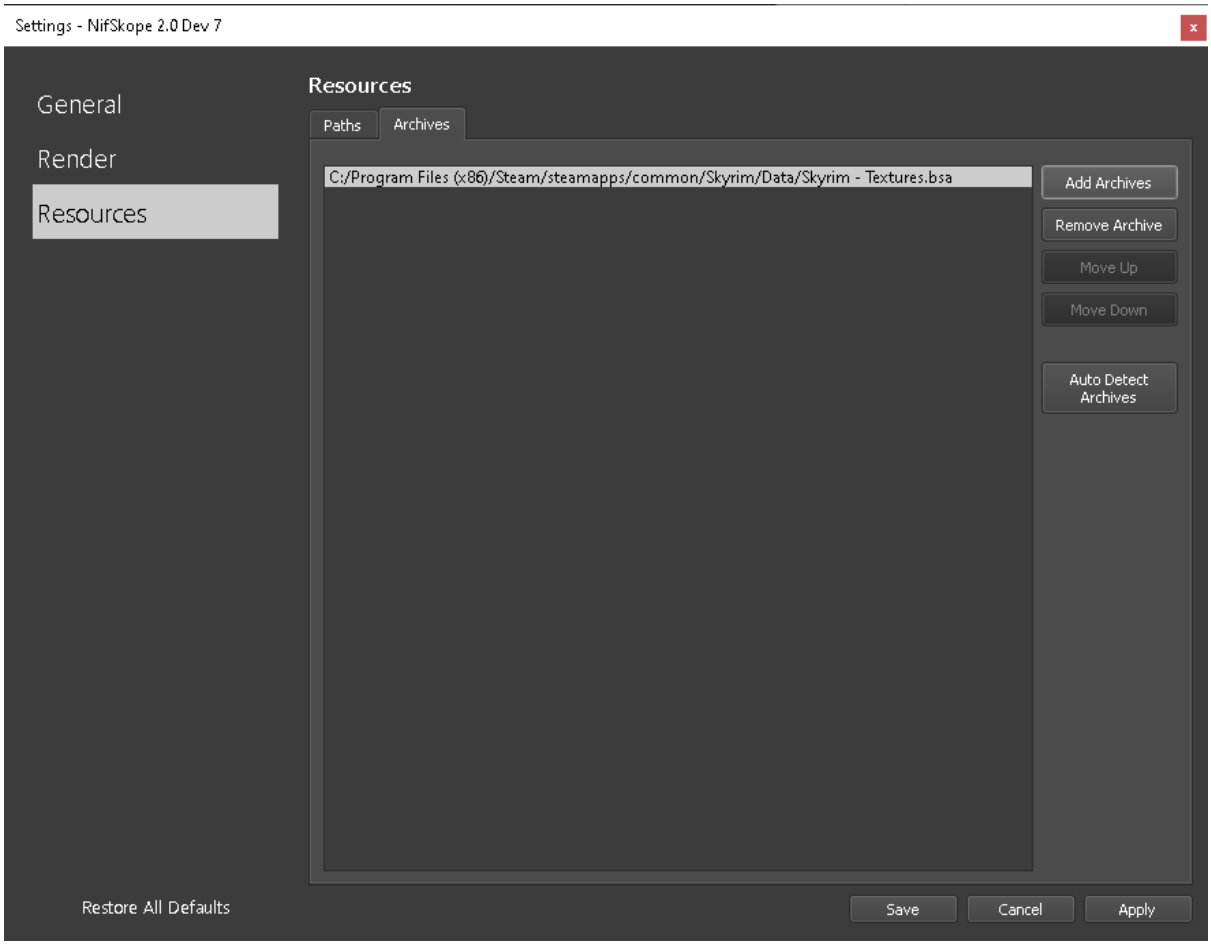


Figure 1432 - .bsa added to NifSkope.

Once you've adding paths and/or .bsa archives, click Save to close out of Settings.

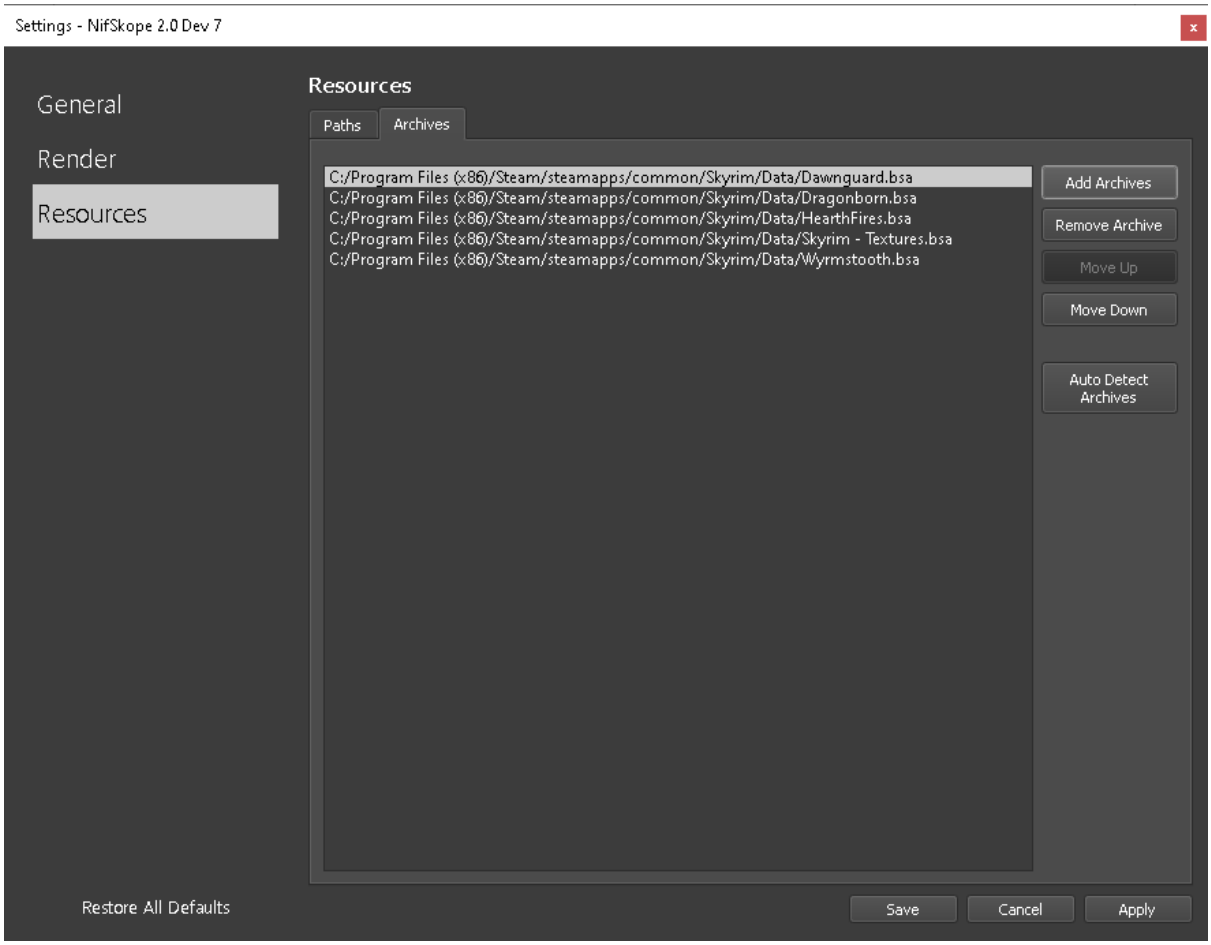


Figure 1433 - More .bsa archives added.

Restart NifSkope and load a .nif file. You should now be able to see its textures.

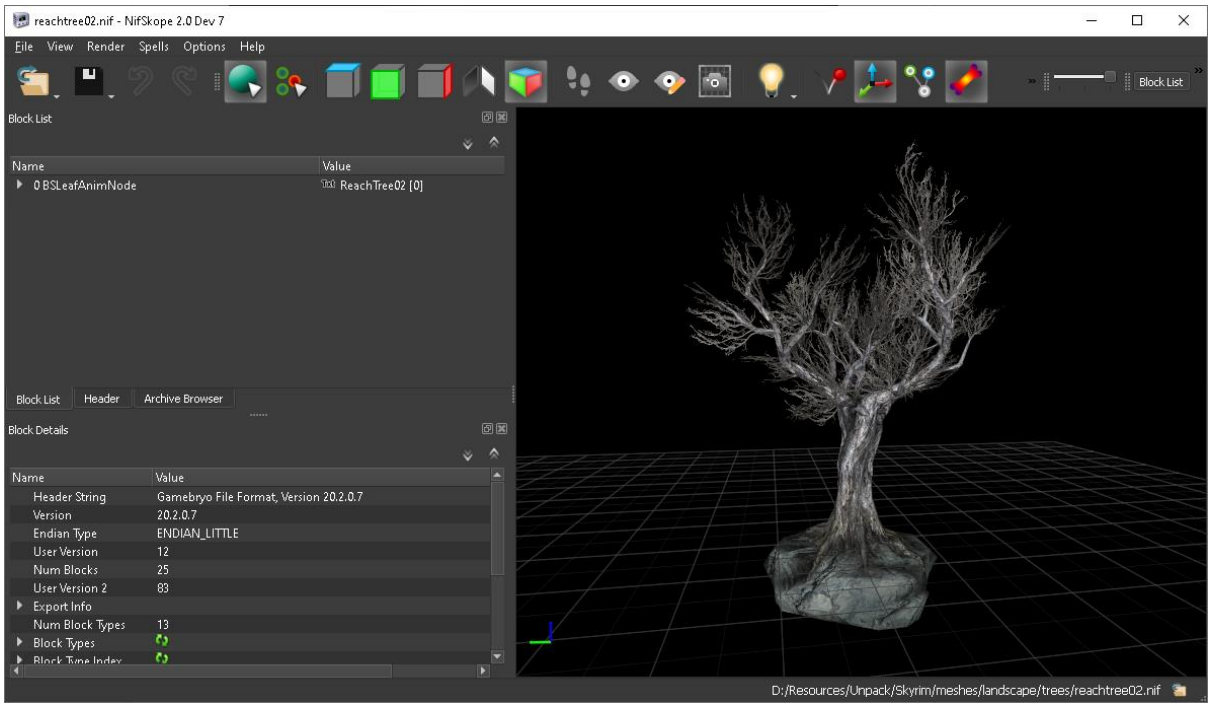


Figure 1434 - Textures shown on imported .nif.

OPTIMIZING INTERIOR CELLS WITH ROOMBOUNDS

We can optimize the performance of an interior cell with roombounds. Roombounds allow us to hide and show parts of an interior cell based on what the player has line-of-sight to.

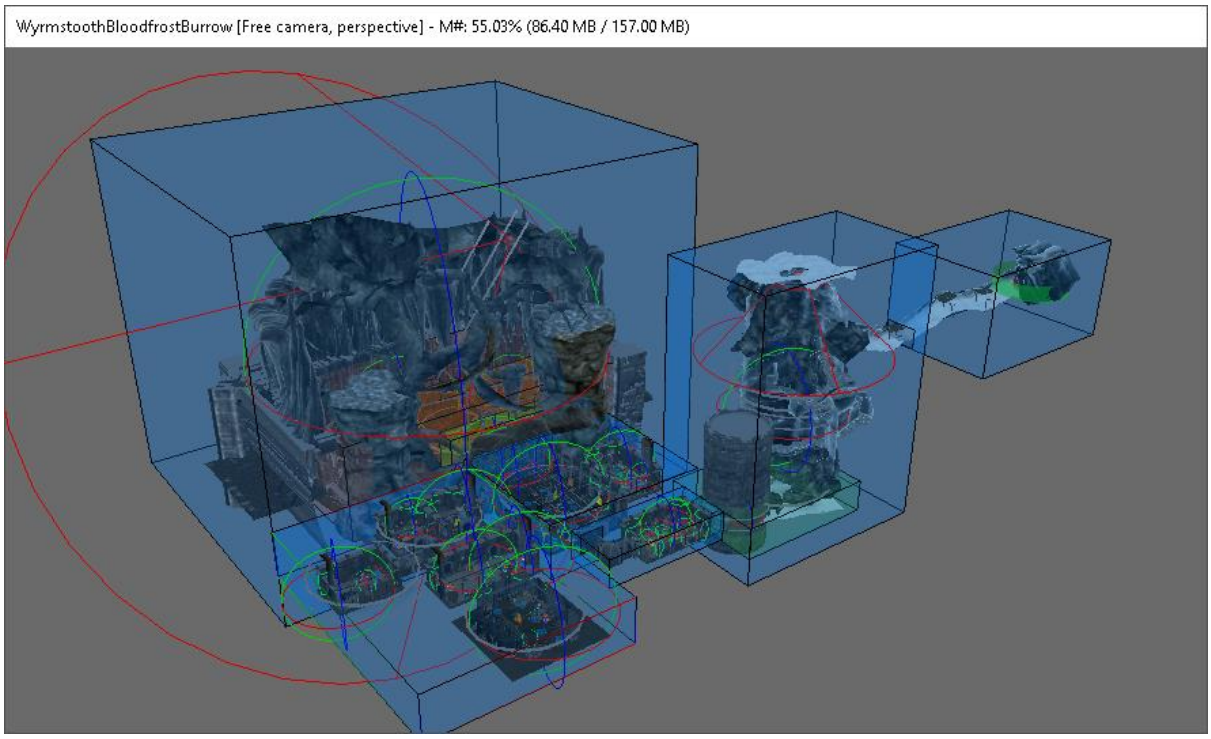


Figure 1435 - Roombounds in a dungeon.

As per the screenshot above, we can encapsulate a room or collection of rooms in a roombound and link them together using portals.

Important: You will need to ensure all parts of your interior are in a roombound and that each roombound is linked to another via a portal or parts of your dungeon may not be visible to the player.

Press 'M' to toggle markers. By default, roombounds won't be shown so you'll need to go to View > Show/Hide Window and tick 'Portals and Rooms'.

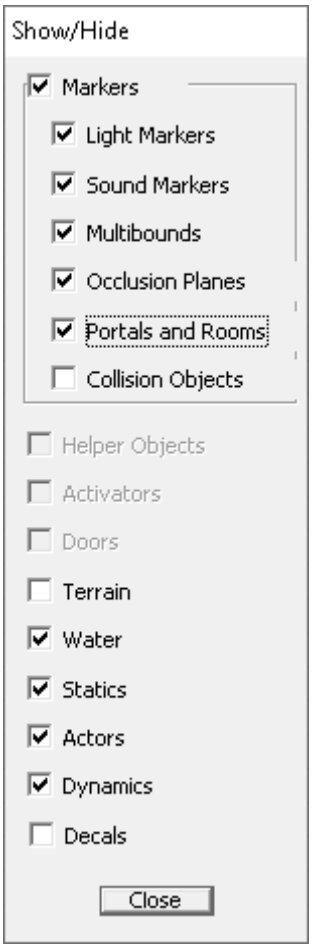


Figure 1436 - Portals and Rooms ticked.

Click Close.

To add a roombound, select a part of your interior's tileset and click on the 'Create a Roombound' button in the toolbar.

You should see a blue box appear in the render window. The resize gizmo should also be toggled on allowing you to resize it. If it isn't, you can toggle it on by pressing '2'.

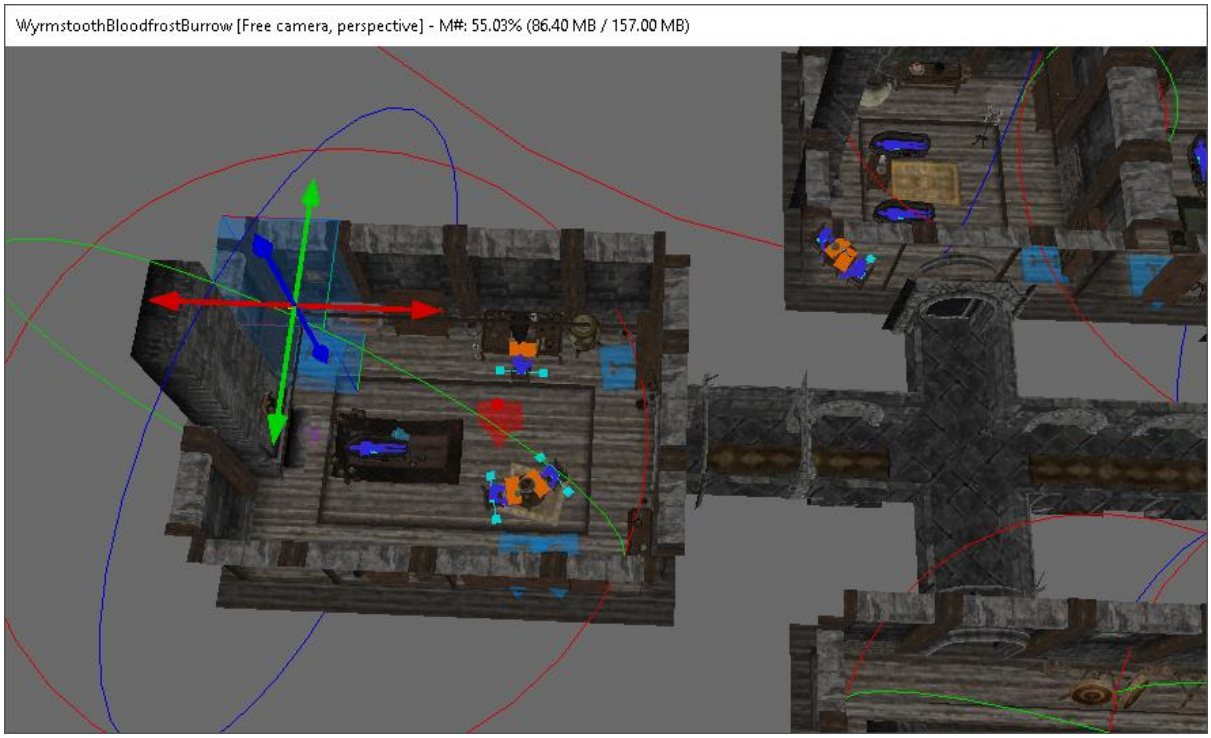


Figure 1437 - Adding a roombound.

Use the resize gizmo to resize the roombound to cover a portion of your dungeon.

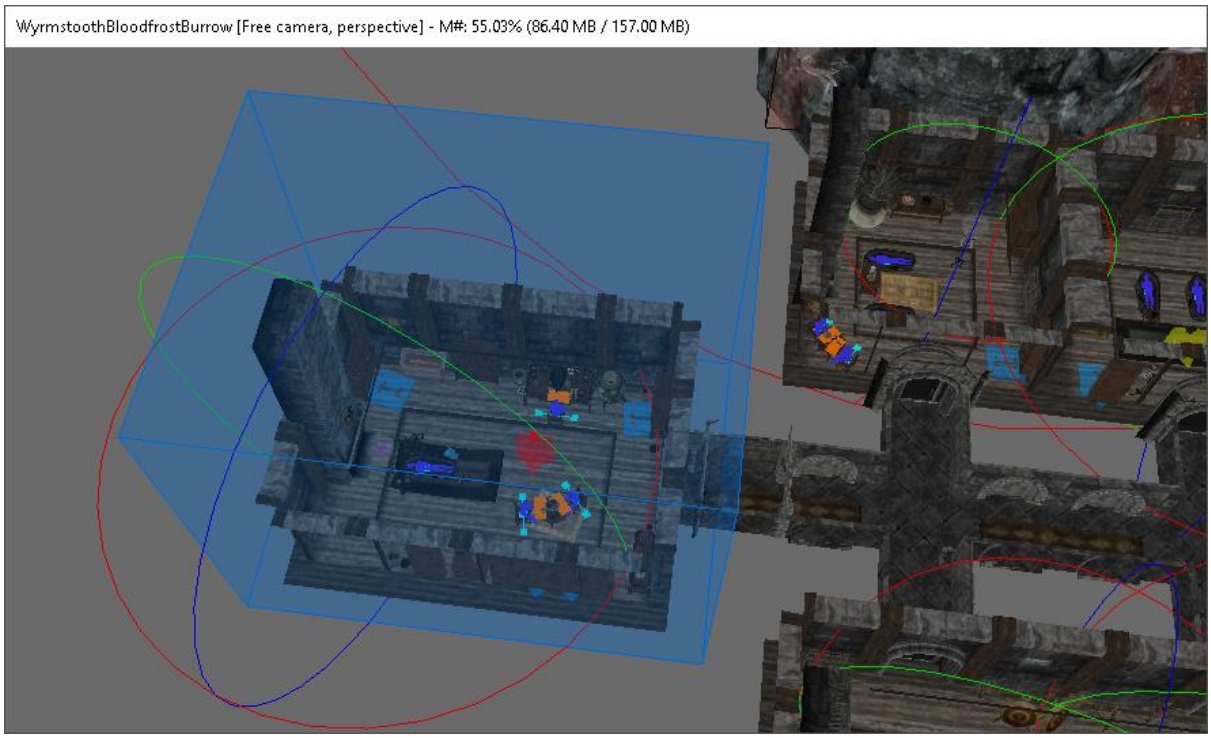


Figure 1438 - Roombound resized.

Important: What's in or out of a roombound is determined by an object's pivot point.

For example, the corner selected in the screenshot below is inside of the roombound.

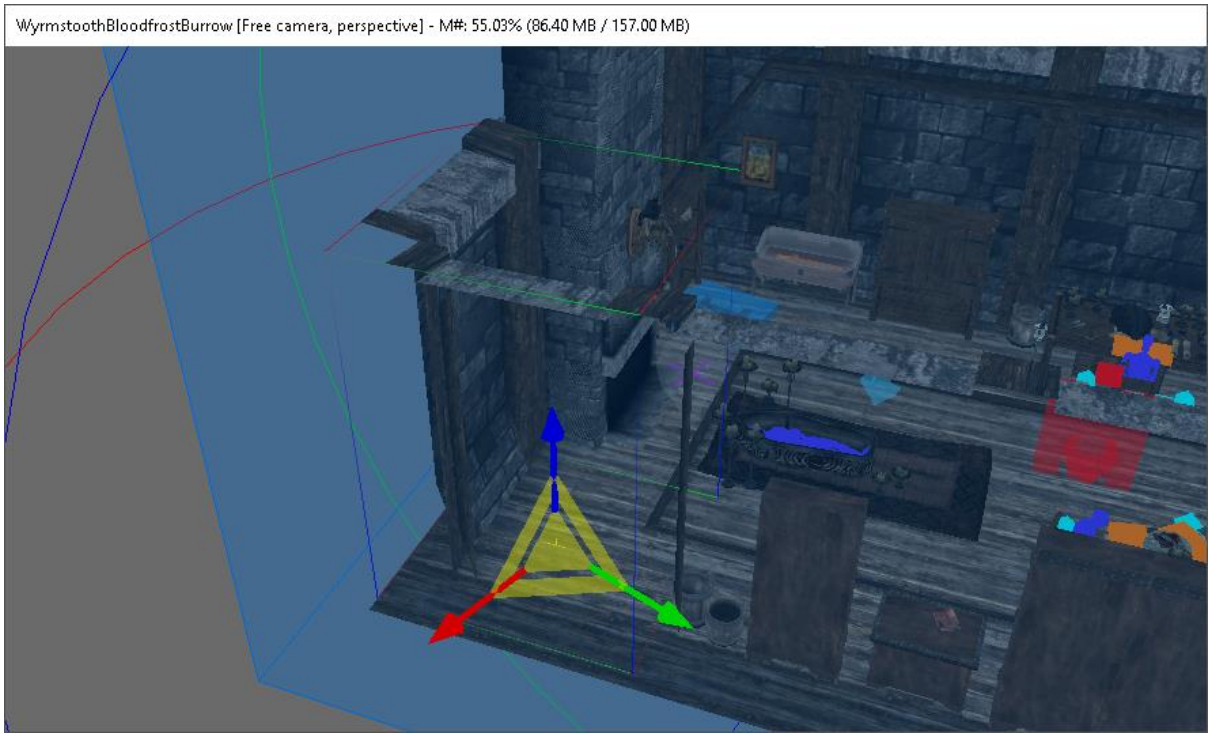


Figure 1439 - An object inside of the roombound.

Repeat this step to add more roombounds to cover the rest of your interior.

Note: It doesn't matter at this stage if they overlap or not.

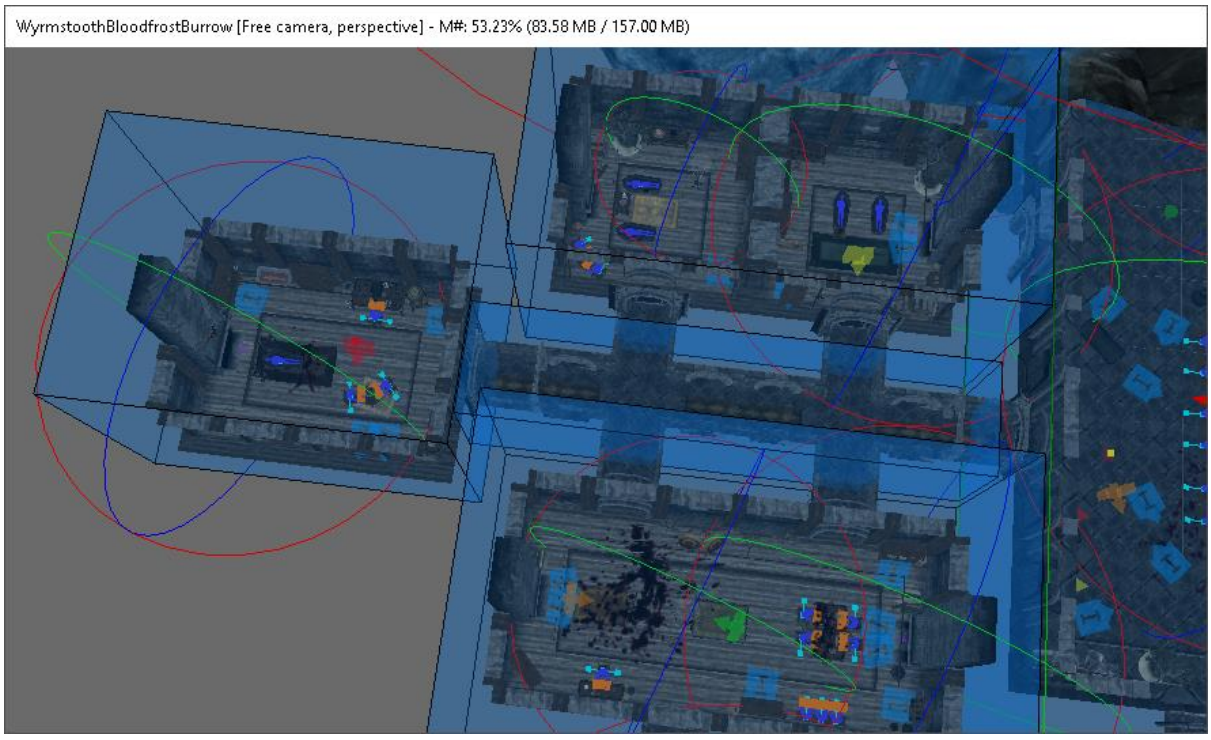


Figure 1440 - More roombounds added.

We can join two roombounds together using portals.

Select the first roombound to add a portal to.

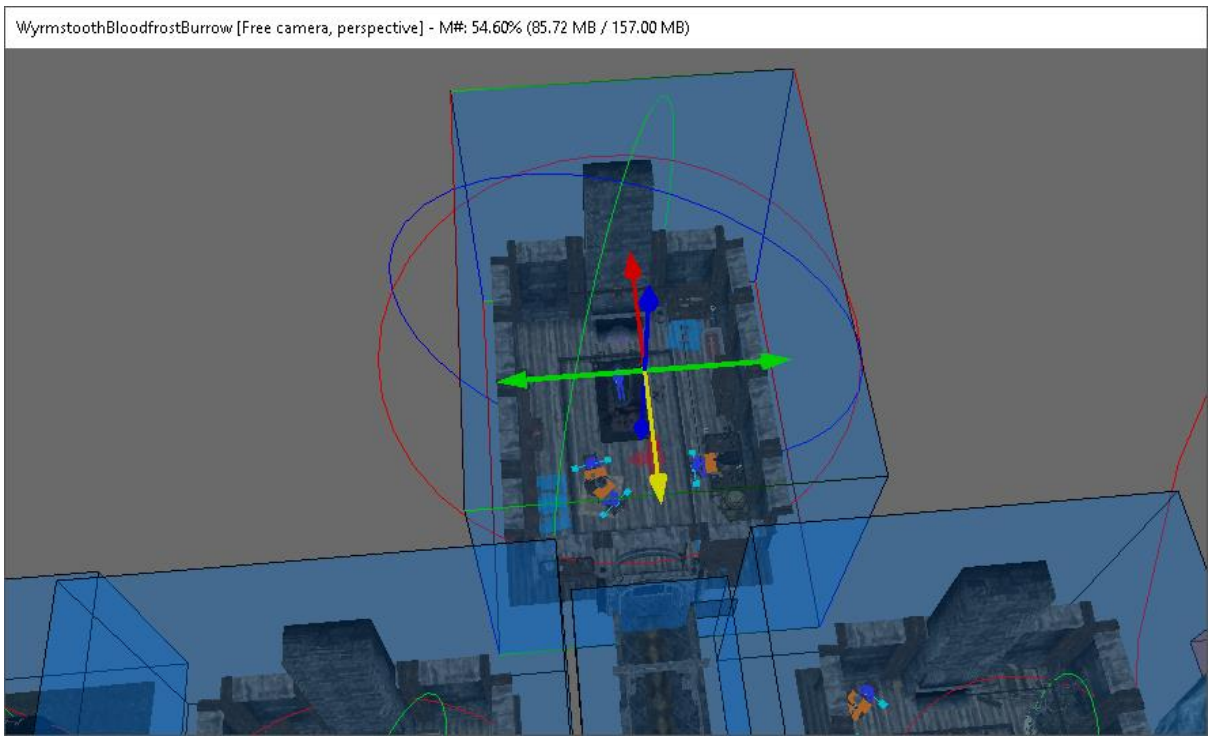


Figure 1441 - Roombound selected.

Click on the 'Create a Portal' button in the toolbar and click on the selected roombound in the render window.

Important: You need to click on the side of the roombound you want to add a portal to.

The portal should appear as a flat 2-dimensional gray rectangle.

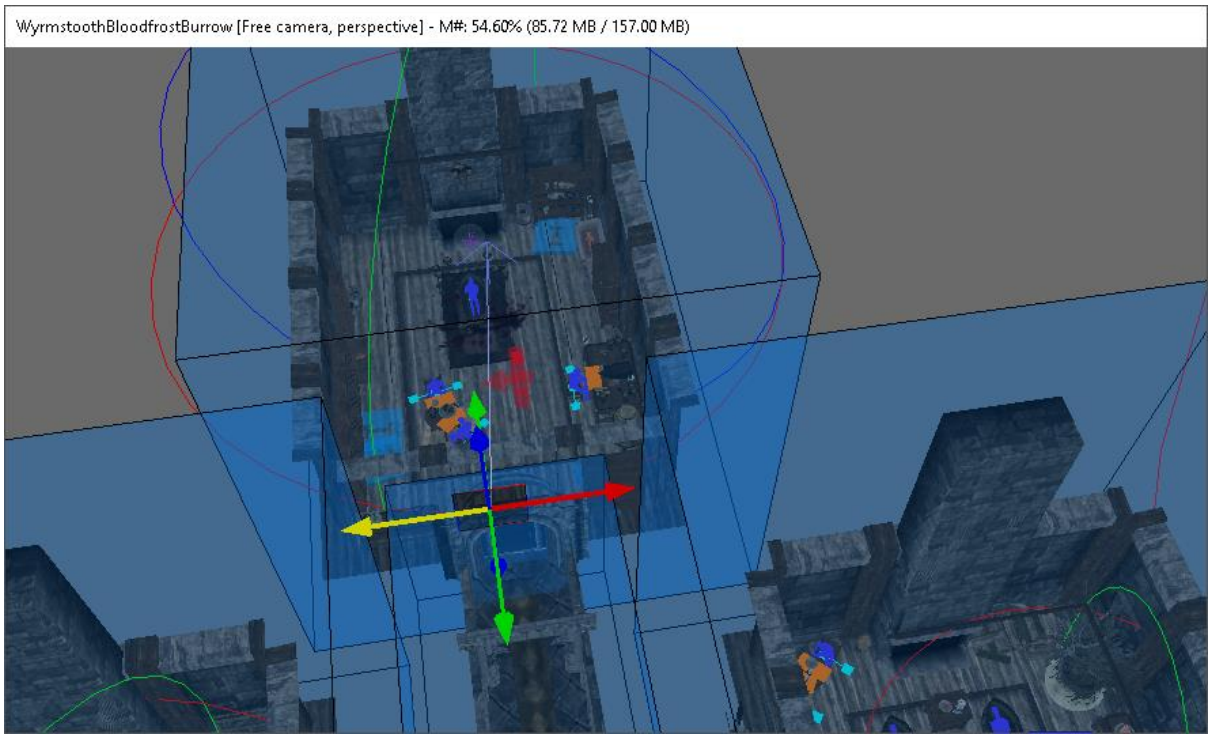


Figure 1442 - Portal added to roombound.

In the screenshot above we can see that the portal was automatically linked to the topmost roombound. Next, we'll need to link it to the roombound below it to link both roombounds together.

With the portal still selected, CTRL-click on the roombound you want to connect it to.

Important: You should have both the portal and the roombound selected.

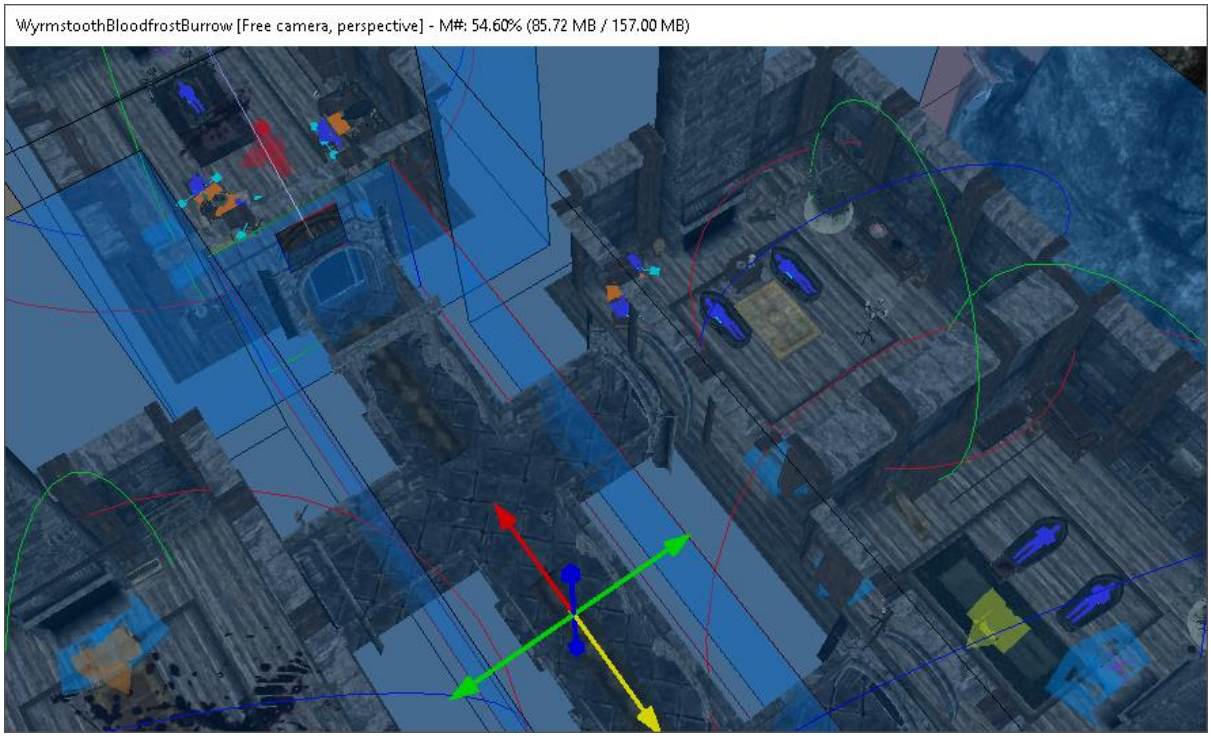


Figure 1443 - Selecting the roombound and portal to connect.

Click on the 'Link Portal to Room' button in the toolbar. The white/blue arrow on the portal should now be pointing to both roombounds.

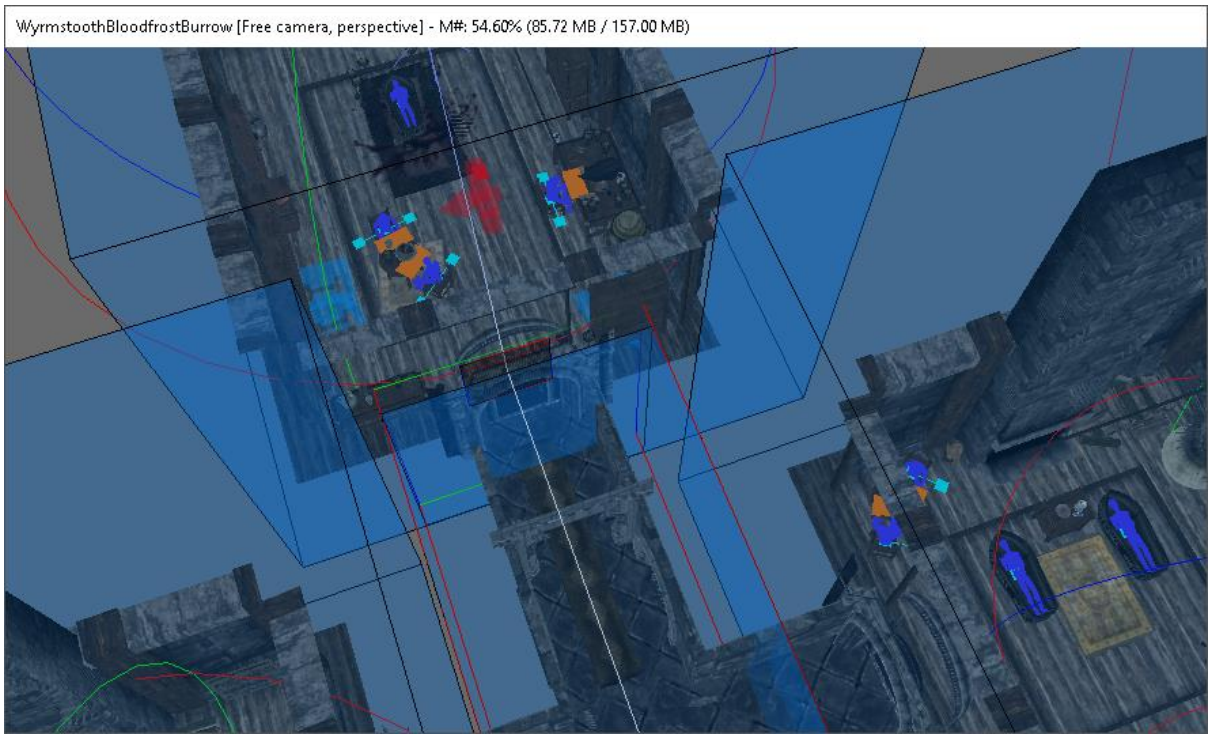


Figure 1444 - Connecting the portal to the selected roombound.

Note: Clicking on the 'Enter Portal Mode' button in the toolbar prevents you from selecting anything else in the render window except portal objects.

Select the portal object.

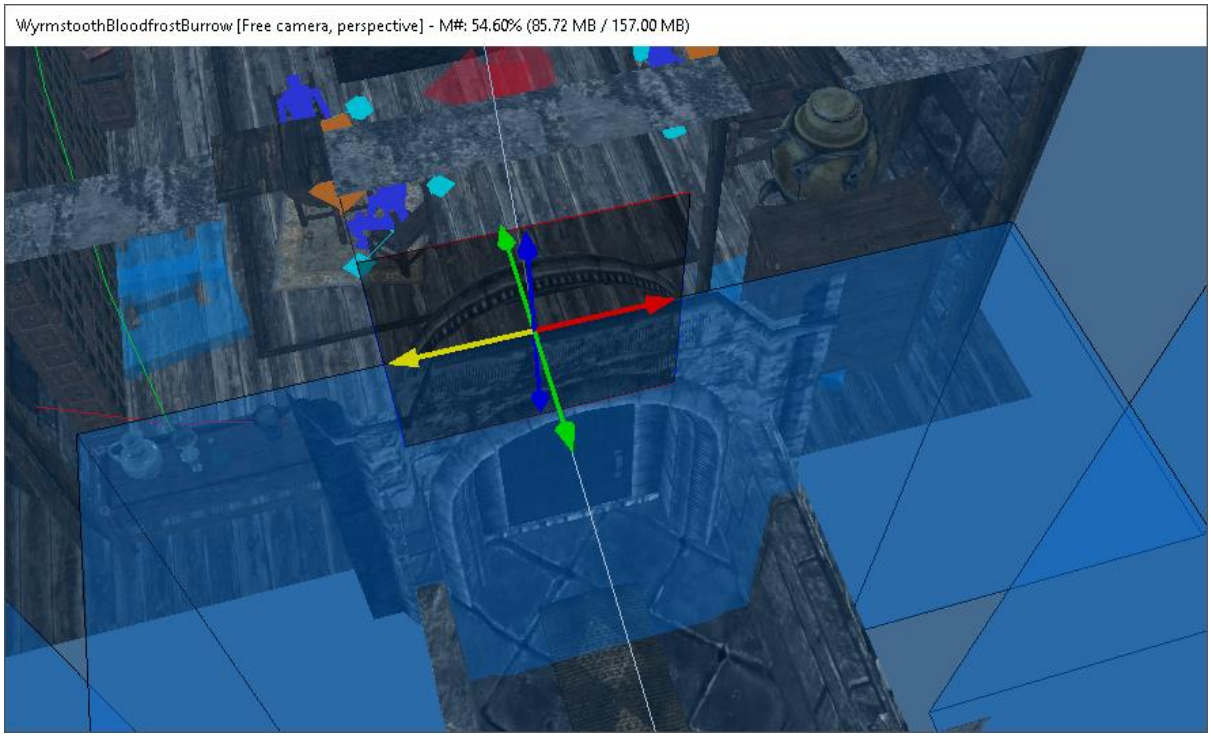


Figure 1445 - Selecting the portal object.

Use the resize gizmo to scale the portal just outside the bounds of the passageway.

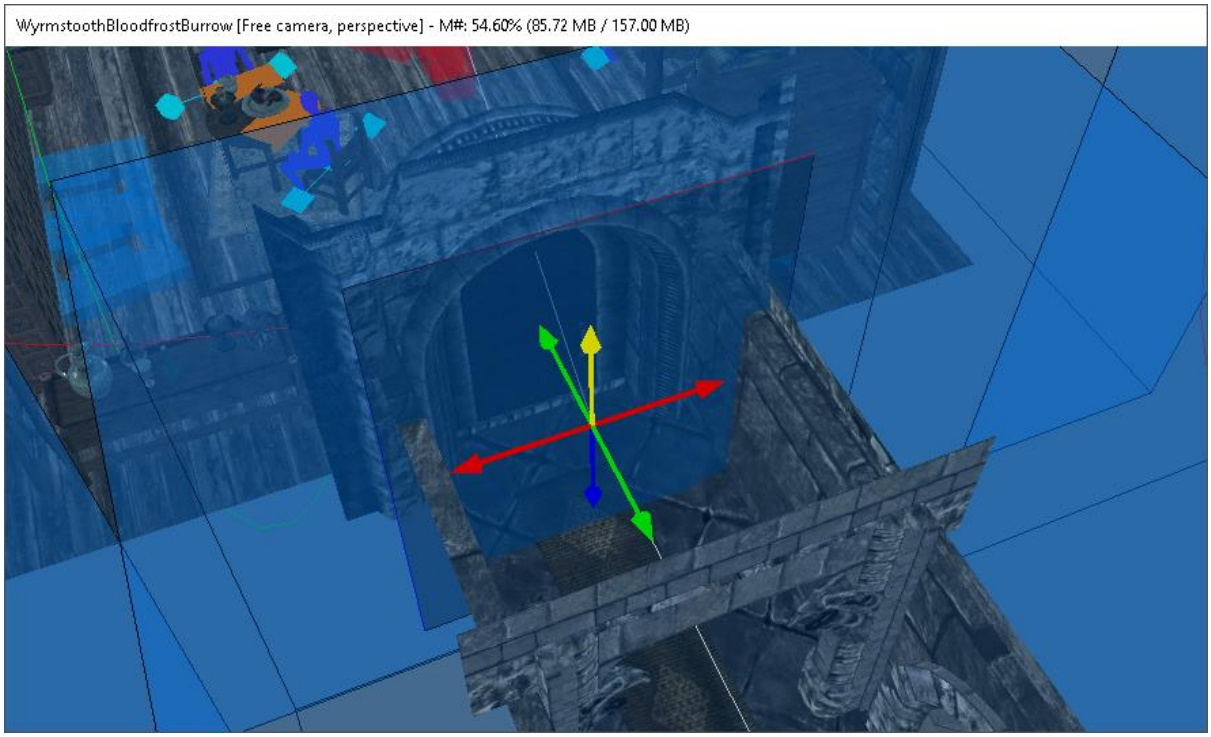


Figure 1446 - Portal resized.

Here's a screenshot of the portal from within the passageway:



Figure 1447 - Portal from inside the passageway.

Important: When you linked the roombound to the portal you may have noticed that the side of the roombound facing the portal was moved to sit flush with it, automatically removing the gap if there was one.

After linking a roombound to a portal do not move either roombounds or the portal or you will need to delete and relink the portal.

Sometimes the portal will be created along the wrong edge as per the screenshot below:

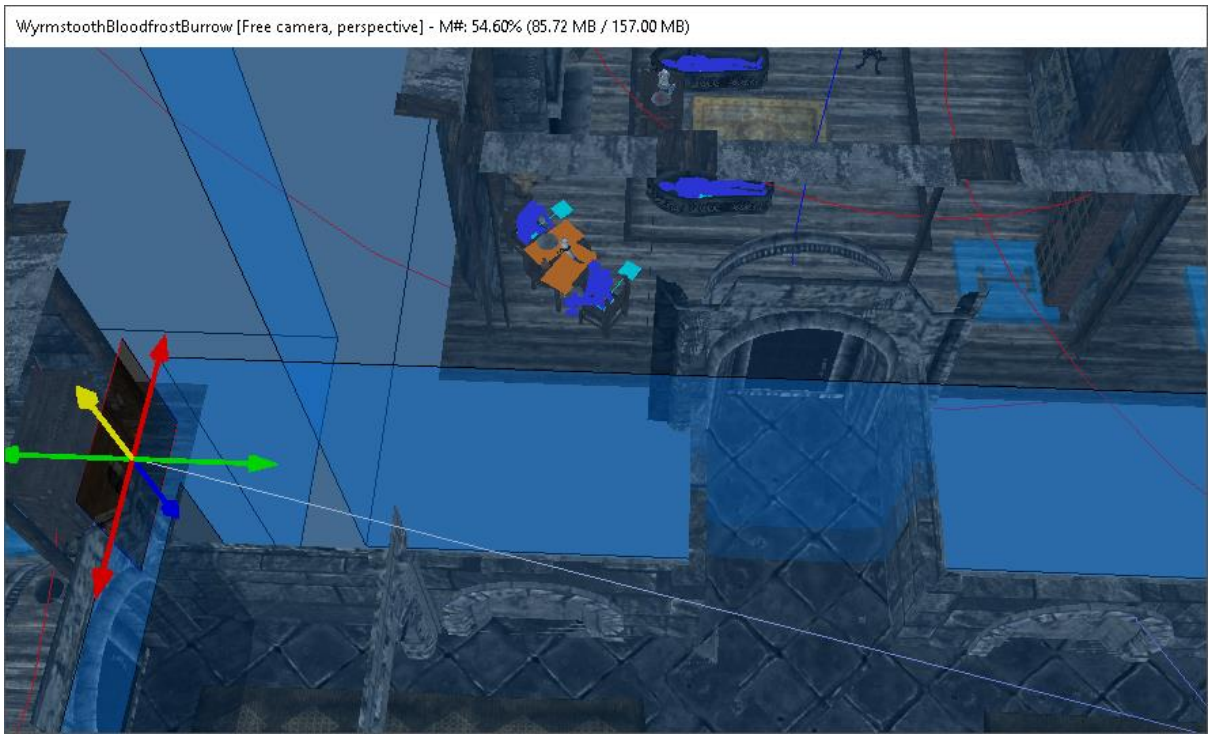


Figure 1448 - Portal created along the wrong edge.

To correct this, click on the 'Snap to Angle' and 'Local gizmo orientation' buttons in the toolbar to toggle them both on and rotate the portal to face the right direction.

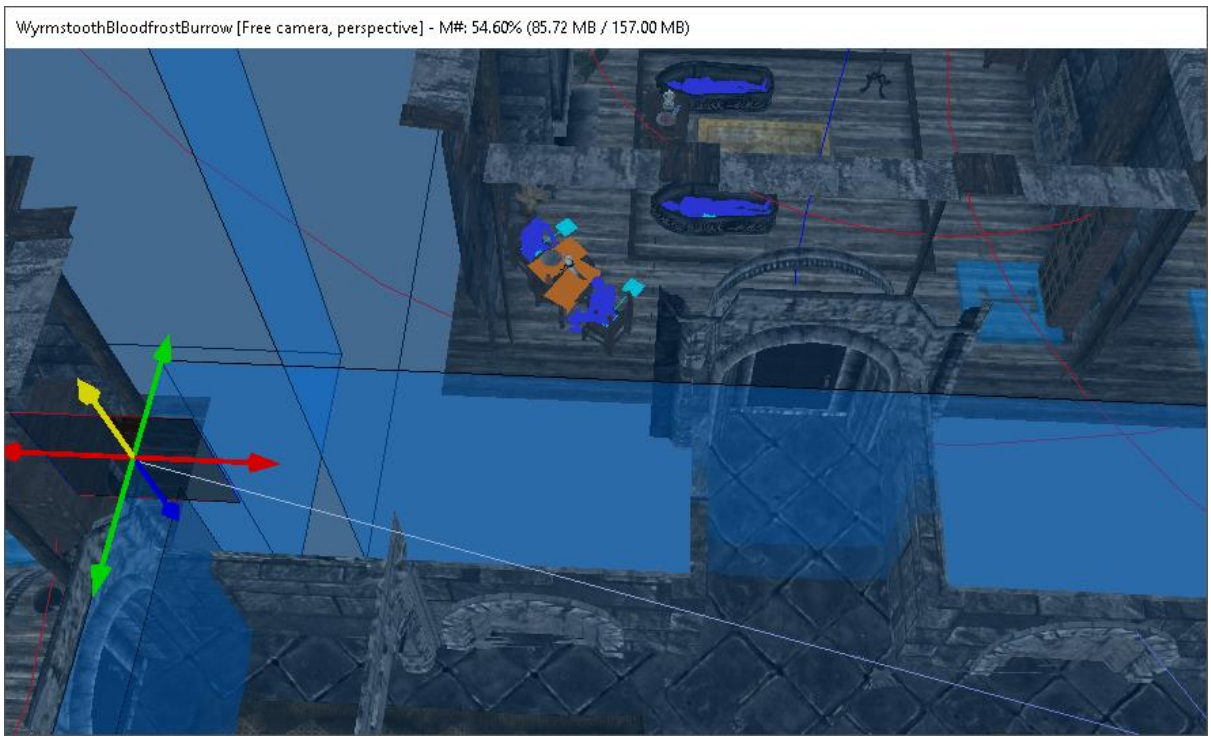


Figure 1449 - Portal rotated.

Then use the resize gizmo to move it into position and resize it to cover the width and height of the hallway.

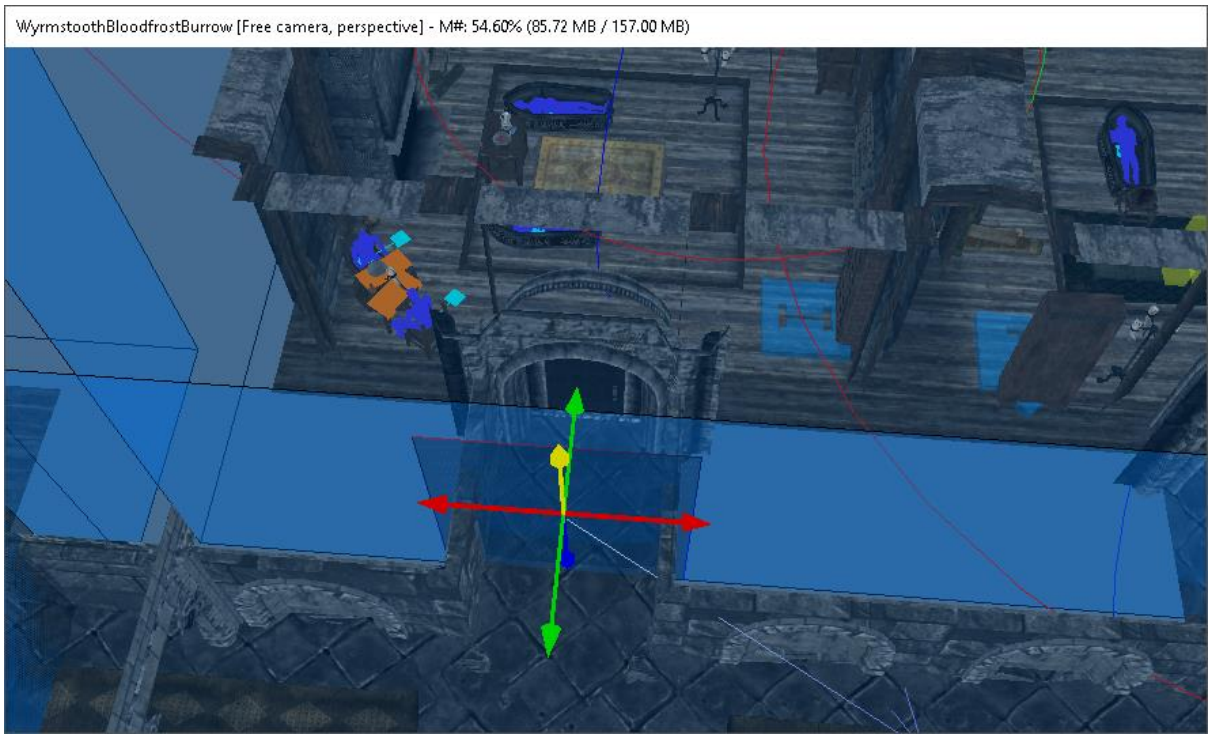


Figure 1450 - Portal moved into position.

And here it is linked to the roombound above it.

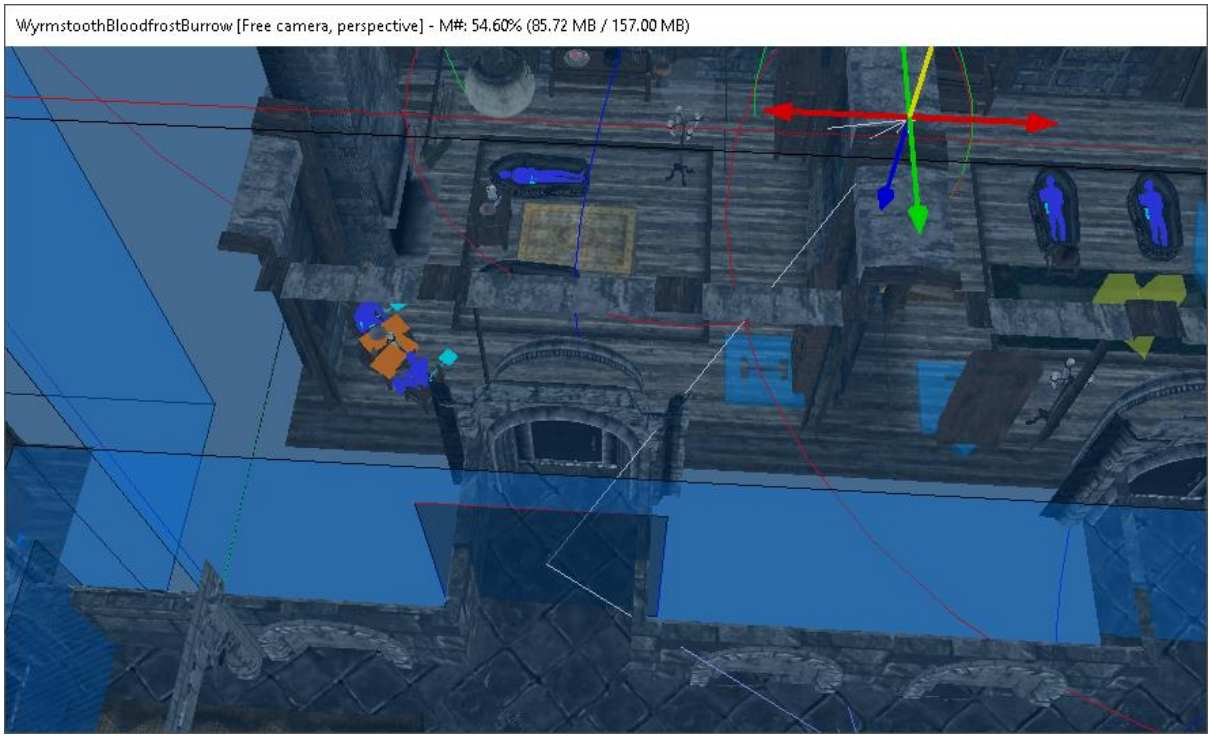


Figure 1451 - Portal linked to another roombound.

Portals can be duplicated as per the screenshot below to cover multiple passageways to a room:

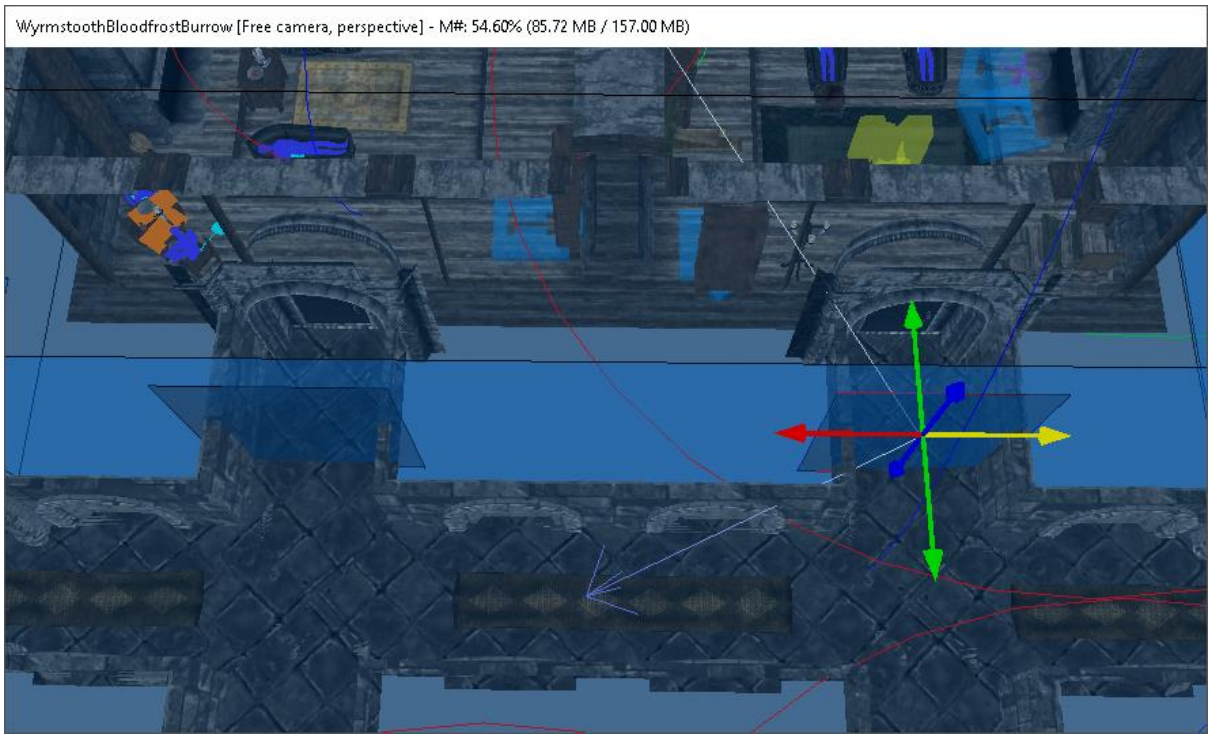


Figure 1452 - Duplicating portals.

Repeat these steps to link the rest of the roombounds to portals.

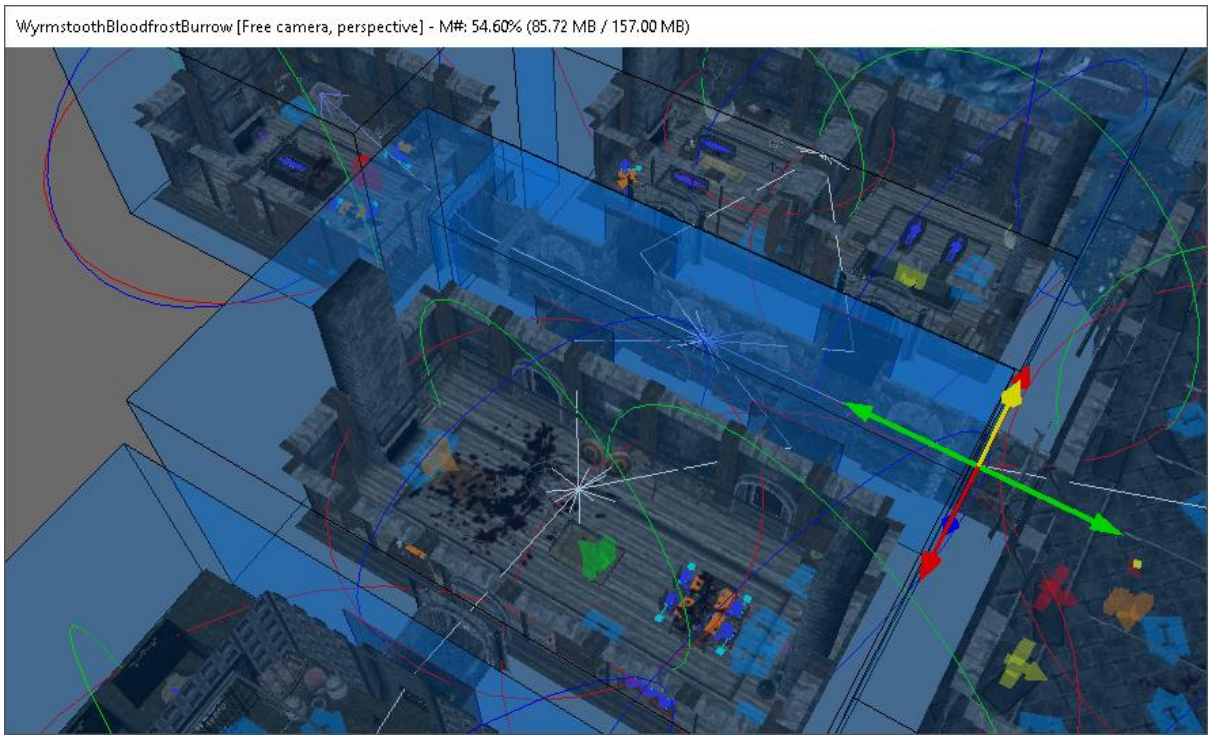


Figure 1453 - Roombounds linked together.

When working with roombounds, I would strongly recommend keeping an eye on your EditorWarnings.txt file.

You can find this file in your Skyrim or Skyrim Special Edition installation folder.

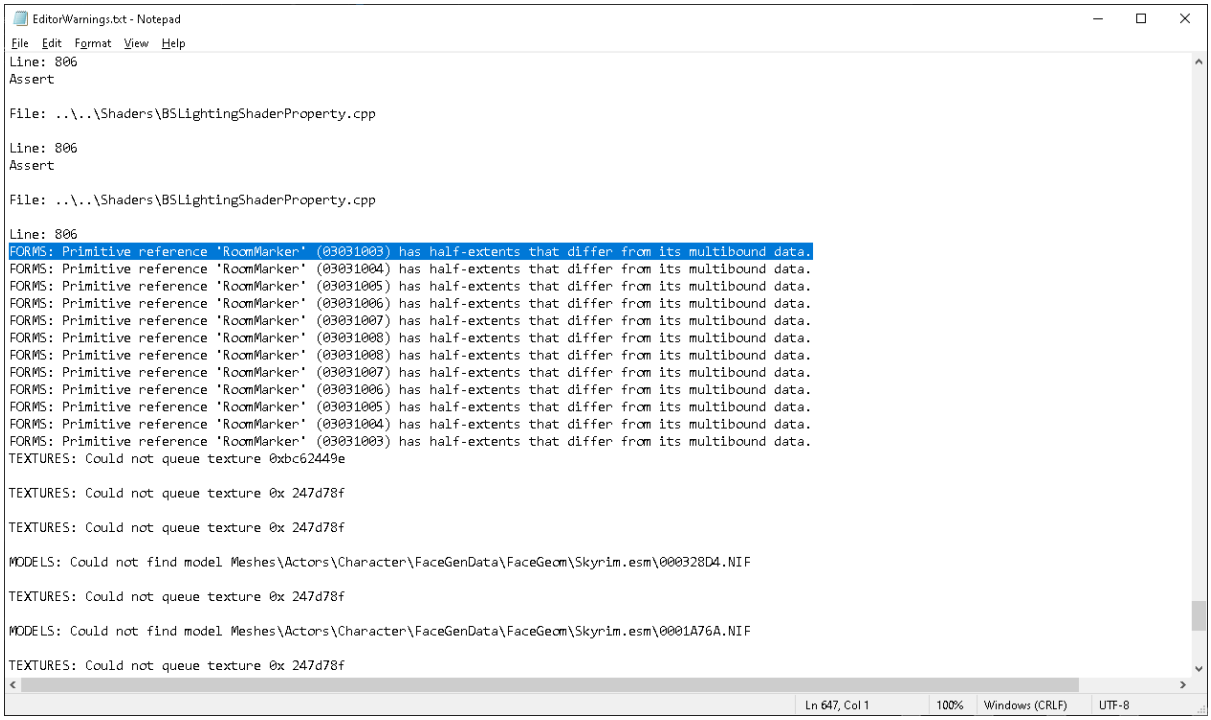


Figure 1454 - EditorWarnings.txt.

If you spot a warning that says 'FORMS: Primitive reference 'RoomMarker' (<Ref ID>) has half-extents that differ from its multibound data.' you may need to delete and recreate any portals attached to that roombound.

If that doesn't fix the issue, you may have to go so far as to delete the roombound and recreate it.

VERSION CONTROL AND MERGING CHANGES

Version control allows multiple developers to work on the same master file by merging changes from separate .esp files.

Go to your Skyrim or Skyrim Special Edition installation folder.

Create a new folder called Merging.

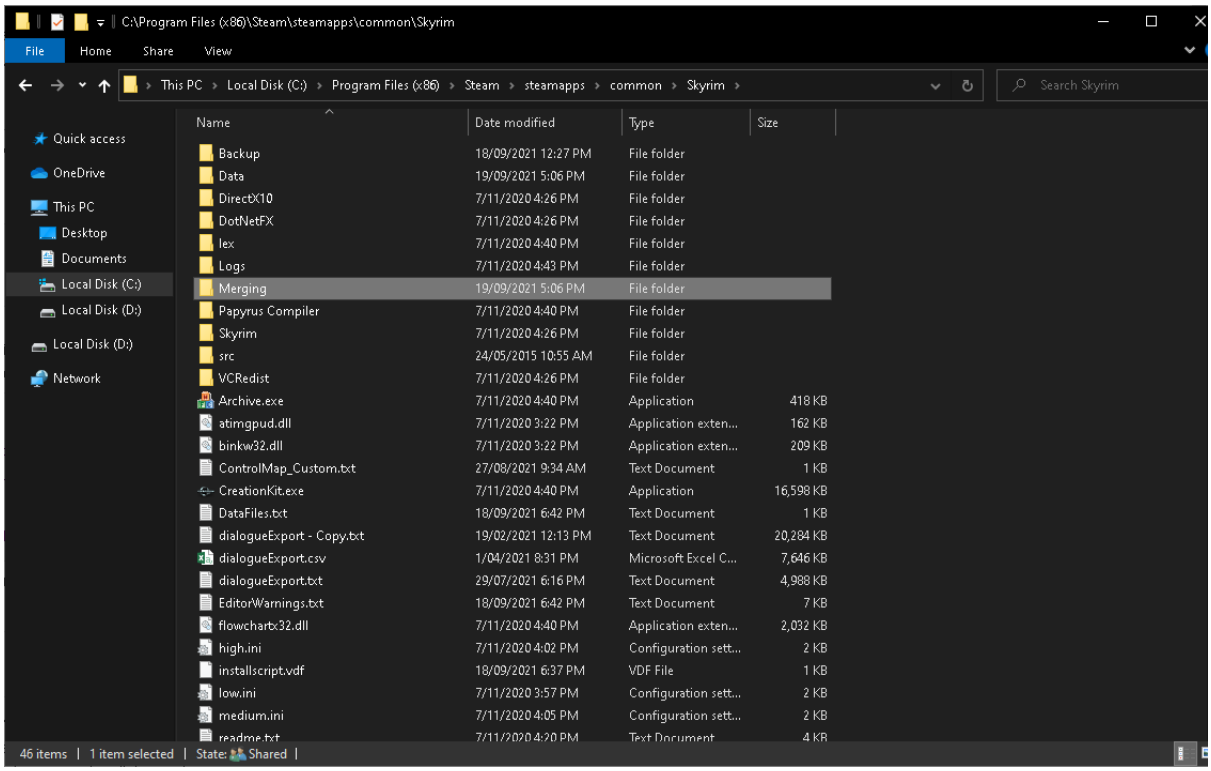


Figure 1455 - Merging folder.

In the Merging folder, create two folders: Data and VersionBackup.

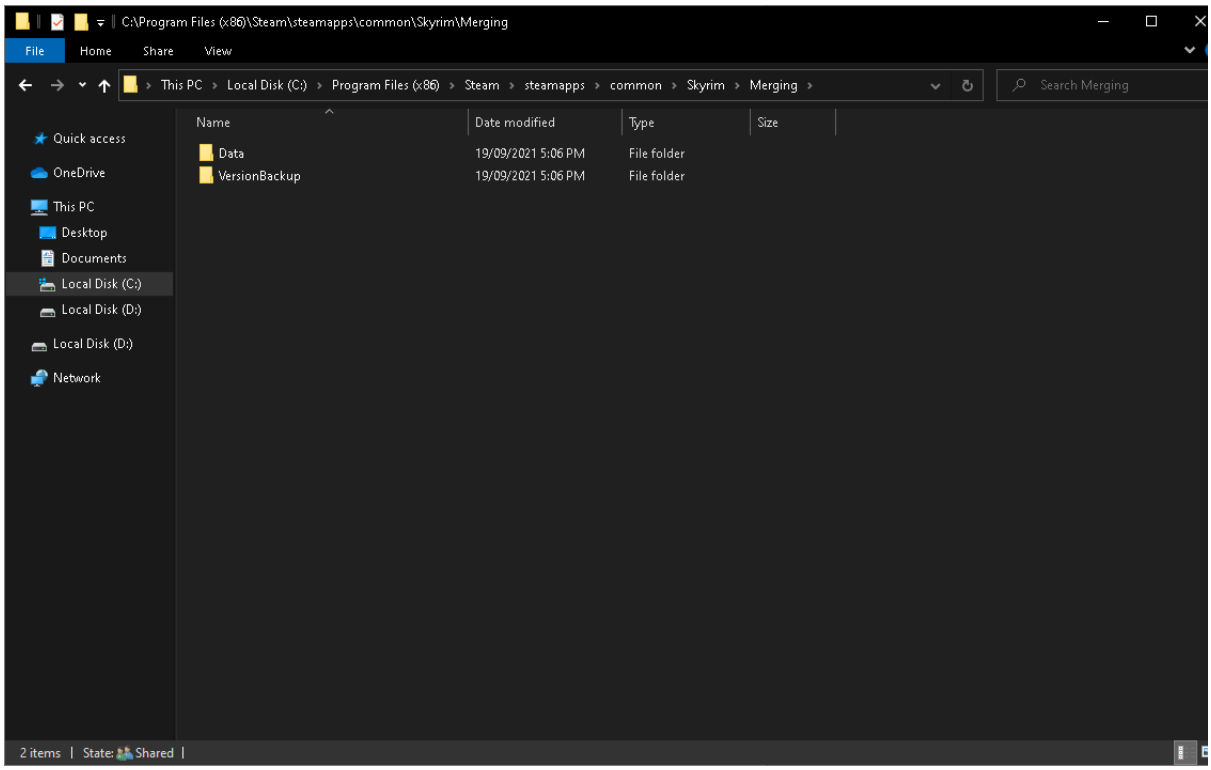


Figure 1456 - Data and VersionBackup folders.

Now go to your Skyrim\Data or Skyrim Special Edition\Data folder. In the Data folder, create a new folder called 'CheckInBackup'.

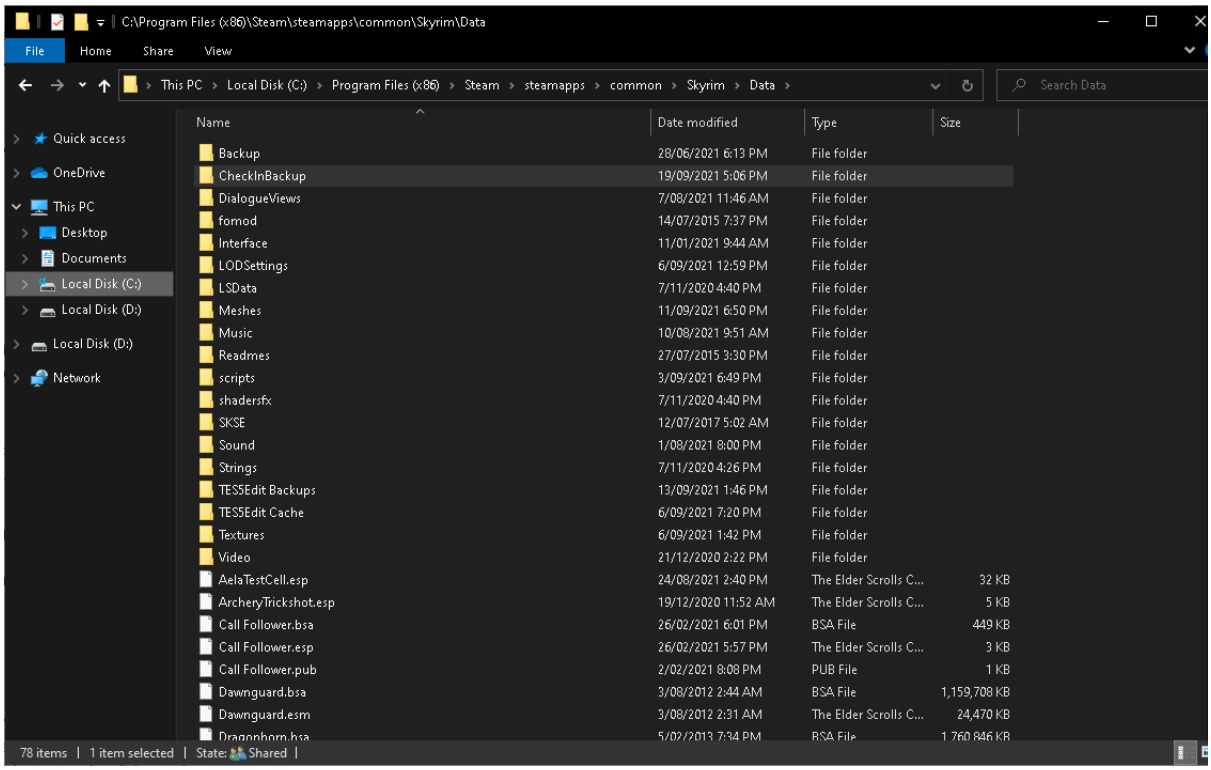


Figure 1457 - CheckInBackup folder.

Important: Your .esp/.esm file should have the 'esm' flag. For steps on adding the 'esm' flag to your mod, see ESM-ifying your mod in xEdit.

Copy your mod and any master files it relies on to the Skyrim\Merging\Data folder. For Skyrim Special Edition that's Skyrim Special Edition\Merging\Data folder.

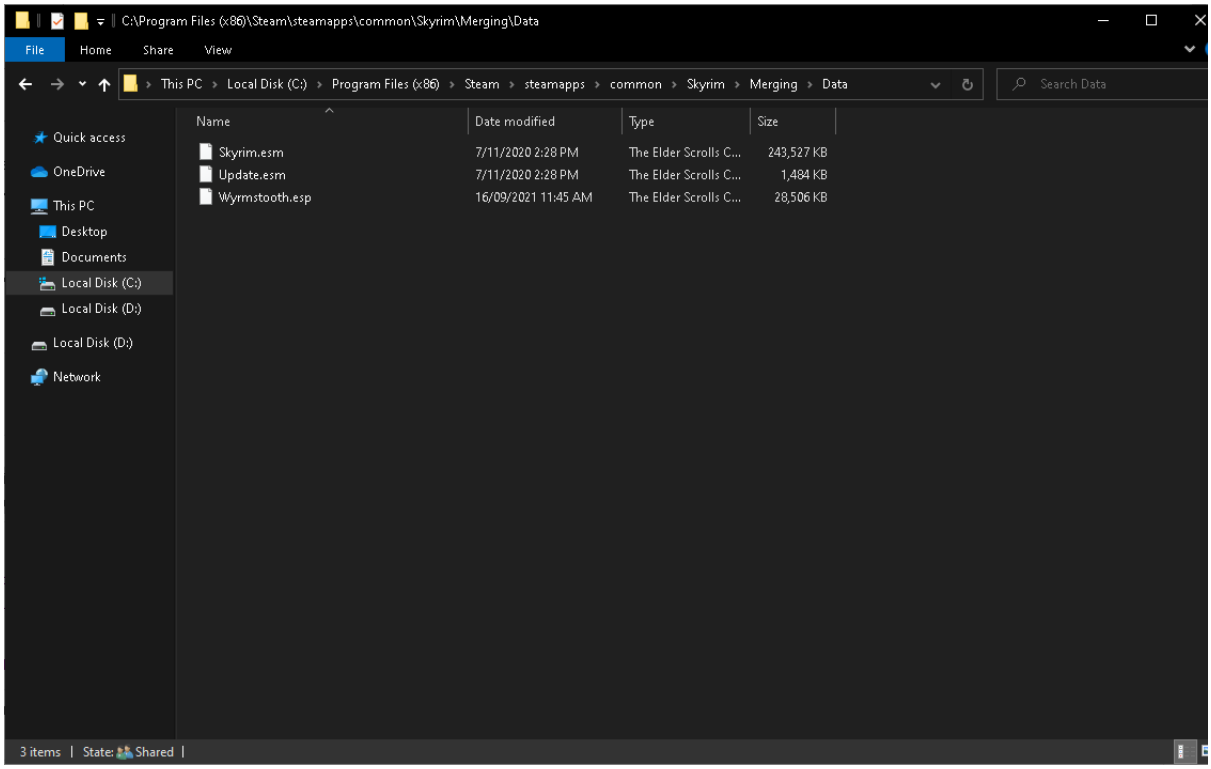


Figure 1458 - Skyrim.esm, Update.esm and our master file copied to the Skyrim\Merging\Data folder.

Open your SkyrimEditor.ini file.

Under the [General] tab, add the following lines:

```
bUseVersionControl=1
SNetwork Path=\\<YourPCName>\Merging\
SNewVersionBackupPath=\\<YourPCName>\Merging\VersionBackup\
SNetworkMasterPath=\\<YourPCName>\Merging\Data\
SLocalSavePath=Saves\
SLocalMasterPath=Data\
SLocalBackupPath=Data\CheckInBackup
```

Replace <YourPCName> with the name of your PC. To find your PC's name on Windows 10, run msinfo32 and check the 'Device Name' property.

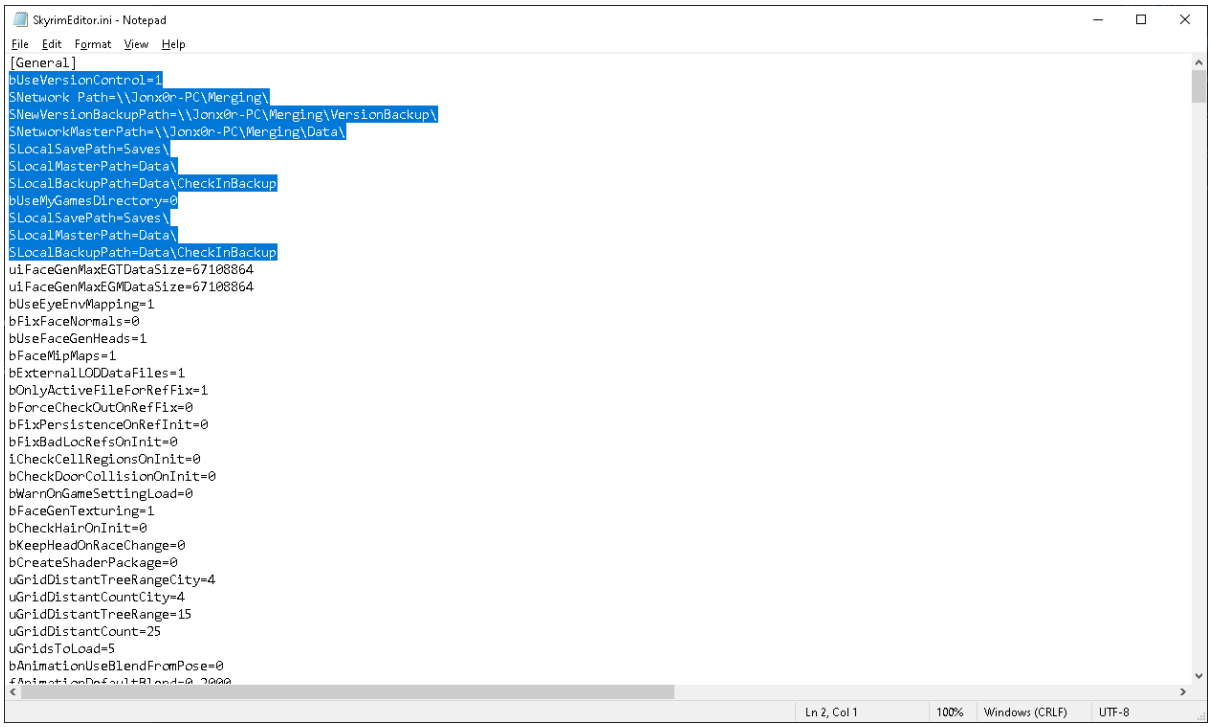


Figure 1459 - SkyrimEditor.ini with Version Control set up.

Open your SkyrimEditorPrefs.ini file and add the following lines:

```
[WhoCanMerge]
<YourUsername>=1
```

Replace <YourUsername> with the name of your Windows user account.

To get your user account name, run CMD.exe and look at the command prompt.

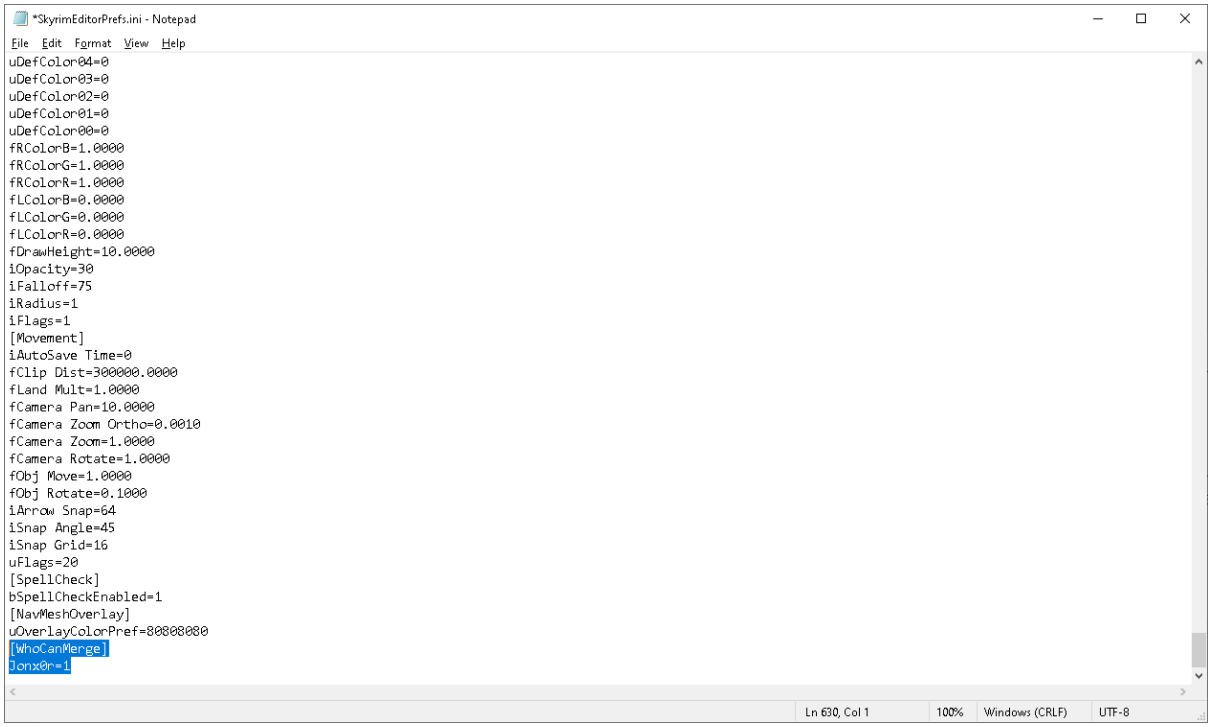


Figure 1460 - User account line added to SkyrimEditorPrefs.ini

Back in your Skyrim or Skyrim Special Edition install directory, right-click on the Merging folder and select Properties.

Go to the Sharing tab.

If no Sharing tab is visible, run 'services.msc' and make sure the 'Security Accounts Manager' and 'Server' services are Running and have their start-up set to Automatic.

If you need to enable any of these services, you will need to reboot your PC before continuing.

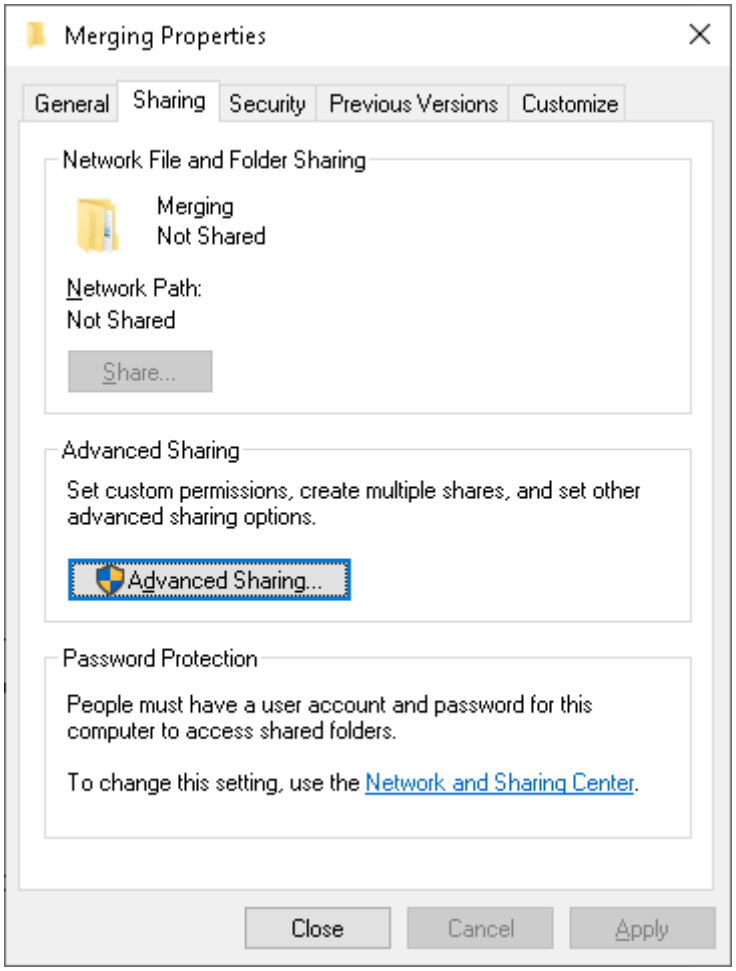


Figure 1461 - Sharing tab.

Click on the Advanced Sharing button.

Tick 'Share this folder'.

Make sure the 'Share name' is set to 'Merging'.

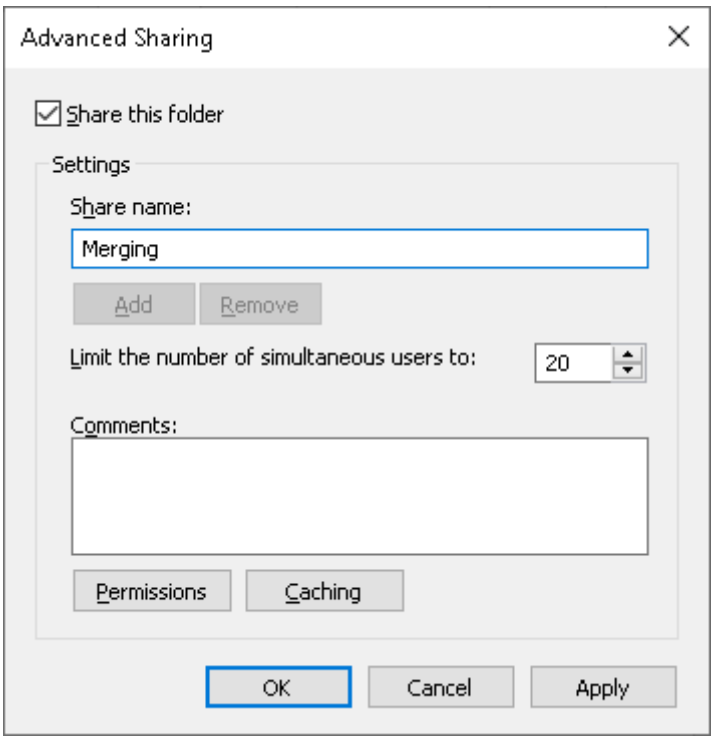


Figure 1462 - Sharing the Merging folder.

Click Permissions.

Click Add.

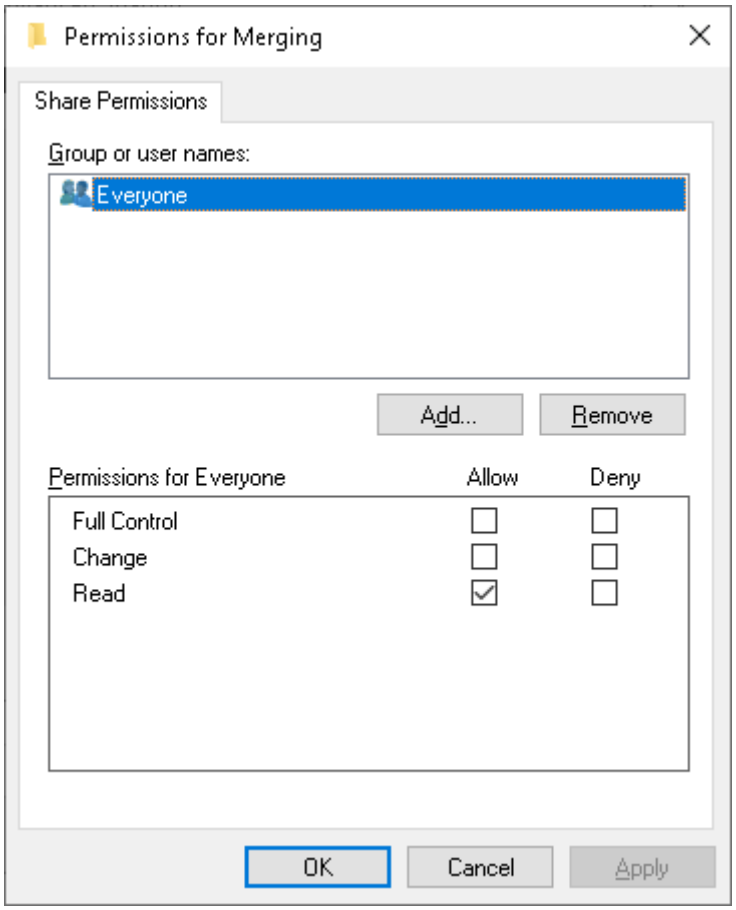


Figure 1463 - Share permissions.

Enter in your user name and click Check Names. The computer name should be automatically added if the user account you typed was found.

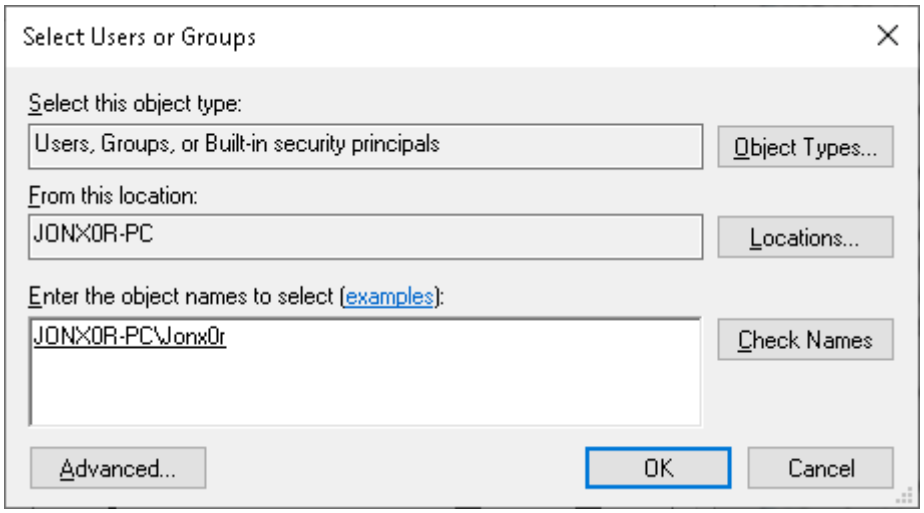


Figure 1464 - Adding your user account.

Click OK.

With your user account selected, tick Change in the Permissions list below.

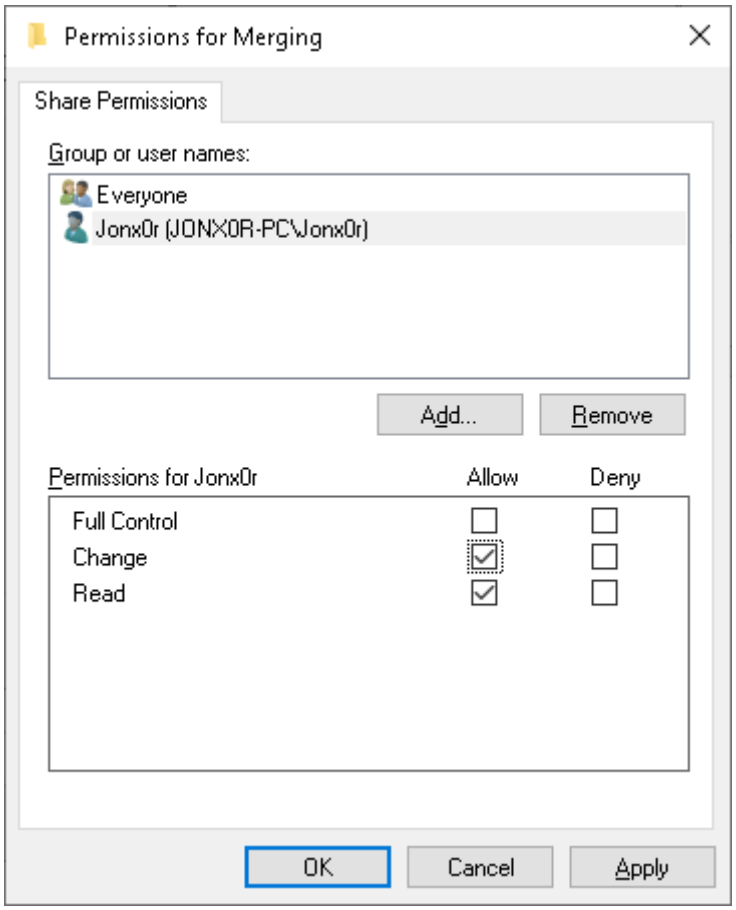


Figure 1465 - Giving yourself Change permissions.

Click OK to close out of Share Permissions.

Click OK to close out of Advanced Sharing.

Click OK to close out of Merging Properties.

Now we need to share the CheckInBackup folder.

Go to Data, right-click on CheckInBackup and select Properties.

Go to the Sharing tab.

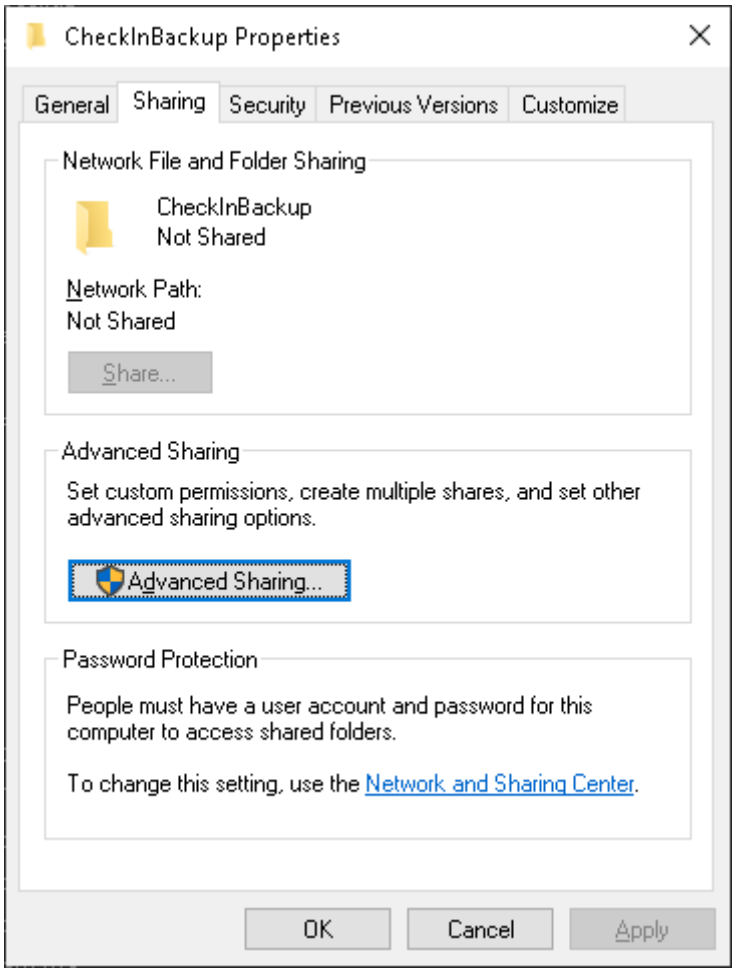


Figure 1466 - Sharing tab for the CheckInBackup folder.

Click on Advanced Sharing.

Tick 'Share this folder'.

Make sure the 'Share name' is set to 'CheckInBackup'.

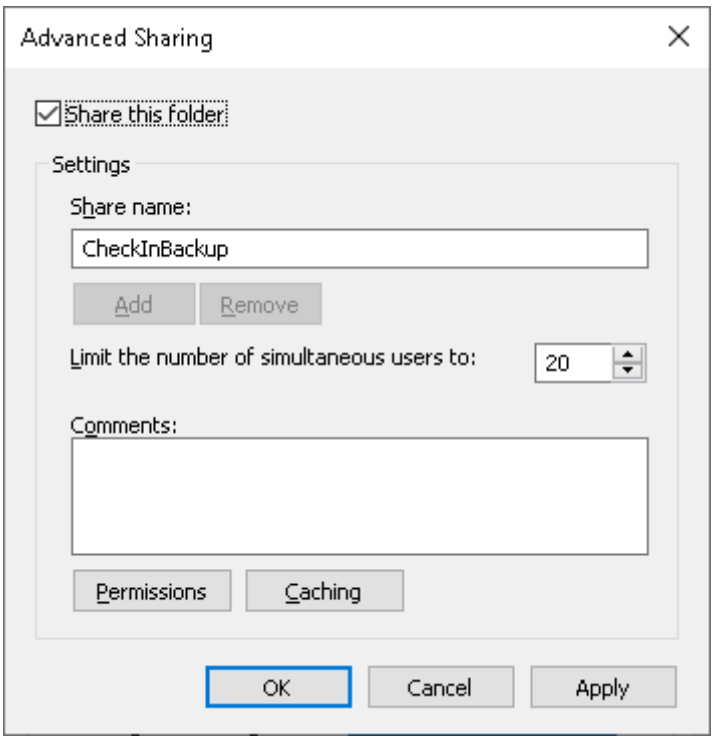


Figure 1467 - Creating the CheckInBackup share.

Click Permissions.

Click Add.

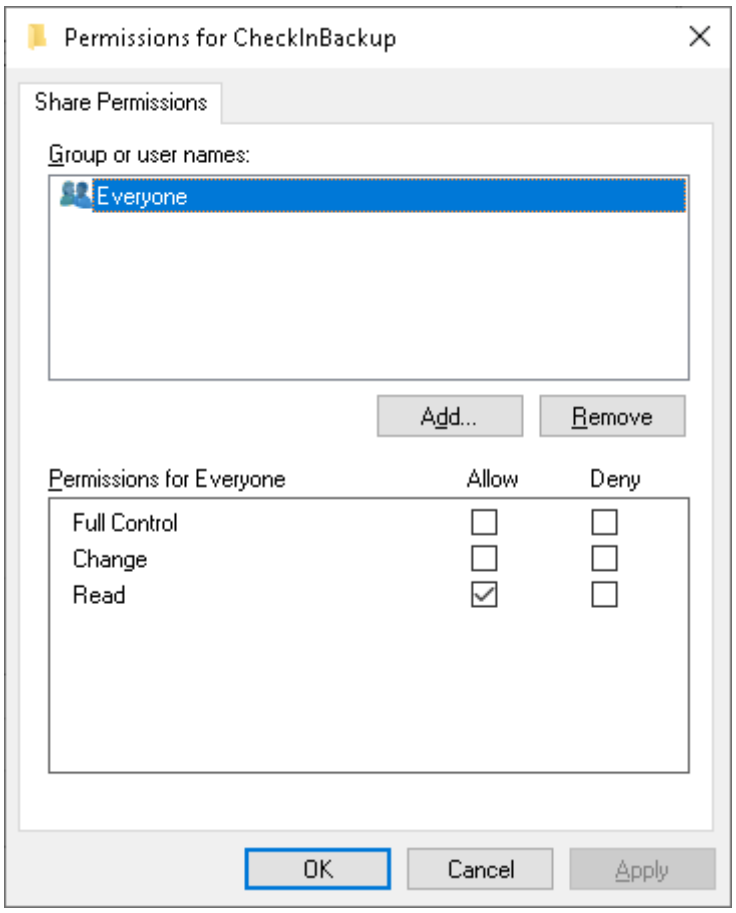


Figure 1468 - Adding share permissions.

Enter in your user name and click Check Names. The computer name should be automatically added if the user account you typed was found.

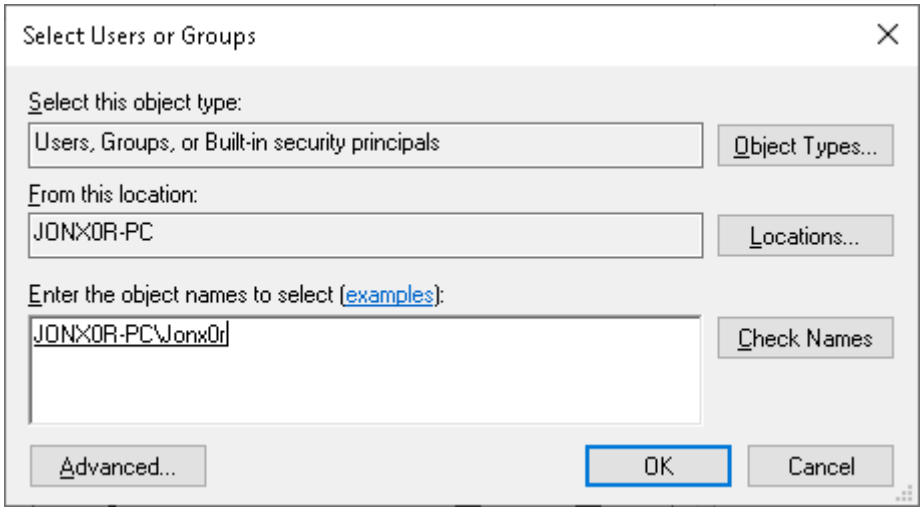


Figure 1469 - Adding your user account.

Click OK.

With your user account selected, tick 'Change' in the Permissions list.

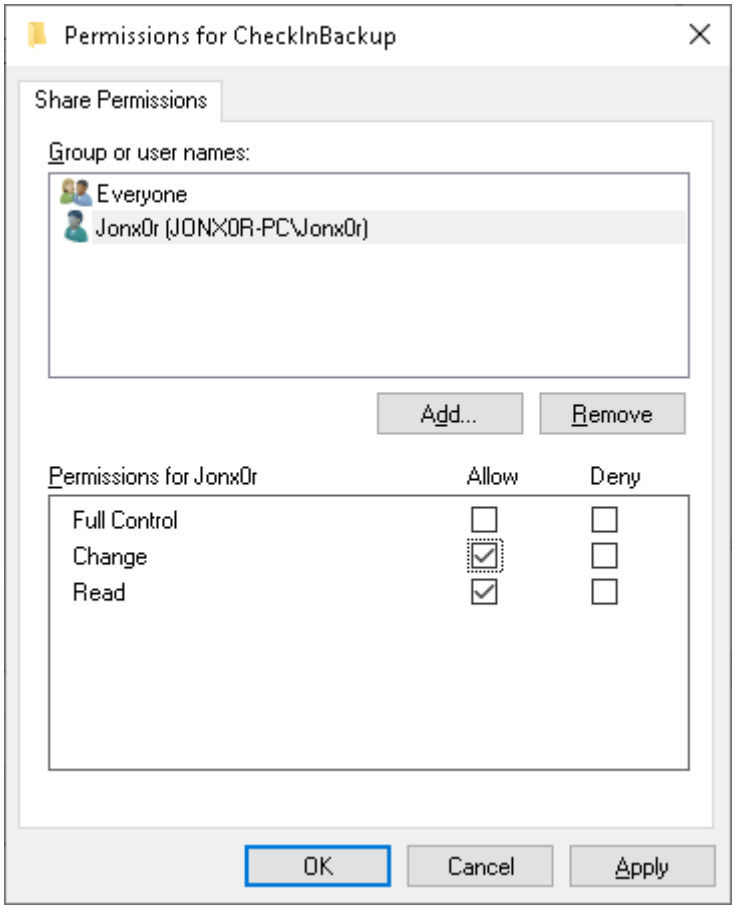


Figure 1470 - Giving yourself Change permission on the share.

Click OK to close out of Share Permissions.

Click OK to close out of Advanced Sharing.

Click OK to close out of CheckInBackup Properties.

If you can't connect to your shares by running \\<YourComputerName>\Merging and \\<YourComputerName>\CheckInBackup, read on.

Click on 'Turn Windows features on or off'.

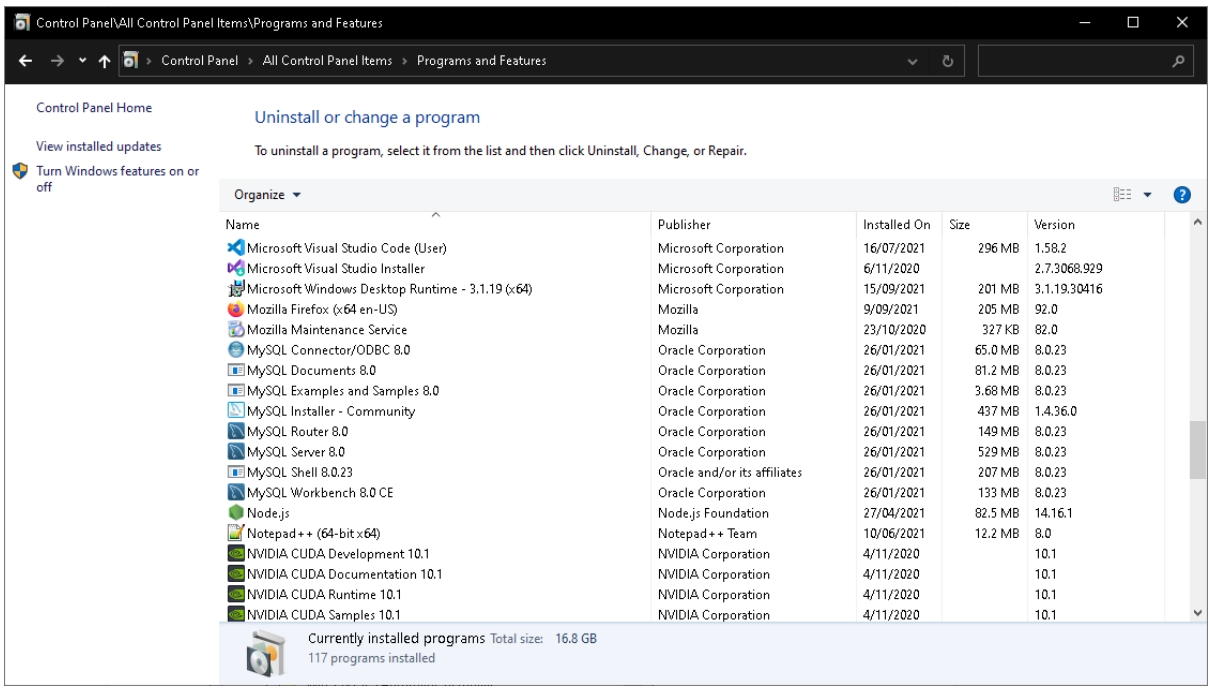


Figure 1471 - Control panel.

Browse down to 'SMB 1.0/CIFS File Sharing Support' and make sure 'SMB 1.0/CIFS Client' is ticked. If it isn't, tick it, click OK and reboot your PC.

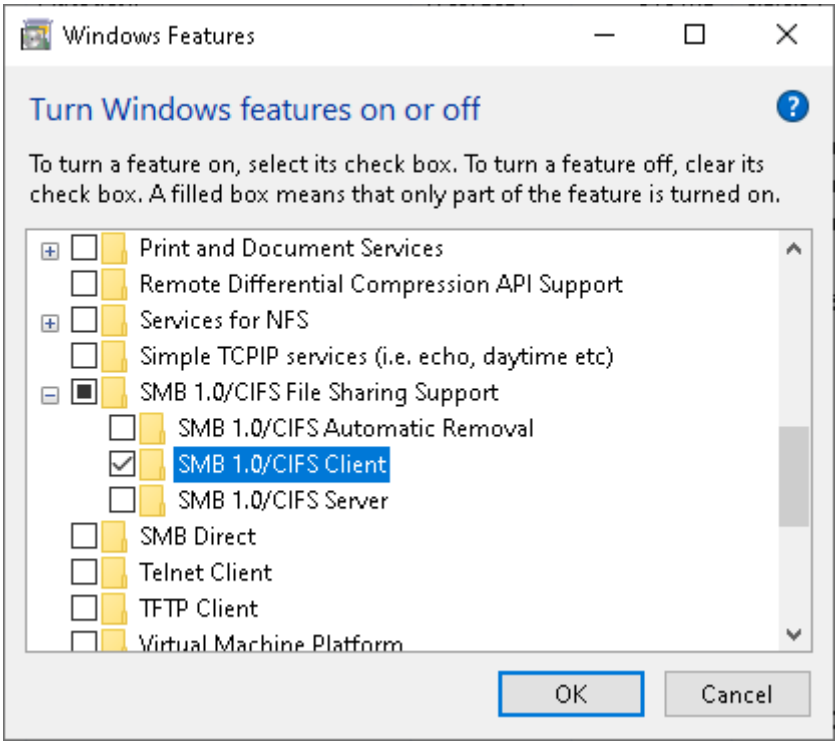


Figure 1472 - SMB 1.0/CIFS Client.

Go to Control Panel > Network and Sharing Center.

Click on 'Change advanced sharing settings'.

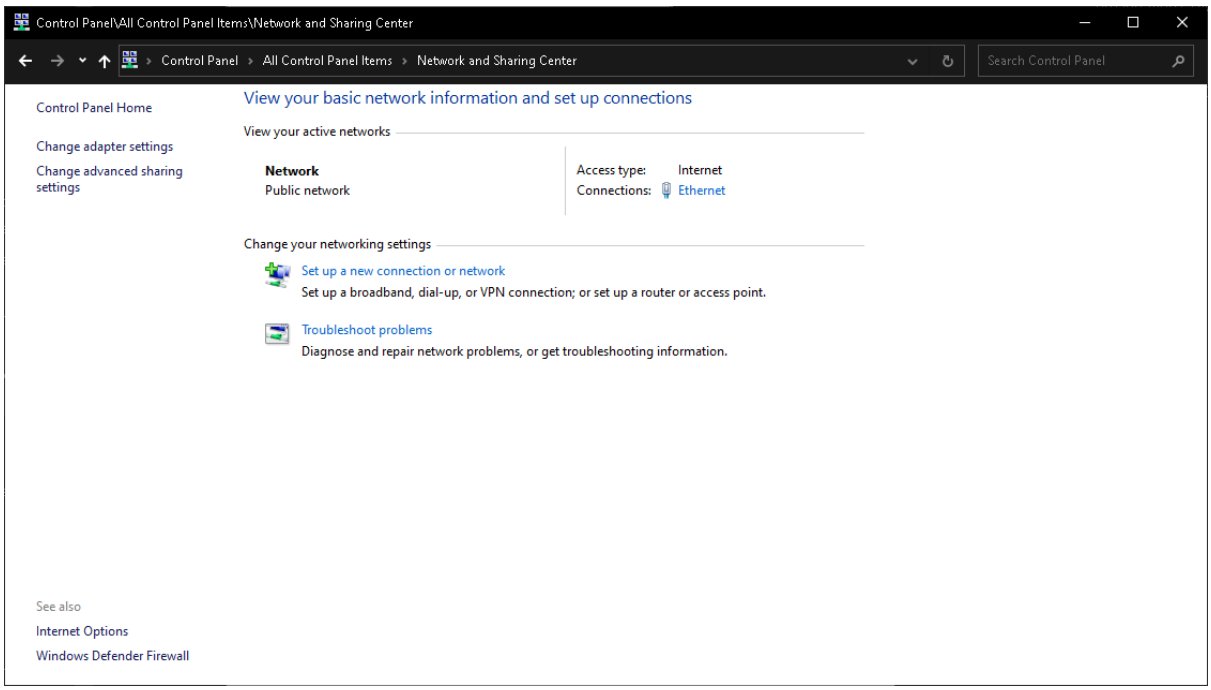


Figure 1473 - Network and Sharing Center.

Unde Private, tick 'Turn on file and printer sharing'.

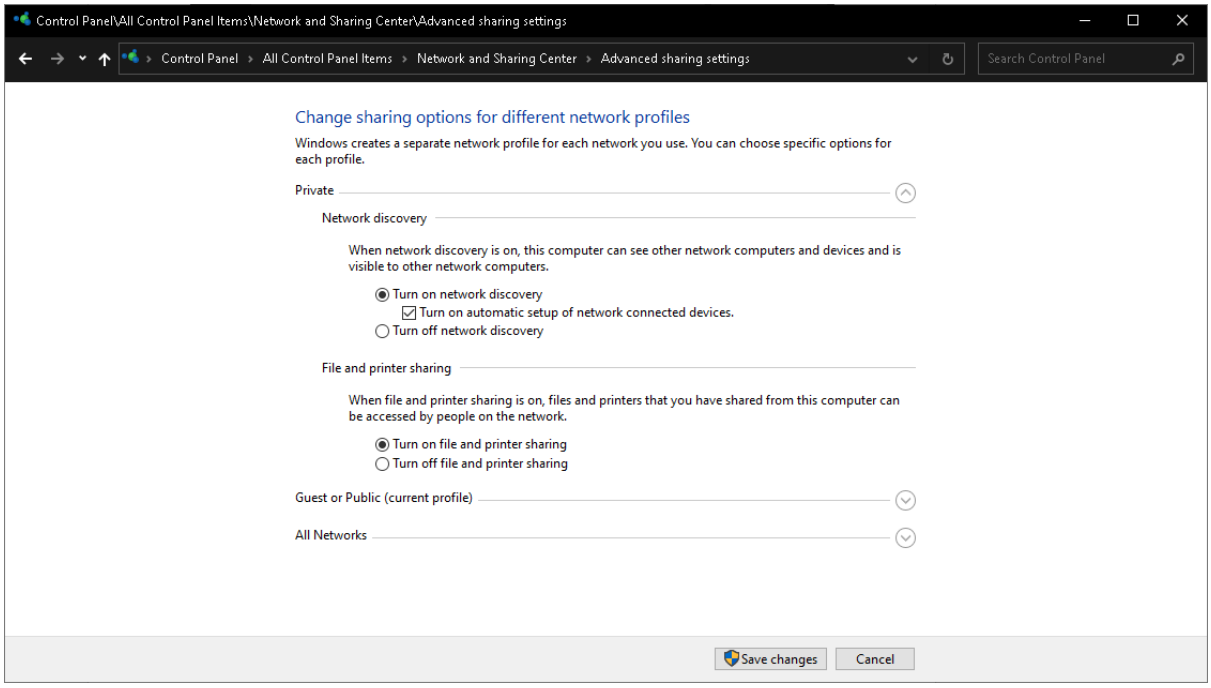


Figure 1474 - Enabling file and printer sharing.

Click 'Save changes'.

Run services.msc and make sure the following services are running: Function Discovery Provider Host, Function Discovery Resource Publication, SSDP Discovery, UPnP Device Host.

Also make sure 'File and Printer Sharing for Microsoft Networks' is enabled on your network adapter.

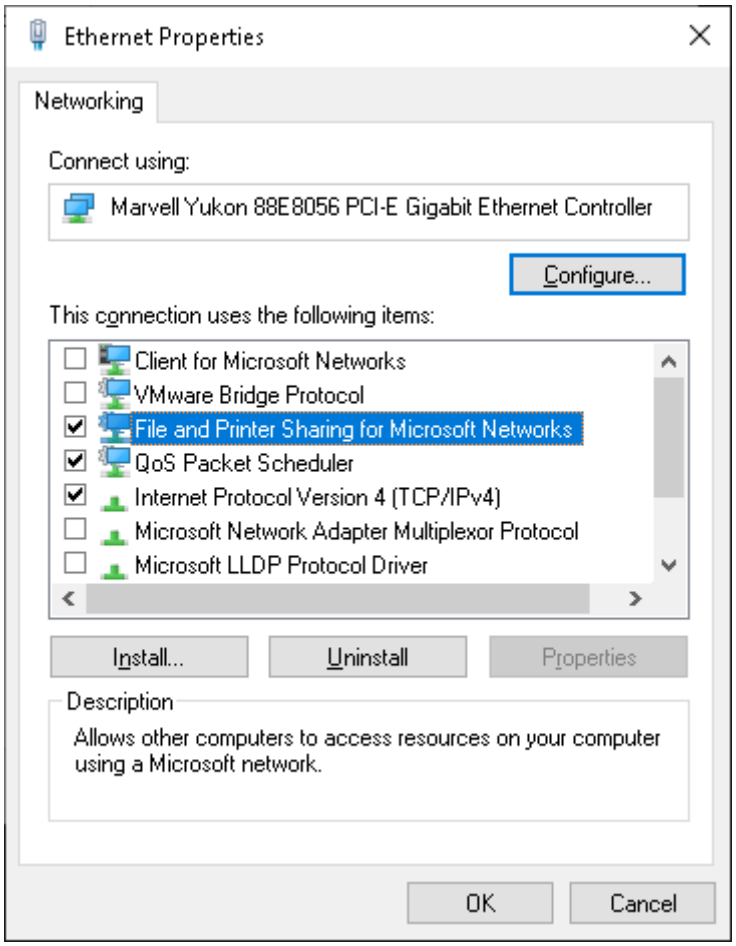


Figure 1475 - Enabling File and Printer Sharing for Microsoft Networks.

You should now be able to connect to your share by running \\<YourComputerName>\Merging and \\<YourComputerName>\CheckInBackup.

e.g.: \\jonx0r-pc\Merging and \\jonx0r-pc\CheckInBackup.

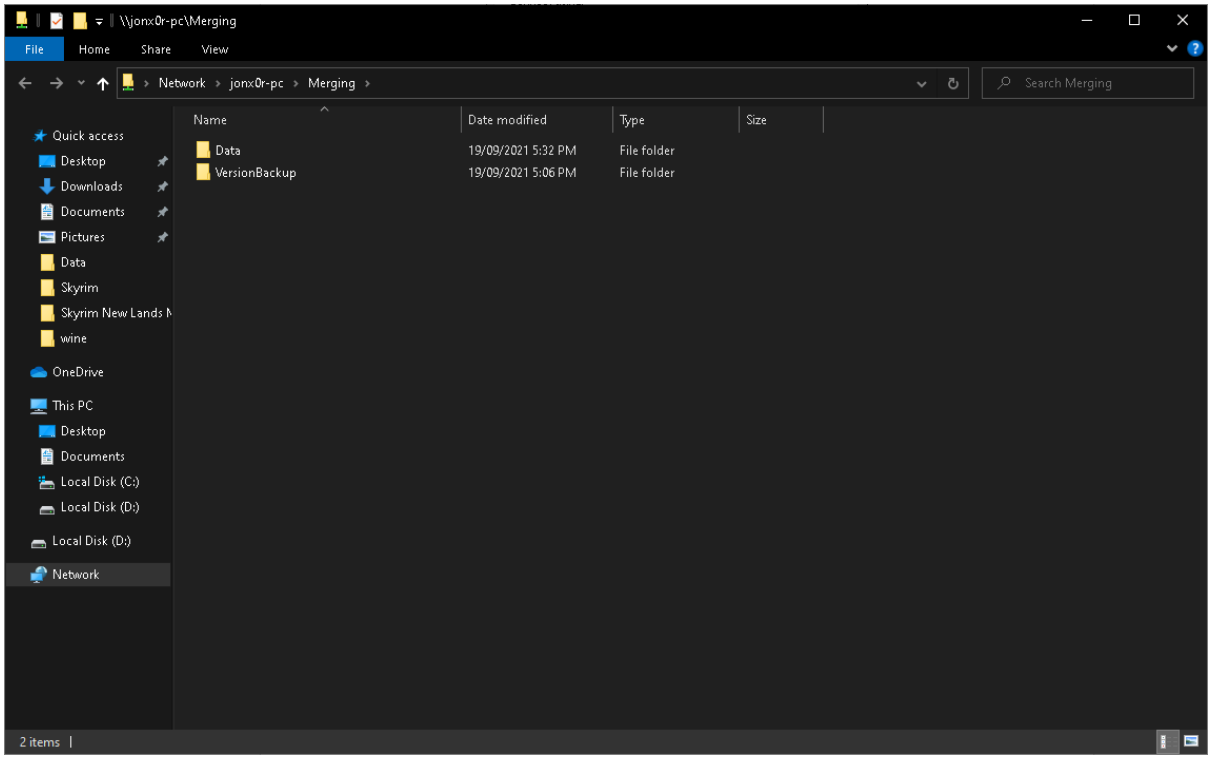


Figure 1476 - Connecting to a network share.

By default, you won't be able to merge changes made to forms in Skyrim.esm. To be able to do this, we'll need to make a small edit to Skyrim.esm with a hex editor.

We'll need a copy of frehed which you can download from Sourceforge.

Important: Before doing anything, make a backup of your Skyrim.esm file from the Skyrim\Data or Skyrim Special Edition\Data folder.

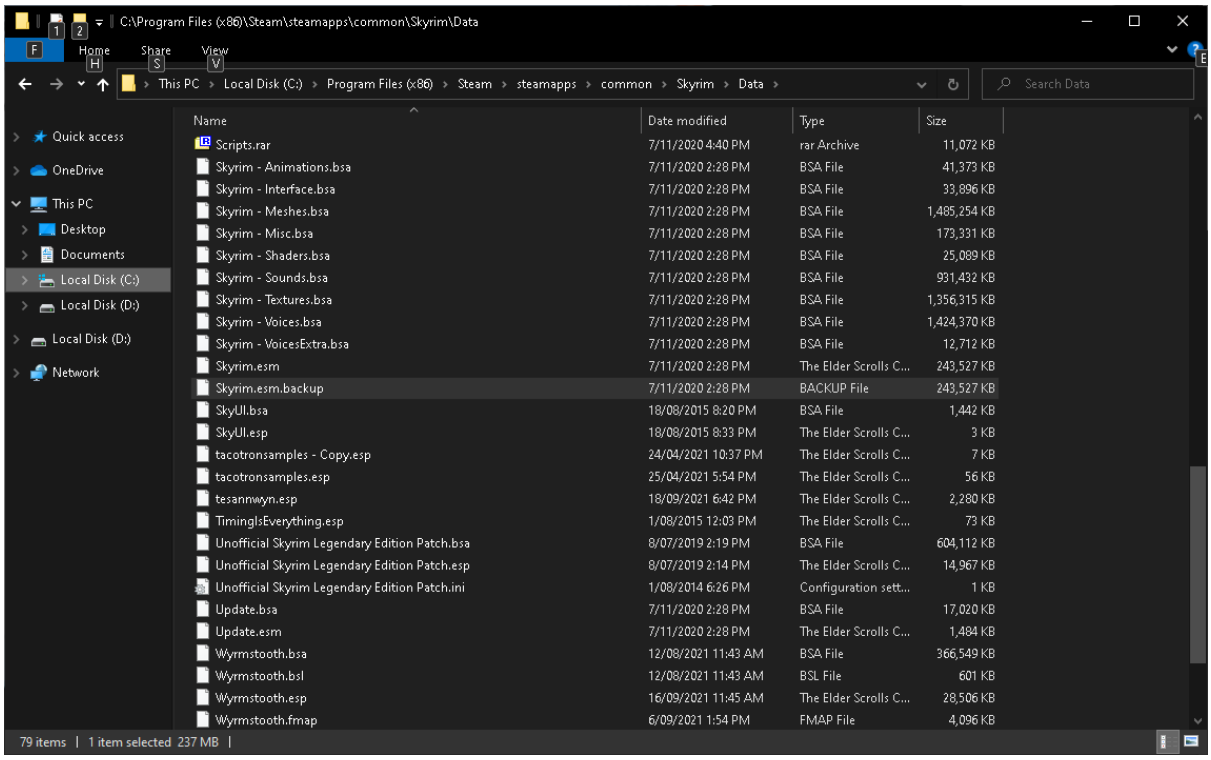


Figure 1477 - Backup of Skyrim.esm.

Open frehed.



Figure 1478 - Time for some fun with frehed.

Go to File > Open.

Select the Skyrim.esm file in your Skyrim\Data or Skyrim Special Edition\Data folder and click Open.

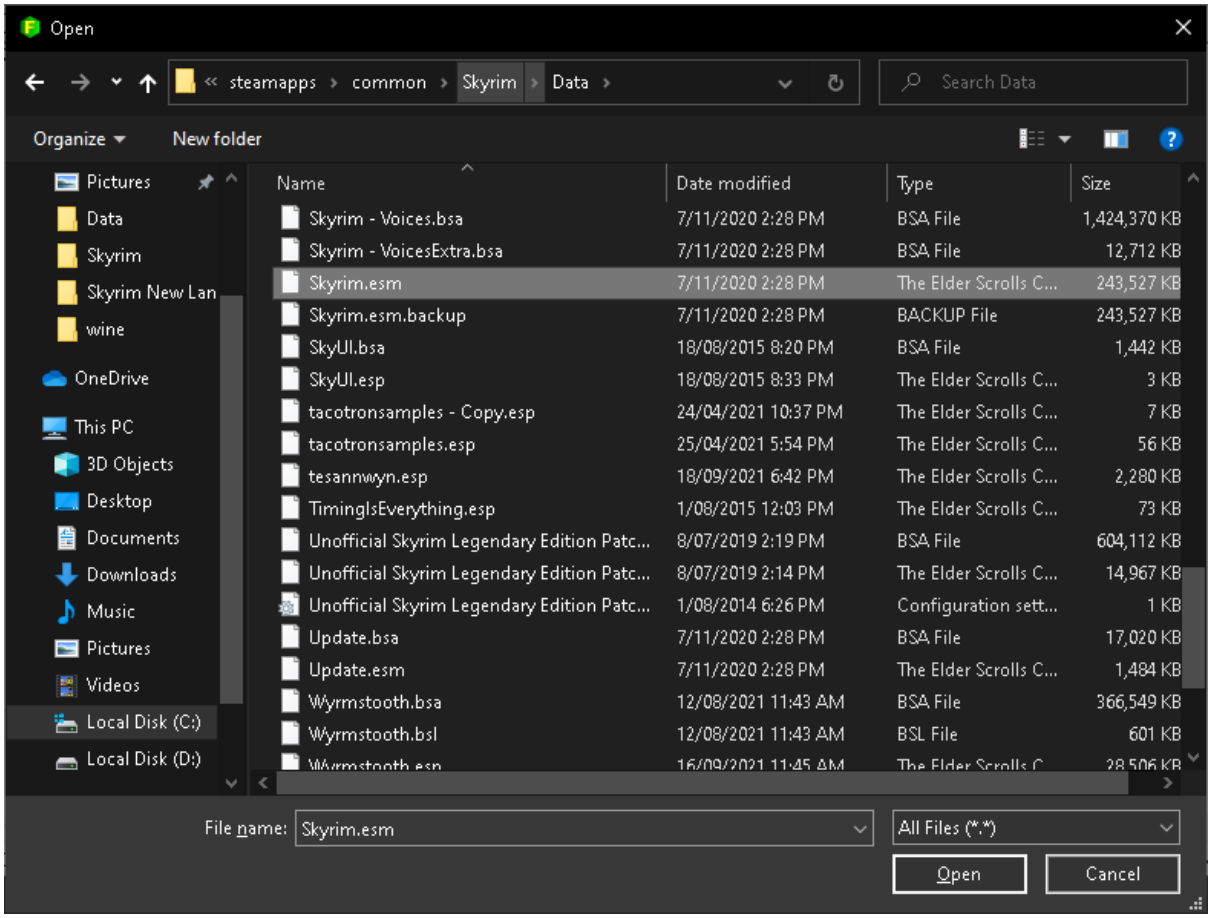


Figure 1479 - Opening the Skyrim.esm file in frehed.

By default, the value of the ninth byte in Skyrim.esm is 81.

To find this, go to the first row 0000000. Starting from the top-left, go right and look for '81'. It should be the ninth value pair from the top-left.

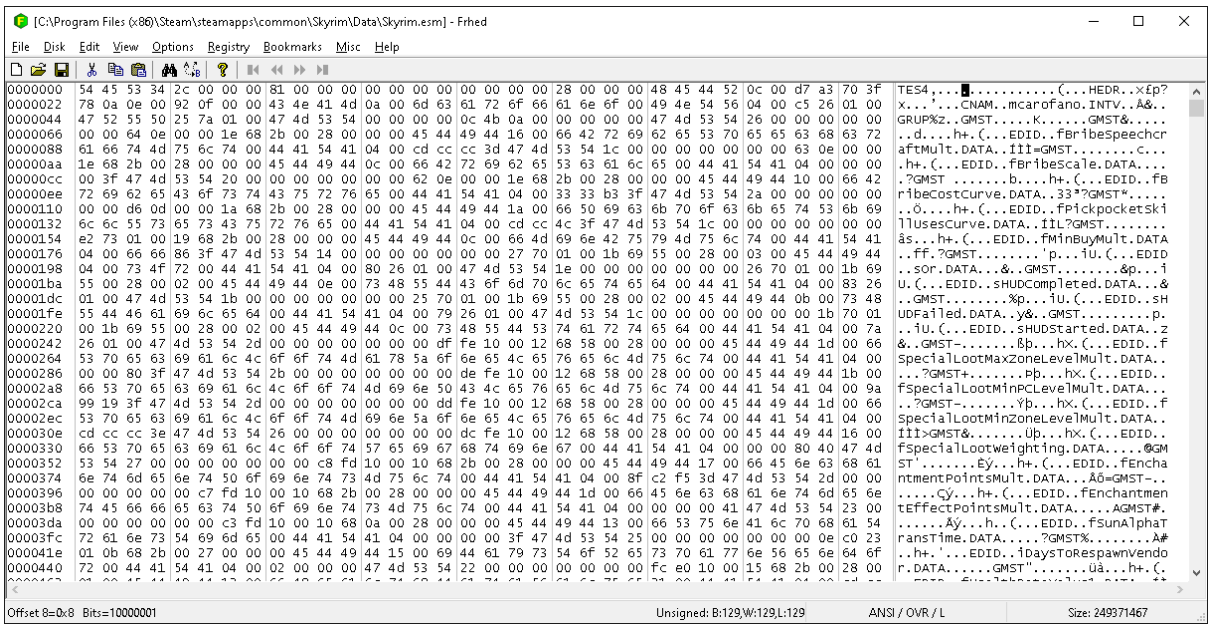


Figure 1480 - Default value of the ninth byte.

Change 81 to 01.

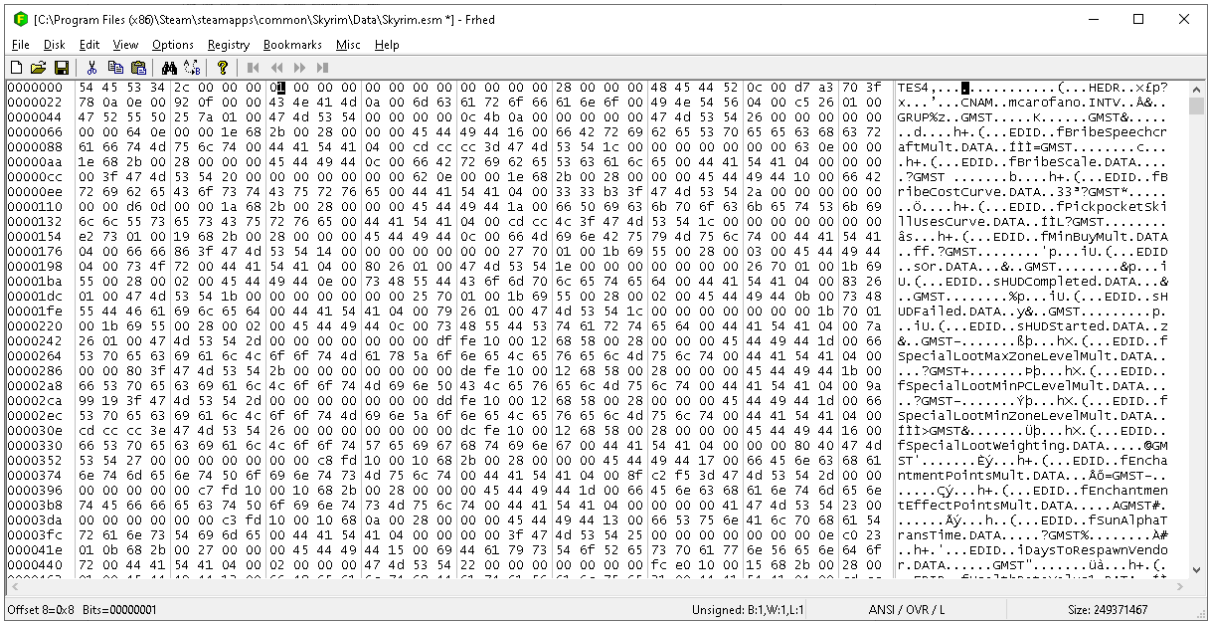


Figure 1481 - Changing the value of the ninth byte.

Go to File > Save.

Close frehed.

Important: If you want to play the game, rename the current Skyrim.esm file to Skyrim.esm.hexedited, and rename the Skyrim.esm.backup file back to Skyrim.esm.

Copy the Skyrim.esm file from your Skyrim\Data or Skyrim Special Edition\Data folder to your Skyrim\Merging\Data or Skyrim Special Edition\Merging\Data folder.

Overwrite the Skyrim.esm file we copied there previously.

Launch the Creation Kit.

Click OK to the Perforce Interface warning.

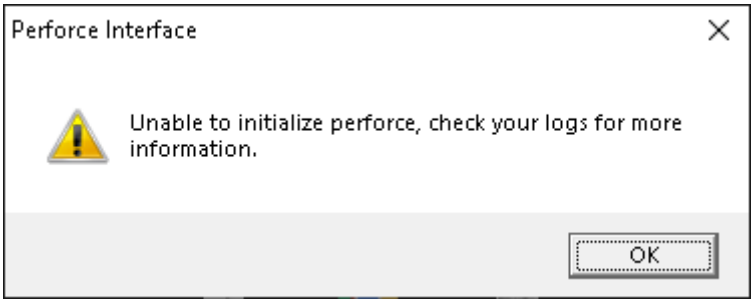


Figure 1482 - Perforce Interface warning.

Things will look a bit different.

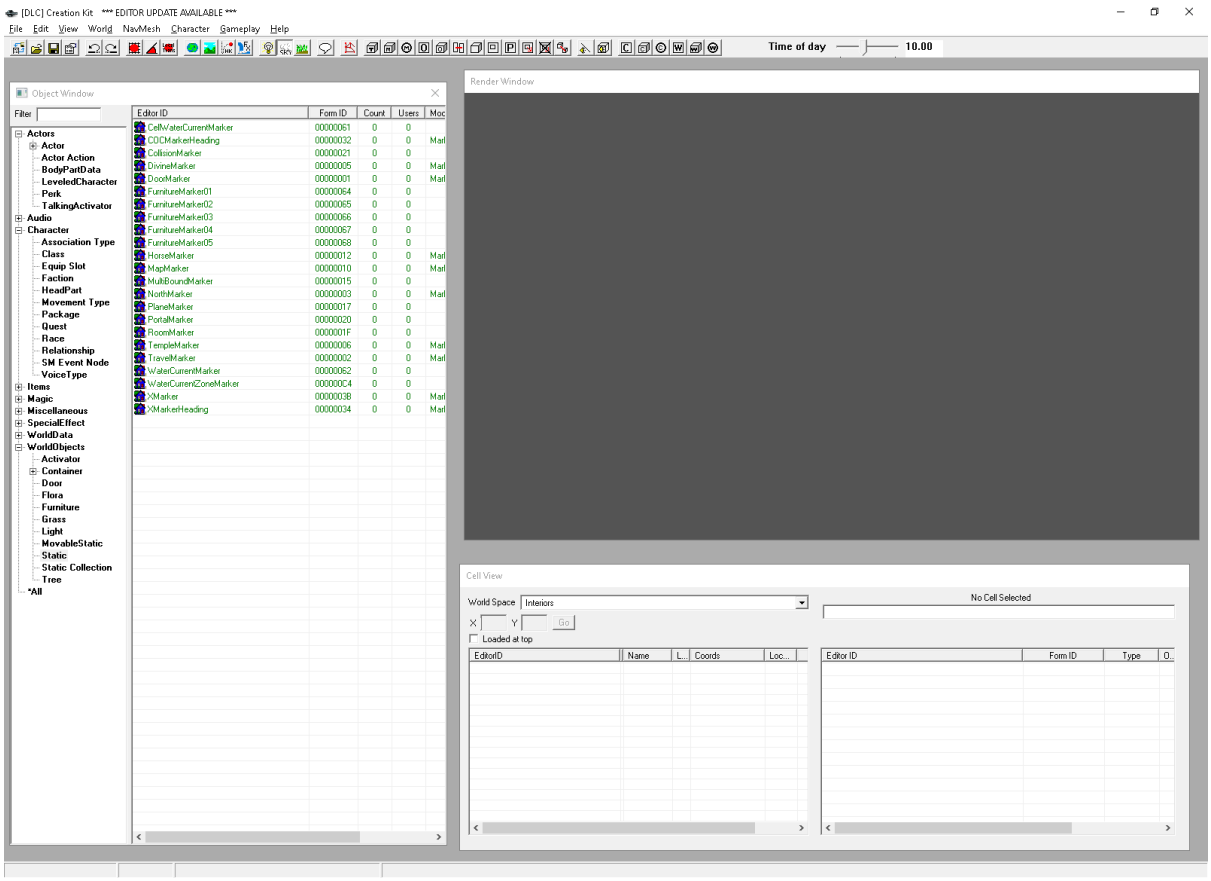


Figure 1483 - Creation Kit with Version Control enabled.

Go to File > Data.

Tick your master file in the Plugin/Master Files list.

Click OK to begin loading.

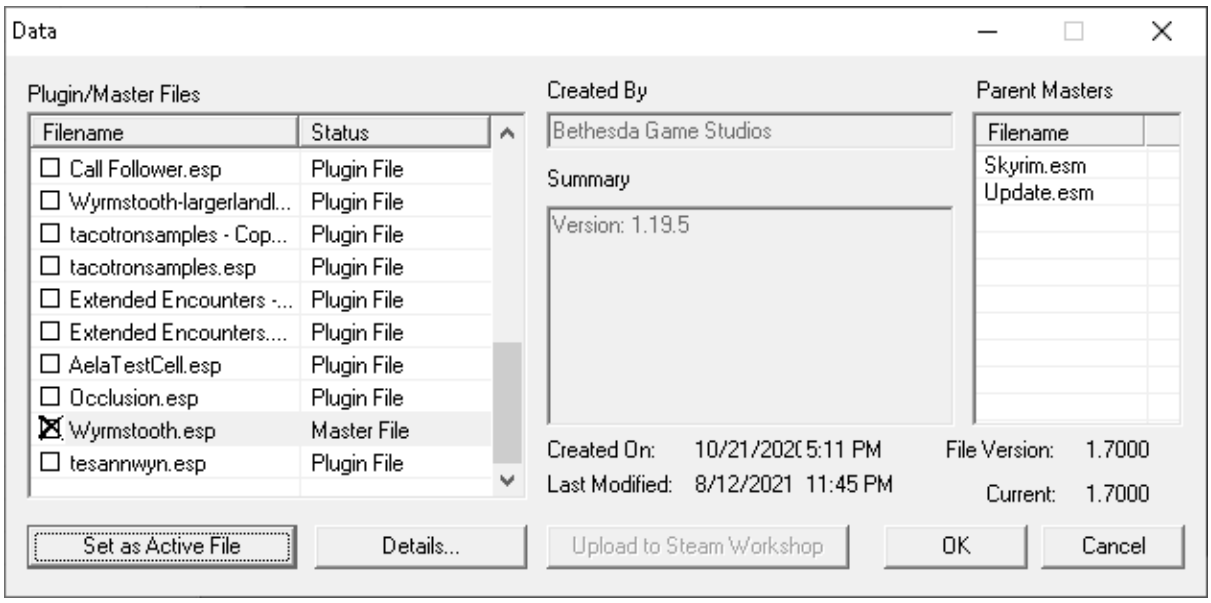


Figure 1484 - Loading your master file.

Click 'Yes To All' to all the warning pop-ups.

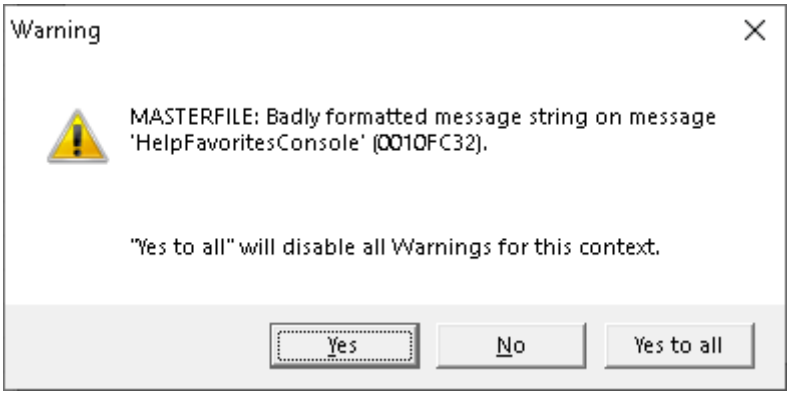


Figure 1485 - Ignore the warnings.

Go back to File > Data.

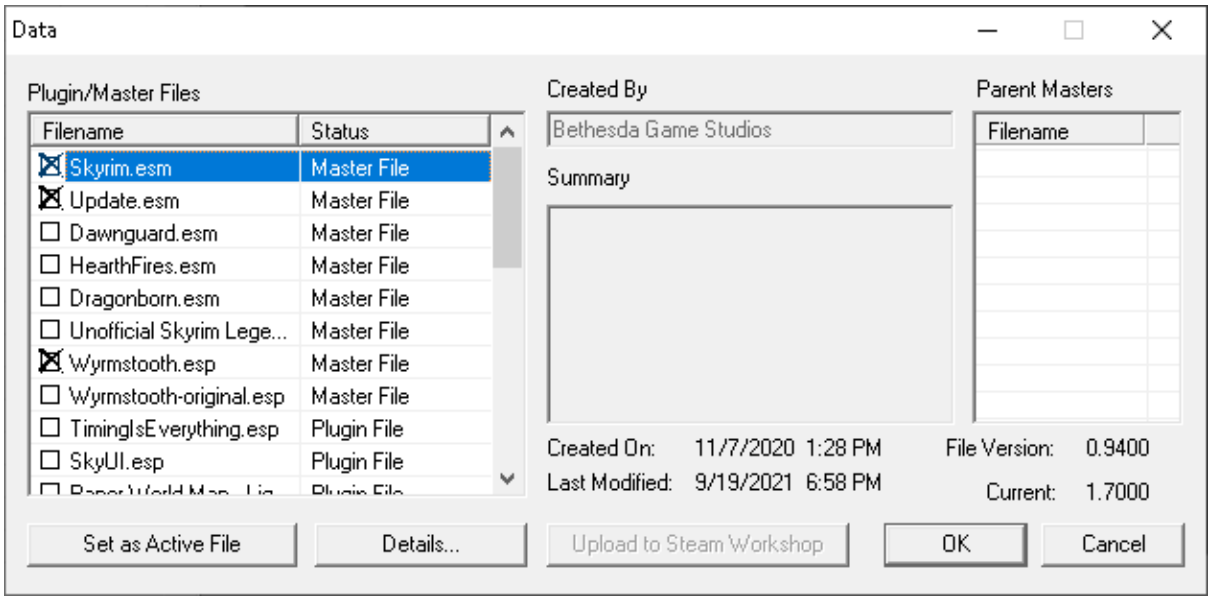


Figure 1486 - Plugin list.

Highlight Skyrim.esm and click on the Details button.

Click Yes to the 'Empty List?' pop-up.



Figure 1487 - 'Empty List?' popup.

You should see a blank File Details list.

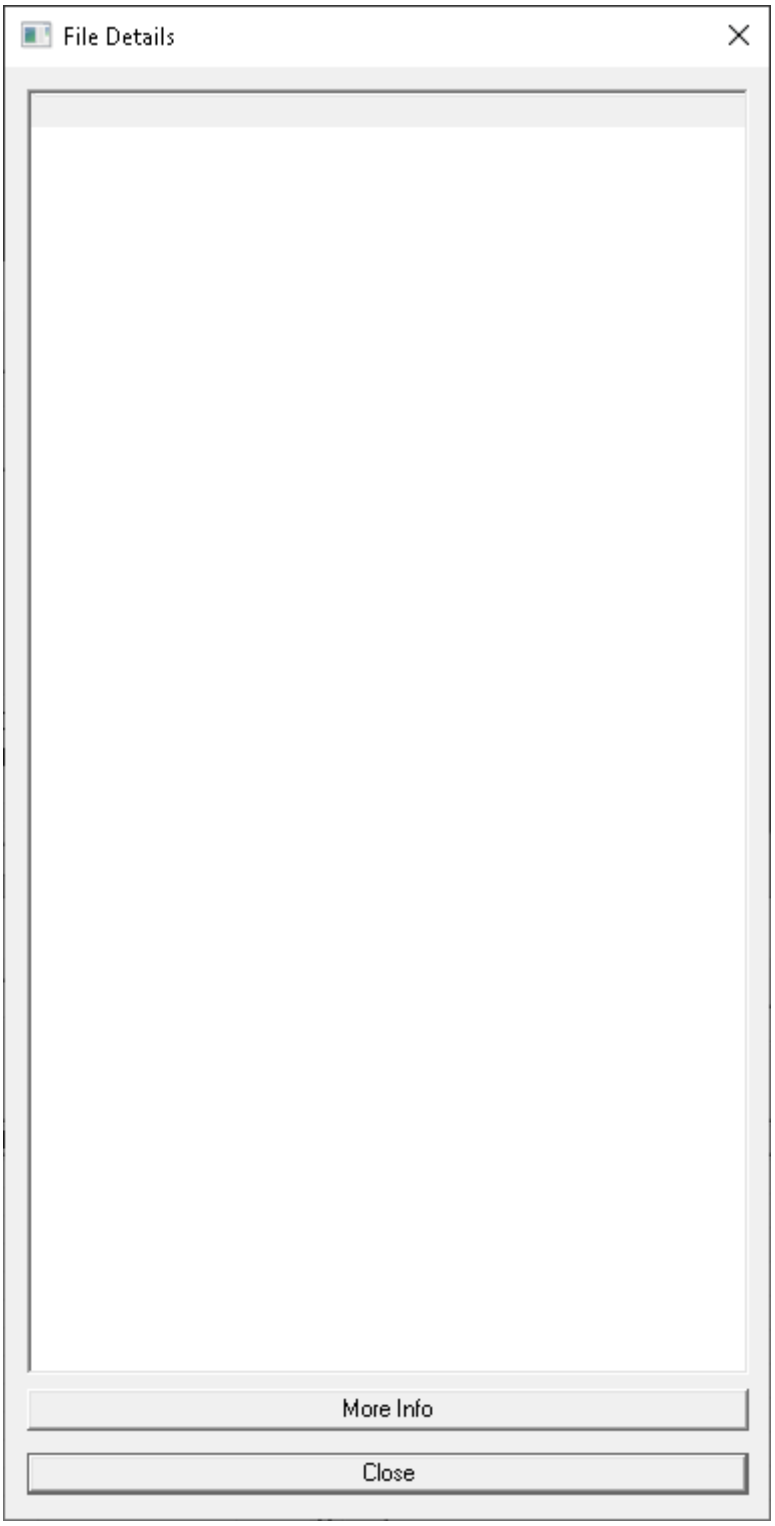


Figure 1488 - Blank File Details.

Press CTRL + Shift + B.

Click Yes to the 'Confirm Bit Array File Update' pop-up.

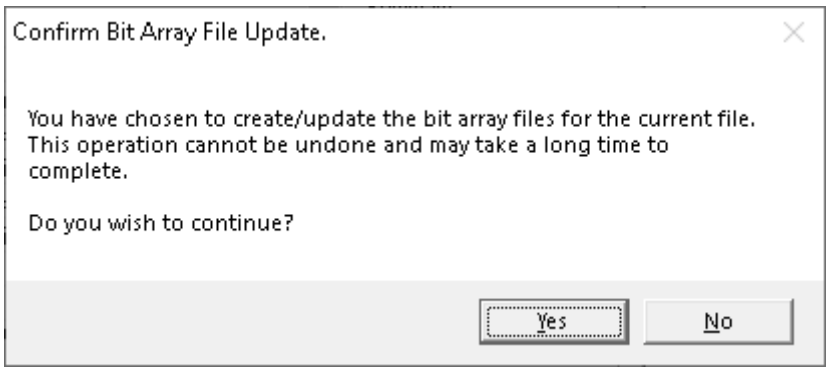


Figure 1489 - Confirm Bit Array File Update pop-up.

Now click Close to the File Details list.

You should now see the following files in your Skyrim\Merging\Data or Skyrim Special Edition\Merging\Data folder: Skyrim.fid, Skyrim.fud, Skyrim.fvd.

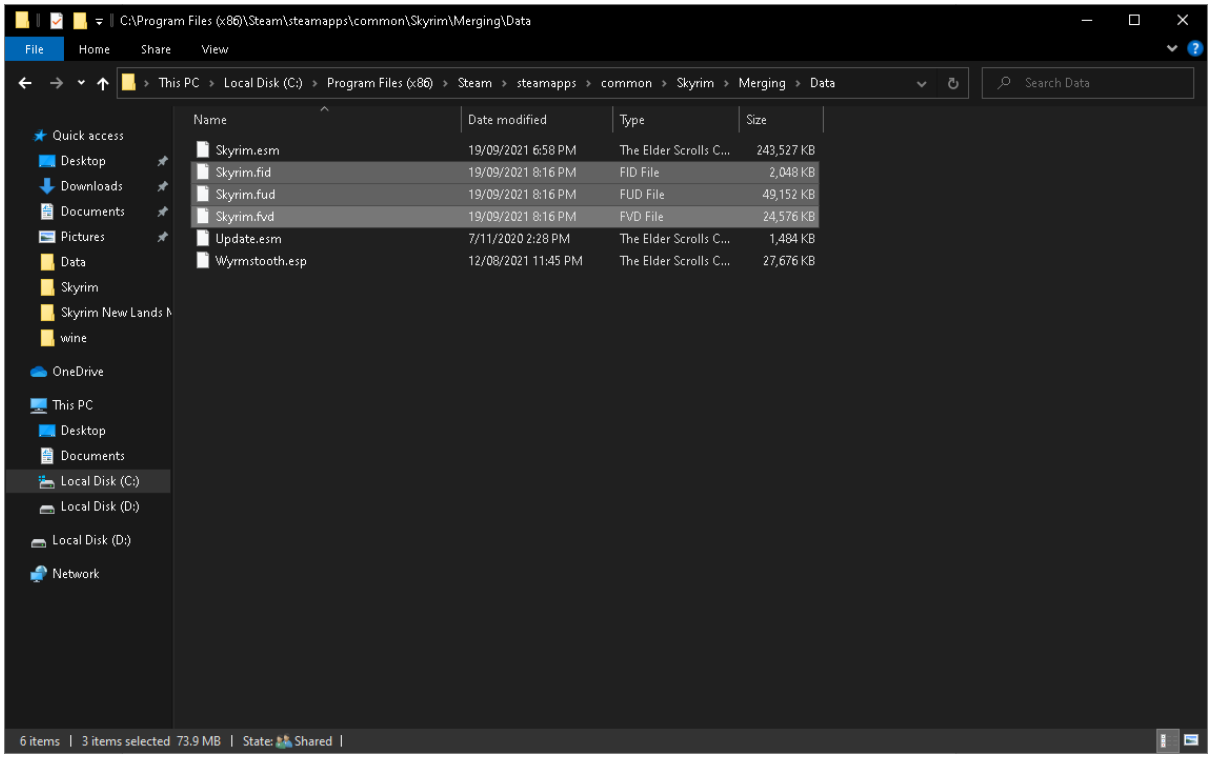


Figure 1490 - Skyrim.fid, Skyrim.fud, Skyrim.fvd files created.

Next, with Data still open, highlight your mod and repeat these steps to create a set of .fid, .fud, and .fvd files for your mod.

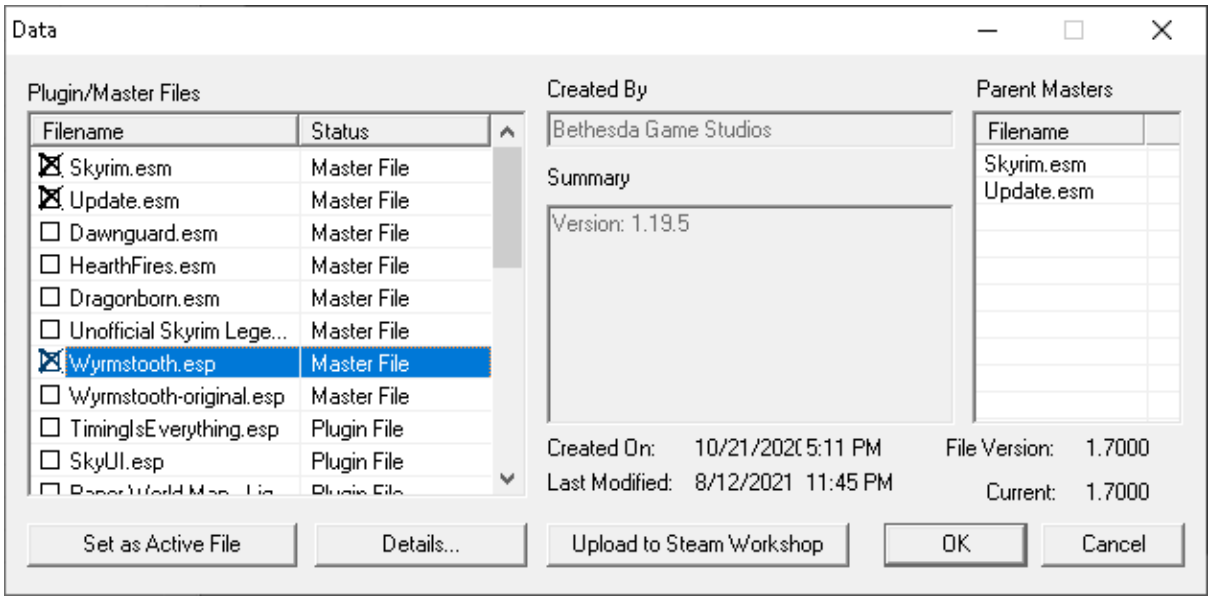


Figure 1491 - Generating the .fid, .fud and .fvd files for our mod.

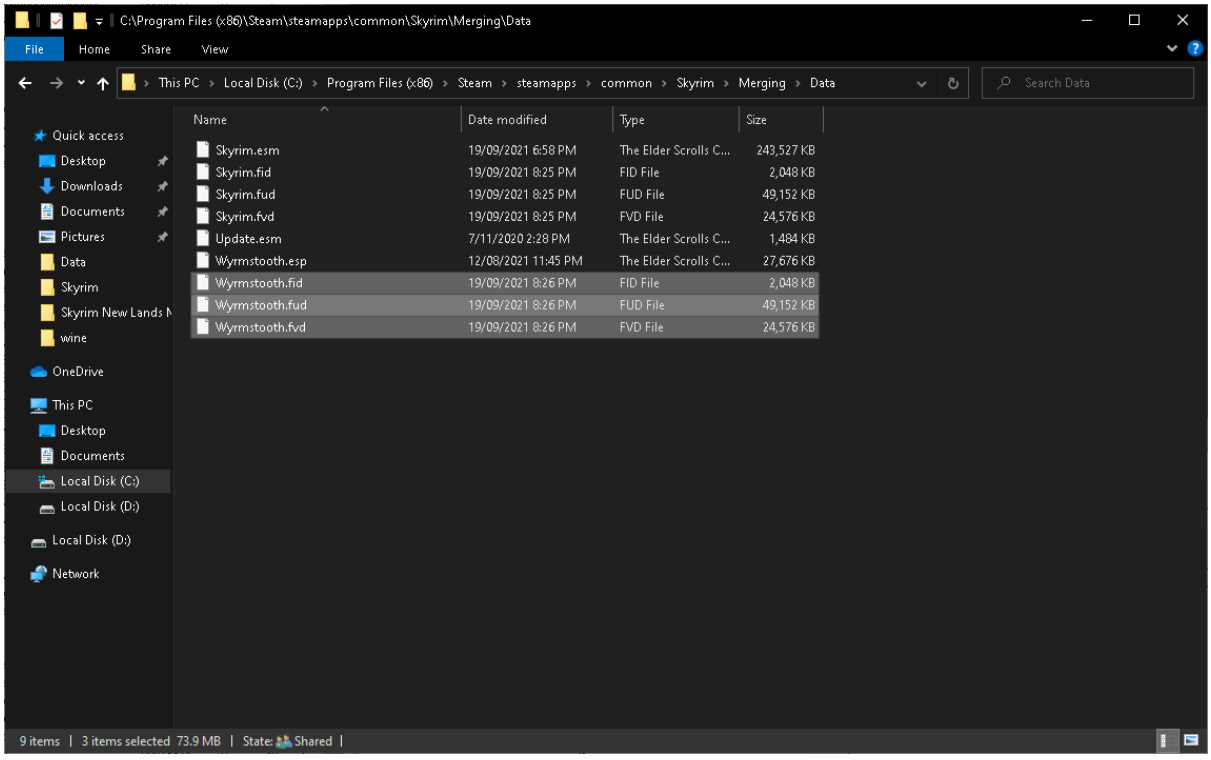


Figure 1492 - .fid, .fud, .fvd files have been created.

Important: Close and re-open the Creation Kit. If we don't do that at this stage, the Creation Kit may crash.

Also, if the file extension of your master file is .esp, you'll need to change it at this point to .esm in both your Skyrim\Data or Skyrim Special Edition\Data and Skyrim\Merging\Data or Skyrim Special Edition\Merging\Data folders.

Once again, tick your master file in the Plugin/Master Files list.

Click OK to begin loading.

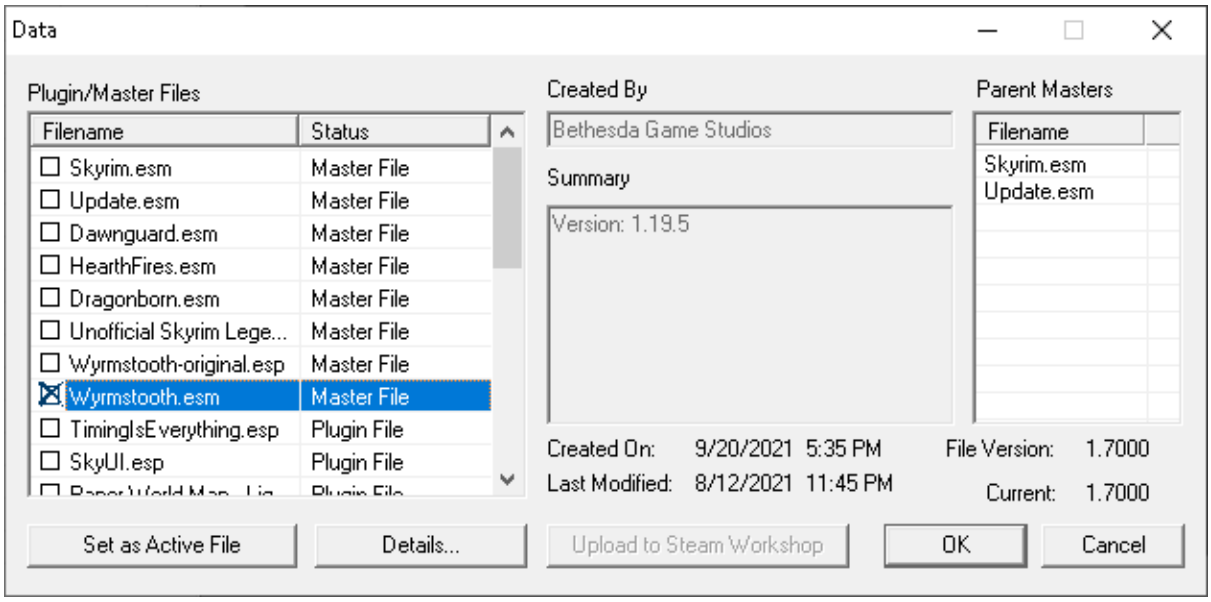


Figure 1493 - Loading your master file.

Right, now we should be ready to merge changes. But before we can do that, we need to make some changes to merge.

Just for the sake of this example, I'm going to add a barrel to an existing cell just to create a new record that we can merge into our master file.

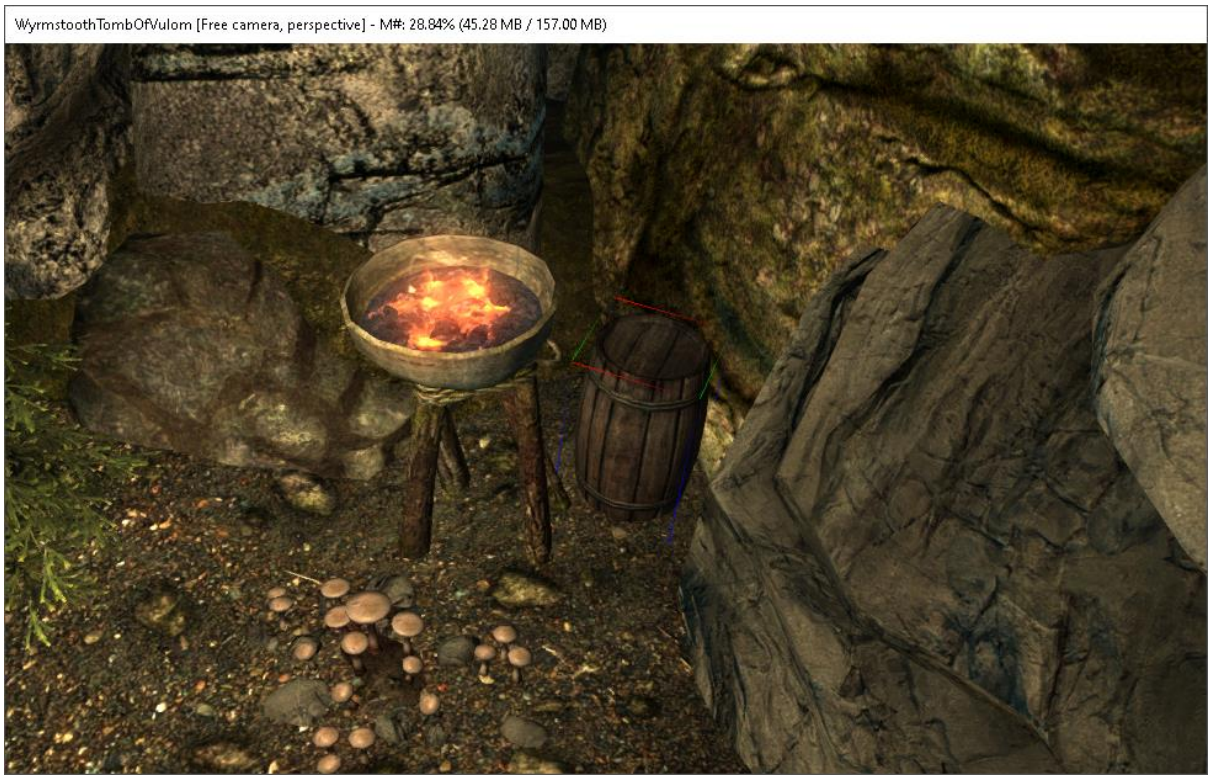


Figure 1494 - New barrel.

Go to File > Save.

Set the name of the .esp file. In my example, I just called it 'testmerge'.

Next we're going to merge this testmerge.esp file into our master.

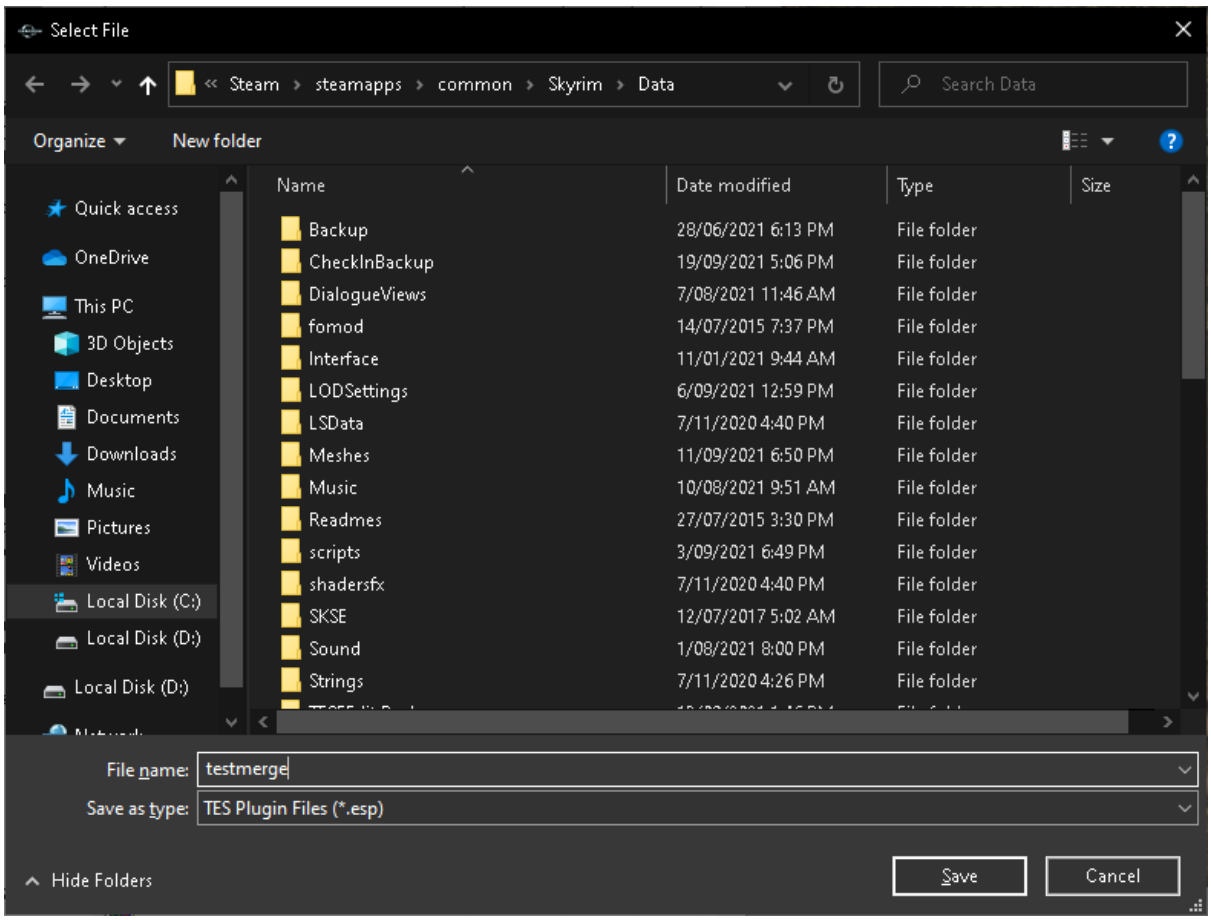


Figure 1495 - Saving a new plugin file to merge with our master file.

Click on the Version Control button at the top-left.

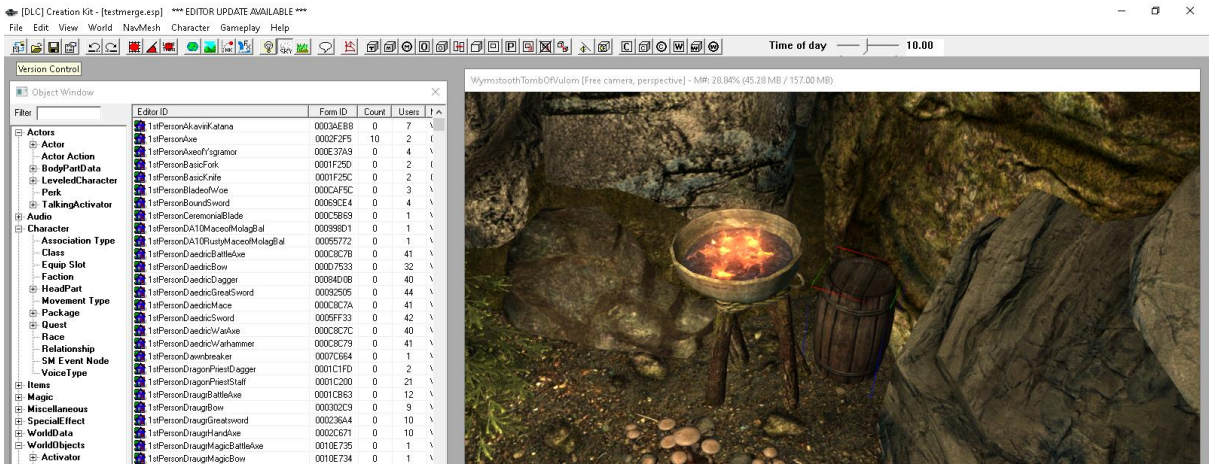


Figure 1496 - Version control button.

Click on the Select All button, then click on Check Out.

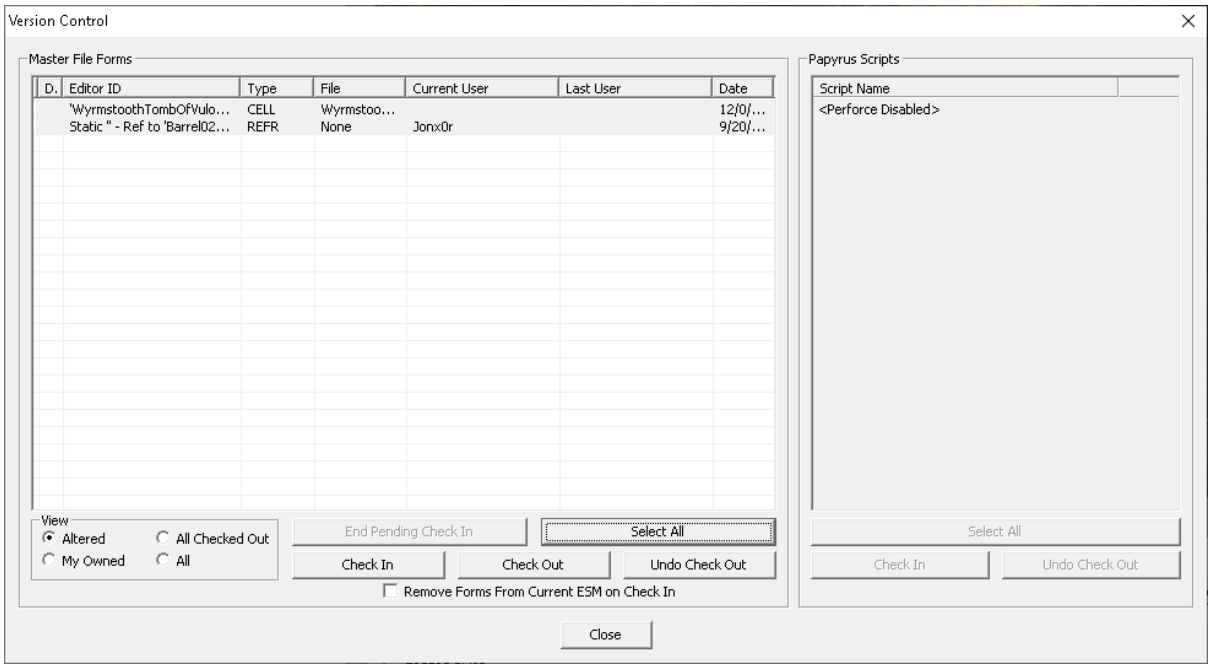


Figure 1497 - Checking out selected changes.

Click on the Select All button again, then click on Check In.

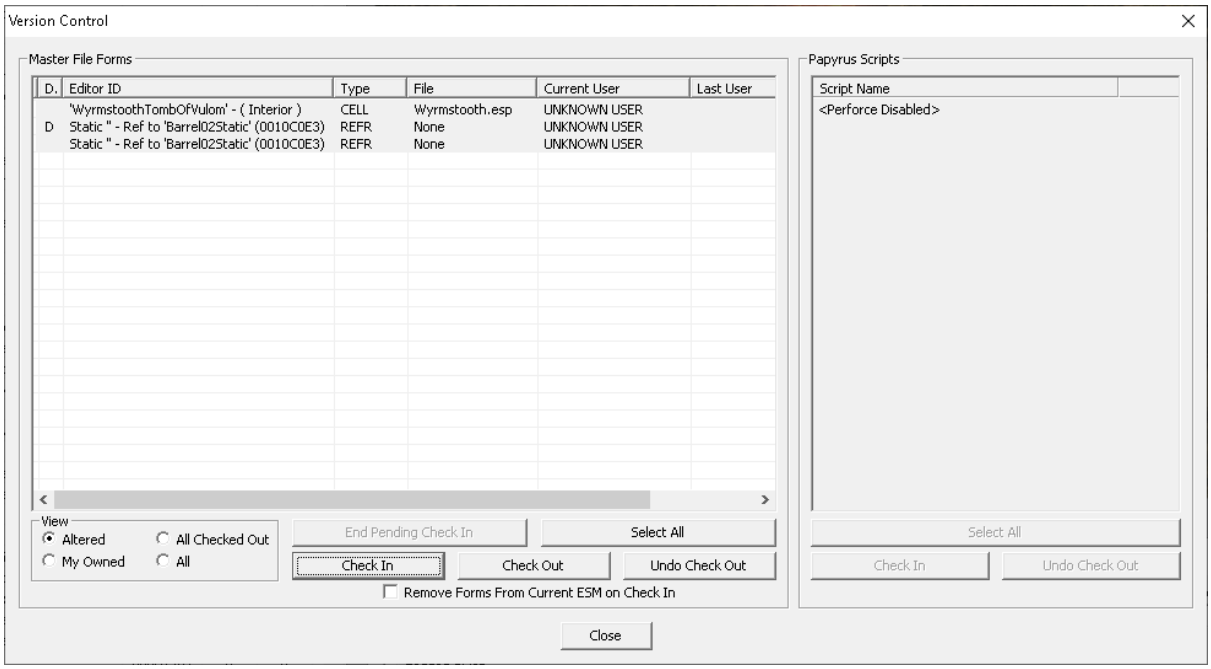


Figure 1498 - Checking in selected changes.

Enter a check in comment if you want to then click OK.



Figure 1499 - Check In comment, something to describe the changes.

Select the master file to merge these changes to.

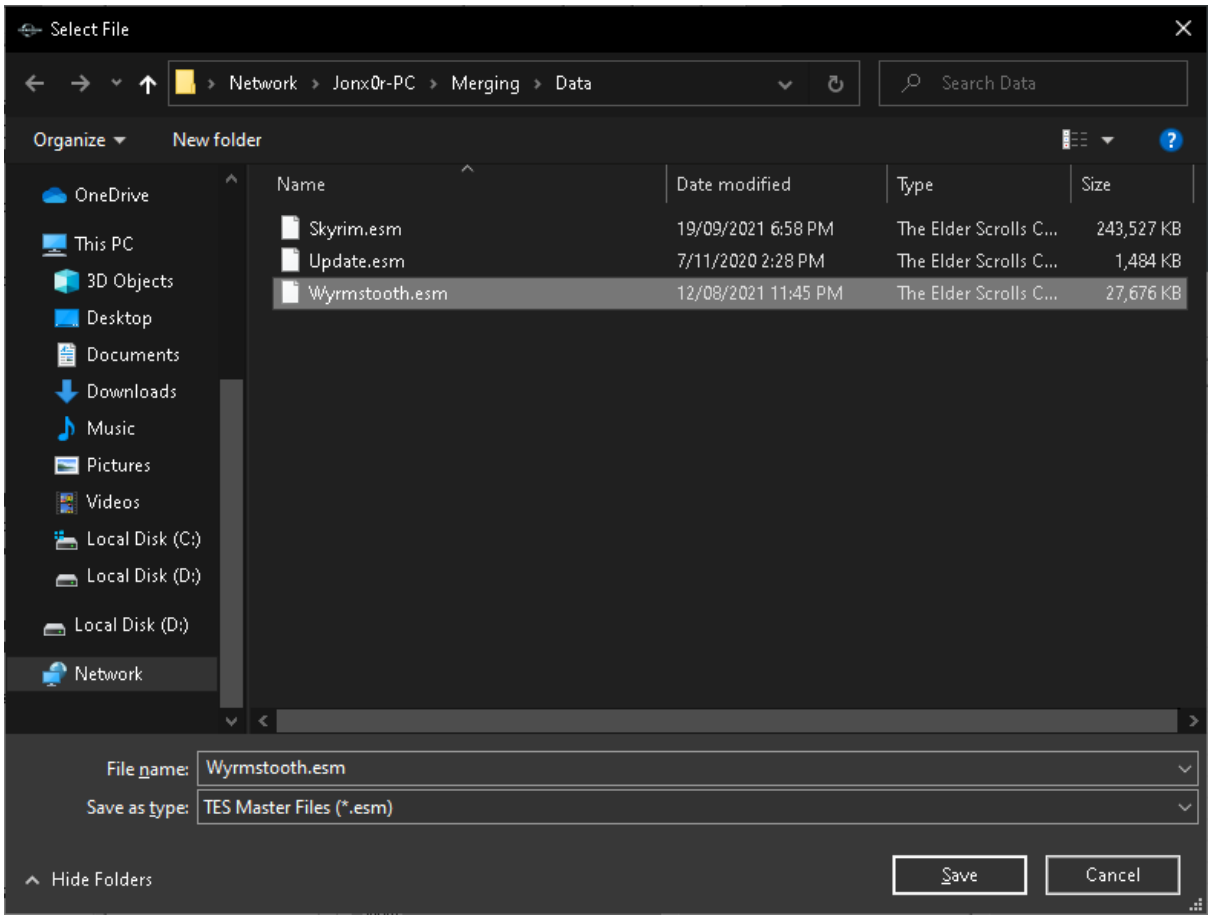


Figure 1500 - Selecting the master file to merge changes to.

Click No to the 'Remap FormIDs' pop-up.

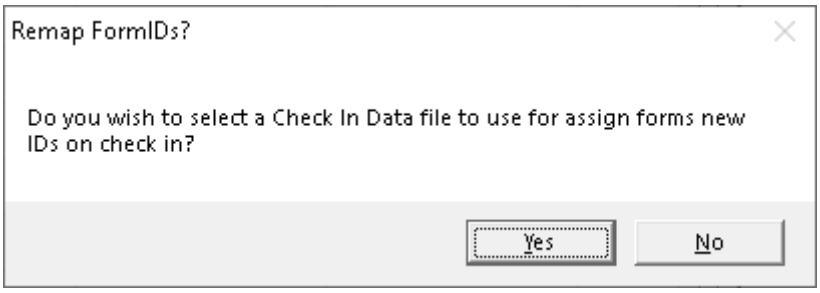


Figure 1501 - 'Remap FormIDs?' popup.

Click Close.

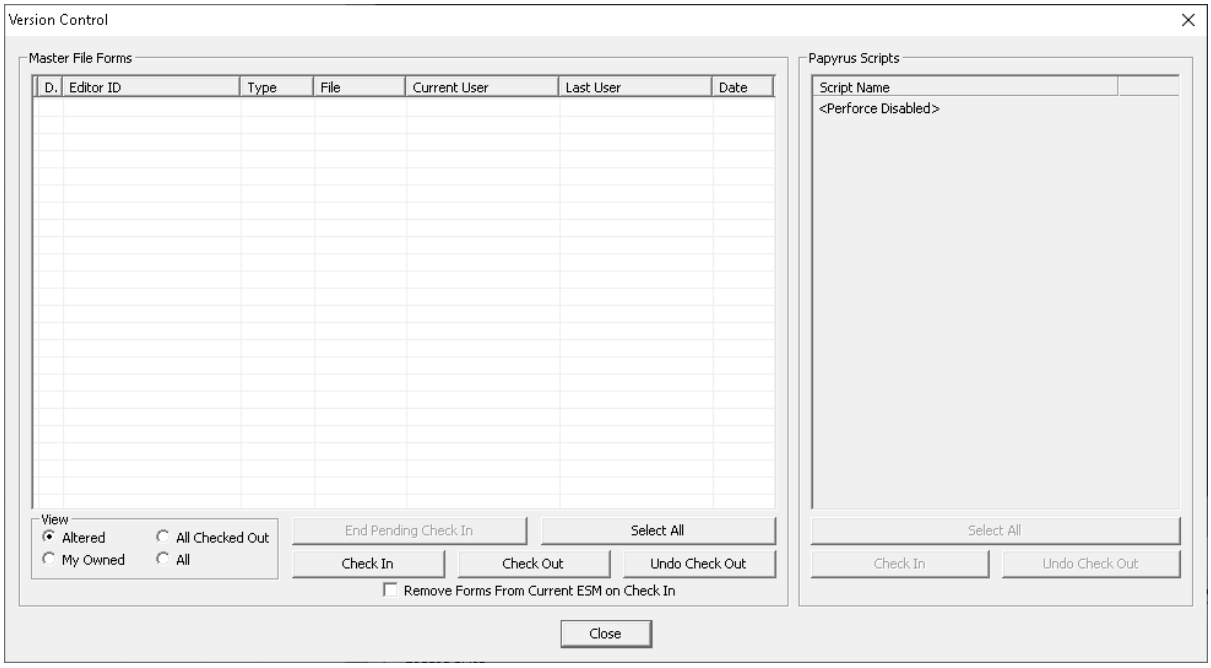


Figure 1502 - Version control after changes merged.

To test the merge, copy your master file from your Skyrim\Merging\Data or Skyrim Special Edition\Merging\Data folder to your Skyrim\Data or Skyrim Special Edition\Data folder.

Overwrite the existing file.

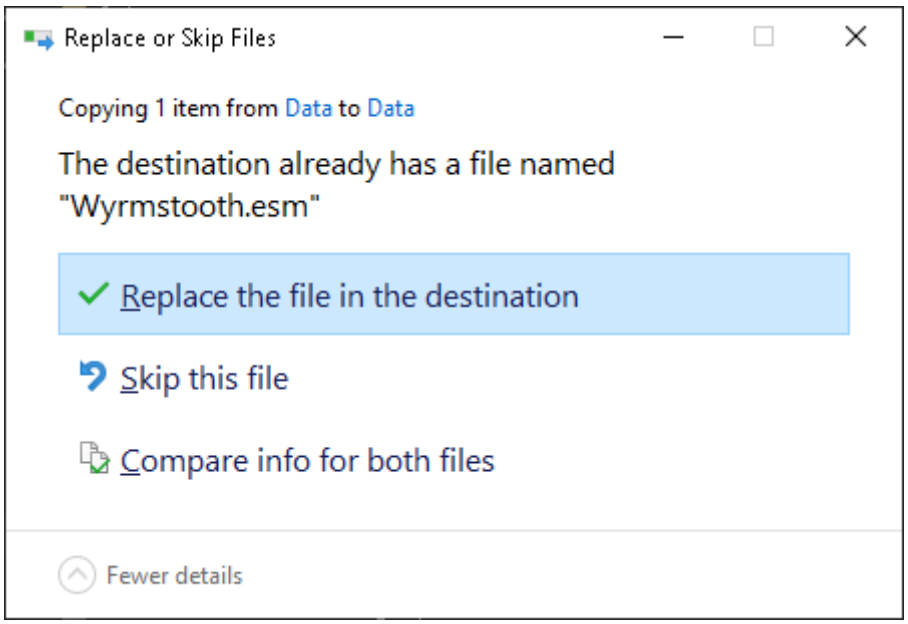


Figure 1503 - Overwrite existing master file.

Go to File > Data.

Tick your master file in the Plugin/Master Files list.

Ensure the merge plugin you created earlier isn't ticked.

Click OK.

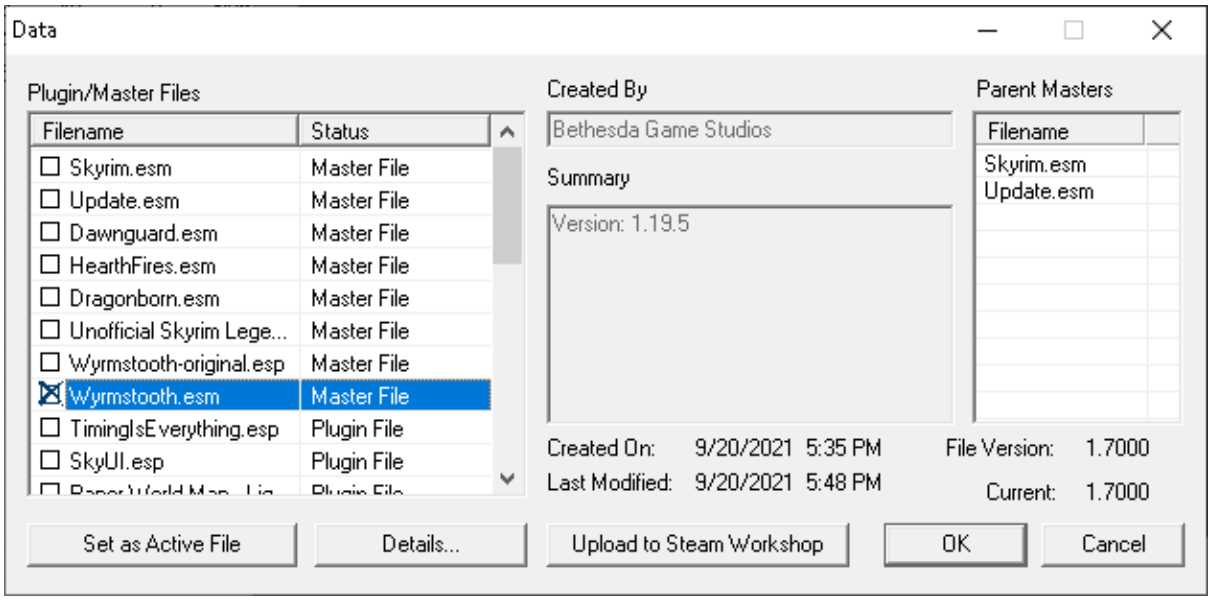


Figure 1504 - Plugin and master files.

And there's the barrel we added earlier, now part of the master file.



Figure 1505 - The barrel merged with the master file.

CLEANING YOUR MOD IN XEDIT

Cleaning will remove temporary records from your mod that are created as you edit cells, and will reduce the file size of your mod. First, go ahead and download a copy of TES5Edit or SSEEdit from Nexusmods and extract it from the .zip archive.

Launch TES5EditQuickAutoClean.exe/SSEEditQuickAutoClean.exe.

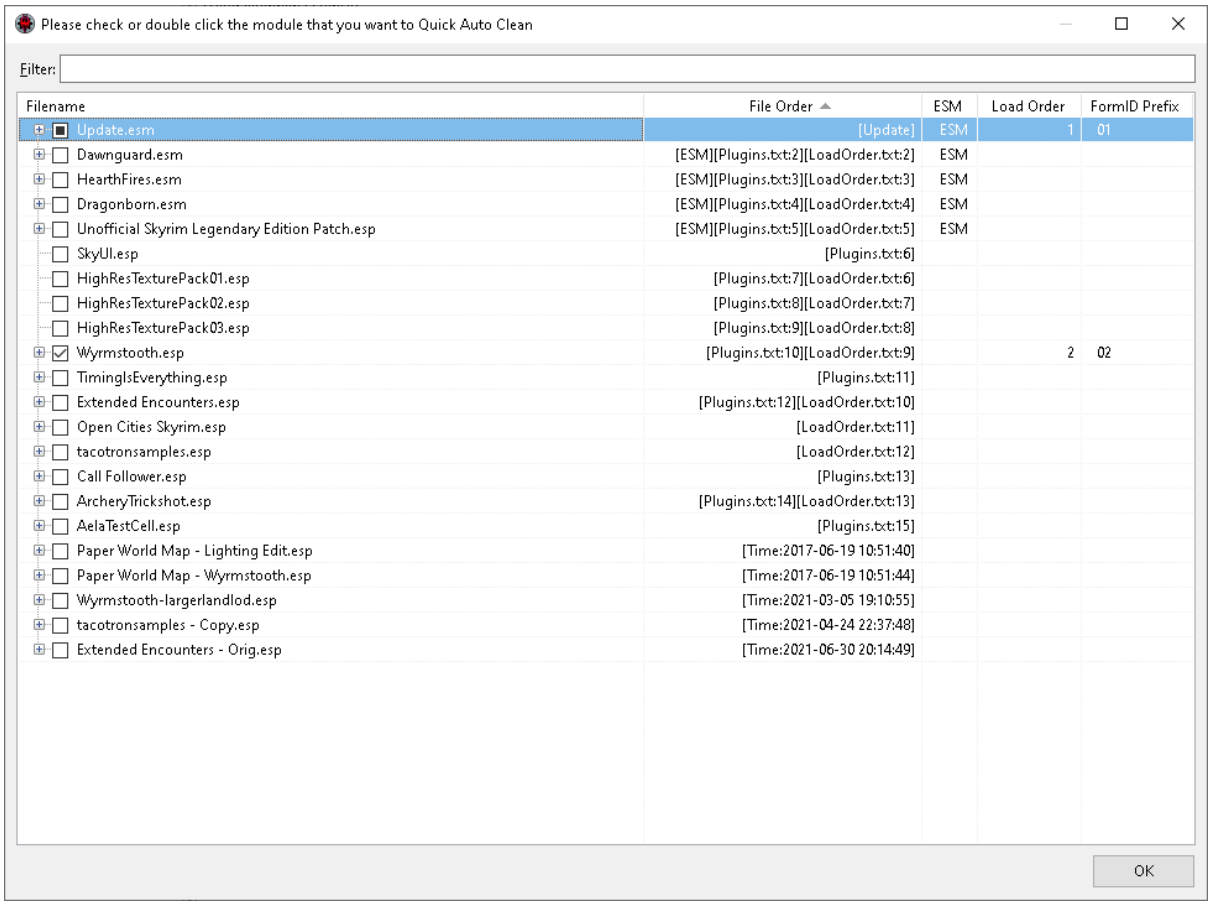


Figure 1506 - TES5Edit Quick Auto Clean module selection.

Select your mod then click OK.

Once it's done, click on the burger menu at the top-left and select Save or press CTRL + S just to make sure the changes have been saved before closing the application.

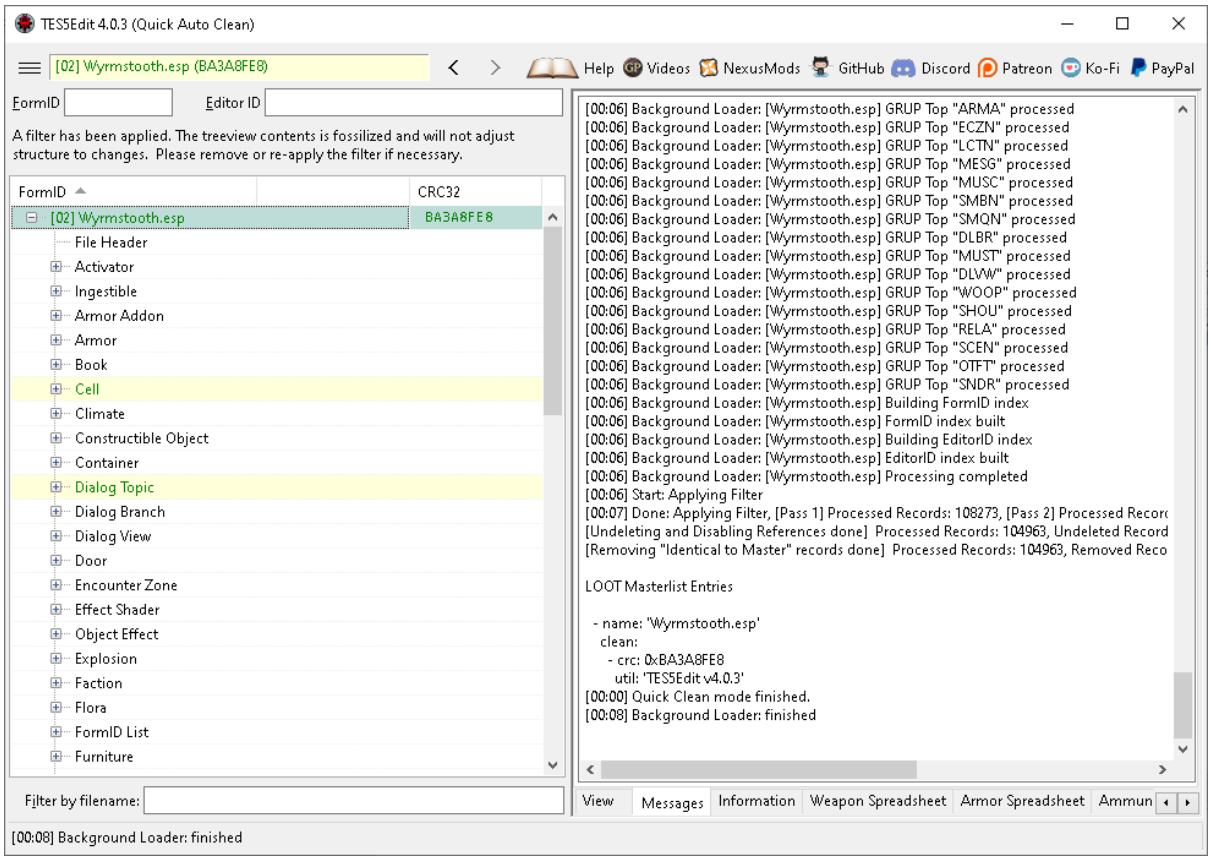


Figure 1507 - TES5Edit Quick Auto Clean Complete.

CHECKING FOR ERRORS IN XEDIT

The error checking feature in xEdit can be used to check for bad records which may potentially cause crashes in-game.

Launch xEdit.

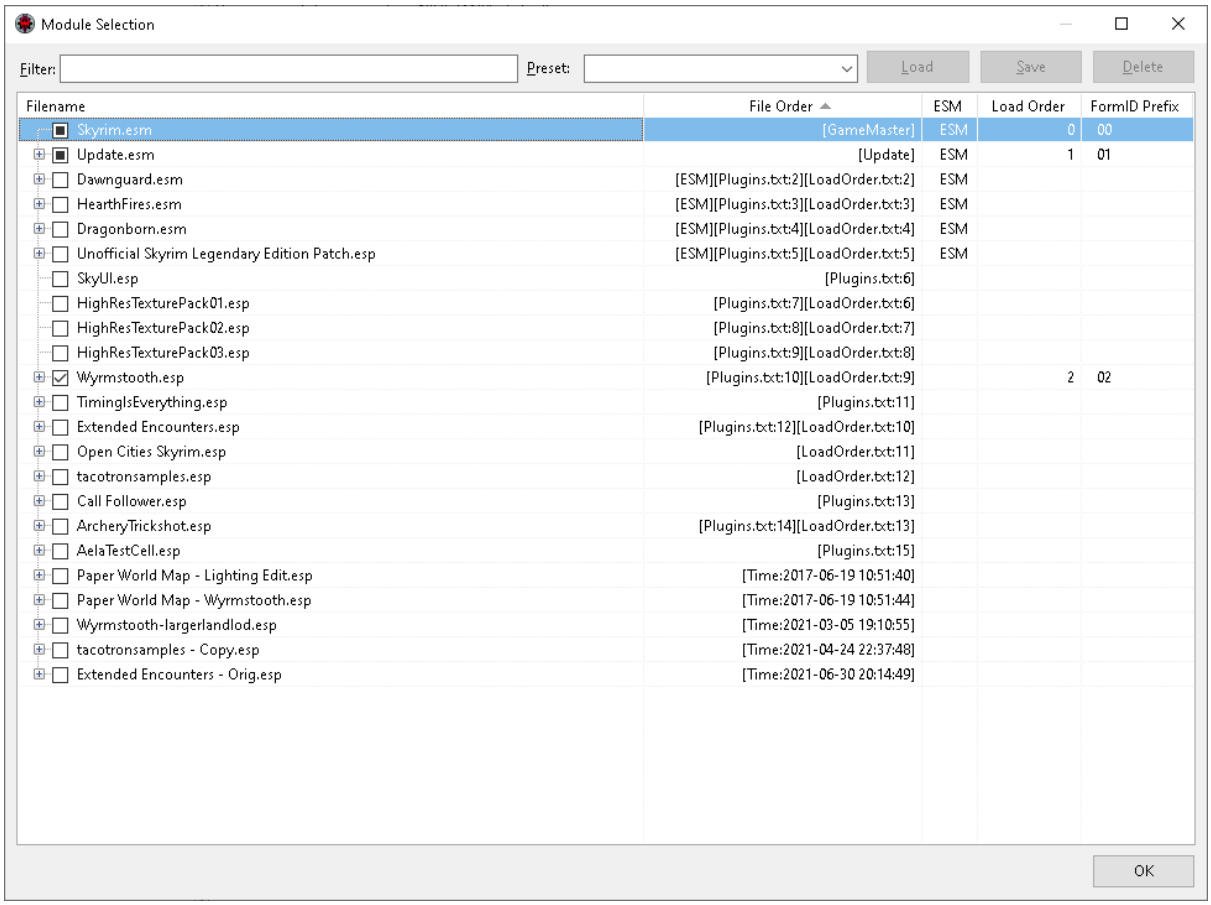


Figure 1508 - TES5Edit Module Selection.

Make sure your .esp file is selected and click OK.

Wait until the background loader has finished.

Right-click on your mod and select ‘Check for Errors’.

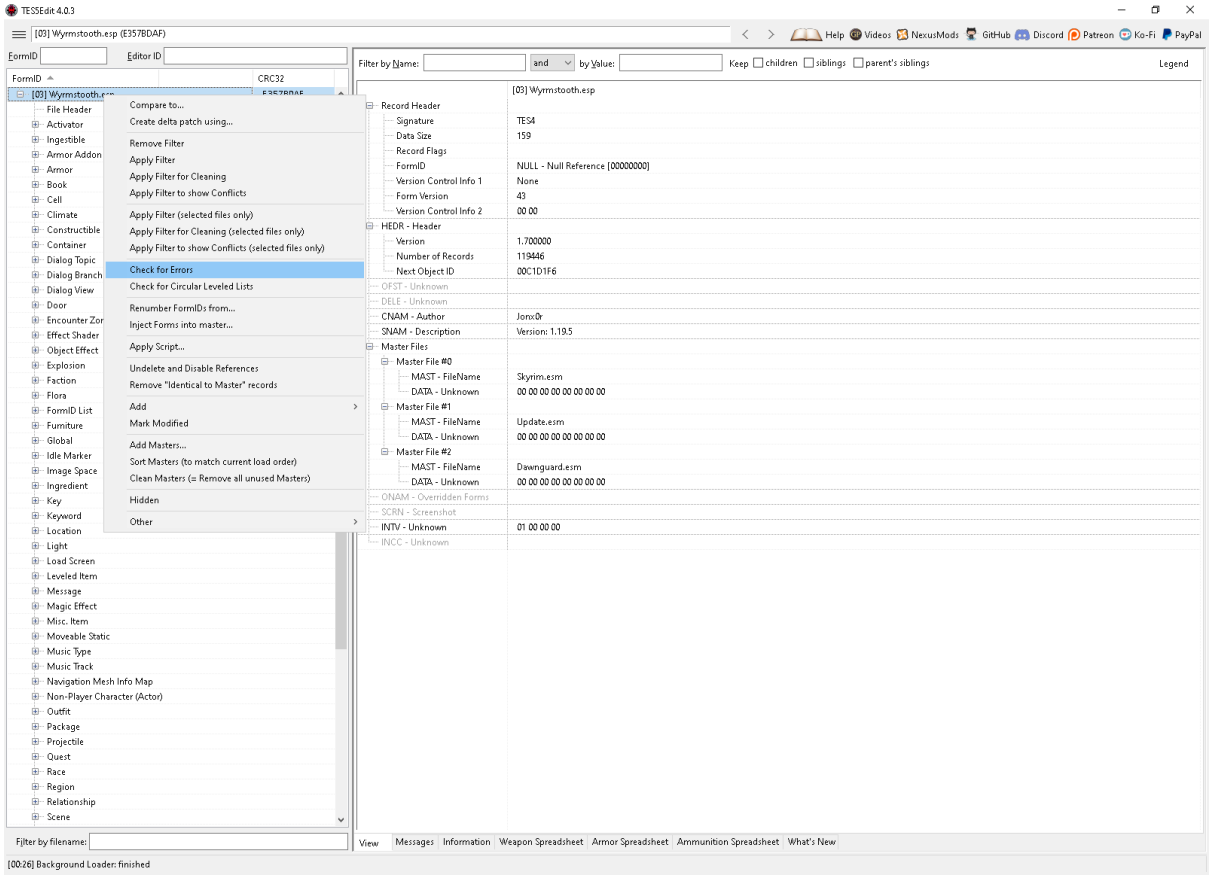


Figure 1509 - Selecting 'Check For Errors'.

Once the process is complete, the output should state ‘Done: Checking for Errors’.

For good measure, right-click on your mod and select ‘Check for Circular Leveled Lists’ as well.

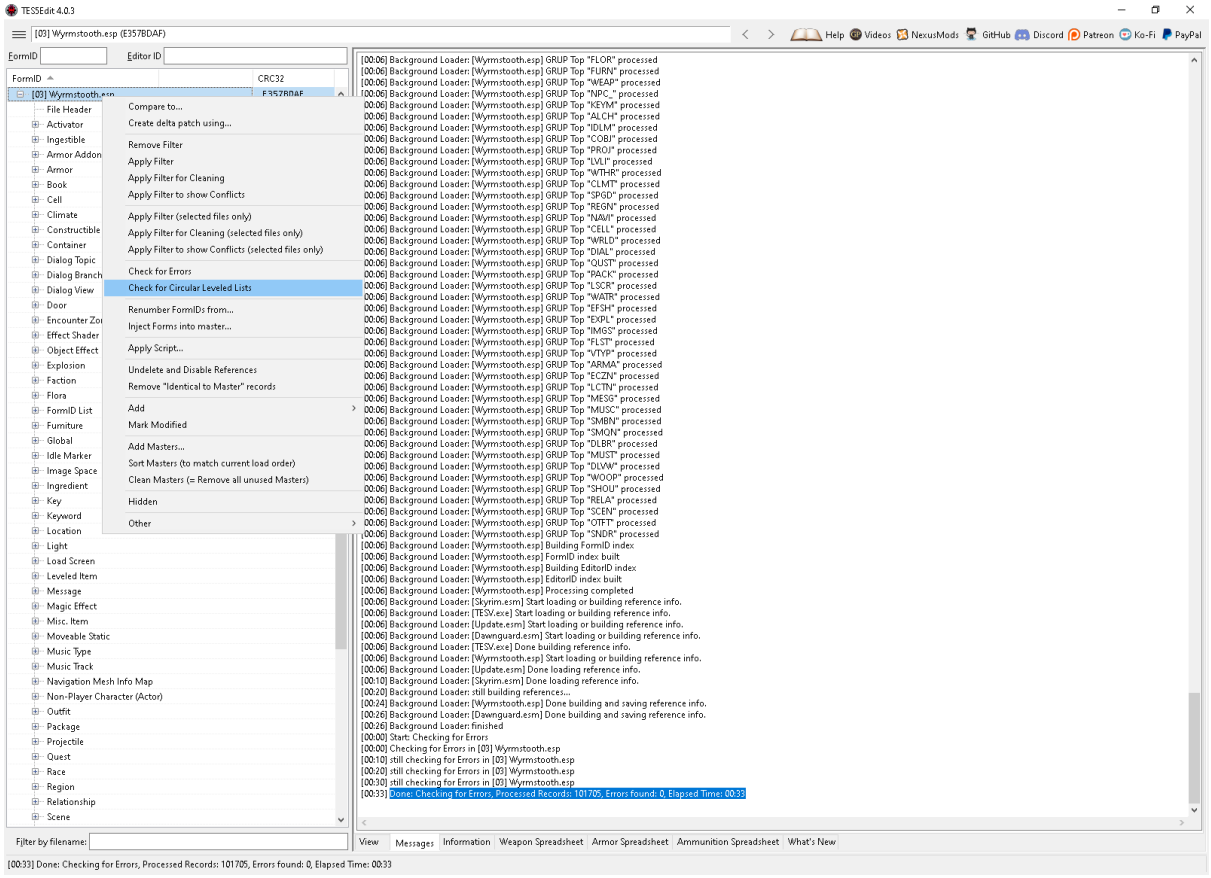


Figure 1510 - Check for Circular Levelled Lists.

Click on the Burger Menu and select Save, or press CTRL + S to save any changes.

ESM-IFYING YOUR MOD IN XEDIT

Skyrim's DLC (Dawnguard, Hearthfire, Dragonborn) contain an 'esm' flag. This flag allows other mods to treat the DLC as master files, so you can use content from them in your mods. You can add the 'esm' flag to your own mod to allow other mods to build upon it in the same manner.

Launch xEdit.

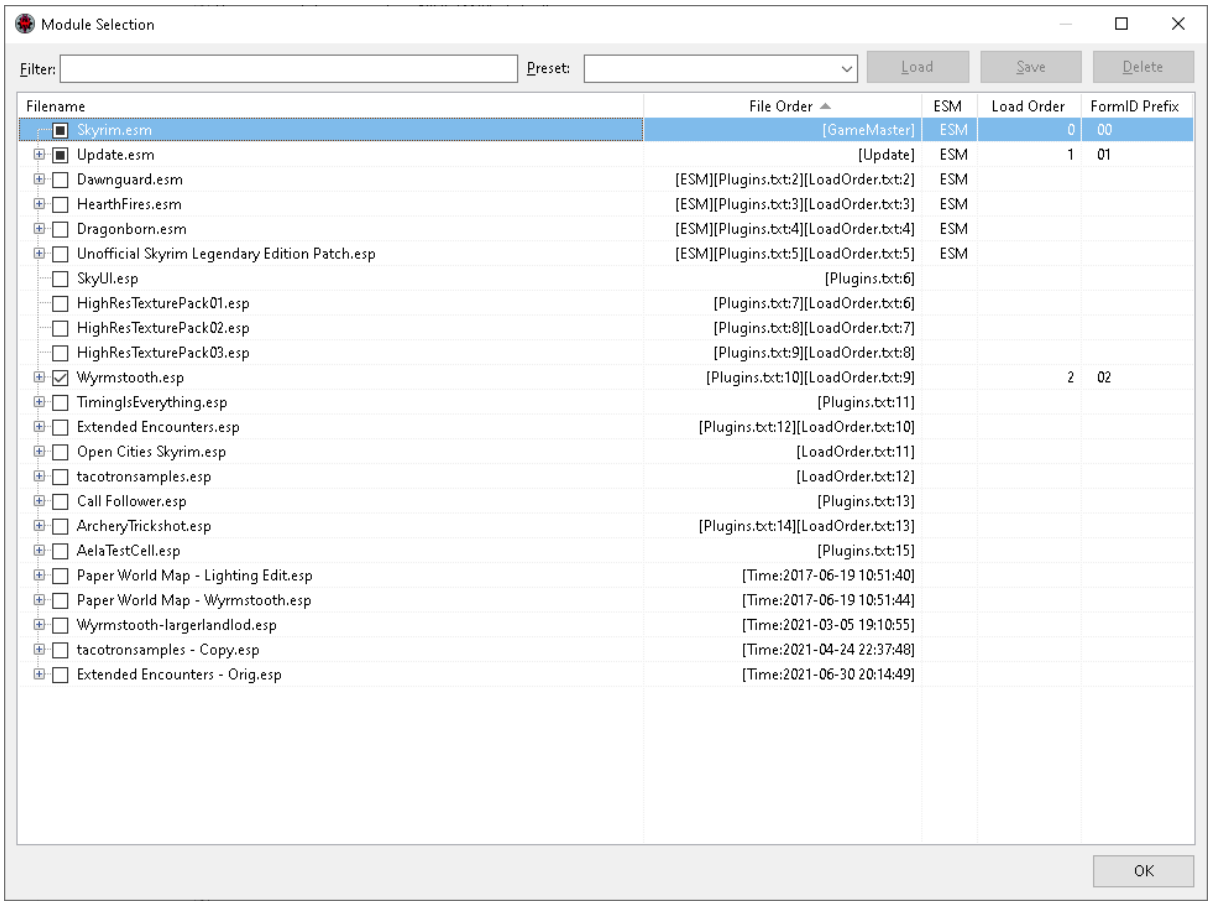


Figure 1511 - TES5Edit Module Selection.

Make sure your .esp file is selected and click OK.

Wait until the background loader has finished.

Right-click on the column to the right of Record Flags and select Edit.

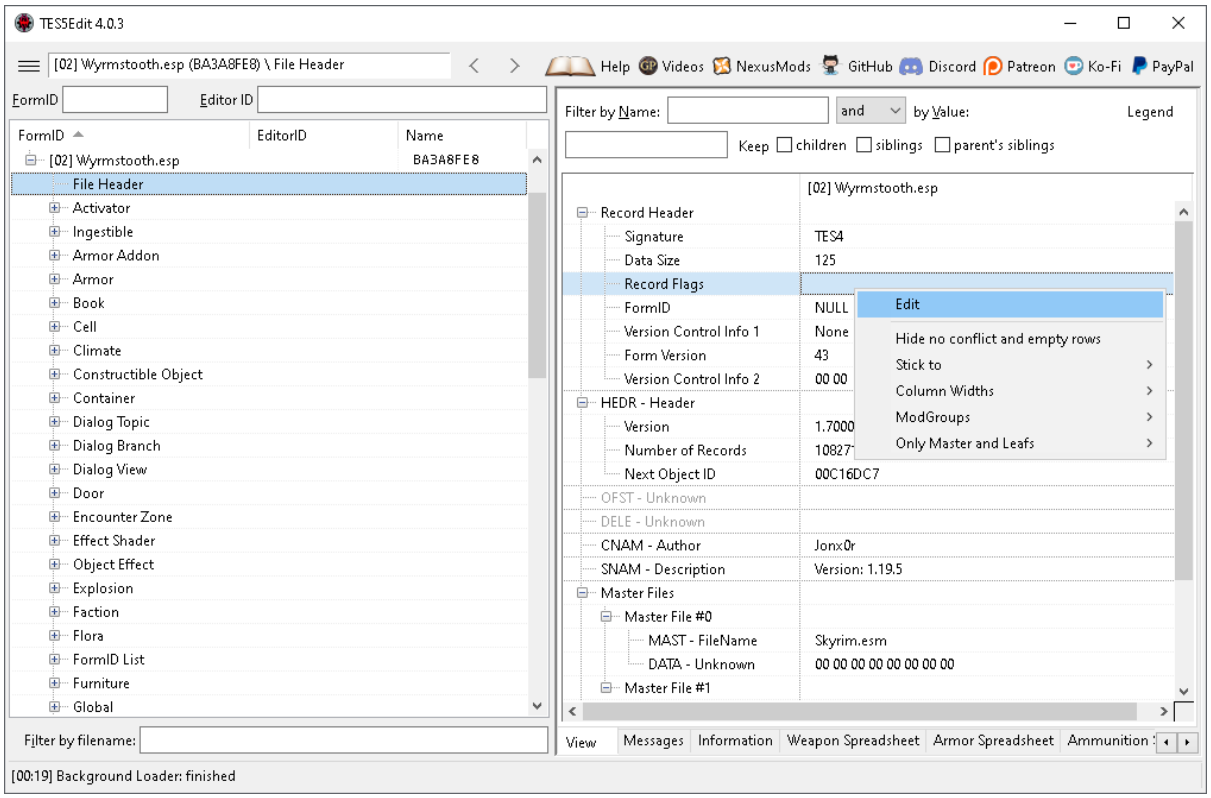


Figure 1512 - TES5Edit.

You'll see a warning about editing a module file. Click 'Yes I'm absolutely sure'.

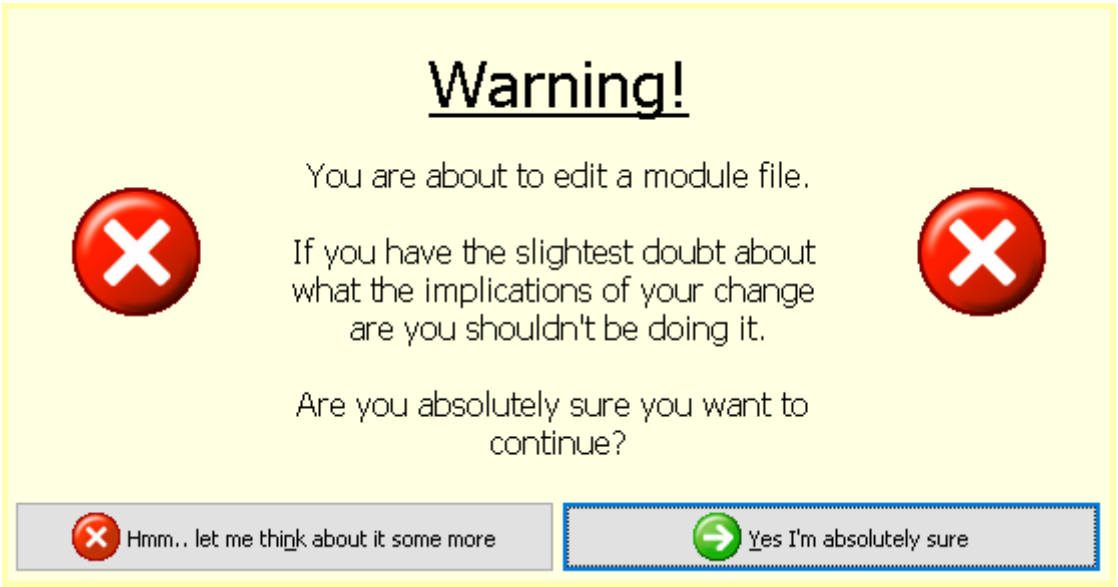


Figure 1513 - TES5Edit change warning.

Tick the 'ESM' tick box then click OK.

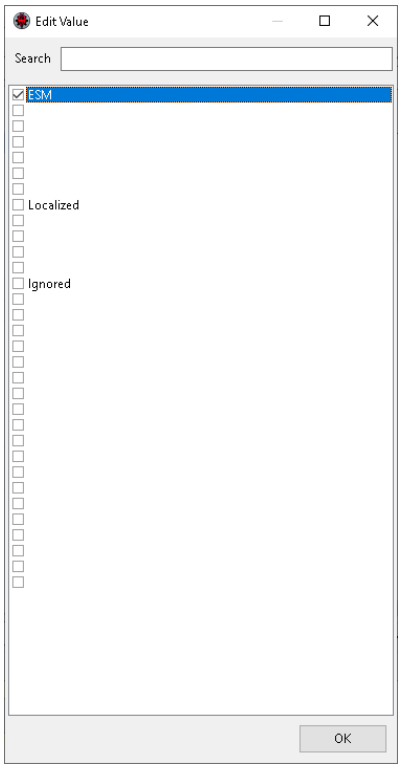


Figure 1514 - TES5Edit Edit Value.

Go to the burger menu and select Save, or press CTRL + S to save this change. Optionally, you can also change the file extension of your mod from .esp to .esm.

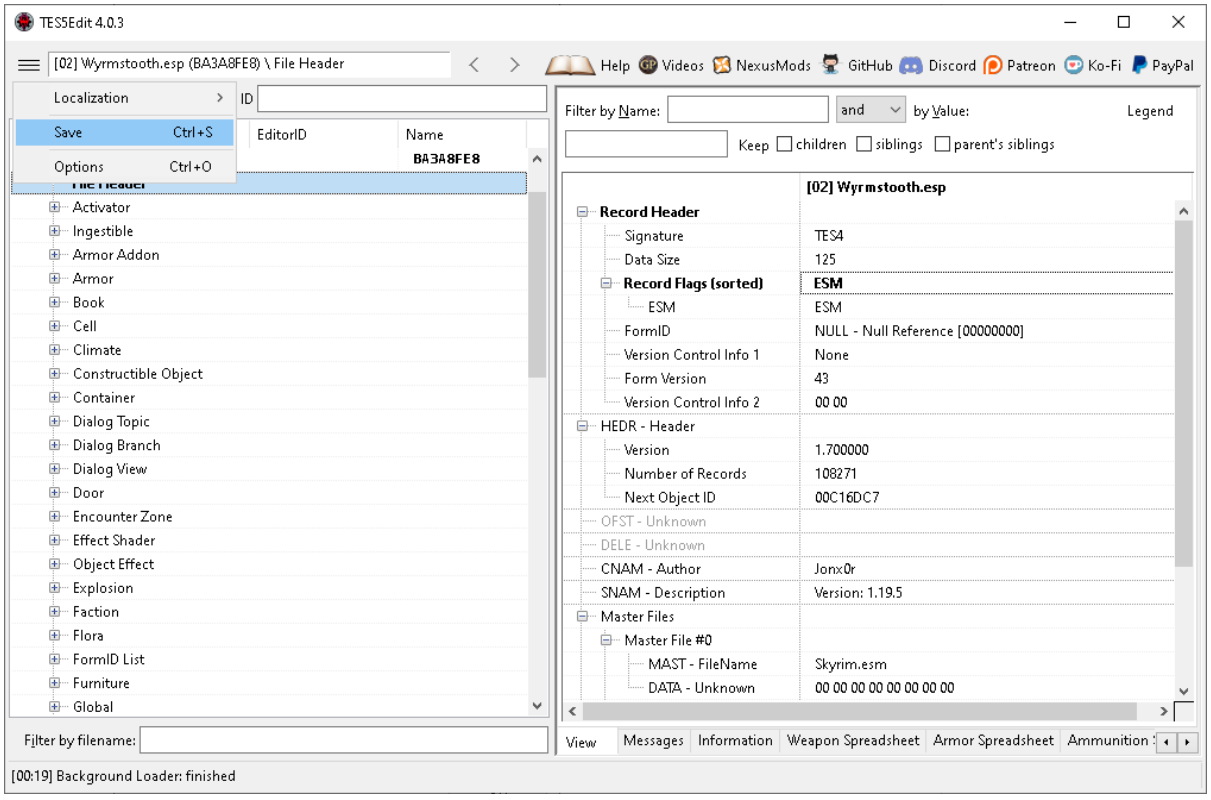


Figure 1515 - TES5Edit Save.

GENERATING PAPYRUS DEBUGGING TRACE

Before packaging your mod, you should do a playtest with papyrus debugging enabled. Papyrus debugging will help identify issues with any of your scripts.

You can enable papyrus debugging in your Skyrim.ini file under the [Papyrus] section.

The Skyrim.ini file is located under C:\Users\<YourUsername>\Documents\my games\Skyrim for Skyrim and C:\Users\<YourUsername>\Documents\my games\Skyrim Special Edition for Skyrim Special Edition.

Set bEnableLogging to 1 and bEnableTrace to 1 as per the screenshot below.

If these entries don't exist, add them in.



Figure 1516 - Enabling papyrus debugging in the Skyrim.ini file.

When you launch the game, a Logs\Script folder will be created containing a Papyrus.#.log file. Each time you launch the game, a new .log file is created, up to a total of four log files.

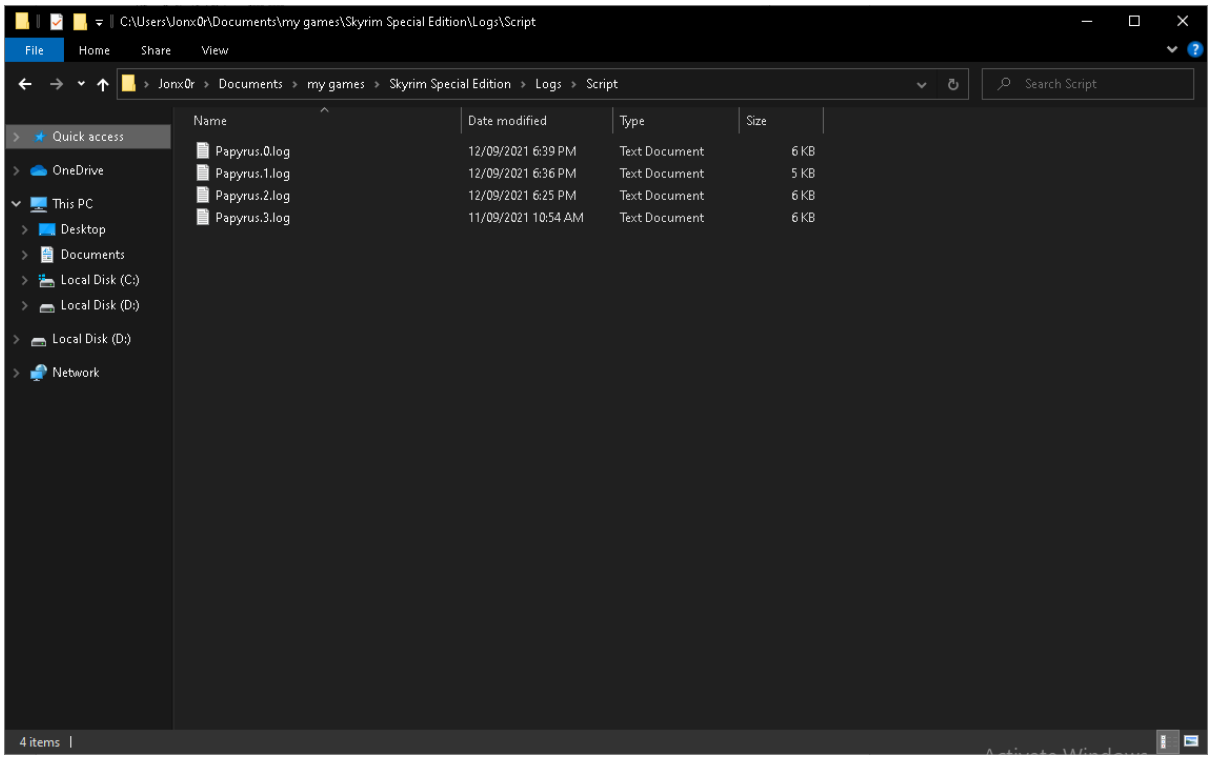


Figure 1517 - Log files.

Before each release, I do a full play-through of the main quest in Wyrmstooth and any side-quests that changed since the last update, then check the log file for warnings or errors.

Each log file contains the papyrus trace.

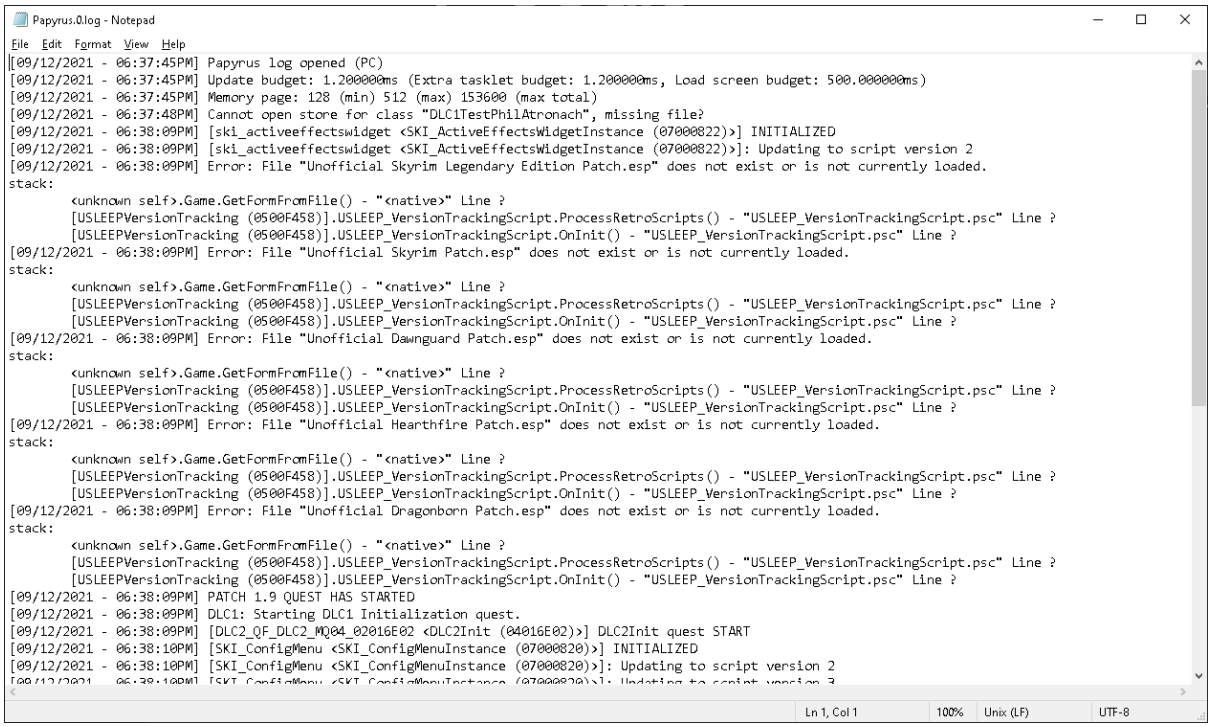


Figure 1518 - Papyrus debug.

Most of the trace will come from the base game, DLCs, and other mods you have loaded.

All my scripts for Wyrmstooth are prefixed with 'WT' so I search for that prefix to look for warnings or errors that are relevant to my mod.

When you're done testing your papyrus scripts, don't forget to set bEnableLogging and bEnableTrace back to 0.

CREATING A .BIK VIDEO FILE

This section will cover the steps required to compress an .avi video in the .bik format so that it can be played in-game.

In order to do this, we'll need a copy of RAD Game Tools.

For this example, I'll be working with a video I put together in Adobe After Effects. This video is used in Wyrmstooth and plays when the player travels to the island for the first time.

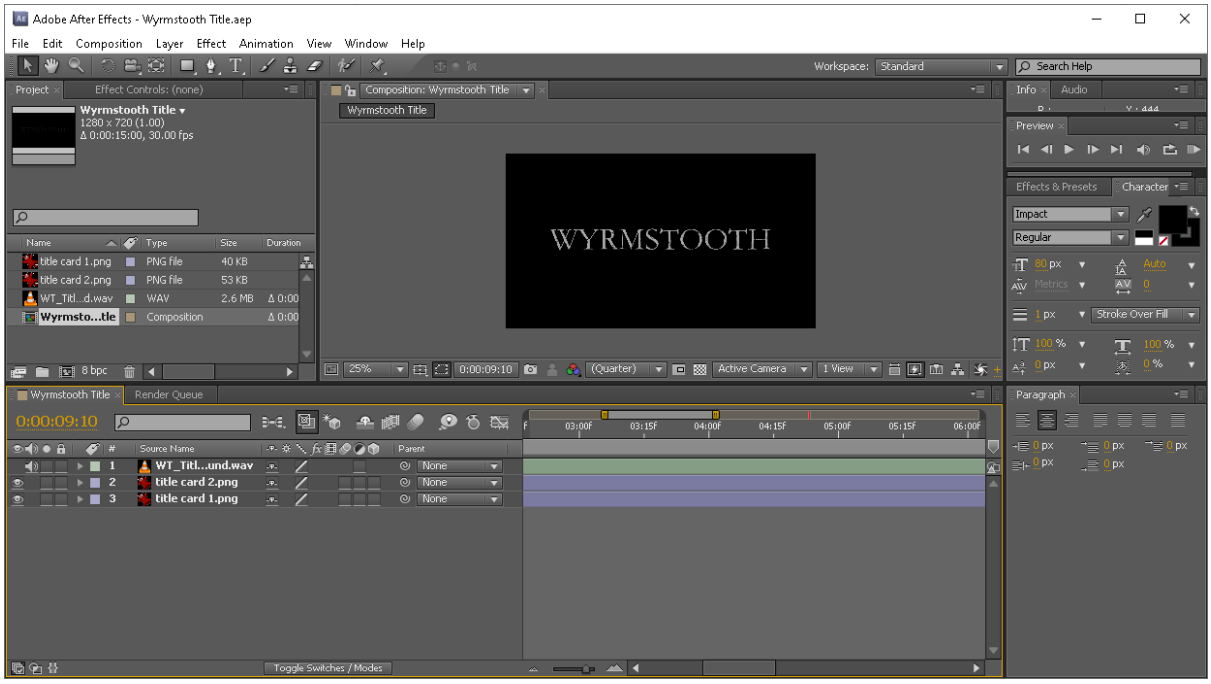


Figure 1519 - The video to convert to .bik.

The first thing we'll need to do is render an .avi file at lossless quality.

As per the screenshot below, I set Render Settings to Best Settings and Output Module to Lossless.

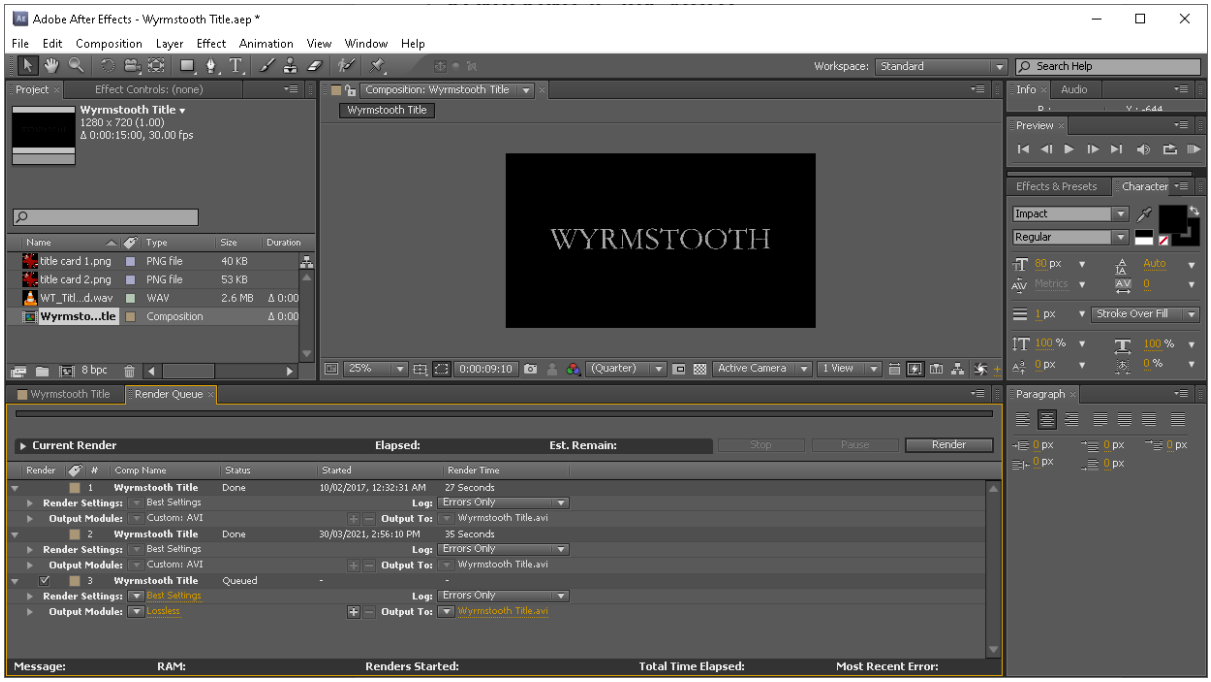


Figure 1520 - Rendering from After Effects.

Important: This will result in a very large file, so make sure you have enough disk space to work with!

Once the .avi file has rendered, open RAD Video Tools.

Select the .avi file and press the 'Bink it!' button.

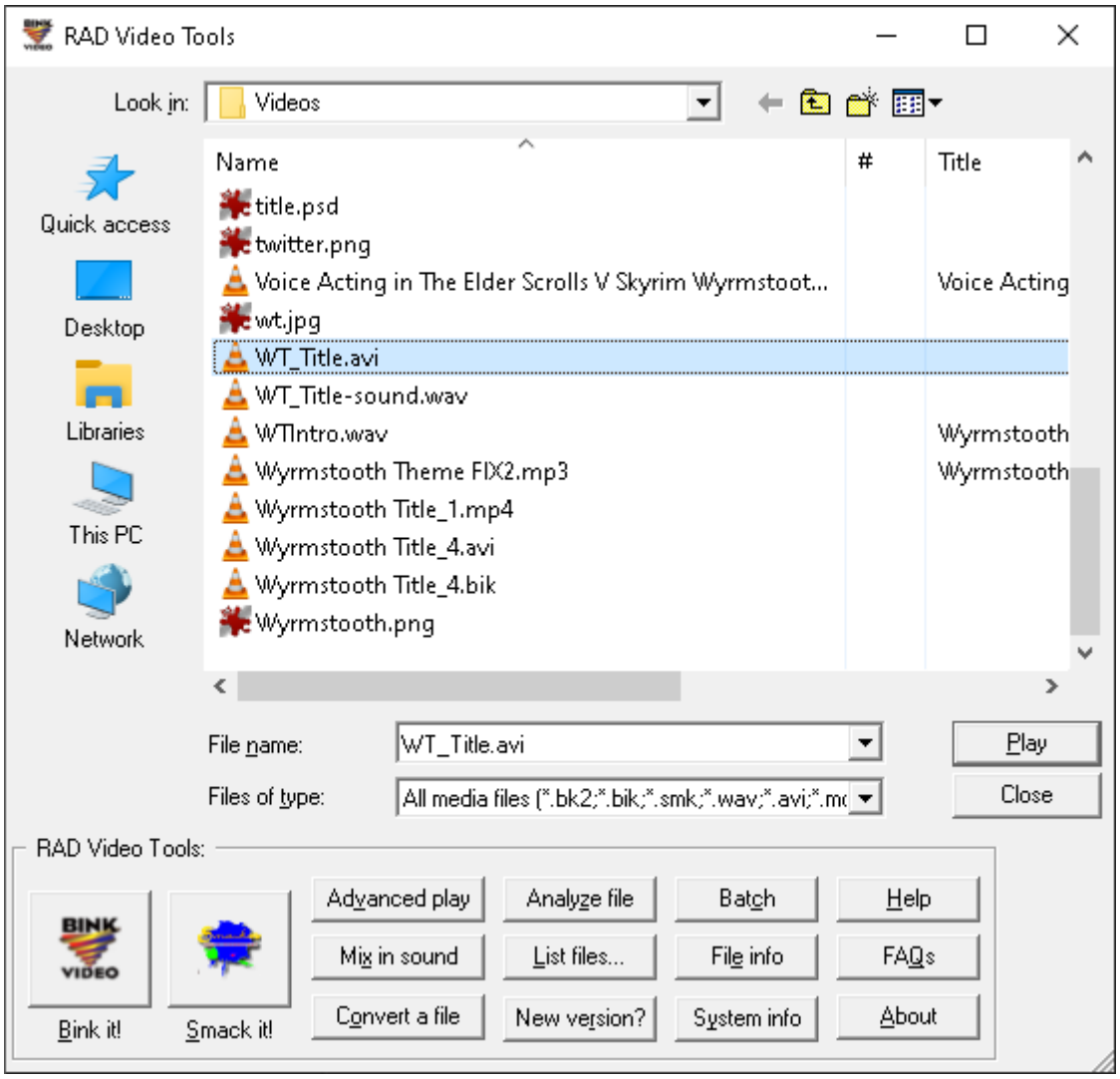


Figure 1521 - Binking the .avi file rendered from After Effects.

Ensure 'Compress audio' is ticked. Set the 'Compress level' to 0 and 'Convert to rate' to 44050. For 'Convert to what sound format', tick '16-bit/stereo'.

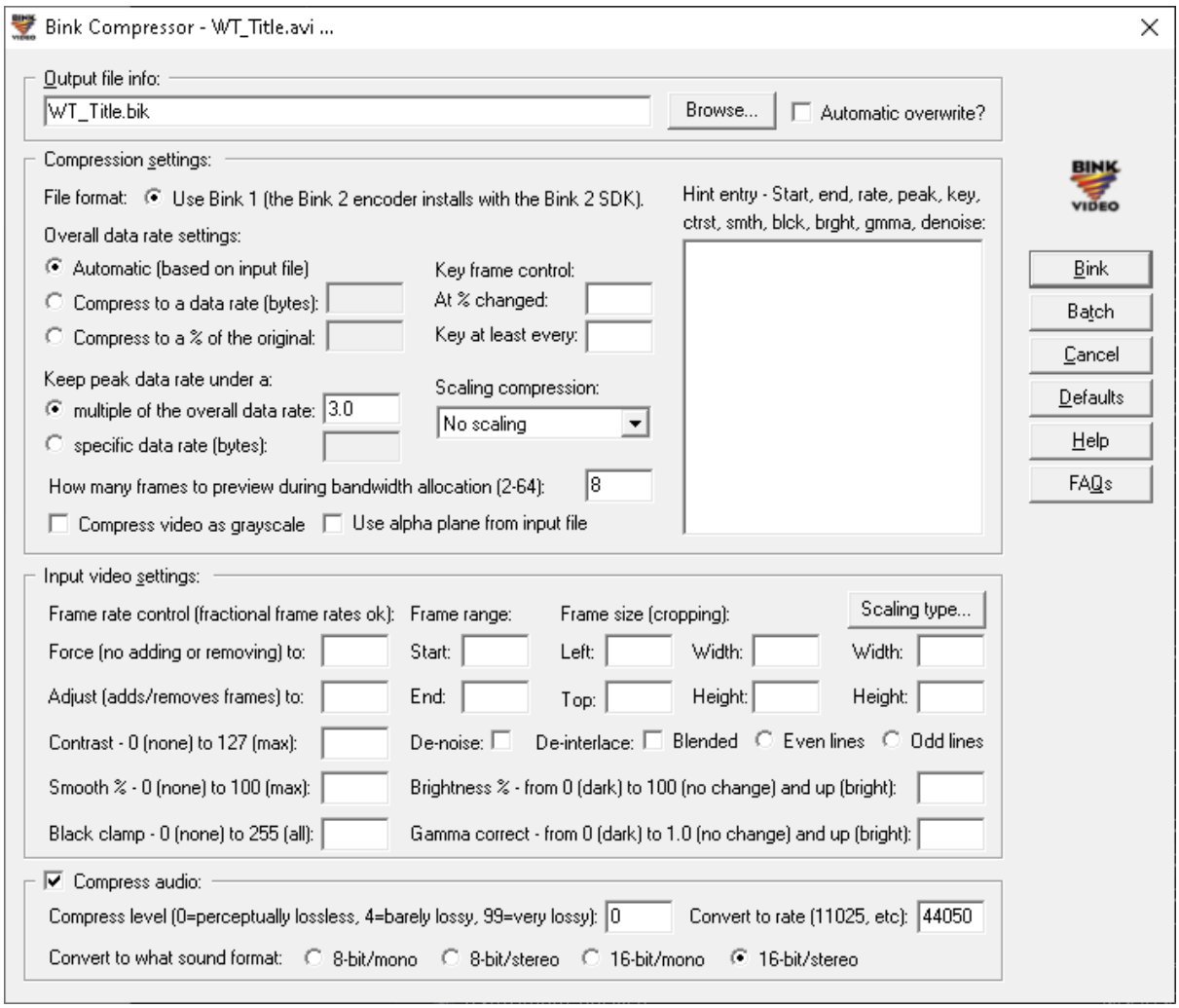


Figure 1522 - Bink settings.

Click Bink to begin compressing the video.

Once it's done, confirm you have a .bik file in the same directory as the .avi file.

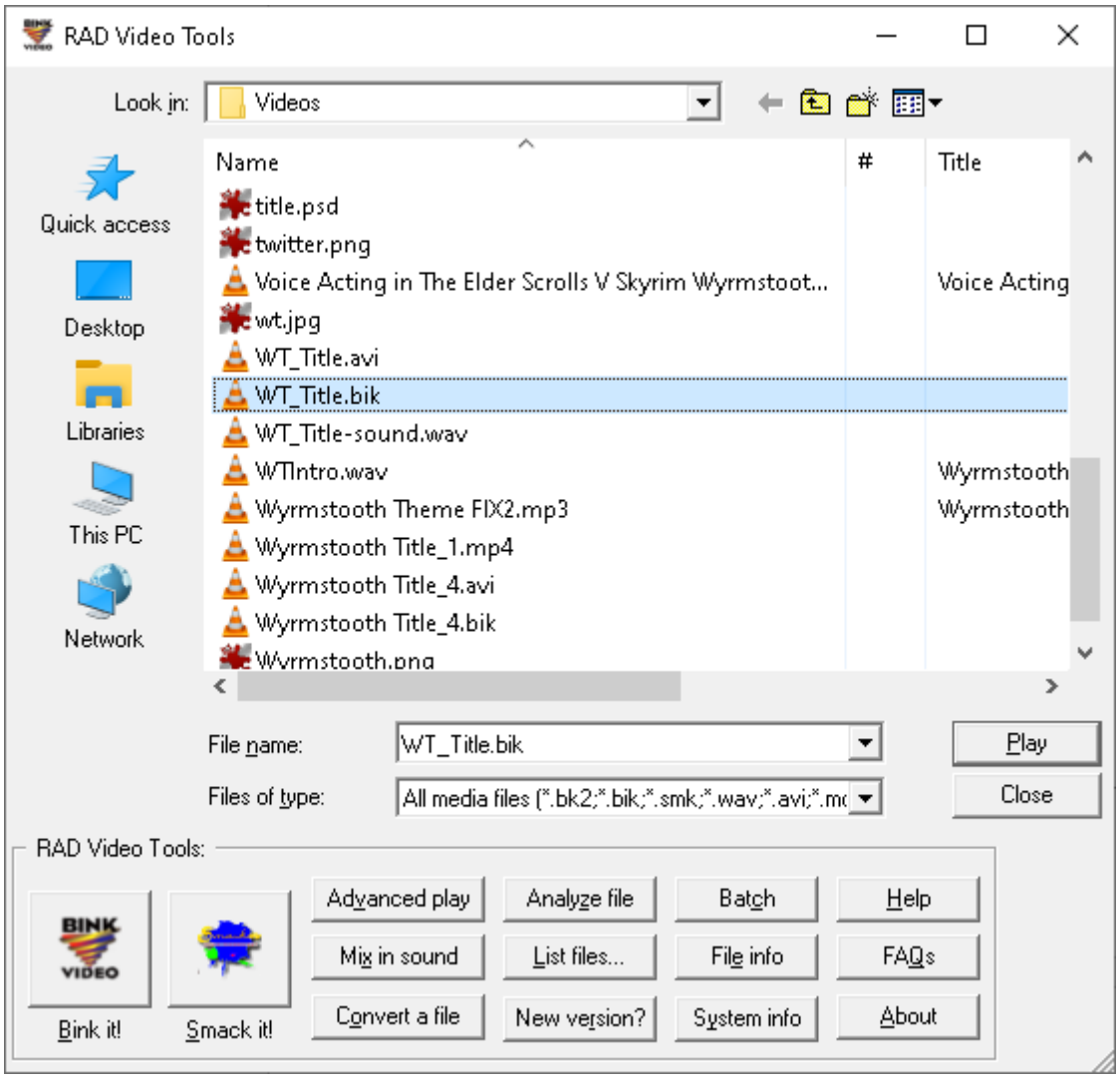


Figure 1523 - .bik file created.

Now we can add it to our mod.

Note: .bik videos cannot be packed into .bsa archives. I include my .bik video in my mod as a loose file.

Copy the .bik file to your Skyrim\Data\Video or Skyrim Special Edition\Data\Video folder.

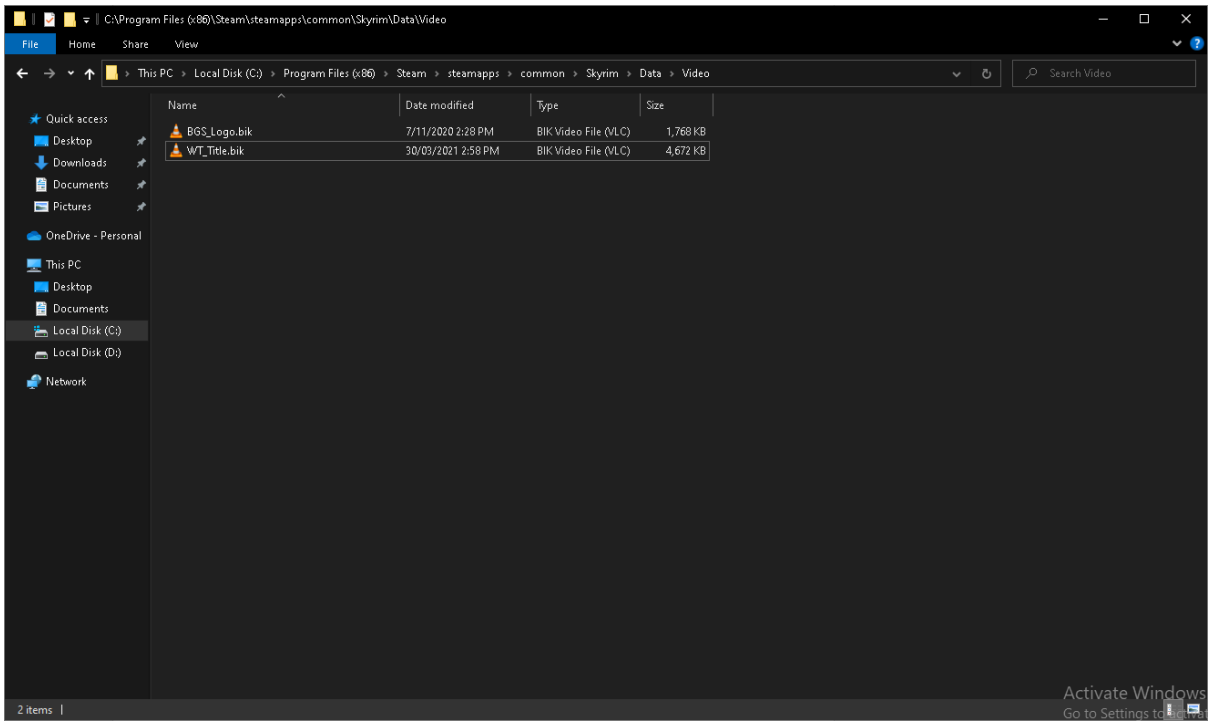


Figure 1524 - .bik file copied to Skyrim directory.

To play a .bik video, use the following papyrus command:

```
game.PlayBink("WT_Title.bik", false, true, true, true)
```

For more information on the PlayBink command, see the article on the Creation Kit wiki.

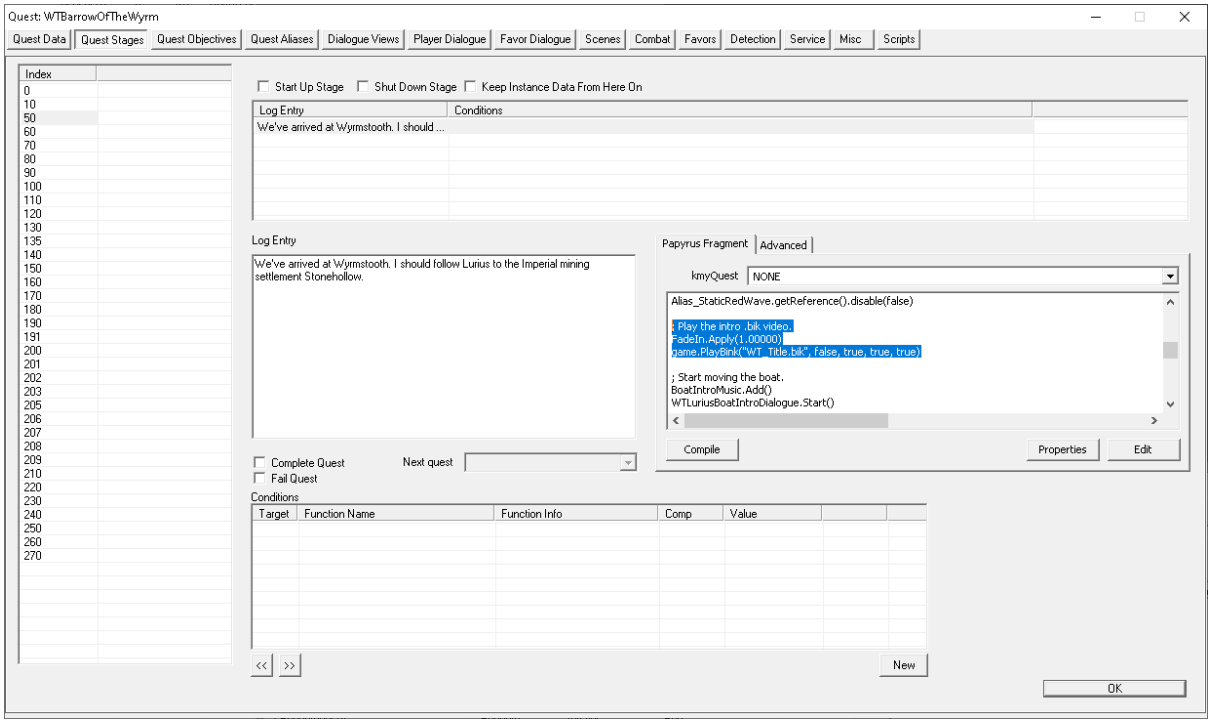


Figure 1525 - Papyrus to play a .bik video.

CHAPTER 12: PACKAGING THE MOD FOR RELEASE

PORTING YOUR MOD TO SKYRIM SPECIAL EDITION

If you developed your mod for the original version of Skyrim and would like to port it to Special Edition, this section will guide you through that process.

The first thing you need to do is copy the .esp file from your Skyrim Data folder to your Skyrim Special Edition Data folder.

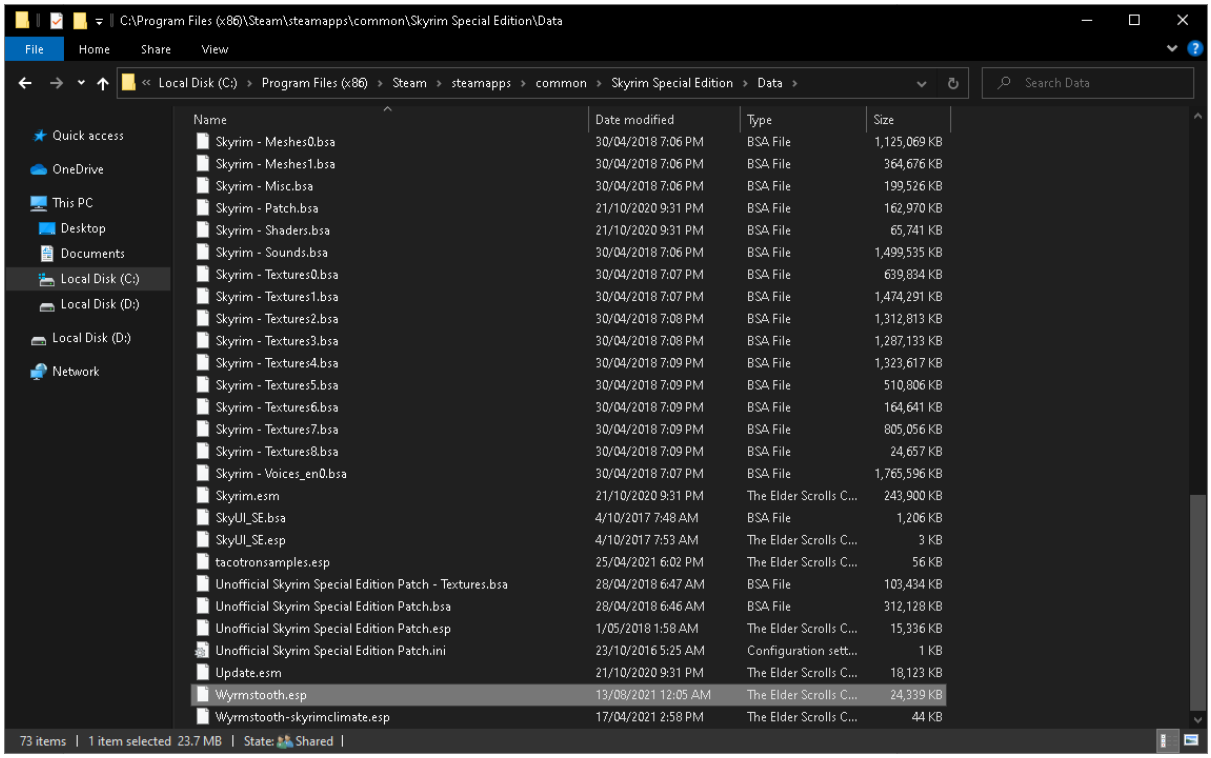


Figure 1526 - Copying the .esp to the Skyrim Special Edition data folder.

Launch the Skyrim Special Edition Creation Kit.

Go to File > Data.

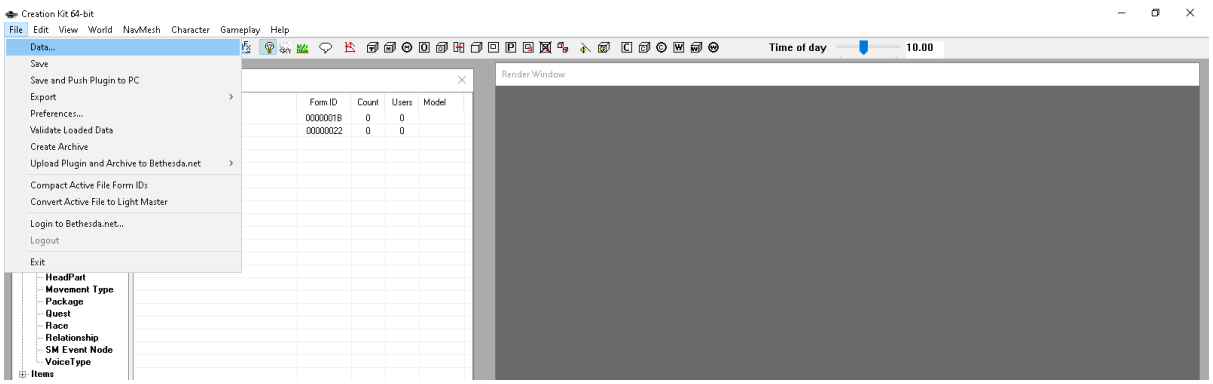


Figure 1527 - The Special Edition Creation Kit.

Tick your mod in the Plugin/Master Files list, click 'Set as Active File' and click OK.

Note: If you used xEdit to set the ESM flag in your mod already, you'll need to remove it first, convert your mod using the following steps, and then you can add it back.

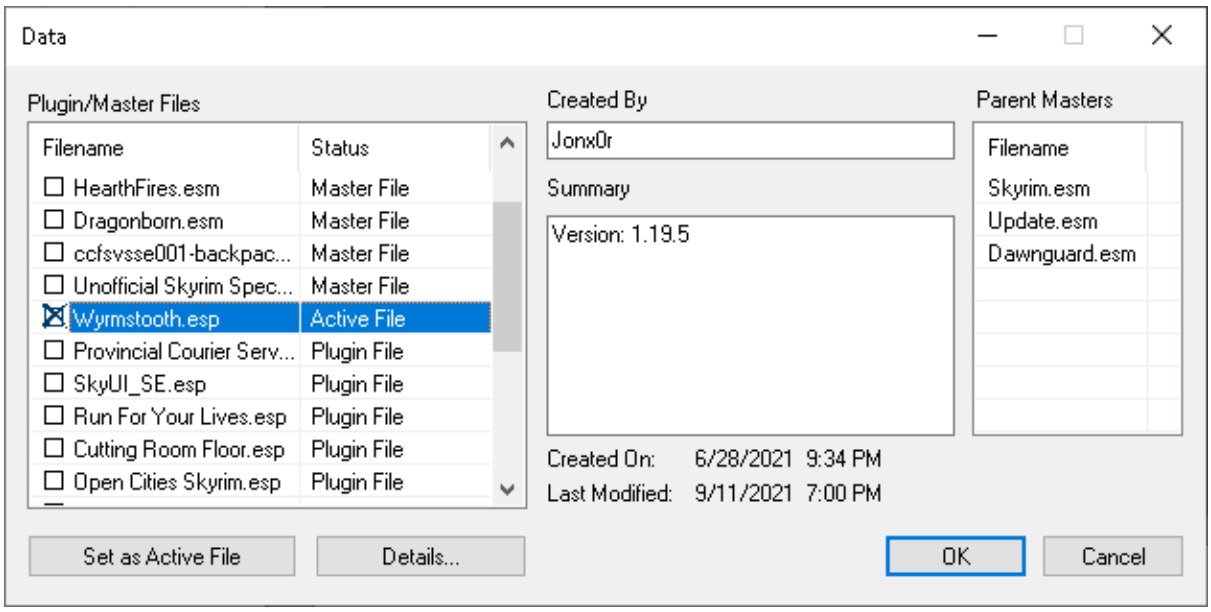


Figure 1528 - Loading your mod.

Let the Creation Kit load your mod.

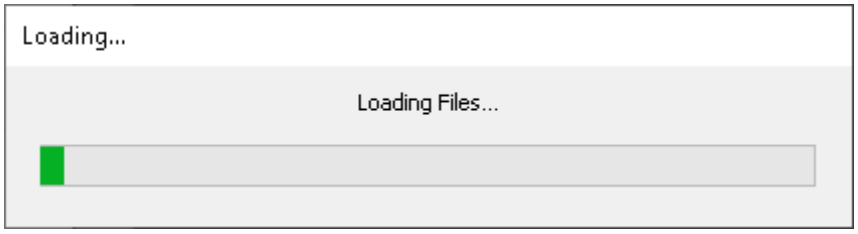


Figure 1529 - Loading progress.

Click 'Yes to All' to the countless warning pop-ups.

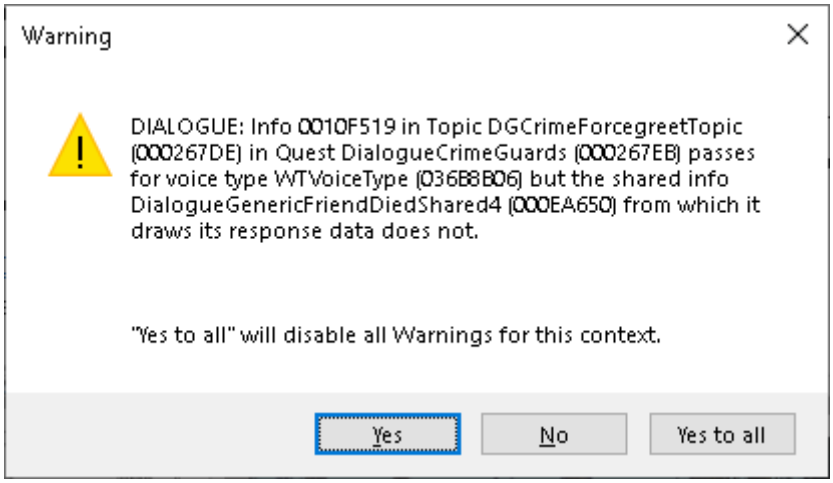


Figure 1530 - Load warnings.

Once the mod loads, take note of the warnings that appear. Most of the time they'll be related to issues in the dependencies.

Some warnings can be ignored, like the one highlighted in the screenshot below.

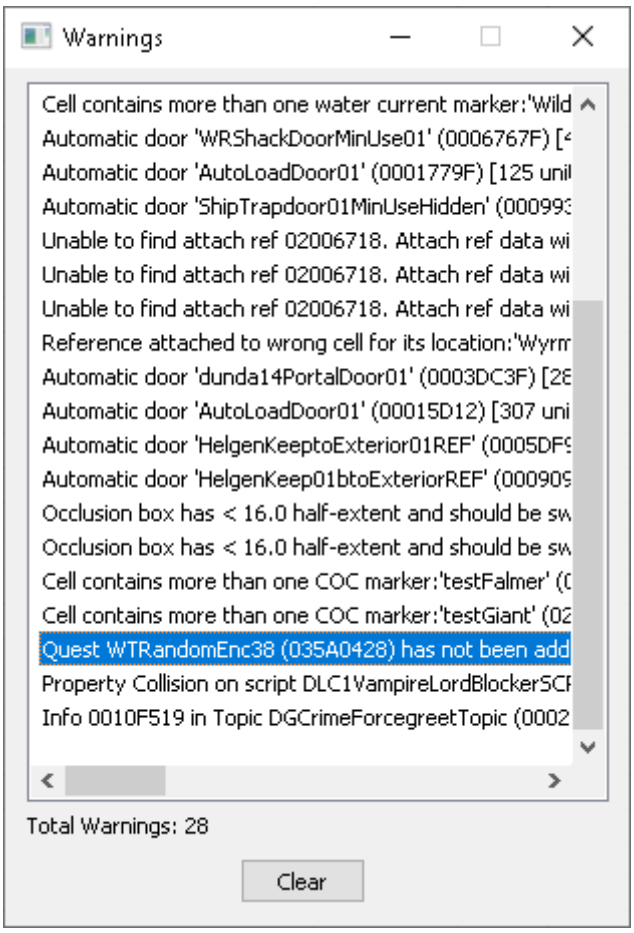


Figure 1531 - Warnings.

Click on the Save button to save your mod.

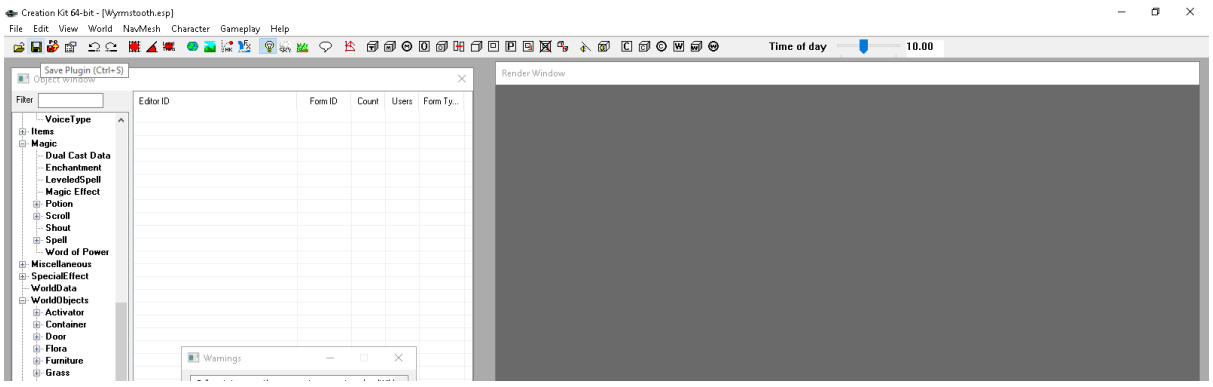


Figure 1532 - Saving the mod.

Now we need to regenerate the facegen data.

You can do this by selecting your NPCs in the Object Window under Actors > Actor, then press CTRL + F4.

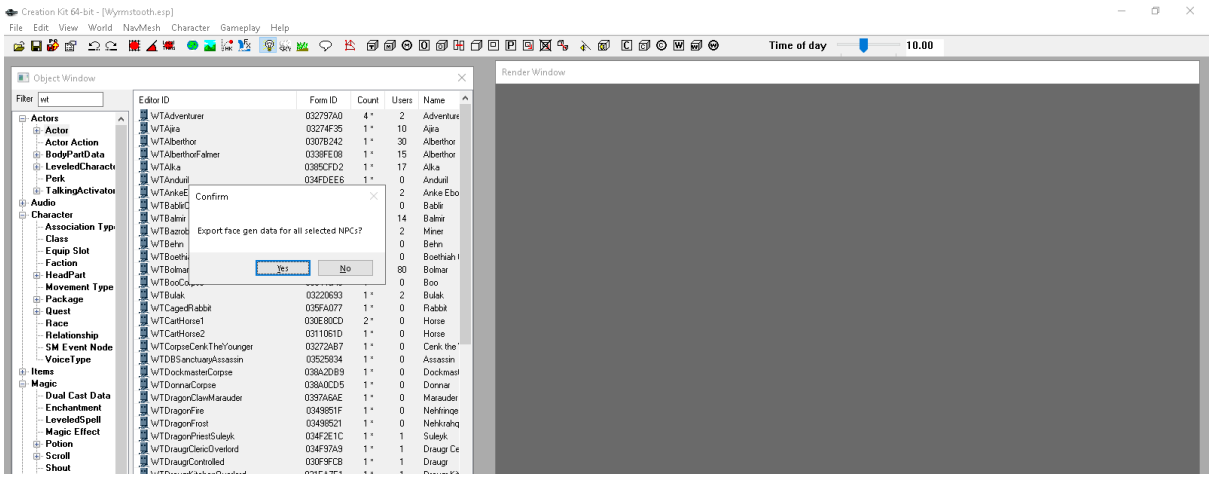


Figure 1533 - Regenerating facegen data.

Click Yes to the confirmation pop-up.

Next we need to convert our .nif files to the SSE format. Before doing this, copy your mod's files from your Skyrim\Data folder to your Skyrim Special Edition\Data folder.

We convert the .nifs automatically using the SSE NIF Optimizer tool available on Nexusmods.

Run SSE NIF Optimizer.exe.

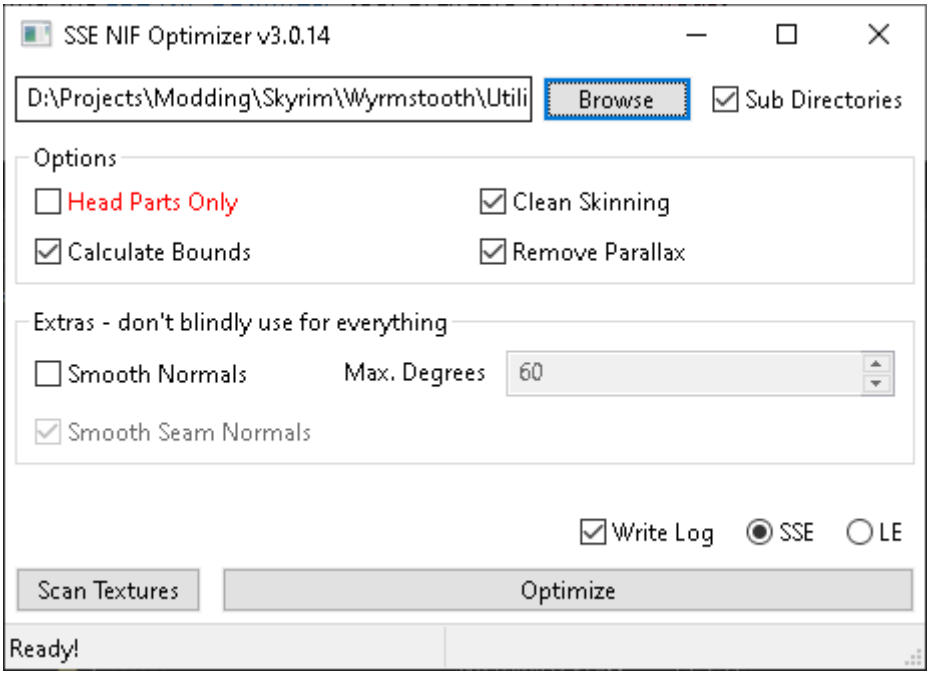


Figure 1534 - SSE NIF Optimizer.

Click on Browse.

Navigate to your Skyrim Special Edition\Data\Meshes folder.

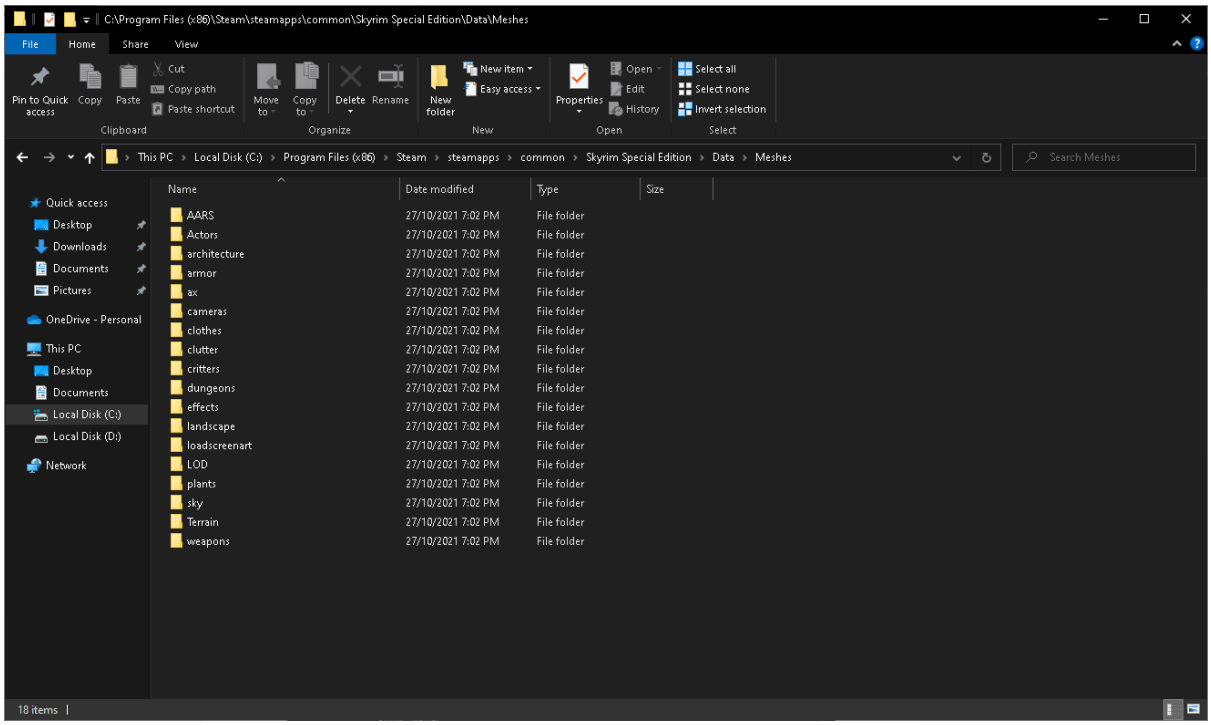


Figure 1535 - Skyrim Special Edition meshes folder.

I would recommend selecting one folder at a time, rather than selecting the entire meshes folder and running the conversion process on all subfolders.

Important: The .nif files under the Skyrim Special Edition\Data\Meshes\Actors\Character\FaceGenData\FaceGeom\<YourModName>.esp folder shouldn't be converted using this tool.

There are also no .nif files under Skyrim Special Edition\Data\Meshes\Terrain to convert, so you don't need to run the SSE NIF Optimizer for that folder.

Click Optimize.

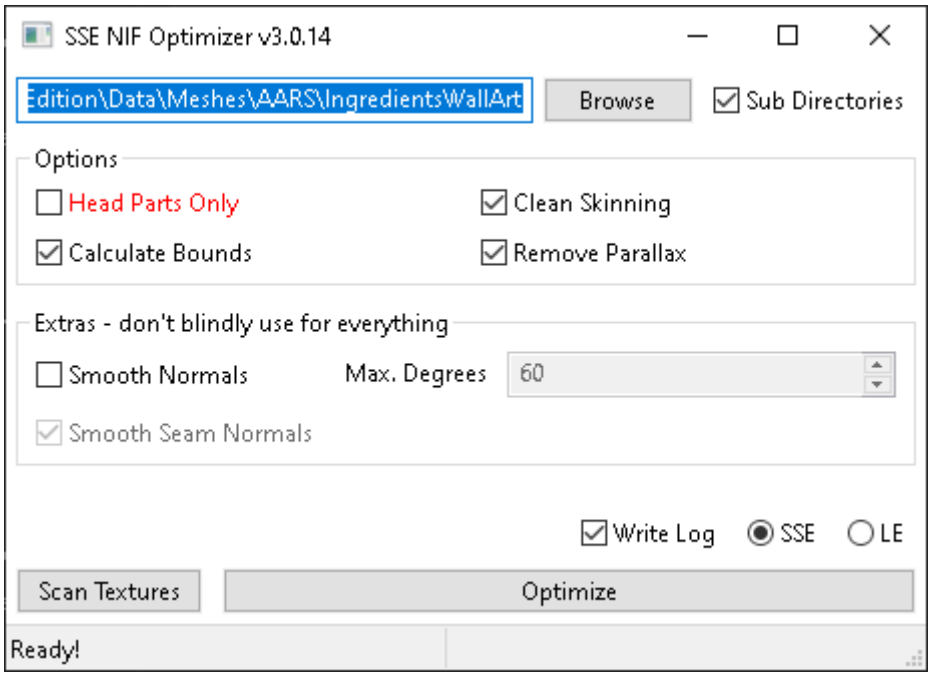


Figure 1536 - Optimizing.

Now you can go ahead and Package your mod for Skyrim Special Edition.

PACKAGING YOUR MOD FOR LEGENDARY EDITION

This section will cover packaging your mod using Bethesda's archive utility.

You can find archive.exe in your Skyrim folder.

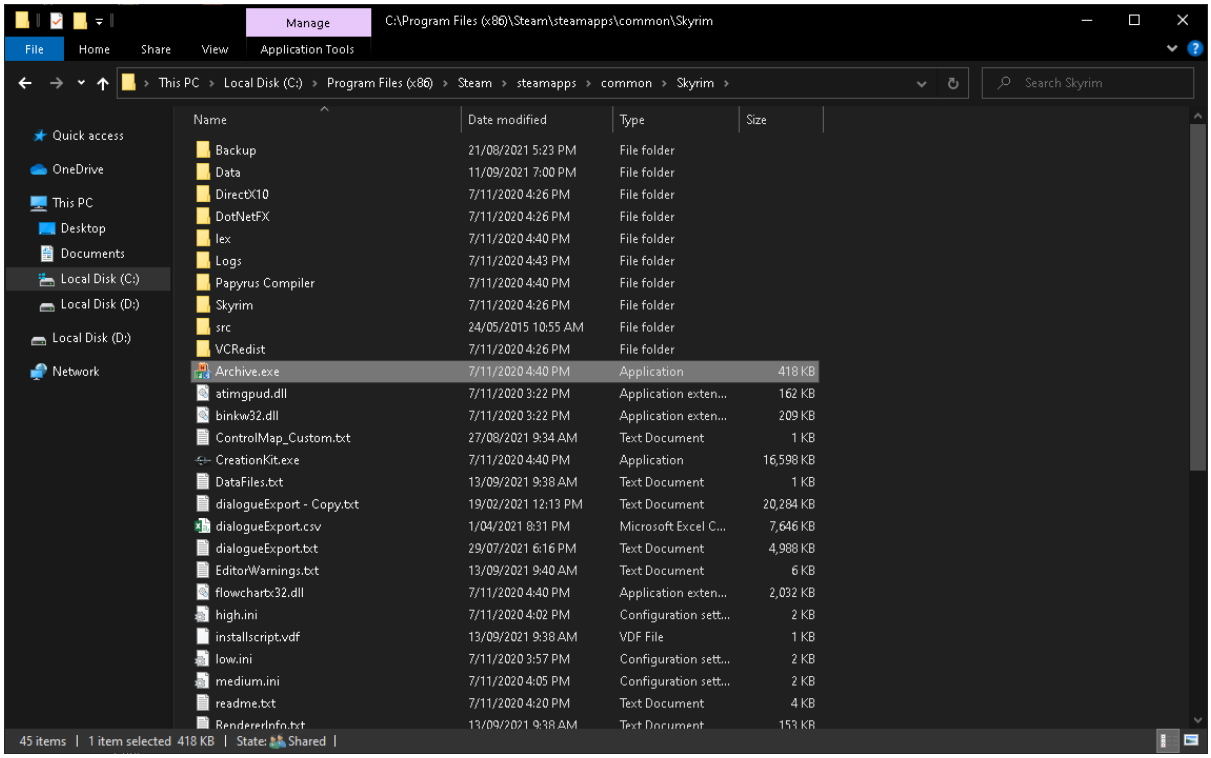


Figure 1537 - Bethesda's archive utility.

Launch archive.exe.



Figure 1538 - TESArchive.

Go to Edit > Add Directory.

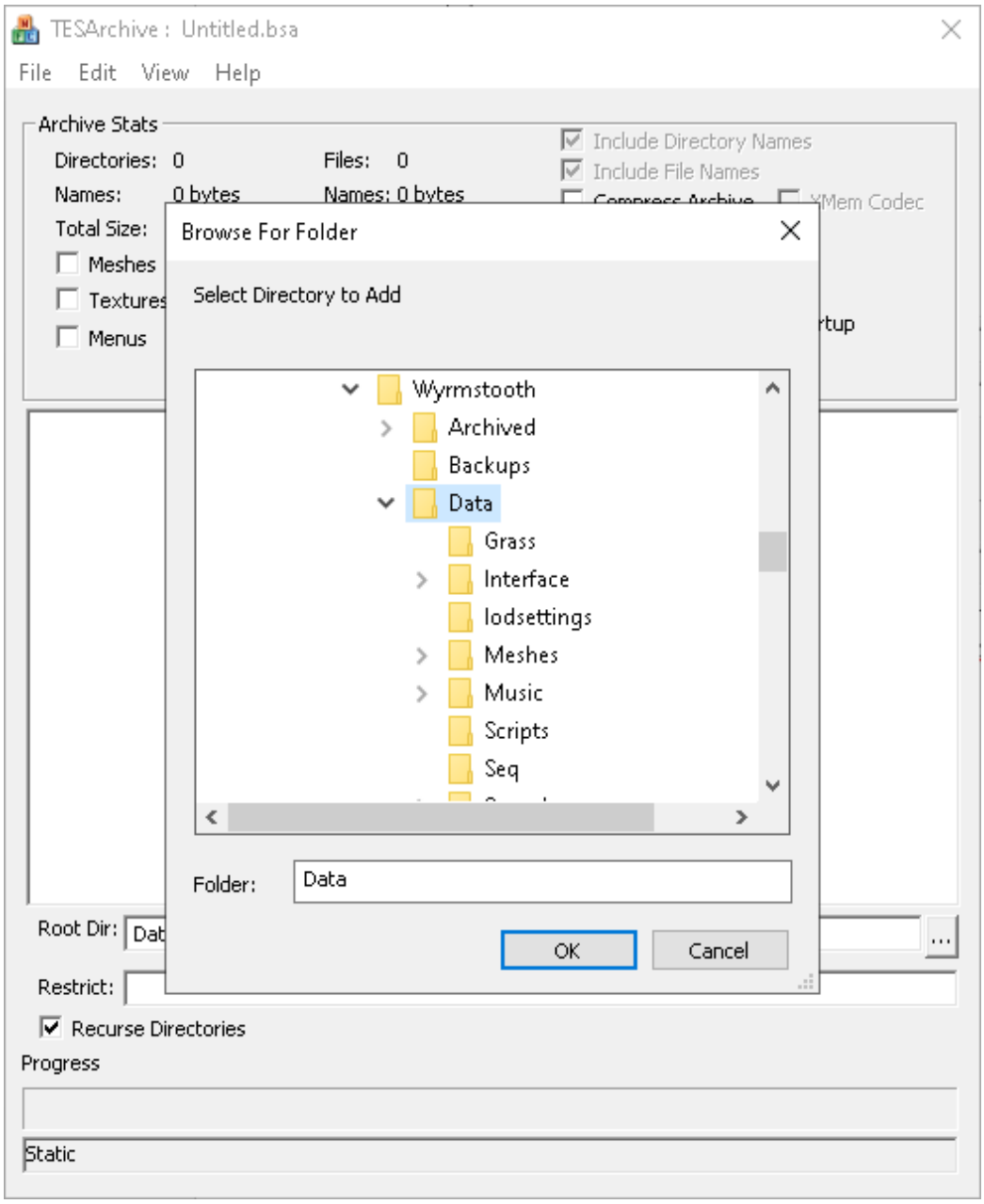


Figure 1539 - Adding your mod assets folder.

Select your mod assets folder. This folder should be named Data and should contain all the assets relating to your mod and ONLY the assets relating to your mod.

Click OK.

Go to Edit > Check All Items to select all assets.

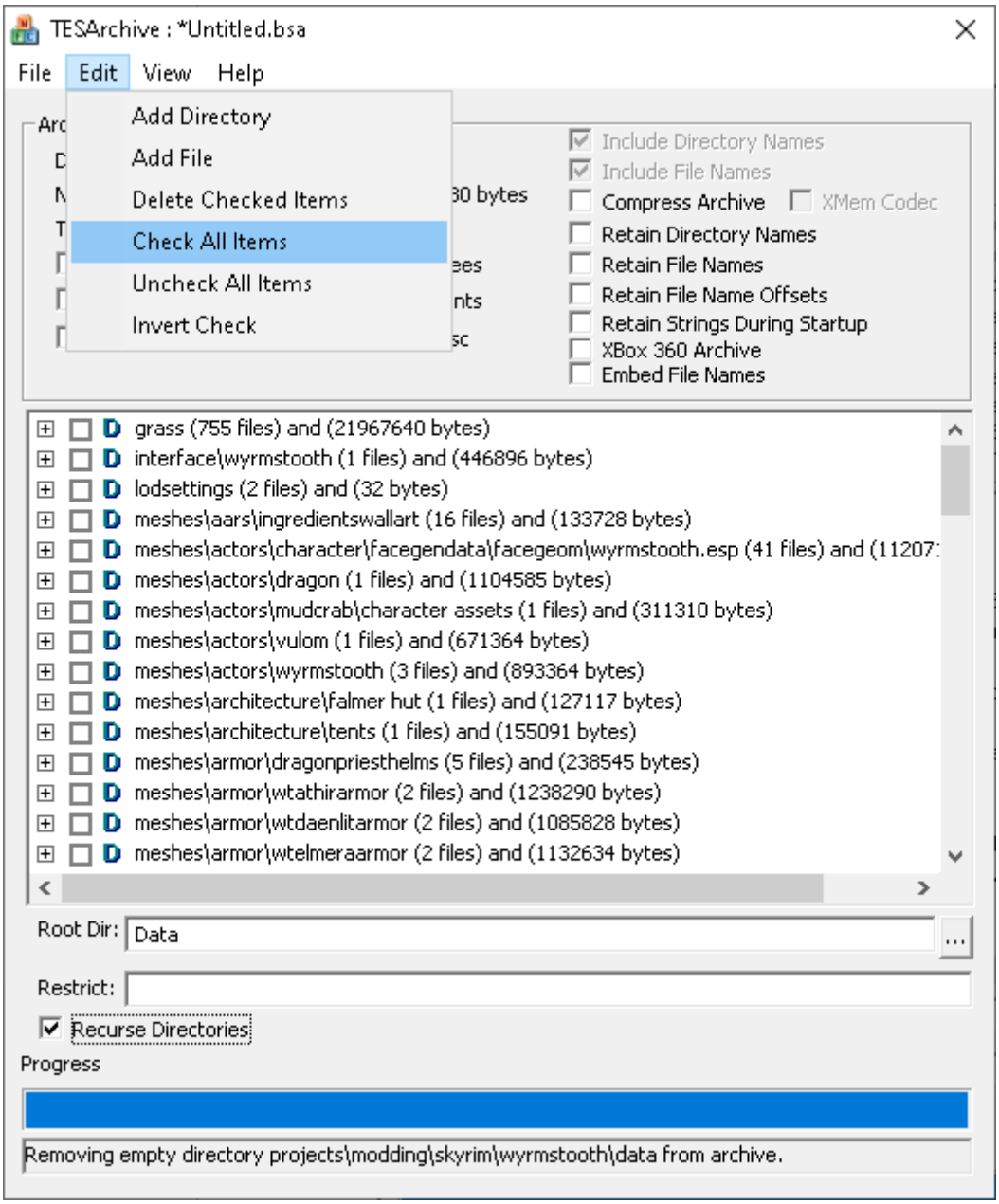


Figure 1540 - Selecting all assets to package.

Under Archive Stats, tick the boxes relating to assets included in your mod. Wyrmstooth, for example, includes Meshes, Textures, Sounds, Voices and some other Misc assets, so I ticked those boxes.

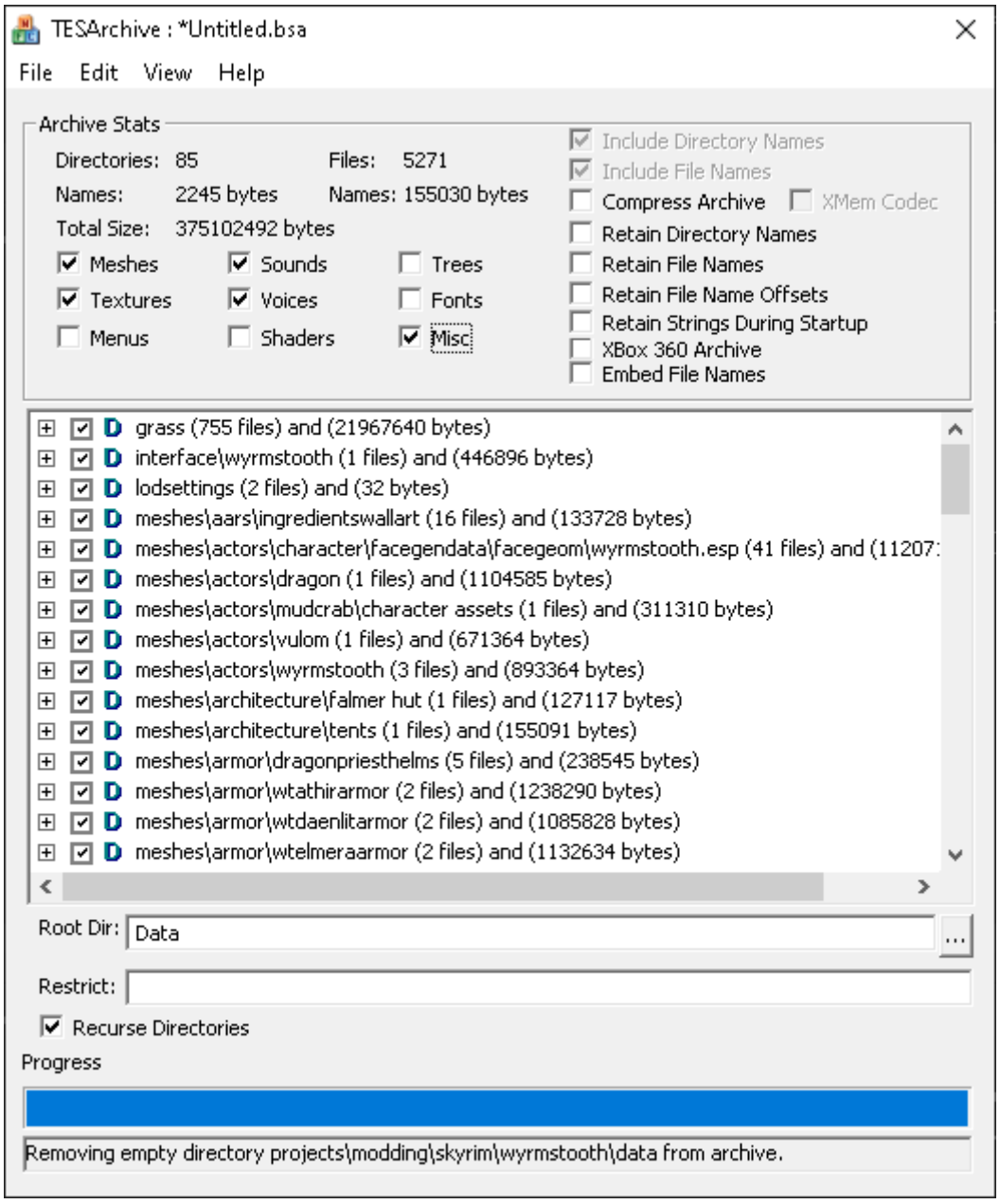


Figure 1541 - Setting the archive type.

Go to File > Save As.

Make sure the name of your .bsa archive matches the name of your .esp file.

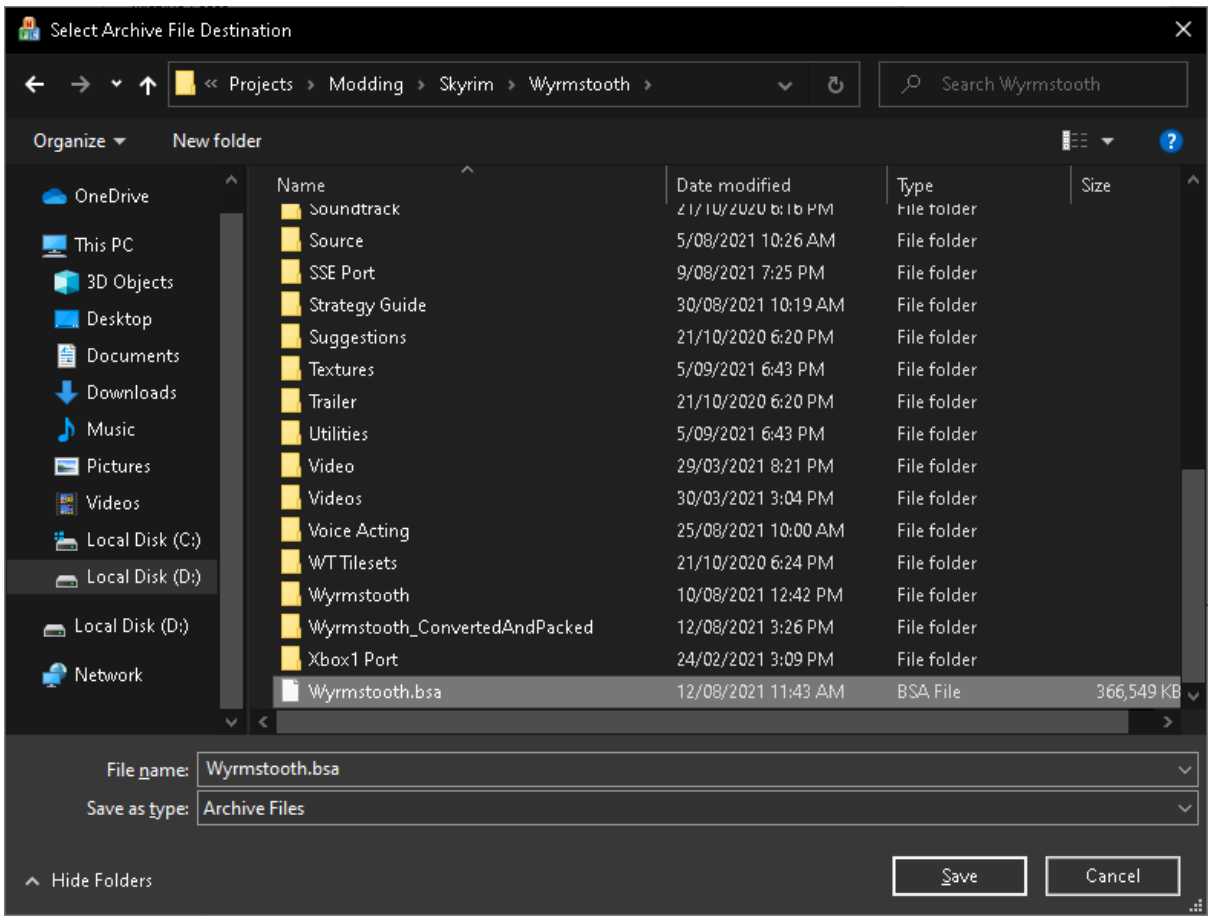


Figure 1542 - Setting a location to save the new archive to.

Click Save to begin packing the archive.

The current process is done when the current task is left at 'Renaming temporary archive'.

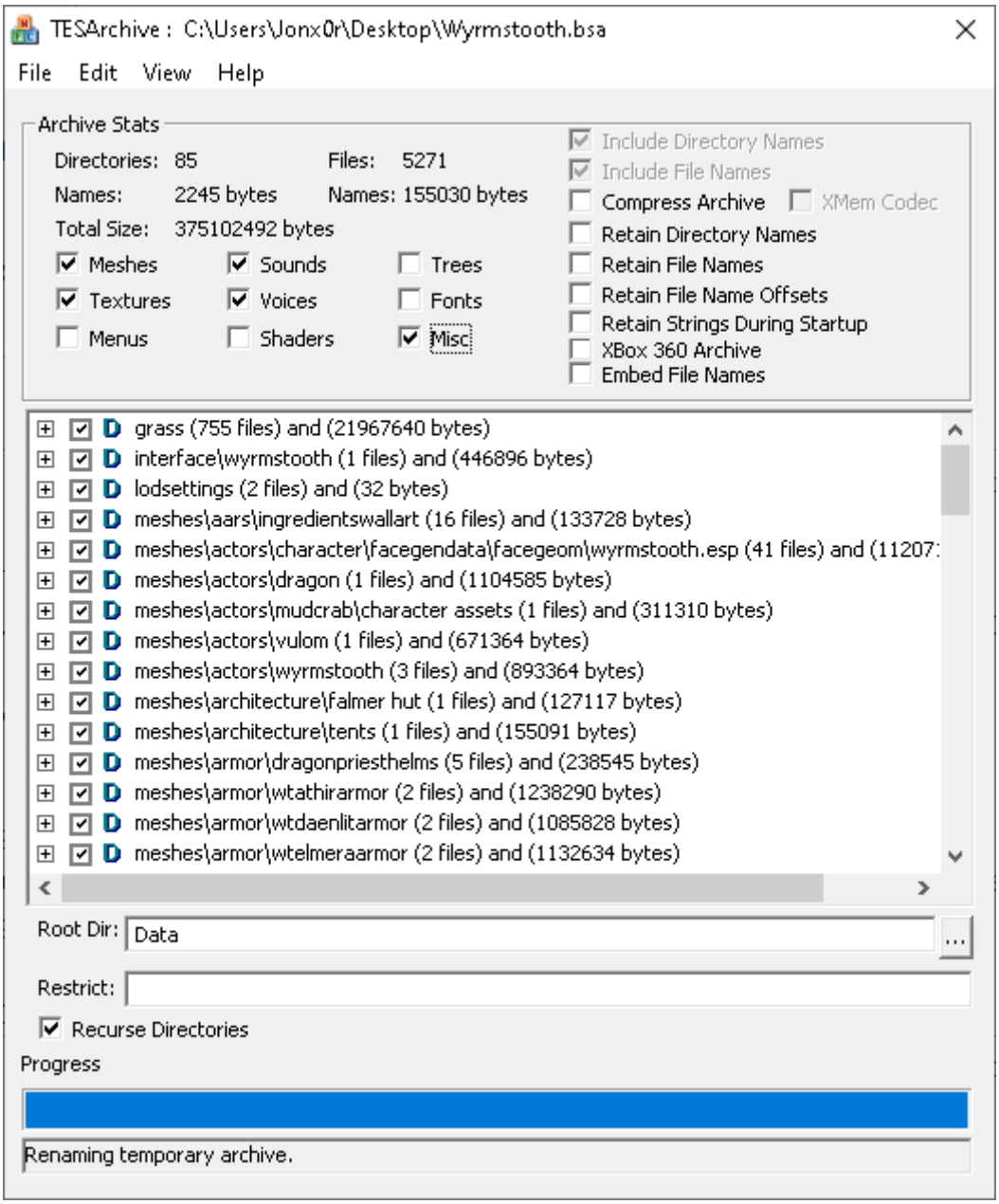


Figure 1543 - Done.

You should have a new .bsa and .bsl file.

I usually copy these files to my Skyrim\Data folder as well.

PACKAGING YOUR MOD FOR SPECIAL EDITION

This section will cover the packaging your mod in the Skyrim Special Edition version of the Creation Kit.

Before launching the Creation Kit, if you already previously packaged a .bsa archive for your mod, you will need to remove it from the Skyrim Special Edition\Data folder first. If you don't, the Creation Kit will hang when trying to make a new archive.

Launch the Creation Kit.

Go to File > Data.

Tick your mod in the Plugin/Master Files list. If your plugin isn't a master file, click 'Set as Active File'.

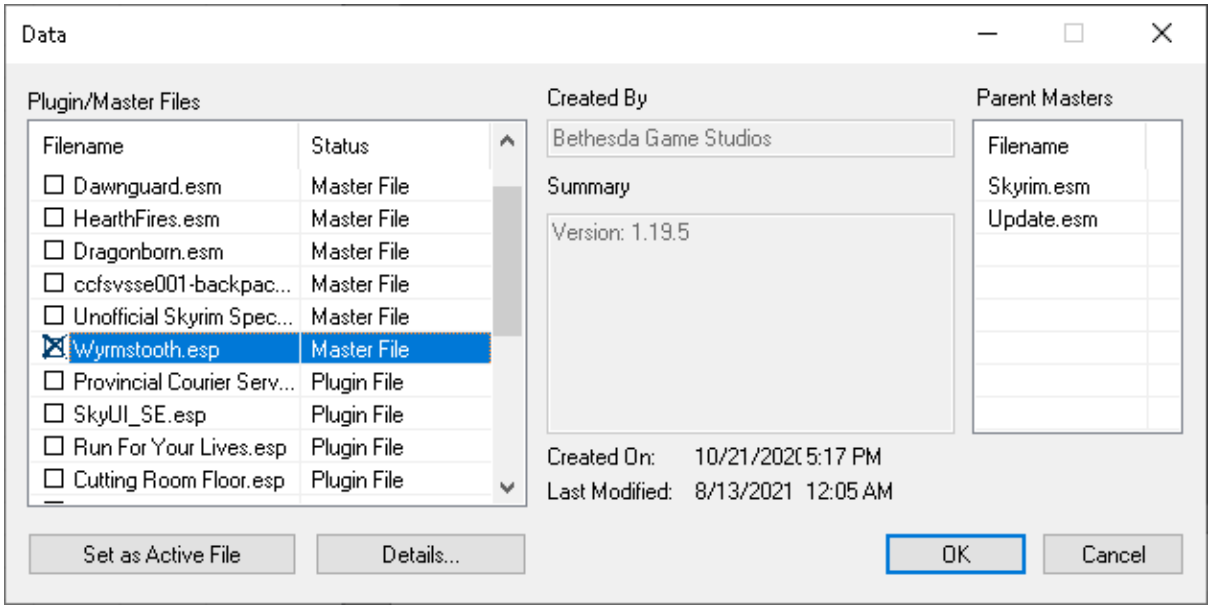


Figure 1544 - Loading your mod.

Click OK to begin loading your mod.

Once the Creation Kit is done loading your mod, go to File > Create Archive.

If your loaded mod is a master file, this list should appear straight away and should be empty.

If your loaded mod isn't a master file, it may take a while for the Creation Kit to search through your mod and automatically detect assets.

If any files are listed here, select them all and remove them as we'll be adding them in manually.

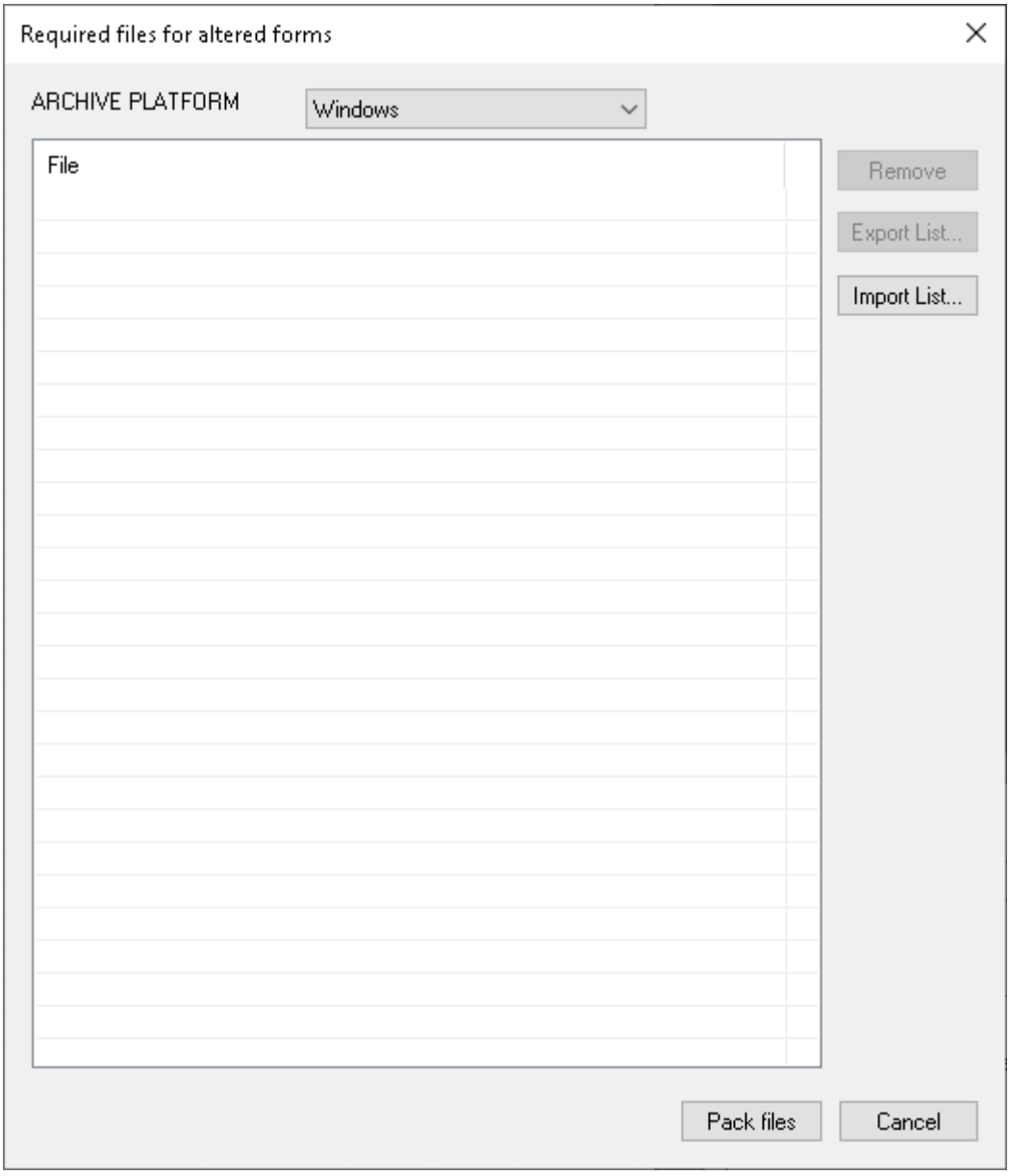


Figure 1545 - Asset list.

To add in your assets to the archive, drag and drop them in from the Skyrim Special Edition\Data folder.

You will need to select the files themselves to drag and drop into the Creation Kit, not the folders or subfolders.

Do not drag and drop in files from a separate project directory. These files need to come from the Skyrim Special Edition\Data folder.

Do not include assets that are not part of your mod.

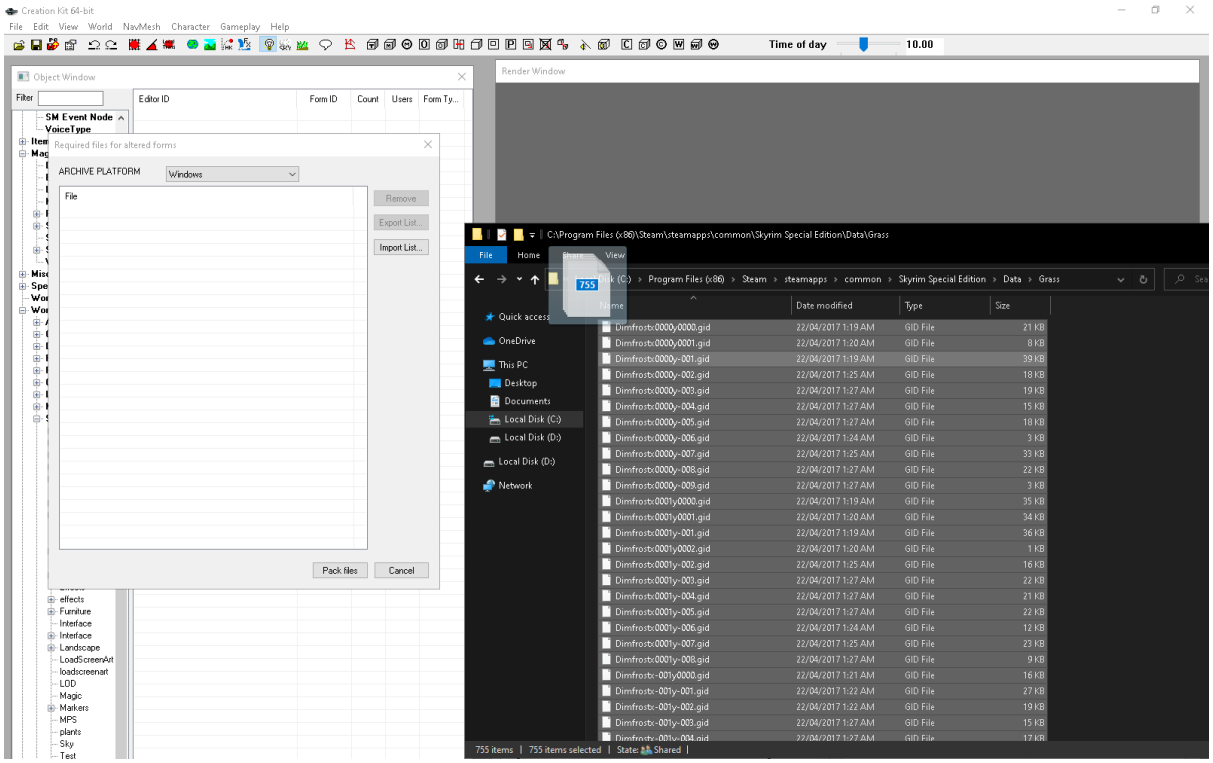


Figure 1546 - Dragging in the .gid grass files into the asset list.

Assets dragged and dropped into the asset list should appear as per the following screenshot:

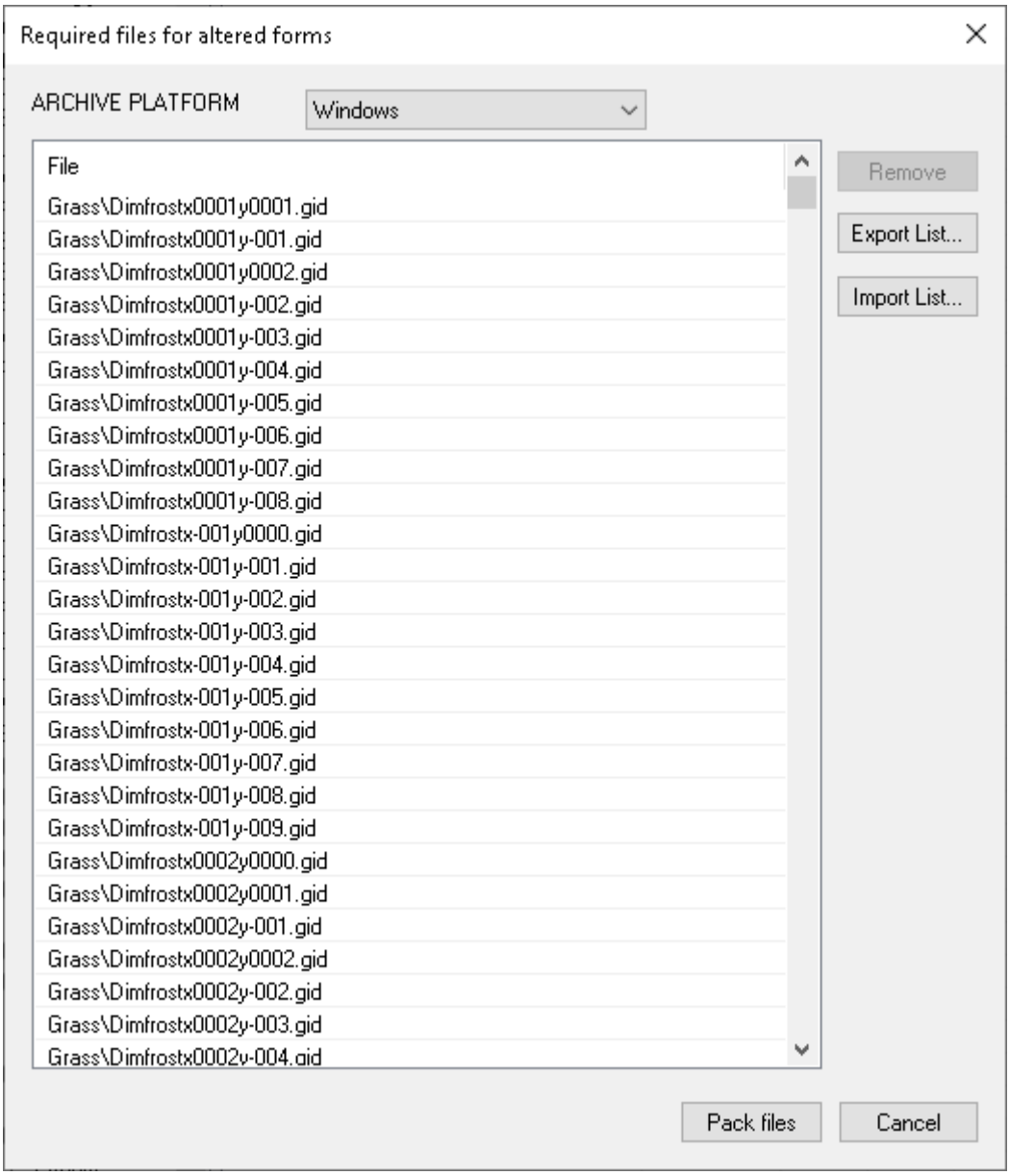


Figure 1547 - .gid grass files added to the asset list.

To make things easier, I only keep loose files belonging to the current mod I'm working on in the Skyrim Special Edition\Data subfolders.

This is so I can use wildcard search terms like *.* to select all files in all subdirectories at once.

In the screenshot below, I searched for '*.*' in the Skyrim Special Edition\Data\Sound folder to select all ambient sounds and voice files at once.

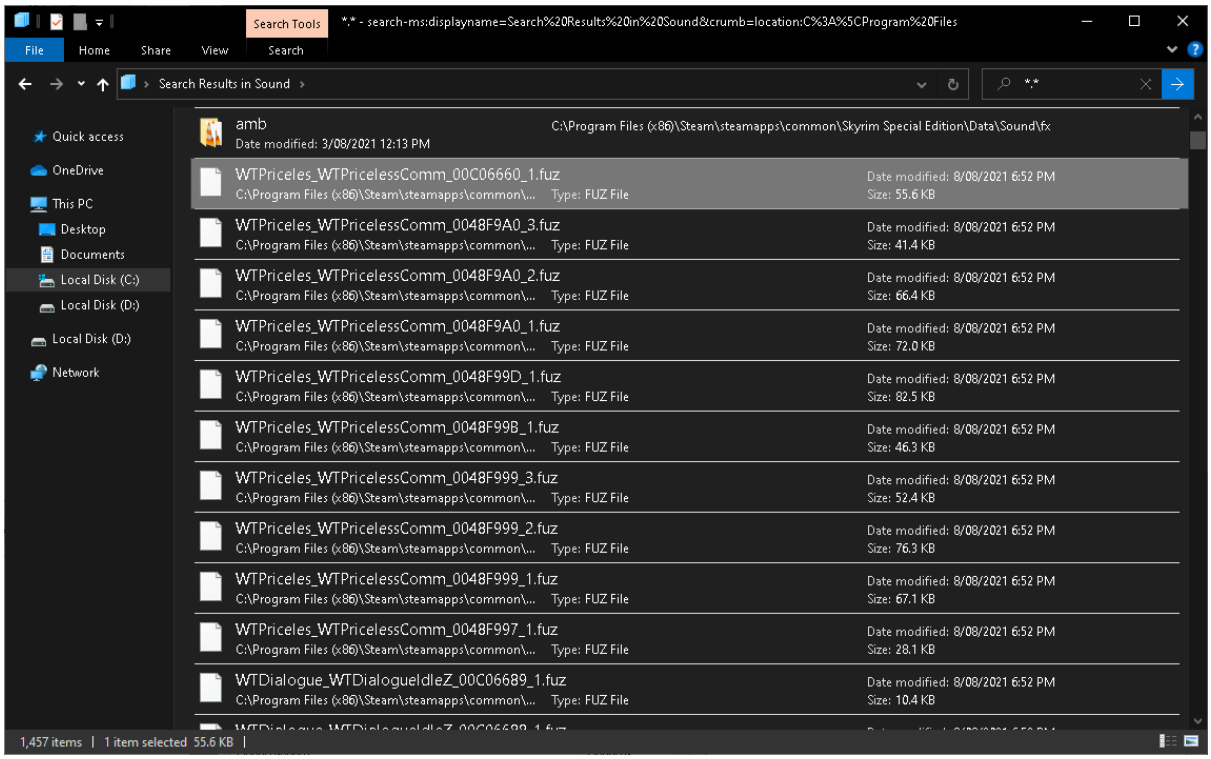


Figure 1548 - Using wildcard search terms.

For texture files, search for '*.dds' to ensure that only .dds files are being added to the list.

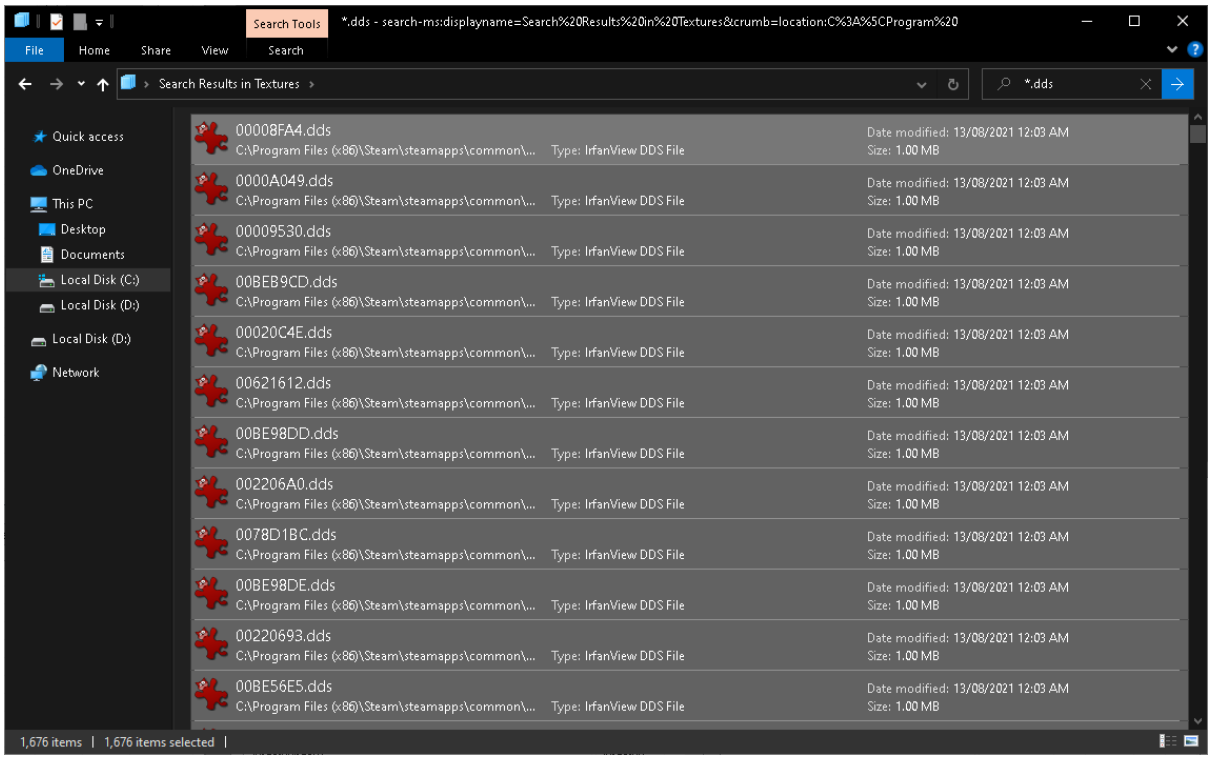


Figure 1549 - Searching by a specific file extension.

Once you've added all assets to the asset list, click on the 'Pack files' button.

You will be prompted to set the name of your main archive first.

Ensure the name of your .bsa archive matches the name of your .esp file.

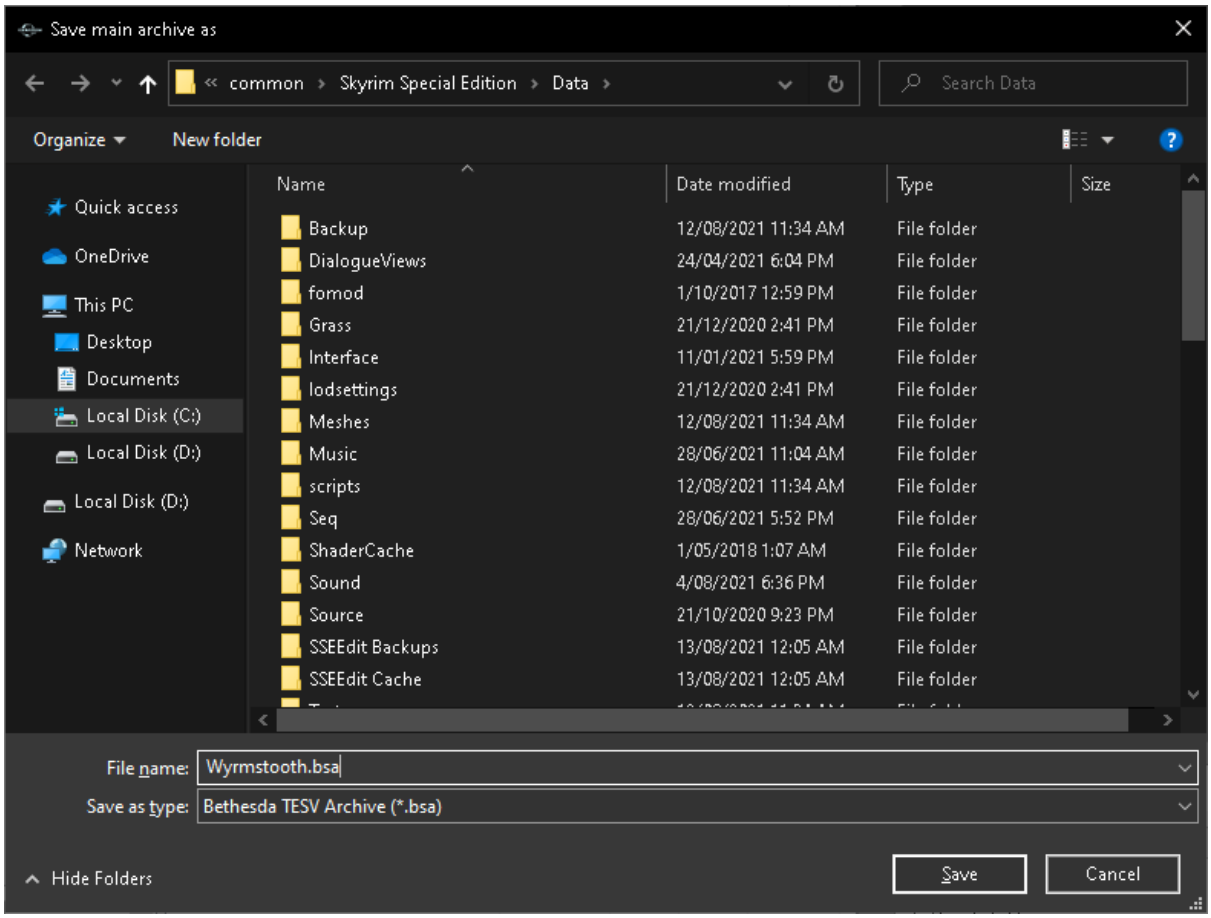


Figure 1550 - Setting the name of the main archive.

Click Save.

If your mod includes textures, you'll need to set the name of the textures archive as well.

This should match the name of your .esp file with ' - Textures' appended to it.

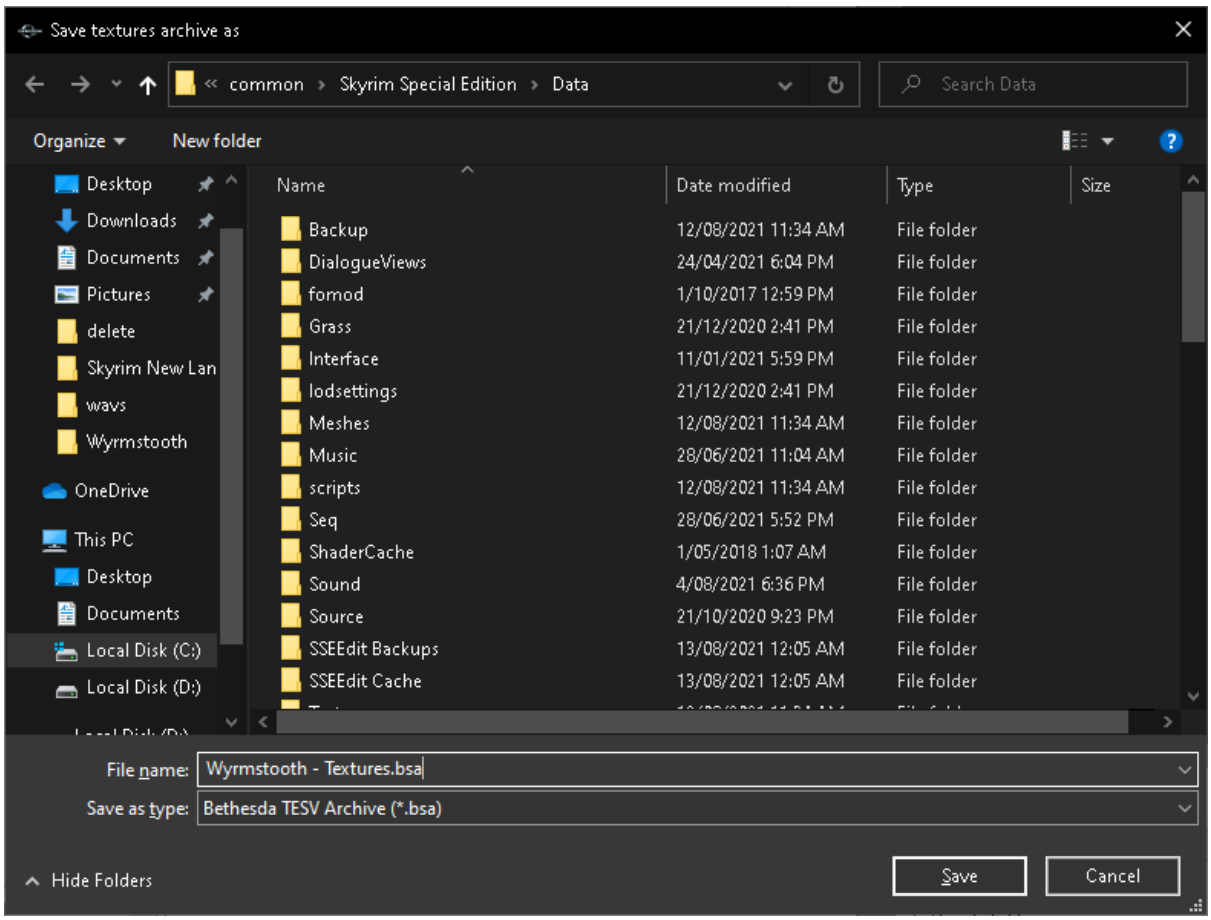


Figure 1551 - Setting the name of the textures archive.

Click Save.

Let the conversion and packing process complete.

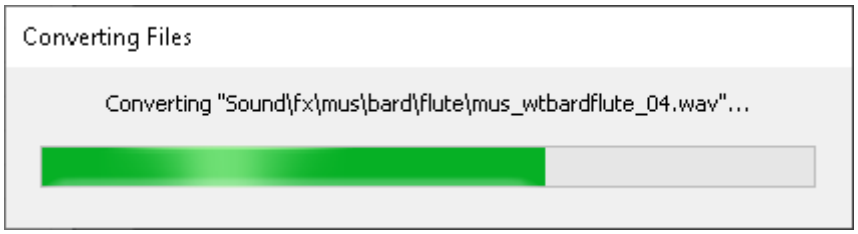


Figure 1552 - Converting files for packing.

Don't touch anything, just let it continue automatically.

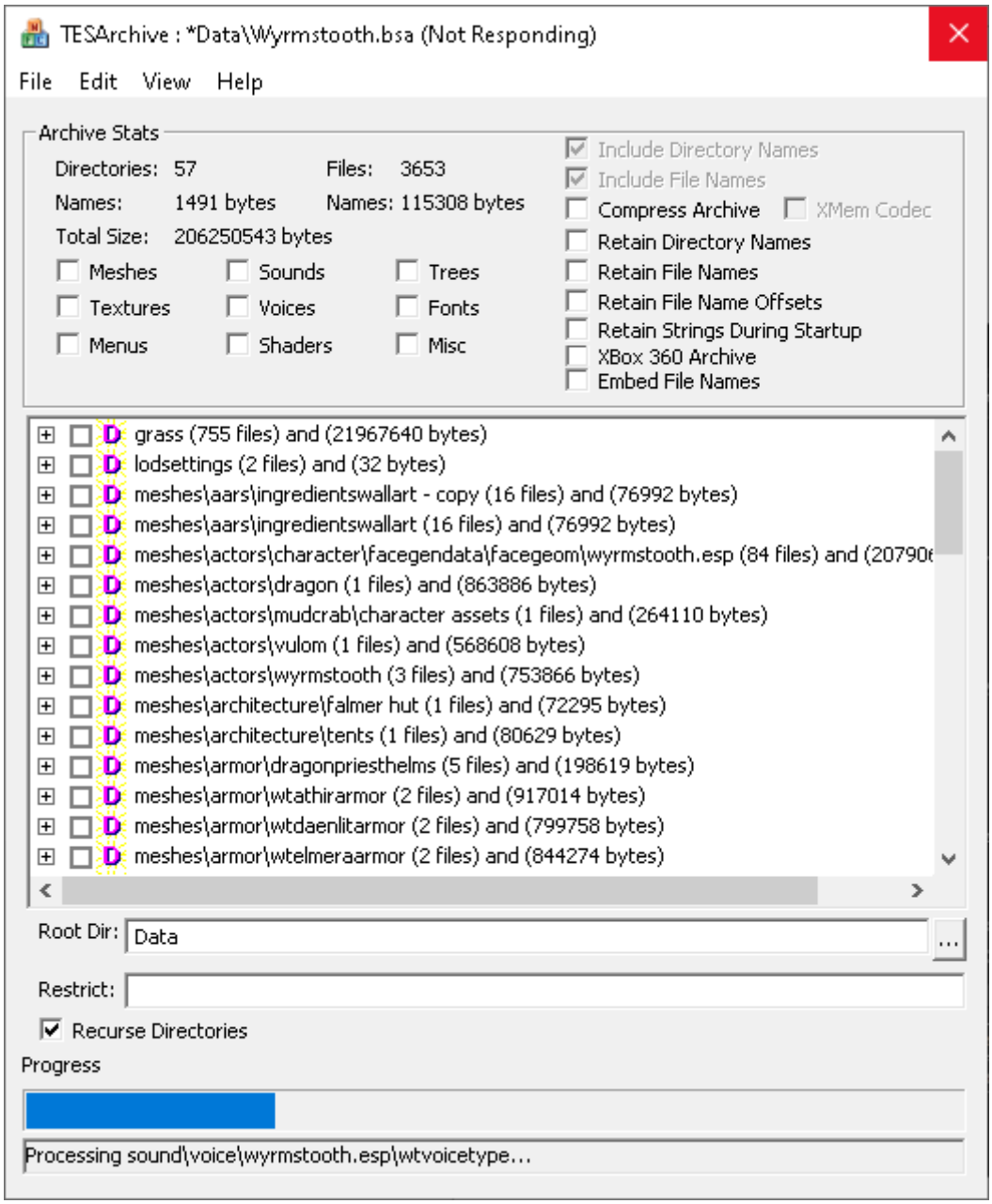


Figure 1553 - Packing the archive.

When it's done, your .bsa archive(s) should appear in the Skyrim Special Edition\Data folder.

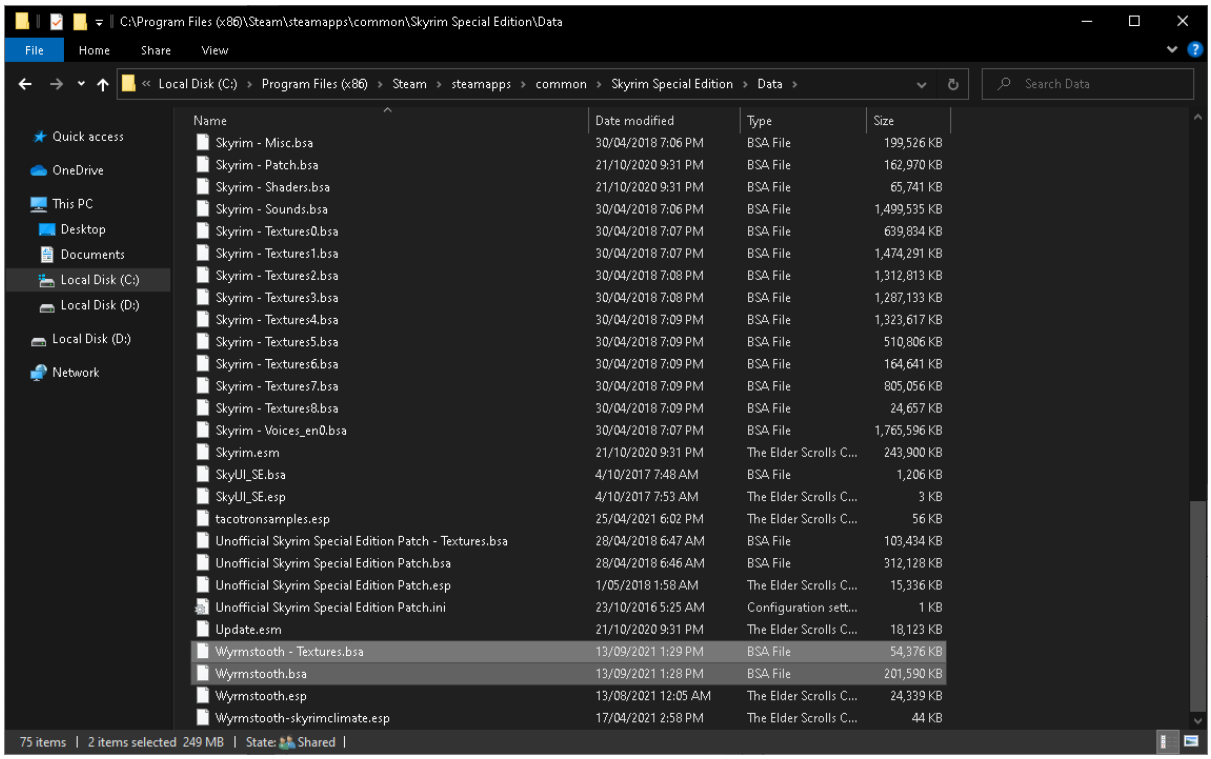


Figure 1554 - The .bsa archives are ready.

PACKAGING YOUR MOD FOR NINTENDO SWITCH

This section will cover packaging your mod using the NX Toolkit which is available on Nexusmods.

Download the NX Toolkit and extract it to a new folder.

Create a new folder in your projects directory and name it NX Port. Copy the .bsa archives and the .esp/.esm file from the SSE version of your mod into this folder.

If you created a .bik video, create a new folder called Video and copy the .bik video into it.

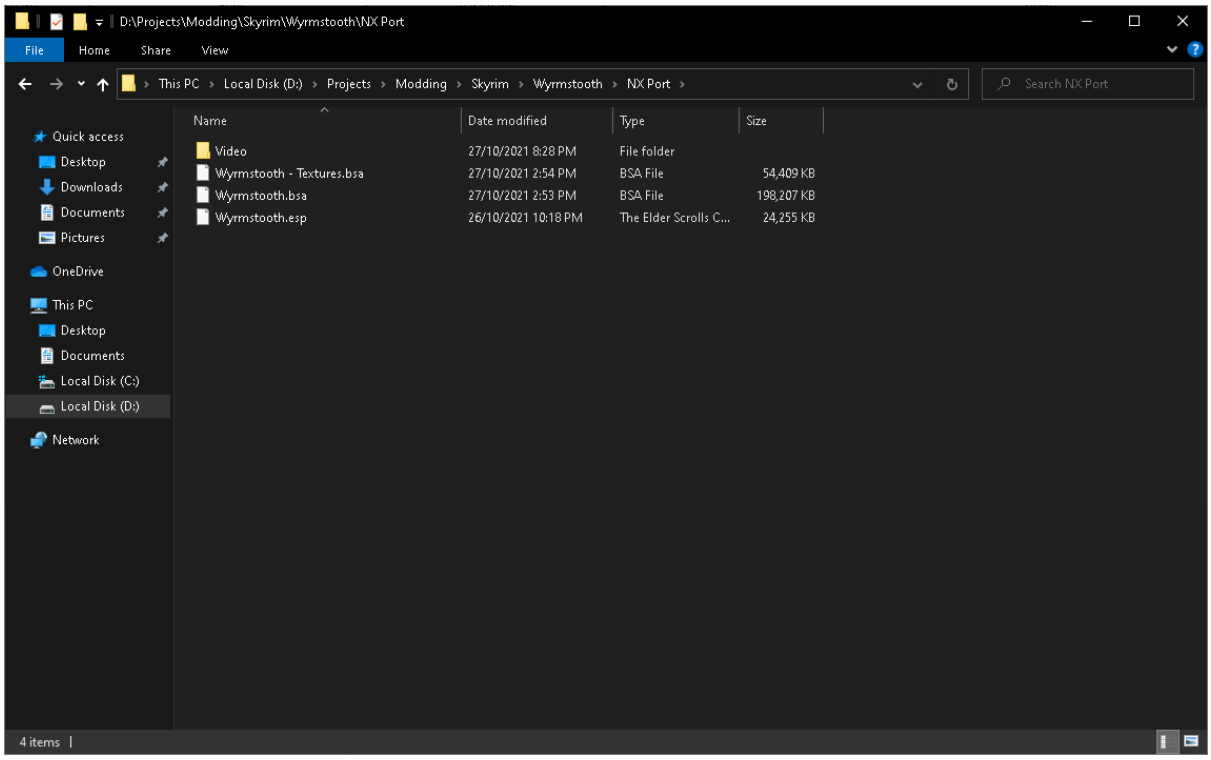


Figure 1555- NX Port folder.

Open a command prompt and navigate to the folder you extracted the NX Toolkit to.

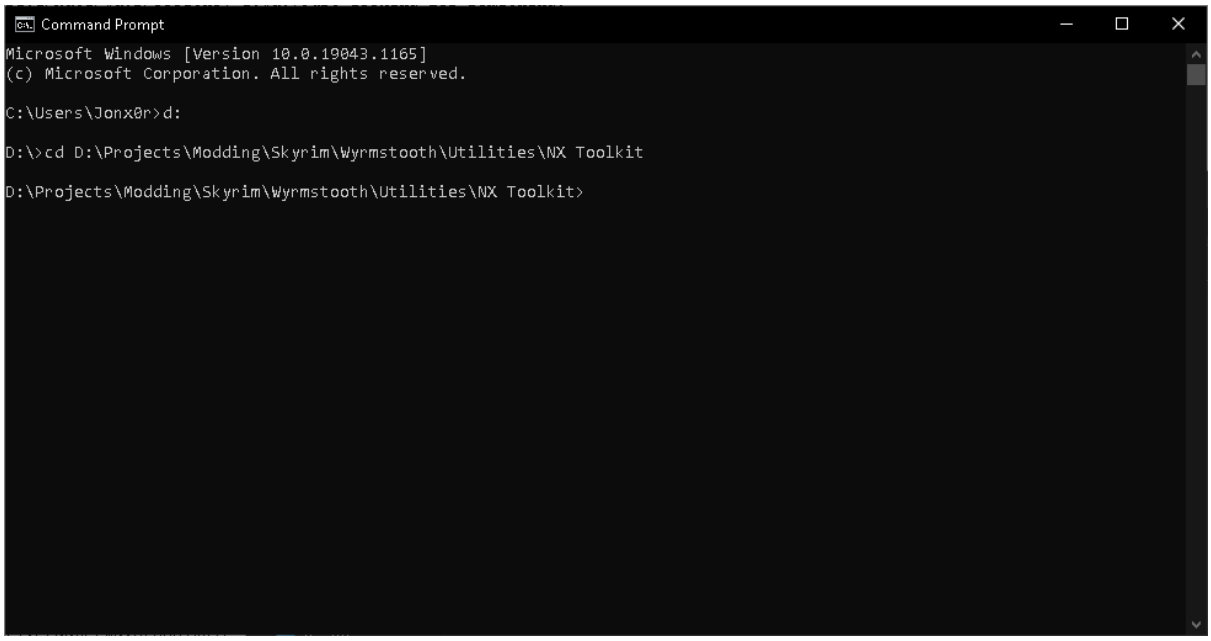


Figure 1556 - Folder containing the NX Toolkit.

Run the following command: CONVERT_MOD.BAT "<PathToYourNXPortFolder>"

For example: CONVERT_MOD.BAT "D:\Projects\Modding\Skyrim\Wyrmstooth\NX Port"

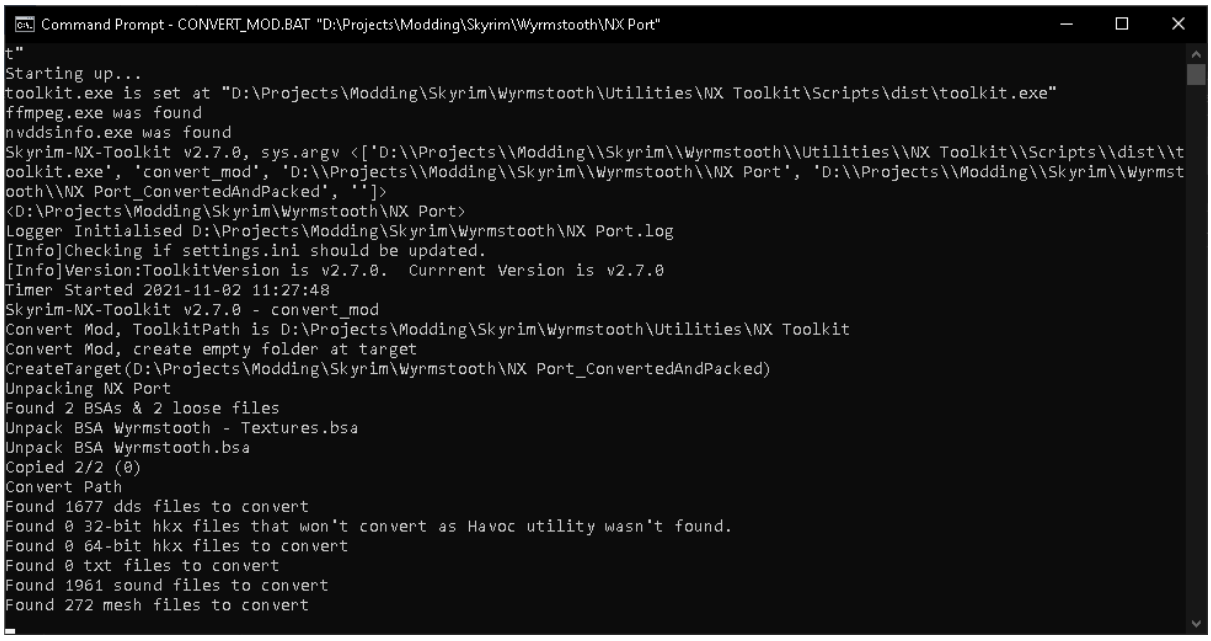


Figure 1557 - Starting up the conversion process.

The converted files for you to distribute will appear under a folder called 'NX Port_ConvertedAndPacked'.

This process may take a while to complete. Just let it run until it's finished.

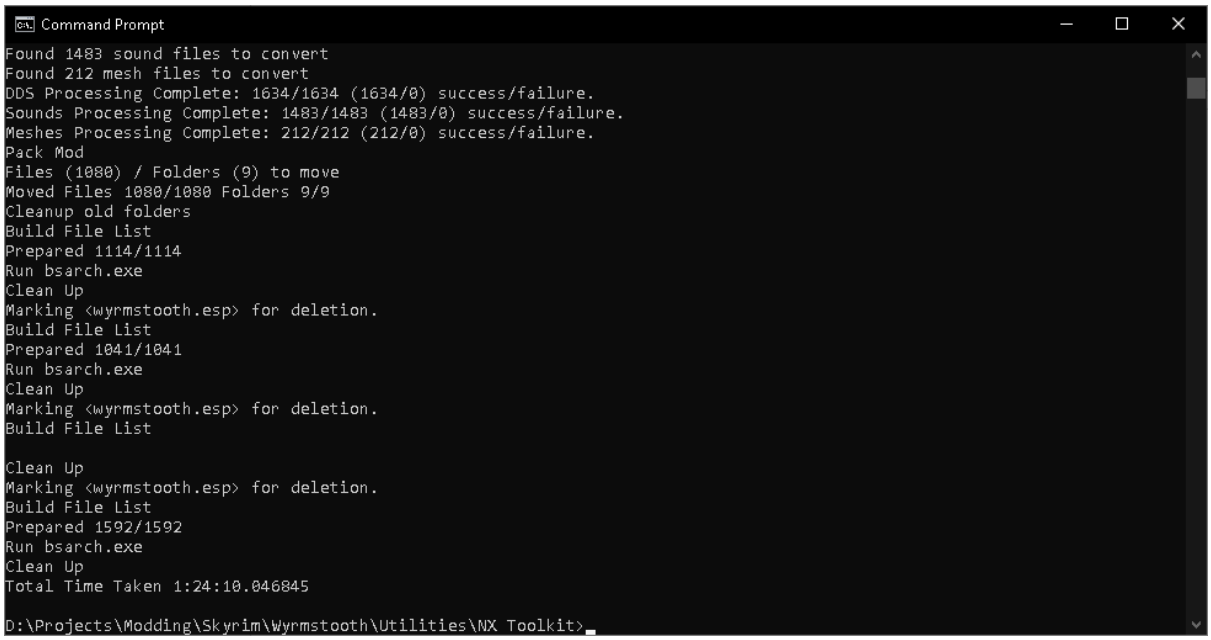


Figure 1558 - Conversion complete.

The NX Port_ConvertedAndPacked folder should contain new .bsa archives and your .esp/.esm file ready for upload.

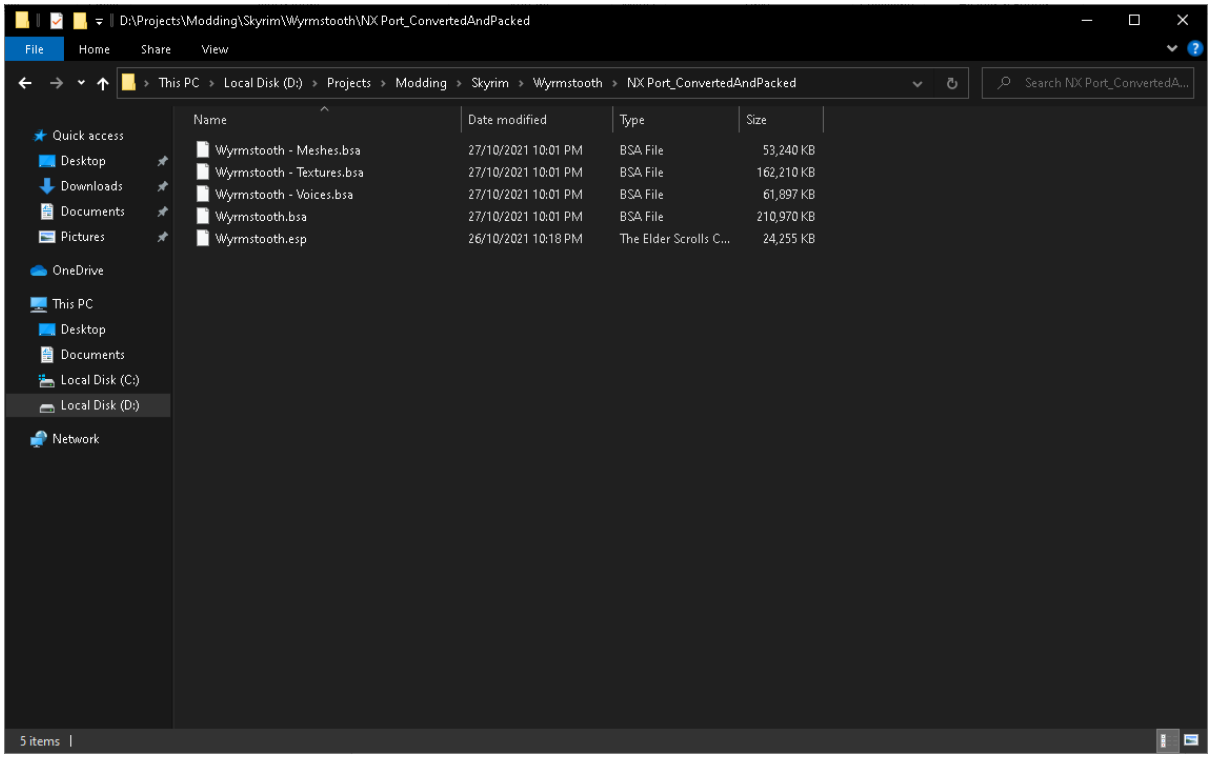


Figure 1559 - Converted and packed for switch.

If you need to port your mod from Legendary Edition first, see the section on Porting your mod to Skyrim Special Edition.

INDEX

3ds Max, 789, 790, 791, 796, 800, 810, 827, 828, 830, 831, 832, 840, 842, 844, 851, 856, 857, 858, 860, 892, 901, 909, 910, 916, 918, 919, 925, 928, 929, 937, 940, 966, 977, 1068, 1071
After Effects, 1139, 1140, 1141
Audacity, 297, 298, 1046, 1048, 1049
BAE, 928, 1006, 1007, 1008, 1058, 1059, 1060
Bethesda.net, 2
Crash, 8, 55, 62, 99, 310, 325, 331, 373, 919, 925, 1120
Creation Kit, 8, 11, 12, 14, 38, 46, 48, 55, 62, 91, 99, 122, 131, 182, 213, 305, 310, 320, 325, 327, 331, 340, 375, 377, 533, 535, 537, 576, 614, 641, 649, 675, 788, 811, 817, 819, 910, 987, 995, 996, 1062, 1063, 1114, 1120, 1144, 1146, 1147, 1159, 1160, 1161
Facegen, 372, 1149
Havok Content Tools, 928, 975
hkxcmd, 934, 935, 936, 938
Nexusmods, 2, 322, 631, 764, 776, 782, 789, 827, 910, 928, 934, 1006, 1009, 1058, 1128, 1149, 1169
NifSkope, 789, 801, 802, 827, 843, 848, 919, 920
njob, 83, 84, 87
NVIDIA Texture Tools Plugin, 74, 83, 90
NX Toolkit, 1169, 1170
Oscape, 764, 768, 771, 776, 782
Papyrus, 12, 339, 405, 407, 413, 434, 435, 436, 438, 440, 540, 561, 575, 579, 581, 586, 735, 1136, 1138, 1144
Photoshop, 74, 75, 76, 88, 845, 893, 897, 898, 899, 900
RAD Game Tools, 1139
Skyrim, 2, 8, 9, 14, 19, 38, 45, 48, 75, 85, 87, 90, 94, 112, 304, 307, 322, 326, 617, 619, 632, 751, 764, 765, 772, 773, 774, 779, 780, 782, 783, 786, 787, 788, 789, 792, 800, 810, 824, 825, 827, 829, 831, 832, 842, 853, 886, 890, 901, 902, 903, 909, 917, 918, 919, 928, 929, 934, 966, 977, 986, 990, 992, 995, 996, 1003, 1006, 1007, 1008, 1030, 1042, 1059, 1068, 1069, 1070, 1071, 1094, 1095, 1096, 1099, 1111, 1112, 1113, 1114, 1116, 1118, 1120, 1126, 1133, 1136, 1144, 1146, 1149, 1150, 1151, 1152, 1158, 1159, 1161, 1163, 1168, 1170, 1171
Skyrim Audio Converter, 322
SpeedTree, 831, 832, 834, 839, 840, 841, 842
SSE NIF Optimizer, 1149, 1150
SSEEdit, 776, 782, 819, 1128
Steam, 2, 8, 632
Substance Painter, 8, 851, 857, 883, 884, 887, 905, 907
Tacotron, 992, 993, 994, 995, 1012, 1022, 1027, 1028, 1029, 1030, 1033, 1034, 1039, 1040, 1041, 1043, 1045, 1046, 1049
Tensorboard, 1039, 1040, 1041
TES5Edit, 614, 776, 782, 819, 1128, 1129, 1130, 1133, 1134, 1135
TES5LODGen, 782
TESAnnwyn, 19, 45, 46, 47, 48, 49, 50
Unfuzer, 618, 631, 1009
WaveGlow, 992, 993, 995, 1012, 1022, 1029, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1045
World Machine, 14, 19, 38, 41, 44, 45, 46
Wyrmstooth, 2, 8, 12, 45, 128, 184, 192, 225, 226, 227, 231, 237, 323, 326, 336, 337, 339, 412, 434, 490, 505, 536, 537, 543, 546, 620, 644, 661, 674, 676, 680, 723, 752, 814, 815, 992, 1137, 1138, 1139, 1156, 1170
xEdit, 614, 776, 782, 784, 819, 1061, 1096, 1128, 1130, 1133, 1147
xLODGen, 819
Zbrush, 851, 856