

---

# **Atlas Devkit Documentation**

***Release 1.0***

**Joshua Hines**

**Oct 02, 2019**



---

Links:

---

<b>1</b>	<b>General</b>	<b>3</b>
<b>2</b>	<b>Asset Creation</b>	<b>15</b>
<b>3</b>	<b>World Building</b>	<b>117</b>
<b>4</b>	<b>Miscellaneous</b>	<b>189</b>







## 1.1 Getting started with the ATLAS Devkit

So you've decided to be a mod author for ATLAS, but you aren't sure where to start?

Well, you have come to the right place.

### 1.1.1 Installing the ATLAS Devkit

To get started the first thing any new ATLAS mod author will need to do is install the devkit.

- Install the editor from the Epic Launcher
- Download an *OPTIONAL* package of Ocean Map Content Files by selecting it in the installation options on the Epic Launcher.
- The base level file for creating Non-Seamless (ARK-Style) mod maps is now included with the core Epic Launcher installation of the editor
- Extract it to some location where you have 180 GB of free space (SSD recommended), and then run it with ATLASDevKit.bat
- Extract the patch over the devkit installation (directories need to be properly aligned).

### 1.1.2 Running ATLAS Steam Workshop Content

If you're hosting a seamless grid, all Mod Steam Workshop ID's that you wish to use (also including any new persistent level Map Mod ID's, and any Island Extension Mod ID's) ought to be specified here in the Grid json or via the World Grid editor (in Mod stacking order):

Edit project
✕

Atlas Id

Friendly Name

Cells X  Y  CellSize   
*In Unreal Units*

Column UTC Offset

Additional CmdLine Params:

TransitionMinZ  (ex: -12000)

Mod Ids  (Use comma as separator between the Ids)

Map Image URL  (optional: If you want to use URL instead of UE4)

MetaWorldURL  (optional: Base URL for http hosted map images)

AuthListURL  (optional: URL for clients to download cluster IPs)

Atlas Password

BaseServerArgs

Override ShooterGameMode DefaultGame.ini Values

	Key	Value
•		

New Database Settings

<input type="text" value="S3 URL"/>							
<input type="text" value="KeyId"/>							
<input type="text" value="SecretKey"/>							
<input type="text" value="BucketName"/>							
<input type="text" value="Region"/>							

Tribe Log Config

Shared Log Config

Travel Data Config

	Name	URL	Port	Password
DB Entry 1:	Default	127.0.0.1	6379	foobared
DB Entry 2:	TribeDB	127.0.0.1	6379	foobared
DB Entry 3:	TravelDataDB	127.0.0.1	6379	foobared
DB Entry 4:	TerritoryDB	127.0.0.1	6379	foobared
DB Entry 5:	LogDB	127.0.0.1	6379	foobared

Use UTC Time

Day-0

globalGameplaySetup:

```
{
  QuestEntries=
  ((QuestID=0,CompletedIcon=Texture2D'/Game/Atlas/UI/Icons/Icon_PowerStonesQuest_Complete.Icon_PowerStonesQuest_Complete',UncompletedIcon=Texture2D'/Game/Atlas/UI/Icons/Icon_PowerStonesQuest_Uncomplete.Icon_PowerStonesQuest_Uncomplete',QuestName="Voyage of
```

Which is equivalent to adding ID's directly into the ServerGrid.json here:

```

ServerGrid.json x
1  {
2    "BaseServerArgs":
3      "%MapName%%GridLocation%?AltSaveDirectoryName=%AltSaveDir
4      -log -server -culture=en -NoCrashDialog -NoBattleEye",
5    "gridSize": 1400000.0,
6    "MetaWorldURL": "",
7    "WorldFriendlyName": "ATLAS Official Pvp NA 15x15",
8    "WorldAtlasId": "1202965888",
9    "AuthListURL": "",
    "WorldAtlasPassword": "",
    "ModIDs": "1625331561,1626832642,1626876784,162688118",
  }

```

However if you're running a Non-seamless-grid Mod map, then you need to specify all the Mod ID's (including the Mod Map's ID) that you wish to use via commandline like so:

```

ShooterGameServer.exe OceanTestLevelMod?SessionName=WhateverSessionName?
↪AltSaveDirectoryName=WhateverSaveDirectory?
↪ServerAdminPassword=whateverserverpassword?MaxPlayers=70?QueryPort=57557?Port=5757?
↪GameModIds=1627890405,1625331561,1627715666?MapPlayerLocation=true -log -server -
↪culture=en -NoCrashDialog -NoBattleEye -NoSeamlessServer

```

You can also add more mods in the Game.ini with this section:

```

[ServerSettings]
ActiveMods=12334593945,875445124

```

Upon being launched, the server will then automatically download and install the Mods upon being launched, as well as update them each time the server is run. If you wish to disable the autodownload/update functionality, you can run the server with `-manualmanagedmods`

### 1.1.3 Working with the Devkit

The first thing to know is that the ATLAS Devkit is a branched version of Unreal Engine 4. Specifically, it is derived from version 4.5.1 with custom modifications.

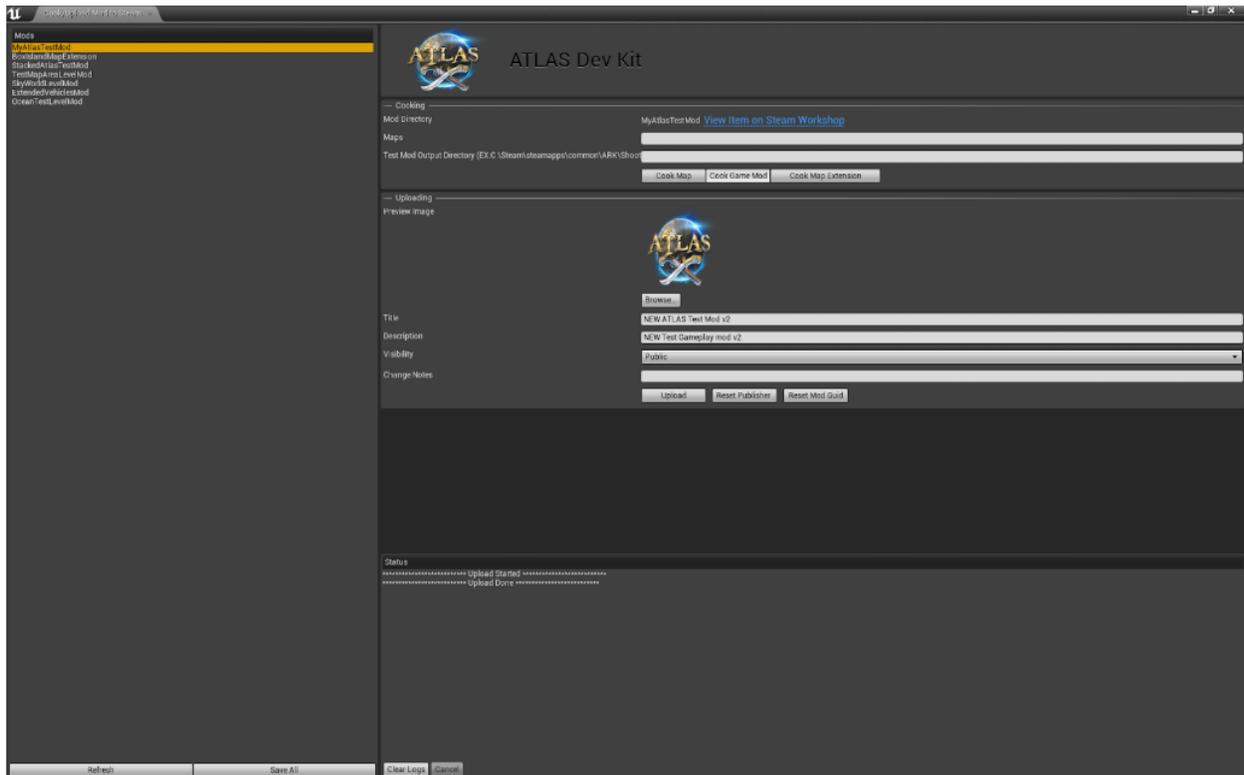
**Note:** While not everything is in parity with standard UE4, most of the functionality can be found within the [Unreal Engine 4 documentation](#). It is recommended to familiarize yourself with standard usage of Unreal Engine as this documentation will general not cover packaged features/systems.

Upon first opening the devkit you will likely see what might be an overwhelming scene if you are unfamiliar with Unreal Engine.

#### Creating Mods

To make a Mod using the ATLAS Dev Kit, as you may remember from ARK, simply create a child of "Blueprint'/Game/PrimalEarth/CoreBlueprints/PrimalGameData\_BP.PrimalGameData\_BP'" into your "Mods\YourModName" folder. We recommend following the naming

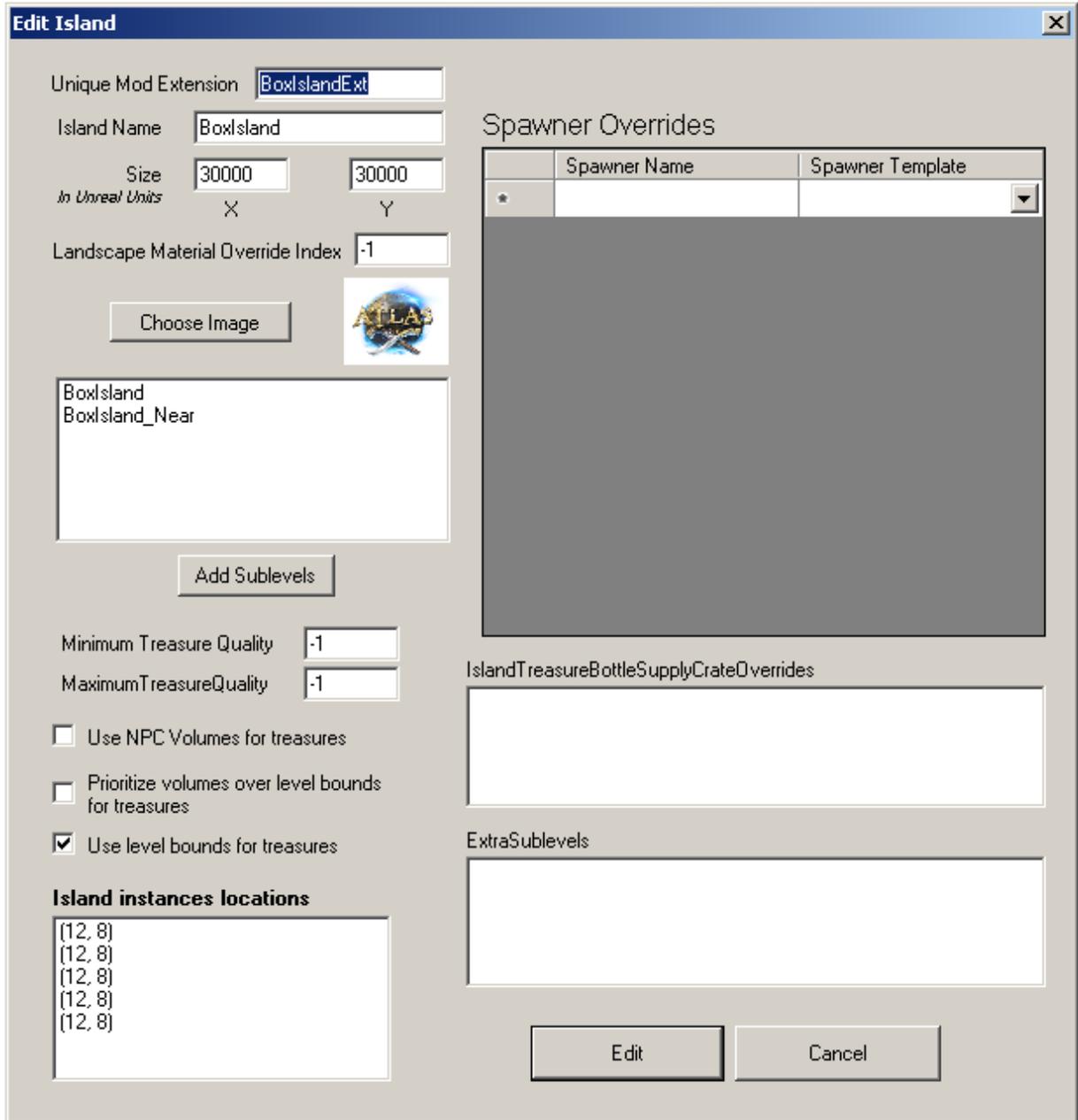
scheme of "PrimaGameDataBP\_YourModName.uasset". Then change whatever values you in that new GameData asset to alter the game functionality (note there are various new fields for ATLAS, including "AdditionalInventoryComponentAppends" to add new craftable item types to existing inventories). Then click the "Steam Upload" button in the toolbar, select the folder of your Mod in the Steam Upload dialog, and click "Cook Game Mode". After the Cook process is done (if you get a crash dialog at the end of the Cook process, just close the crash dialog as it's harmless), browse for a small preview PNG (256x256 or so), add a Title, Description, Visibility Setting, and Upload to Steam (input your Steam Username and Password when prompted).



## Creating Map Extensions

One of the most exciting new concepts for Workshop Content in ATLAS is the idea of "Map Extensions", which allows you to add new sets of Islands and landmasses that server hosts can place upon their seamless grid layouts. To create a "Map Extension" for use in seamless grid layouts, simply open "SeamlessTest\Ocean.umap" in the Editor, and add a set of new uniquely named worldcomposition sublevels to the "SeamlessTest\" folder (yes, put them there, not in your Mod folder). These should all be centered on the ORIGIN 0,0,0 (where they'll be placed in the actual seamless grid layout is up to the server host). Then, set the WorldComposition layer for each those Sublevels to "Farthest", "Far" or "Near" depending on whether they're long-distance or short-distance visible levels. Finally, create an empty "Mods\YourExtensionName" folder, and then select it in the Steam Upload dialog. Then input the names of each of your new sublevels into the Maps field like so, and click "Cook Map Extension":

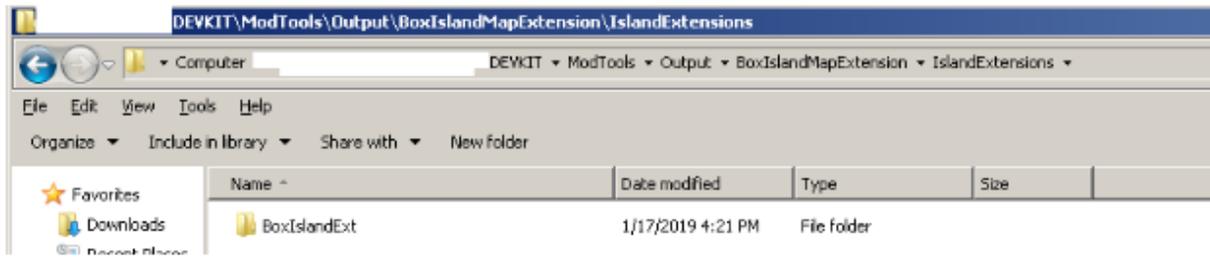




Be sure to add a uniquely named “Extension” for your Island Template, because this will create a unique directory here:

```
\ServerGridEditor\IslandExtensions\[EXTENSION_NAME]
```

We recommend that you copy that unique extension directory into “ModToolsOutputYourMapExtensionNameIslandExtensions” after you have cooked, like so:



And then instruct server hosts, after subscribing to your Map Extension on their Steam clients, to copy the “IslandExtensions” folder you’ve included with your files, into their local ServerGridEditor application’s directory so that they can place your new Island Templates within their own server layout (as many times as they wish).

So after a host downloads your content on their Steam client, the hosts should copy this folder:

```
\SteamLibrary\steamapps\workshop\content\834910\[YOUR_WORKSHOP_CONTENT_ID]\IslandExtensions\
```

Over onto this in their ServerGridEditor application directory:

```
\ServerGridEditor\IslandExtensions\
```

hey’ll then see your custom Island Template appear in the list of available Island Templates when they reload the Server Grid Editor.

Note that server hosts should remember to add your Map Extension’s ID to their Grid’s Mod ID’s list in the Server Grid settings! And yes it’s entirely possible for a single Map Extension package to include multiple new Island Templates and landmasses for server hosts to use, you don’t have to limit to just a single new Island Template! (Note: if you set the sublevels to UNCATEGORIZED layer, they will be always-loaded, once, on every grid server – that may be useful if you want, say, a persistent weather system addition to be present in every grid region.)

## Creating Maps

Making a new “base” Map is similar to cooking a Mod, with the uniquely-named map file .umap needing to be placed in your Mods\YourMapName\ folder. Then click “Cook Map” in the Steam Uploads dialog. Typically, new custom Maps will ARK-style (and not seamless grid persistent levels), so you’ll want to run those with the following server commandline:

```
ShooterGameServer.exe OceanTestLevelMod?SessionName=WhateverSessionName?
↳AltSaveDirectoryName=WhateverSaveDirectory?
↳ServerAdminPassword=whateverserverpassword?MaxPlayers=70?QueryPort=57557?Port=5757?
↳GameModIds=1627890405?MapPlayerLocation=true -log -server -culture=en -
↳NoCrashDialog -NoBattleeye -NoSeamlessServer
```

However, you can certainly make new “seamless grid persistent levels” too, such as this SkyWorldLevel which removes the Ocean (and of course it could new things too, such as a different DayCycleManager for new weather systems!). To make new “seamless grid persistent levels”, we suggest resaving SeamlessTest\Ocean.umap into your Mods\WhateverCustomMapName directory, and then modifying it as you see fit. We think it’d be very cool to make some Avatar-esque floating islands Map Extensions to go along with that. . .

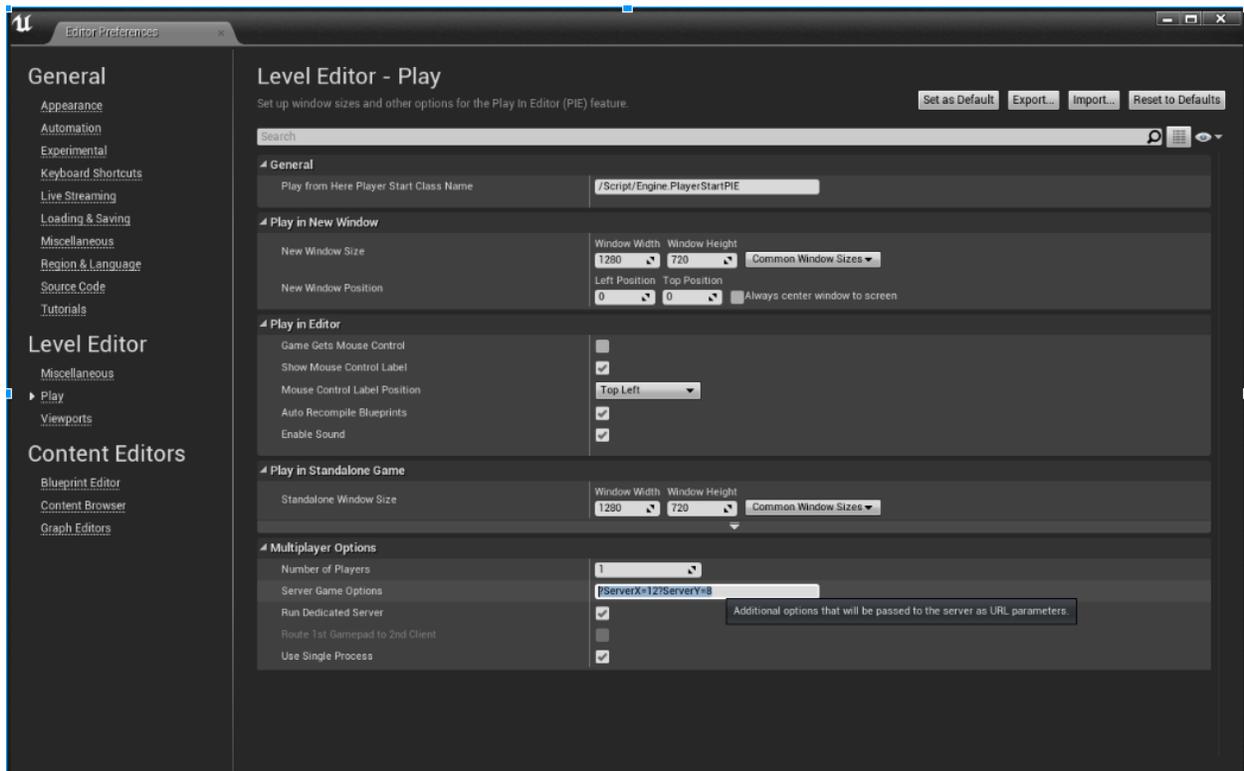
New “seamless grid persistent levels” are cooked the same way, but then simply run with the grid commandline like so (presuming that the Map Steam Workshop ID for the level content is specified within the ServerGrid’s json, or via the [ServerSettings] ActiveMods section of the Game.ini!):

```
ShooterGameServer.exe SkyWorldLevel?ServerX=12?ServerY=8?AltSaveDirectoryName=JerTest?  
↪ServerAdminPassword=grapeshot?MaxPlayers=200?ReservedPlayerSlots=50?QueryPort=57557?  
↪Port=57557?SeamlessIP=37.10.127.116?MapPlayerLocation=true -log -server -culture=en -  
↪NoCrashDialog -NoBattleEye -automanagedmods
```

### Steam Workshop ID 1626926342 for ‘SkyWorldLevel’

**Note:** ATLAS currently only supports client-and-dedicated server mode. Running in single-player or as a listen-server will have issues.

To experiment with Seamless Grid levels (as a client) in the Editor, you can run Play-In-Editor on “SeamlessTest\Ocean.umap” with these Server Game Options in the Advanced Settings dialog:



You can substitute whatever ServerX ServerY coordinates you want to play on different grid regions (the game uses whatever ServerGrid.json is located in the ShooterGame folder). Alternatively if you just wish to try your new Islands, or a specific combination of sublevels, without setting them up on the json grid layout, you can run Ocean.umap PIE without any ServerX/ServerY commands and it'll Play-In-Editor without being on any specific region. If you have any sublevels VISIBLE in the Editor Viewport before you do this, they'll remain loaded in Play-In-Editor, presumably at the origin where you constructed your Islands. In this way, you can test and iterate on the gameplay of your new islands without having to set them up on the servergrid (of course, this method is only practical for testing out one new island set of sublevels at a time).

You can also run the Seamless Grid outside the Editor (useful for testing transition testing), with the following batch files:

- DedicatedServerAllFourAndClient\_CLEAN.bat
- And alternatively ClientSub2.bat if you wish to connect a second client.

You can also run a single non-SeamlessGrid server (such as TestMapArea or OceanTest) outside the Editor with the

following batch files:

- GameDedicatedServer.bat
- GameClient.bat
- And alternatively GameClient2.bat if you wish to connect a second client

### 1.1.4 Sample Mods

Here are some example simple Mods/Maps we've provided, the sources for these are included in the DevKit ShooterGameContentMods folder to help you get started:

#### ATLAS OceanTest Level

<https://steamcommunity.com/sharedfiles/filedetails/?id=1627890405>

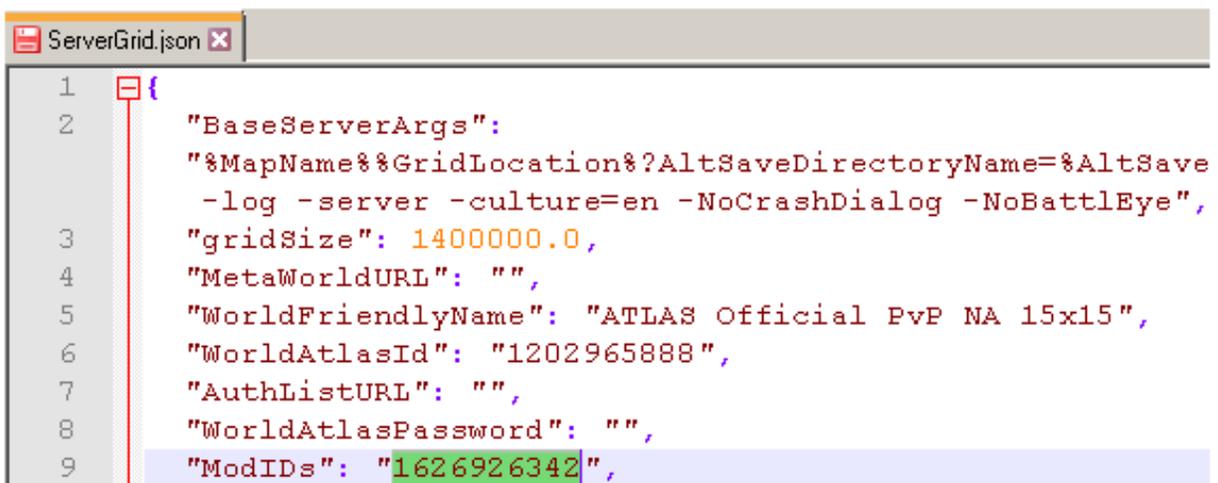
This is a non-seamlessgrid level, a traditional ARK-style single-server level that has an ocean and some basic box floors for testing, as an example. Servers can host with this:

```
ShooterGameServer.exe OceanTestLevelMod?SessionName=WhateverSessionName?
↪AltSaveDirectoryName=WhateverSaveDirectory?
↪ServerAdminPassword=whateverserverpassword?MaxPlayers=70?QueryPort=57557?Port=5757?
↪GameModIds=1627890405?MapPlayerLocation=true -log -server -culture=en -
↪NoCrashDialog -NoBattleEye -NoSeamlessServer
```

#### ATLAS Sky World

<https://steamcommunity.com/sharedfiles/filedetails/?id=1626926342>

This is a resave of OceanTest.umap, but with the Ocean water removed. It's used as an example of how to have an alternate seamlessgrid persistent level. You can host it by adding this to your ServerGrid.json:



```
1  {
2    "BaseServerArgs":
3    "%MapName%%GridLocation%?AltSaveDirectoryName=%AltSave
4    -log -server -culture=en -NoCrashDialog -NoBattleEye",
5    "gridSize": 1400000.0,
6    "MetaWorldURL": "",
7    "WorldFriendlyName": "ATLAS Official Pvp NA 15x15",
8    "WorldAtlasId": "1202965888",
9    "AuthListURL": "",
    "WorldAtlasPassword": "",
    "ModIDs": "1626926342",
```

Or by setting that value here in the Server Grid Editor:

**Edit project**

Atlas Id:

Friendly Name:

Cells X:  Y:  CellSize:  In Unreal Units

Column UTC Offset:

Additional CmdLine Params:

TransitionMinZ:  (ex: -12000)

Mod Ids:  (Use comma as separator between the Ids)

Map Image URL:  (optional: If you want to use URL instead of UE4)

MetaWorldURL:  (optional: Base URL for http hosted map images)

AuthListURL:  (optional: URL for clients to download cluster IPs)

Atlas Password:

BaseServerArgs:

**Override ShooterGameMode DefaultGame.ini Values**

Key	Value
*	

**New Database Settings**

S3 URL:

KeyId:

SecretKey:

BucketName:

Region:

Name	URL	Port	Password
DB Entry 1: Default	127.0.0.1	6379	foobared
DB Entry 2: TribeDB	127.0.0.1	6379	foobared
DB Entry 3: TravelDataDB	127.0.0.1	6379	foobared
DB Entry 4: TerritoryDB	127.0.0.1	6379	foobared
DB Entry 5: LogDB	127.0.0.1	6379	foobared

Tribe Log Config | Shared Log Config

Travel Data Config

Use UTC Time

Day-0:

globalGameplaySetup:

```
{QuestEntries=
((QuestID=0,CompletedIcon=Texture2D'/Game/Atlas/UI/Icons/Icon_PowerStonesQuest_Complete.Icon_PowerStonesQuest_Complete',UncompletedIcon=Texture2D'/Game/Atlas/UI/Icons/Icon_PowerStonesQuest_Uncomplete.Icon_PowerStonesQuest_Uncomplete',QuestName='Voyage of
```

Edit | Cancel

Or in the Game.ini:

```
1 [ServerSettings]
2 ActiveMods=1626926342
```

And then run the servers with:

```
ShooterGameServer.exe SkyWorldLevel?ServerX=12?ServerY=8?AltSaveDirectoryName=JerTest?
↪ServerAdminPassword=grapeshot?MaxPlayers=200?ReservedPlayerSlots=50?QueryPort=57557?
↪Port=5757?SeamlessIP=[THEIP]?MapPlayerLocation=true -log -server -culture=en -
↪NoCrashDialog -NoBattleye
```

(continues on next page)

(continued from previous page)

### Extended Vehicles Mod (Stackable)

A very simple box-island-with-trees “Map Extension” for seamless grids, placeable as an island on your servergrid layouts using the included Island Template files. To use this in a server layout, in addition to adding the ID to the Mod Ids for the grid, subscribe to the files on a client and then copy this folder:

```
SteamLibrary\steamapps\workshop\content\834910\1626922438\IslandExtensions
```

Over this folder onto hosts’ Server Grid Editor application directory:

```
\ServerGridEditor\IslandExtensions\
```

Then relaunch the Server Grid Editor and you’ll see the new “BoxIsland” Template available for use in your layout wherever you want:

### MyAtlasTestMod

<https://steamcommunity.com/sharedfiles/filedetails/?id=1625331561>

A very simple “Base” or non-stackable Mod that just adds a few different Default Feats, used in testing that a base non-stackable Mod works.

### Stacking MOD Test

<https://steamcommunity.com/sharedfiles/filedetails/?id=1626876784>

This is a stackable Mod that adds an additional unlocked Feat (testing stacking functionality).

Hopefully that covers a lot of information in order for you guys to get started! Something you’ll notice is that we didn’t strip the majority of ARK content from the kit, and that’s because it can be used as a basis for building new content in ATLAS, especially for those of you familiar with the ARK DevKit, and some of you may find it helpful when trying to port over your existing ARK Mods.

As for the question regarding can ARK content be used in ATLAS and vice versa? The overall answer is that we’re fine with it, however, the only thing that would not be allowed and will cause your mod to be removed from the Steam Workshop is using any Wildcard specific maps. This means that you are not allowed to port over The Island, Scorched Earth, Aberration, Extinction, or Ragnarok.

We’ll be working on some additional documentation that will help guide users through some of the new systems in the DevKit, so be on the lookout for that. Good luck, we can’t wait to see all the epic content you guys create!

## 1.2 Frequently Asked Questions

**Q:** Why does my devkit lock up when I try to open files?

**A:** When the devkit opens a blueprint file, it also opens any of the assets and blueprints that it references, loading them into memory. The first few files that you open any time you restart the devkit can take a sizable amount of time. This can be drastically mitigated by installing the devkit onto an SSD.

## 1.3 Tools

### 1.3.1 Python

#### Overview

Below are some general installation instructions for setting up Python and any modules you might need. ATLAS tools use the latest version of python 2.7.X. (<https://packaging.python.org/installing/>)

1. Upgrade to latest version of pip

```
python -m pip install -U pip setuptools
```

2. Run `python get-pip.py`

3. Add pip folder to PATH environment folder

4. `pip install numpy`

5. `pip install -U PySide`

6. [http://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_setup/py\\_setup\\_in\\_windows/py\\_setup\\_in\\_windows.html](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_setup/py_setup_in_windows/py_setup_in_windows.html)

7. `pip install opencv-python`

Sample Assets and Content Tools can be found on GitHub: <https://github.com/GrapeshotGames/AtlasDevKitContent>

## 2.1 Testing Content in the Game

### 2.1.1 Overview

This section will go over how to test content in Atlas PIE (Play-In-Editor).

### 2.1.2 Best Practices

To trigger content while playing the game you usually need to use 'Cheat Codes'. These are typed in at the command prompt that comes up when you hit the tilde (``) key in the top left of your keyboard.

Things to take note of before testing content:

- **You should always test your content in an actual playable level for all times of day to make sure the lighting is good during**
  - cheat settimeofday 12:00
- **To avoid your character from getting attacked by creatures or NPC players or having health issues due to cold/hunger/etc**
  - cheat leavemealone
  - cheat setinfinitestats
- **Tame the ocean- often the Ocean's waves are too big for testing on a platform or on a ship, so you can change the wave's h**
  - cheat setwind 3
- **Sometimes you want to be able to watch the game from any viewpoint without interaction. To do this, enter spectator mod**

- cheat enablespectator

Often you need to spawn something else first to test your content. Like for Cannon shot you might need to spawn a boat and then spawn a cannon on it. So some of the examples below will need to be used in combination to do what you want to do.

### 2.1.3 Spawn FX

#### Spawn a Ship

To create a boat it is best to type in this Cheat Code to fly out over the water:

- cheat fly

To walk on the Boat again, fly down near the top deck and enter this cheat code:

- cheat walk

Once you are over the water in a good, spot type this to create each type of ship:

- **Brig**
  - cheat spawnshipfast “brig”
- **Schooner**
  - cheat spawnshipfast “schoo”
- **Sloop**
  - cheat spawnshipfast “sloop”
- **Dinghy**
  - cheat spawnshipfast “ding”
- **Raft**
  - cheat spawnshipfast “raft”
- **Galleon**
  - cheat spawnshipfast “gall”
- **AoD Ship**
  - cheat spawnactor “Blueprint’/Game/Atlas/AtlasCoreBP/AI/ShipNPC\_Brig.ShipNPC\_Brig” 4000 0 0
  - If you have a Brig or other ship in the ocean the AoD ship will start firing at you automatically

To spawn ships to sail easily to test wakes and be able to orbit around to test FX, etc:

- cheat ssf “brig” (or “gall”/”schoo”/”sloop”, “ding”, “raft”...)
- cheat setignorewind true
- spacebar to open sails (edited)
- z to close sails
- K to orbit cam
- Mousewheel down to zoom out

## Swivel Cannon

Once you have created a boat, you need to mount a Swivel Cannon on it. This is usually accomplished by hitting a function key or a Num key or some combination with Shift and a Function key.

Once your Swivel Cannon is created, use these cheat codes to get Ammo for Grapeshot.

- cheat gfi “grapeshot” 20 1 0
- cheat giveitem “/Game/YARK/ShipCombat/Test\_Christian/PrimalItemAmmo\_GrapeShot.PrimalItemAmmo\_GrapeShot” 10 0 0 false

**Mount and Fire:** To fire you must ‘mount’ the Cannon by moving close to it and looking down, and then hitting the ‘e’ key to mount. You fire by hitting the left mouse once button to load and then once to fire. Repeat to keep firing.

## Canister Shot

- use cheat gfi canistershot 50 1 0
- Hold E at the swivel gun and select change ammo type to canistershot
- mount the swivel gun and fire as noted above

## Barshot

- use cheat gfi barshot 50 1 0
- Hold E at the swivel gun and select change ammo type to canistershot
- mount the swivel gun and fire as noted above

Another method someone else posted:

- For Testing Swivel cannon in Test Level on TRUNK: Spawn cheat structure box (F8). Grab Primal Wood floor and primal wood wall. Place a few wood floors and select & place 1/2 walls (press T to select 1/2 wall option from menu- lower lft screen). Spawn and place swivel cannon on the 1/2 wall (shift F6). Again, Swivel gun uses grapeshot and canister shot

## Cannon

- **Cannon**
  - **To get a cannon**
    - \* cheat gfi “shipcannon” 1 1 0
  - **to get cannon ammo**
    - \* cheat gfi “cannon” 50 1 0
- **Barshot**
  - **To get BarShot Ammo**
    - \* cheat gfi “barshot” 50 1 0
  - place cannon, get on cannon, hit shift to change active ammo type to barshot
  - To do a Large Cannon, use shipcannon\_large
- **SpikeShot**

- cheat gfi “spikeshot” 50 1 0
- place cannon, get on cannon, hit shift to change active ammo type to spikeshot

### Mortar

The Mortar doesn't require a ship so could be spawned on land or on a ship

- **Mortar:**

- cheat giveitem “Blueprint’/Game/YARK/ShipCombat/Test\_Christian/PrimalItemStructure\_ShipMortar.PrimalItemStruc  
1 0 0

- **Ammo:**

- cheat giveitem “Blueprint’/Game/Atlas/Weapons/Mortar/MortarAmmo/PrimalItemAmmo\_MortarShot.PrimalItemAmm  
1 0 0

### Pistols

- cheat giveitem “/Game/YARK/CoreBlueprints/Items/PrimalItem\_YarkWeaponDualPistols.PrimalItem\_YarkWeaponDualPistols”  
4 0 0 false
- Get a few of those and put them in hotbar
- Bullets: cheat giveitem “/Game/PrimalEarth/CoreBlueprints/Weapons/PrimalItemAmmo\_SimpleBullet.PrimalItemAmmo\_Simpl  
10 0 0 false
- Equip two
- Then LMB and RMB will shoot so you can see the effect

### Bow

- cheat giveitem “/Game/PrimalEarth/CoreBlueprints/Weapons/PrimalItem\_WeaponBow.PrimalItem\_WeaponBow”  
1 0 0 false | cheat giveitem “/Game/PrimalEarth/CoreBlueprints/Weapons/PrimalItemAmmo\_ArrowStone.PrimalItemAmmo\_Arro  
10 0 0 false

### Power Hit

- **To use Power Hit**
  - “cheat giveengrams” and then ctrl + 3

### Two Handed Mace

- cheat giveitem “/Game/Atlas/Weapons/TwoHandedMace/PrimalItem\_TwoHandedMace.PrimalItem\_TwoHandedMace”  
1 0 0 false

### Sword

- Cheat giveitem “/Game/YARK/CoreBlueprints/Weapons/PrimalItem\_YarkWeaponSword.PrimalItem\_YarkWeaponSword”  
1 0 0 false

- You can also go to any inventory item blueprint (PrimalItem\_YarkWeaponSword) and right click it and say give item ingame

## Sextant

- **cheat giveitem “/Game/Atlas/Weapons/Sextant/PrimalItem\_WeaponSextant.PrimalItem\_WeaponSextant” 1 0 0 false**
  - Hold right click to start using the sextant

## Compass Visual Marker

To spawn a compass use this cheat code:

- cheat giveitem “Blueprint’/Game/PrimalEarth/Test/PrimalItem\_WeaponCompass.PrimalItem\_WeaponCompass” 1 0 false
- Then click ‘r’ or middle mouse button to get the Compass Visual Marker to come up

## Human NPC

Spawn Human NPC to fight you and test block FX.

- **To get a sword use this cheat:**
  - cheat giveitem “/Game/YARK/CoreBlueprints/Weapons/PrimalItem\_YarkWeaponSword.PrimalItem\_YarkWeaponSword” 1 0 0 false
- cheat spawnactor “/Game/Atlas/Test/HumanNPC\_BP\_Male\_NewMelee.HumanNPC\_BP\_Male\_NewMelee” 0 200 0 0
- Block with alt+LMB or alt+RMB (will block an attack from the left or right respectively)
- Block with either the sword or the fists and see the effect
- U and Y toggle between FPV and TPV
- **Here is another way to spawn an NPC, but I can’t remember what I used this for**
  - cheat spawnactortamed “Blueprint’/Game/Atlas/Test/HumanNPC\_BP\_Crew\_Male.HumanNPC\_BP\_Crew\_Male” 200 1 0

## Skeleton

- climbs out of the ground and fights
- cheat spawnactor “Blueprint’/Game/Atlas/Test/HumanNPC\_BP\_Enemy\_Melee.HumanNPC\_BP\_Enemy\_Melee” 400 0 -80

## Treasure/Shovel & Digging

- Start editor from GameEditor\_SeamlessOcean\_Single.bat
- Play
- cheat givetreasuremap 1

- cheat giveitem “Blueprint’/Game/Atlas/Weapons/Shovel/PrimalItem\_WeaponShovel.PrimalItem\_WeaponShovel” 1 1 0 false
- cheat leavemealone
- Equip shovel and treasure map from inventory into hotbar (edited)
- Cheat fly and look for the glowing column on the same island
- Equip shovel, look at ground and press lmb
- Cnce you are set up you can leave pie running though if you just iterate on fx
- Use cheat givetreasuremap 1 again to get another map for the same island (edited)

### Cyclone FX

Load the Ocean Map and an Island level and hit play (the Cyclone doesn’t seem to work with the ‘Auto Play’ technique or on a Test Map). Then fly far away from the island and spawn a boat far away from land. Cheat walk to get onto the boat and enter these cheat codes to get the bad weather to ramp up and the Cyclones to start:

- *cheat ce daycycle startrain*
- *cheat slomo 100*
- **Once you see the Cyclones starting enter:**
  - *cheat slomo 1*
- Takes a little time to ramp up

### Ship Scaffolding

- press F9 to place the small shipyard
- *cheat setwind 1 (I dont like waves)*
- *cheat gfi wood 100 1 0*
- put wood in shipyard (press E on it, and you will open its inventory and inside you will see the blueprints to build the ships)
- *cheat fly*
- go to ship yard, build a ship (this will spawn the hull and scaffolding)
- hold E on the shipyard and select the demolish ship option

### Water Bucket

- **cheat giveitem “Blueprint’/Game/Atlas/Weapons/Bucket/PrimalItem\_WeaponBucket.PrimalItem\_WeaponBucket” 1 1 0**
  - (equip bucket, fill up in water with LMB, 2x for full)

### Oil Jar Fire

- cheat giveitem “Blueprint’/Game/PrimalEarth/CoreBlueprints/Weapons/PrimalItem\_OilJarNew.PrimalItem\_OilJarNew” 5 0 0 | cheat gfi flame 10 0 0 | cheat gfi bow 1 0 0

## Kraken

- load Ocean.umap
- load sublevel - endboss
- make sure “Run dedicated server” is checked
- Launch the game
- “cheat ghost” to stop falling
- cheat ce startboss (spawns the Kraken)

## GhostShip

- cheat ssf ghost
- cheat settimeofday 23:59
- two times
- should start to fade in then
- if you do cheat settimeofday 23:59 again it should fade out

## Spawn random Crew

- cheat spawnactortamed “Blueprint’/Game/Atlas/Test/HumanNPC\_BP\_Crew\_Male.HumanNPC\_BP\_Crew\_Male”  
200 1 0

## Misc Commands

- Cheat SetWind ## (1-15) Changes the wind and wave speed. Higher is greater. 0 resets it to the default server random cycling.
- Cheat Slomo ## (0.01 - 15) Changes server speed, 0.1 is 1/10th speed, 10 is 10x speed.
- Cheat SSF Ding Spawns a Dinghy / Rowboat for fast iteration.
- Cheat SSF Sloop Spawns a Sloop ship with some structures pre-placed for faster iteration.
- Cheat SSF Schoon Spawns a Schooner ship with some structures pre-placed for faster iteration.
- Cheat SSF Brig Spawns a Brigantine ship with some structures pre-placed for faster iteration.
- Cheat SSF Gall Spawns a Galleon ship with some structures pre-placed for faster iteration.
- “Cheat SDF Crew\_M 1” Spawn NPC Crew for your ship. The \_M is necessary or you will get Ken doll versions.
- Good startup equip command - copy/paste this:

```

cheat giveitem "/Game/Atlas/Weapons/GrapplingHook/PrimalItem_WeaponGrapplingHook.PrimalItem_WeaponGrapplingHook"
1 0 0 | cheat giveitem "/Game/Atlas/Weapons/RepairHammer/PrimalItem_WeaponRepairHammer.PrimalItem_WeaponRepairHammer"
1 0 0 | cheat giveitem "Blueprint'/Game/PrimalEarth/Test/PrimalItem_WeaponSpyglass.PrimalItem_WeaponSpyglass'"
1 1 0 | cheat giveitem "/Game/YARK/CoreBlueprints/Weapons/PrimalItem_YarkWeaponSword.PrimalItem_YarkWeaponSword"
1 0 0 | cheat giveitem "/Game/YARK/CoreBlueprints/Items/PrimalItemArmor_YarkWoodShield.PrimalItemArmor_YarkWoodShield"
1 0 0 | cheat giveitem "Blueprint'/Game/Atlas/Weapons/Shovel/PrimalItem_WeaponShovel.PrimalItem_WeaponShovel'"
1 1 0 | cheat giveitem "/Game/YARK/CoreBlueprints/Items/PrimalItem_YarkWeaponDualPistols.PrimalItem_YarkWeaponDualPi"
4 0 0 false | cheat giveitem "/Game/PrimalEarth/CoreBlueprints/Weapons/PrimalItemAmmo_SimpleBullet.PrimalItemAmmo_Sim"
10 0 0 false

```

## 2.2 Particles

### 2.2.1 Overview

This page covers some best-practices and things to look out for when creating particle systems.

### 2.2.2 Naming Convention

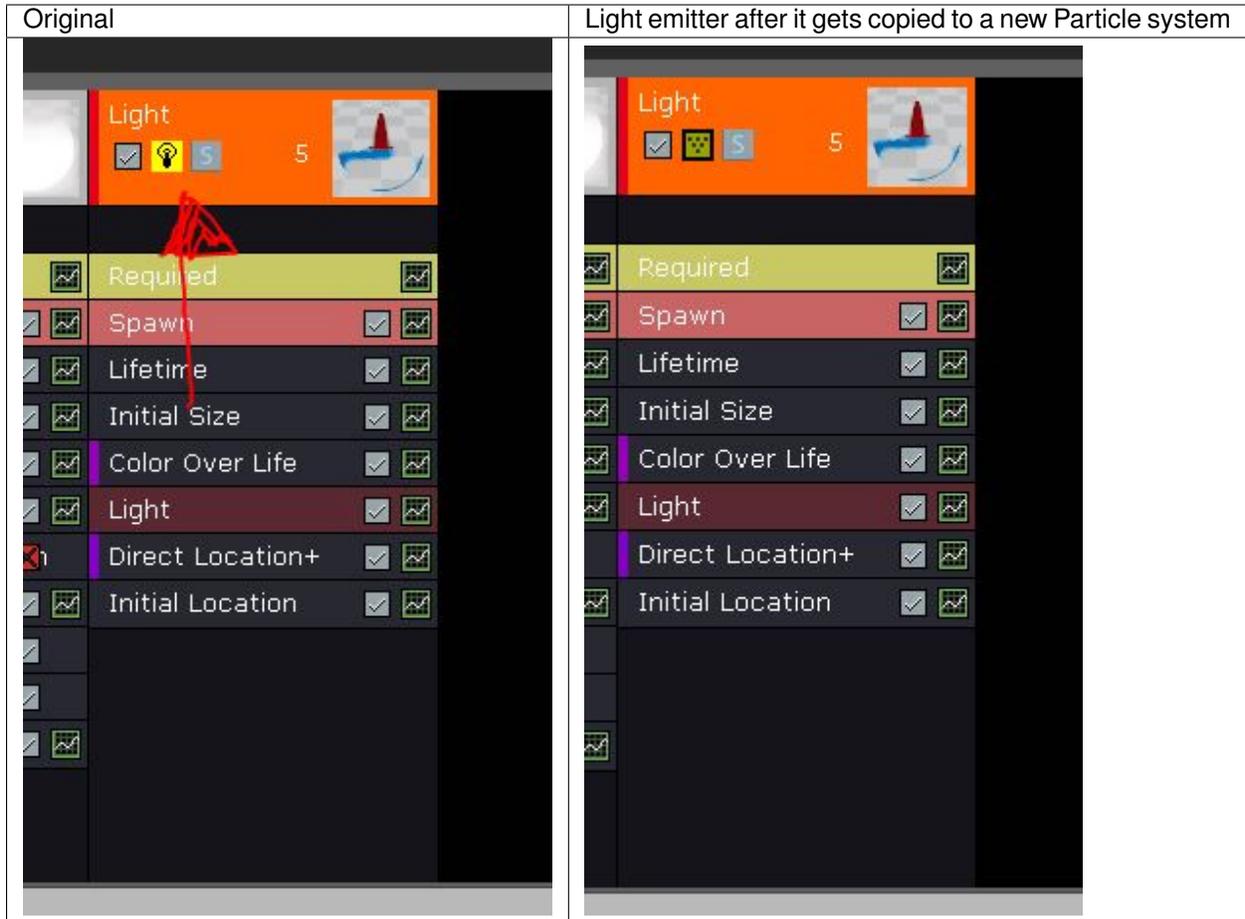
Please, follow [unreal naming convention](#)

Textures:	<b>T_whateverName_D</b> (D for Diffuse, N for Normal, etc.)
Master Material:	<b>MM_WhateverName</b>
Material Instance:	<b>MIC_WhateverName</b>
ParticleSystem:	<b>P_WhateverName</b> (don't forget to name your emitter inside the system)
Meshes:	<b>FX_whateverName</b>

### 2.2.3 Miscellaneous

**Warning:** Careful copying emitters from one particle system to another

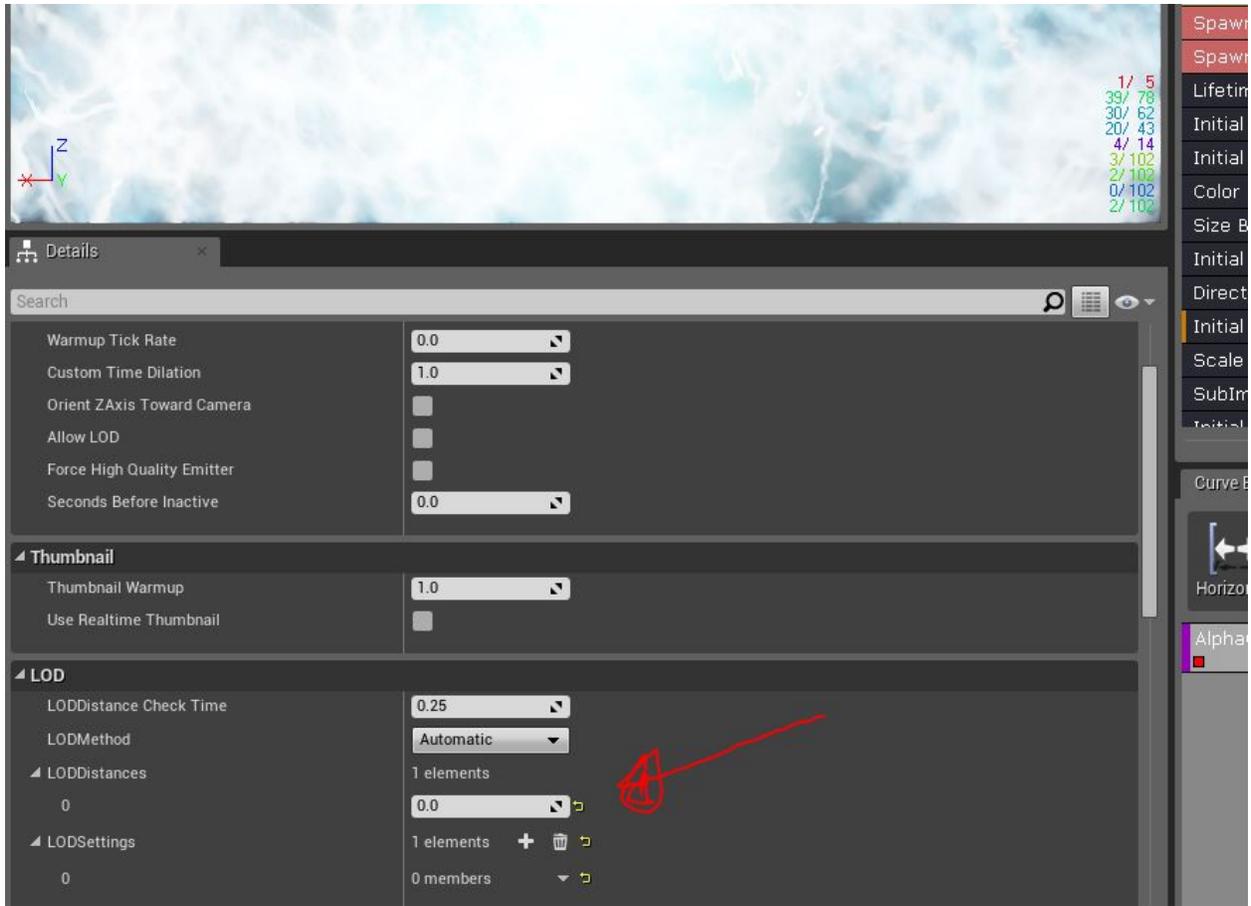
Particle Lights: Copying a light emitter that toggles the light type module at the top into a new particle system will toggle the emitter back to a normal particle. The light bulb icon means the emitter will hide the material that is associated with the emitter. When it toggles back to the down arrow the material will show up and thus needs to be switched back or else unintended visuals result in the particle.



Particle LODs: Copying emitters from a particle system that has LODS will transfer those LODS to the new particle system with weird settings on what's turned on or off on the higher LODS. This could cause weird visuals when you move far enough from the particle. Particle systems should be cleaned up in this case.

NOTE: The standard for ATLAS VFX is to not make use of UE4 Cascade LODs, and use only one main LOD.

Where to find how many lods are in the particle system



Change lods levels above and manually delete ones not needed.



### FX for Cinematics

Make sure that FXs that are only used in cinematics (make duplicates if necessary) have “**Sec before inactive**” in the particle system set to -1 so it always plays.

## 2.2.4 Detailed Guides

### Base Space

#### Overview

ATLAS has added new functionality for World Space particles to be “based” on another actor’s transform, such as a ship!

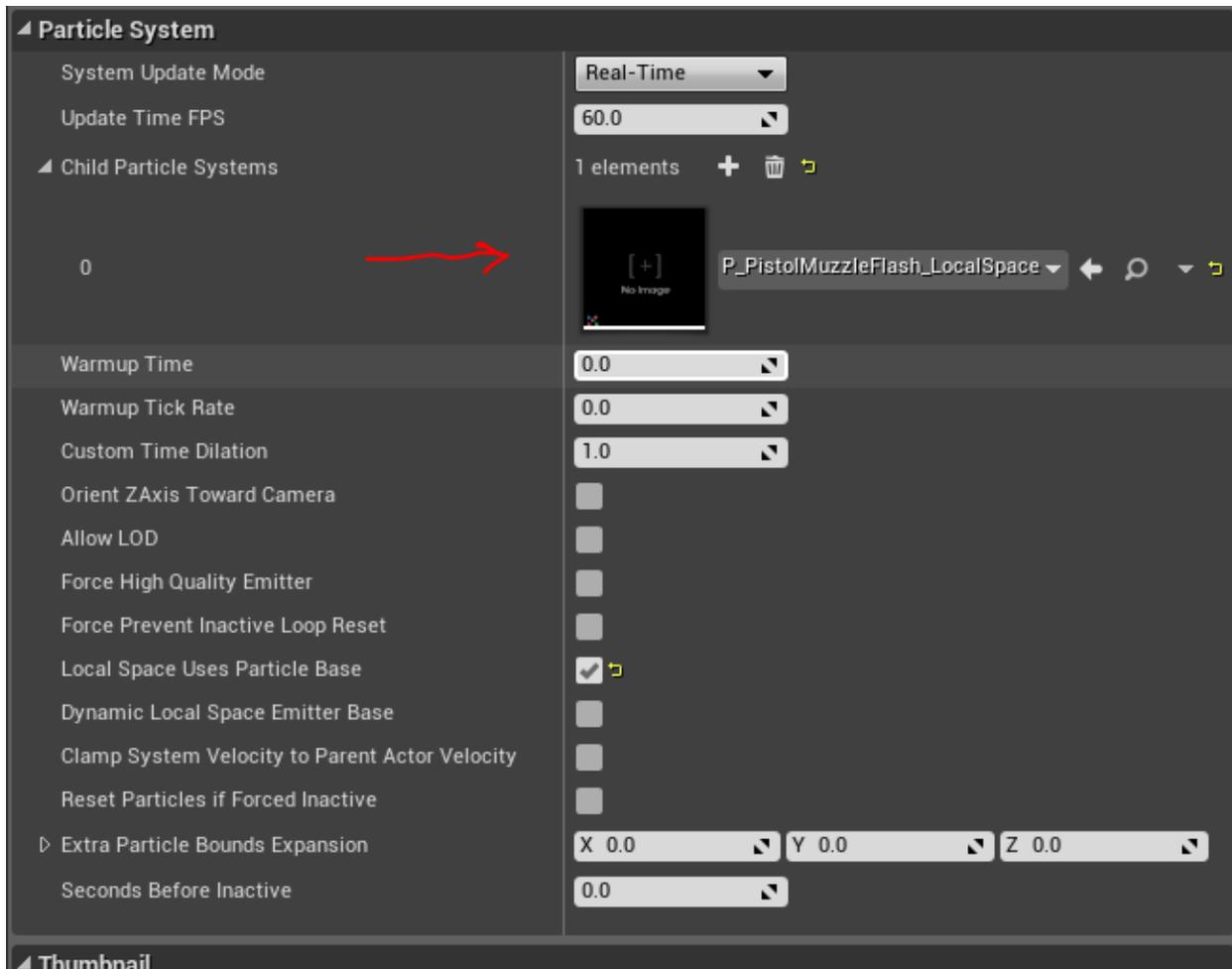
Particle ‘Base Space’ is basically how Atlas does ‘World Space’ behavior for particles when on a moving ship. A fast moving ship will leave particles behind in World Space, but ‘Base Space’ allows those particles to stay in the coordinate space of the ship and have the particles behave in a realistic manner relative to players on board the ship.

For World Space particles, it’s just a matter of setting:

1. **LocalSpace=true** on the emitters
2. **bLocalSpaceUsesParticleBase=true** on the particle system

#### Local Space & Base Space together:

Because Base Space overrides Local Space in this way, if a visual effect requires both Base Space and Local Space emitters, it must be separated into two particle systems - one for Base Space only and one for Local Space only - using new child particle system spawning functionality. So this means that the Base Space Effect will be the main particle system and it will point to a ‘Child Particle System’ like below. When the Effect is triggered, only the Main Base Space Emitter needs to be called and both will trigger.



**Important Note:** This technique is useful for VFX that fire in a burst (like a Muzzle Flash) and need to be aligned properly with the parent it is assigned to (like a Weapon), but if the effect needs to emit over a period of time, the BaseSpace Emitter as set up above will not move with the parent object (it will stay positioned and oriented to where it was initially fired). To have Base Space Emitters emit over a period of time and stay connected to the parent object as its position moves, see below under “Moving Emitters or Projectile Trails”- although in this case, the orientation will not update if the parent rotates, etc. An Emitter that emits particles over time and stays connected to the parent as it animates and stays aligned in direction with a rotating parent object is currently not supported.

- **Examples**

- ParticleSystem'/Game/PrimalEarth/Effects/Particles/Weapons/RocketLauncher/P\_RocketLauncher\_Projectile.P\_RocketLauncher\_Projectile'
- ParticleSystem'/Game/PrimalEarth/Effects/Particles/Weapons/WeapPistol/Muzzle/pistolMuzzle\_01\_TPV\_SpaceBased.pistolMuzzle\_01\_TPV\_SpaceBased'
- ParticleSystem'/Game/PrimalEarth/Effects/Particles/Weapons/WeapPistol/Muzzle/pistolMuzzle\_01\_FPV.pistolMuzzle\_01\_FPV'
- ParticleSystem'/Game/PrimalEarth/Effects/Particles/Weapons/WeapPistol/Muzzle/pistolMuzzle\_01\_FPV\_SpaceBased.pistolMuzzle\_01\_FPV\_SpaceBased'

- ParticleSystem'/Game/PrimalEarth/Effects/Particles/Weapons/WeapPistol/Muzzle/pistolMuzzle\_01\_TPV.pistolMuzzle\_01\_TPV'
  - ParticleSystem'/Game/PrimalEarth/Effects/Particles/Weapons/RocketLauncher/P\_RocketLauncher\_Impact.P\_RocketLauncher\_Impact'
  - ParticleSystem'/Game/PrimalEarth/Effects/Particles/Dust/PS\_DFootImp\_MED.PS\_DFootImp\_MED'
- 

**Note:** The pistol effects above are split into two particle systems: one that is true Local Space and the other using Base Space.

---

## Moving Emitters or Projectile Trails

For Projectile Trail particle systems or Emitters that need to follow the parent object, instead of `bLocalSpaceUsesParticleBase=true`, set:

**“Dynamic Local Space Emitter Base”=true.**

Projectile trails and Emitters that need to follow the Parent Object needed a different bool because they have to properly update a distinct emitter location as the projectile moves relative to the base. Test the rockets to see that they have correct local trail on a moving ship.

- **Example**

- ParticleSystem'/Game/PrimalEarth/Effects/Particles/Weapons/RocketLauncher/P\_RocketLauncher\_Projectile.P\_RocketLauncher\_Projectile'

---

**Note:** Projectile Trails still need the `UseLocalSpace=true`, and `spawn-per-unit` doesn't work with that of course, so we can just have bigggg spawn rates!

---

## Summary

Every character- or structure-spawned particle effect should have:

- **bUseLocalSpace=true**
- **bLocalSpaceUsesParticleBase=true**

Any true Local Space emitters should be moved into their own child particle systems with

- **bLocalSpaceUsesParticleBase=false.**

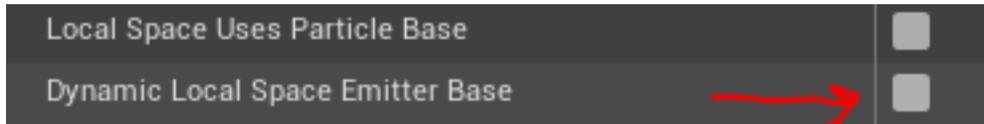
Any projectile trails or emitters that need to move with parent objects, should have:

- **bUseLocalSpace=true**
- **bLocalSpaceUsesParticleBase=false**
- **Dynamic Local Space Emitter Base=true.**

## Quick Run down

If you want to emit something that is oriented to the parent like a MuzzleFlash, but is just a QUICK BURST (not following the parent object over time) use the standard 'Local Space Uses Particle Base' If you want the particles to emit from a moving parent in BaseSpace continuously over time, but NOT FOLLOW IT'S ORIENTATION- use the 'Dynamic Local Space Emitter' option. Right now we don't have a solution for Base Space that will allow an emitter to follow the parent's motion AND orientation over time.

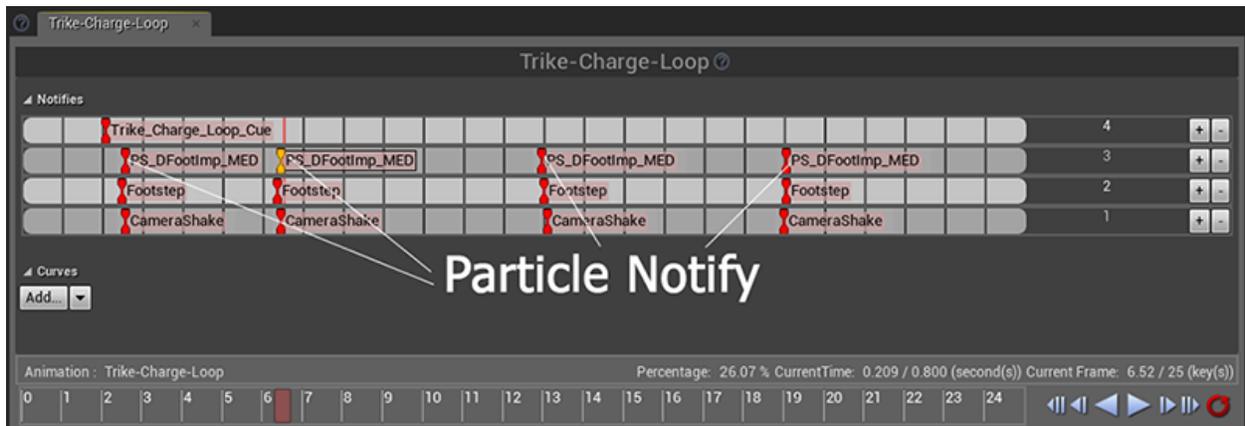
This is where the 'Dynamic Local Space; option is:



## Creature Footstep FX Setup

### Footstep Effects Setup

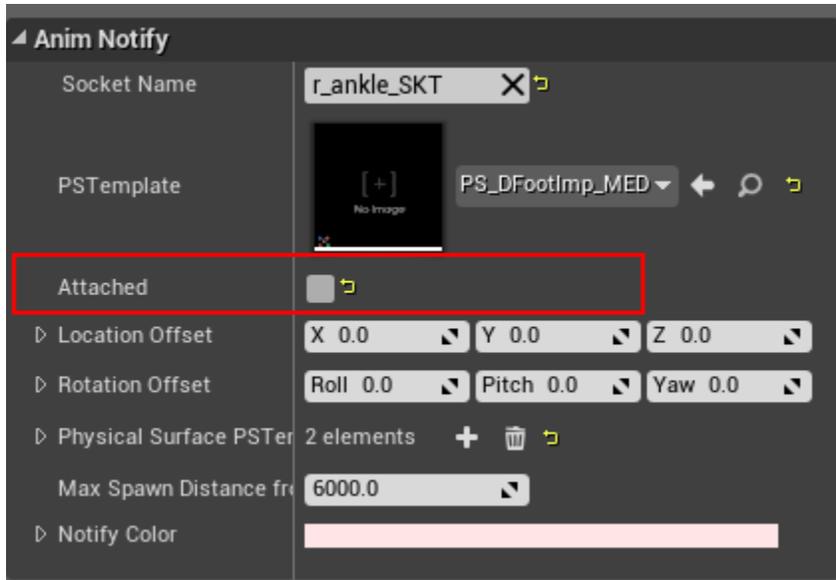
1. Add a Play Particle Effect notify



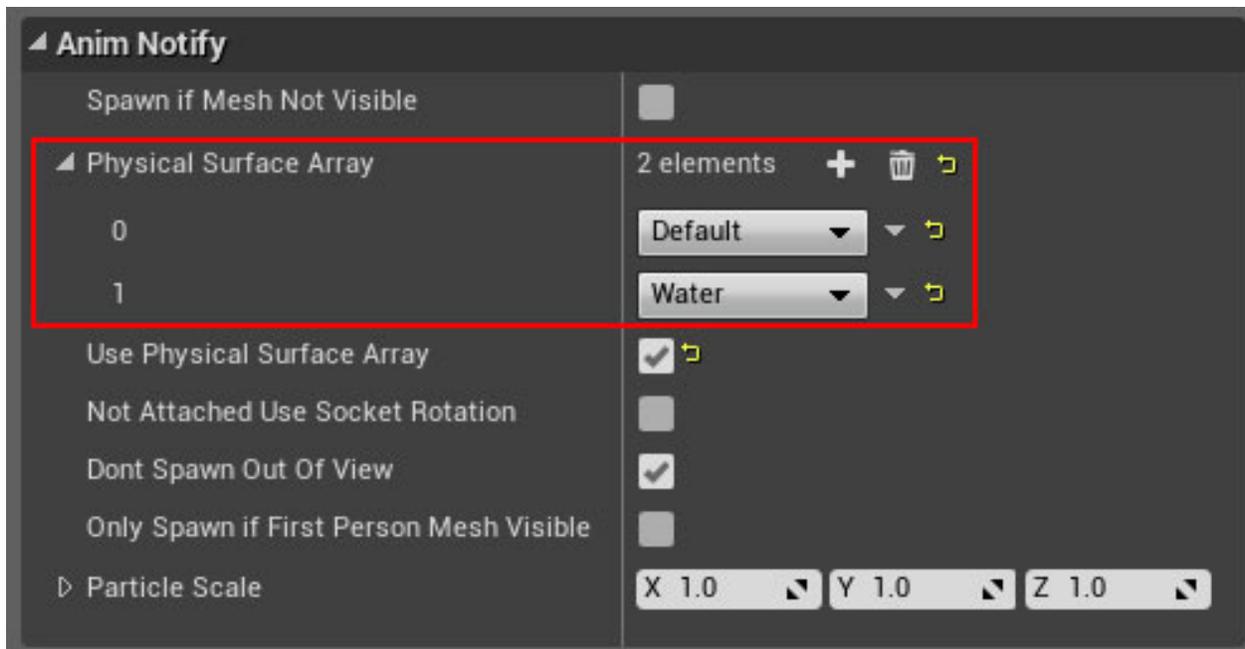
2. Attach to a **Socket/Bone Name** generally a bone at the base of the foot



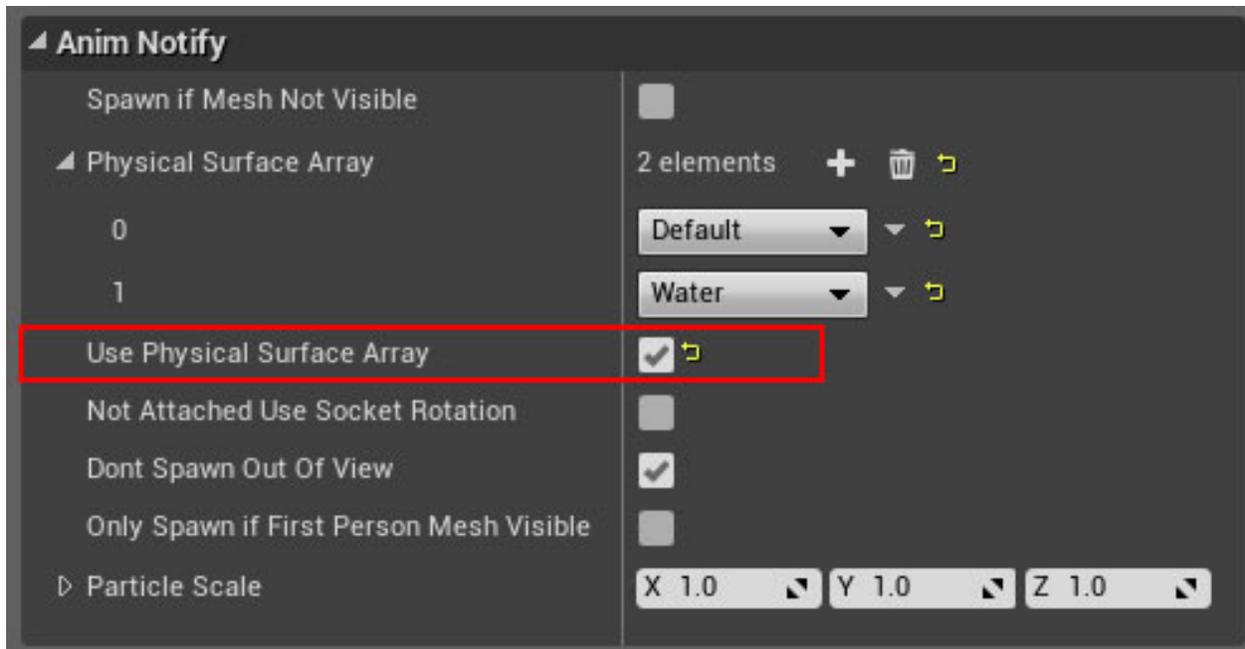
3. Add the default footstep particle
4. set **Attached** to **false** (uncheck) this makes sure the particle never takes any orientation of the bone/socket



5. Add two slots to the **Physical Surface PS Template** array slot [0] is for generic surfaces, 1 slot is for water, this is based on what you setup below. More could be added later. Add in the appropriate particle slot [0] is generally the same as the default



6. Add the same number of elements in the **Physical Surface Array** as in slot [0] is default. slot [1] is water.
7. Make sure to set **Use Physical Surface array** is TRUE



Adjust any **particle scale** as necessary if generic particles are used **AFTER** an appropriate based sized particle is added (7)

Repeat for all appendages that touch the ground and for all movement animations

**Walking footsteps are NOT needed for small creatures (dodo, penguin sized)**

---

**Note:** You can edit multiple of the same input at once by ctrl selecting multiple particle systems and editing them all at once. ie pretty much everything except for socket/bone name

---

## Fog in VFX Materials

### Overview

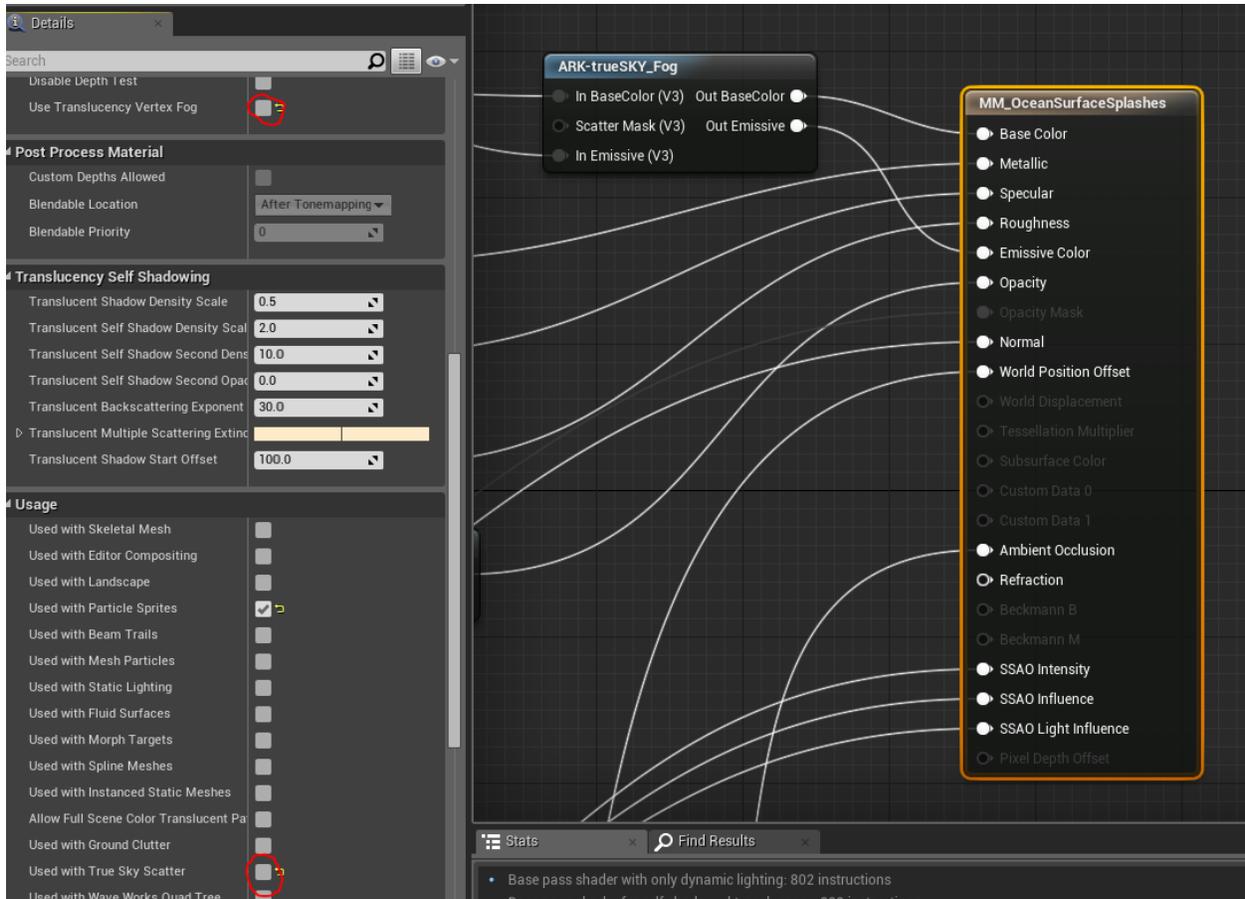
Currently standard TrueSky Fog in standard Translucent Materials don't match well to Fog on Opaque objects and Ocean in the scene at varying distances and lighting, and you get better visual results disabling the standard Fog and using a Custom Fog function.

### Implementation

To get best Fogging results for Translucent Materials you need to disable the standard Material Fogging :

- Uncheck "Use Translucency Vertex Fog"
- Uncheck "Used with True Sky Scatter"

Then add the **ARK-trueSKY\_Fog** Function and hook up the BaseColor and Emissive Color Inputs and Outputs as shown here:



## Mesh Particles

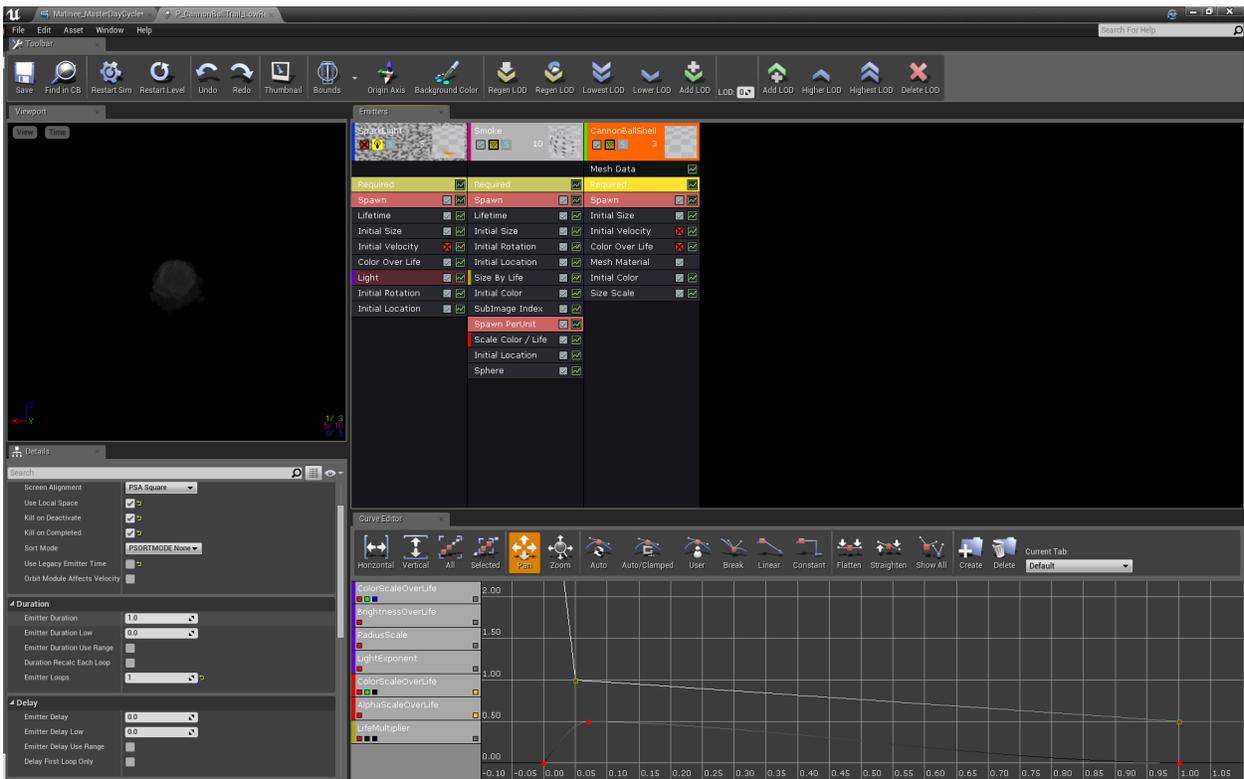
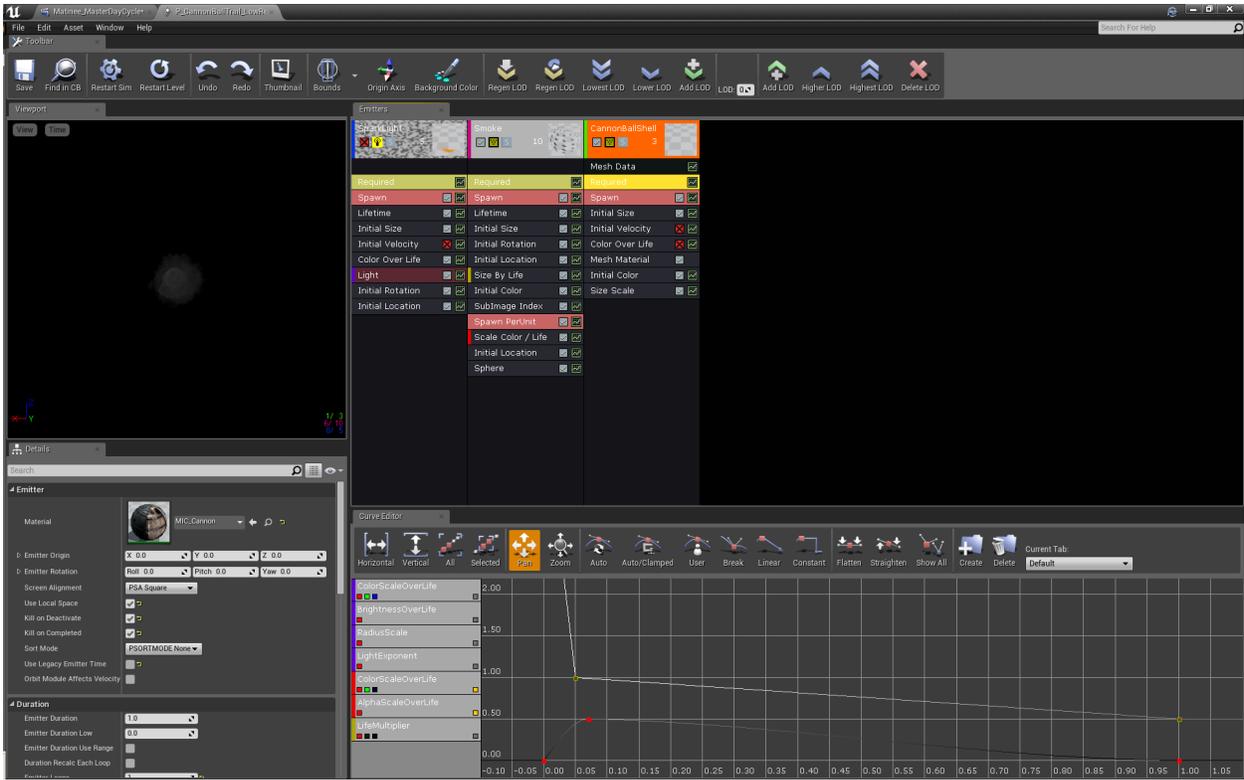
### Overview

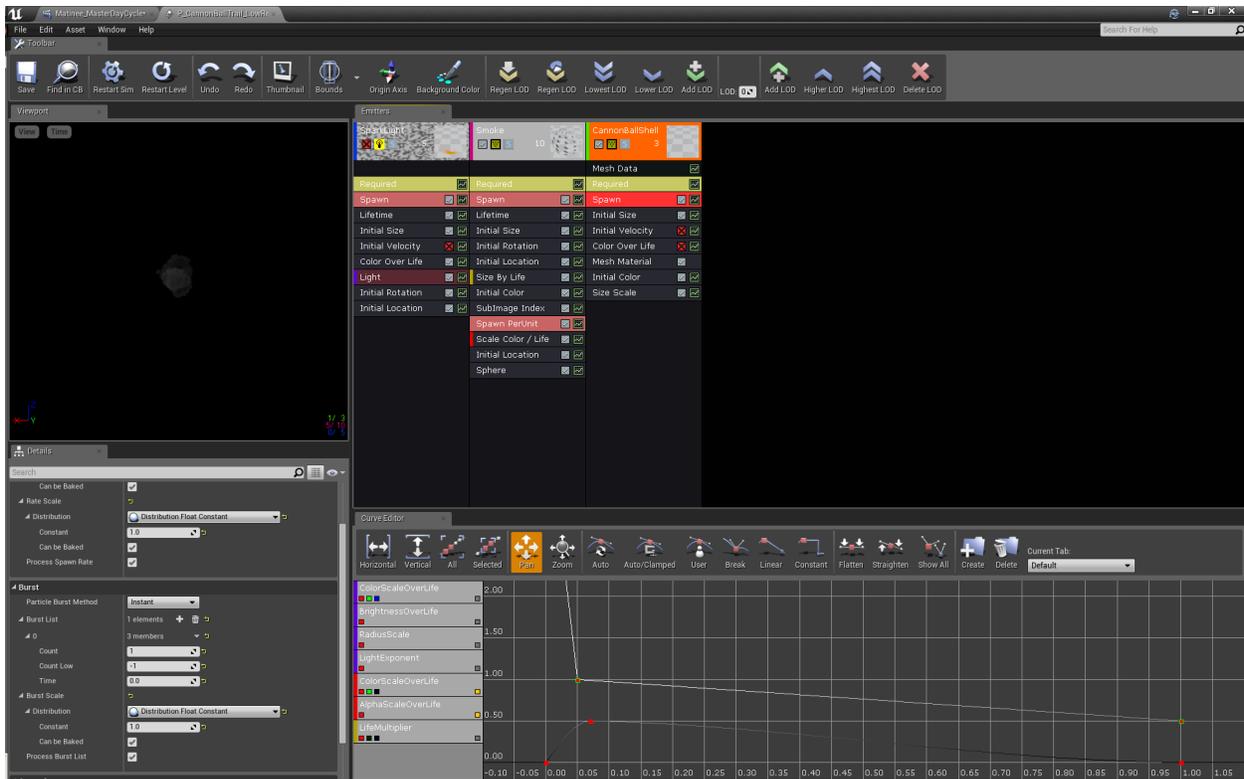
Information about Mesh Particles In Atlas

### Infinite Life Mesh Particles

If you are making a Mesh Particle in Cascade that has no Fade-In or Fade-Out and will live until the game ‘kills’ it, here are the correct settings to use:

**Warning:** Make sure you have **Kill-on-deactivated** and **Kill-on-completed** checked! (otherwise they stay around forever!)

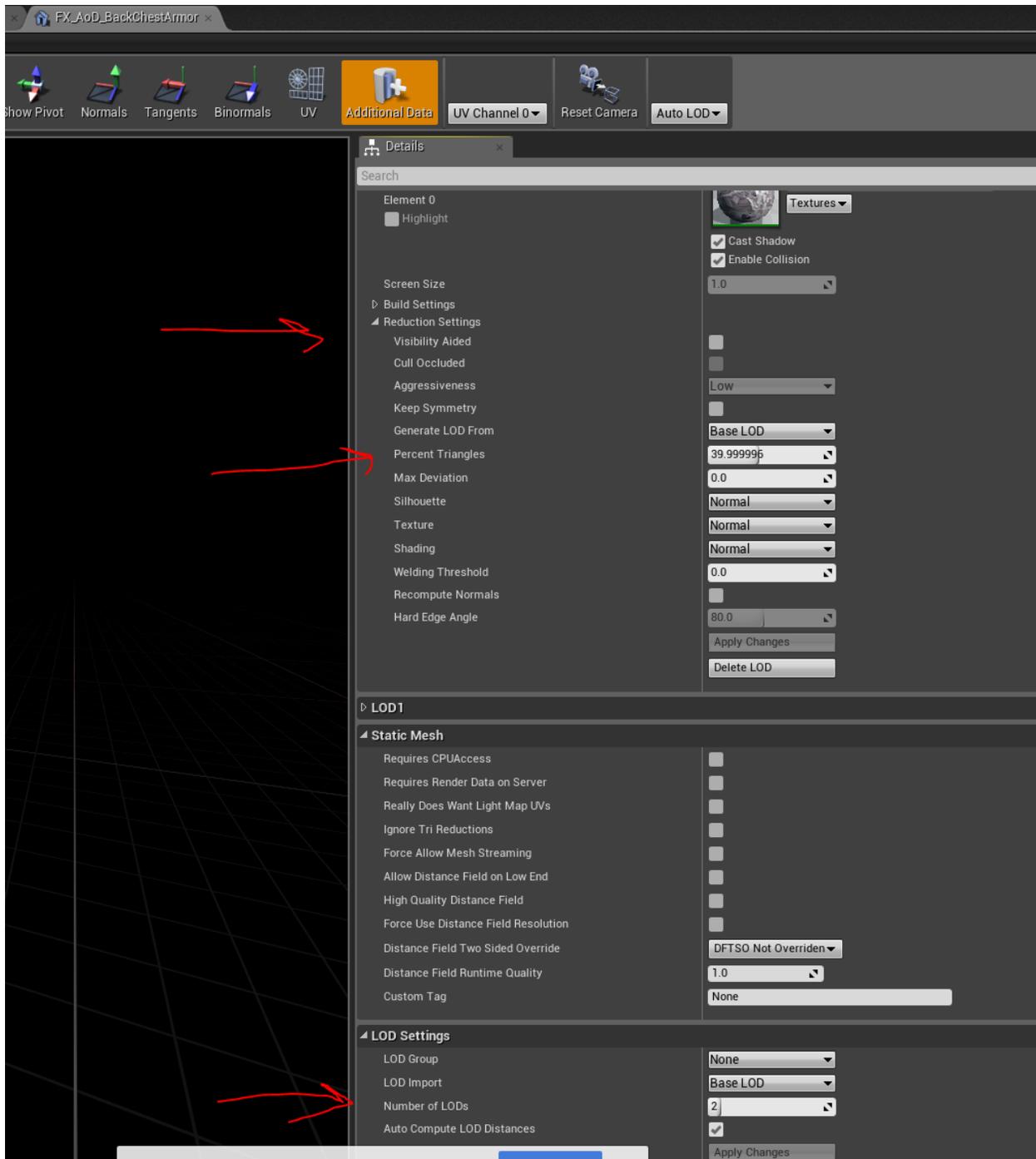




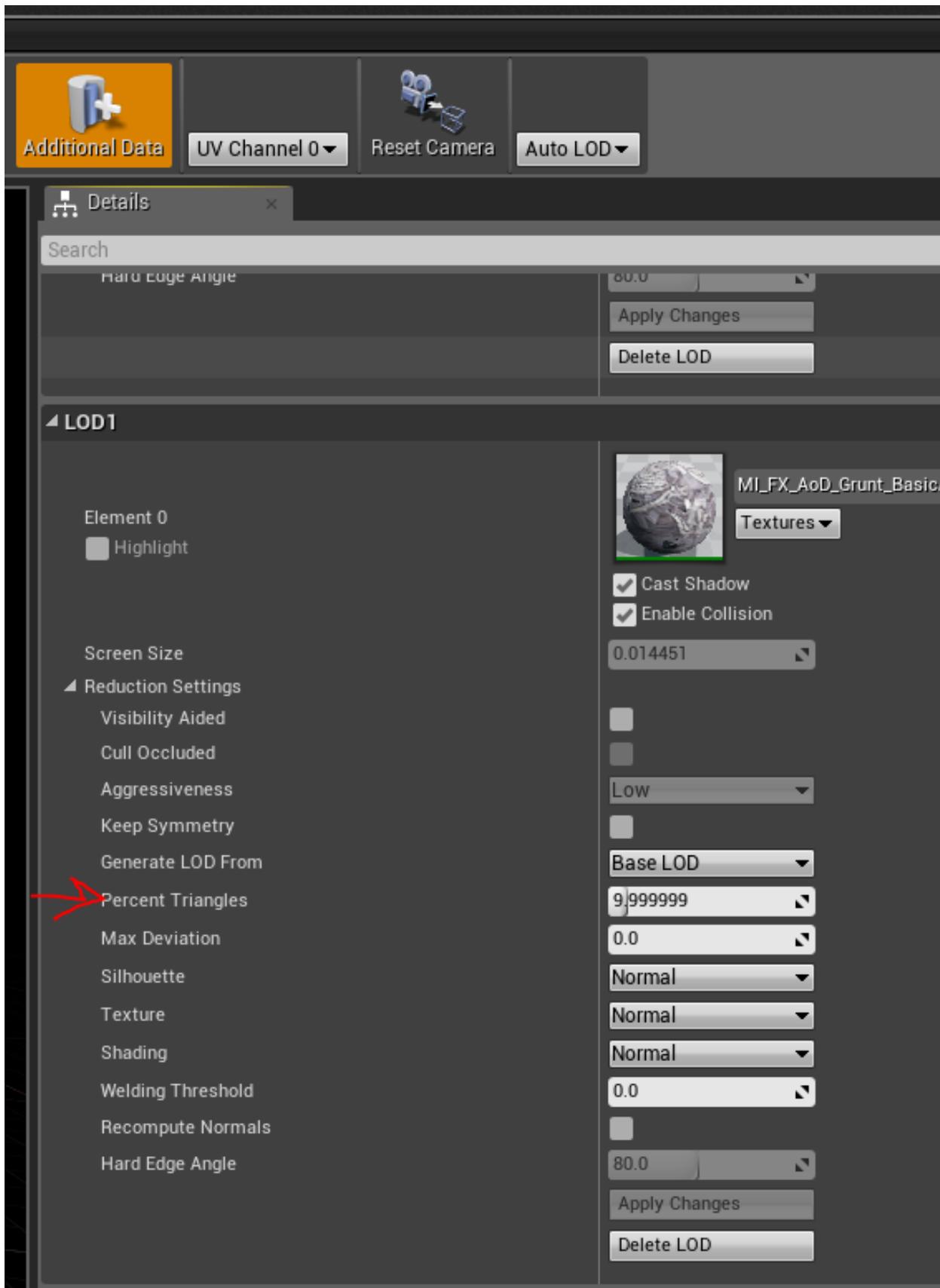
## Importing Meshes & LODs

When you import meshes into UE4 for use as Mesh Particles, generally you need to Poly Reduce and create a very low resolution LOD for streaming performance.

To do this, open the Mesh Editor and in 'Details' open '**Reduction Settings**', set the '**Percent Triangles**' to something that gives you the Poly Count you want (this uses SimplyGon to do the Poly Reduction) and set '**2**' for number of LOD's in the 'LOD Settings'. Then hit 'Apply Changes':



Once the new LOD shows up after hitting the changes are applied, set a really low 'Percent Triangles' for the LOD1 (which is only used for loading into memory/streaming- not for when it is actually used) :



## Misc

- For meshes that are less or equal to 300 tris, make sure to turn on “ignore tri reduction”
- When importing vertex color make sure you delete the mesh first BEFORE reimporting it.

## Animation Driven Mesh Particles

To keep as much of the production in the Art Department’s hands we have a process where all the Meshes that our characters use for things like reloading/etc are being accomplished with Mesh Particles instanced in AnimNotifies. There is a specific way to accomplish this!

Here is the basic order of accomplishing this:

- **Mesh**

- Get the final Mesh or Meshes that will be used to connect to the Animation. Generally they are the same for TPV and FPV, but there is always a chance they are different.
- Create new version of each of the meshes for use as a Particle Mesh. Rename them “FX\_whateverMesh” and put them in the Atlas/Effects/Meshes area.
- Create a new Material (and likely Material Instance) for the Mesh and enable compile for Particle Meshes. Put these in the Atlas/Effect/Materials area

- **Particle System**

- Create a new Particle System in Cascade and make a Mesh Particle emitter. Put this in the Atlas/Effects/Particles area
- **Bring in the Mesh you created into the Particle System and create a Mesh Material Node and assign the Material**
  - \* The Mesh relying on its own assignment or overridden in the ‘Required’ node will not work!!

- **Animation**

- **Open the Animation(s) you need to add the Particle Mesh to. This is usually named something like “\_Montage”**
  - \* There are generally two versions- a TPV one and an FPV one
- **Add AnimNotifies and hook up the Particle Meshes where they need to be triggered on and where they need to be**
  - \* Point the AnimNotifies to the appropriate Particle Mesh and assign it to the proper socket/bone
  - \* Generally the proper Socket/Bone was created or set up by the Animator to work properly with the Mesh, but there is a chance you might need to create a socket and/or position/rotate it or tweak it some.
  - \* There are a bunch of options that need to be checked/set a certain way to make this work properly and have the Mesh Particle correctly
  - \* **Here is a link to an example with everything set up correctly:**
    - **AnimMontage' /Game/PrimalEarth/Human/Male/Animations/**  
TPV\_NoWeapon\_NotifiesExample\_Anim\_Montage.  
TPV\_NoWeapon\_NotifiesExample\_Anim\_Montage'
    - Note: this has the extra StoppedEvent in the array there on the left, *as a failsafe*
- **Tweak/iterate- you may need to tweak position/rotation/scale of Mesh Particle.**

- \* If you are using a Socket that has other things attached to it, you should tweak in the Particle System - or create a new Socket and position/rotate as needed - but you should always check with the Animator to make sure you have the everything else correct before modifying.
- \* If you are using a Socket that was made for this, it is okay to modify as needed (but generally best to check with the Animator to make sure there are no issues with changing it)

### - Test

- \* You need to test both TPV and FPV versions **IN GAME!!** The animation editor doesn't really give an accurate view of how the Mesh Particles will be seen by the player, and sometimes things don't trigger as expected.
- \* Usually the Dev assigned to the task can give you notes on how to easily trigger the animation in a test scene

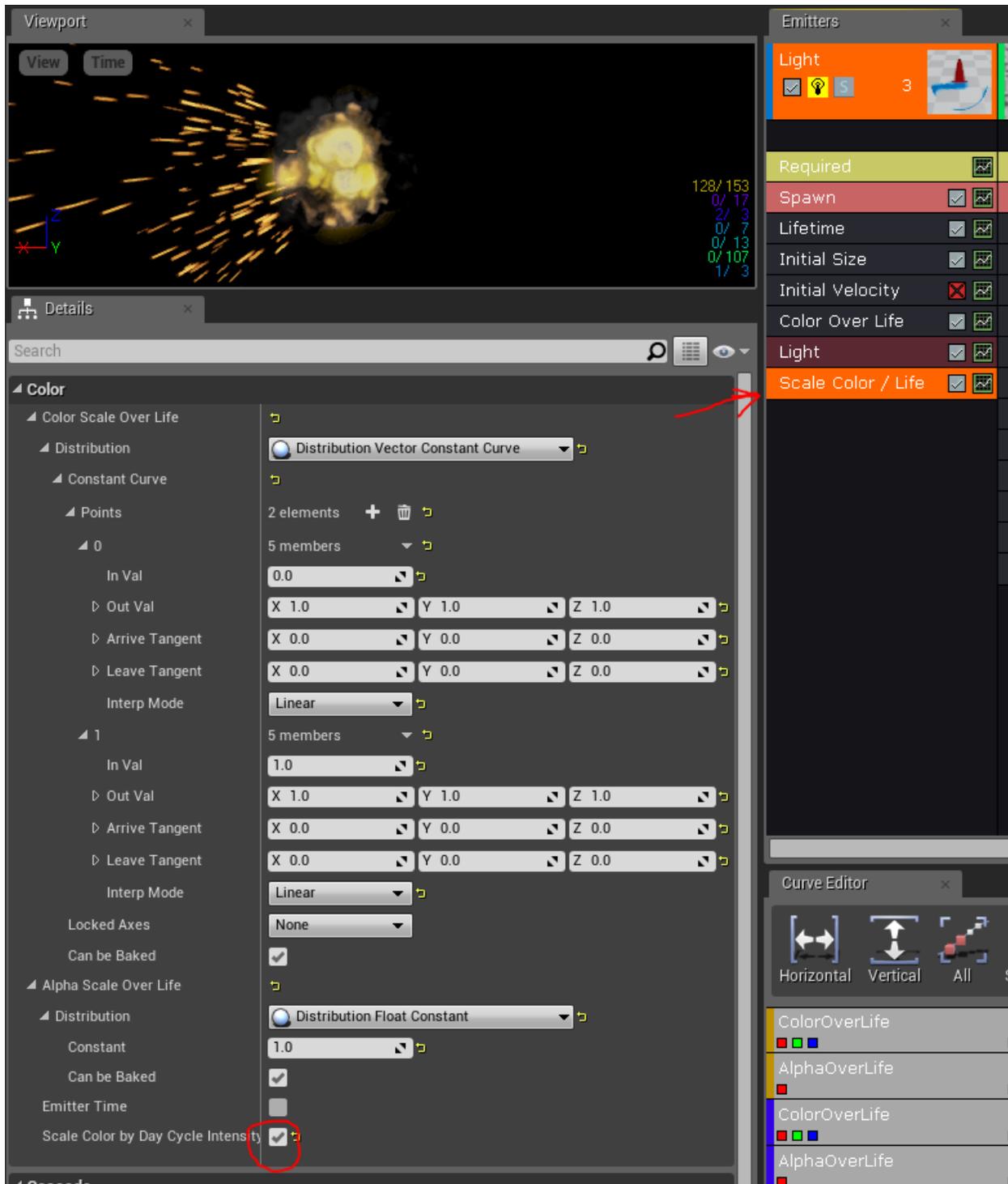
## Particle Lights

### Overview

Since it's difficult to create an intensity for Particle Lights that works well in all Day/Night lighting situations without blowing out too bright at Night or not being very noticeable in the Day, we have a Custom setting that helps scale the Light intensity based on the Intensity of the Sun/Dominant Light for that time of Day/Night.

### Implementation

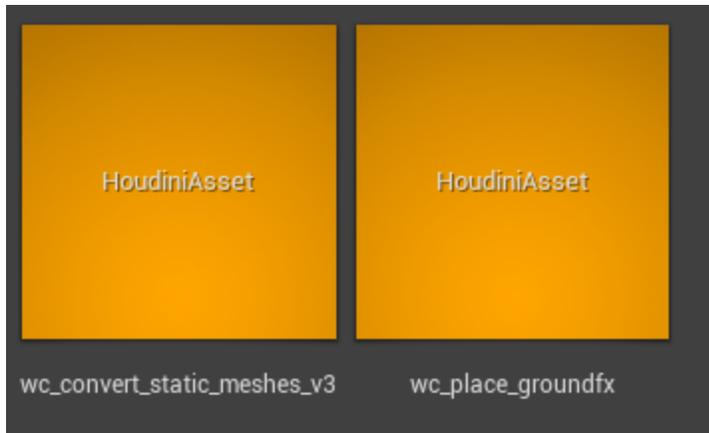
To get Particle Lights to have a better balanced intensity for Day and Night you need to add a "Scale Color Over Life" Node and set the "Scale Color by Day Cycle Intensity" flag like this. After doing this, make sure you test the result for varying times of day (as described in the "*Time of Day Material Function*" section).



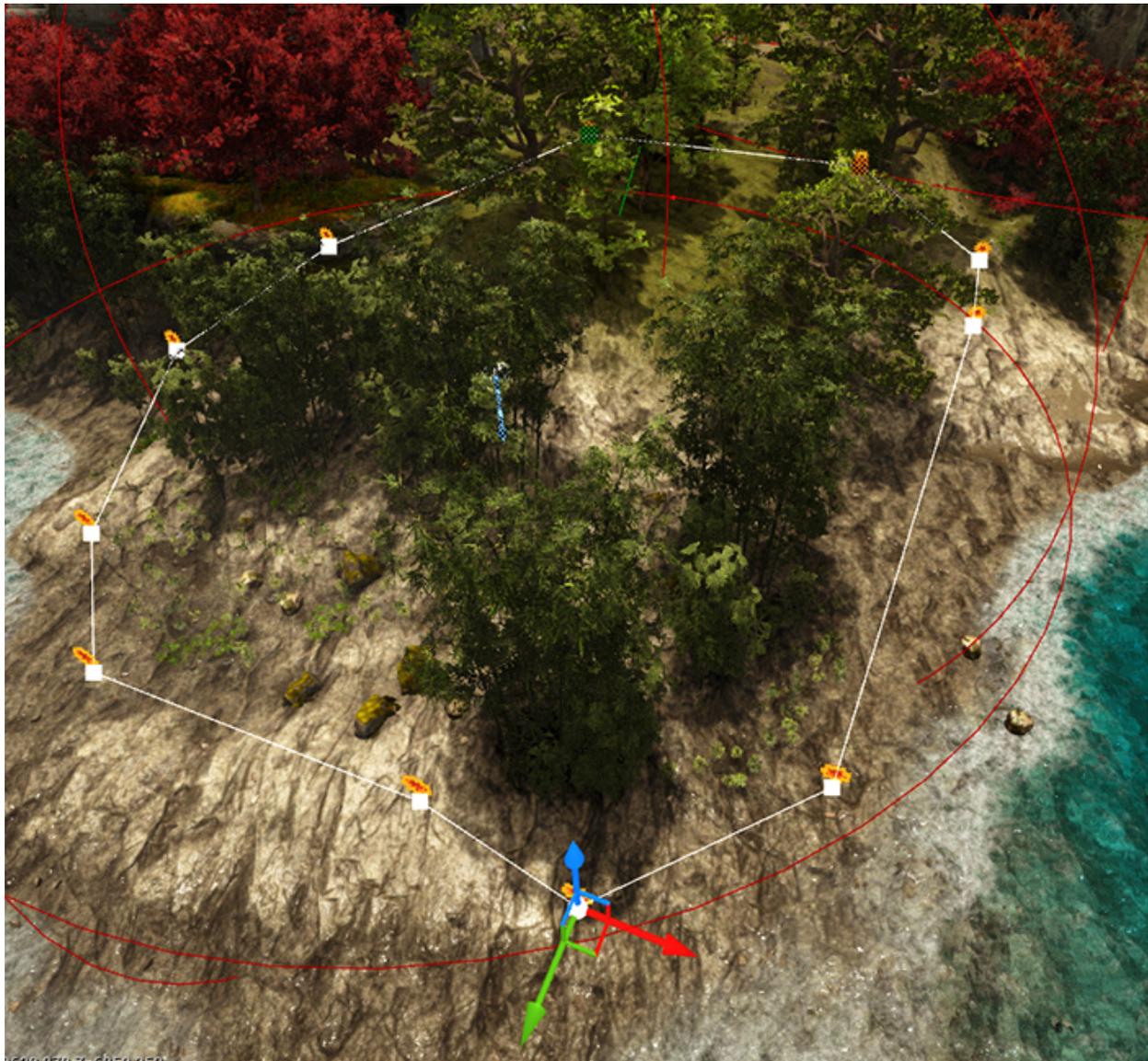
## Particle Placement on Landscape Tool

Tool was developed by **Kenny Lammers** for easier placement of emitters on the terrain for environment fx decoration.

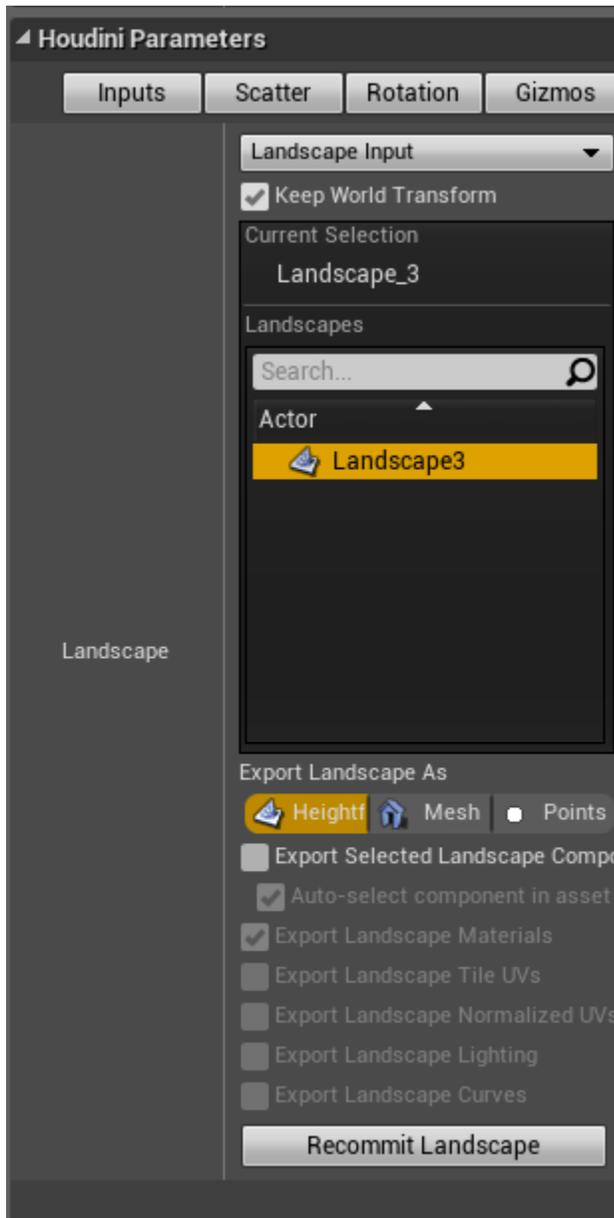
1. Navigate to the HDAwc\_place\_groundfx and drag it into the level.



2. drag into level. - You will see a box with 4 vert points, this is the beginning of the tool.
3. Select a point, hold ALT and drag to add points. - Note you can *ADD* and *Delete* points as necessary.



4. select landscape from the drop down to input, click landscape in the actor area “WAIT!”



5. Hit Scatter Tab, check scatter points



- Wait! while it starts cooking

6. emitters are placed on the terrain in the area assigned.

- changing point count requires a re-cook.

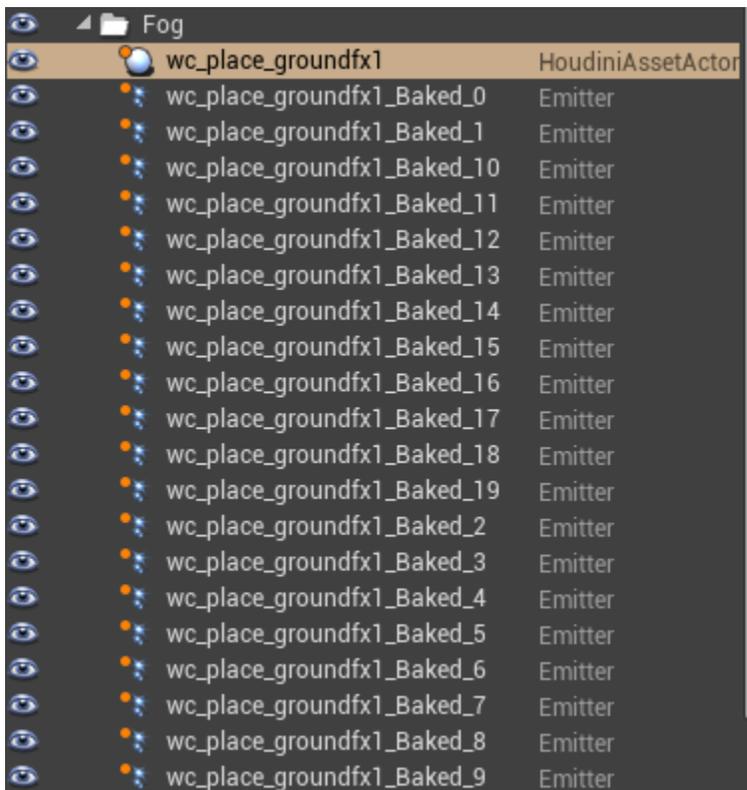
after happy with emitter placement, press gizmos

uncheck “show gizmos”

- wait to cook

once our happy and the gizmos are removed “bake to Actors”

Navigate to wc\_Place\_groundfx folder to see the baked emitters



Deleted HDA file



The Baked emitters will contain all the same emitter, so you will want to replace them with something else. (you may have some missing deleted files, hide and unhide the layer to remove the references)

## Particle Wind

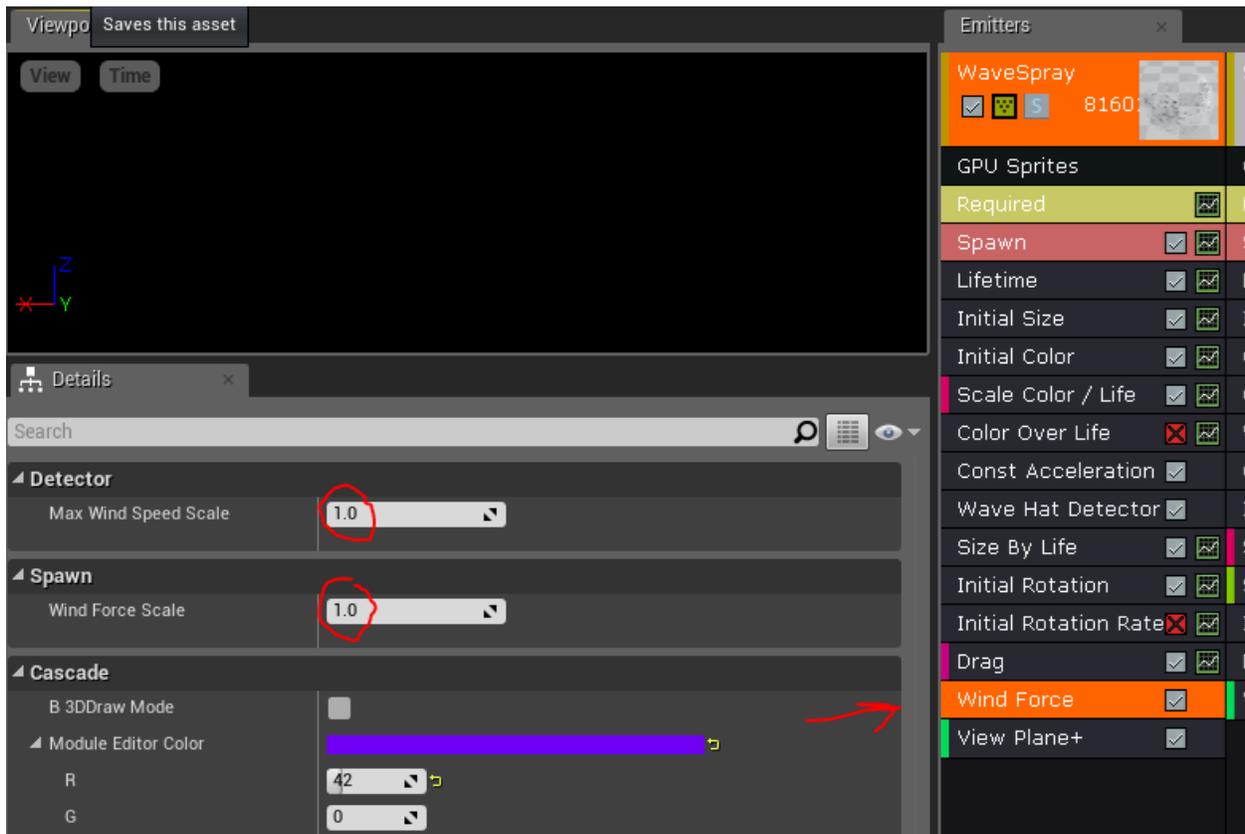
### Overview

Our WaveWorks Ocean simulation is driven by Wind and that same Wind can be used to drive Particles so Smoke/etc can automatically fly off in the same direction that sails/waves/etc are being blown

### Implementation

To make Particles be affected by the WaveWorks Wind, add a “Wave Force” node to your Particle Emitter and set the values for “Max Wind Speed Scale” and “Wind Force Scale”.

- “Max Wind Speed Scale” is how fast the particle will end up once at full speed compared to the Wind - “1.0” means 100% of the Wind Speed.
- “Wind Force Scale” is a factor for how quickly the Wind accelerates the the particles to help represent physical characteristics like mass and cross-section of the particles. Here is an example:



## Particles in Animations

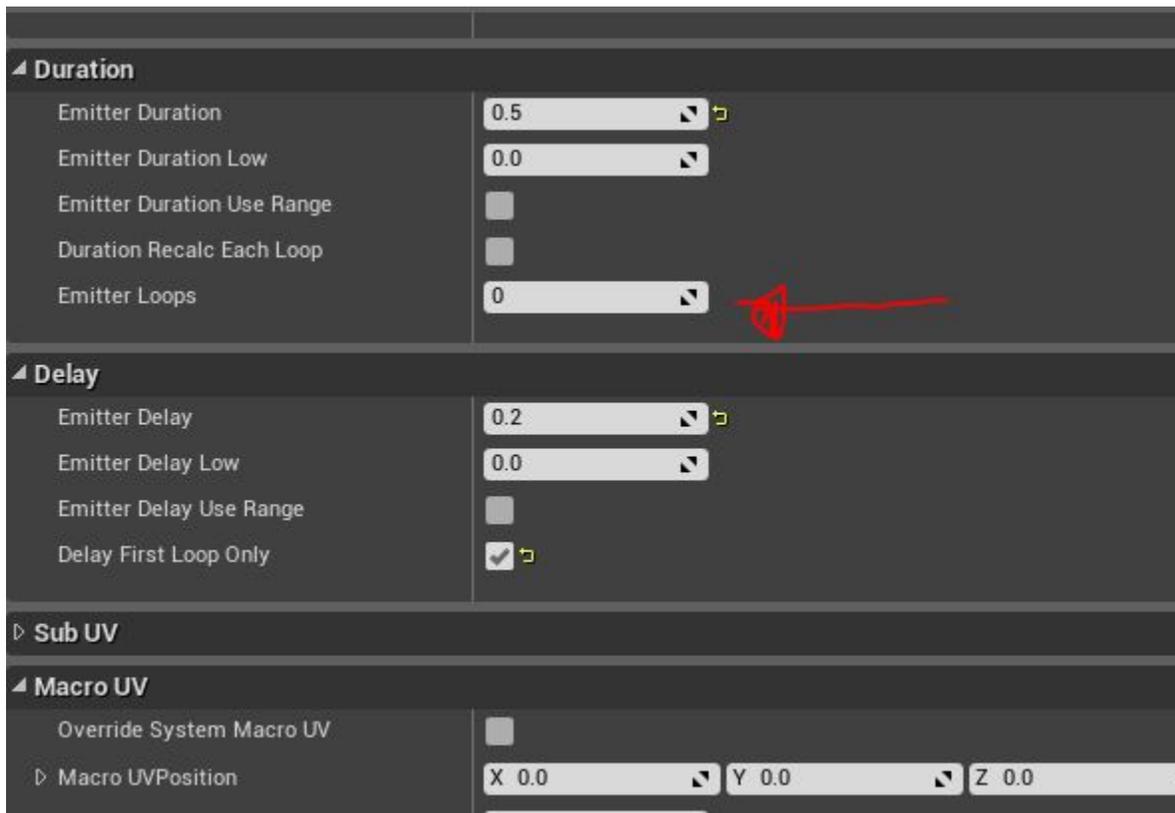
### Overview

Setup and issues with adding Particles to Animations

## Particles in Animations

### No infinite looping emitters

Particles that are used in animation should not have infinite looping emitters in them.



There is no way for an animation to stop a particle once it is playing. Therefore an infinite looping emitter will just continue to spawn and stack. The engine has been coded to look for animations that have infinite looping emitter and will kill/disable all particle systems in animations that have any infinite looping emitters in them permanently. go to “Particles not playing in Animations” for a simple fix.

**Warning:** For a Non-Looping ParticleSystem, ALL sub-emitters need to have non-Zero Emitter Loops!

If not, the ParticleSystem will be considered Looping, which means it will never STOP!

### Removing all looping Emitters

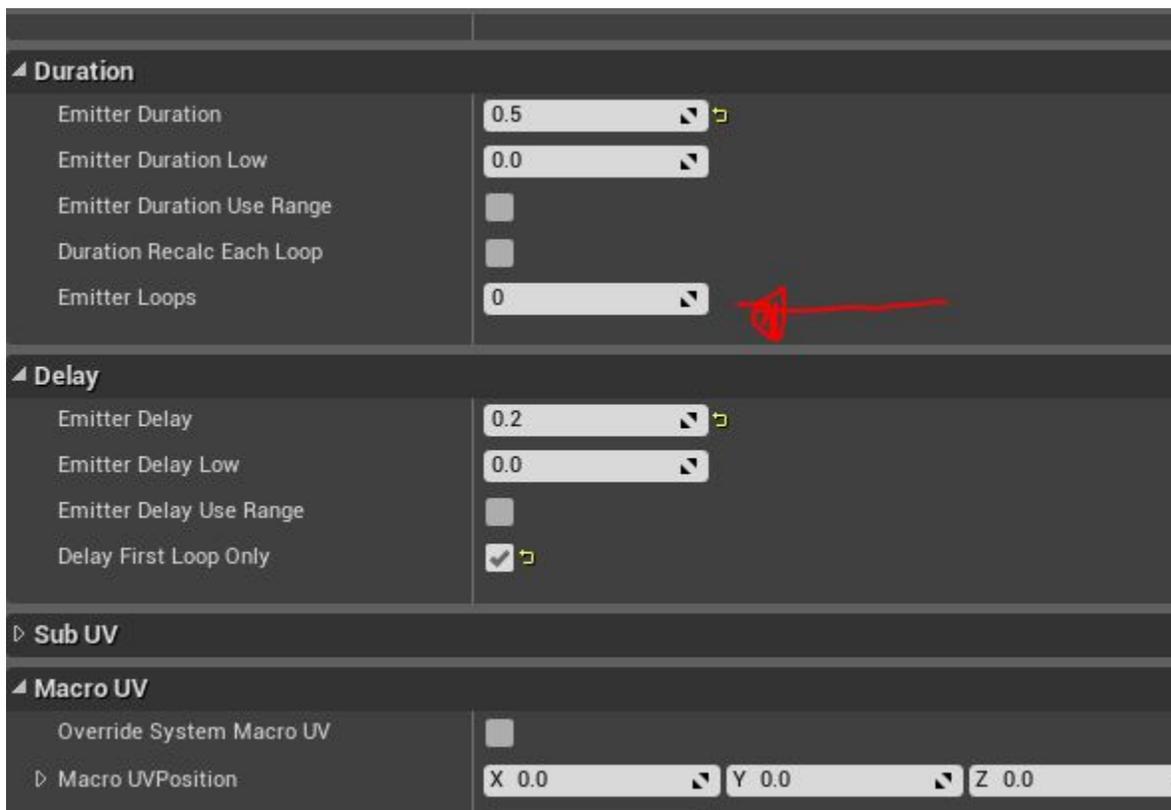
Disabled and collapsed emitters that are looping will still activate the looping flag on the particle system. Even though they are disabled and or collapsed they must be given a non zero loop value. Once you have done that you MUST

restart the editor to clear the Looping flag set in cascade. You can verify this by clicking on the empty black area inside cascade and looking at the “looping” check box. It should be unchecked, if not you may have a emitter with a zero loop value hiding somewhere.

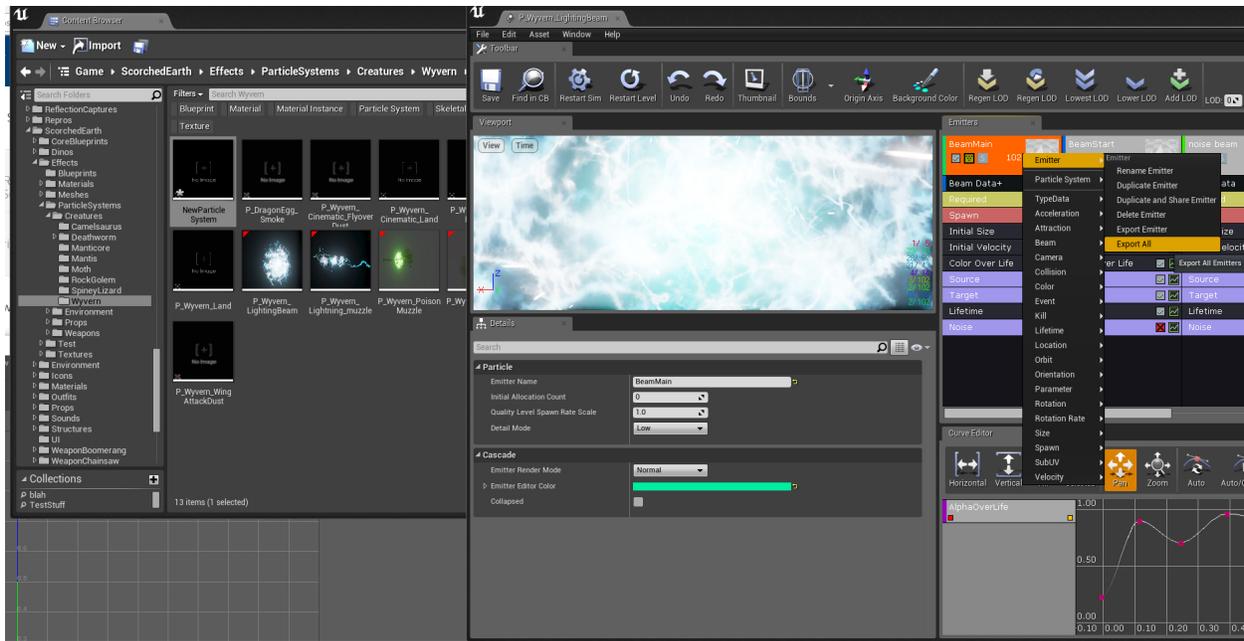
## Issues to be aware of...

### Particles not playing in Animations

One possible reason that a particle will not play properly in an animation if it has any infinite looping emitter in its particle system. Particle Systems that get added into Animations through notifiers **CAN'T** have emitters that infinite loops. There is code in the background that will check if the animation has infinite looping effects and will kill the entire particle and prevent the particle system from working properly in animations permanently. The particle will never play in the editor or in game after this happens.



When this happens you must make a new particle system and copy over all the emitters and change all infinite looping particles before putting the emitter back into the animation in order to fix the issue.



## Ribbons/Trails for Swords/etc

### Overview

Info and Caveats about using Ribbon Emitters in ATLAS.

### Reference

<https://docs.unrealengine.com/en-us/Engine/Rendering/ParticleSystems/Reference/TypeData/Ribbon>  
 <<https://docs.unrealengine.com/en-us/Engine/Rendering/ParticleSystems/Reference/TypeData/Ribbon>>

### Parameters

#### Max Trail Count

- Should be '1' - you are only creating one trail for a weapon. '30' would be if you have 30 parent particles flying around and they are the 'Source' and you want to emit trails from all of them.

#### Spawn Rate

- Is supposed to equate to 'Particles Per Second', so best guess is making the Spawn Rate 30 fps (with Rate Scale set to 1.0) might help the ribbon emission match up to the Animation better which likely updates at the lowest at 30 fps (not sure if it is playing at 30 fps or 60 fps).
- (Try to make sure you are emitting every frame, because the tessellation can help smooth it out, but those areas pop in/out when particles birth/die. So it needs a combination of higher emission rate and then tessellation enough to smooth out the in-between areas)

## Tiling Distance

- Is “The (estimated) covered distance to tile the 2nd UV set at.” So if you have ‘5’ you will be creating a new UV 0-1 Tile every 5 units in World Space. If you have it set to ‘30’ you will be creating a new UV tile once every 30 units of World Space. The Tiling is based on world units and not any parameterization of the Ribbon. So for less tessellation (and therefore polygons) you need a higher ‘Tiling Distance’ value.

## Distance Tessellation

- Step Size that can help round out some areas in-between for fast moving swishes.

## Ribbons UV co-ordinates

- You can fade ribbons with with standard particle point color/opacity with ColorOverLife in Cascade as long as that is hooked up in your Ribbon shader.
- **The other thing you can do is if you checked ‘Tiling Distance’ it creates a second UV set. The standard UV set (UV0) goes**
  - So for the Tiled UVs it might go from 0 to 5 (not 0 to 1 repeating over the ribbon) and your repeating texture just tiles nicely. This does allow for a better ability to select just the first tile by clamping the second UV to 0-1 to avoid a hard beginning edge, by the way.

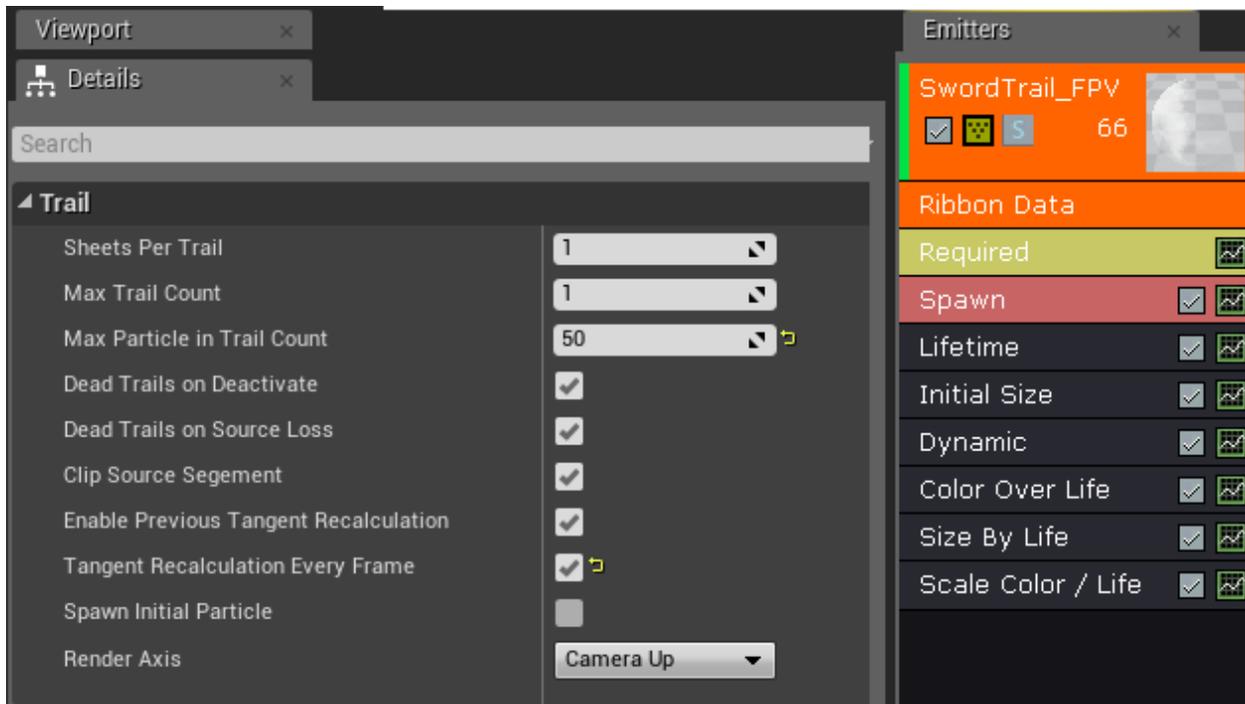
## Best Practices

### Ribbons Bounds Issues

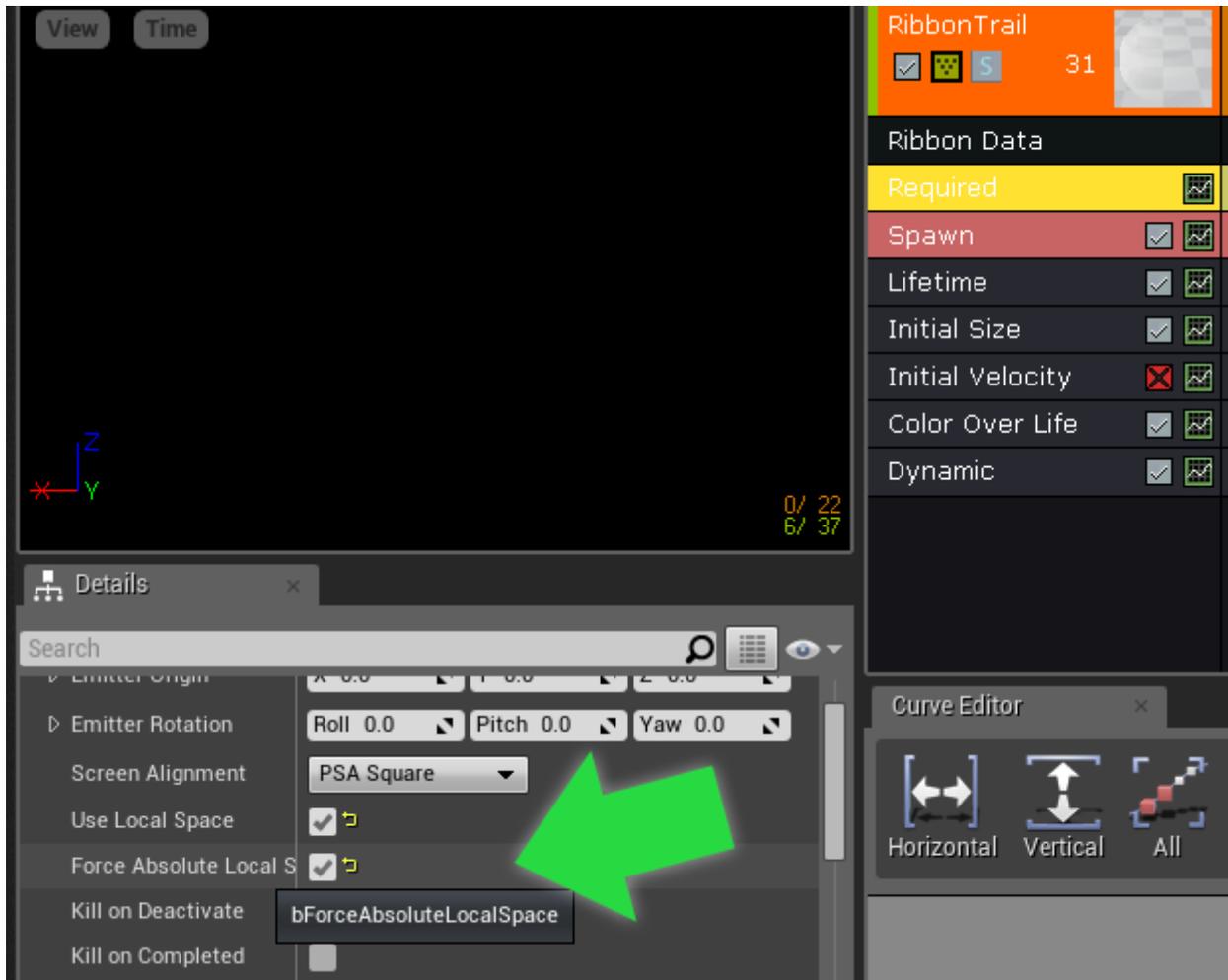
- Another issue that happens with Ribbons is they will not render due to the size of the game world without fixed bounds.
- In the emitter properties check “Use Fixed Relative Bounding Box” and set the dimensions to 1000 for X, Y and Z- or whatever is necessary depending on the size the effect takes up in World Space

### Best Working Settings

- Select the Ribbon module in Cascade and use these settings in the details tab for the best ribbon functionality in game.



- Use either Camera Up or Source up whichever works best for your fx and socket orientation



## Known Issues

- Ribbons on ships are still a problem and do not work correctly with base space.
- Use Local Space and Force Absolute Local Space without local space uses base space enabled.

## Ribbons with projectiles (World Space vs Local Space)

- It was noticed during testing that Ribbons had segmented, hard edges when they were emitted in 'World Space', but when set it to 'Local Space' the ribbons silhouettes were nice and smooth. But the 'Local Space' caused some other issues, so test and make sure it works properly for your use.

**Warning:** read section about *Base Space* for creating Ribbons for Projectiles

**Note:** Using Ribbons for Projectile Trails is normally frowned upon if there ever might be a large number of projectiles in a scene at once due to performance concerns.

## Skel Vert Loc Particles

Using the Skeletal Mesh location module in Atlas requires a quick fix and some blu print work to get the correct results in game.

## Cascade Settings

You will needs to setup the module in cascade to work in game by doing the following:

1. add the Skel Vert/Loc surf location module to your emitter
2. Set the skel mesh actor reference
3. Set the parameter name. This will also be used in the BP

---

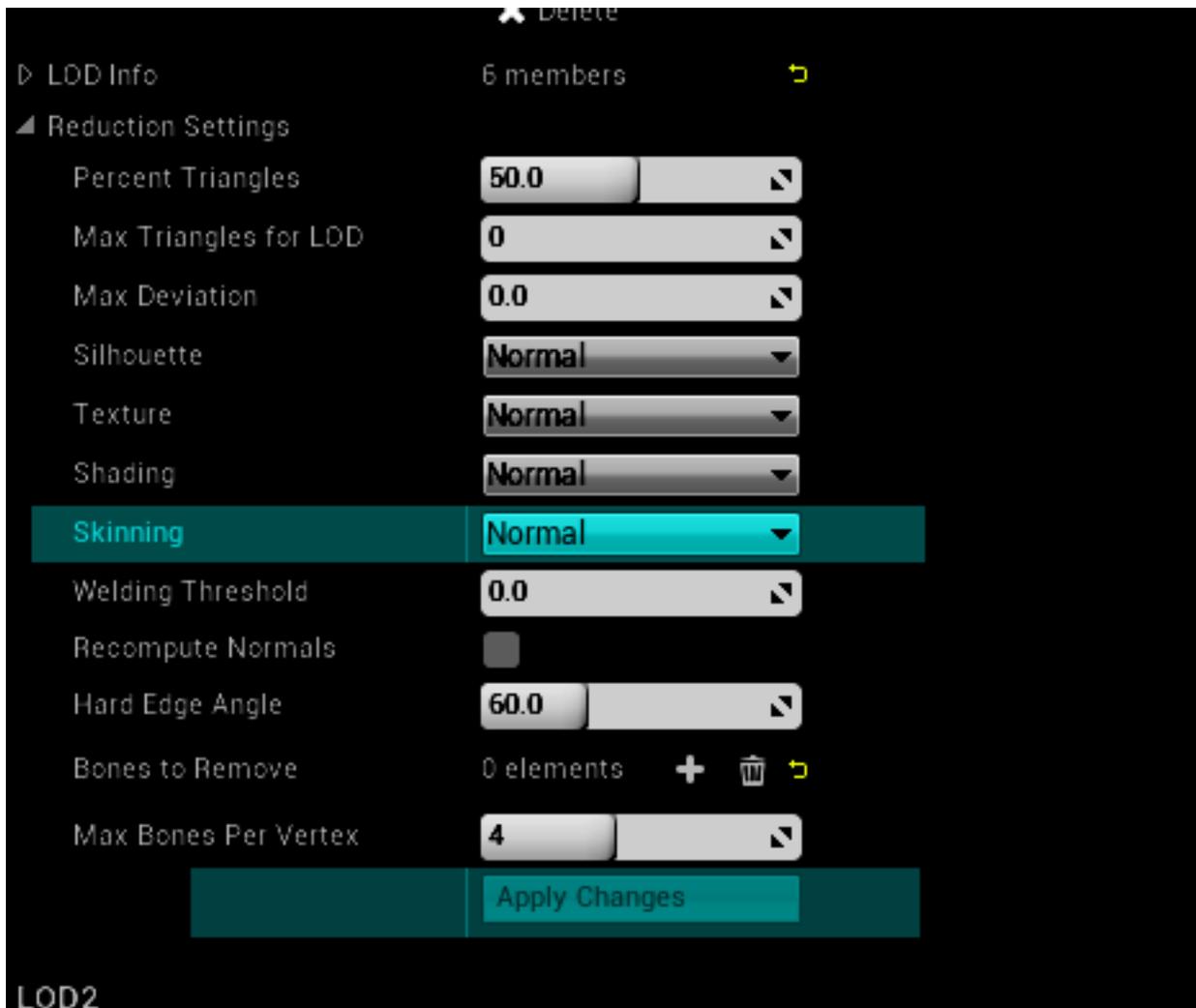
**Note:** to preview in editor you can follow this thread <https://answers.unrealengine.com/questions/7423/questionrequest-how-use-skel-mesh-vert-location-fo.html>

---

## Skeletal Mesh Fix

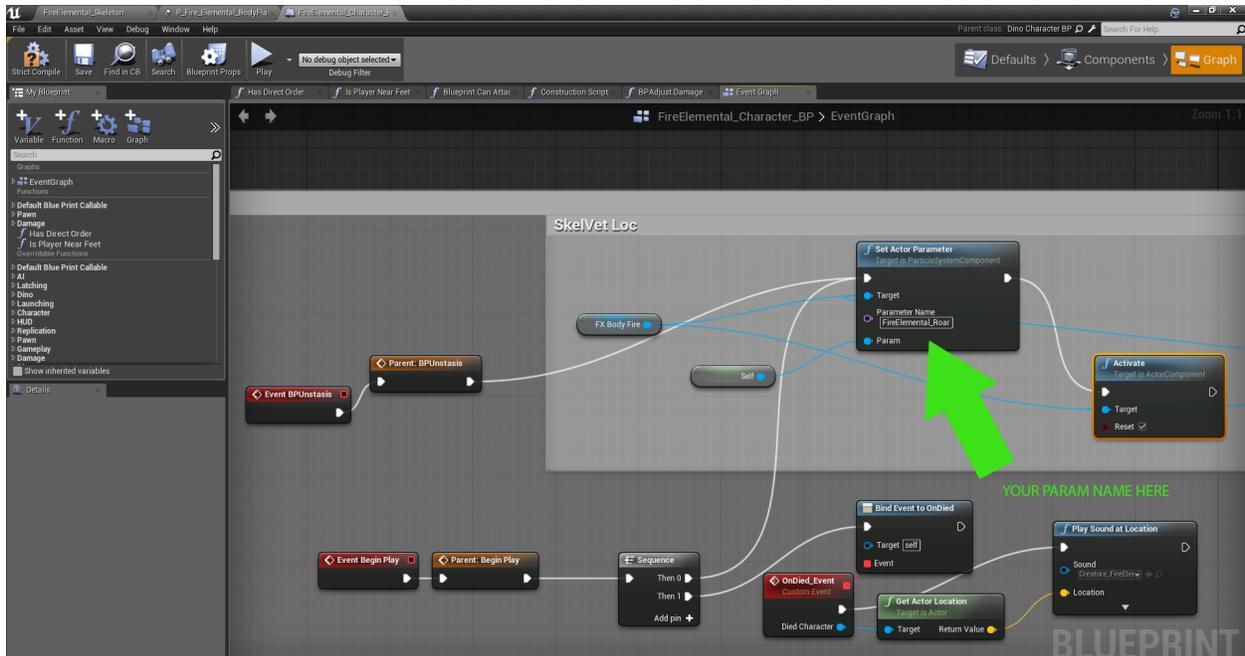
By default the particles will not spawn as expected this is a UE4 bug or a issue with skinning with too many bone influences. When enabled particles with spawn with strange offsets on the joints and in empty areas between joints .

To solve this issue open up the skel mesh and to into “Reduction Settings” set the default skinning from Normal to low. Hit “Apply Changes” now set it back to “Normal” and apply changes. This will fix the skel vert spawning positions.



## Blueprint Setup

For the particles to spawn and continue to spawn on the actor you will need to add some references in the creature's blue print. Open up the BP of the creature you are working on. Go into the event graph and add the nodes in the image below. Use the parameter name that you setup earlier in cascade and save.



## Specific Particle Setups

### Overview

Info on setting up particles for different implementations

## Specific Particle Setups

### Placement/Spawning/Destruction Effects for Structure Objects

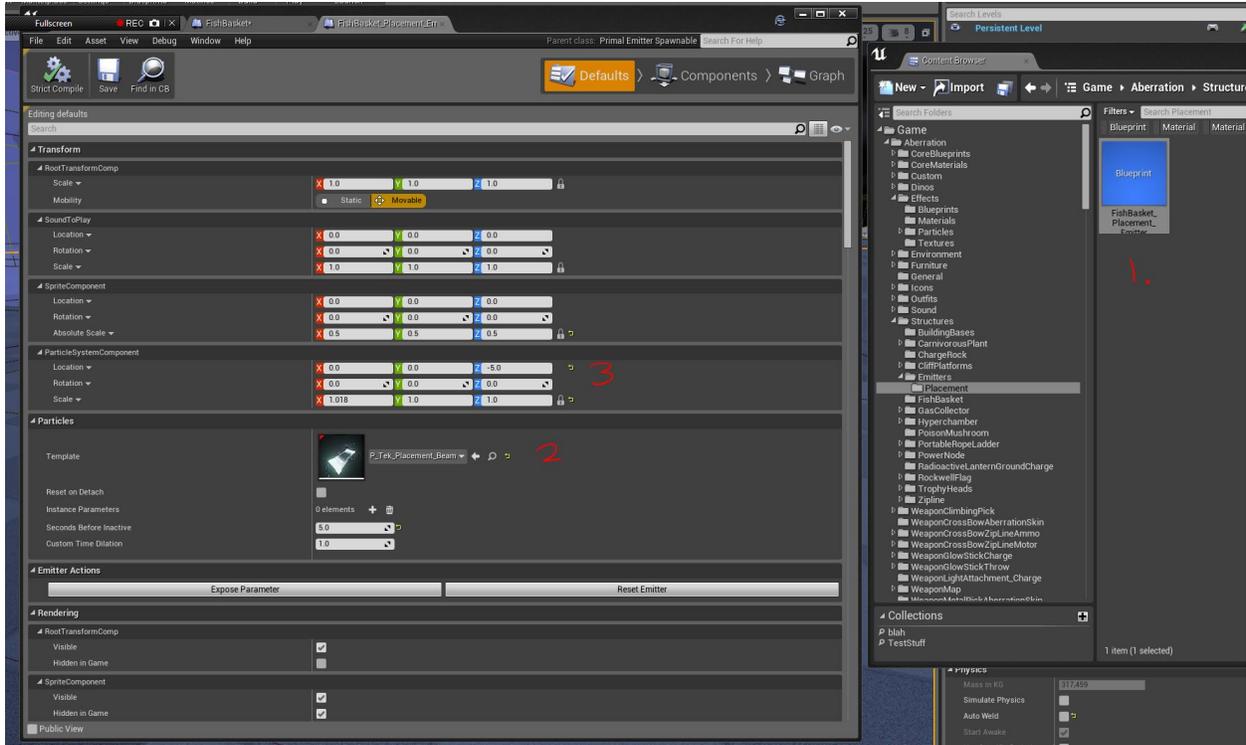
Create/Copy a Emitter Blueprint (possibly from a preexisting placement emitter) **(1)**

Plug in the correct particle in the **template (2)**

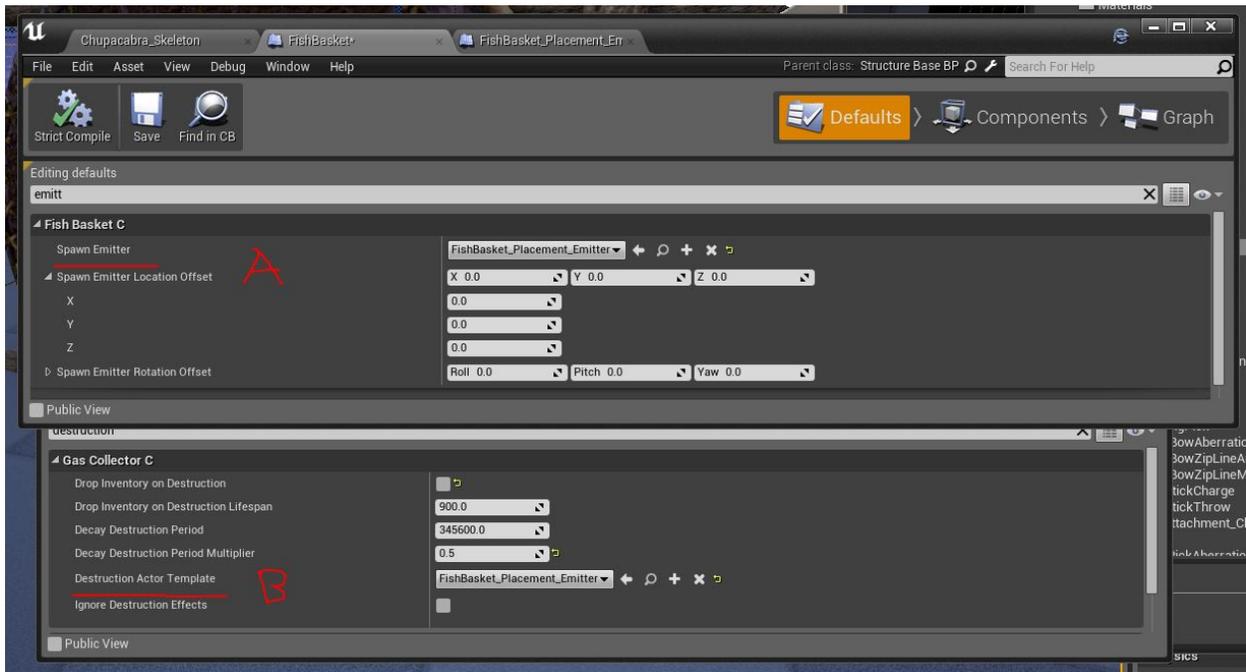
Make sure **Location, Rotation, and Scale** is as desired **(3)** **NOTE:** this will play at the **origin location of the structure/object** if everything is set to default

Plug the Placement emitter in its appropriate structure blueprint in the as the **Spawn Emitter (A)**

Plug the Destruction Emitter Blueprint in its appropriate structure blueprint as the **Destruction Actor Template (B)**  
(Particle Emitter Blueprint setup)



(Placement/Spawn and Destruction Emitter Setup for Structures)

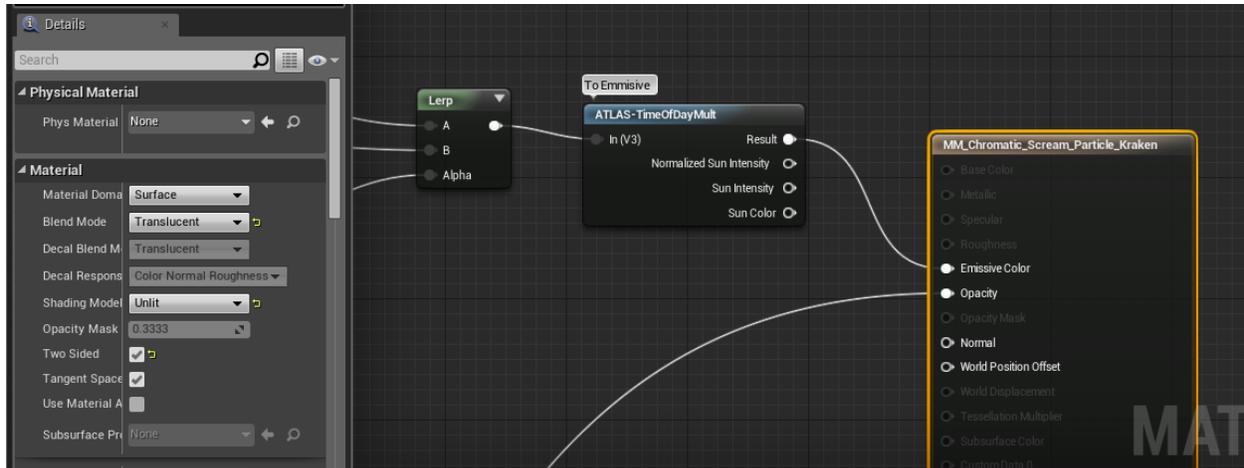


## Time of Day Material Function

For most 'Physical VFX' (Smoke, Water, Blood, etc.) materials in Atlas, we want to rely on the standard UE4 lighting functions but for Non-Physical Emissive VFX Materials (like Magic, Fire, etc.) it can be a challenge to make the object 'fit into' the environment properly throughout different lighting conditions that happen throughout the day

without blowing out super bright at night or being too dim during the day.

To handle this, we have the **ATLAS-TimeOfDayMult** function (this has replaced the **ARK-TimeOfDayMult** and **ARK-EyeAdaptationCorrection** Functions). To use this function in your Material, you just feed your Emissive Color into the input and feed the ‘Result’ output into the Material’s “Emissive Color”:



To use it properly, you should create a Material Instance (MIC) that you point to this Material and the MIC is what is used as the Material. This exposes these parameters that help tune the result during the Day and Night:



Tune the ‘Intensity’ values to tweak overall ‘brightness’ of the object during the Day and Night and tune the ‘Saturation’ values to tweak how much of the Sun Color will tint your effect (often this will be 0).

## Testing

As always, please test in a level (like VFXOceanTest) that has proper Lighting and Light Matinee set up to make sure everything works properly under all day/night lighting conditions before checking in. You test the various times of day lighting by scrolling through the various times in the **Matinee\_MasterDayCycle** Matinee. You can also test this by playing the game and setting the cheat code like this to various times of day using Military Time (ex: 1:00- 24:00) : **cheat settimeofday 12:00**

## Useful VFX Material Functions and Graphs

### Material Functions

#### Custom

**Atlas-DistFieldClip** - this Function is used to make particles invisible inside a ship so that splashes, etc don’t show up under the decks of a ship as it sails through them. To use this function, just place it in your Material in the Material Editor and feed your Opacity or OpacityMask output into this function and then take that output out to the Material

Opacity or OpacityMask output. A Material Instance (MIC) for the Material should be created so the useful parameters are exposed for tweaking as needed.

**Atlas-ProjectOntoWater** - this Function is used to project the Vertices of a Particle Sprite or Particle Mesh onto the surface of a WaveWorks Ocean Surface. To use this properly, a Material Instance (MIC) for the Material should be created so the useful Projection parameters (like distance to offset the verts to avoid interpenetration, etc) are exposed for tweaking as needed.

**Atlas-NormalBiasUpward\_WS** - this Function is used to force the Normals of a particle sprite/etc to point upwards to get a better sense of stable overall scene lighting rather than having the normals point at the camera and get lighting from that direction (which can change a lot as camera rotates and sometimes have color tints that don't fit well with sky color, etc.). As always this function is best used when a MIC is created so the Bias Amount can be tuned.

## Waveworks

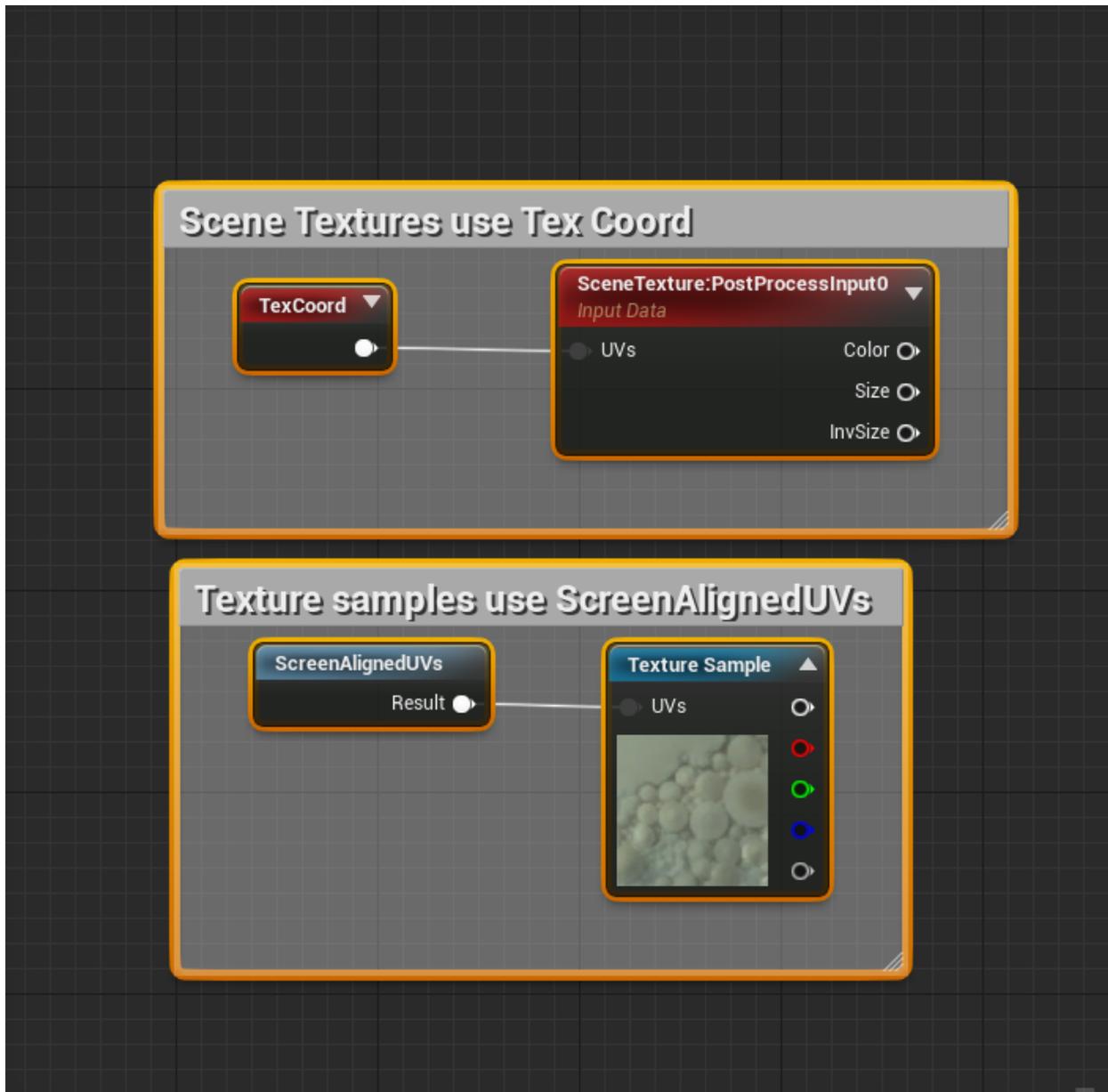
**WaveWorksHeight** - gives the value of the WaveWorks Ocean surface at the current position. See **ATLAS-ProjectOntoWater** function for example of how to use.

**WaveWorksDisplacement** - gives the value of the fully displaced WaveWorks Ocean surface including the horizontal motions. See **ATLAS-ProjectOntoWater** function for example of how to use.

## Material Graphs

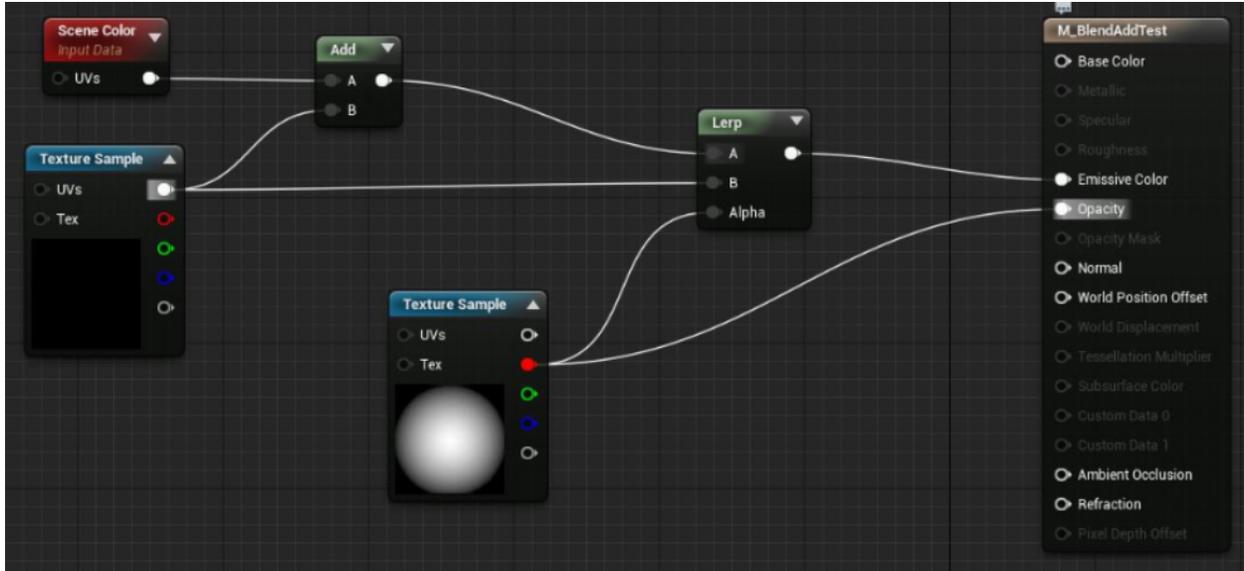
### PostProcess material good practice

SceneTexture needs textcoord node Texture sample needs screenAlignedUVs node



## AlphaComposite

AlphaComposite is a blend state like 'Translucent' or 'Additive' in the latest Unreal Engine, but it is currently not in ours. It allows for much better results for blending bright colored VFX that you want to seem 'Additive' over a bright background- where Additive VFX tend to blow out to white or disappear. Here is an example graph:

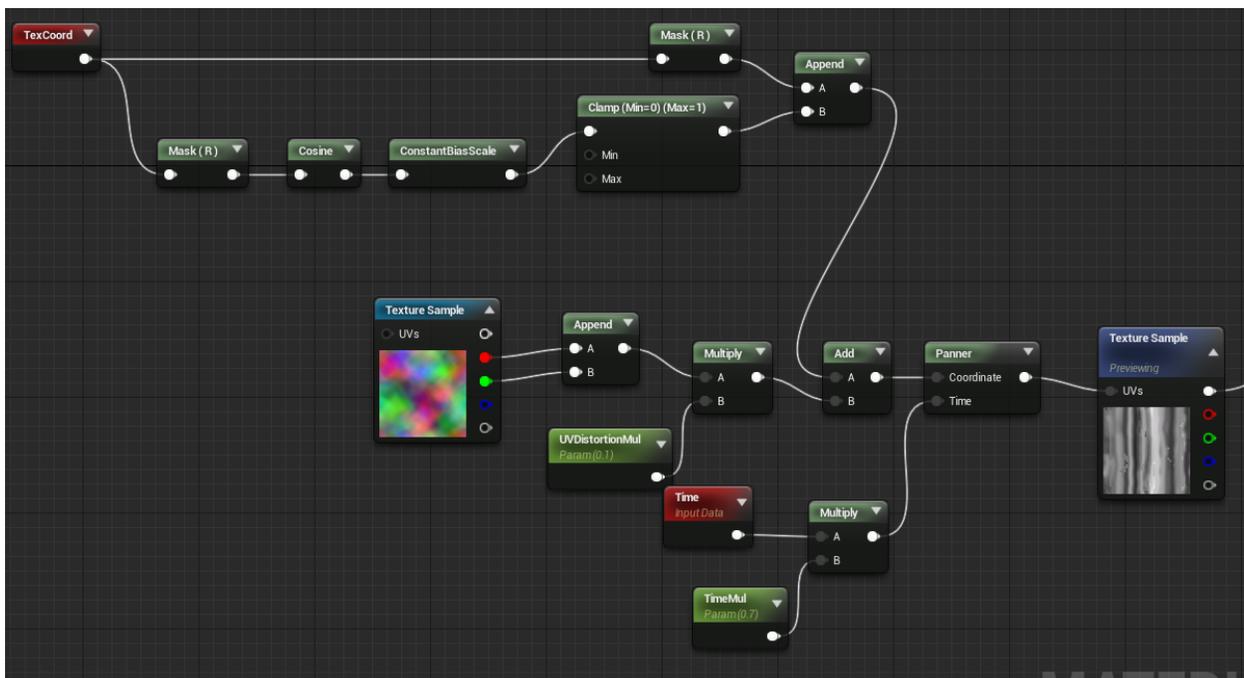


Note that this is a simple example with just two textures defining the color and opacity, but basically the color and opacity outputs of your shader math should be used how these are.

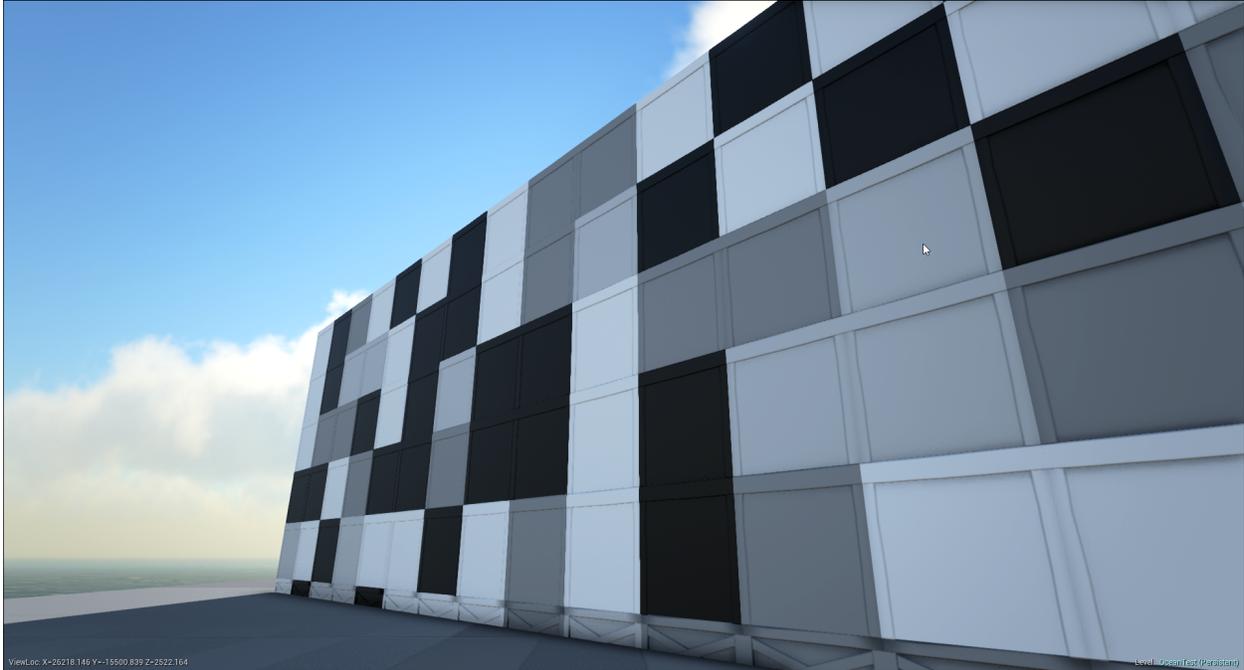
Here is some more info on it: <https://forums.unrealengine.com/development-discussion/rendering/74467-alphacomposite-finally?102548-AlphaComposite-finally!>

**Note:** There is now an 'AlphaComposite' Blend Mode available (where you set Masked/Translucent/Additive/etc) that should have similar functionality as this Material Graph.

## Scroll texture from out to in



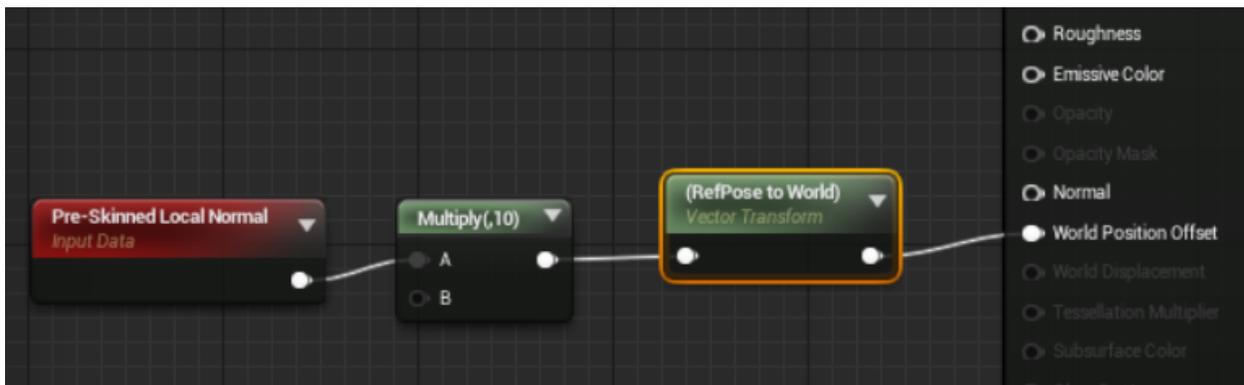




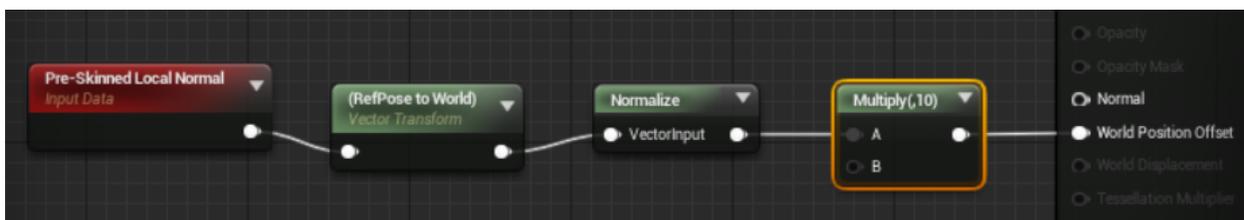
### RefPose to World Vector Transform

We created a TransformVector source space for “RefPose” which means you can transform from skeletal mesh bind-pose/refpose into world space. The transform would include the scaling, though, so you may want to transform an offset direction vector and then apply any additional world space scale after normalizing the output. Note that this node only works in the vertex shader, so if you want to use it in the pixel shader you must pass it via Custom UVs.

Basic use case for RefPose to World that includes scaling:

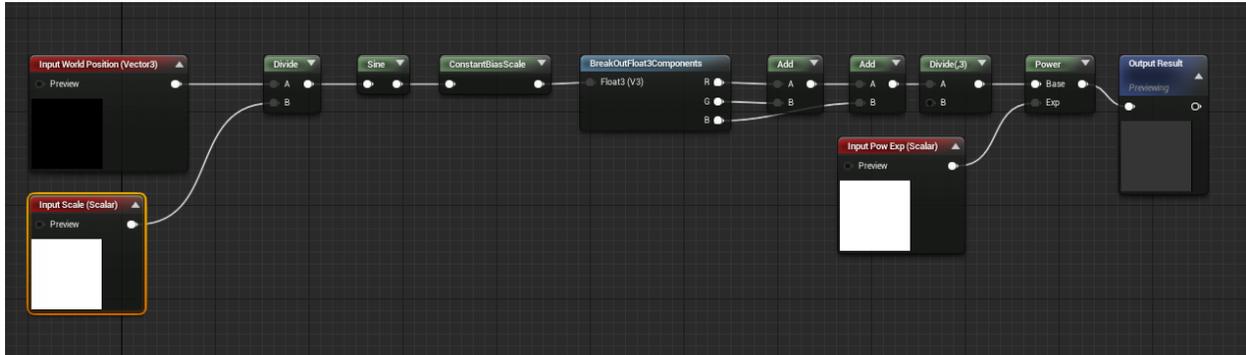


This version applies your world space scale after rotation and normalization:

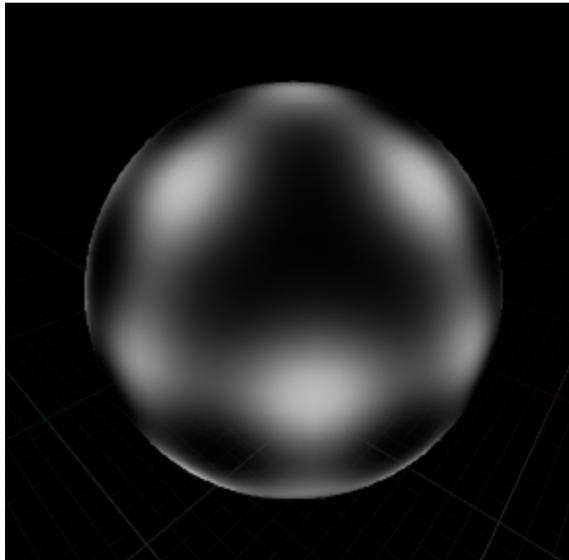


## 2.3.3 Functions

### World Sine



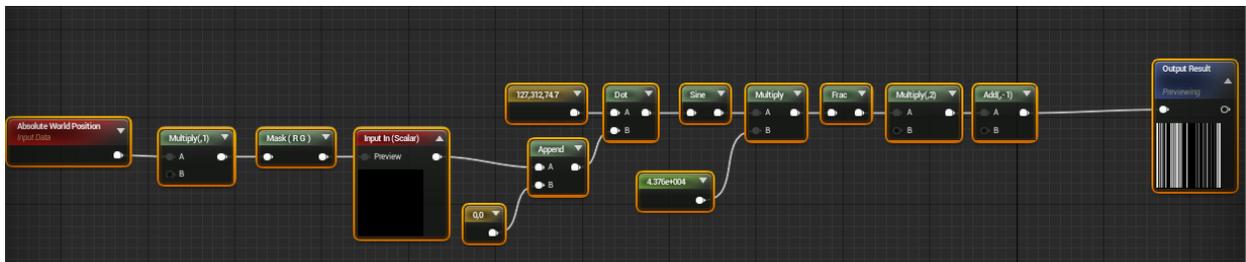
**Description:** Creates sines pattern in world space.



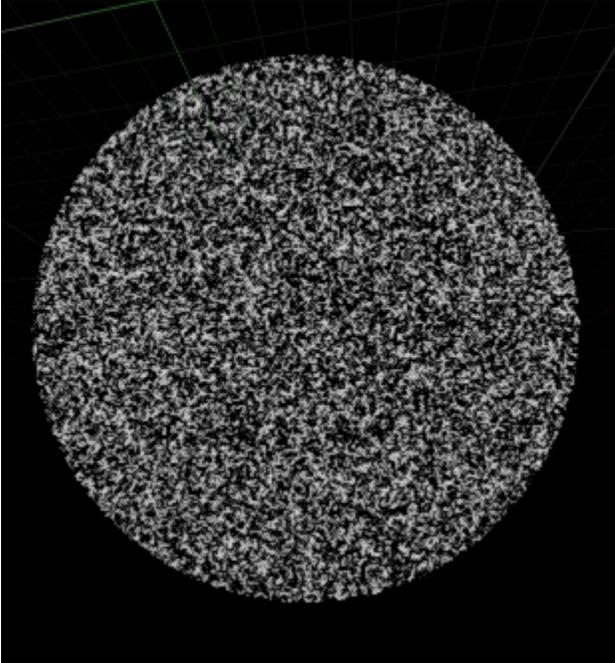
#### Use Cases:

- Creating a soft pattern for fake lights (Ex: Window Light)
- WPO for character wavy bits (Ex: Desert Kaiju wings)

### Gradient Hash



**Description:** Creates a unique value for each point in world space.

**Use Cases:**

- Create random values for every point in the world
- Provide random values as a replacement for Object ID in instances where meshes are combined

## 2.4 Colorization

### 2.4.1 Overview

**What is Colorization?**

Colorization defines up to 6 dye-able zones on assets that players can dye in game. For example, why the seat of a saddle can be dyed one color, while the base can be set to a different one. This is done through a texture mask that defines up to 7 zones (6 dye-able and 1 non-dyed). These mask zones are defined by 6 separate colors (RGB and CYM) plus black for anything that should always stay the original color. The strength of the colorization (for example, wanting the wear on the edge of a saddle to retain a similar color no matter how the saddle is dyed) can be modified through the intensity of the masking color on those zones.

This is ARK's dye system in game:



One color in the colorization map defines each of those listed color regions. In this case, region 5 is the area defining the front straps, and when the dye is applied to that zone it is colored in game.

### What do we colorize?

The general rule is that weapons, structures, clothing, and dinosaurs can be colorized. Not every asset within these categories is colorized however. For example, stackable items, special structures that players use (like workbenches), and weapon attachments are all not colorized. Icons of colorized objects are also colorized, with the exception of structures and dinos. For a more in-depth explanation of what the sub-categories are, see the general reference section on this page.

## 2.4.2 Colorization Texture Breakdown

Here is an example of the textures for colorizing the Megalania saddle.

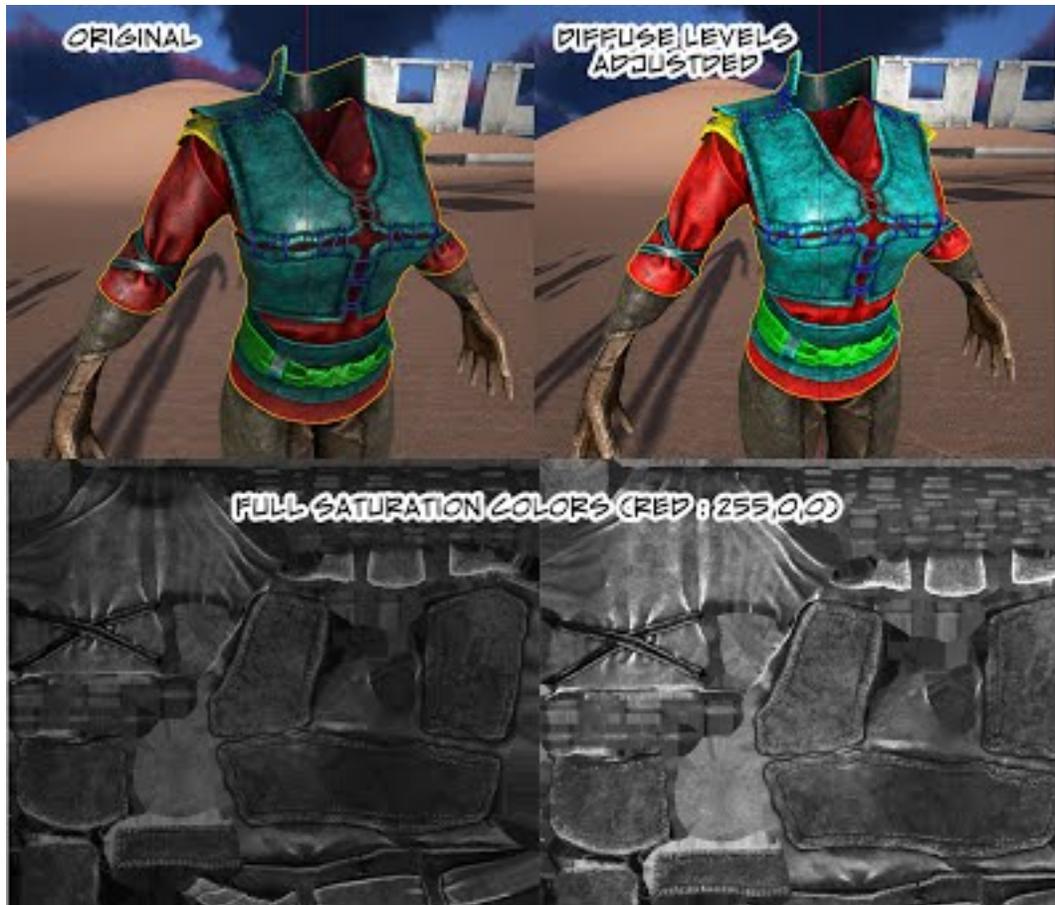
Colorization Mask	Colorized Diffuse	Original Diffuse	In-editor colorization regions

This mask utilizes all 6 color zones, and everything on the saddle is possible to dye (hence no black in the texture). The colorized diffuse retains some of the color in the rubbed leather edges, and is darkened in the same zones on the mask so that the dye effects those areas less intensely. The greyscale value of the diffuse has also been lightened, as the dyed version can become too dark when using the original value levels.

**Something to note about color placement choices in this example:** Many of these zones are defined by separate mesh shells in which case there are no restrictions on bordering colors. If you look at the use of green (which in this case is defining the metal sections) it is within the same mesh shell as two other color zones. That is a case where

the color choices become important, because some colors that do not blend well cannot border each other on the same mesh shell without creating a noticeable white line on the final asset at the blend point. In this case, green can border on cyan and yellow, so the straps with metal brackets are dyed those colors to accommodate this need. For more information on what colors can border others, see the reference section.

**Why desaturate the diffuse?** Areas of the Albedo texture that will be colorized should be desaturated so that the colorization is not affected by the base Albedo (unless this is specifically intended). If an area is not going to be colorized it should not be desaturated in the base Albedo texture. It can be OK to add a small amount of base Albedo color variation to desaturated color regions to better match the spirit of the original asset (see saddle example above). However, one should be careful doing so as adding a color in the Albedo region that is colorized will make it harder to accurately color that area with certain colors, like white.



### 2.4.3 How to Colorize (Non-Creature Assets)

#### What you need for colorization

1. An Asset to colorize
  - Make sure the mesh is in it's final version, otherwise work may need to be repeated. Find the fbx file for the mesh.
  - Locate the diffuse texture file for the asset to be colorized
  - In both of these cases, it may be best to export from the in-game version to be sure of having the latest as source can sometimes be outdated.

### 2. Software

- Photoshop
- Maya/Max
- Optional: Substance Painter

### 3. Useful additions

- Get the photoshop action for setting up layers for each colorization color.

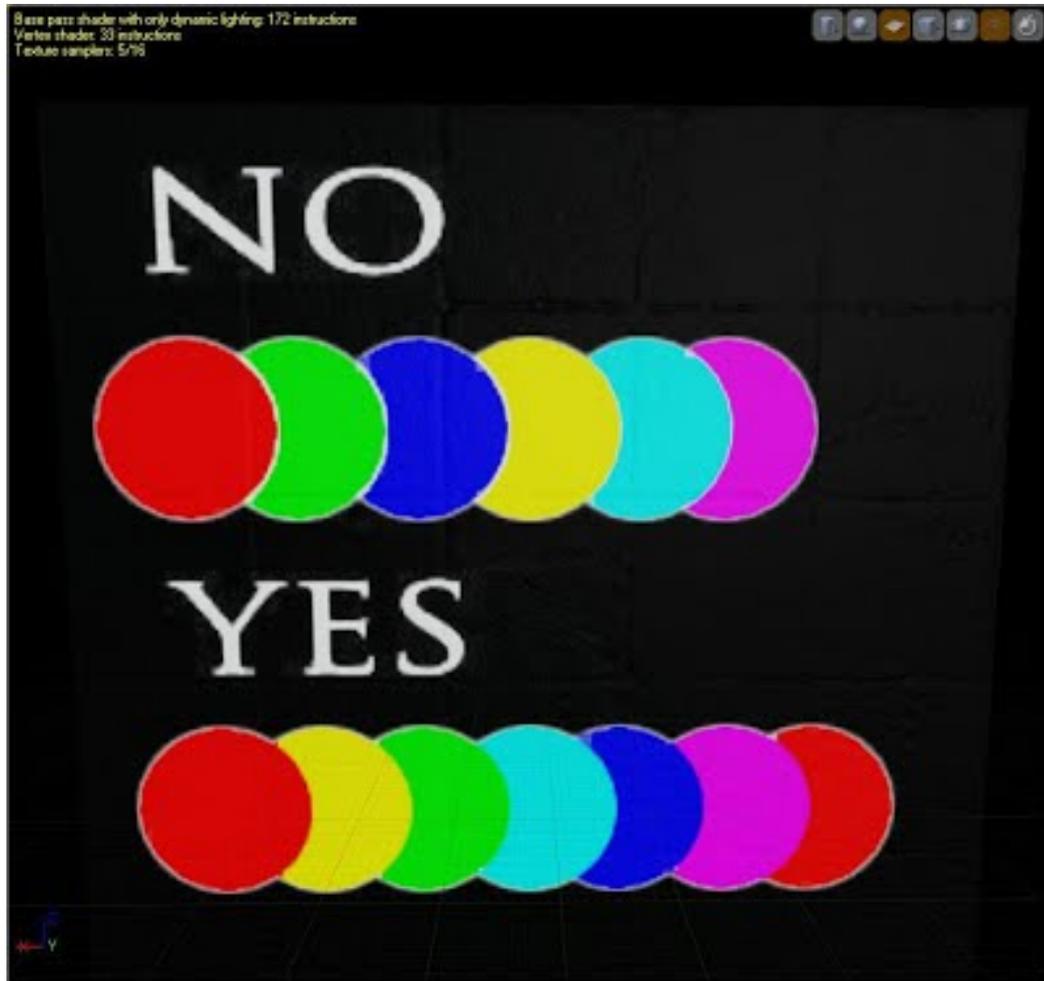
## How To Colorize Mesh Textures

### 1. Decide on color zones

- **Color bordering restrictions:** Choosing dyeable zones is one of the most important steps. Only certain colors can border each other on the same UV shell without causing a white line.

For example, red and blue should not border with each other because where they mix is magenta. If magenta is used elsewhere in the mask, the magenta colorization will be applied everywhere red and blue mix. *Even if magenta is not used elsewhere it will still create a white line at all blending points due to white being the default colorization value.*

Here is a chart showing what colors can touch (**Note:** The chart does not apply for separate mesh shells, where any colors can be on neighboring shells. Only colors present on the same shell need to follow this) You can see the white border that appears between the colors in the first example, and how it is not present in the second.

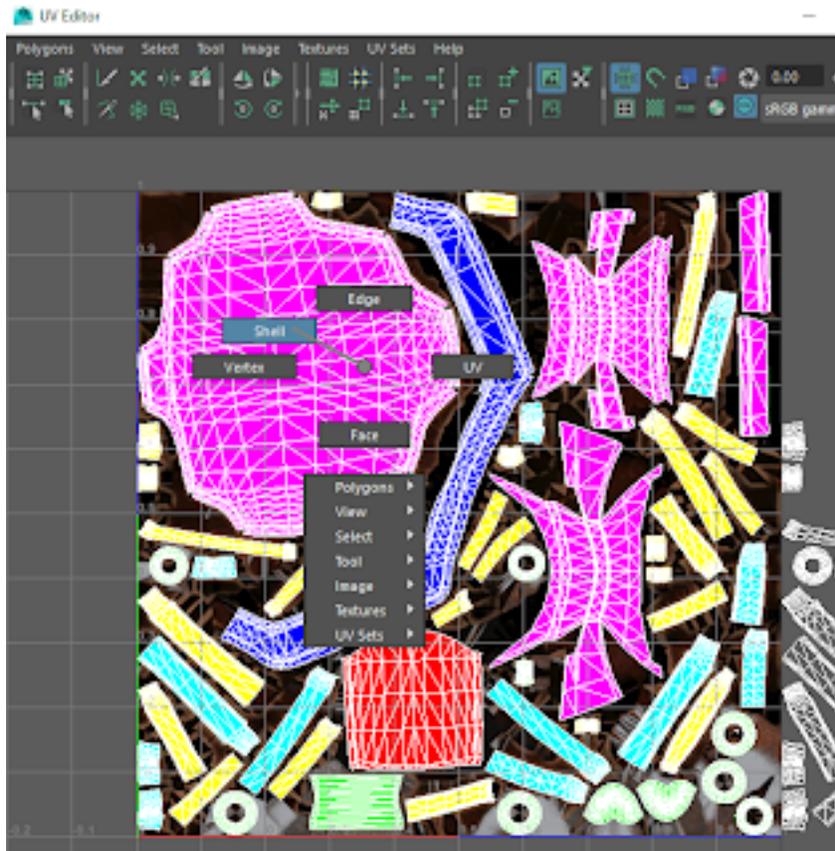


- **Choosing zone divisions:** Often zones themselves will be defined by the materials used. For example, all the metal parts on a saddle may be one color. On a weapon, maybe the wood handle is one section and the metal blade is another, while the leather wrap around the handle is a third. Everything in a zone will be colorized with the same color (like the brown on saddle leather), so it is important to only have similarly colored surfaces in a single defined zone. Metal and leather in the same zone would not work due to requiring two separate colors (grey and brown). Often the mesh will have natural divisions in the shells used that can be very useful to base colorization off of as well.
- **Number of zones:** The goal is to always have at least 3 color zones unless the mesh is extremely simple (look at gloves with only one material, for example). The ideal is to aim for 4-6, and use your judgement on when a smaller object may not need that many.

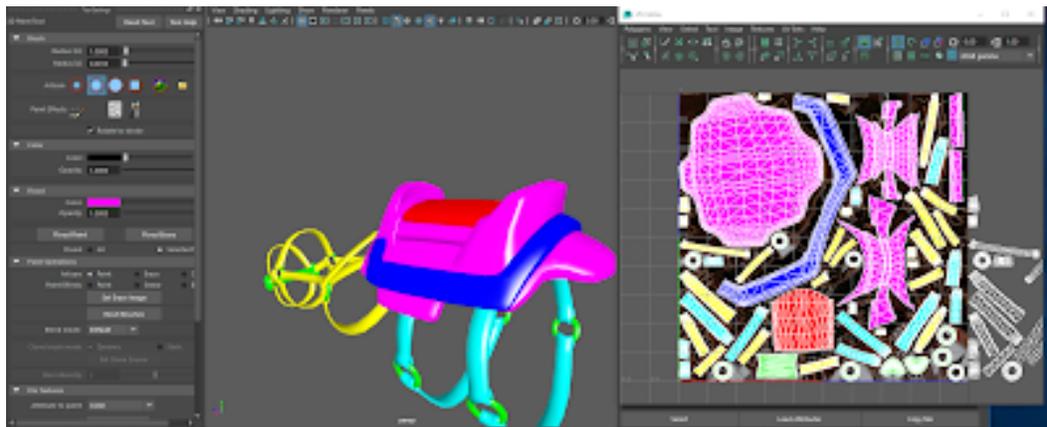
## 2. Create the colorization mask

- **Initial color masking**

- This can be done in several different ways. The most efficient is to first import the asset fbx in maya
- Apply the original diffuse texture to the mesh
- Open the UV Editor and choose the selection tool. Set this to select shells.

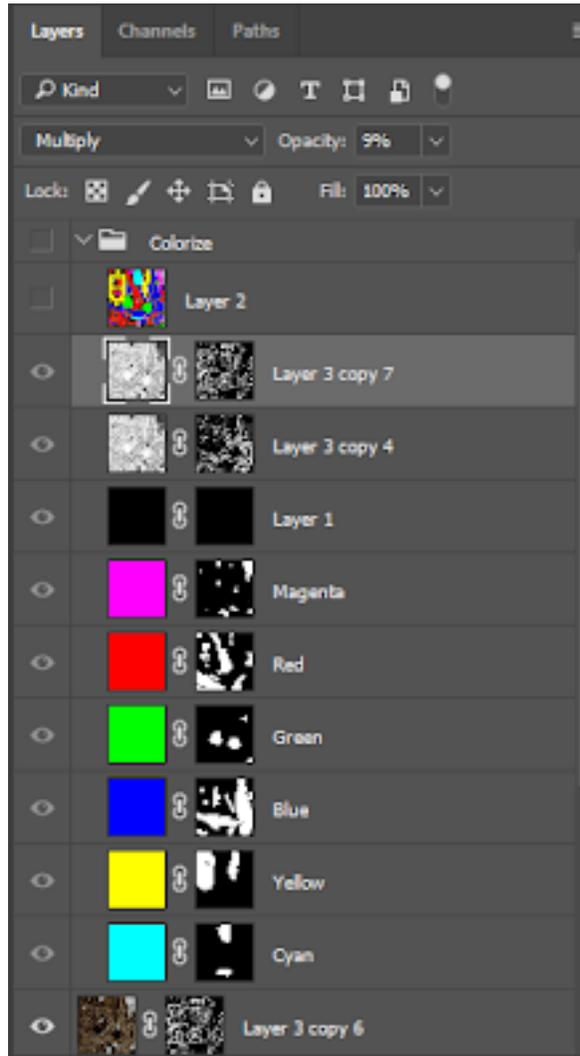


- Select the shells that will be in the same dyeable zone (for example, all the metal rings on a saddle)
- Using the 3D Paint Tool, flood paint the selected area one of the six colors. Remember to take into account what colors can border. If there are small metal rivets for example, then whatever color assigned to metal needs to be compatible with the color assigned to the shells that will have rivets on them. So in this example, if the metal is green, then the straps (which have metal rivets on the strap mesh shells) need to be cyan and yellow to border with green on the same shell.



- Once done (make sure all of the shells are colored, including backfaces) save the texture out.
- **Optional: Take into Substance Painter to fill in the gaps**

- Import the mesh and the created texture into substance painter
  - Create a new fill layer and make the mask texture the base color (UV scale should be set to 1)
  - Export the texture with dilation on
- **Build mask photoshop file** The colored mask portion of a finished file looks like this (the diffuse layers are below these):

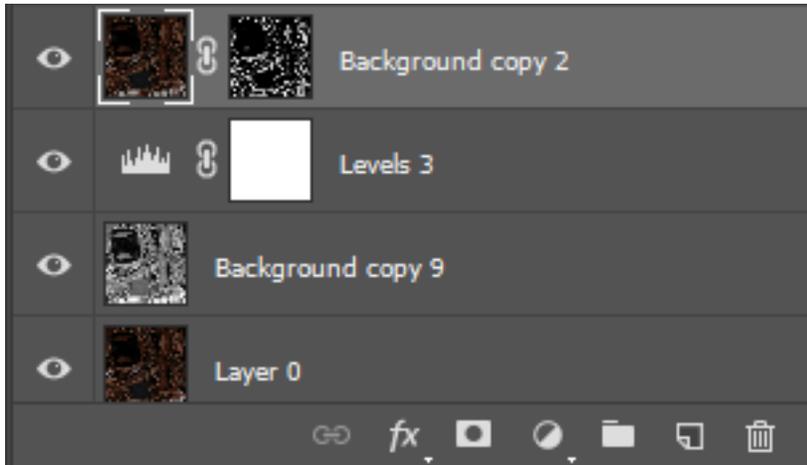


- Open the asset's original diffuse and run the photoshop colorize action on it, this will create the layers for each color channel.
- Open the mask file you created, and copy-paste it into the diffuse's file above the colorization layers. You can see this just below the Colorize folder. It is only there to copy from into the individual layers, and is hidden afterwards.
- Copy the colors on the mask into each matching channel layer. Anything that you want to have show on that layer (red, for example) needs to be masked with white on the red's mask. You can see this mask just to the right of each channel's color fill layer. Hide the texture imported from maya/substance when done (click the eye to the left to turn it off).
- Do any necessary hand-painting (for example: a shell that has metal buttons on it, or any mesh shells

with two dye-able zones) that was not included in the flood painted areas. To do this hand painting, use paintbrush set to white, and paint into the mask of the color you want to show (for example, if metal is green, paint the metal pins in the green channel above the yellow and cyan leather layers)

### 3. Create the diffuse texture

Example of a finished set of diffuse layers:



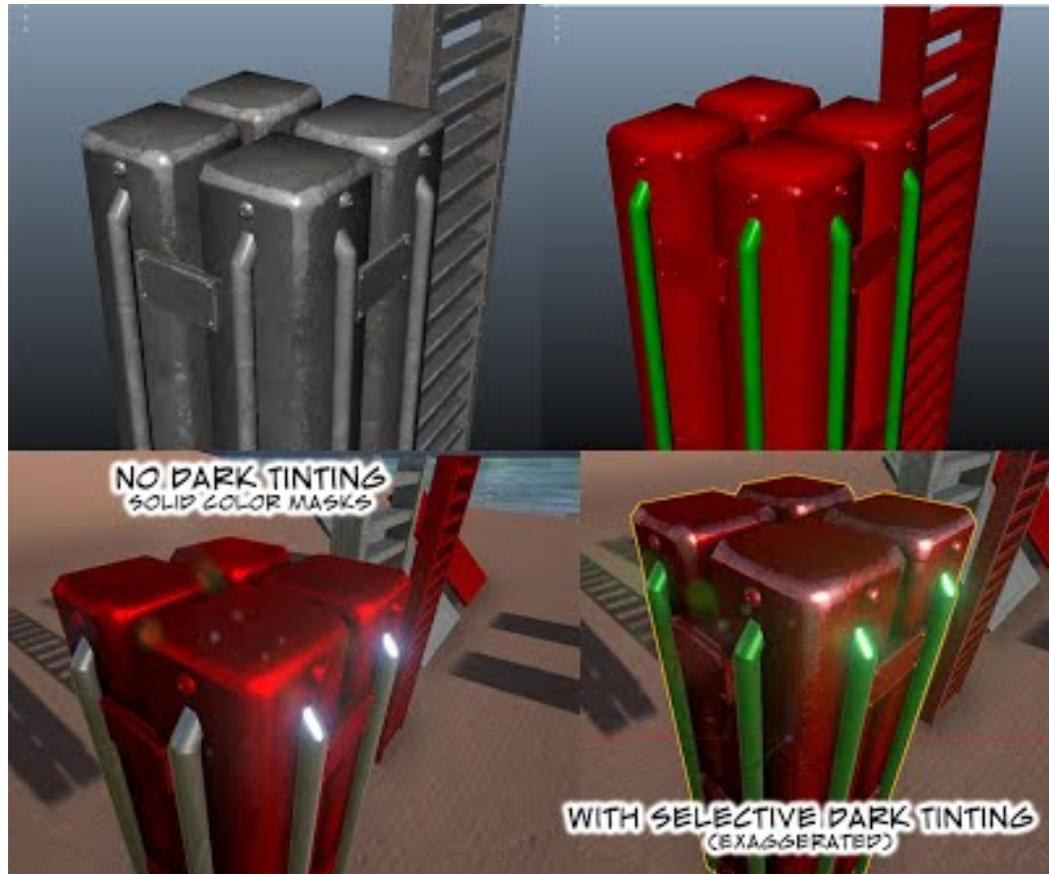
Not every diffuse layer set will need something like the “Background copy 2” layer, that is used when of the original color needs to be kept to show through the dye. Some diffuse layer setups will also get more complex, such as if there are multiple selections of original colors (say the edges on a saddle, plus the wear on metal), or if adjustments need to be made to a diffuse masked layer (Background copy 2) for the original color to interact correctly with the dye. Sometimes this means bringing the saturation up on that section, or decreasing it.

Creating the diffuse:

- See if there are any special circumstances that apply (wood, stone, etc) and if so, follow the instructions for those, found further down this page.
- Duplicate the diffuse layer and desaturate it.
- **Apply a levels adjustment layer. The result should be no extremely dark spots in dye-able areas, the goal is a light to mid g**
  - Try not to lose too much of the value variation, flattening the range will drop some of the visual interest in the texture.

### 4. Additional adjustments

- **Decide what fine tuning is necessary**
  - Some things will match the original asset better with less intense dye (for example, worn edges, patterns with distinct color differences, color variations within a surface). Identify these. This image shows an example of the difference between worn metal edges when this has not been done (lower left) and when the dye impact on the edges has been reduced (lower right)



- Some of these areas will also need the original diffuse color to come through (like worn leather showing through). This is the difference between no darkening on the mask (left) and allowing some of the base worn color to show through (right).



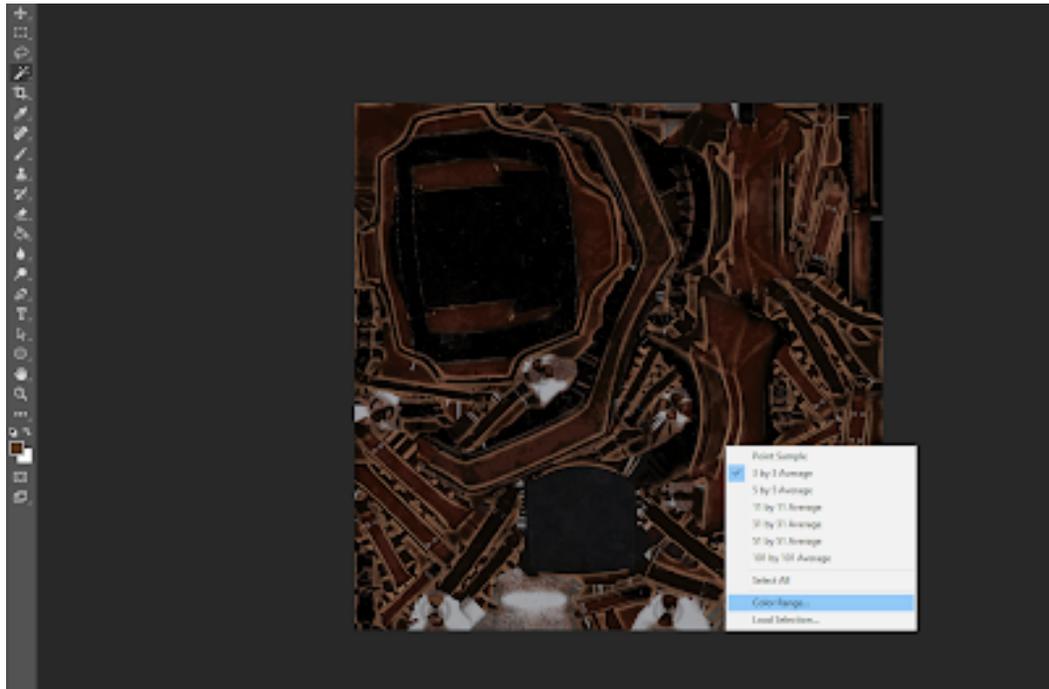
You can see that on the left, the dye does lighten at the edges because the diffuse value is lighter, but to get the original color to show through, the amount of dyeing in that area needs to be reduced and the original color need to be kept in that part of the diffuse texture.

- If the color just needs to be dyed less then just part b is necessary. To retain original color,

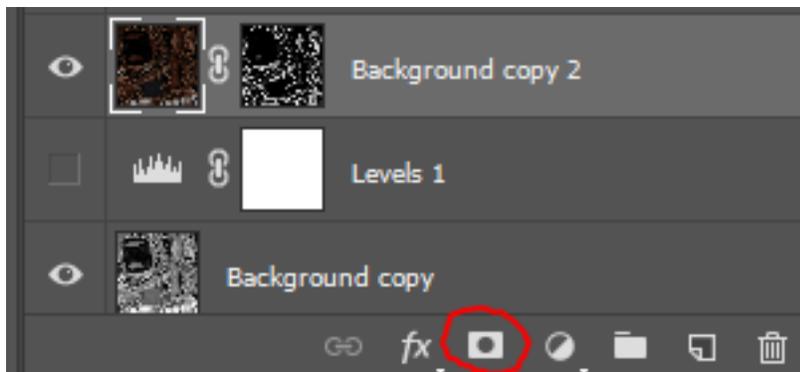
also follow part c.

- **Reducing the impact of dye on specific areas**

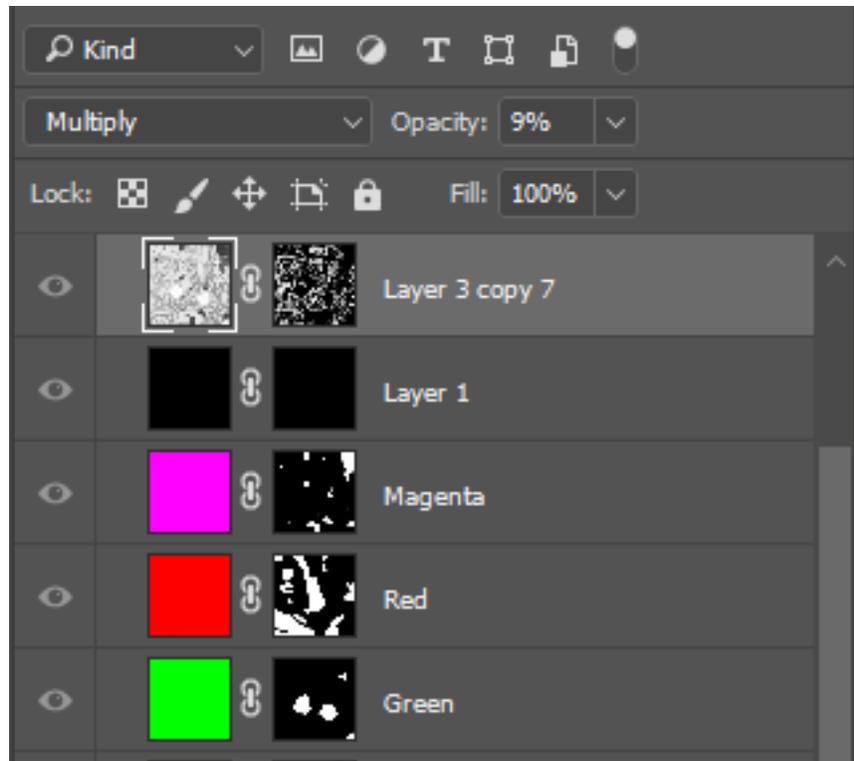
- For each part to be adjusted, select that on the diffuse (selecting by color works well here). For this example, assume the worn edges of the saddle are selected. Having a different layer for each thing adjusted (saddle edge vs worn metal, for example) can help with later adjustments. To select by color, use the magic wand, right click, and select Color Range, as shown below.



- Create a mask of this selection. To do this, hit the button circled in the image below and it will mask the current selection on the current layer.



- **Desaturate this layer. This layer should be moved above the colorization mask layers and multiplied.** The opacity on this layer should be set to around 10% or less. On a very rare occasion higher might be needed, but 10 is usually enough.



- **Retaining some of the diffuse color**
  - To retain the original color for an element, select the masked area for that. Right click on the layer mask and select add to selection.
  - Duplicate the layer with the original diffuse (not desaturated) and apply the mask to that layer. Move this above the desaturated diffuse layer.
  - This will likely need some tweaking once the colorization is in engine and you can determine whether this color is too bright, dark, etc.

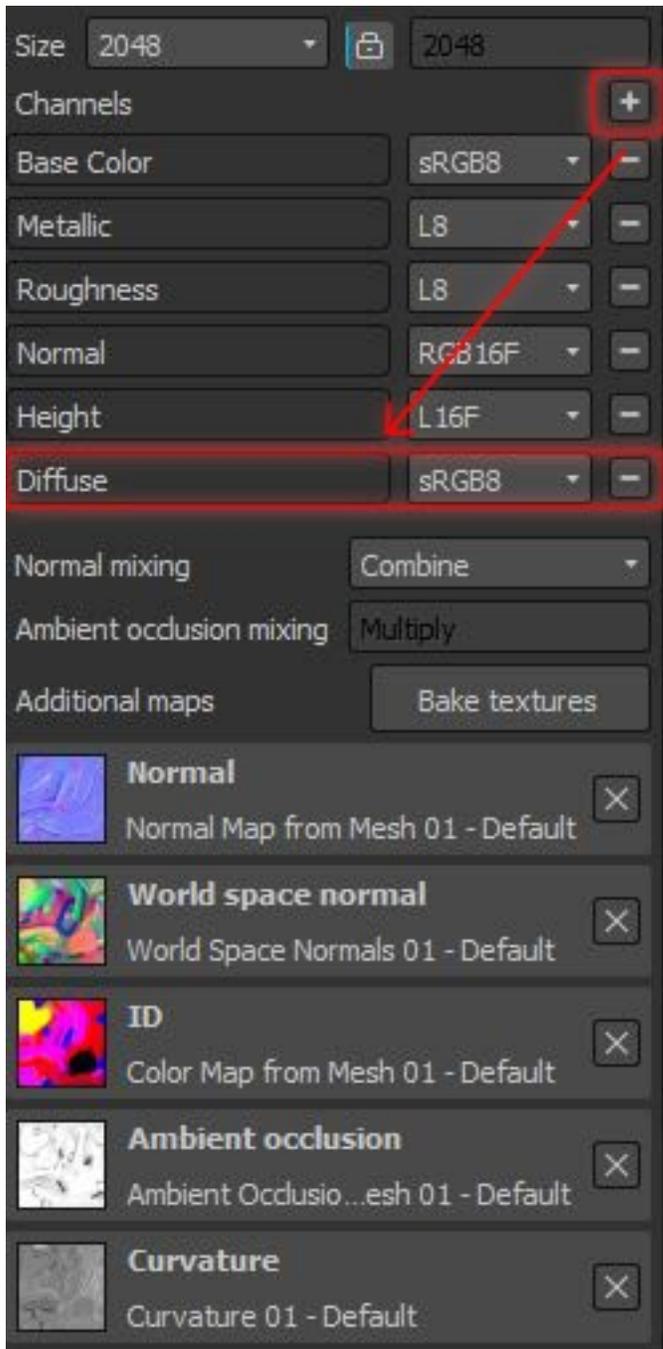
#### 4. Save files

- **Mask settings**
  - Naming: suffix: “\_Colorize\_m.tga”
  - After hitting save, choose 24 bit
- **Diffuse settings**
  - Naming suffix: “\_Colorize\_d.tga”
  - After hitting save, choose 32 bit

### Using Substance Painter for colorization

- **Add vertex color to the high poly**
  - Make sure when adding the vertex colors that you keep in mind the Color Zones mentioned above.
- **Create a substance file and then duplicate**

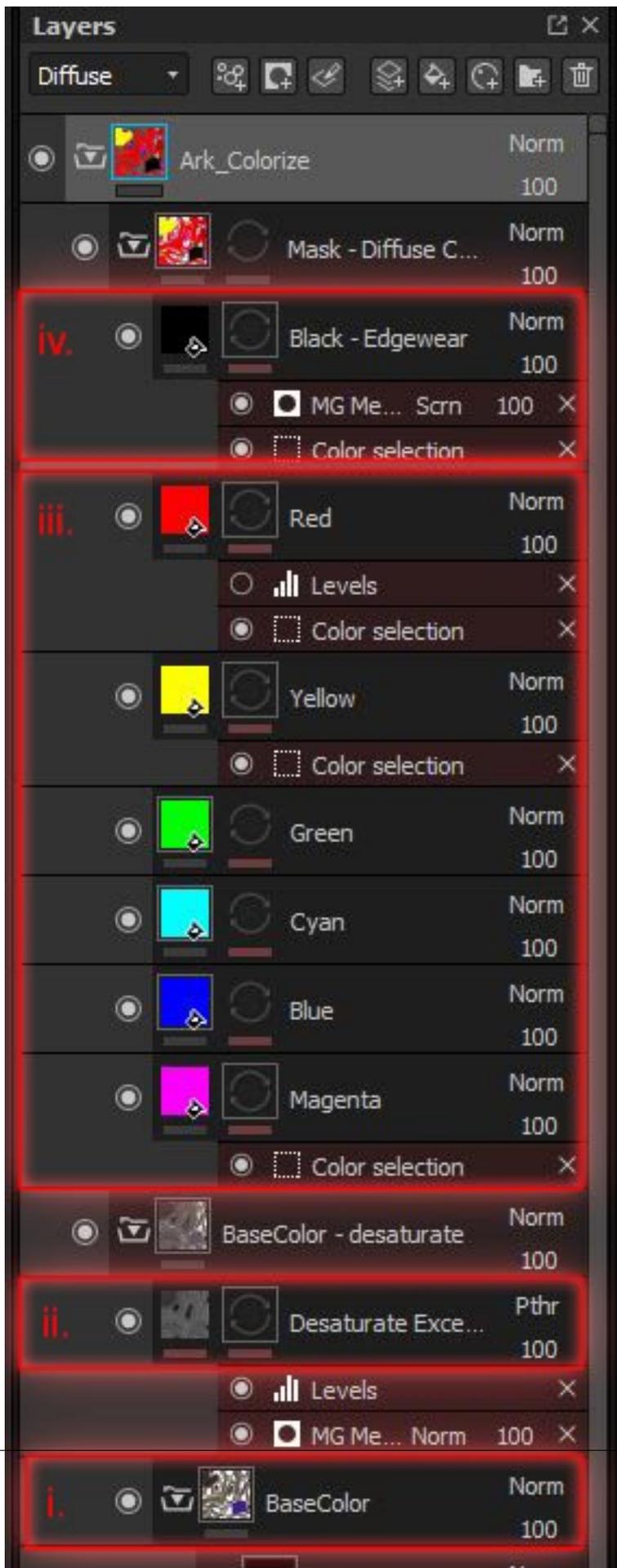
- Create a substance file to generate the regular textures (Basecolor, Normal, Layered, etc.)
- Once finished with the first substance file, save a copy of the file and add \_Colorize as a suffix to the name.
- Important: Add the 'Diffuse' channel to this new colorize file.



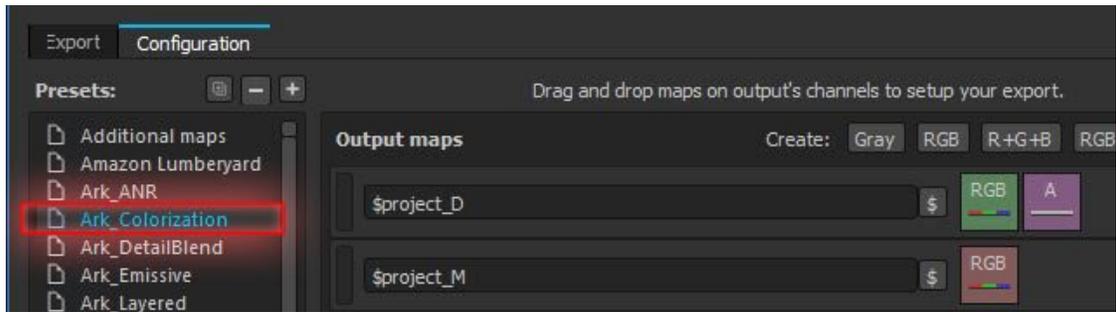
- **Add the Ark\_Colorize smart material**

- Add any layers used in the generation of the original basecolor, to the Folder named Basecolor inside the Ark\_Colorize smart material, just below the layer named Desaturate

- Adjust/change the masking on that Desaturate layer to desaturate the basecolor, except for the edgewear and/or any other areas that shouldn't be affected by the colorize process.
- Within the Mask folder there are several fill layers that only have a diffuse color and a mask. On each of these masks, Color select from the generated ID mask for the appropriate material. If the color zone selection has already been properly decided on when applying the vertex color, this should all be relatively simple.
- The black layer on the top of the stack should be used to partially mask out any edge wear (aligning with the edge wear from the 'Desaturate' layer and/or other areas that should not be colorized.)

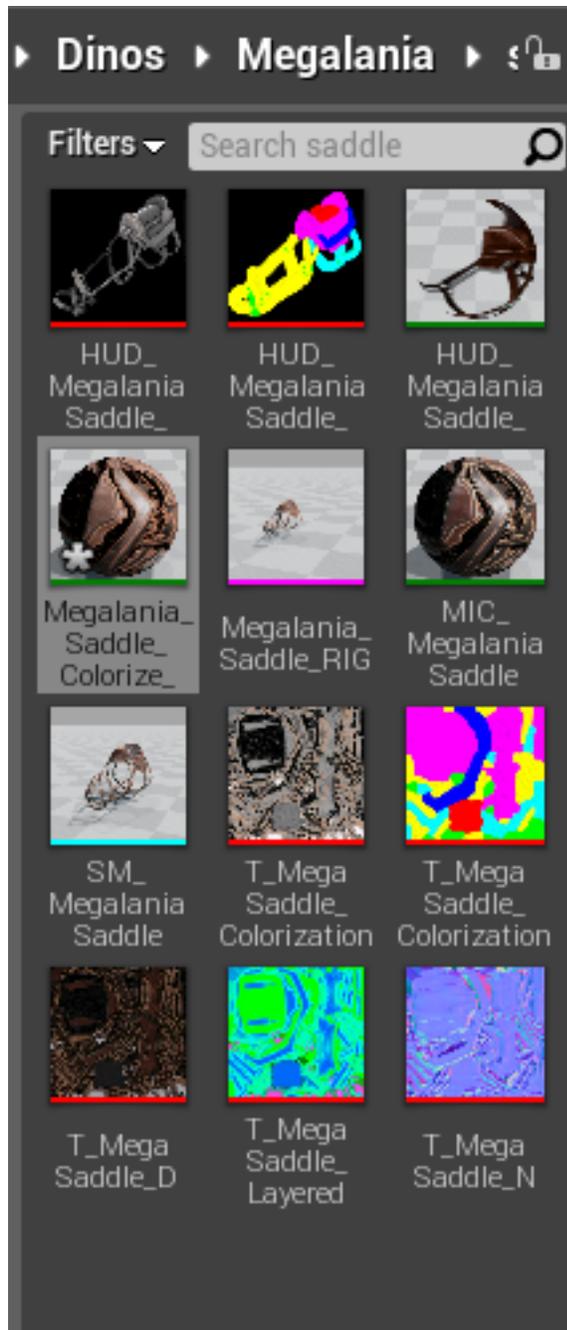


- **Export textures using Ark\_Colorization preset**
  - Export out the mask (T\_\*Filename\*\_colorize\_m) and desaturated basecolor (T\_\*Filename\*\_colorize\_d) textures to the appropriate folder.

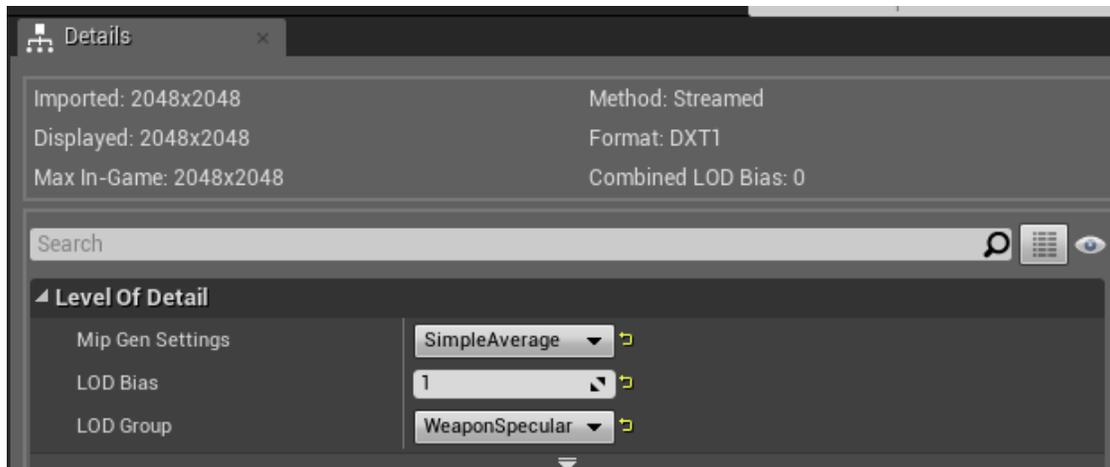


### How to set up asset colorization in Unreal

- Import the mask and diffuse into Unreal, in the same folder as the asset (or in a texture folder if it has one). Here is what it may look like once everything is in (colorization textures, duplicated material, icon textures and material)



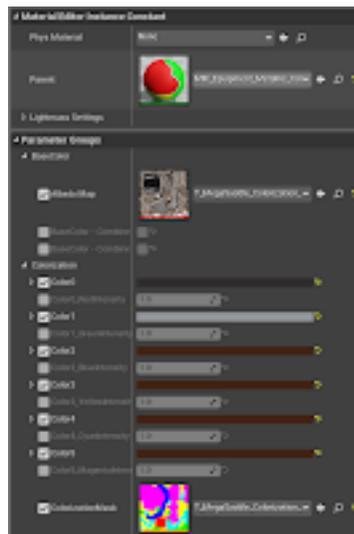
- Set the LOD settings for the mask. They should be Simple Average in the Mip Gen Settings. The LOD Group should be set to a specular group, depending on the type of asset. A weapon would be weaponSpecular, for example.



- **Create colorization material(s)**

- Make a duplicate of every material applied to the mesh that will need to be colored. Name these duplicates with a “\_Colorize” suffix
- **Open the material and find the parent. There are colored versions of most base materials, select the m**

\* For example: MM\_Equipment becomes MM\_Equipment\_Metallic\_Colorized\_MIC for an asset with metal parts.

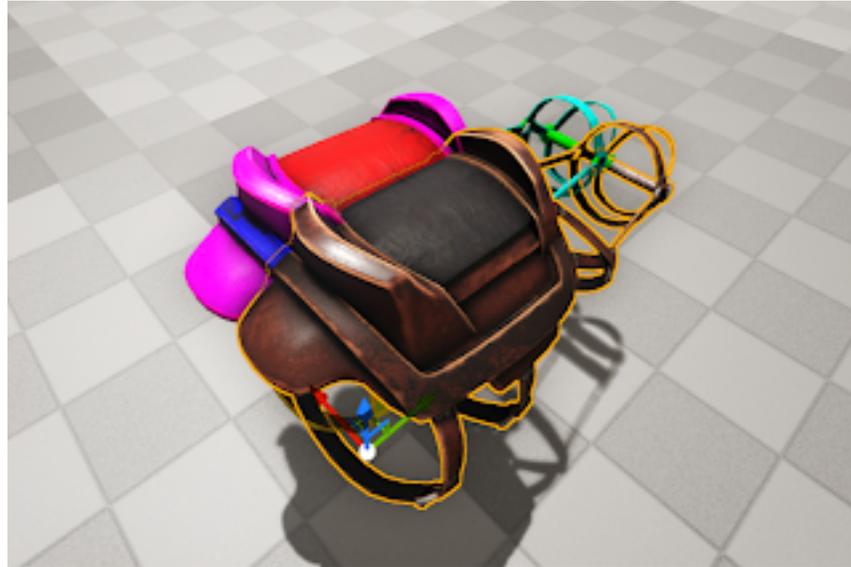


- Under BaseColor and Colorization sections, plug in the diffuse (albedo) and colorization masks.
- **The Colorization section contains all of the color channels and is where the settings for them will be chan**

\* **Choose what channels you are using and turn those on so that you can see them. Be aware that**

- If the colors are not showing, you sometimes need to refresh it to colorize by scrolling down to the static switch parameters and checking that on, then back off.

- Note: If there are any special case materials here (wood, stone, etc) look at the special cases section for how to handle their setup.
- \* Place the asset in a level with game lighting and create a duplicate. Assign one the colored material. This gives you a reference for the color you are trying to match and a way to make sure it looks as close as possible to the original by comparing side-by-side. For example:



- \* **Start adjusting the color of each channel until the mesh surface color matches the appearance**

- The base color buffer can be a great way to see the exact color without being effected by light. This is where to find that setting:



- \* If a surface does not match well using the color wheel options (the surface can't get light enough, the worn edges are the wrong color, etc) going back into the textures for further adjustment will be necessary. Repeat that back and forth process until happy with the match between original and colored.
- Once happy with the appearance, open the mesh file in Unreal and replace the non colorization material with its colored version.

- **Set up the blueprint**

- If the asset has a blueprint, open that up.
- Search for “color”. You will need to turn on any color channels (set them to 1) that the

asset is using before it will be possible to dye in game. Any set to 0 will not be used.



## 2.4.4 Testing

- **Appearance**
  - Make sure the colors appear the same in game lighting at around 12 pm game time
- **Play the game in editor (this needs to be done whenever the blueprint has been set up)**
  - If equipable, drag and drop one of the dyes on it to open up the dye menu. The dye can be found here, drag this in your inventory to drop onto the item to be colorized.



- \* Check for only the used zones being dyeable. Any that are not dye-able will not be selectable in this menu. If you can dye them (the dye color fills the upper right box on the selected region) but that channel was not used then it needs to be turned off in the blueprint.



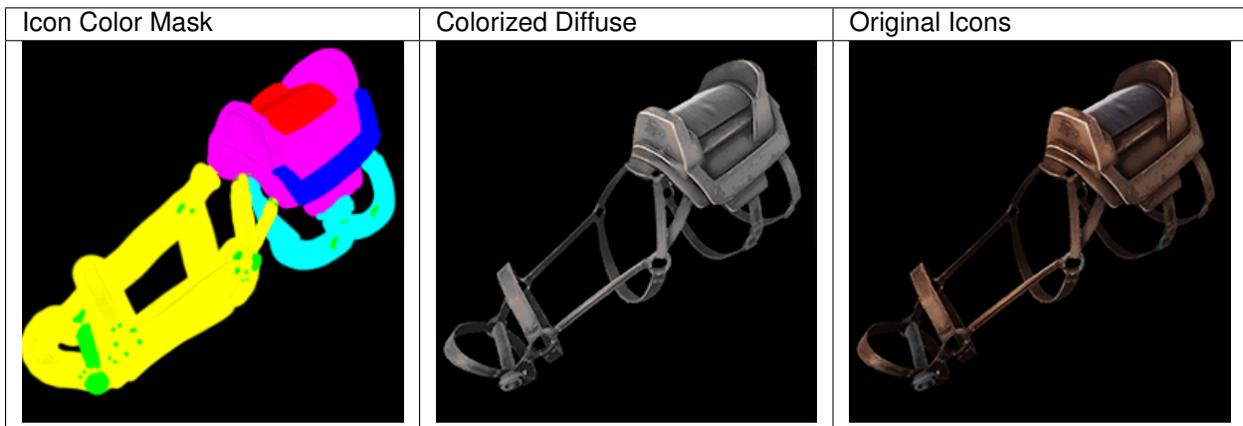
- \* Make sure all of those zones can be dyed and that the color shows up well. The above image is an example of what a dyed zone should look like.
- \* If the icon is done at this point, make sure that the icon is also showing up and dyeing correctly. Check this by applying the dye, leaving that menu, and looking at the icon in your inventory.

**– If a structure, place the structure in game**

- \* Drag the dye onto the paintbrush in the inventory and equip the paintbrush. Run up to the structure and hit it with the paintbrush to dye.
- \* Check for the same issues as in part a.

### 2.4.5 How To Colorize Icons

Example of a colorized icon:



**ATLAS DOES NOT REQUIRE COLORIZED ICONS. (IT USES DYNAMIC SYSTEM)**

**What requires a colorized icon?** Any colorized asset with an inventory icon (so weapons, clothing, some items). Structures never have colorized icons.

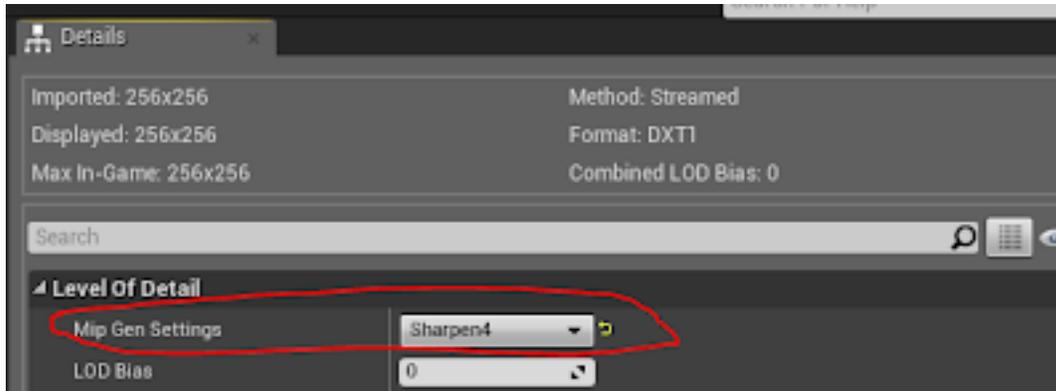
**Colorizing an icon**

- Open the icon in photoshop and do the same initial setup as the diffuse (run the photoshop action to create the channels)

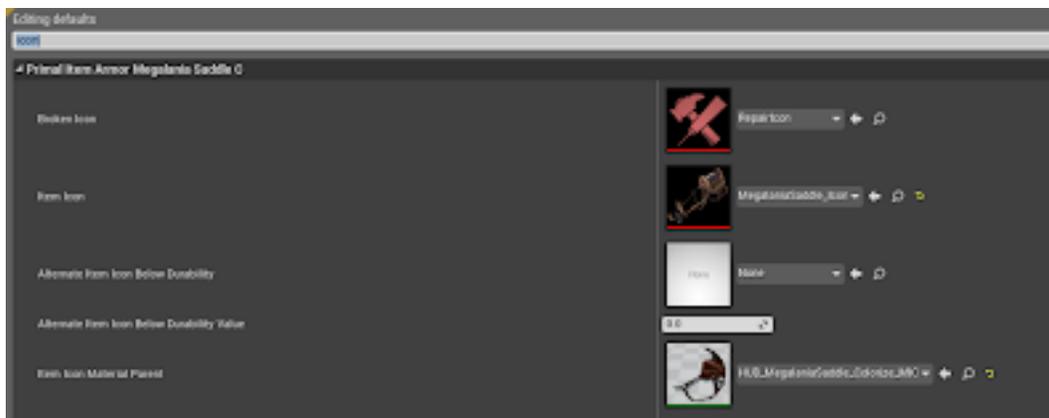
- Paint the sections with the colors they have in the colorization mask. This is done the same way as the asset’s colorized mask. Colors that shouldn’t border on a mesh shell can border here, it will have a minor white edge along the border of the colors but due to the size of the icons in game this is negligible.
- Create a desaturated diffuse in the same way as the asset texture.
- When saving out, name the textures with the prefix “**HUD\_**” and suffix “\_Colorize\_m” or “\_Colorize\_d” depending on being the mask or the diffuse.

### Setup in Unreal

- Import mask and diffuse into Unreal. Set the LOD Mip Gen Settings to Sharpen4 on both.



- Find a HUD material for a similar asset and copy that over to the asset folder. Rename for current asset.
- Copy paste the colors from the base asset colorization material so that the icon colorization matches. It is possible to right click on the color bar and choose copy to copy the color, then the same again (selecting paste) to paste the color into the color channel in the colorization section of the icon material.
- In the blueprint for the asset, search for “icon” and replace the icon material with the colorized version.



### Testing

- Play game in editor and give yourself one of the items. Make sure the icon shows up in the inventory.
- Dye the item, checking that each dyed zone also correctly changes the icon appearance.

## 2.4.6 Specific Cases

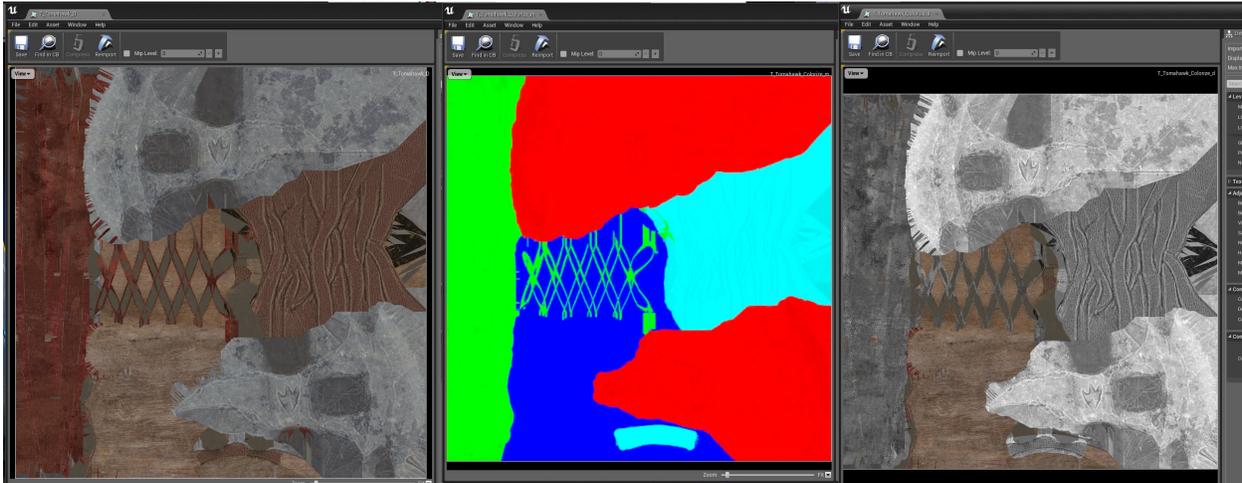
### Wood and Stone

Wood and Stone have a slightly different method to colorization. In general, wood and stone don't look as good with the usual approach of coloring over a b/w diffuse. Wood and Stone uses paint effects.

Any region of an asset that has wood is treated specially. It should be masked in full color but the underlying Albedo texture should not be desaturated. The color region simply tells the material which area will have a special paint effect colorization applied and the paint effects will also show the original base color.

#### Example asset with both normal and wood colorization

Note that everything is desaturated except for the wood part.



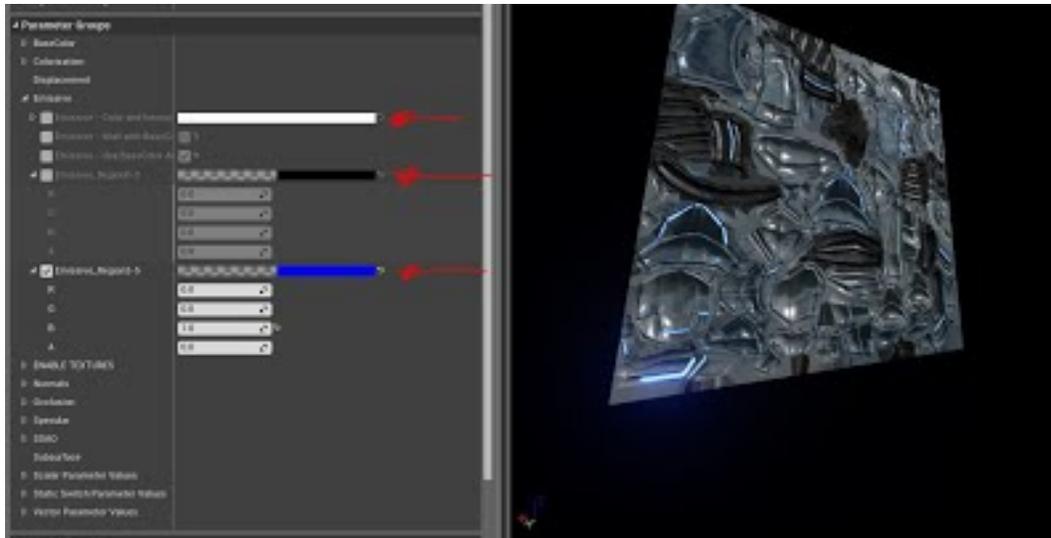
Settings to note for Wood and Stone colorization.

- Paint Effect is used to tell the material which regions are going to be colored as wood/stone (First red arrow)
- Paint Effect must be enabled (second arrow)
- UsingColor# turns on and off the paint effects per region. Should be set to 0 for any region that uses paint effects (third arrow)
- Edge effect should just be set to 0, the effect it leaves generally doesn't look too great
- Mask Scale - allows you to adjust the Stroke Mask tiling



## 2.4.7 Emissive Colorization (Such as TEK)

- Use a base material that supports emissive colorization
- Keep the Emissive Color to white use intensity for intensity brightness.
- Set the emissive\_region that should be emissive to 1 and allow the color set in the colorization channel to determine the color of the emissive.



## Glass

Glass is not currently colorized

## Clothing

Clothing is colorized the same way as a normal asset, but should also be checked for the presence of a skin mask texture for each piece of clothing. If a skin mask is missing, this should be created once the colorization is done. Instructions for this can be found here: [Creatures and Humans](#)

## Bosses

Bosses often have three different difficulty level that need colorization for each one.

## 2.4.8 General Reference

A collection of general information about colorizing. Ranging from what assets we colorize, to what the naming conventions are, etc.

### Assets Colorized

Colorized asset types:

- Weapons
- Clothing
- Structures
- Dinos
- Dino saddles and attachments
- Sometimes food is colorized for variations (like multiple colors of berries)

Exceptions not to colorize:

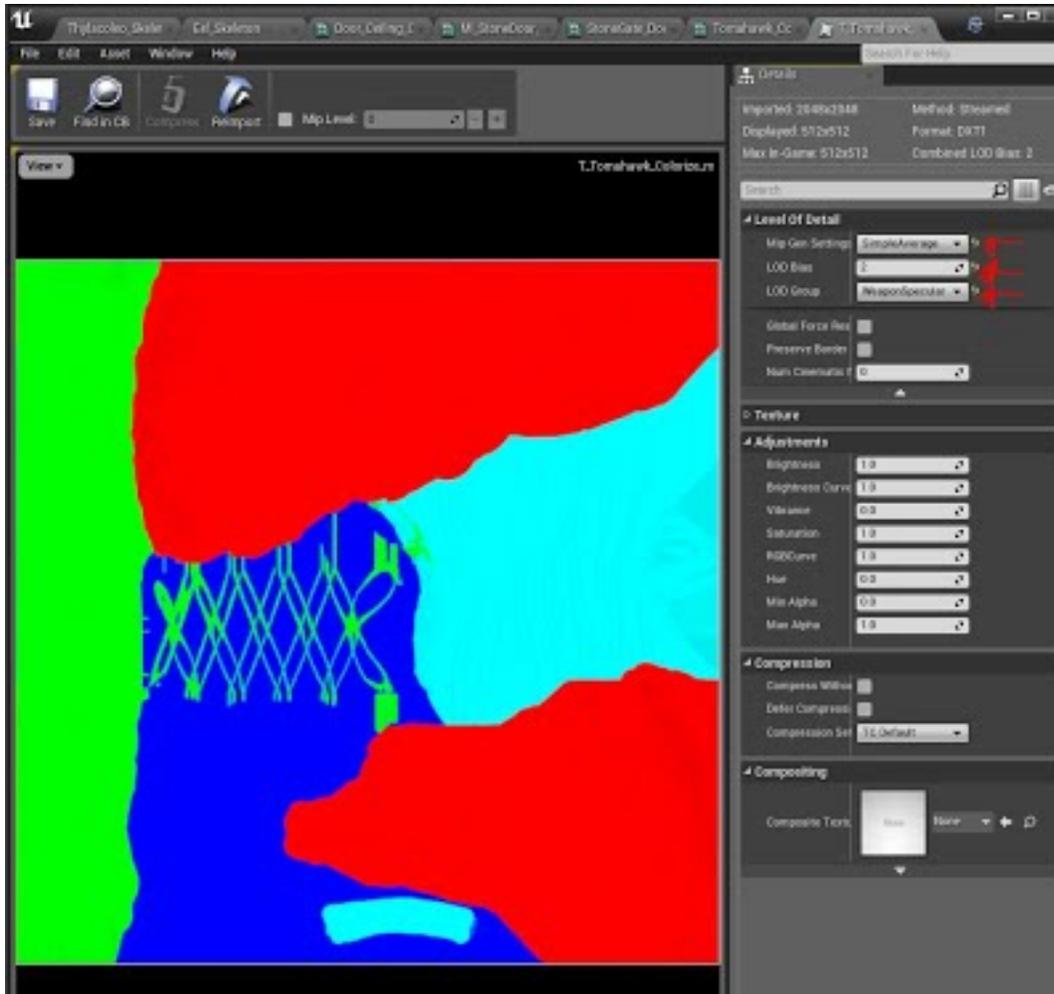
- Structure icons
- Dino icons
- Special structures. This is generally anything that is more function focused (like a workbench) while anything decorative (like a wall or chair) tends to get colorized
- Anything stackable
- Weapon attachments (unless it is part of the actual gun, ex Tek Rifle)
- Mirrors
- Glass

### Naming Conventions

- “\_Colorize\_d” , “Colorize\_m” suffix, depending on whether it is a mask or diffuse texture
- Materials also get the “\_Colorize” suffix
- Icons have a “**HUD\_**” prefix on textures and on the material

### Special Texture settings

- MIP Gen Settings - SimpleAverage
- LOD Bias - generally set enough to reduce the Displayed value to 1024 or 512 or even 256 if possible without it looking too bad
- Colorization mask textures belong in the Specular LOD Group for their respective asset. (ie tomahawk is a weapon -> WeaponSpecular)



## Notes about Colorization in Unreal

Cyan and Yellow switch: In the material in Unreal, the yellow and cyan color channels will switch regions, so what was colorized as cyan in the texture will actually show up as the yellow region in editor.

### 2.4.9 Notes on Dino Colorization

Dino colorization at its foundation is very similar to asset colorization, with a few differences to note:

- There is much more blending in dino colorization, where there can be layers of colorizing on the same zone. In assets, the zones are rarely overlapped, that can be done on dinosaurs.
- Teeth, nails, claws, eyes, and the inside of the mouth are never colorized.
- Dinos tend to have fewer color regions due to needing to be careful about blending on the same mesh zones. Bugs/Insects may have more, but many dinos use just 3 color channels.

#### 2.4.10 Common Issues and Edge cases

**Structures with multiple states** Anything with multiple states will need extra care when colorizing.

- Multiple states may include things like TEK that has an active and inactive state
- To colorize, two materials are needed (one for each state) with attributes that suit each state. For example, activated may have emissive, while deactivated does not.
- Emissive masks don't always update properly in-game, so if it doesn't seem to be working try turning the structure off and on again and it will often fix itself.

**Magenta artifacts on icon in Unreal editor** Sometimes magenta artifacts will show up in the editor window when looking at the HUD material. These do not show up in game and can be ignored. This is what that looks like:



## 2.5 Collision

### 2.5.1 Static Meshes

- Prioritize “generally correct” collision over absolute accuracy. Walking surfaces should be the most-precise, but remember that ATLAS is a large open world game and complex collision can cause performance problems for the server.
- Attempt to create collision for most assets with primitives (box, sphere, cylinders) if possible.
- If custom convex shapes (UCX\_) are required, limit them to a maximum of 32 triangles. Try to stay below 4 - 5 shapes with a maximum of 8 shapes.
- If auto-convex is necessary, try to use only 2 hulls, then increase only if 2 is not adequate.
- If more complex collision is required, you may use per-triangle collision with the last LOD, between 200 - 4,000 triangles depending on size of object. *More Info Here.*

## 2.5.2 Skeletal Meshes

- Use as few bodies as possible.
- Small creatures use about 10 bodies.
- Larger creatures can use up to 25 bodies.

## 2.6 Large Rocks

### 2.6.1 Overview

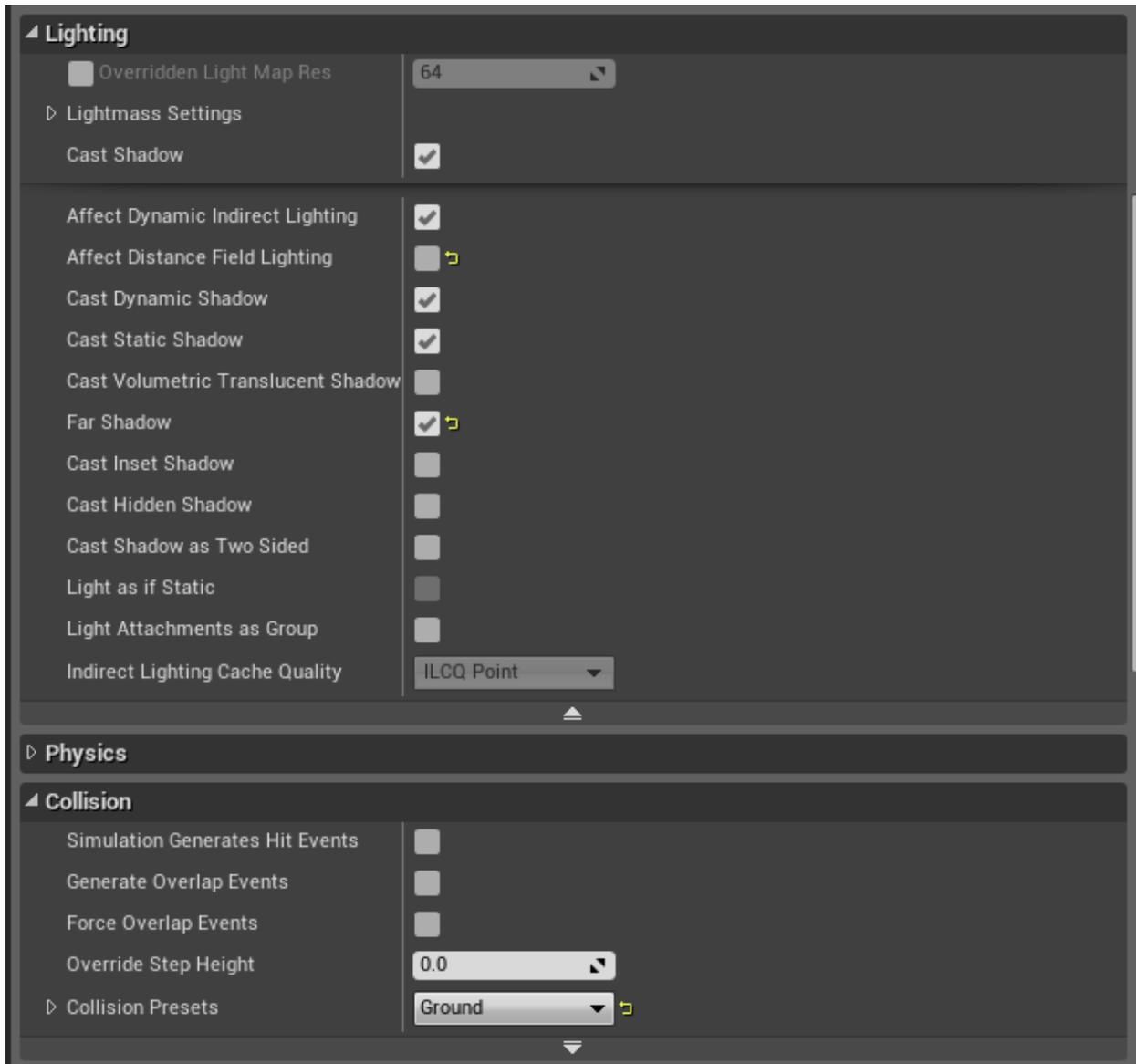
In ATLAS, Large Rocks are a special class of asset. A single Large Rock has the following properties:

- Covers significant gameplay area that would otherwise be Landscape
- Can be considered a major, characteristic terrain feature from far away
- Benefits greatly from receiving and casting far distance shadows

### 2.6.2 Requirements

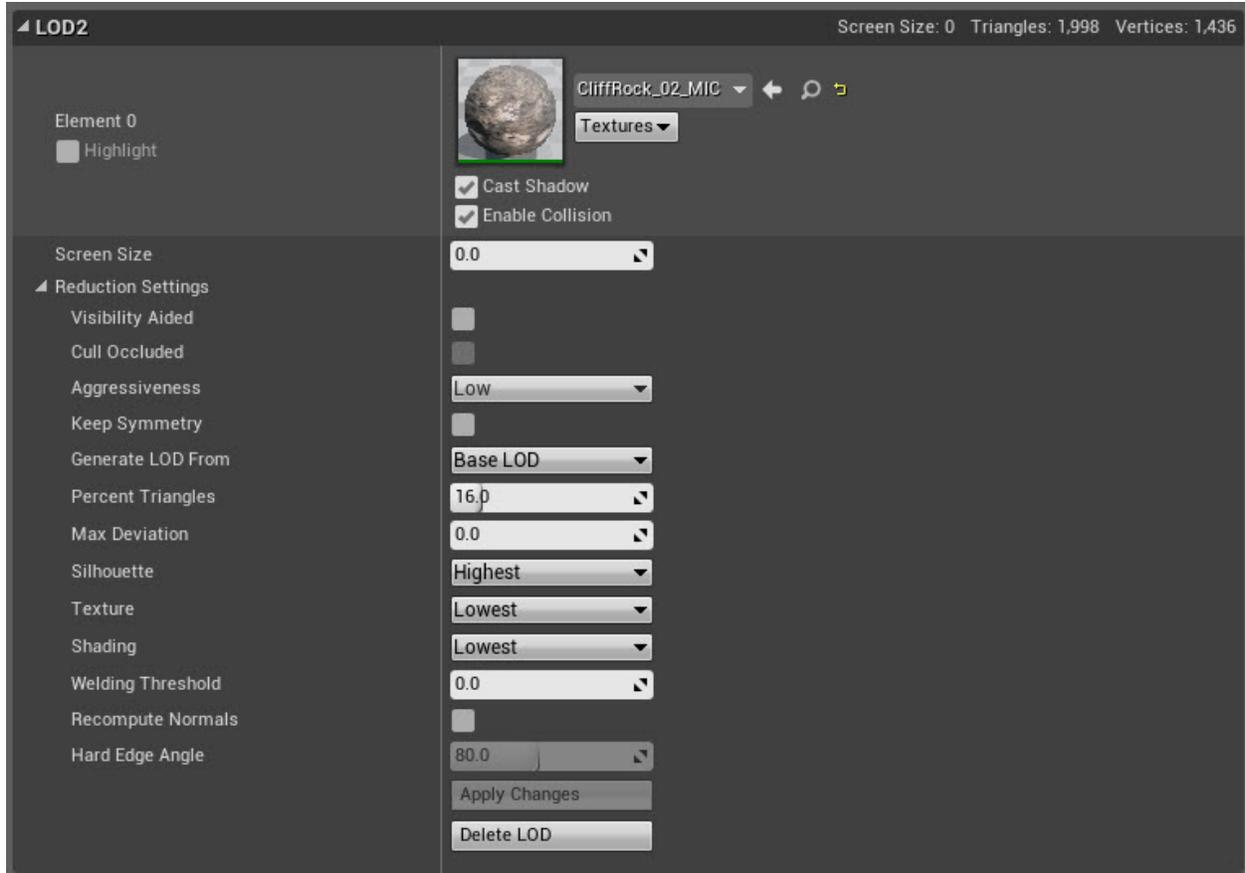
Large Rocks can be placed as StaticMeshes in Farthest sub-levels along with Large Trees, provided:

- Lowest LOD less than 4,000 tris
- **Use Distance Field Texture** = True (in StaticMesh editor)
- **Distance Field Two Sided Override** = DFTSO Override False (in StaticMesh editor)
- **Affect Distance Field Lighting** = True (in level actor - should be set automatically)
- **Collision Preset** = Ground (in level actor)
- **Desired Max Draw Distance** = 0 (Should be part of a proxy mesh parent when set to 0)
- **Use Absolute Max Draw Distance** = True



### 2.6.3 Collision

Large Rock collision is generated using Simplygon as the lowest LOD. This last LOD should NOT be more than 4k triangles but an exception can sometimes be made (ask). Silhouette should be set to Highest and both Texture/Shading should be set to Lowest. Screen Size should be set to 0 so that it is never used visually. Below are the correct settings.



## 2.7 Ships

### 2.7.1 Ship Components

#### Overview

Ships in Atlas are comprised of many components. This document covers them and related specifications.

#### Ribbing

The “ribbing” is comprised of a boat keel and the skeleton support beams that make up the shape of the hull. The ribbing is the first item that must be crafted and placed in a drydock in order to begin building a boat or ship. The ribbing mesh is also the root of all the other meshes, as far as the ship Blueprint is concerned (hull pieces and decks are offset from the ribbing).

The ribbing also has a Distance Field override mesh that serves to “cut out” the ocean water inside of the ship. “Used with Distance Field Lighting” must be set to true in order for the distance field to be generated.

#### Hull Pieces

Hull pieces are the individual structures that players must place to build a ship, and they can be individually damaged, repaired and replaced during the course of gameplay. All hull pieces must be placed on the ribbing before a new ship

can be launched to sea, and any hull pieces that are removed or damaged during gameplay will cause a ship to start sinking!

Hull pieces can all be completely unique.

## Decks

Decking are not required to launch a ship, but can be added to a ship at any time. Decks are comprised of a “frame” and snappoints, which can be used to attach ceiling and wall structures to. The overall size of each deck is arbitrary, but decks must be 220 units apart in height (standard Atlas wall height). The inner portion of decks should also be broken into as many 220x220 tile pieces as possible and removed, so ceilings can be placed inside the edges of each frame.

## Gunports

Specific hull pieces on a boat may be swapped out for a gunport, which is a window-wall made for housing a shipboard canon. Gunports have a standard size and standard height off the deck they are associated with.

## Gunport Doors

Coming soon...

## Sails

Coming Soon...

## Naming Conventions

SM_<Ship>_<Component>_Ribbing	Root Component of ship, must be unique! Creates a StaticMesh in UE4.
SM_<Ship>_<Component>_Frame	Sub-Component Root. Creates a StaticMesh in UE4.
SM_<Ship>_<Component>_Frame	Sub-Component Root. Creates a StaticMesh in UE4.
SM_<Ship>_<Component>_<Piece>	Unique Pieces of name-matched Component. Creates a StaticMesh in UE4.
SM_<Ship>_<Component>_<Piece>_Alt	Alt Piece of name-matched Piece. Creates a StaticMesh in UE4.
SM_<Ship>_<Component>_<Piece>_Inst_###	Instance of name-matched Piece. Uses StaticMesh from existing Piece.

## Example

- SM\_Brigantine\_Hull\_Ribbing
- SM\_Brigantine\_Hull\_Port\_Plank01
- SM\_Brigantine\_Hull\_Stbd\_Plank01
- SM\_Brigantine\_Hull\_Stbd\_Plank01\_Alt
- SM\_Brigantine\_Deck00\_Frame
- SM\_Brigantine\_Deck00\_FloorA

- SM\_Brigantine\_Deck00\_FloorA\_Inst\_01
- SM\_Brigantine\_Deck01\_Frame
- SM\_Brigantine\_Deck01\_FloorA
- SM\_Brigantine\_Deck01\_FloorA\_Inst\_01

## 2.8 Armor

Guides for creating new armor and armor skins.

### 2.8.1 Armor Skins

#### Naming Conventions

Source Folder Structure

- **AtlasSkins**
  - **Humans**
    - \* **Male**
      - **Outfits**
        - ArmorClass\_ExampleOutfit**
          - HM\_ArmorClass\_ExampleOutfit\_RIG.ma
          - SK\_HM\_ArmorClass\_ExampleOutfit\_Pants.fbx
          - SK\_HM\_ArmorClass\_ExampleOutfit\_Shirt.fbx
          - SK\_HM\_ArmorClass\_ExampleOutfit\_Pants.fbx
          - SK\_HM\_ArmorClass\_ExampleOutfit\_Shirt.fbx
        - Textures**
          - HM\_ExampleOutfit.spp
          - T\_HM\_ArmorClass\_ExampleOutfit\_Pants\_BC.psd
          - T\_HM\_ArmorClass\_ExampleOutfit\_Pants\_BC.tga (Base color)
          - T\_HM\_ArmorClass\_ExampleOutfit\_Pants\_L.tga (Layered texture)
          - T\_HM\_ArmorClass\_ExampleOutfit\_Pants\_colorized\_M.tga (Clothing mask)
          - T\_HM\_ArmorClass\_ExampleOutfit\_Pants\_colorized\_BC.tga (Desaturated Base Color)
          - T\_HM\_ArmorClass\_ExampleOutfit\_Pants\_M.tga (Colorization mask)
        - RAW**
          - HM\_ArmorClass\_ExampleOutfit.ztl
          - HM\_ArmorClass\_ExampleOutfit\_Baking.max

## Terms

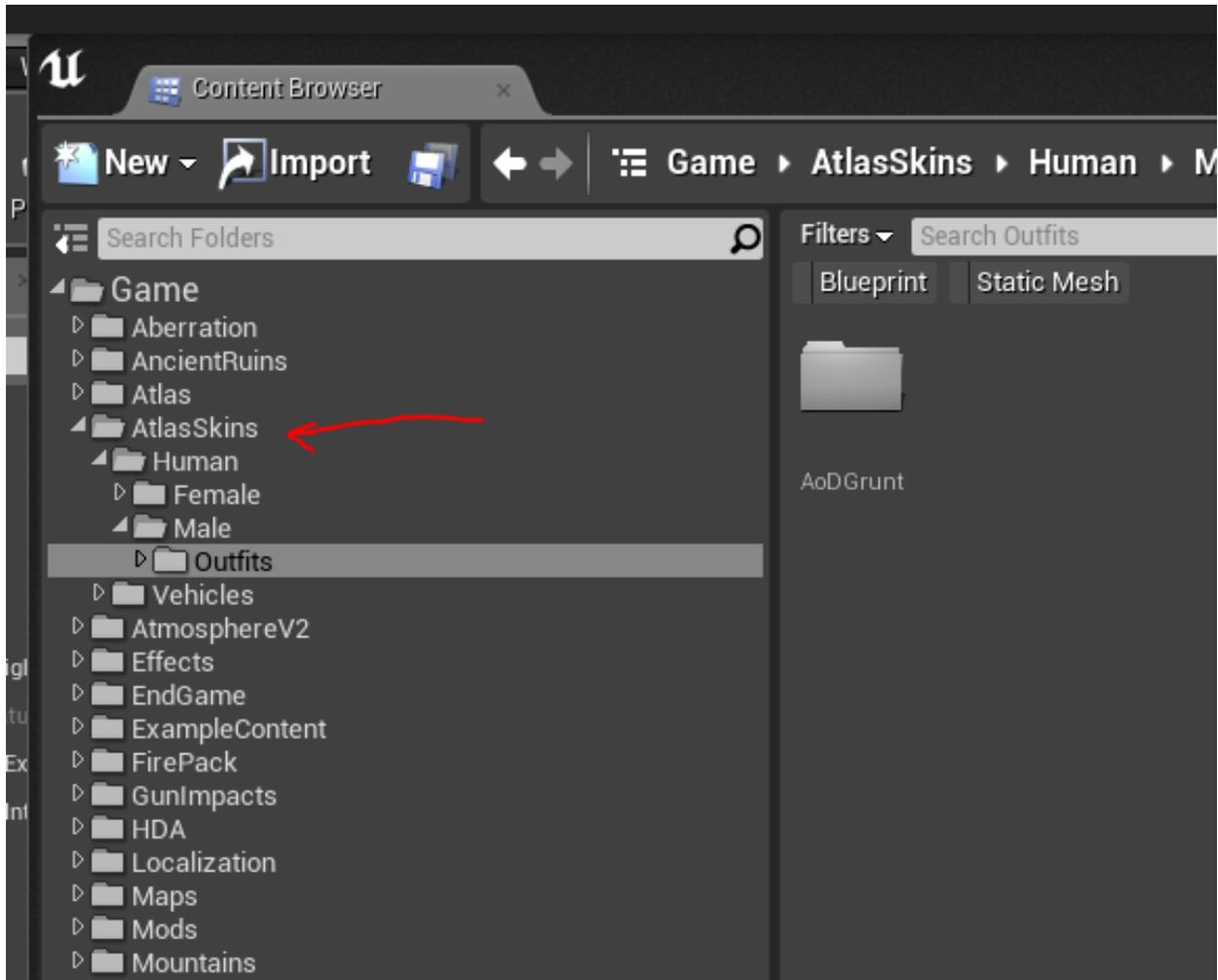
**Classes:** Heavy Hide Fur Cloth

**Pieces:** Helmet Shirt Gloves Pants Boots

HM= Human Male HF= Human Female

## Importing Armor Skins

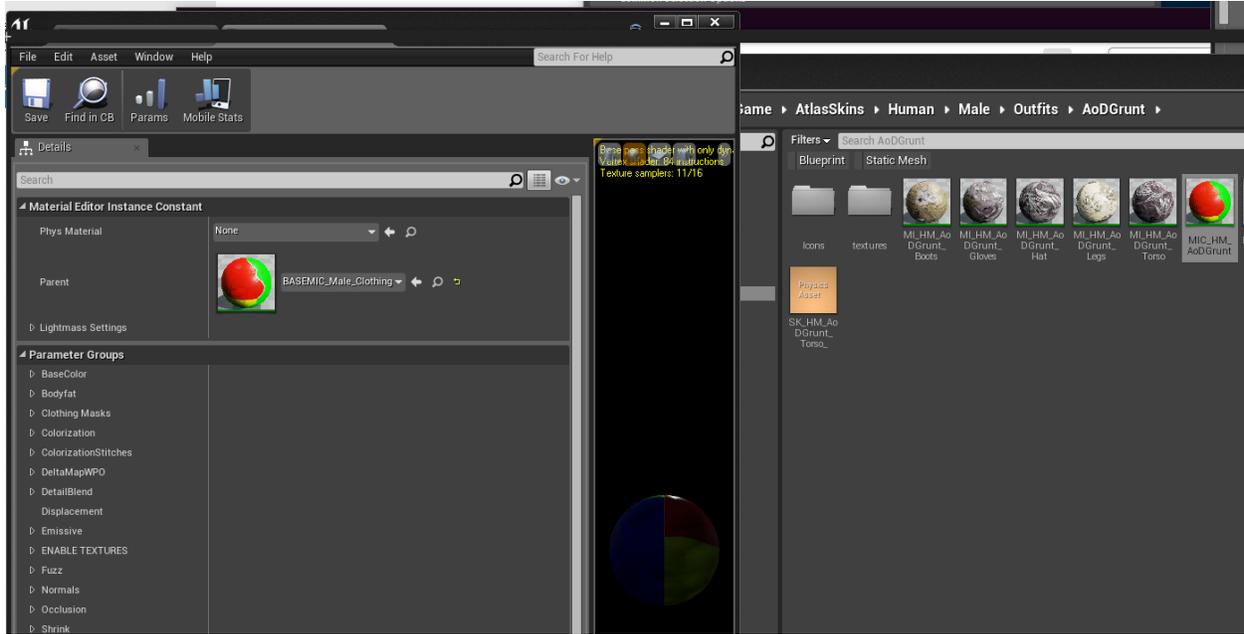
Import skinned FBX files to the AtlasSkins folder, which is on the same level as the Atlas folder.



Import path should be AtlasSkins>Human>Male(or Female)>Outfits>"SkinSetName"

## Assigning Materials

Create a material that will be the MIC for the entire armor skin set. This should be a material instance parented to the material "BASEMIC\_Male\_Clothing" for male, and "BASEMIC\_Female\_Clothing" for female. *Pay close attention to whether the BASEMIC is for male or female.* Male BASEMIC will not work for female and vice versa. Name this material as MIC\_HM(or HF)\_ArmorSkinName



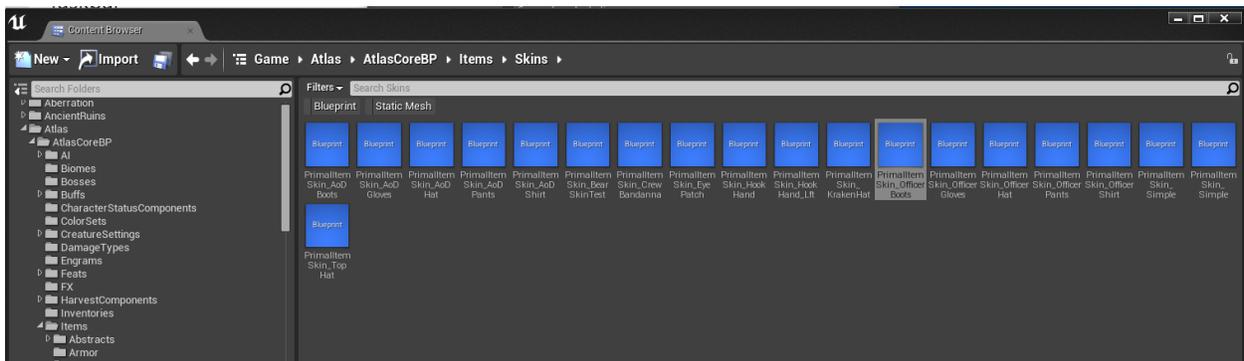
For each piece of armor, make a material instance of the MIC you just made : MI\_HM\_ArmorSkinName\_Boots, MI\_HM\_ArmorSkinName\_Shirt, for example.

Put textures for the set in a separate textures folder

## Creating Blueprints

### Creating a New Blueprint

Armor Skin Blueprints can be found in Atlas>AtlasCoreBP>Items>Skins



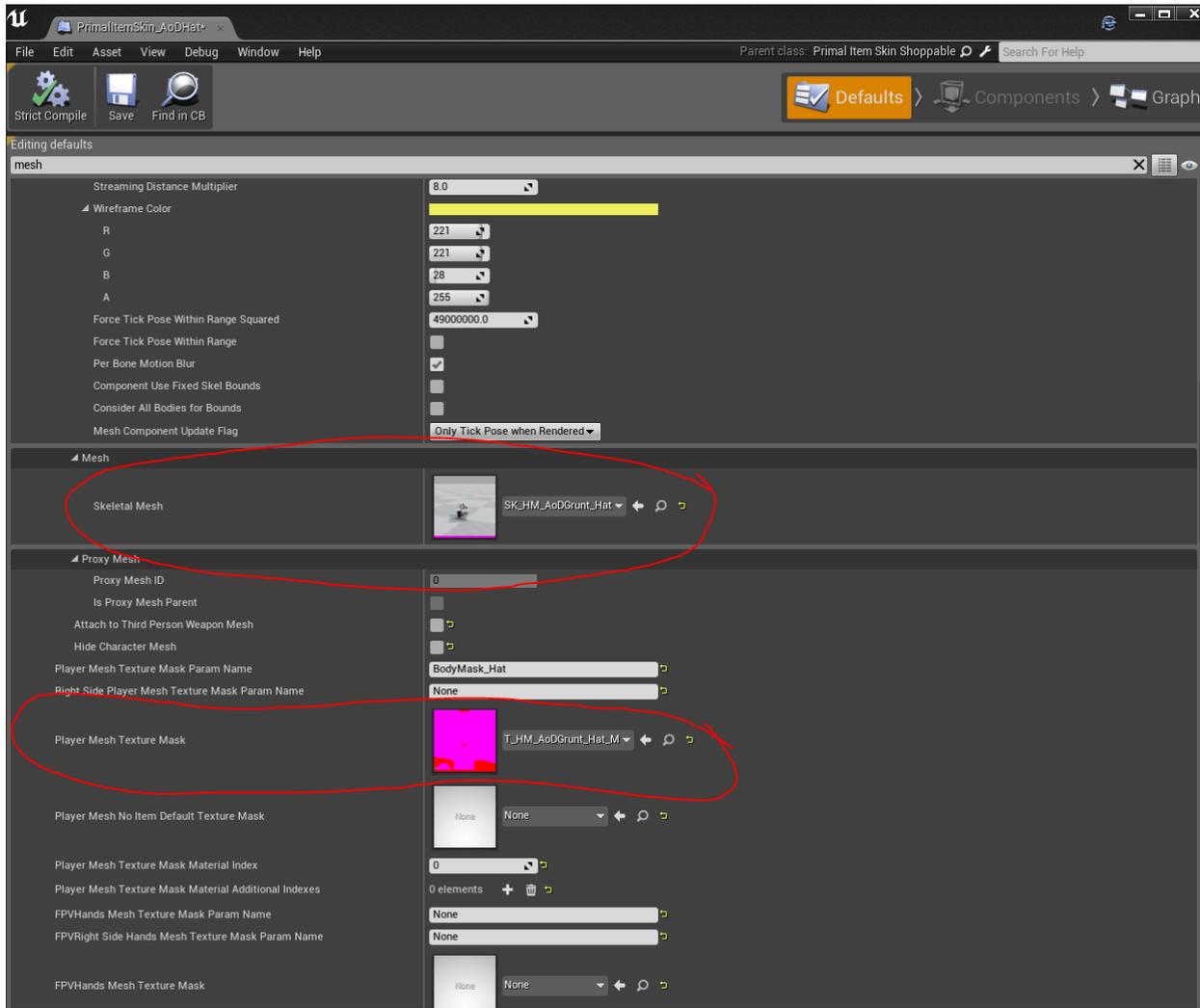
To make a new skin, duplicate one of the AoD skins for the armor piece you are making and rename it to match your asset.

It should be named “PrimalItemSkin\_ArmorClassArmorName\_ArmorPiece” For example, that might be “PrimalItemSkin\_HeavyHydra\_Gloves”

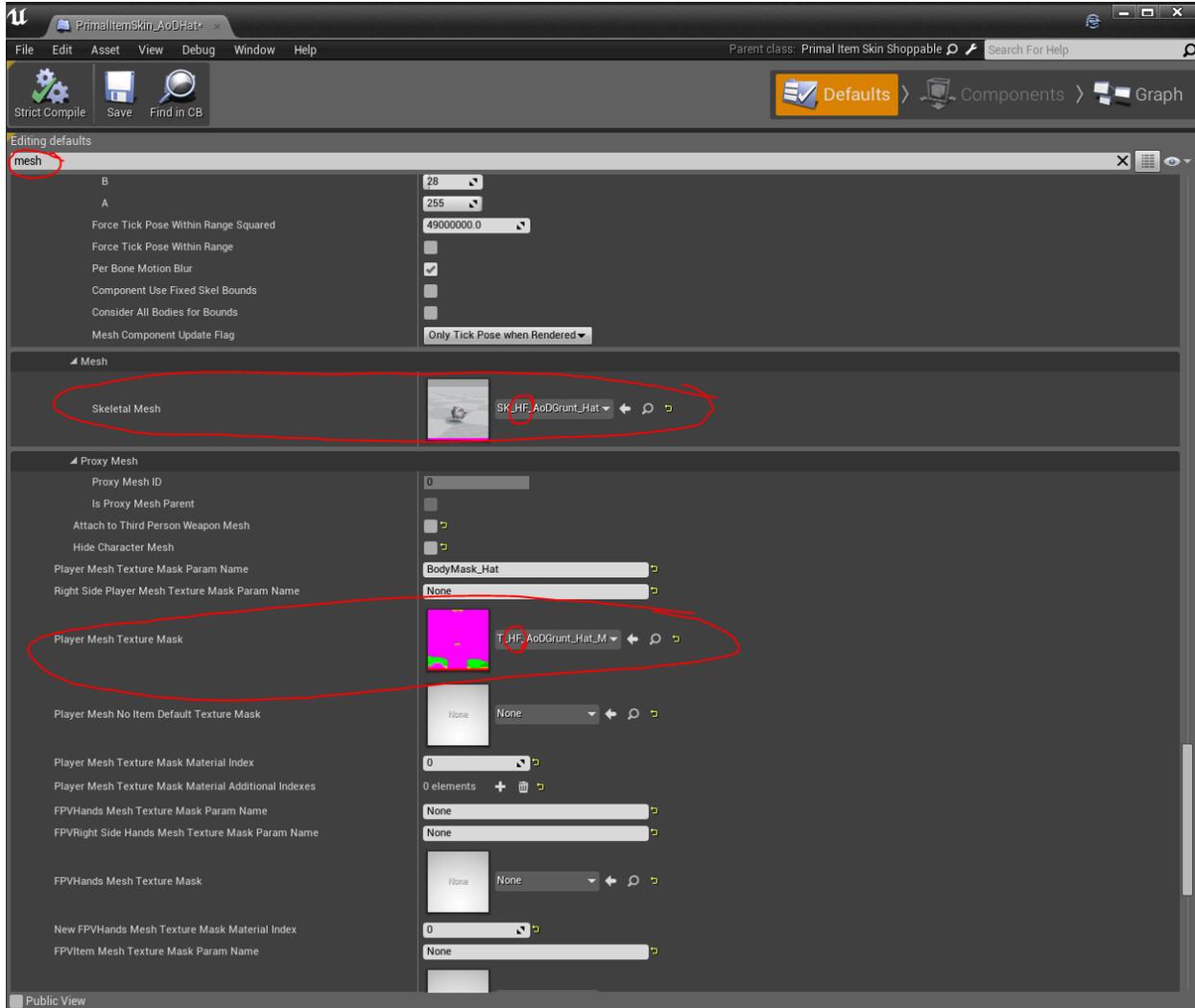
### Editing the New Blueprint

Open the blueprint you just created. Type “mesh” into the search bar to narrow things down and find the parameters you will need to edit.

As you scroll down, the first set of mesh and texture slots you come to will be for the “male” version of the armor. Hook up your SK mesh and your clothing mask here.



As you scroll down farther the next set of mesh and mask slots will be for the female version of the armor



This process is going to be different for **glove assets**

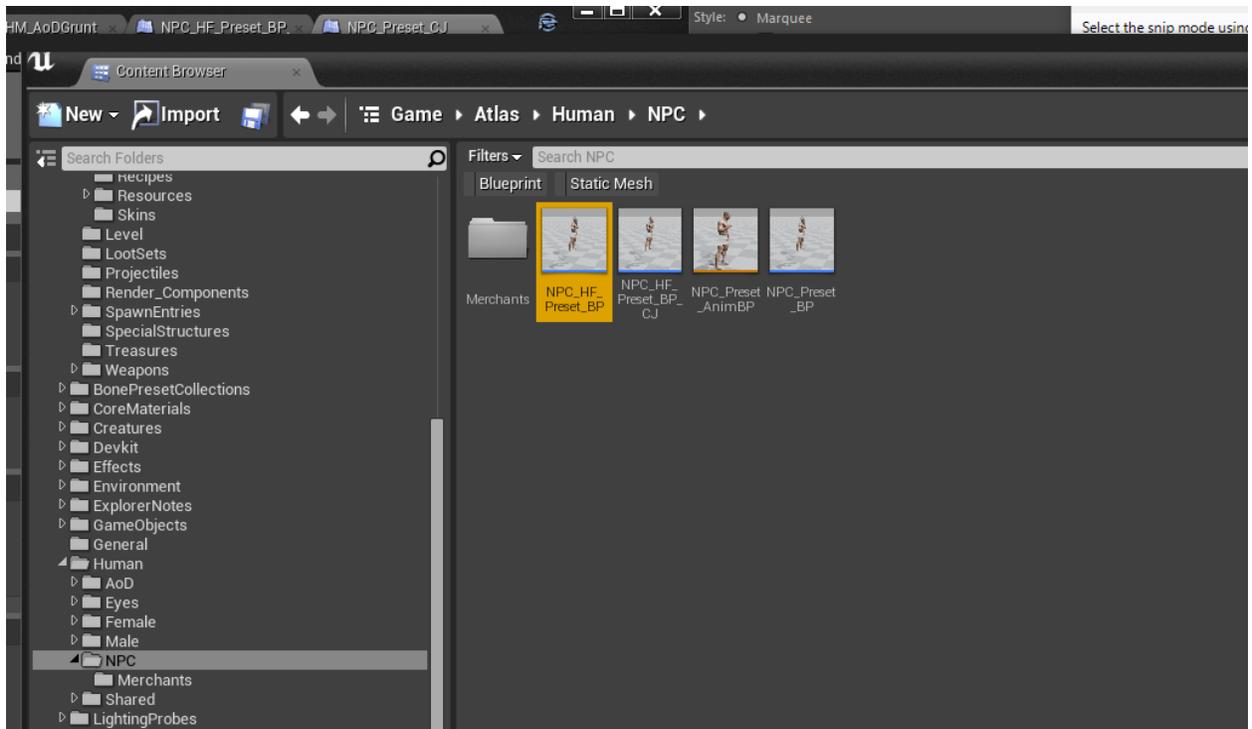
For gloves, there are two slots for each mesh and texture, one for first person and one for third person. So if you are scrolling down in a glove blueprint, you will need to hook up the male assets for the first two sets of slots for mesh and mask and the female assets for the second two sets

## Testing Armor Skins

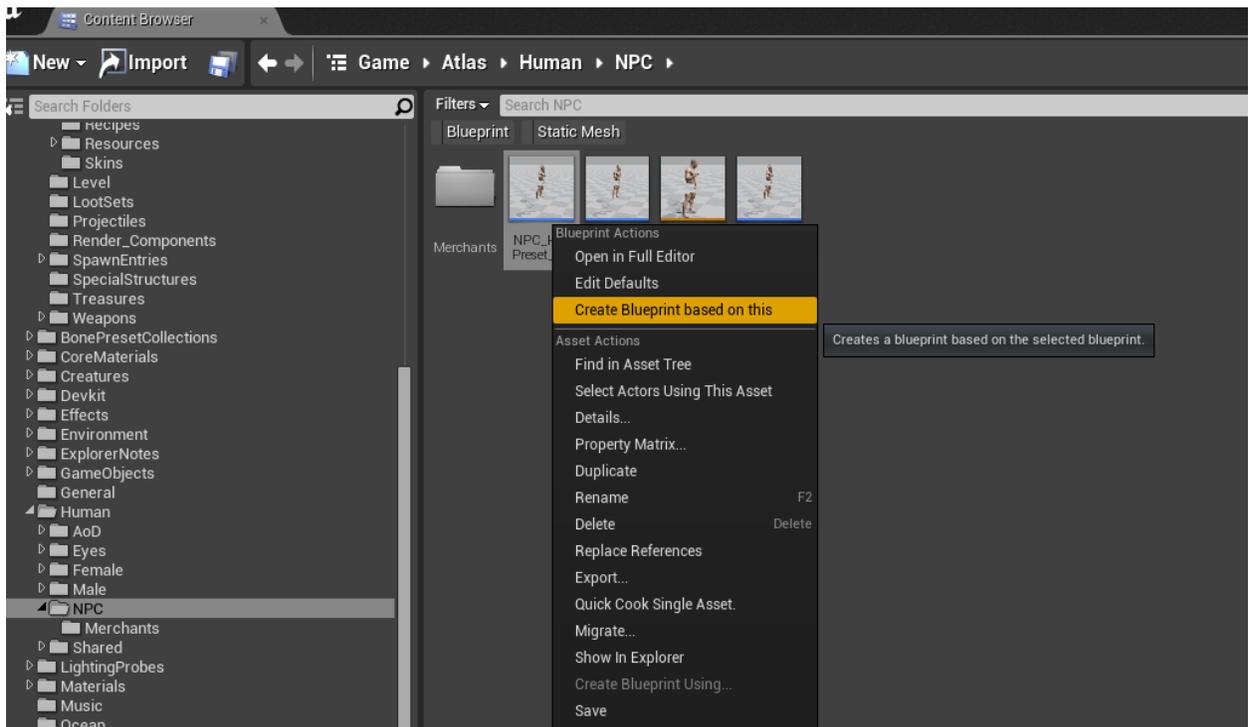
### Creating an NPC Preset

It is easiest to test new armor assets on an NPC. You should make your own test NPCs on your local machine using these steps. Do not commit your test NPCs to SVN

For Female NPCs, find NPC\_HF\_Preset\_BP



To make your own test NPC, right click and select “Create Blueprint Based on this”



Name your new NPC Blueprint whatever you want, since you **won't be committing it**

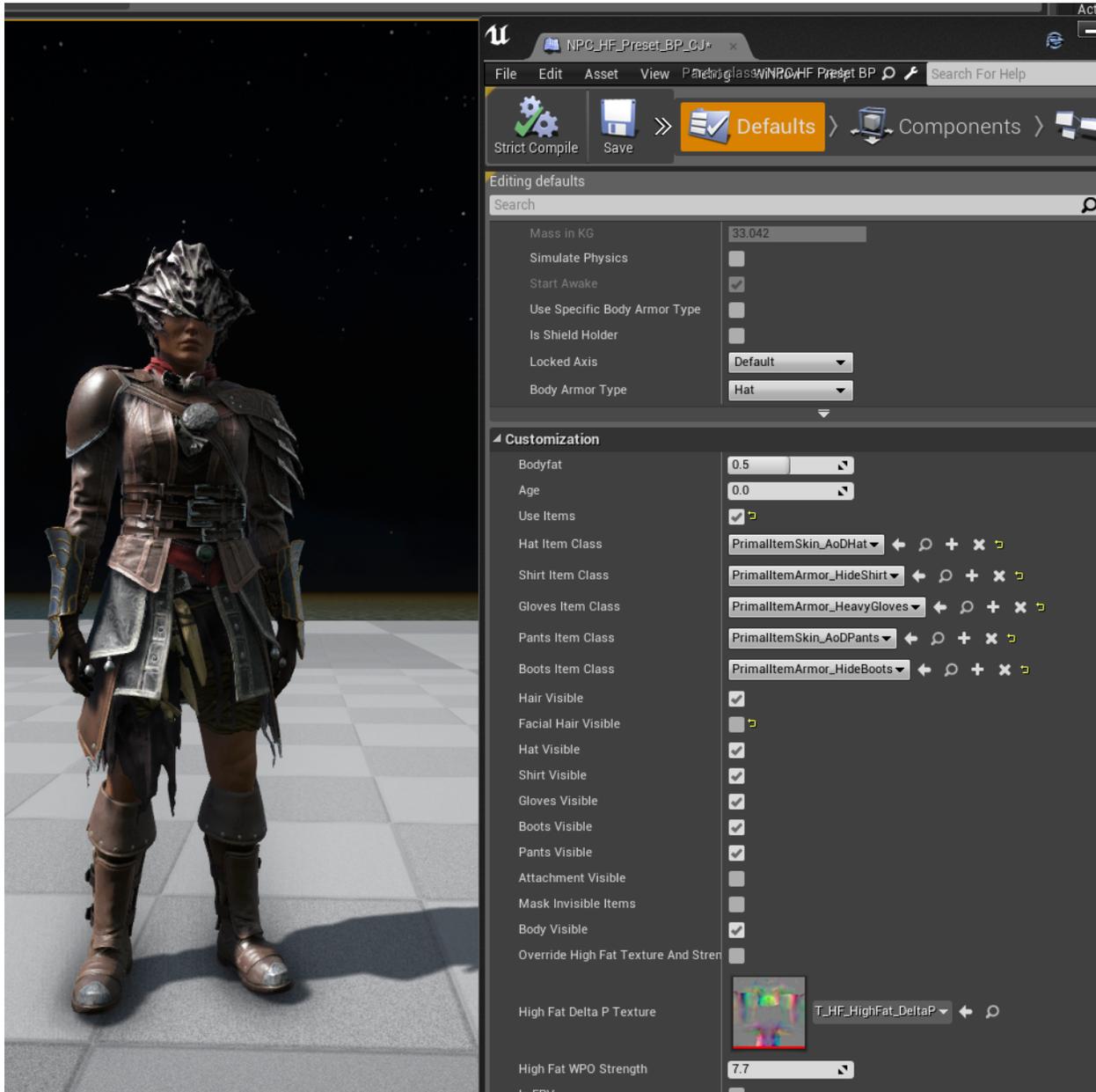
For male NPCs, create a blueprint based on “NPC\_Preset\_BP

## Using an NPC preset to test clothing

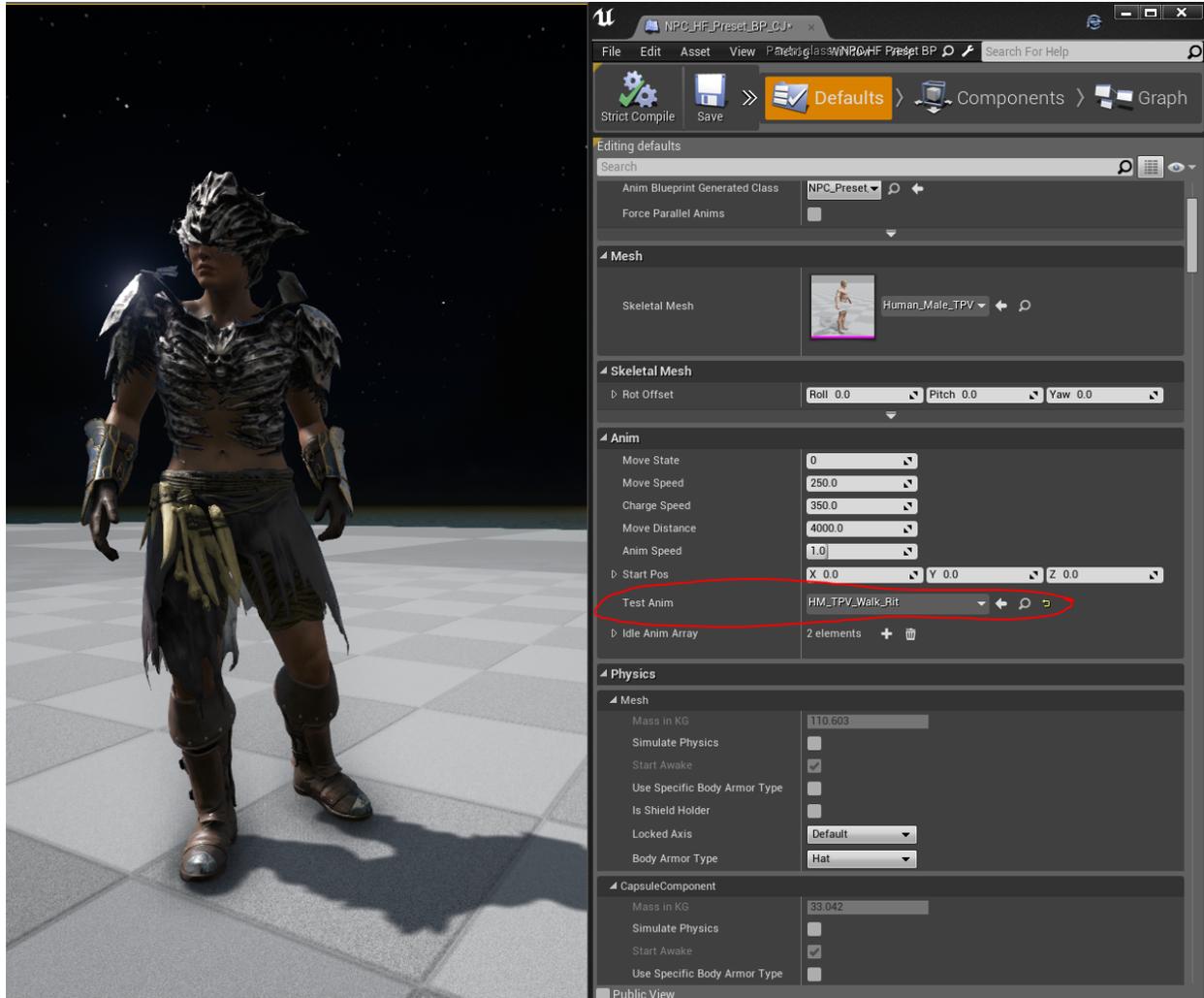
If you put your newly-created NPC in the scene, you will be able to test a bunch of different things.

Under “Customization” you can adjust the Bodyfat to make sure your delta UVs are working correctly. You can also choose which items will be visible on the character in the scene.

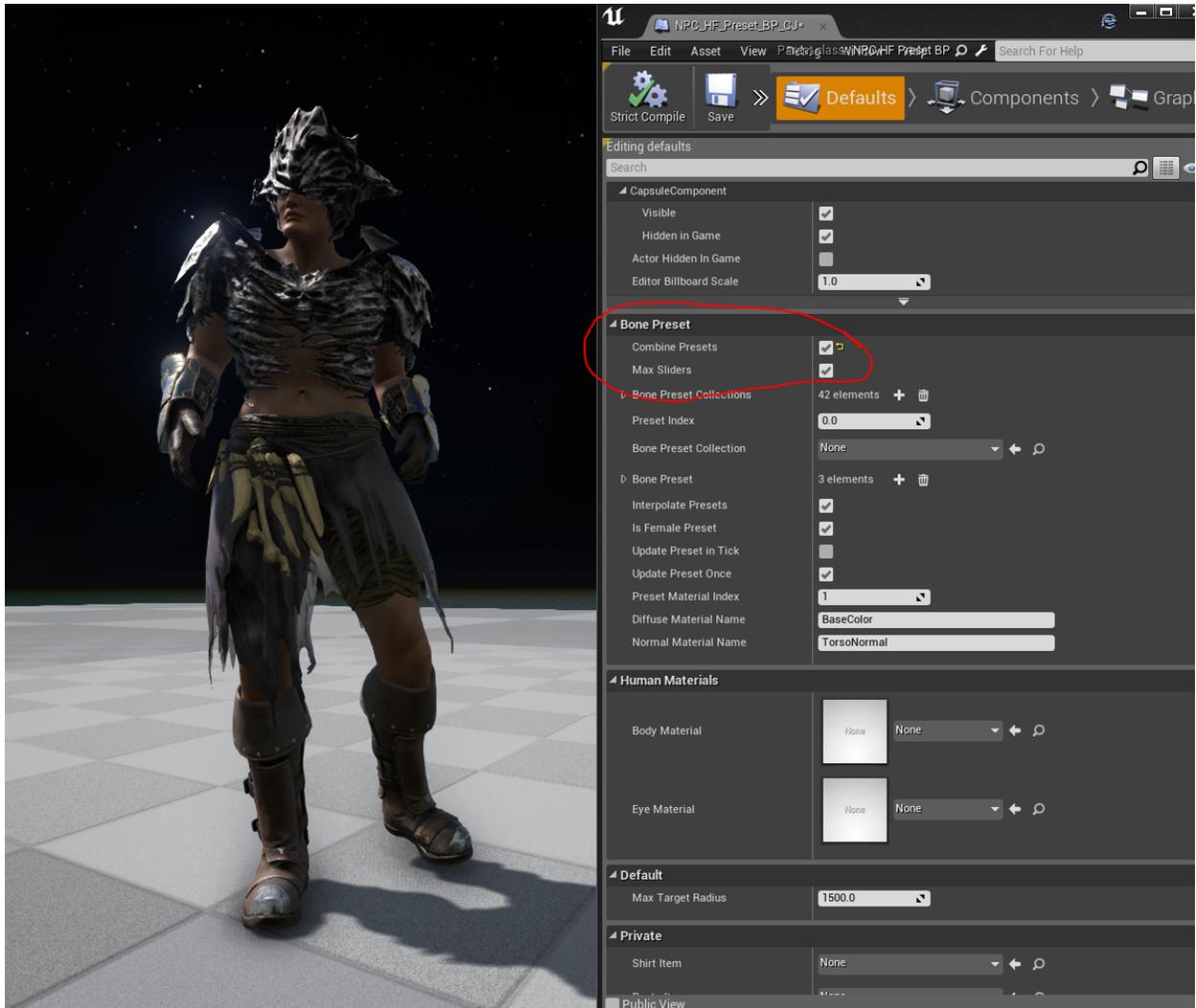
If you have “Use Items” checked, you can simply plug in the blueprints you made for the skin items and test them against other clothing items. This will help visualize if shrink masks and clothing layers are set up properly.



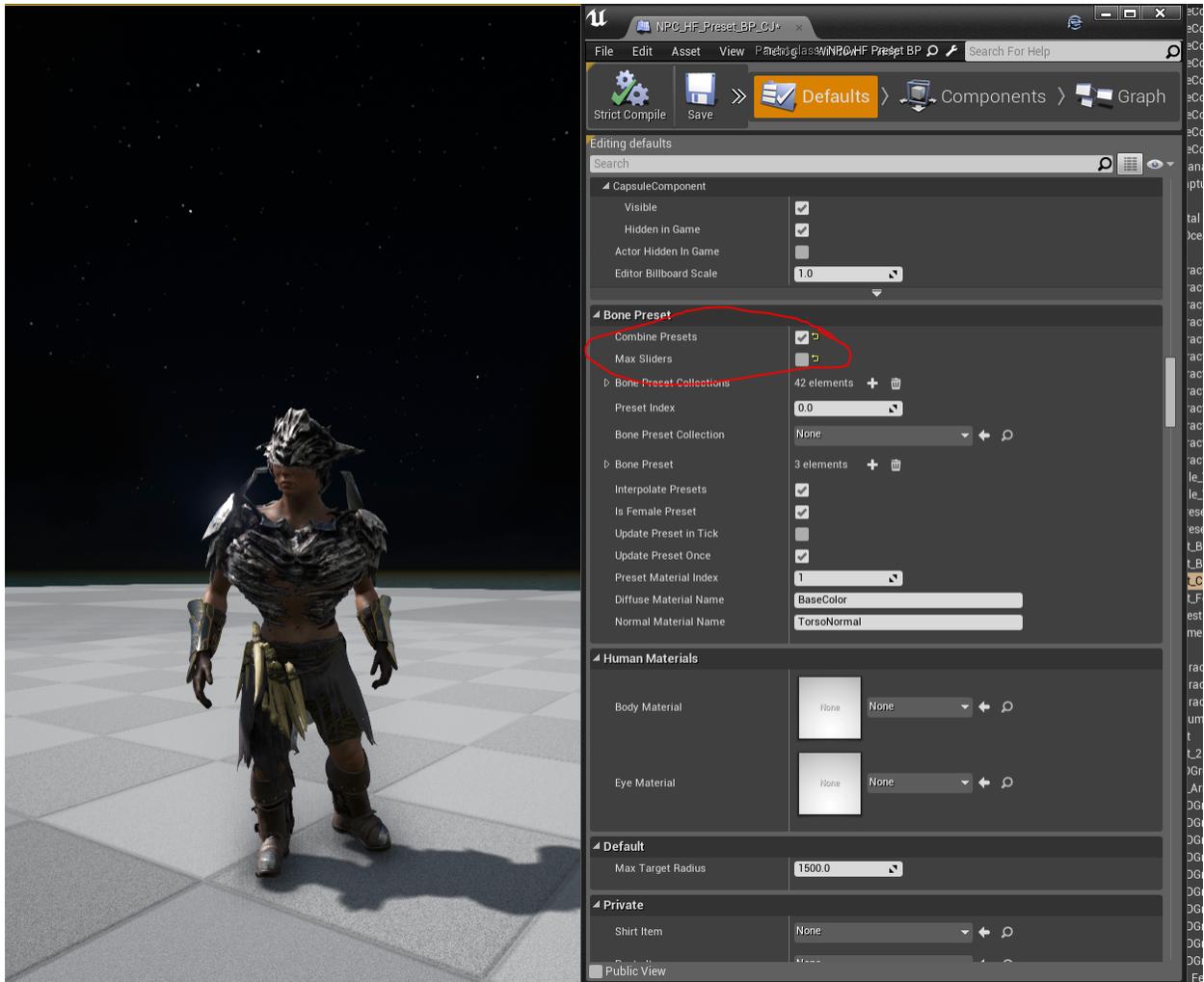
Under Anim>Test Anim you can change the animation that is playing in order to better test skinning



You can see how the armor looks with maximum bone scale settings by going to the Bone Preset dropdown and checking on “Combine Presets” and “Max Sliders”



Minimum bone scale settings can be looked at by having “Combine Presets” checked and “Max Sliders” unchecked



## Testing while Playing in Editor

Test while playing in editor to check on things like FPV of gloves and further testing of skinning and combined clothing pieces.

Once you are in play mode, give yourself the base armor sets by copying and pasting this list after hitting “~”

```

cheat giveitem "/Game/PrimalEarth/CoreBlueprints/Items/Armor/Cloth/PrimalItemArmor_
↵ClothBoots.PrimalItemArmor_ClothBoots" 1 0 0 false |
cheat giveitem "/Game/PrimalEarth/CoreBlueprints/Items/Armor/Cloth/PrimalItemArmor_
↵ClothGloves.PrimalItemArmor_ClothGloves" 1 0 0 false |
cheat giveitem "/Game/PrimalEarth/CoreBlueprints/Items/Armor/Cloth/PrimalItemArmor_
↵ClothHelmet.PrimalItemArmor_ClothHelmet" 1 0 0 false |
cheat giveitem "/Game/PrimalEarth/CoreBlueprints/Items/Armor/Cloth/PrimalItemArmor_
↵ClothPants.PrimalItemArmor_ClothPants" 1 0 0 false |
cheat giveitem "/Game/PrimalEarth/CoreBlueprints/Items/Armor/Cloth/PrimalItemArmor_
↵ClothShirt.PrimalItemArmor_ClothShirt" 1 0 0 false |
cheat giveitem "/Game/Atlas/AtlasCoreBP/Items/Skins/PrimalItemSkin_OfficerBoots.
↵PrimalItemSkin_OfficerBoots" 1 0 0 false |
cheat giveitem "/Game/Atlas/AtlasCoreBP/Items/Skins/PrimalItemSkin_OfficerGloves.
↵PrimalItemSkin_OfficerGloves" 1 0 0 false |

```

(continues on next page)

(continued from previous page)

```
cheat giveitem "/Game/Atlas/AtlasCoreBP/Items/Skins/PrimalItemSkin_OfficerHat.  
↪PrimalItemSkin_OfficerHat" 1 0 0 false |  
cheat giveitem "/Game/Atlas/AtlasCoreBP/Items/Skins/PrimalItemSkin_OfficerPants.  
↪PrimalItemSkin_OfficerPants" 1 0 0 false |  
cheat giveitem "/Game/Atlas/AtlasCoreBP/Items/Skins/PrimalItemSkin_OfficerShirt.  
↪PrimalItemSkin_OfficerShirt" 1 0 0 false |  
cheat giveitem "/Game/PrimalEarth/CoreBlueprints/Items/Armor/Cloth/PrimalItemArmor_  
↪HeavyBoots.PrimalItemArmor_HeavyBoots" 1 0 0 false |  
cheat giveitem "/Game/PrimalEarth/CoreBlueprints/Items/Armor/Cloth/PrimalItemArmor_  
↪HeavyGloves.PrimalItemArmor_HeavyGloves" 1 0 0 false |  
cheat giveitem "/Game/PrimalEarth/CoreBlueprints/Items/Armor/Cloth/PrimalItemArmor_  
↪HeavyHelmet.PrimalItemArmor_HeavyHelmet" 1 0 0 false |  
cheat giveitem "/Game/PrimalEarth/CoreBlueprints/Items/Armor/Cloth/PrimalItemArmor_  
↪HeavyPants.PrimalItemArmor_HeavyPants" 1 0 0 false |  
cheat giveitem "/Game/PrimalEarth/CoreBlueprints/Items/Armor/Cloth/PrimalItemArmor_  
↪HeavyShirt.PrimalItemArmor_HeavyShirt" 1 0 0 false |  
cheat gfi hideboots 1 0 0 | cheat gfi hidegloves 1 0 0 | cheat gfi hidehelmet 1 0 0 |  
↪cheat gfi hidepants 1 0 0 | cheat gfi hideshirt 1 0 0 |  
cheat gfi furboots 1 0 0 | cheat gfi furgloves 1 0 0 | cheat gfi furhelmet 1 0 0 |  
↪cheat gfi furlpant 1 0 0 | cheat gfi furshirt 1 0 0
```

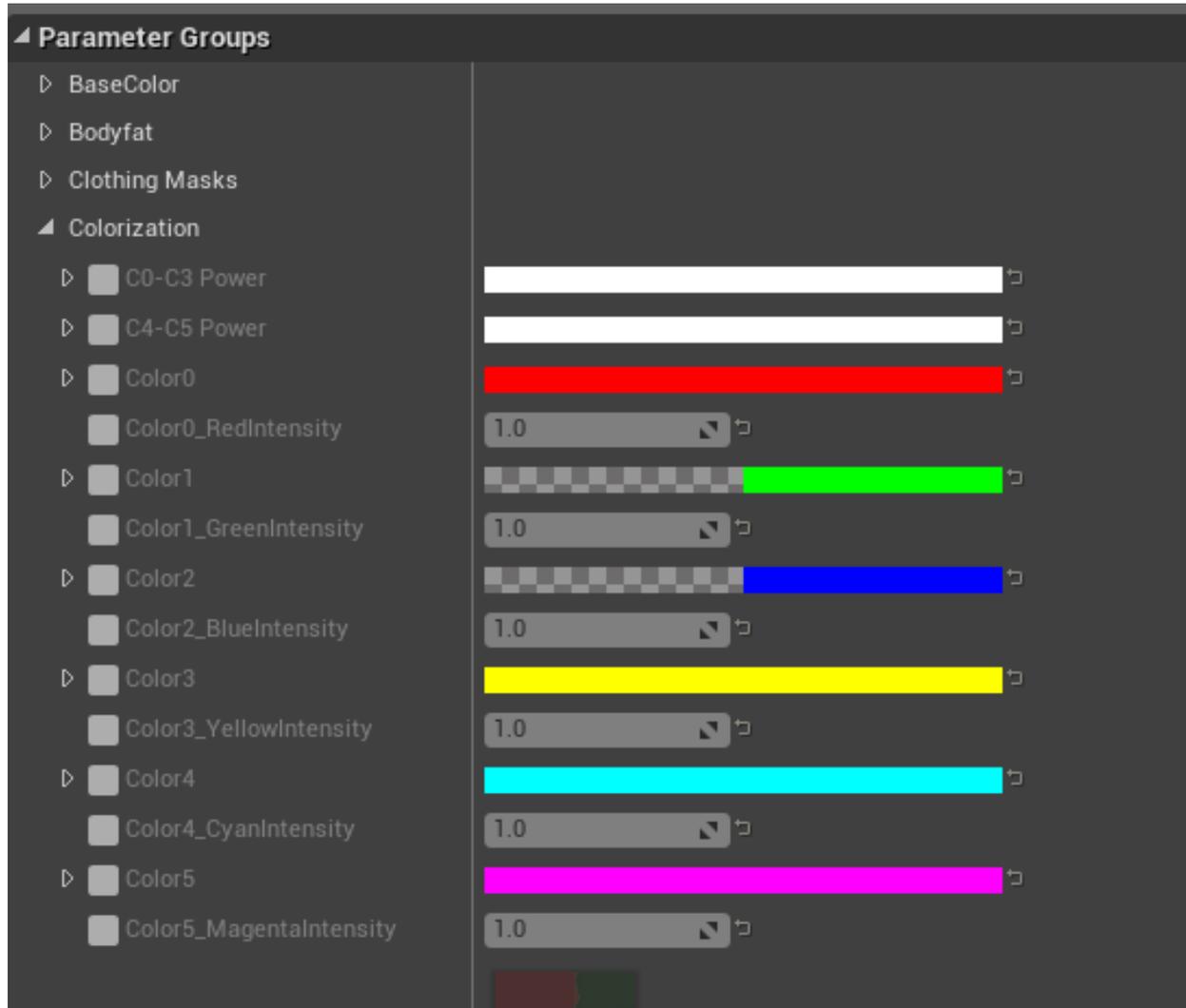
This will put all of these items in your inventory. To give yourself your new Skin items you want to test, right click on the BP in the content browser and give yourself the item in game. Skins will need to be applied to existing clothing pieces in your inventory for you to view them.

Pay close attention to how gloves look in first person while testing this way.

To make a female character you will need to type “suicide” into the command bar and create a new character. Same thing with any new bone slider adjustments

## 2.8.2 Clothing Material

## Colorization



Colorization works on clothing material the same way that it works for other materials in the game. However, the color regions drive a lot of other features like detail tiling and cloth shading.

See here for an overview on colorization: [Colorization](#)

Parameters that refer back to color regions are often named differently in the material, so consult this guide to figure those out

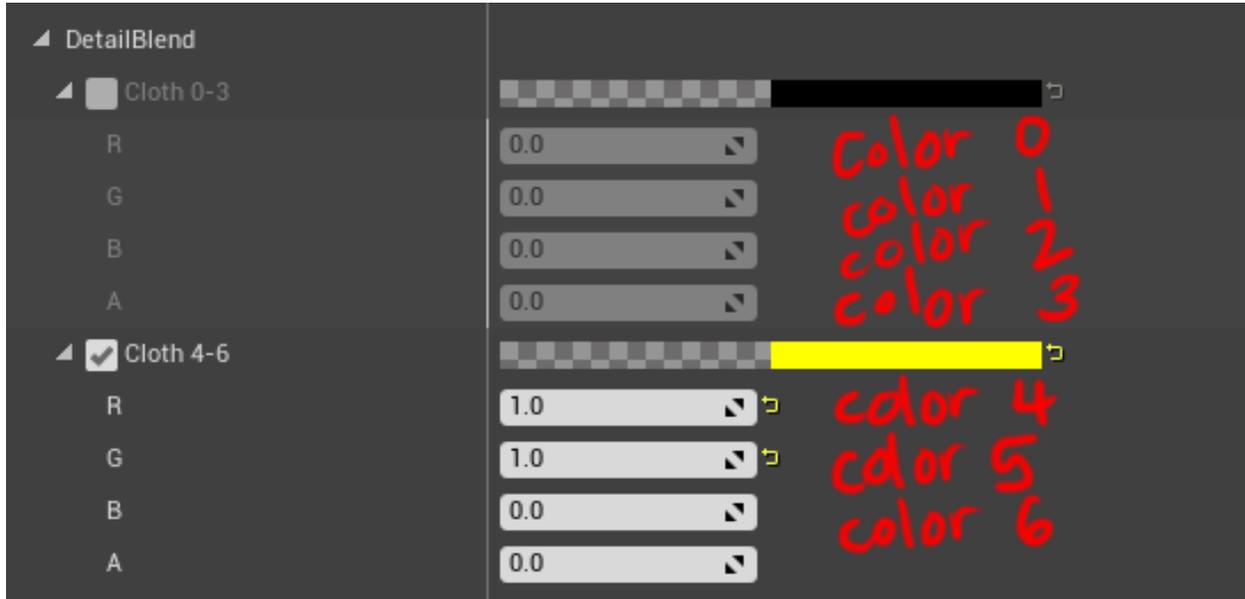
## Delta Map WPO

Some bugs involving body fat can be adjusted a little bit in this dropdown. High Fat WPO strength can be used to customize how much the body fat will affect that item of clothing. Be careful with how this ends up interacting with other pieces of clothing, though. Sometimes bugs with the smoothness of the surface of the clothing due to body fat can be helped by checking on “Use Body Distance”. This option prevents clothing parts that already fit around the High Fat body from inflating further.

## Detail Blend

### Cloth

You can determine whether a color region will have “cloth fuzz” applied by setting the parameters here. 0=no cloth, 1=cloth. The RGBA selections correspond to the color regions this way:



### Detail Textures

Plug in texture maps for detail textures here. Detail L\_0 corresponds to color region 0, Detail L\_1 corresponds to color region 1, etc.

The image shows a configuration panel for material details, organized into four sections: Detail\_L0, Detail\_L1, Detail\_L2, and Detail\_L3. Each section contains a checked checkbox, a texture preview, a dropdown menu, and several sliders. Red circles highlight the texture preview and dropdown menu for each level.

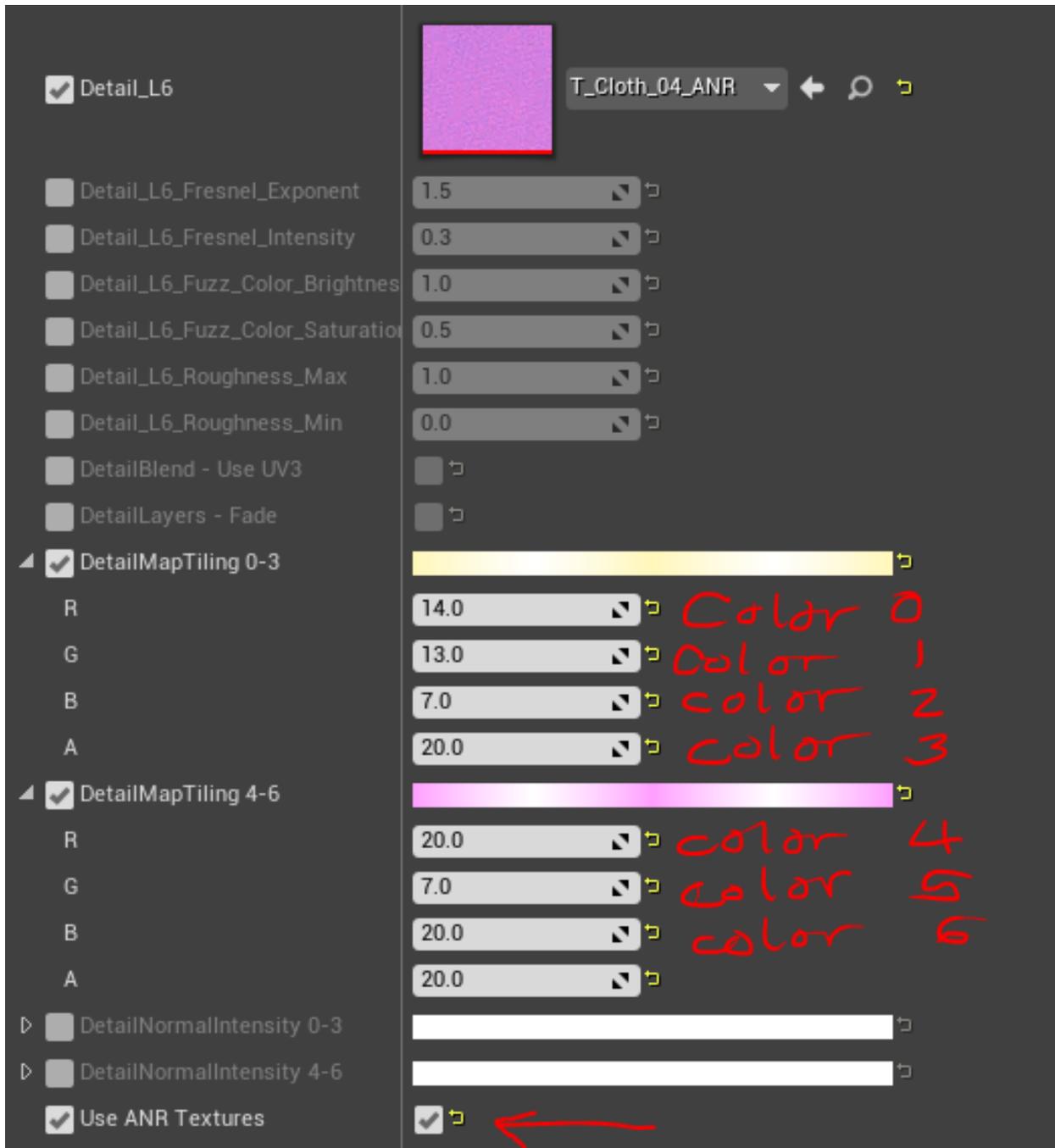
Level	Detail	Texture	Dropdown	Fresnel_Exponent	Fresnel_Intensity	Fuzz_Color_Brightness	Fuzz_Color_Saturation	Roughness_Max	Roughness_Min
Detail_L0	<input checked="" type="checkbox"/>		T_Leather_03_L	1.5	0.3	1.0	0.5	1.0	0.0
Detail_L1	<input checked="" type="checkbox"/>		T_Leather_04_L	1.5	0.3	1.0	0.5	1.0	0.5
Detail_L2	<input checked="" type="checkbox"/>		T_Ornamental_02_L	1.5	0.3	1.0	0.5	1.0	0.5
Detail_L3	<input checked="" type="checkbox"/>		T_Metal_01_L	1.5	0.3	1.0	0.5	1.0	0.0

Normal maps can be used here, but so can layered ANR textures. ANR textures are set up in this way:

- R - AO / Cavity
- G - Normal G
- B - Roughness
- A - Normal R

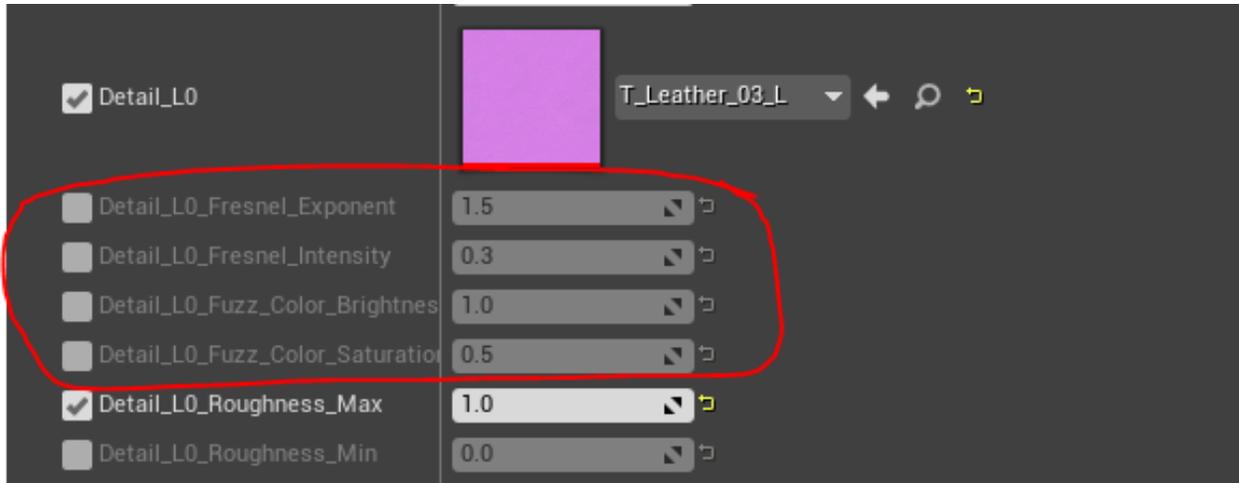
New ANR textures can be saved in `Game>Atlas>Human>Shared>TilingTextures`. They should have a suffix of “ANR” (Some textures in that folder currently are named with the suffix “L” even though they are ANR textures).

To use ANR textures, make sure “Use ANR textures” is checked on all the way at the bottom of the Detail Blend dropdown. Control the detail tiling and the detail normal intensity at the bottom of the dropdown as well, with the color regions corresponding in the way illustrated below



## Fuzz

The appearance of the fuzz in cloth-enabled color regions is controlled here:



The roughness for each tiling texture can also be controlled here.

## Shrink

When you create Delta UVs with either the Clothing Wizard or the Clothing HDA plugin, you will get a number value called “max shrink amount”. Plug that in here.



## Stitches

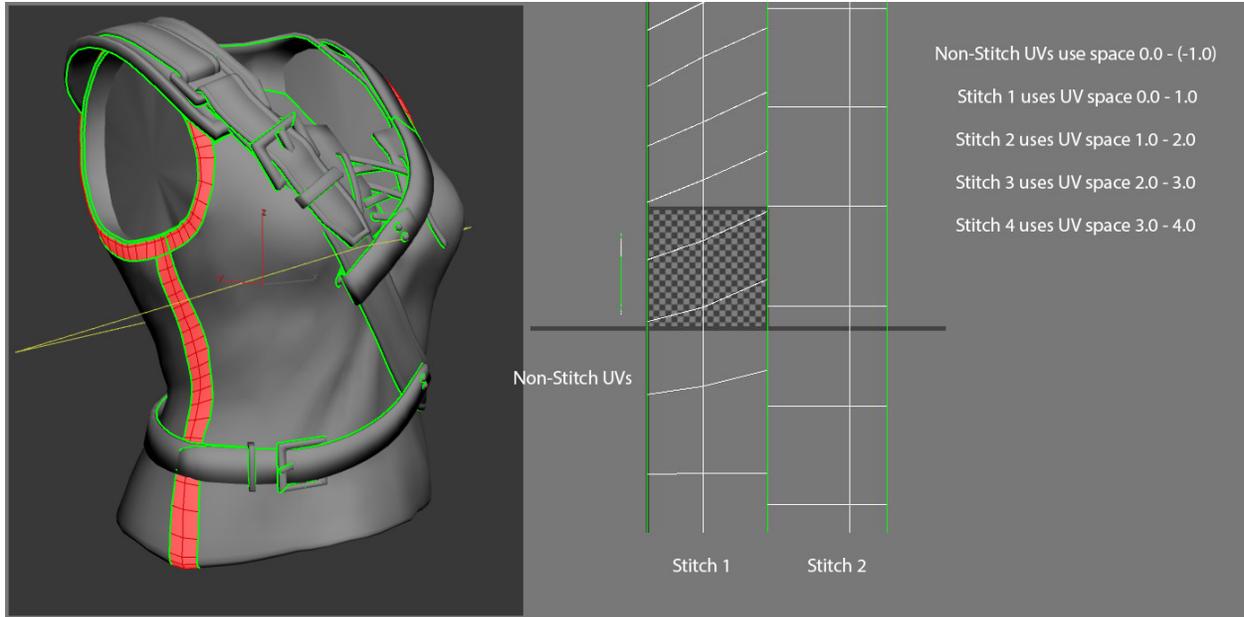
### Laying out Stitch UVs

We use up to 4 types of stitches per clothing part.

Stitches are found here: A:CharactersHumansSharedStitches

The UVs should be spread out into 4 regions: 0-1, 1-2, 2-3, 3-4

Parts that don't belong to a stitch type should be placed in the negative U space region: (0, 0)(-1,1)



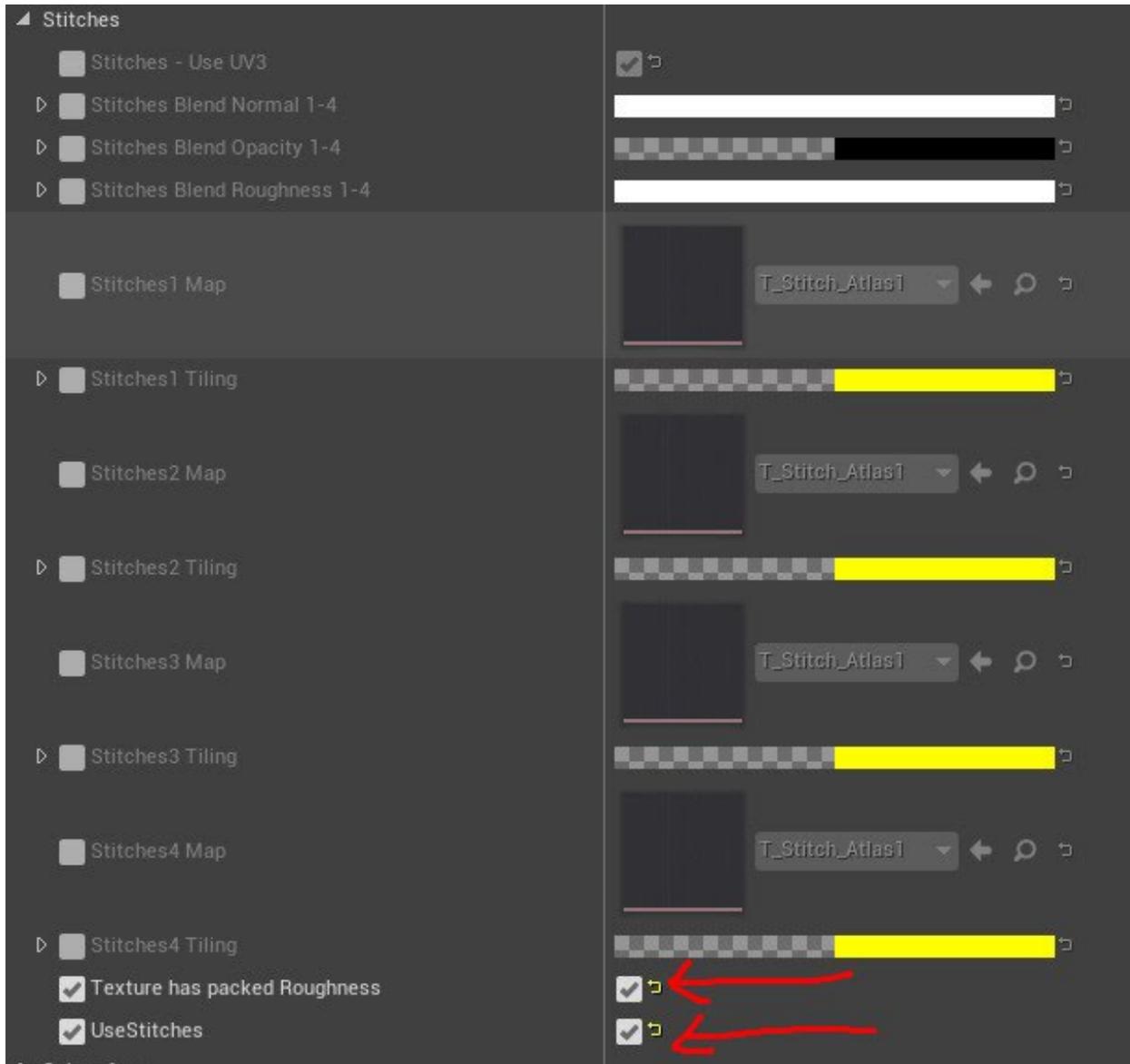
## Stitch Textures

We use a packed NRO map for stitches:

NRO

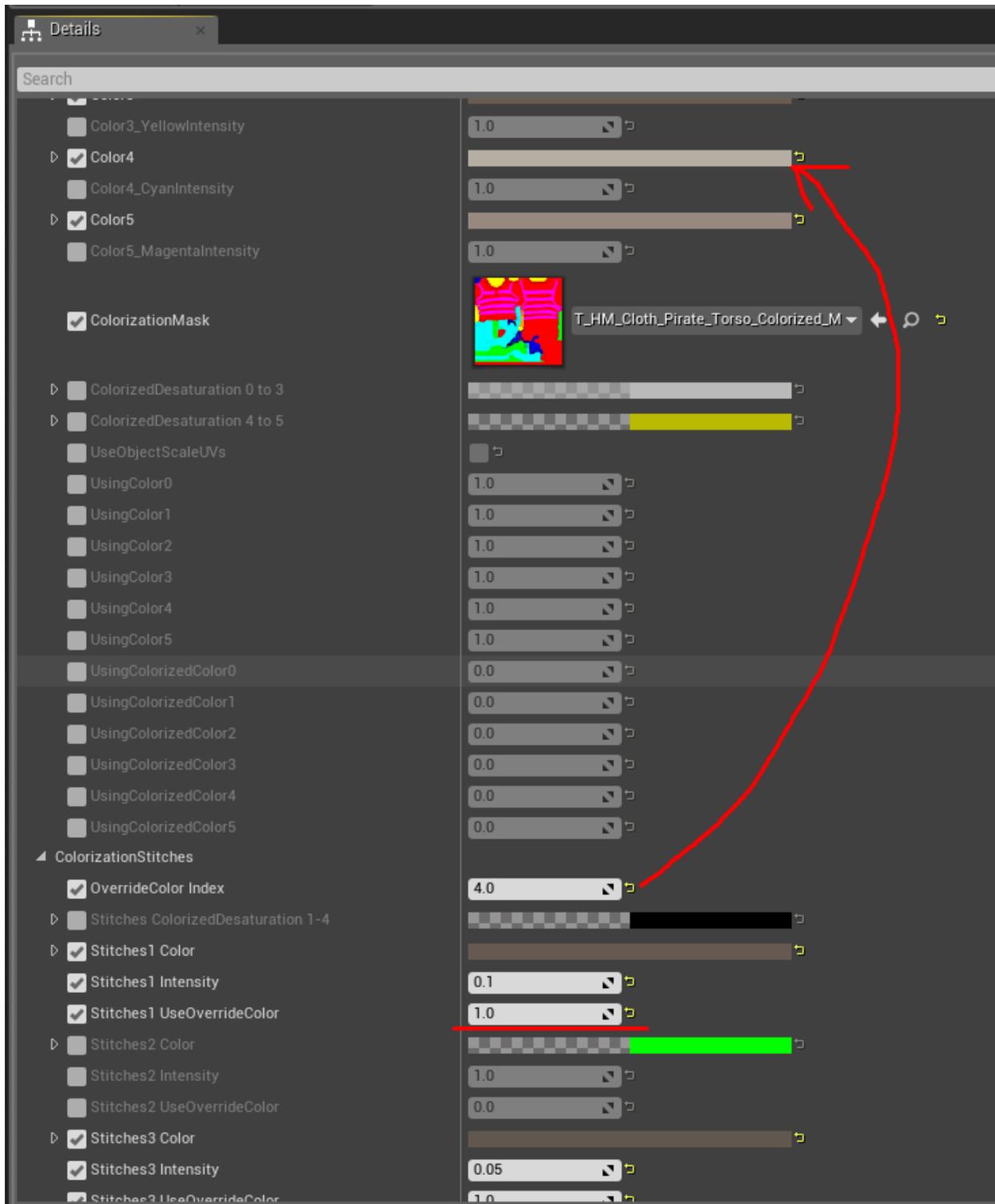
- R - Normal R
- G - Normal G
- B - Roughness
- A - Opacity

To enable stitches, you need to enable the switch “UseStitches”. If you do not need Roughness in the texture, you can disable the switch “Texture has packed Roughness”.



## Colorizing Stitches

Each of the 4 regions can be colorized manually in the “ColorizationStitches” category. To make stitches colorizable by the player, you can use the “OverrideColor”. You can enable it for each of the 4 stitches separately. For example to enable it for the first stitches region, set “Stitches1 UseOverrideColor” to 1 and set the color index to use with the “OverrideColor Index” parameter. In the below example, when the player dyes color region 4 this will also colorize the first Stitch region because it is set to be overridden by Color4.



## FPV

Parts that you want hidden in FPV should be placed in the negative U space region of the primary UV channel: (0, 0)(-1,1) This is useful for shoulder pads for example that would otherwise obstruct the FPV view.

## 2.8.3 Delta UVs and Clothing Masks

### Delta UVs

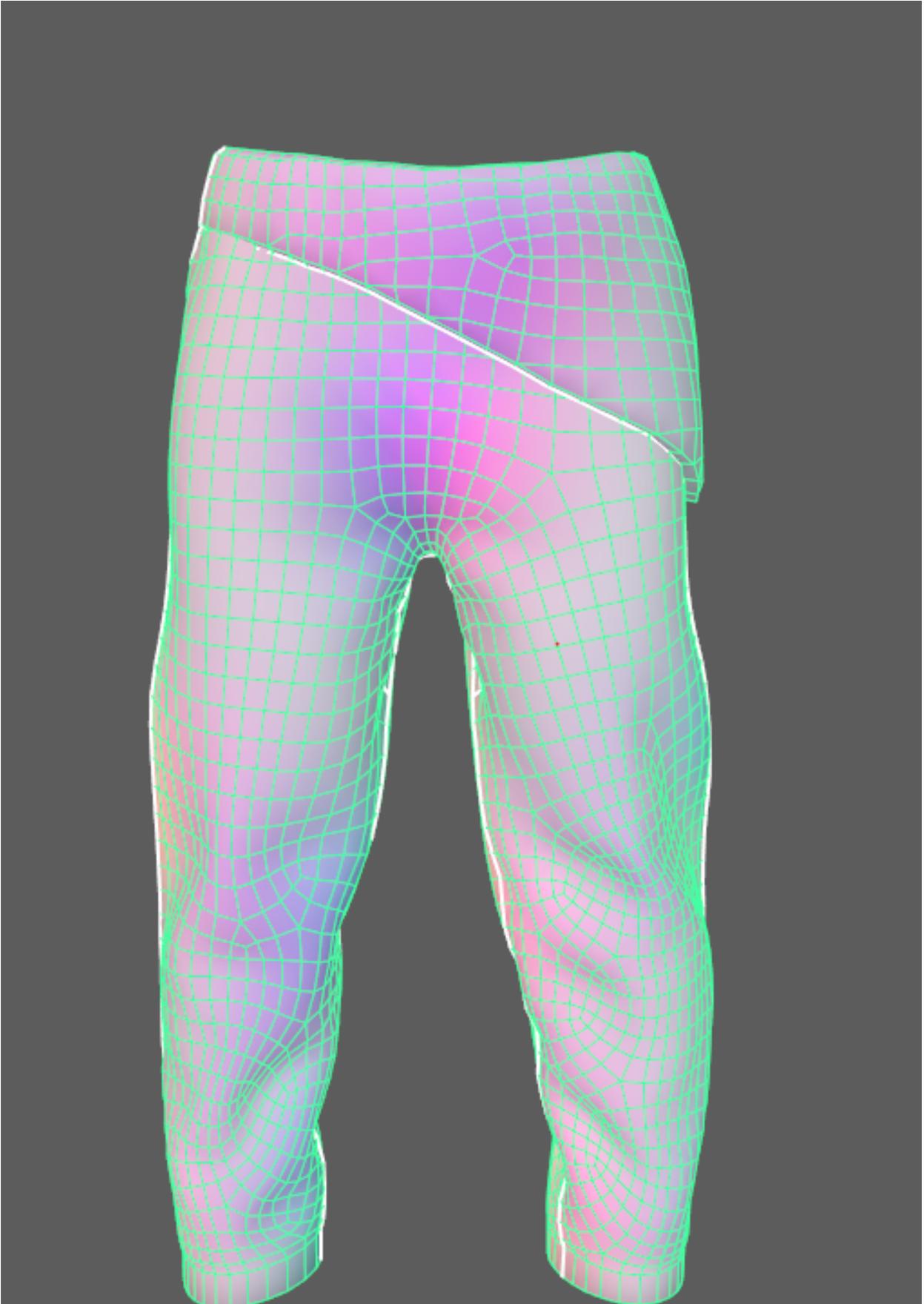
The Atlas clothing material uses Delta UVs to project delta textures from the base character body onto clothing so that clothing will react to the body fat slider. They are also used so that multiple articles of clothing can react the same way to the same masks.

Delta UVs are written to the second UV channel of a clothing asset by using the *Clothing Wizard* (Found in the AtlasDevKitContent tool repo)

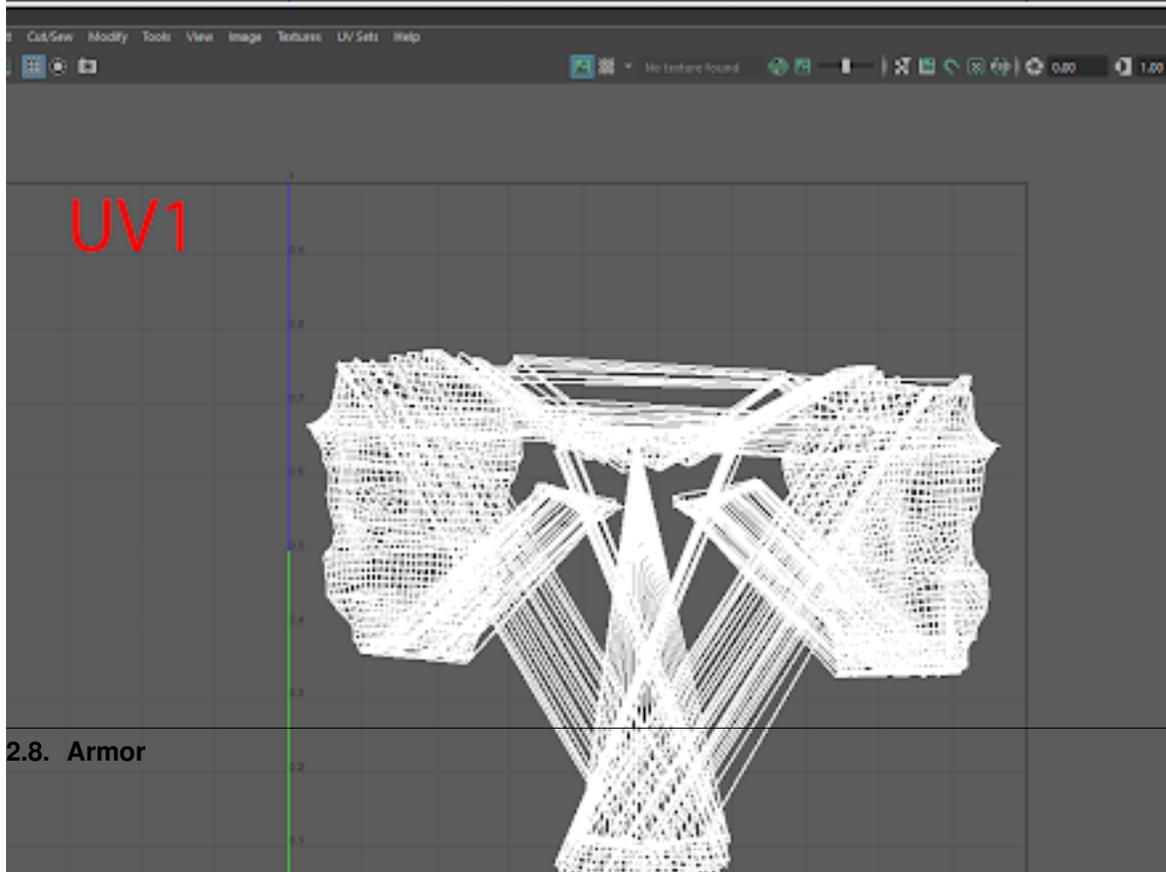
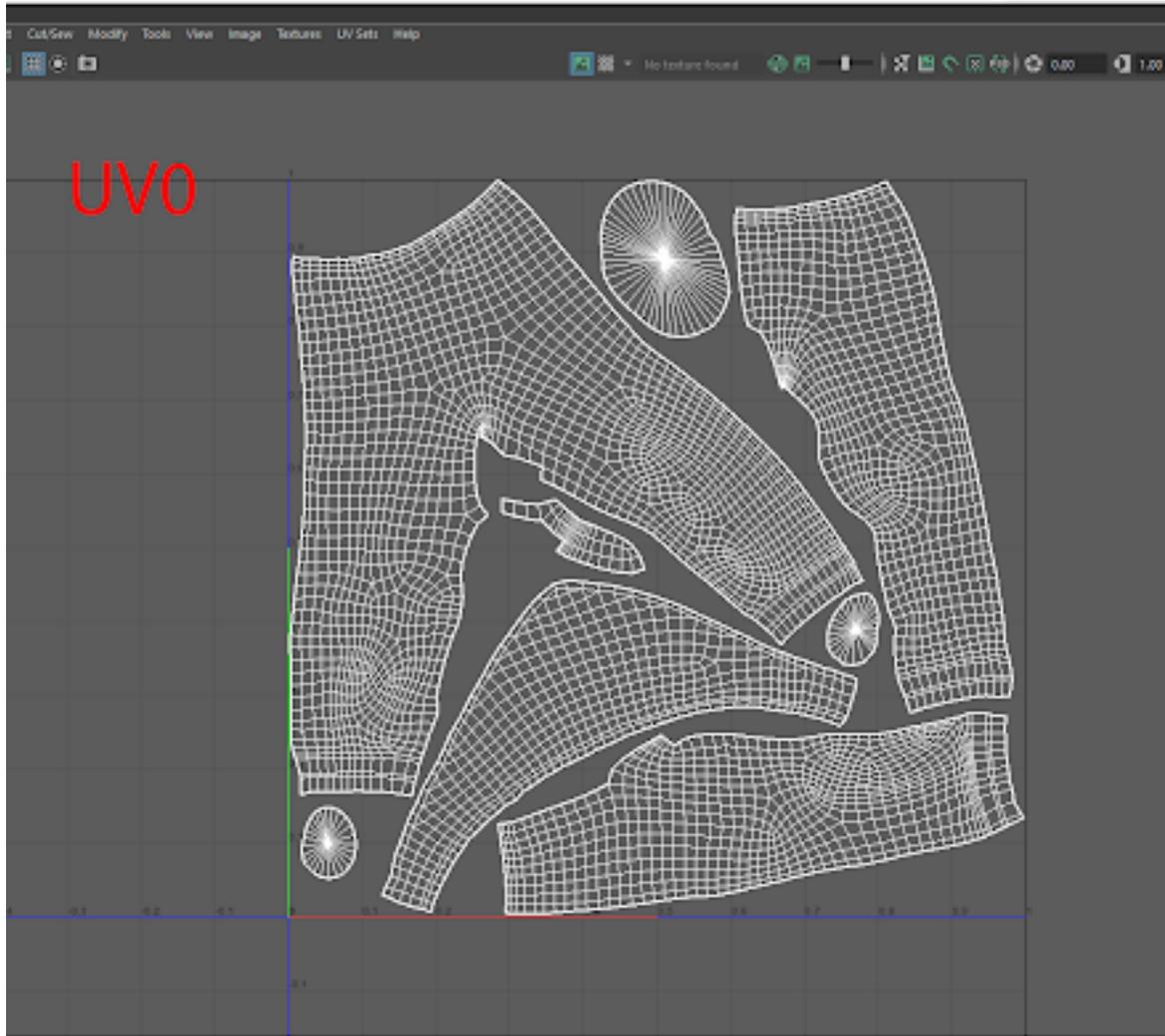
The clothing HDA will also write vertex colors and Body Distance UVs.

Any clothing asset will need these UVs and vertex colors to work properly with our shrink mask and body fat system.

A properly processed piece of clothing will display vertex colors if “Display Colors Attributes” is toggled on in Maya



UVs will look like this on a properly processed mesh:



## Clothing Masks

Clothing masks give us the ability to mask out the player skin where it intersects with the armor, shrink one piece of clothing into another, and mask out other armor meshes where they intersect with the armor. They also carry the information that tells what “layer” a piece of armor is on. The layer number lets the engine know the hierarchy of which armor pieces are going to mask which.

The shrink mask texture is a layered mask texture that is set up like this:

- R = Skin Mask (Human UV0) <- Human skin opacity
- G = Shrink Mask (UV1)
- B = Clothing Mask (UV1) <- Opacity

The clothing layer is displayed as a small gray square at the lower left corner of the texture

Currently we have 9 layers:

- layer 0 = boots, gloves small = 0% gray
- layer .1 = torso extra small = 1% gray
- layer 1 = pants small = 5% gray
- layer 2 = torso small = 10% gray
- layer 3 = gloves small = 15% gray
- layer 4 = pants big = 20% gray
- layer 5 = boots big = 25% gray
- layer 6 = torso big=30% gray
- layer 7 = gloves big = 35% gray
- layer 8 = hats = 40% gray

Masks should be named as “T\_HM(HF)\_ArmorSetName\_ArmorPiece\_M”

### 2.8.4 Clothing Wizard

Introduction - 01

Tool Installation into Houdini - 002

Loading your Clothing FBX - 003

Delta UV's and Mask Painting - 004

Exporting Clothing to FBX - 005

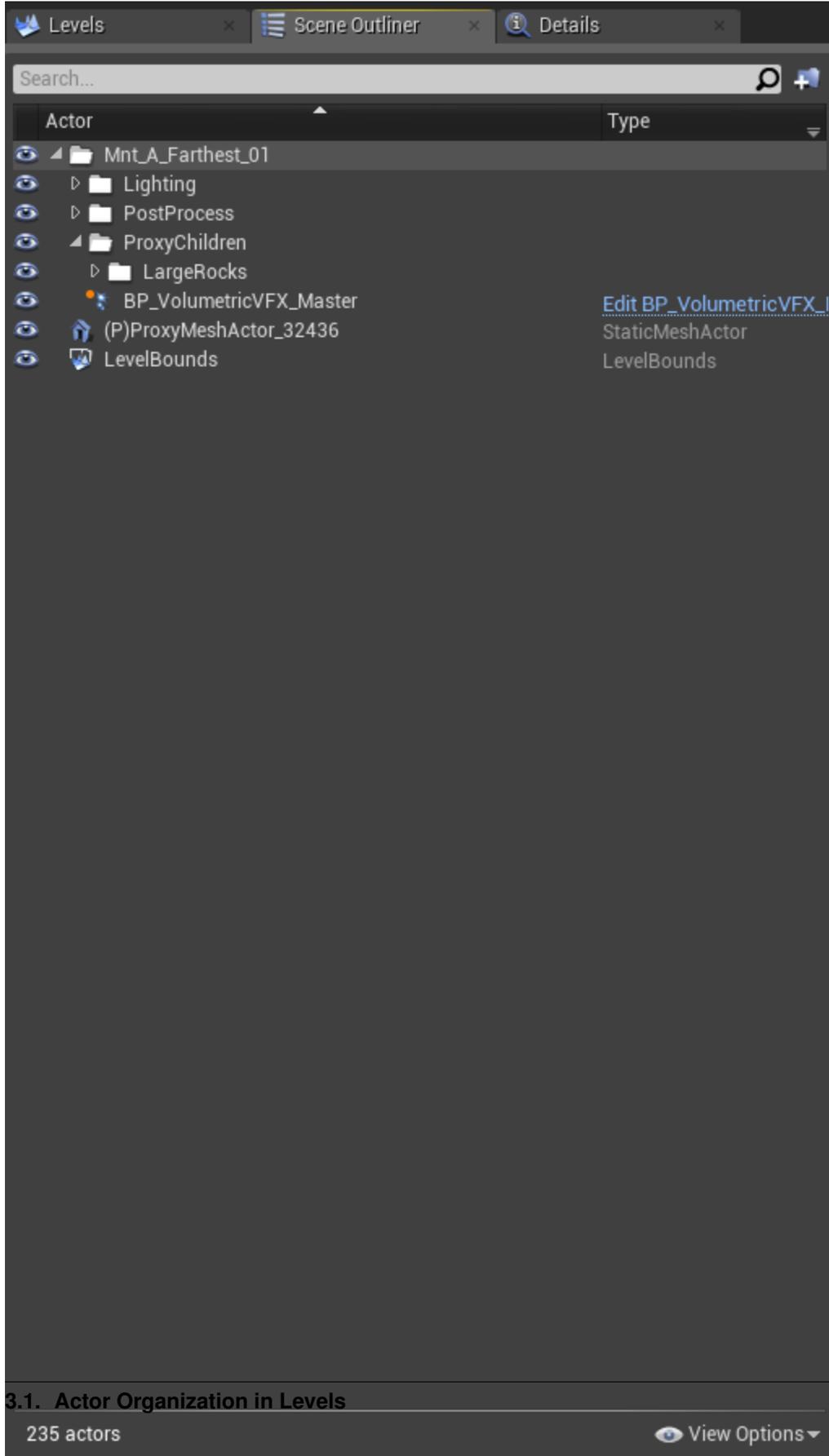
### **3.1 Actor Organization in Levels**

#### **3.1.1 Overview**

This page describes the general organization of assets in a level so things are standardized.



### 3.1.2 Folder Names



Levels should have a root folder that is named after the level. All other folders are inside this one. Below are some pretty standard folders you might have. Not all levels will have these folders. for example, Architecture will usually be in Far, and not Farthest. Exceptions will sometimes be made.

**[Level Name]: This is the root folder and will have the same name as the level. This will make finding assets easier if you h**

**[Lighting]:** Right now only Farthest and Far levels have this folder. All IBLs and any potential point lights should go in here. It's okay to have more folders inside here if you want to keep certain type of IBLs/lights together. **[PostProcess]:** PP volumes for a particular level. This includes the clutter BP volumes. **[ProxyChildren]:** Typically only farthest levels have this folder or any levels that will have a proxy mesh generated. Right now only the farthest levels have this folder. Assets in this folder will eventually be converted to proxy meshes. For more complicated levels this folder may be split up further into separate groups based on different quadrants of the level.

**[LargeRocks]:** Large deco rocks should go here that will eventually be part of a proxy mesh. **[Misc]:** Any assets that will be part of a proxy mesh that are not rocks (Sunken ships, giant statues, etc)

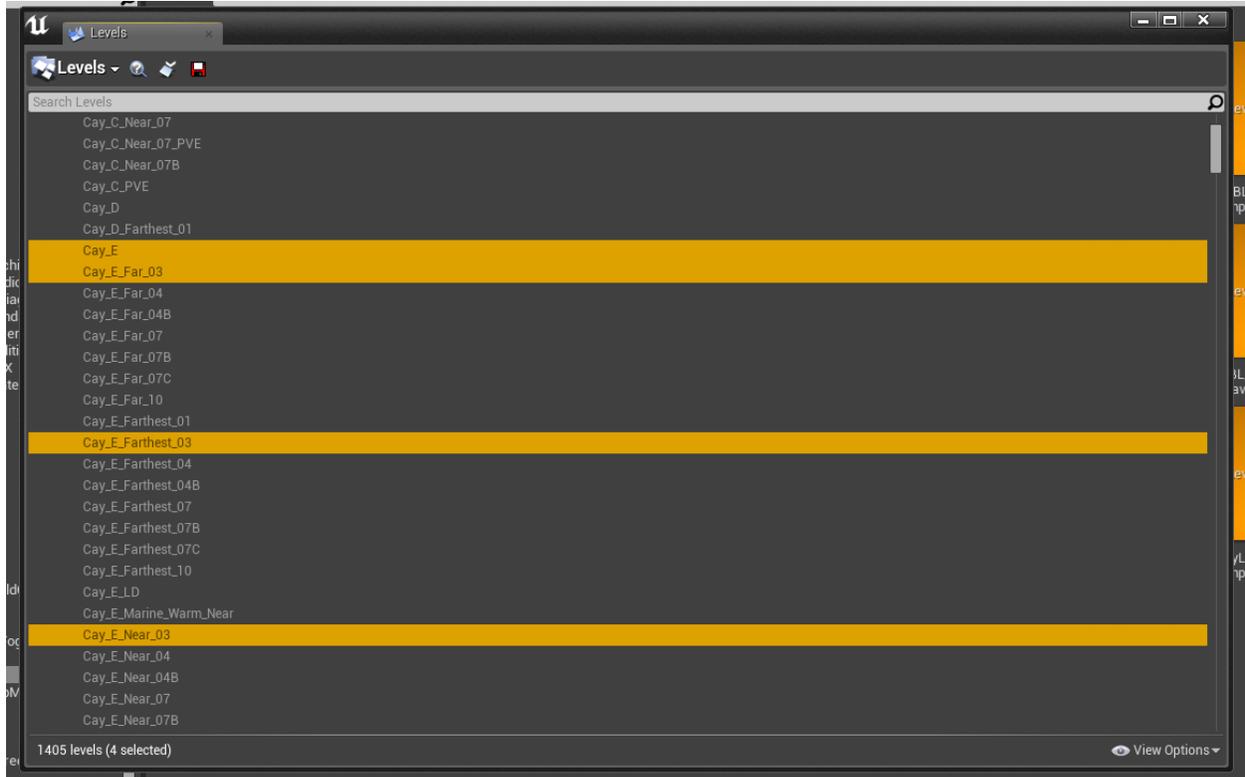
**[Architecture]:** Various architecture assets can go in here. It's okay to have various subfolders in here. **[Sounds]:** Any SFX should go in here. It's okay to have various subfolders in here. **[VFX]:** Any VFX should go in here. It's okay to have various subfolders in here. **[BP]:** Various blueprints can go here. It's okay to have various subfolders in here.

## 3.2 Island Icon Generation

The island icon generator gives us the ability to create map icons for the Atlas game. It takes in an existing island landscape and all static meshes associated to that island and procedurally generates an illustrative version of the island. The Tool itself is an HDA, meaning it was created for use with the Houdini Engine. The following steps outline the tools general usage and what each of the current settings does.

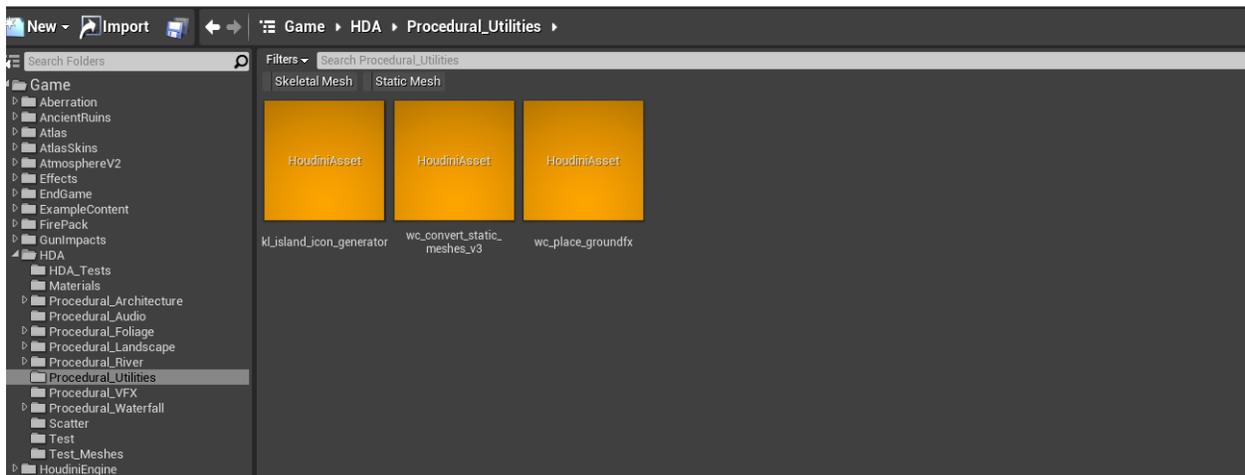
### 3.2.1 Step 1:

To start the icon generation process, load up an island that needs to have an icon generated. You can do this through the "Level" window in Unreal. Load the Island level and all of its near, far, and farthest sub levels so you get access to all the static meshes that are included in the island.



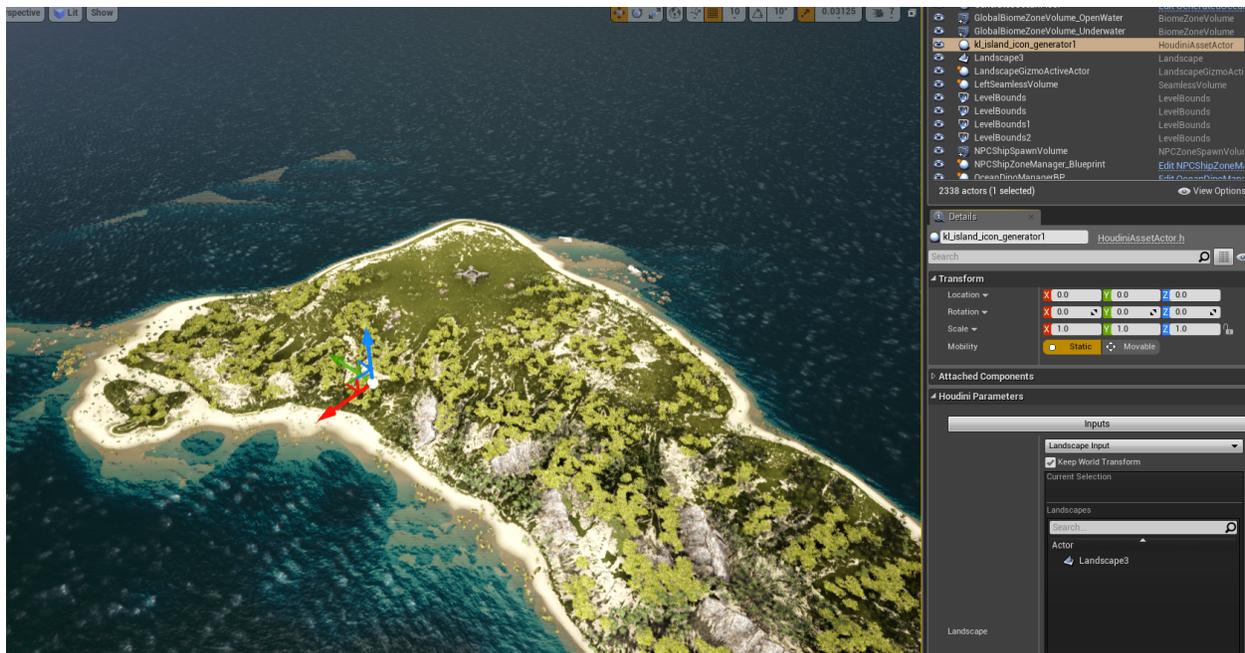
### 3.2.2 Step 2:

Once you have the Level and all its sub levels loaded, it is now time to add the “kl\_island\_icon\_generator” had to the scene. Navigate to the HDA folder located in the Atlas content folder. Then find the “Procedural Utilities” folder inside of that. There you will find the “kl\_island\_icon\_generator” had to assist you in generating the icon for the level.



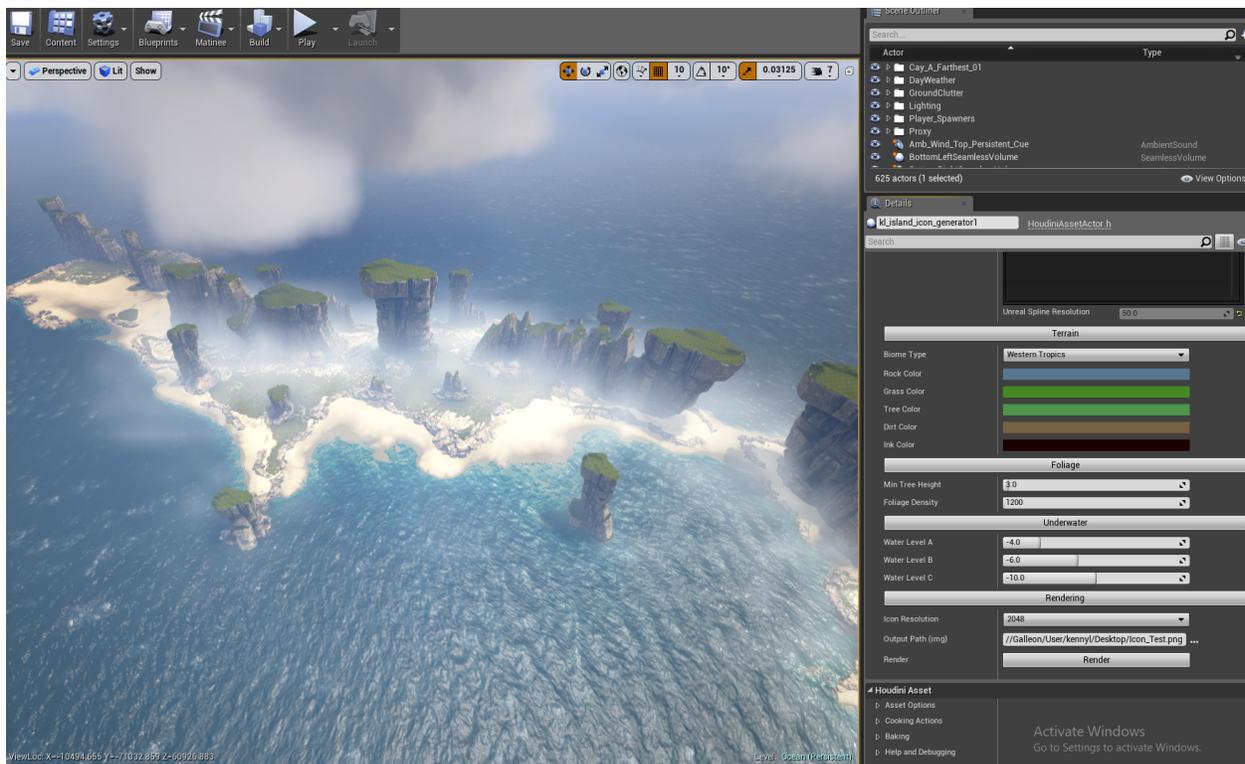
### 3.2.3 Step 3:

Once you have located the HDA, simply drag and drop it into the scene and make sure its location value are set to 0,0,0, for X,Y, and Z. This will instantiate the Had and get it ready for generating the Island Icon.



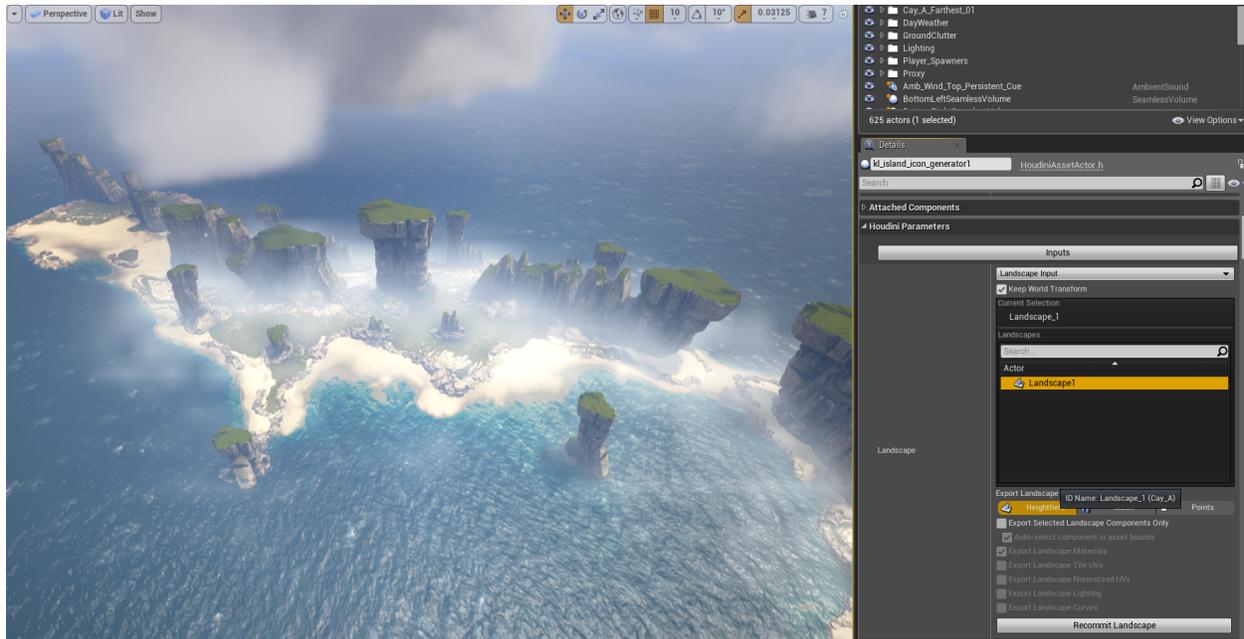
### 3.2.4 Step 4:

Before assigning the landscape or the static meshes to the HDA, we should set the main setting of the HDA, such as the icon size and the location of where we want to save the icon. Start by Setting the Icon size using the Icon Resolution dropdown, then by setting the Output Path to a location on your local machine (Can be a networked drive too).



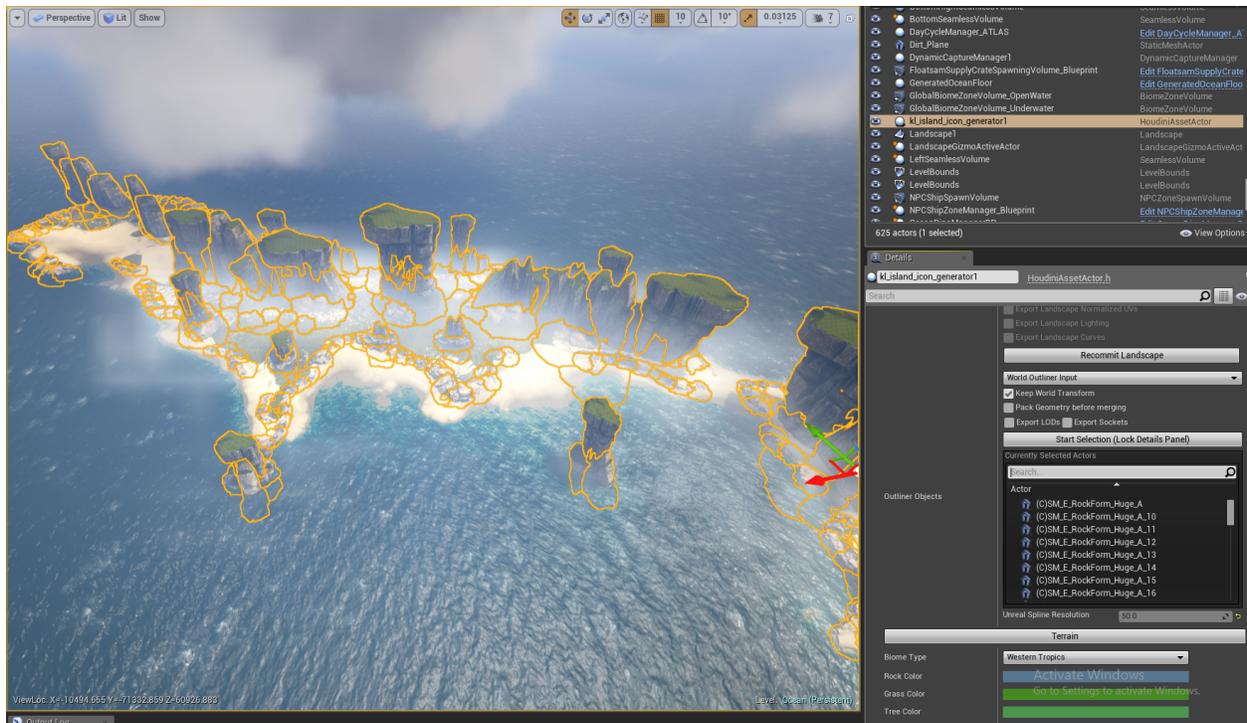
### 3.2.5 Step 5:

With the Icon resolution set and the Output Path set, it is now time to assign the Landscape or Island, and all the static meshes you would like included in the icon. This can be anything but its mostly meant to render out the Larger rocks of the island. Be careful not to add too many static meshes as this is an icon and we don't want the icon to be too busy looking when scaled down to small resolutions.



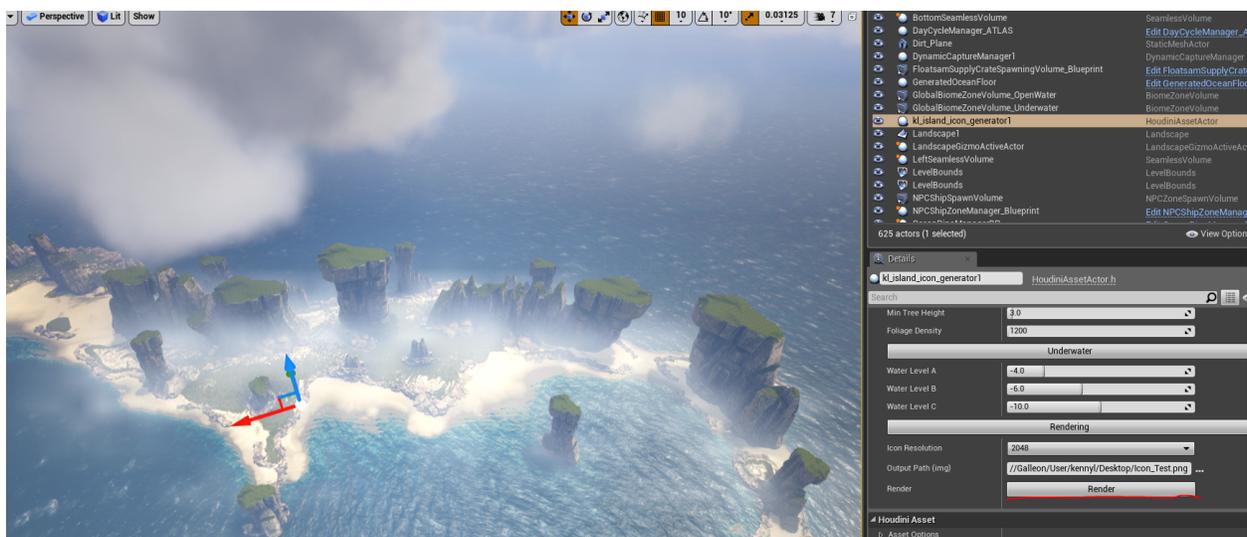
### 3.2.6 Step 6:

With the Landscape assigned to the HDA, it is time to select all the static meshes we want to be included in the final rendered icon. This is a bit of a weird workflow, but follow along and it'll get easier the more it is done. To start the process, it is recommended to find the Static meshes you want assigned in the Scene Outliner. Once you are ready hit the Start Selection Button once. This puts the HDA into assignment mode. Shift select all the static meshes you want assigned, then hit the "Start Selection" button again to commit the changes.



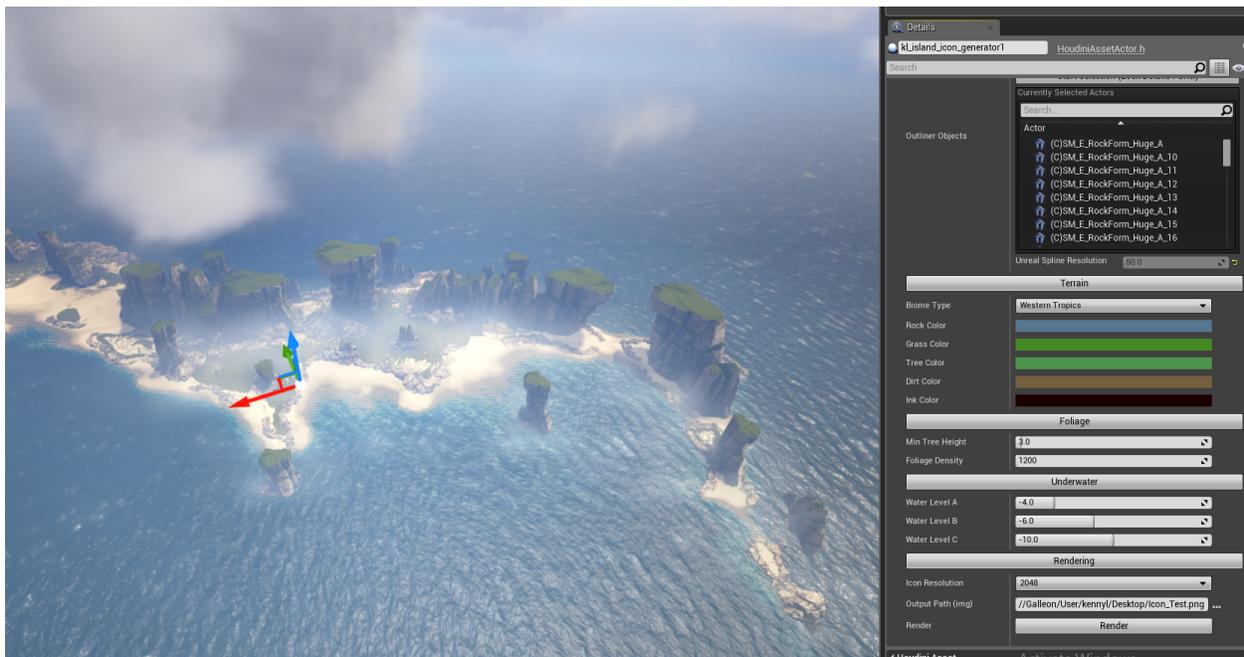
### 3.2.7 Step 7:

with all the preceding steps complete, we are now ready to render out the icon. It might take a few tries to get comfortable with the steps, but the tool itself allows us to control the overall look of the icon using the exposed settings provided. When ready hit the “Render” button at the bottom of the Tools Details panel, and let the icon render out. This can take anywhere from 3 to 5 minutes depending on the size of the island and the amount of static meshes assigned. So give it some time. The tool will display a “Finished” message when complete and you should see an icon at the location chosen for the output path. If you don’t see the icon right away, wait a few seconds and it should pop up.



### 3.2.8 Tool Settings:

- Biome Type: Lets you switch from different biome presets (Currently in progress)
- Rock Color: Allows us to change the global color tint to the rocks
- Grass Color: Allows us to change the global color tint to the grass areas
- Tree Color: Allows us to change the global color tint of the trees
- Dirt Color: Allows us to change the global color tint of the dirt
- Ink Color: Allows us to change the ink color of the icon
- Min Tree Height: Allows us to change at what height the trees start to draw into the icon. (This is in Meters from water level)
- Foliage Density: Allows us to change how dense the foliage is
- Water Level A: The depth at which the first shore area is drawn (In Meters)
- Water Level B: The depth at which the second shore area is drawn (In Meters)
- Water Level C: The depth at which the third shore area is drawn (In Meters)
- Icon Resolution: Allows us to change the resolution at which the icon renders out. (Recommended 2K)
- Output Path: Allows us to set the path the icon will save to.
- Render Button: The button that starts the rendering of the icon.



## 3.3 Ground Clutter

### 3.3.1 Overview

The Ground Clutter system in Atlas is a proprietary method for placing a large number of small static meshes on Landscape layers with minimum performance overhead. Meshes are placed procedurally in a fixed radius around the

player by a compute shader at runtime, using simple rules based on the associated IDs of painted Landscape layers.

### 3.3.2 Setup

Adding clutter requires the following steps:

- A landscape is required with terrain layer weights painted.
- Each layer of a landscape references a LayerInfo asset. This asset will need to have its Landscape Info ID be equal to an integer (int) that is positive and not 0. (0 defaults to no clutter)
- For each non-zero unique Landscape Info ID used you will need to add a Ground Clutter Layer Actor in persistent. This Ground Clutter Layer Actor will have a setting for Layer ID. This number should match the same number as the Landscape Info ID. This creates a 1:1 relationship between the Ground Clutter Layer Actor and the LayerInfo asset.
- A post process volume should be added to the farthest sub level of a particular island with it scaled to encompass the landscape. This post process volume will actually have your clutter settings. When you enter the volume it will push the clutter settings to the Ground Clutter Layer Actors and you will see clutter. It is highly recommended that you use a blueprint for the clutter post process volume. This will allow you to share clutter settings for a particular biome across multiple islands. You will also only need to make adjustments to a single blueprint when refining clutter settings instead of making changes to many different islands.
- Multiple landscape layers can share the same clutter settings by having the Landscape Info ID in the LayerInfo settings be identical.

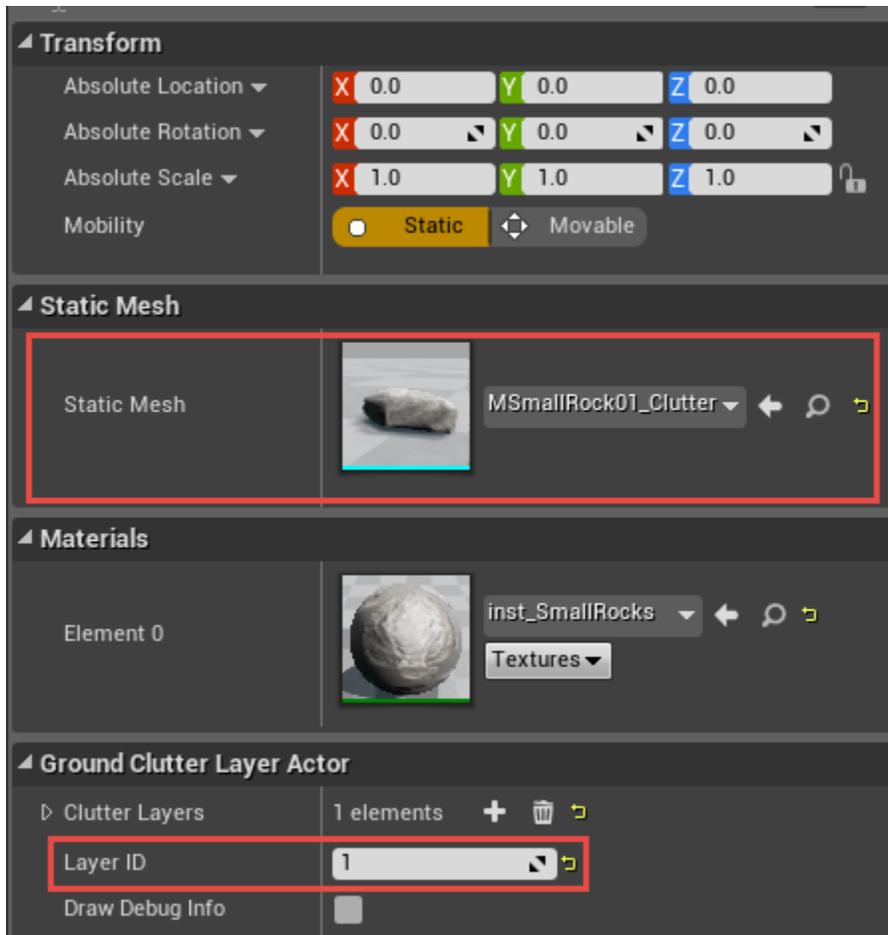
#### Ground Clutter Layer Group

The ground clutter layer actors that must be placed in persistent can be found by searching for it in the modes tab. Ground clutter settings are controlled by a post process volume that is placed in the farthest level for a particular island. The ground clutter layer actors in persistent will inherit their settings from this clutter post process volume. It is **highly** recommended that this post process volume be a blueprint. That way you can have multiple islands that will share the same clutter blueprint for a particular biome. If you need to make changes you will only need to do it once. This will also allow you to make children of the blueprint that are slightly different but inherit most of their settings from the global blueprint.



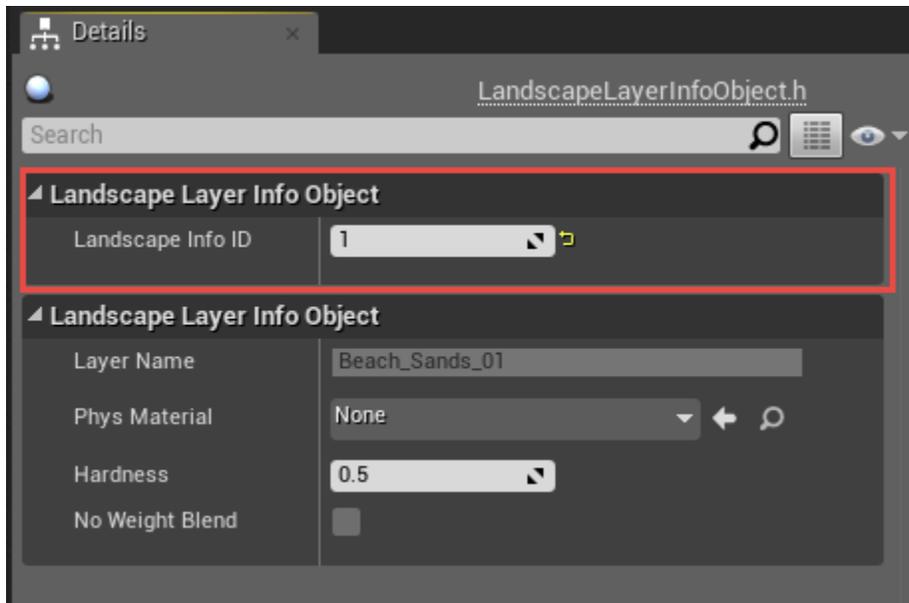
In the properties:

- Make sure each actor's transform is set to the world origin (0,0,0)
- Add a Static Mesh (this will not be used, but is needed for other reasons - just use your first clutter mesh)
- Set the Layer ID for the clutter actor. This should match the int used for the LayerInfo asset where you want the clutter to appear.



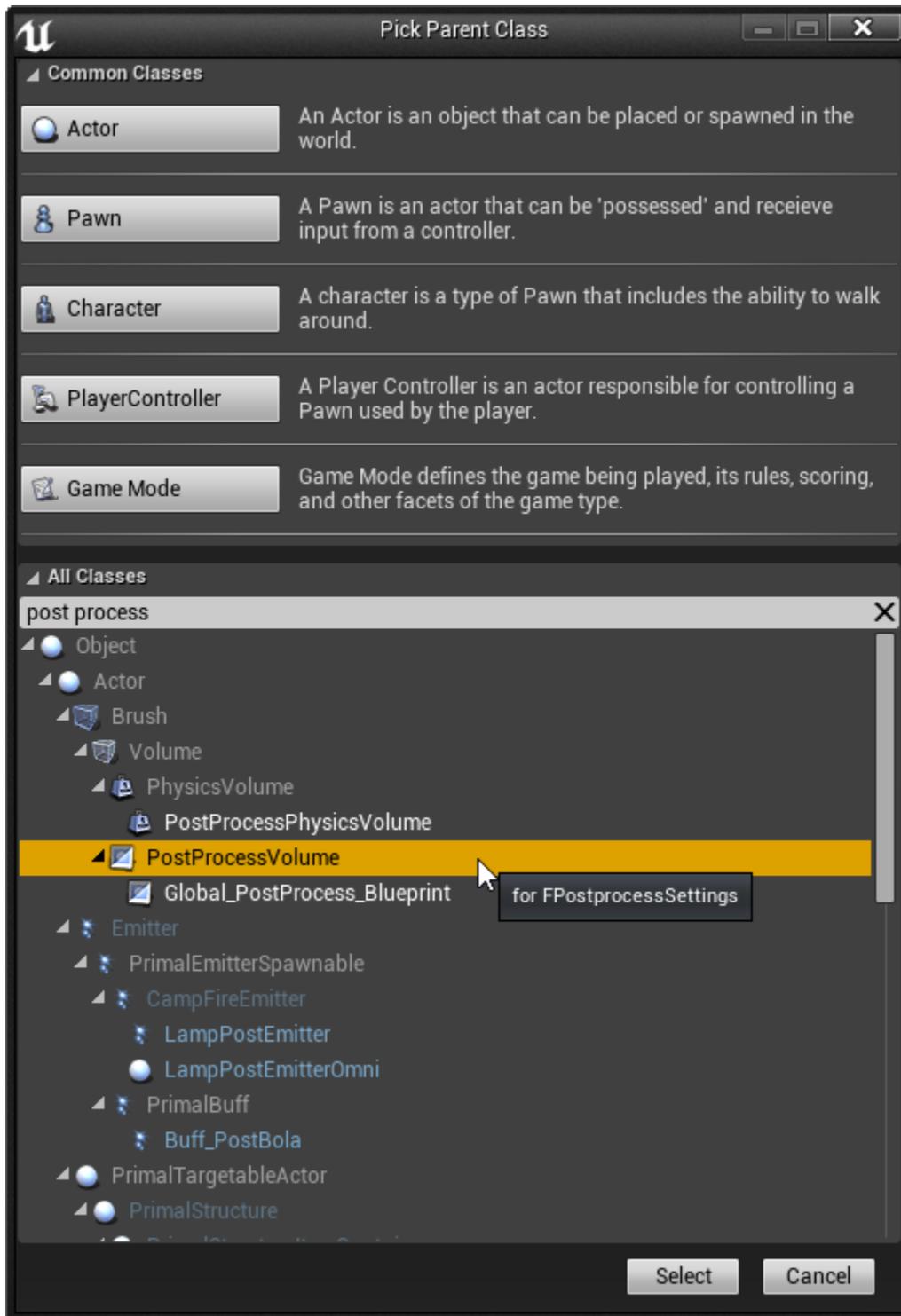
### LayerInfo Assets

For each Landscape Layer Info, set the corresponding Landscape Info ID to match the desired Ground Clutter Layer Actor's Layer ID.



### Post Process Blueprint

It's best to create a global post process clutter blueprint for each particular biome. All islands of that biome can then use that blueprint for clutter.



### Tweaking

Landscape Info ID is cached on load, **so you must reload your map after setting layer IDs**. You should now see some beautiful clutter all over your painted landscape layers! All properties can now be adjusted for each Ground Clutter Layer Actor in real-time.

## Performance

In general they should be **one draw call** and **less than 200 triangles**. Exceptions can be made for lower densities where higher triangle counts can be used on a case-by-case basis. All clutter assets should only have two lods. You do not need to adjust the screen size of the 2nd lod as it is handled in code.

DENSITY	TRIANGLE LIMIT
1	130
0.9	137
0.8	145
0.7	155
0.6	168
0.5	184
0.4	206
0.3	237
0.2	291
0.1	411
0.05	581
0.025	822
0.01	1300

## 3.4 Volumetric Fog

### 3.4.1 Overview

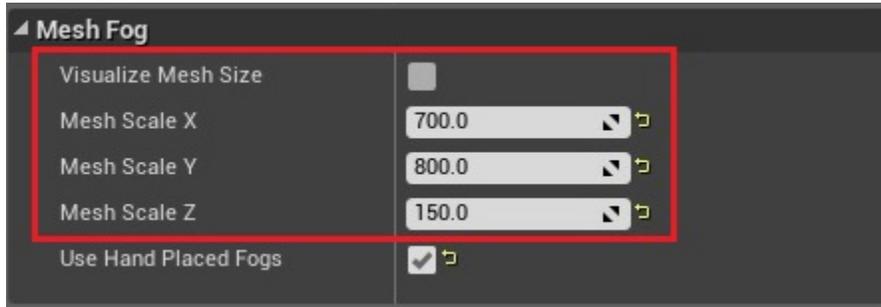
The Volumetric Fog Blueprints allow designers to not only get a nice ground fog that fits the shoreline, but also provides with the ability to place up to 16 hand placed fog spheres per island. The blueprint also gives us density and time of day options to allow designers/artists the ability to adjust fog densities based on time of day. The hand placed fog points give us ultimate control over which areas receive fog and which don't. Generally the fog pass should come after the IBL pass as IBLs do change the overall look of the fog.

### 3.4.2 Setup

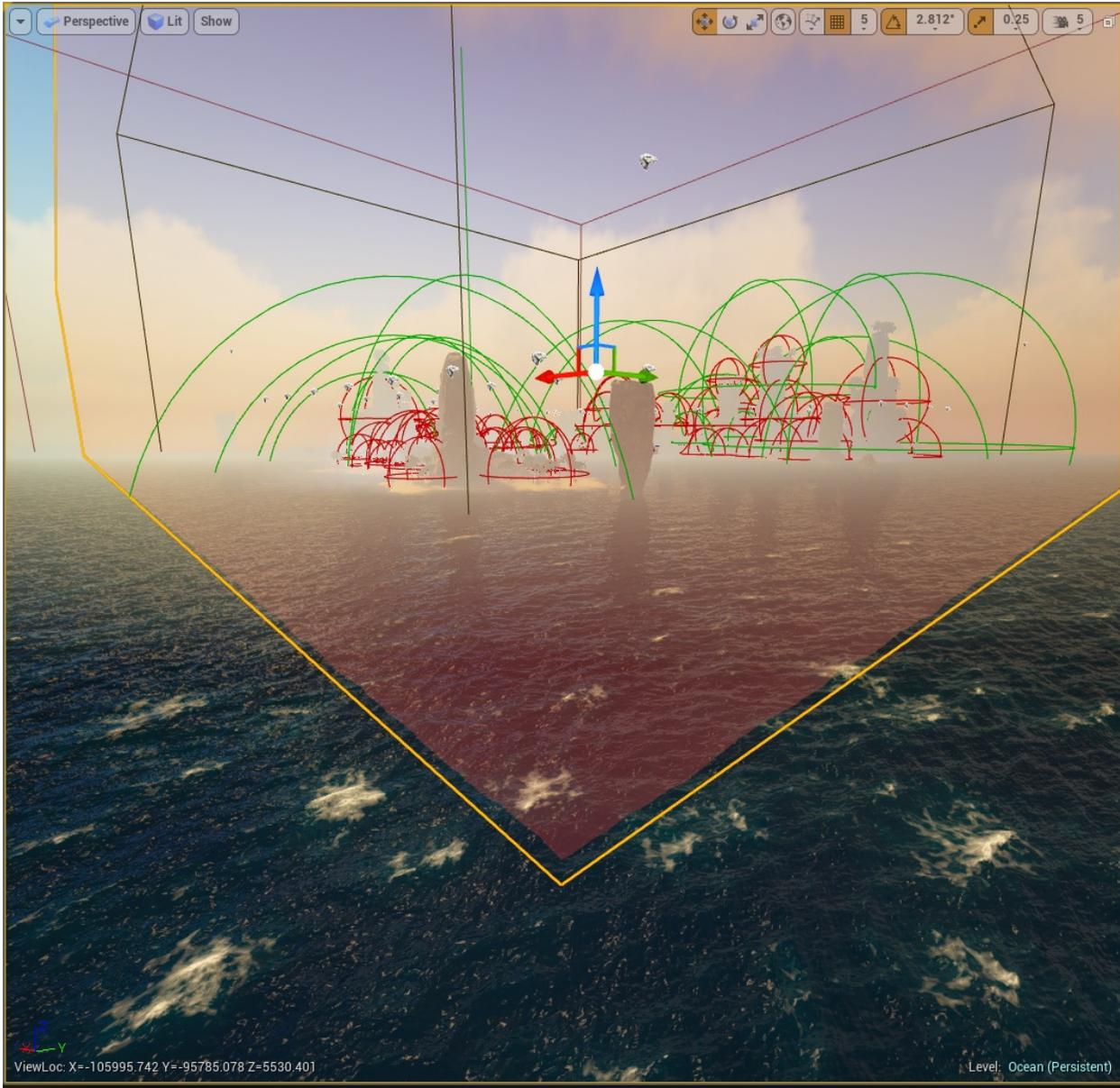
- To setup Volumetric fog on a new island you will need to place 3 unique blueprints in your level (They must all remain in the farthest sublevel for each island)
  - Blueprint '/Game/Atlas/Effects/Volumetric/BP\_FogBlutility.BP\_FogBlutility'
  - Blueprint '/Game/Atlas/Effects/Volumetric/BP\_VolumetricFog\_Master.BP\_VolumetricFog\_Master'
  - Blueprint '/Game/Atlas/Effects/Volumetric/BP\_FogPoints.BP\_FogPoints'
- After placing these three actors select the "BP\_FogBlutility" actor you placed and be sure to set its "Fog Reference" option to match the "BP\_VolumetricFog\_Master" actor that was placed.



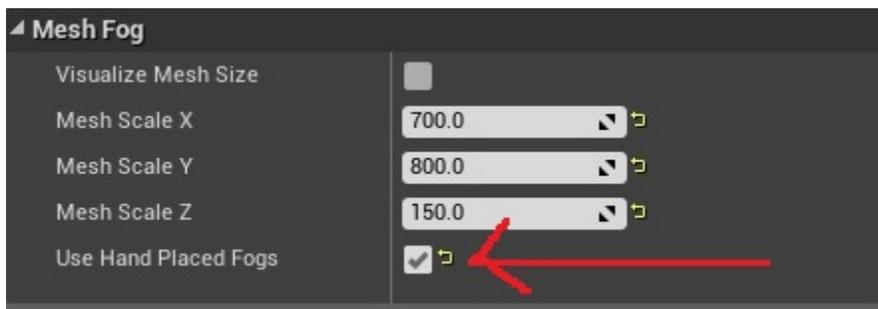
3. Select your BP\_Volumetric\_Master” actor and move the blueprint to the center of the island. This doesn’t have to be exact but on the next step when we are scaling these it works best if this is centered to the island.
4. Select the “BP\_VolumetricFog\_Master” actor and look for the Mesh Scale X,Y,Z options. You’ll want to set the XYZ scales to ensure the mesh is fully covering the entire island with plenty of room to breathe. Fog will only render within this mesh so ensure you have full coverage in all directions or fog will get a very unnatural cutoff at the edges, this includes both ground and hand placed fog.



Generally the volumetric fog mesh should be just below the ocean level.

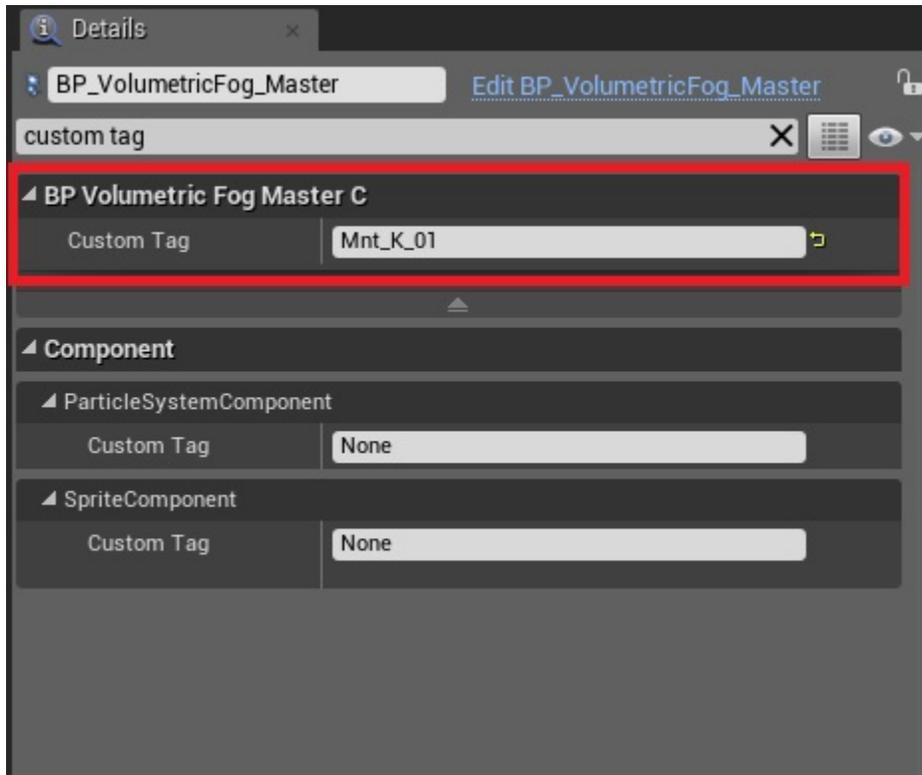


5. Select the “BP\_VolumetricFog\_Master” actor and enable an option called “Use Hand Placed Fogs” This option and the next few steps can be skipped if the map is a very small one such as an islet that we normally don’t want/need individual fog points placed on.



6. With the “BP\_VolumetricFog\_Master” actor still selected do a search for “Custom Tag” There are multiple

Custom Tag options on this actor so be sure to get the one under “BP Volumetric Fog Master C”. This “Custom Tag” Must have a unique tag that does not exist on any other island. This basically ensures that the system is only getting fog point actors from this island and only this island. Generally we have been setting the overall name of the island as the tag for these to ensure they are unique.



7. Repeat the above step and do the same thing for the “BP\_FogPoints” actors. The Custom tags on these actors MUST match the custom tags used on the “BP\_VolumetricFog\_Master” actor exactly or they will not work. Upon setting the custom tag for this actor if you have done everything correctly you should see fog appear at the location of this actor, if not be sure to check the above steps to make sure it is setup properly. You can place up to 16 fog point actors per island. Generally it is better to use fog points with a very large radius and just embed them in the ground than it is to place lots of small radius fog points. Every 4th fog point does increase the cost a very small amount, generally only the largest maps will ever need 16 fog point actors.



### 3.4.3 Settings

## BP\_VolumetricFog\_Master:

### Default

Ground Fog Density	<input type="text" value="0.05"/>
Ground Fog Density Day	<input type="text" value="0.1"/>
Ground Fog Density Night	<input type="text" value="0.4"/>
Hand Placed Fog Density	<input type="text" value="1.0"/>
Hand Placed Fog Density Day	<input type="text" value="1.25"/>
Hand Placed Fog Density Night	<input type="text" value="1.0"/>
Fog Material	<div style="display: flex; align-items: center;"> <div style="display: flex; align-items: center;"> <span style="border: 1px solid #ccc; padding: 2px 5px; font-size: 0.8em;">MaterialInstance</span> <span style="margin: 0 5px;">←</span> <span style="margin: 0 5px;">🔍</span> <span style="margin: 0 5px;">🔗</span> </div> </div>
Overall Noise Scale	<input type="text" value="2.0"/>
Height Falloff Power	<input type="text" value="5.0"/>
Lower Fog Layer Height	<input type="text" value="6000.0"/>
BP Reference Array	0 elements <span style="margin: 0 5px;">+</span> <span style="margin: 0 5px;">🗑️</span>
Update Fog Points	<input type="checkbox"/>
Extinction Near Percent Day	<input type="text" value="1.0"/>
Extinction Near Percent Night	<input type="text" value="1.0"/>

### Mesh Fog

Visualize Mesh Size	<input type="checkbox"/>
Mesh Scale X	<input type="text" value="700.0"/>
Mesh Scale Y	<input type="text" value="800.0"/>
Mesh Scale Z	<input type="text" value="150.0"/>
Use Hand Placed Fogs	<input checked="" type="checkbox"/>

1. **Ground Fog Density:** Sets the lower ground fogs overall density (Works like a global value and will adjust both day/night values)

2. **Ground Fog Density Day:** Sets the lower ground fogs density during the day (Late evening and early morning transition between day/night values)
3. **Ground Fog Density Night:** Sets the lower ground fogs density during the night (Late evening and early morning transition between day/night values)
4. **Hand Placed Fog Density:** Sets the hand placed fog points overall density (Works like a global value and will adjust both day/night values)
5. **Hand Placed Fog Density Day:** Sets the hand placed fog points Day Time Fog Density. (Late evening and early morning transition between day/night values)
6. **Hand Placed Fog Density Night:** Sets the hand placed fog points Night Time Fog Density. (Late evening and early morning transition between day/night values)
7. **Fog Material:** *NO TOUCHY!* Simply a reference to the dynamic material instance that gets dynamically created, you can't change/update this.
8. **Overall Noise Scale:** Adjusts the overall scale of the noise function that is being used for the fog. Probably don't want to touch this value to keep things consistent between maps.
9. **Height Falloff Power:** Normally not used by designers but it can allow us to setup a bit of a softer gradientZ transition for the ground fog. Setting a higher value will basically push the max ground fog height down a bit with a softer transition.
10. **Lower Fog Layer Height:** Default value will be good enough for most maps but this allows us to set the actual max height the ground fog will come up to.
11. **BP Reference Array:** *NO TOUCHY!* This is only visible as an easy way for designers to see how many fog points are currently in the level.
12. **Update Fog Points:** Will update all of the fog point locations/values. You don't need to use this if you have the Blutility actor hooked up properly.
13. **Extinction Near Percent Day:** Controls the near camera fade on the fog during the day. To keep things consistent between maps we normally don't want to touch this setting.
14. **Extinction Near Percent Night:** Same as the above but controls the near camera fade at night.
15. **Visualize Mesh Size:** Toggles visibility for a mesh to better visualize the total area the fog will be covering. Should always disable this option when you are done using it! It won't break anything or show up in game but we don't need lots of levels with giant red boxes enabled all the time.
16. **Mesh Scale X:** Sets the X Scale of the mesh used for Volumetric Fog.
17. **Mesh Scale Y:** Sets the Y Scale of the mesh used for Volumetric Fog.
18. **Mesh Scale Z:** Sets the Z Scale of the mesh used for Volumetric Fog.
19. **Use Hand Placed Fogs:** This needs to be enabled for most of our islands as you can't place the specific fog points without this enabled. For small cay islands or islands where we do not want the specific points placed keep this disabled as it will be a little cheaper without.

### BP\_FogBlutility:



The Fog Blutility actor is pretty straight forward, just need to place it and set the below setting. It does not matter where this actor is located in the world as long as it is in the farthest sublevel with the other fog actors.

- **Fog Reference:** Connect this option to your Volumetric fog master actor to ensure it updates in real-time within editor.

### BP\_FogPoints

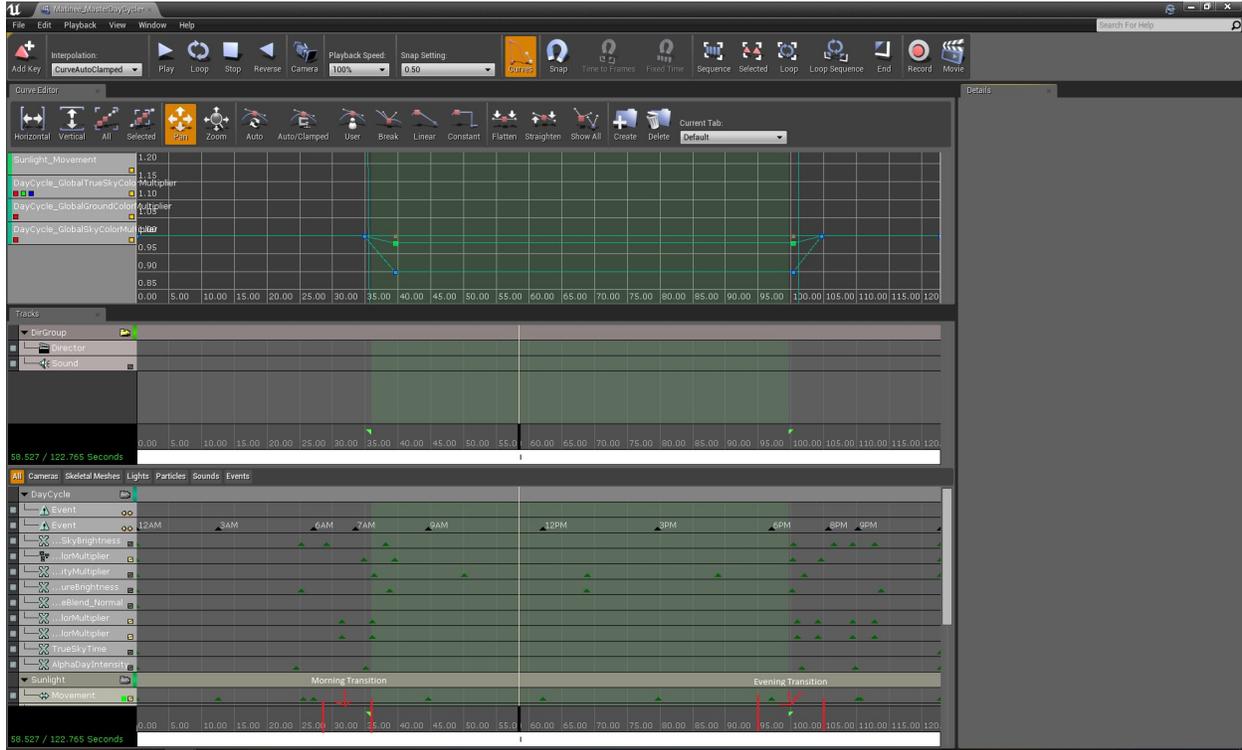


- **Fog Radius:** Total radius that you want this fog point to render fog. Will be visualized in level by a green sphere.
- **Fog Density:** Overall density value used by this fog point. Will be used on top of the day/night values.
- **Day Fog Density:** Adjusts the fog density of this actor only during day time hours. Day/Night values blend between each other during early morning/late evening hours.
- **Night Fog Density:** Adjusts the fog density of this actor only during night time hours. Day/Night values blend between each other during early morning/late evening hours.

### Performance

Every 4th BP\_FogPoints actor you place increases the cost of the system. So if you are using a number not divisible by 4 you probably have a few more points you can use. For instance, if you are using 6 fogpoints you can use 2 more free of charge without increasing the cost. While every 4th fog point actor does increase the cost slightly the fog system has an overall flat cost outside of that. Increasing the scale of these or overlapping them will not increase the cost.

### 3.4.4 Final Notes



When finalizing the fog passes for each level it is always a good practice to check that the values seem good at various times of day (Especially during the transitional times) as they do dynamically update to be different from day to night. In the `Matinee_MasterDayCycle` within the persistent map the night to day transition for fog starts at 28 and will be fully considered day at 36. The Day to night transition starts at 95 and will be considered fully night at 105.

## 3.5 Shorelines

### 3.5.1 Overview

Below is the general overview on generating shoreline data for landscapes. All island footprints should have one generated so that waves near shore are not huge with high wind.

### 3.5.2 Atlas Shorelines - How To Setup

Sublevel need to have their shoreline maps created via a console command. This can be done in the editor. Shoreline data is then saved into the level itself. Shorelines generate 2 different sets of data: height field and distance field. Additional console commands exist to clear out this data, or export into PNGs to aid in verification and debugging scenarios.

#### Creating Shoreline Data

To create shoreline data, load the sublevel and make sure it's the only sublevel visible. Then, using the console, use this command:

---

**BuildShorelineMaps Level=X**, where X is the name of the level (ex: “Isle\_A\_Mnt”)

---

**Note:** Typically we use default settings, but additional parameters can be passed in to modify how the shorelines are created. These are optional and not required, but could be useful for testing and experimentation.

---

- **ShorelineDepth=N**, where N is the Z value where the shoreline is logically at. The default value is 0.
- **MaxDepthForWaveDampening=N**, where N is a positive number that describes the maximum depth where wave dampening takes place. Default value is 3000.
- **MaxDistanceToShore=N**, where N is a positive number that describes maximum distance from shore which impact the distance field created for the shorelines. Default value is 2000.
- **MaxWorldDimension=N**, where N is the maximum size of the world in unreal units. Default value is 600000.
- **MaxTextureDimension=N**, where N is the maximum texture size used by shoreline maps. Default value is 4096.

### Removing Shoreline Data

To remove shoreline data from a sublevel, enter the following on the console:

**RemoveShorelineMaps Level=X**, where X is the name of the level (ex: “Isle\_A\_Mnt”)

### Viewing Shoreline Data

To view the data in PNG format, entering the following on the console:

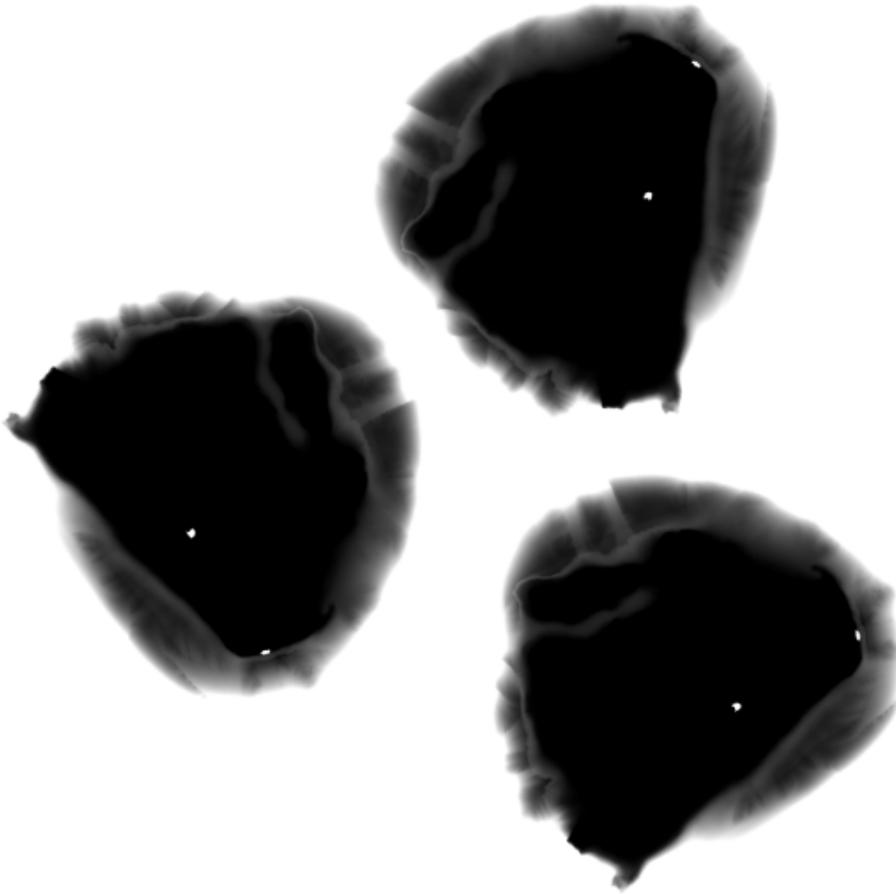
**ExportShorelineMaps {Level=X}**, where X is the name of the level (ex: “Isle\_A\_Mnt”). If you don’t specify a level, all sublevels with shoreline data will be exported to PNG files. A dialog to enter the save location will be displayed so you can save them anywhere on your computer.

You can also export at anytime the world’s composite shoreline maps using this command:

**ExportWorldShorelineMaps {To=PATH}**, where PATH is a location on your PC. This argument is optional, and when not specified the composite world shoreline maps are saved to the games save directory.

### Shoreline Maps

The height map is really more like an inverse height map: black is at/above sea level, and white is deepest ocean depth. Here is an example of the world’s composite height map data:



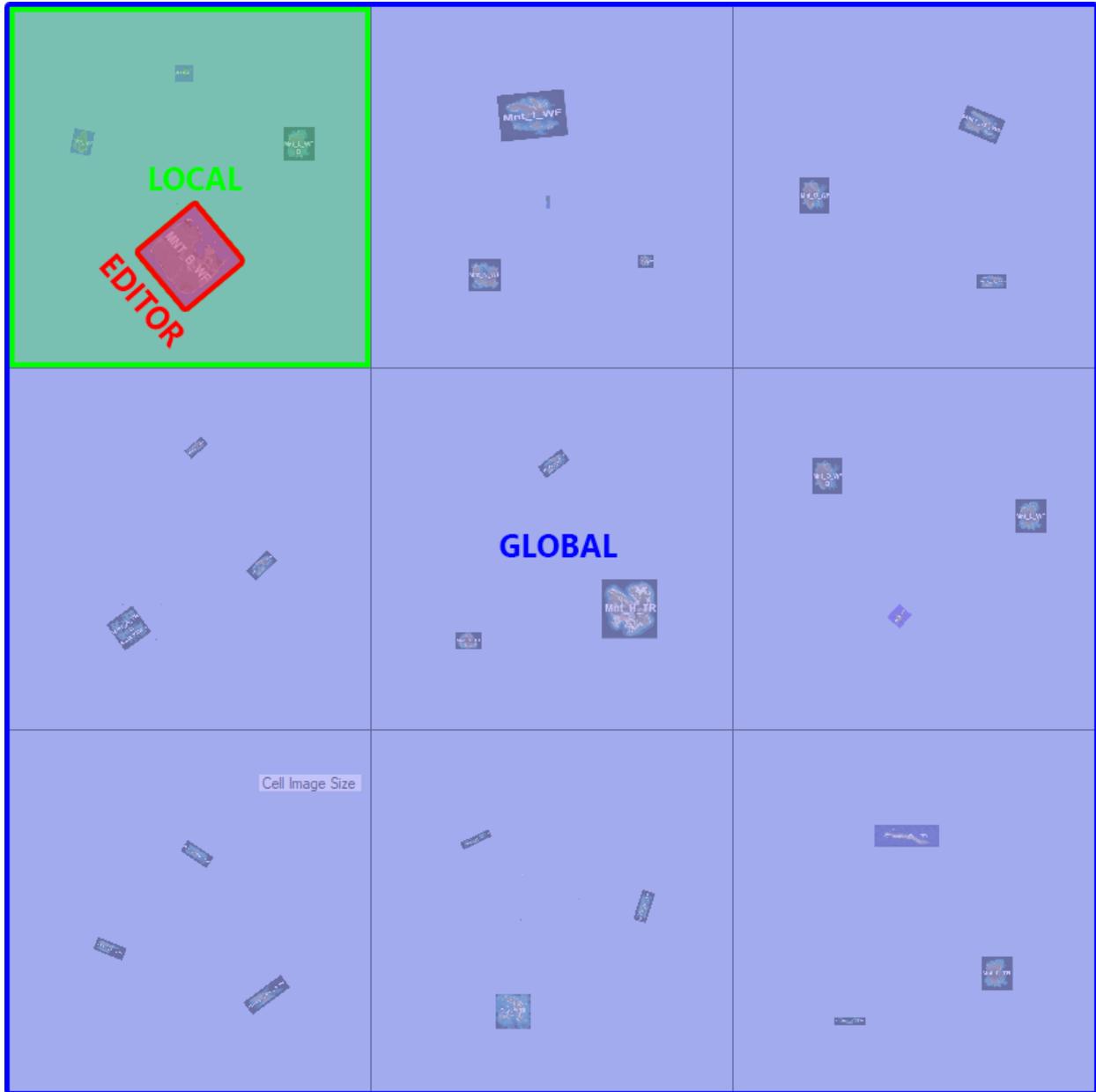
The distance field is black at points farthest from the shoreline, and white at/inside the shoreline. Here is an example:



## 3.6 Coordinates

### 3.6.1 Overview

There are several coordinate systems in ATLAS. This document covers them and how to work with them.



## Editor

These are the coordinates that you are used to from ARK. While constructing a level in the editor, or previewing a single level in PIE (without the server grid), the transform of any objects and the camera location are in this coordinate space. It has an absolute root of 0,0,0 at the center of the world.

## Local

On an ATLAS server, grouped sub-levels are transformed together as part of an “Island Instance”. These islands can be arbitrary translated and rotated, and multiple copies can be made. The resulting coordinates are transformed into the server’s local coordinate space, which has a “local” root of 0,0,0 at the center of the server world. Islands and their coordinates are offset away from the origin, and the EDITOR coordinate values no longer make sense. EDITOR

coordinates such as the location of points-of-interest and other objects must be transformed into a server's LOCAL coordinates in order to work with game commands such as `setplayerpos X Y Z`.

## Global

An ATLAS server grid has an even larger coordinate system which is GLOBAL. Every position in the world can be specified as an absolute and unique coordinate value across the entire grid of servers (however large it may be). The origin is rooted at the top left of the server grid with a value of 0,0,0 and wraps around the edge of the map, meaning there are no negative values. GLOBAL coordinates are not rotated differently from LOCAL values, but a different translation on X and Y axes must be applied in order to resolve them. `setplayerpos` does not currently work with GLOBAL coordinates.

### 3.6.2 Transforming Coordinates

#### Editor to Local

At the time of this writing, this is the most useful transform because there is not a command to instantly travel to GLOBAL coordinates. In order to transform a set of coordinates from EDITOR to LOCAL, the following information must be known from the server grid .JSON file: `additionalTranslationX` `additionalTranslationY` `additionalRotationYaw`. For any given Island Instance, these values can be found in any of the sub-levels used by that instance. To find them, look for the named Island Instances in the server grid file at the bottom of all the sub-levels for each grid. Note that the same Island Instance can be used multiple times and in multiple server grids, so you should use the "id" value of the named Island Instance in order to find its sub-levels, above.

#### Island Instance Definition:



```
"gridX": 2,  
"gridY": 1,  
"ip": "34.212.28.191",  
"name": "",  
"port": 56555,  
"gamePort": 5655,  
"seamlessDataPort": 27005,  
"isHomeServer": false,  
"floorZDist": 0,  
"utcOffset": 0,  
"transitionMinZ": 0,  
"OceanDinoDepthEntriesOverride": "",  
"oceanFloatsamCratesOverride": "",  
"treasureMapLootTablesOverride": "",  
"sublevels": [  
  {  
    "name": "Mnt_C",  
    "additionalTranslationX": -54501.5,  
    "additionalTranslationY": 261384.25,  
    "additionalTranslationZ": 0.0,  
    "additionalRotationPitch": 0.0,  
    "additionalRotationYaw": 131.319382,  
    "additionalRotationRoll": 0.0,  
    "id": 32,  
    "landscapeMaterialOverride": -1  
  },  
  {  
    "name": "Mnt_C_Far_01",  
    "additionalTranslationX": -54501.5,  
    "additionalTranslationY": 261384.25,
```

## 3.7 Procedural Foliage Scattering

### 3.7.1 Overview



The procedural scattering tools allow environment artists to rapidly lay out large areas of Foliage for any given landscape in Unreal. The tools are a set of Houdini Digital Assets, so the Houdini Engine is required, within Unreal, for this pipeline to work. The following outlines the proper usage of the HDA's and the steps needed to get the system up and running.

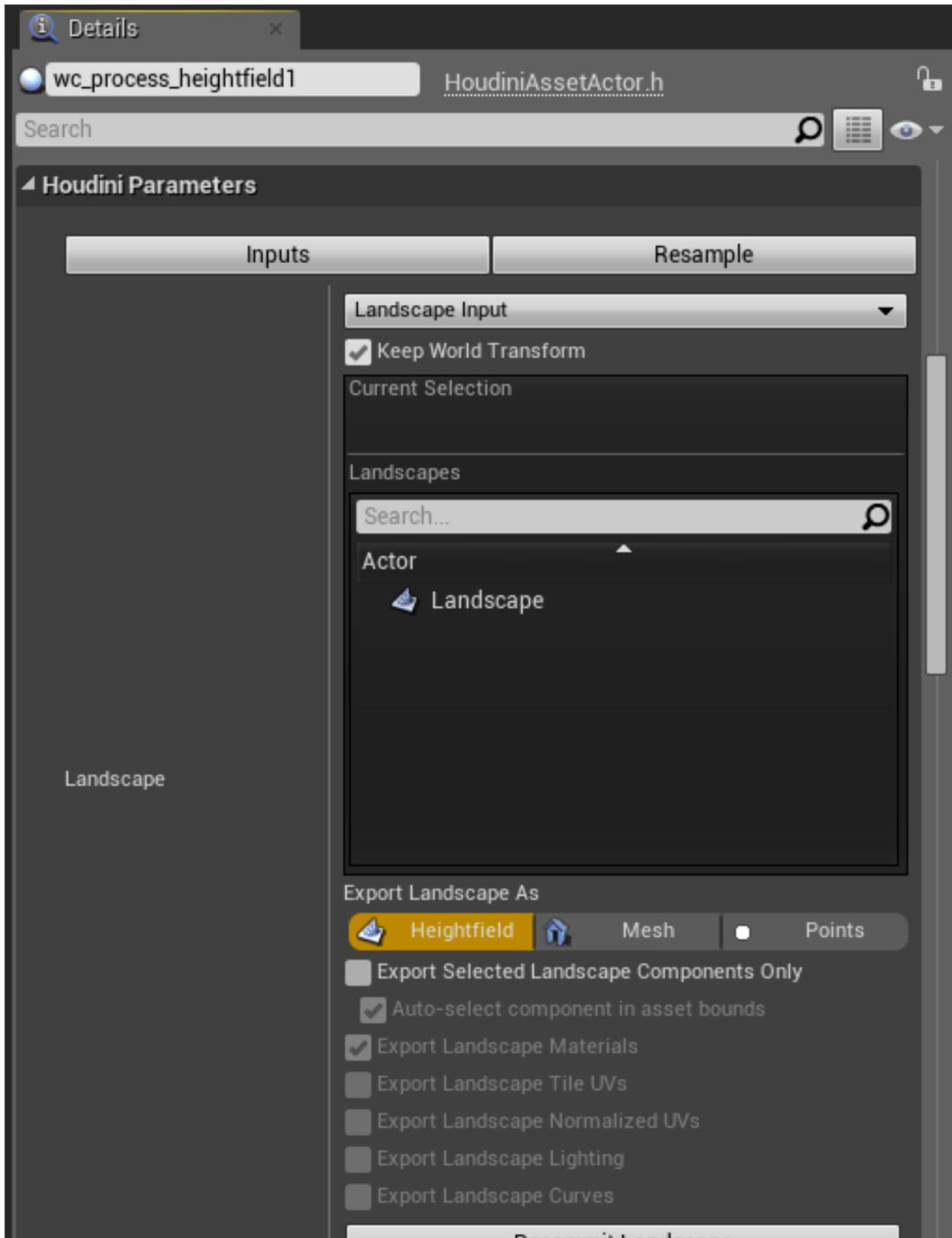
Before beginning it is recommended to create a folder called “HDA” in the Scene Outliner, so you can keep all the Houdini Engine related assets together.

### 3.7.2 Process the Landscape to a Mesh

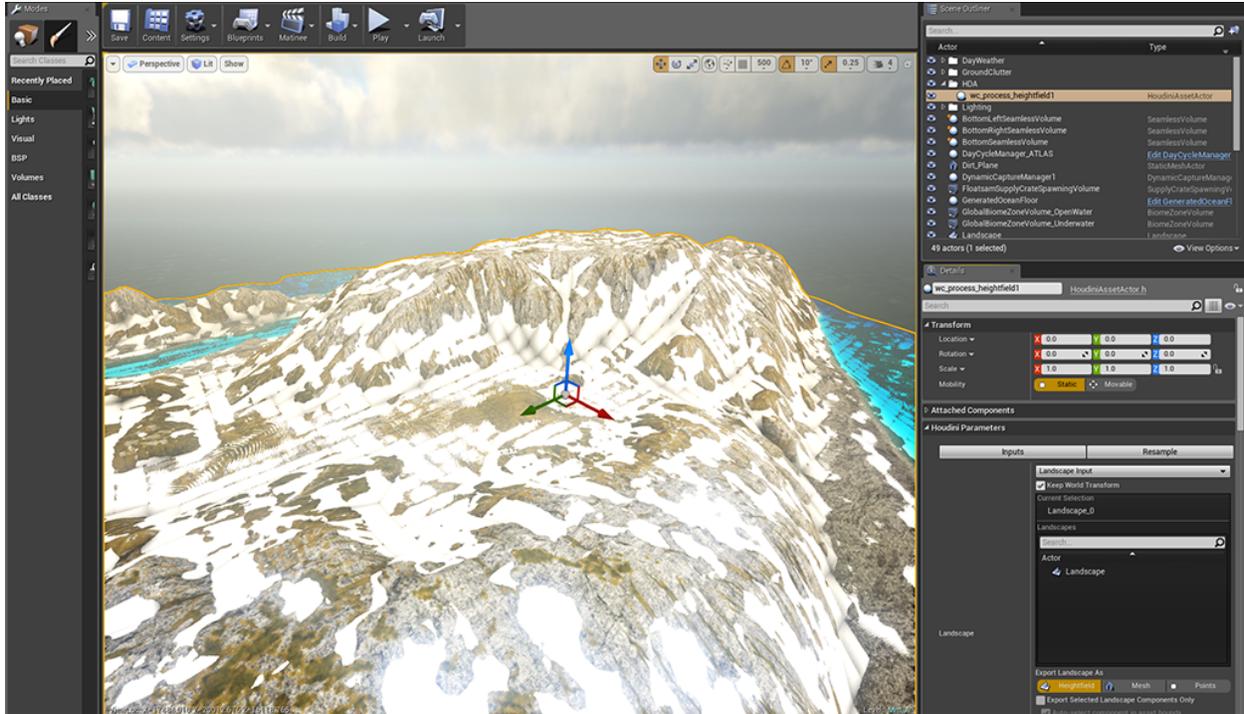
The first step in this process is to turn the landscape in which you want to scatter foliage onto, turned into a mesh. To do this we need to use the `wc_process_heightfield.hda` This is located under `Game/HDA/Procedural_Foliage` folder in the Game Editor. Simply drag this HDA into the scene you want to work with. As a good practice it is recommended to 0 out the world space transform for the asset.



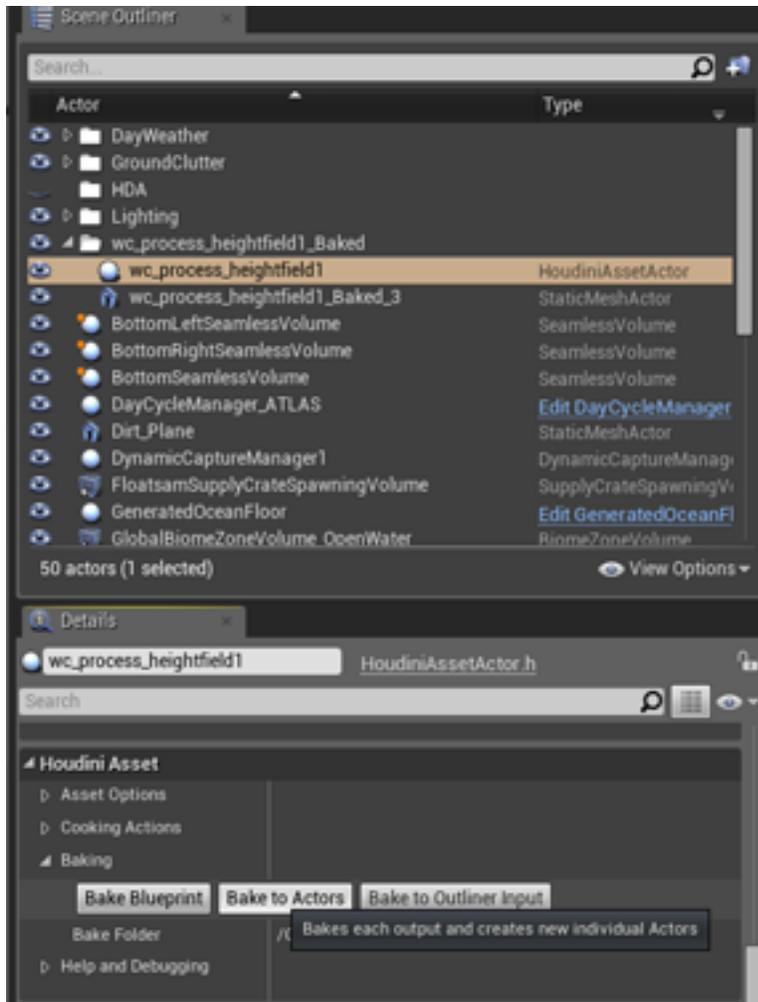
In order to process the Landscape into a mesh we need to set the landscape, we want to convert, in the Inputs tab of the HDA. You can find this in the Details panel for the HDA. Set the dropdown to the “Landscape Input” option. This will allow us to select the landscape we want to convert easily. Select the landscape. The HDA will automatically begin the conversion process. This can take anywhere from a few seconds to a few minutes depending on the “Grid Spacing” value set in the “Resample” Tab. A higher value will take less time but will be less accurate. A smaller value will take more time but be more accurate. The default is 10 and has been working well for most islands that were tested. If you find that you are getting a lot of floating foliage, it is recommended to reduce the Grid Spacing value and re-bake.



When the process heightfield HDA is completed you will see something similar to the image below.



With the Landscape converted to a mesh. We should bake it so we don't accidentally trigger the HDA to cook again. You can keep the `wc_process_heightfield` in the scene if you would like. From here all we need to do is bake the mesh to a static mesh. To do this we simply need to hit the "Bake To Actors" button found in the "Houdini Assets" fold out. When you run this command, it will take the live Houdini Mesh and convert it to a regular mesh that we will use to snap foliage to.

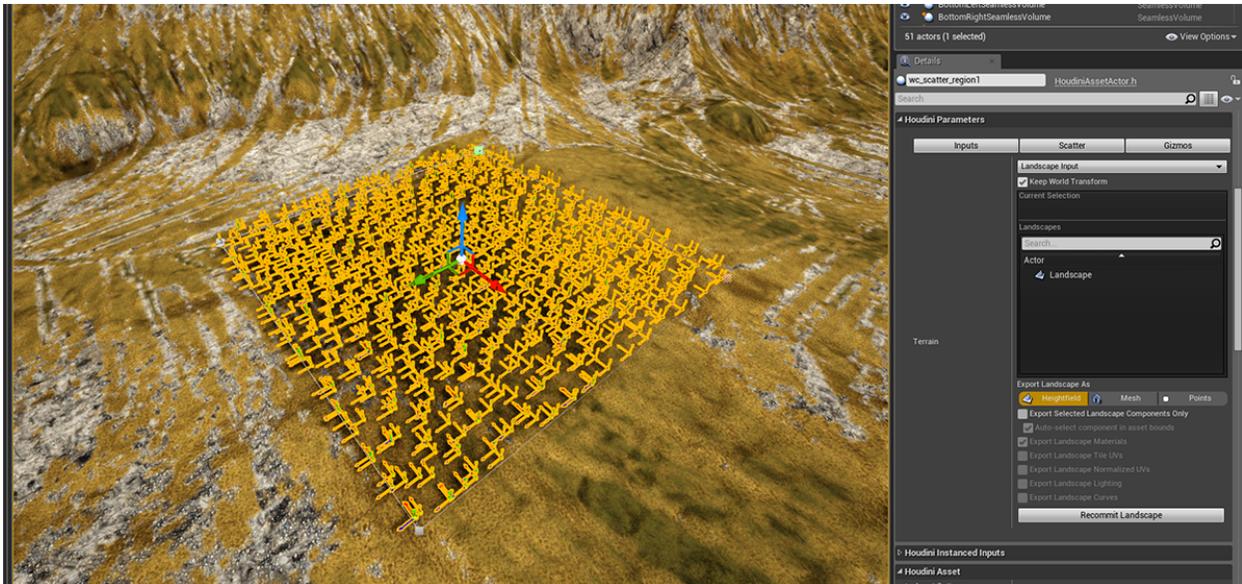


That concludes the setup of the snap mesh we need in order to snap our foliage to. You are completely welcome to keep the original HDA around if you need to re-bake the landscape, or you can delete it and just keep the bake out mesh. It is also possible to import a mesh from another source, if you would like to reduce the mesh in another DCC package.

At this point, you can hide the snap mesh so it doesn't get in the way of your foliage work. Once the generated foliage is created, or the entire island has been covered in foliage, you can delete this snap mesh if you so choose. It is definitely recommended to keep it around if you need to re-generate the scatter results.

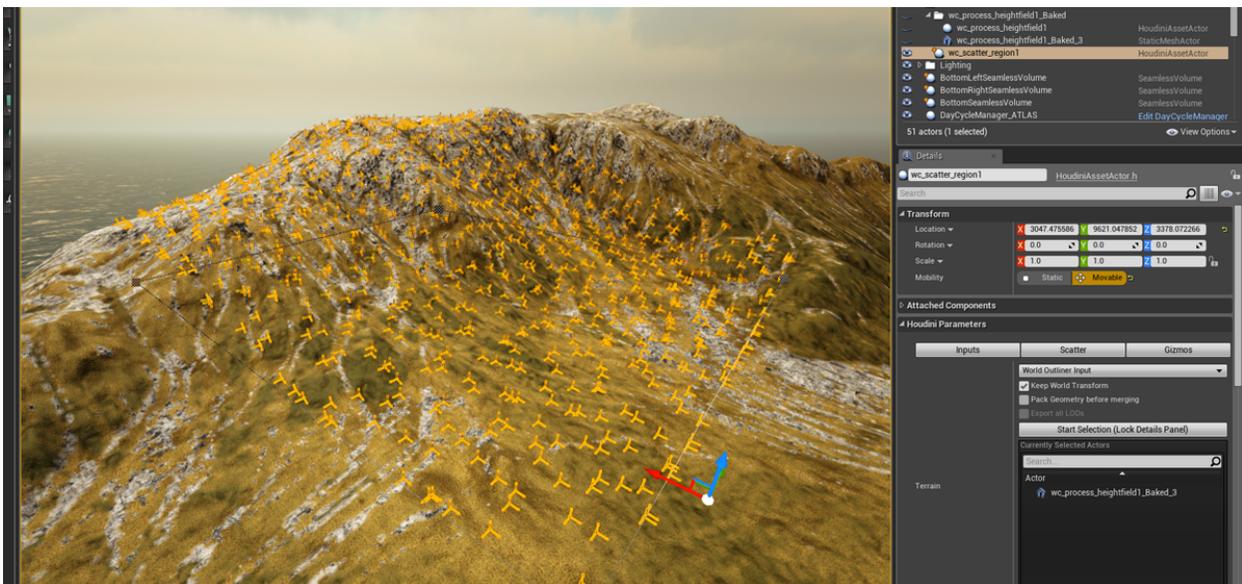
### 3.7.3 Create a Scatter Region

The Scatter region is the start of all scattering of foliage. It is called "wc\_scatter\_region". This HDA, will create a starter curve for you to edit. To edit the curve simply click on the individual points and hit "W" on the keyboard to reveal the transform gizmo. Use the transform gizmo to move a point around in world space. To select more than one point on the curve, hold down the "CTRL" key and select the points you wish to move. If you want to add more points to the curve then select a point on the curve and hold down the "ALT" key. Then drag out the new point.



The idea behind the Scatter Region is to allow us to keep the original scatter curve and points around so we don't lose our placement work. Every individual HDA after this step will simply instance the type of meshes they are responsible for and allow us to modify their properties. You'll notice that the Scatter region comes with small gizmos so you can see if the points within the scatter region are being snapped. This is simply a debug display type of utility and can be turned off in the Gizmos tab of the HDA.

To get the Gizmos and Points of the Scatter region to snap to the landscape we need to assign the Snap mesh to the Scatter Region HDA. To do this we need to switch the Terrain Input for the HDA over to "World Outliner Input". With that selected we need to do the assignment of the snap mesh. To do this, hit the "Start Selection (Lock Details Panel)" button once. This will lock the details panel from selecting anything else and wait for you to hit the button once more. Once hit for the first time, select the baked out landscape mesh from the "Scene Outliner", then come back to the scatter region's details, and hit the "Start Selection (Lock Details Panel)" again. If done correctly you will see the little gizmos, in the scatter region, snapped to the landscape.



From within the Scatter Region Houdini Parameters, you will notice a "Scatter" tab. This is where you will define the amount of points you want to scatter for each region. Rocks, Trees, and Low Level Vegetation points are controlled here. You can also adjust the spacing between each of the scatter points so you can achieve more clumping or to space

out each point evenly.



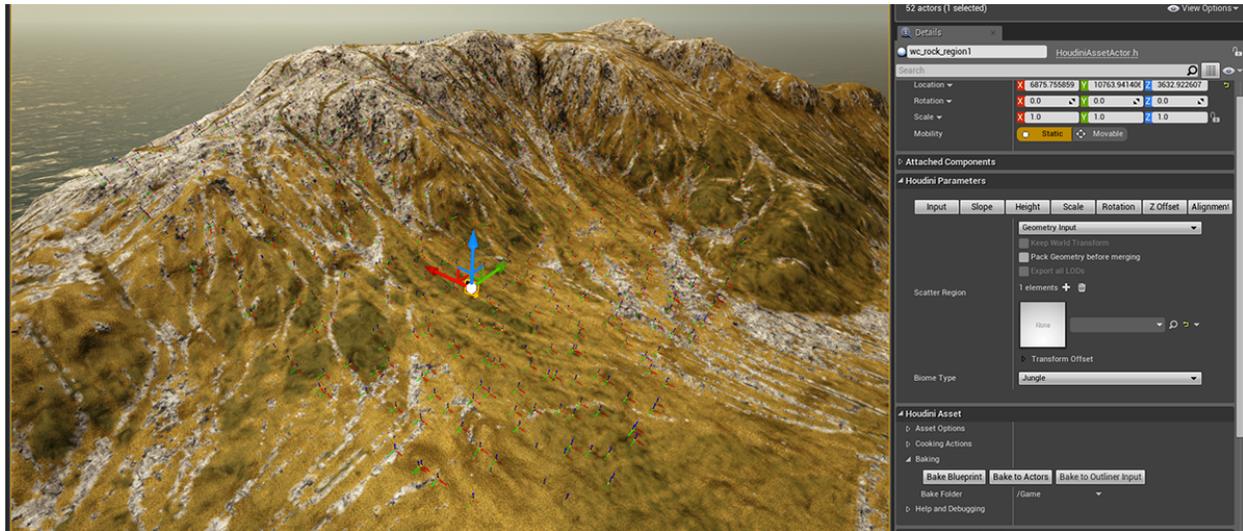
The “Scale Radii” is the slider you are looking for to adjust clumping or to get the points to be spaced evenly from each other.

Always remember that if you need more points for a specific region, you will have to come back to the scatter region to change the amount of scatter points per type of foliage.

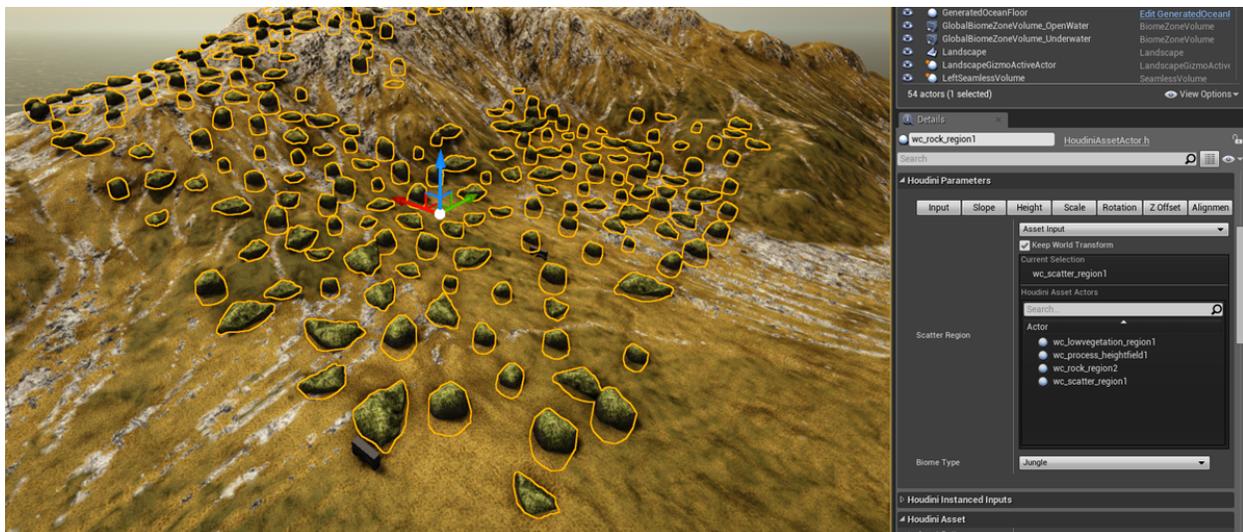
### 3.7.4 Create a Rock Region

The Rock region is dedicated to scattering rocks and rocks only. The Reason behind this is that we want to keep track of where all the scattered rocks are, so we can use that data later on, as obstacles for the trees and low level foliage.

To get started with the Rock Region simply drag out the “wc\_rock\_region” HDA into the scene, ideally close the region that it is associate with. It doesn’t matter where you put it in the scene, but for organizational purposes it is recommended to put it inside the scatter region that will drive the rock region. Once placed the HDA will not look like it did anything. This is because it needs a “Scatter Region” assigned to it. All of the following HDA’s need a scatter region in order for it to work. The scatter region contains the points necessary to instance rocks.



To assign the scatter region make sure you are in the “Input” tab of the rock region. Then change the dropdown to “Asset Input”. this will allow us to assign other HDA’s to this HDA. Hence allowing us to pass the points from the scatter region to the rock region. Once you switch the dropdown to “Asset Input”, you’ll get a list of regions that are currently in the scene. Select the region you wish to associate to this region and select it. Once done you will see rocks being instantiated into the region.



These Rocks can be changed in a couple of ways. You can select from a preset in the dropdown called Biome Type, or you can replace the instanced mesh directly in the “Houdini Instanced Inputs” foldout. This foldout exposes the current set of static meshes being used by the HDA. The Biome Type dropdown includes built in static meshes that are

commonly used in the individual biomes.

The screenshot shows the Houdini Asset Actor interface for the asset `wc_rock_region1`. The interface is divided into several sections:

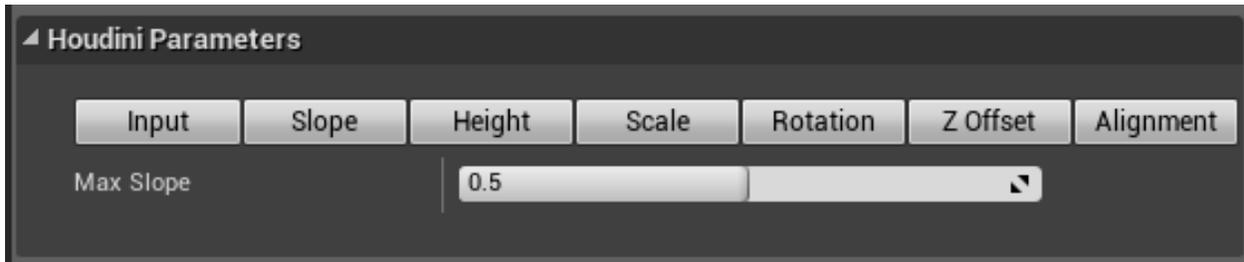
- Header:** Shows the asset name `wc_rock_region1` and the file path `HoudiniAssetActor.h`. A search bar is present.
- Properties:**
  - Keep World Transform
  - Current Selection: `wc_scatter_region1`
  - Houdini Asset Actors: A list of actors including `wc_lowvegetation_region1`, `wc_process_heightfield1`, `wc_rock_region2`, and `wc_scatter_region1`.
  - Biome Type: `Jungle`
- Houdini Instanced Inputs:**
  - `merge_orig_points_0/Over`
    - Asset: `SM_MediumRock_01_M` (with a preview image of a rock)
    - Rotation Offset: X: 0.0, Y: 0.0, Z: 0.0
    - Scale Offset: X: 1.0, Y: 1.0, Z: 1.0 (with a lock icon)
  - `merge_orig_points_0/Over`
    - Asset: `SM_MediumRock_02_M` (with a preview image of a rock)

Each of the HDA's in this system uses the same convention for the instance meshes. Meaning each HDA has the same "Biome Type" dropdown and the "Houdini Instance Inputs" foldout. If ever you need to add a specific static mesh to one of the Biome Types, just let Kenny know and he'll add it to the system.

All of the Foliage regions have almost the same parameters so we will begin with the Rocks Region and walk through each of the individual tabs.

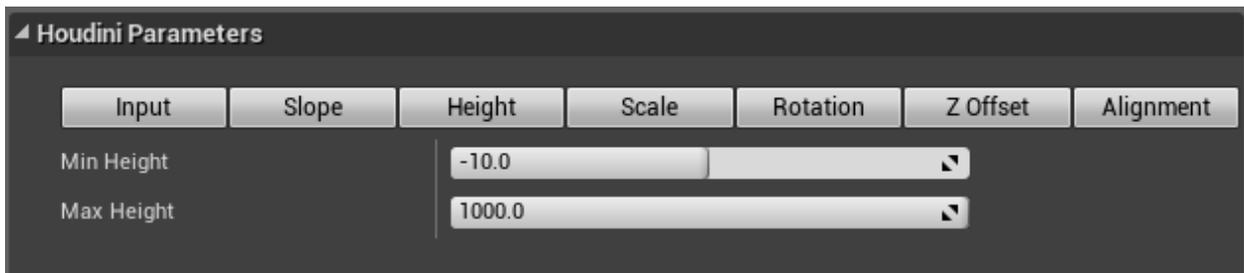
### Slope Tab

The slope tab is where you can adjust the slope for the entire region. A value of 0 means that rocks will only be placed on extremely flat areas. A value of 1 means the region will ignore the slope. The current value is in radians meaning  $1 \approx 90$  degrees. If you set a value of 0.5 rocks will only be placed at a slope of 45 degrees or less.



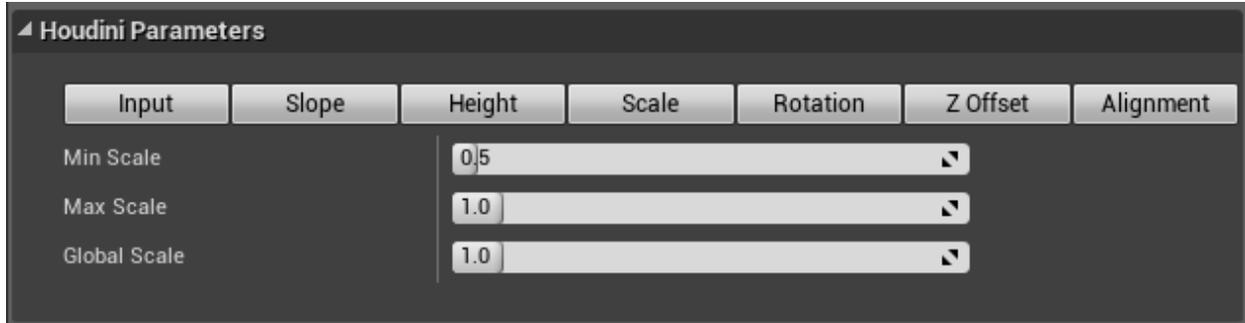
### Height Tab

The Height tab allows you to adjust at which heights a certain foliage can be scattered at. There is a min height and a max height slider. By default the min height slider is set to -10 or just below sea level. The max Height is set to 1000 by default which allows foliage to grow at most heights. Simply adjust the sliders to the height you like. It is recommended to start with the min height since the max height is already well above most landscape heights.



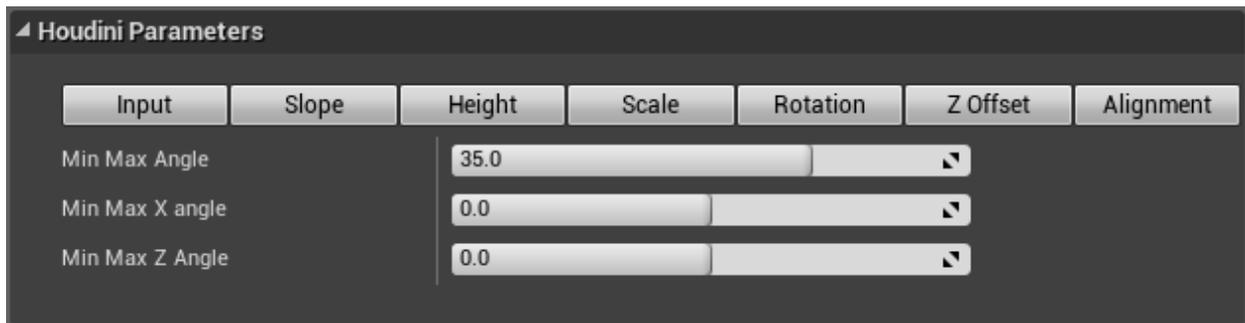
### Scale Tab

The scale tab allows you to control the min and max scale of all the scattered instanced meshes. The min scale is the minimum possible scale an instanced mesh can be. This is based off of the original scale of the object. The maximum scale is the max scale any object can be and is based off of the original scale of the static mesh. Finally the global scale allows you to add scale across the board to each object.



### Rotation Tab

The rotation tab allows you to manage the minimum and maximum rotation angle for each instanced mesh. By default the Min Max Angle is set to 35. Simply adjust this slider to change the random rotation about the z axis of the object. This means that all scattered objects will get a random rotation between -35 degrees and 35 degrees. The Min Max X angle allows you to add some random rotation to the other axes for the mesh. Most of the time the main axis is fine, but these other sliders allow you to tweak as much as possible.



### Z Offset Tab

Here you can set how much z offset you'd like to give to the scattered meshes. Positive values raises the mesh and negative values lower the mesh.



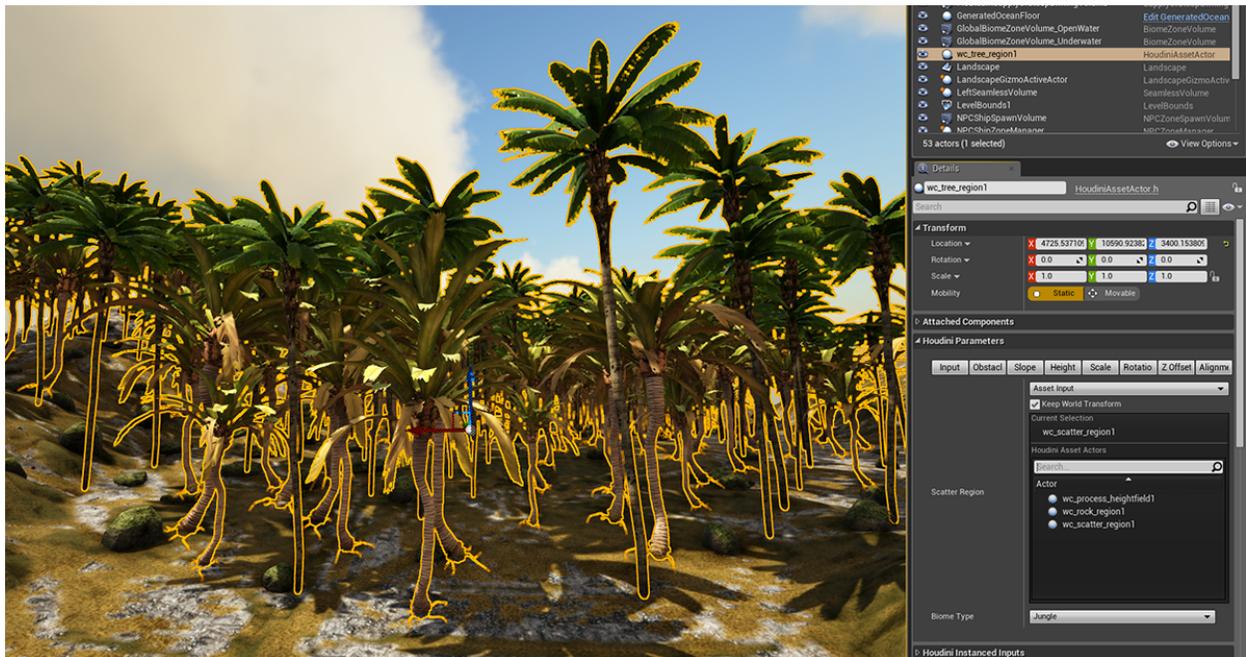
### Alignment Tab

The alignment tab allows you to align to the normal of the landscape or not. If this toggle is turned off then the alignment will be flat to the world.



That concludes the information on what is contained in each tab of the Rock Region HDA. Each of the following HDA's have exactly the same parameters and work in the exact same way. Only features custom to each HDA will be covered from here on out.

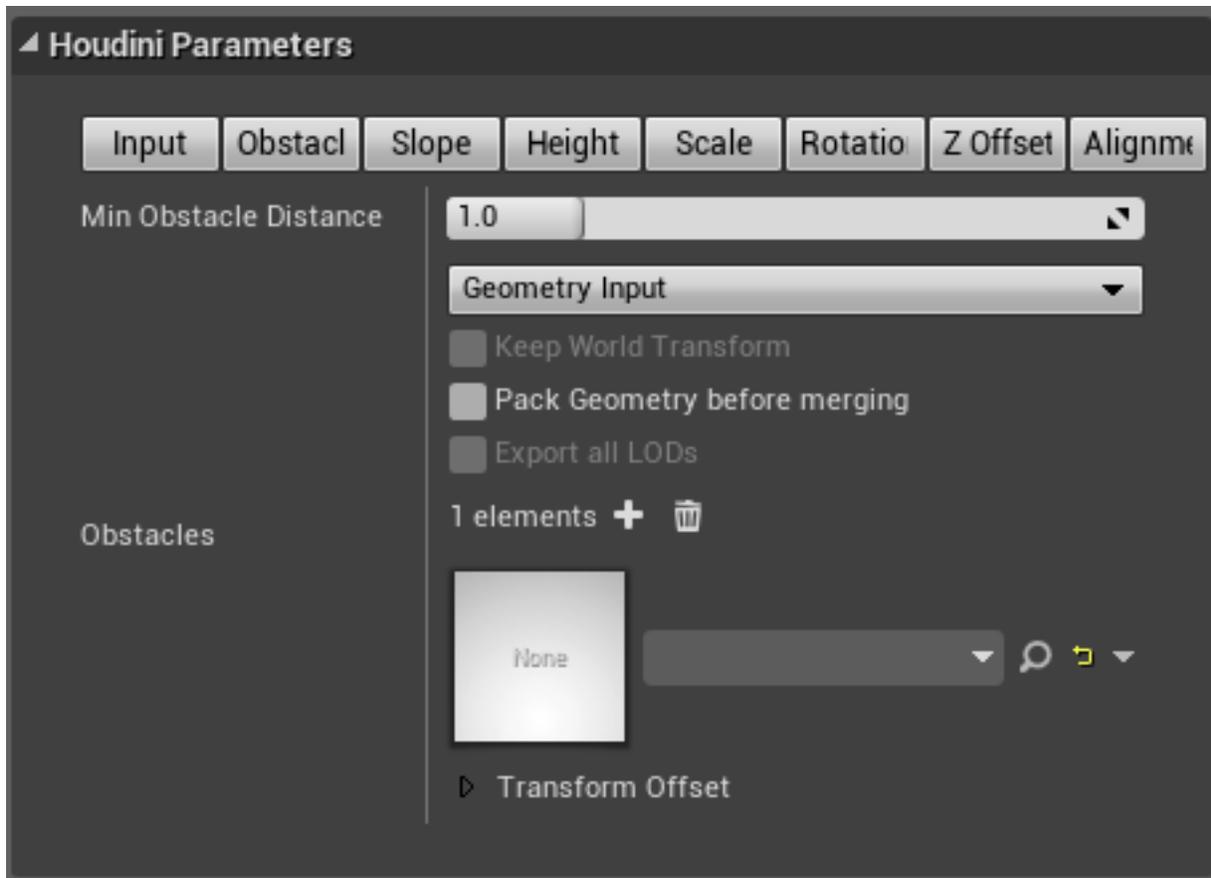
### 3.7.5 Create a Tree Region



The Tree region is dedicated to scattering trees and trees only. The Reason for this is to keep track of where trees are in the scatter region. We will use this data later on when we create the low level foliage region, so as to keep the low level foliage from growing inside a tree.

#### Obstacles Tab

In the Tree Region HDA you have the option to assign a rock Obstacle object or HDA. This can be used to pass the Rock Region into the Tree region so you don't get trees growing inside of rocks. Switch the Obstacles input over to an "Asset Input" to assign the rock HDA to it. Once assigned you can use the Min Obstacle Distance to make sure Trees only grow a certain distance from rocks and not inside of them.



### 3.7.6 Create a Low Level Vegetation Region



The low level vegetation region is meant for scattering things like ferns and larger plants but not clutter foliage. The name of the HDA is “wc\_lowvegetation\_region”. The only different input parameter that is included with this HDA is the Tree Obstacle found underneath the Obstacles Tab. Just like the rock obstacles for the tree region the tree obstacle ensures that no vegetation can grow inside of a tree. If you don’t assign a Tree region to this input the Low Level Veg region will just ignore it and you might get vegetation growing inside of trees.

#### Convert Scatter Foliage to Unreal Foliage

Once you are happy with the results from the scattering process, it is now time to convert all the work to Unreal foliage so it can be edited with the Foliage painter within Unreal. To do this, simply open the command line using the “~” key on the keyboard and type in “meshestofoliage”. This will convert the HISM meshes to foliage meshes. If a Foliage Actor does not exist already in the scene then one will be created. Once completed, you can now use the foliage brush to further refine the foliage placement.

## 3.8 Proxy Meshes

### 3.8.1 Overview

This page will provide a brief overview of using the proxy mesh tool that we use for merging instanced meshes into large pieces that use fewer draw calls. In general the instances should be moved to their appropriate near sub levels (If

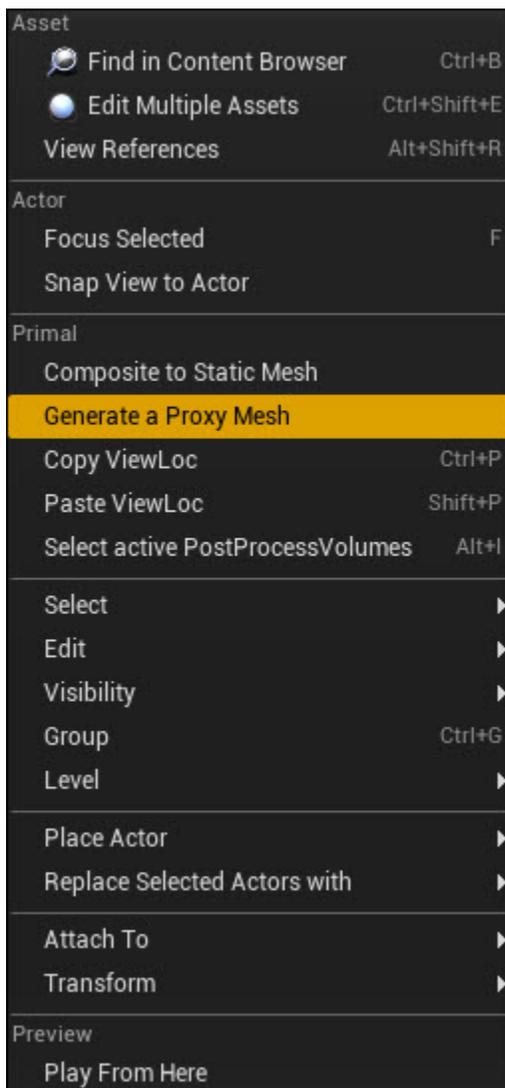
they are not already) to reduce memory and the proxy mesh will be placed in the matching far level. The proxy mesh is arbitrary data saved within the level. No new assets (Textures/Materials/Static Meshes) will need to be committed via the content browser.

### ATLAS Specific

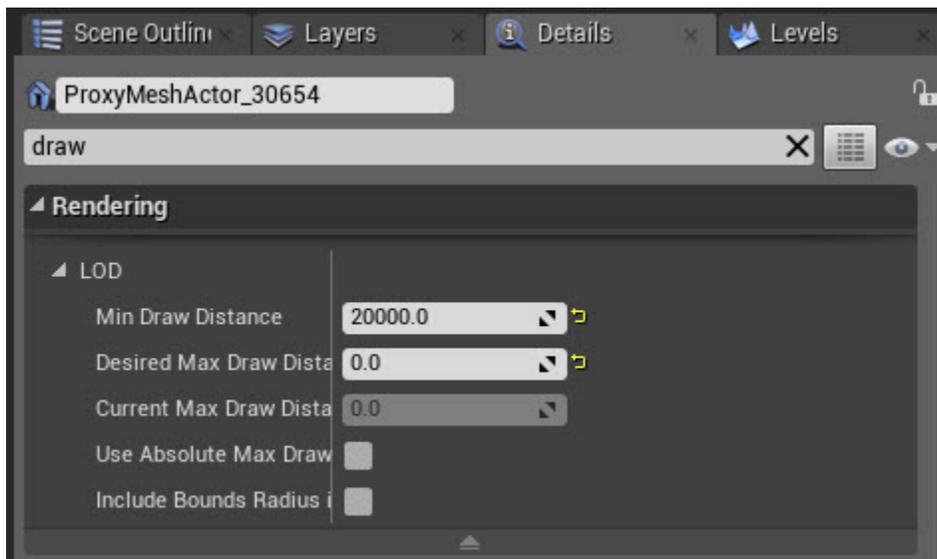
All of the large features in the farthest levels should have a proxy mesh parent that is in the same level. It's max draw distance should be set to 200,000 unreal units with use absolute draw distance set to true. The Islets can have a max draw distance of 100,000 unreal units because they are so small.

### 3.8.2 General Workflow

1. Select the actors you would like to create a proxy mesh from and make sure they are in a near sublevel. Try and make this selection as large as possible but sometimes it might be needed to split a group up into multiple sections if the bounding box of the proxy mesh will be weird.
2. Right click on the selected actors and choose "Generate a Proxy Mesh".



- The proxy mesh tool will display the following options. In general the default values are usually good to use.
  - Set Triangle Percent** - You usually want to keep this as low as possible. 1-2 % is usually a good starting point. If a static mesh has a proxy mesh override in its static mesh settings set triangle percent to 100% (This is very rare to use and 99.99% of the time you don't need to worry about this).
  - Max Draw distance** - Keep this at 0 but it can be adjusted later.
  - Proxy Transition Distance** - Usually it is best to keep this at 30,000 as a starting value but it can be adjusted after the proxy mesh is built to get good results.
  - Select Proxy Level** - Here you will select the matching FAR sublevel for the selected assets. The proxy mesh will be placed in this level.
  - Mesh LOD Bias** - Usually the default value of 0 is fine. If you place a bias it will merge and optimize those mesh lods instead of lod0.
  - Max UVs** - You can leave this at the default value of 8. This is mainly for very specific times when we use the tool for SpeedTree assets.
  - Use Simplygon Remesher** - Enable this if you want to bake the textures down so that the proxy meshes all use one material and texture. Can be useful if the generated proxy mesh has more than a few materials applied. If this is enabled you might have to adjust the Set Triangle Percent higher.
  - Allow Color** - Transfers vertex color data to the proxy mesh. It is usually good to leave this enabled.
  - Texture Size X and Y** - Here you specify the size of the texture the proxy mesh will use when Use Simplygon Remesher is enabled.
- After you adjust any settings click on ok and the tool will run. When it is finished the proxy mesh will be selected and if you move the camera around you will see when it swaps to the near meshes. In the Scene Outliner all proxy meshes will have a (P) prefix and all children will have a (C) prefix. Under the proxy mesh details you can adjust the Min Draw Distance to dial in when the proxy mesh appears and its children disappear. This number is linked with the children so you will only need to adjust the proxy mesh Min Draw Distance.



- If the proxy mesh has more than a few materials applied in its details panel consider enabling Use Simplygon Remesher listed above to bake everything down into a single texture and material. You may need to adjust the triangle percentage to be higher when enabling this option.
- When everything is dialed in don't forget to save both the near and far levels! All proxy mesh parent assets that cover a large area should go into a separate proxy mesh level and the children should stay in the far level.

3. After a proxy mesh has been created additional options will be available when you right click on it, or its children, in the editor.
  - **Re-Generate a Proxy Mesh** - regenerates a proxy mesh from the children (Might be broken)
  - **Re-ID Proxy Mesh Group** - Used to link proxy mesh with correct children in case it somehow gets broken. You will need to make sure that the children AND the proxy mesh are selected for this to work.
  - **Clear Proxy Mesh Data** - Removes the proxy mesh data.
  - **Select Proxy Mesh children** - Selects all of the children of a particular proxy mesh.
  - **Select Proxy Mesh Parent** - Selects the proxy mesh any particular child(ren) might use.

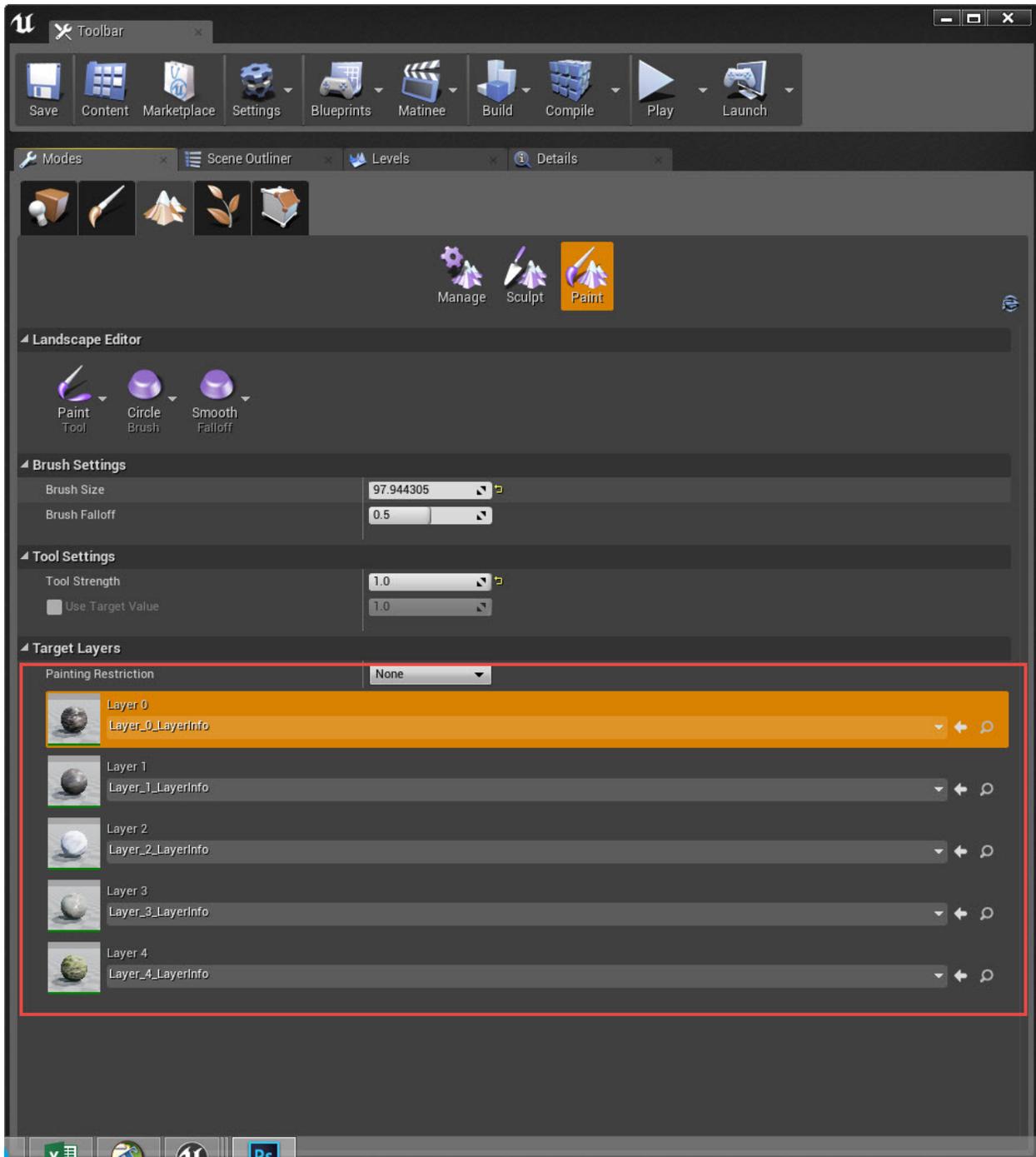
## 3.9 Static Mesh Landscape

### 3.9.1 Overview

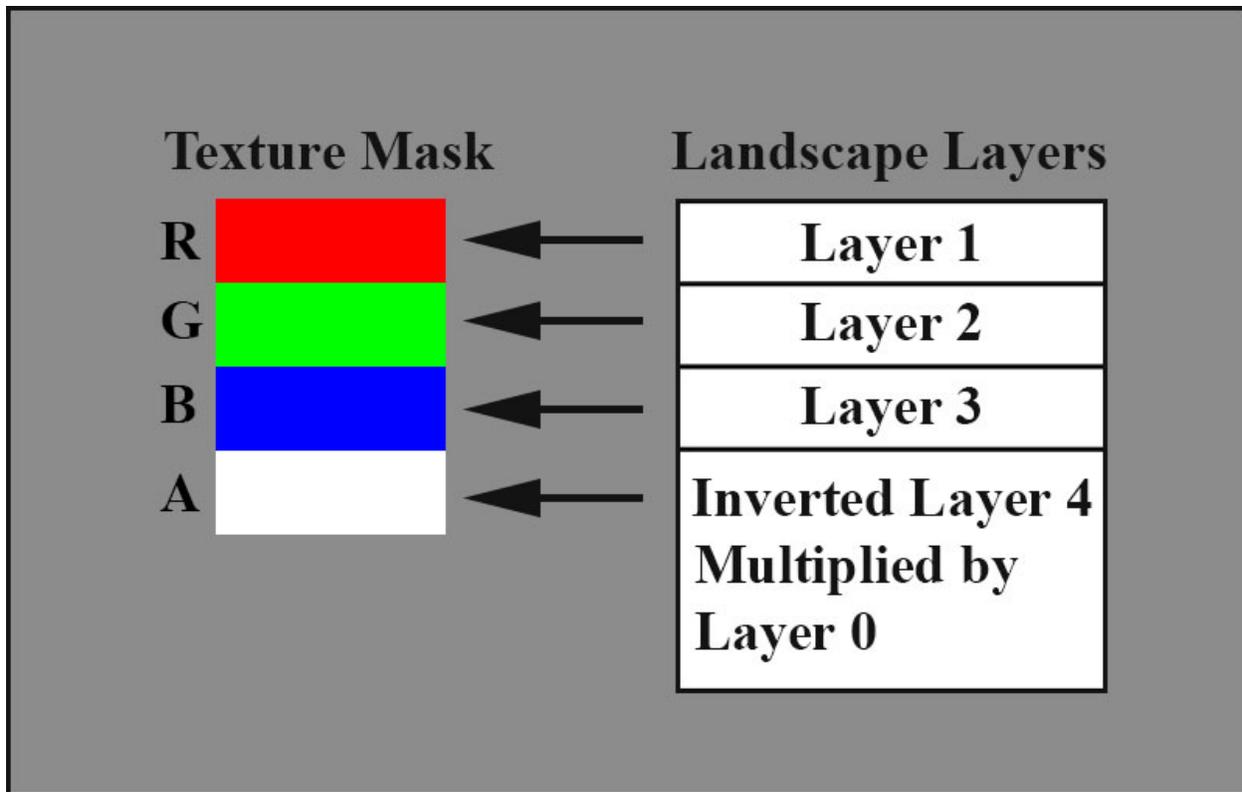
This page describes the process of converting a landscape that uses the MM\_VertexBlending parent material into static mesh geometry and transferring all of the weight blending.

### 3.9.2 Creating Texture Mask

The first step that needs to be done is exporting the landscape layers using the landscape tool. Simply right-click each layer and click on “Export to file”.



Open the five layers inside of Photoshop and create a new image that has the same dimensions as the exported layers. This new image will be the texture mask that the new material will use. In the new image you want to paste the layers into the following channels.



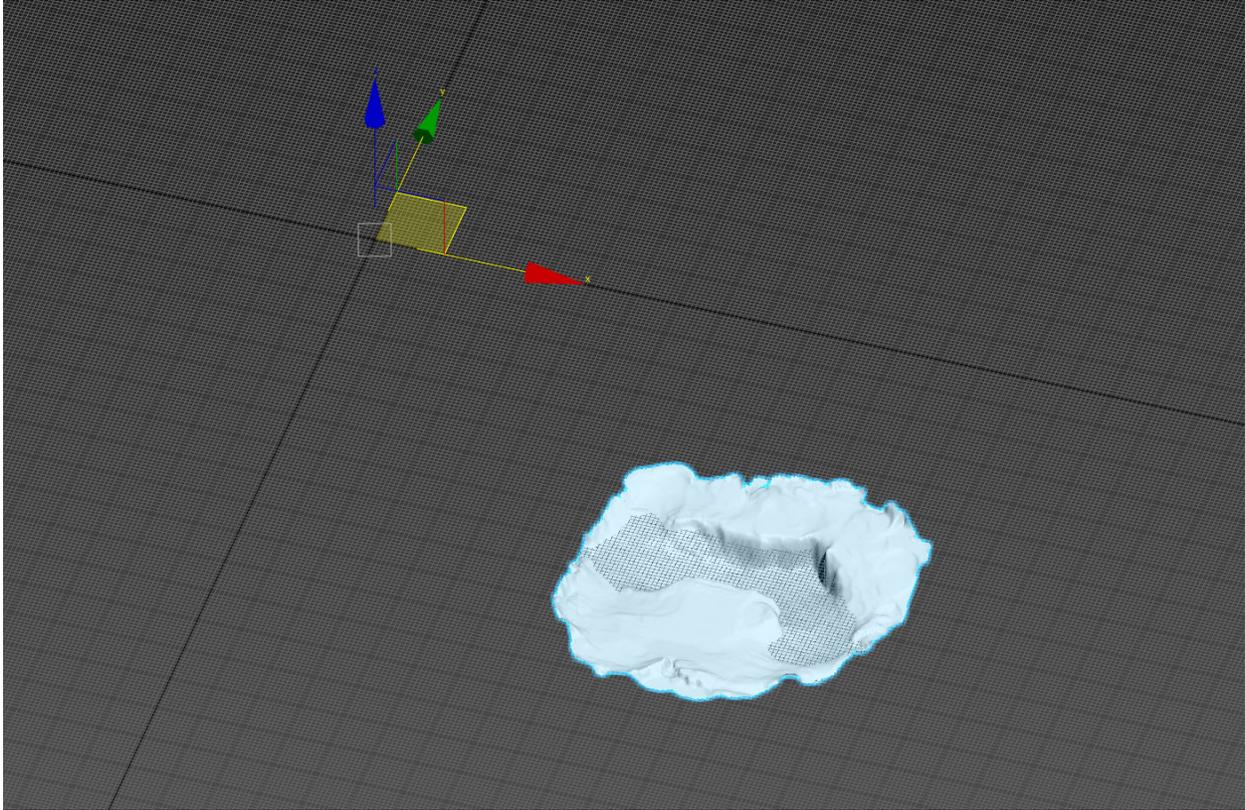
After transferring the landscape layers you will need to flip the new texture mask vertically and save it. Inside of Unreal import the new texture mask. Make sure to disable sRGB and choose TC Vector Displacementmap for compression settings.

### 3.9.3 Export/Import Landscape

Select the landscape and click on “Export Selected...” from the Unreal file menu. Save the file in FBX format so that it saves out all of the UV channels. Import the FBX file into any 3d program. Make sure not to change the pivot! Move the landscape so that its pivot is at the origin and delete any geometry that will not be needed. Copy the UVs from the 2nd UV Channel to the 1st and then delete the 2nd UV channel.

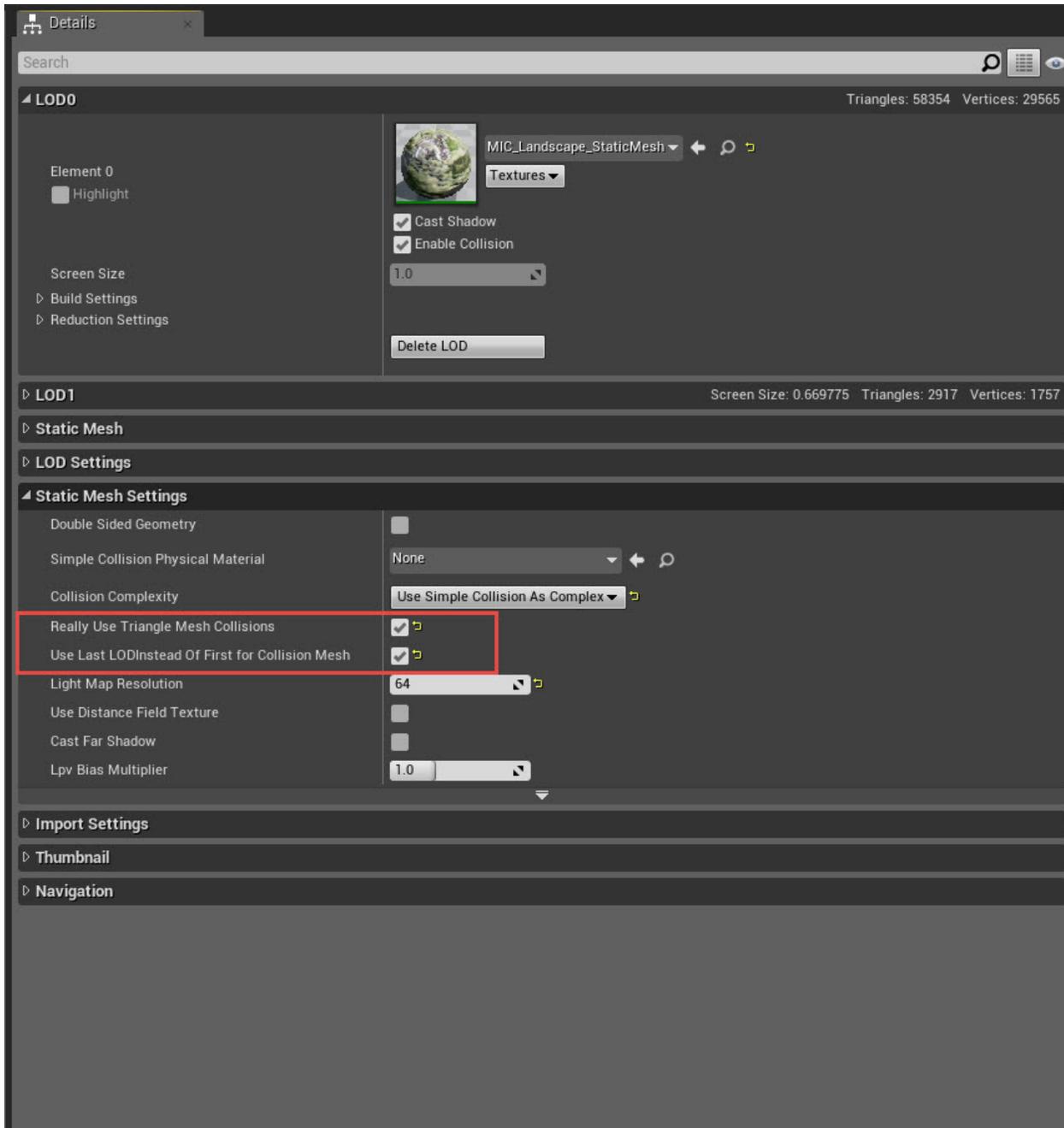
Now you must transfer the colors from the texture mask created above to the vertices. In Maya select the landscape geometry and click Mesh Display -> Paint Vertex Color Tool. In the tool options click on Attribute Maps and then Import. Choose RGBA for import value and then click on the Import... button and choose the texture mask you created above (Make sure you saved it out with the alpha!). After that you should see the landscape mesh with vertex colors applied in the maya viewport.

Now export the landscape geometry as an FBX file.



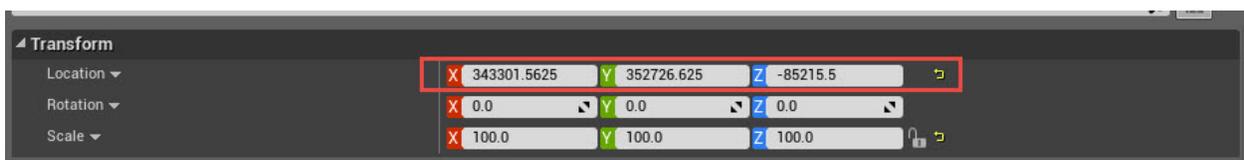
Edited landscape geo that is centered at the origin inside of 3ds max.

Import the new landscape geometry as a static mesh asset inside of Unreal. Open the static mesh properties and create an additional LOD. Reduce the second LOD as much as possible for simple and accurate collision. If the collision mesh is greater than 2k triangles the landscape mesh will need to be cut up into separate pieces. After the second LOD is created enable use triangle mesh collisions and use last LOD for collision mesh under Static Mesh Settings.



### Landscape Collision

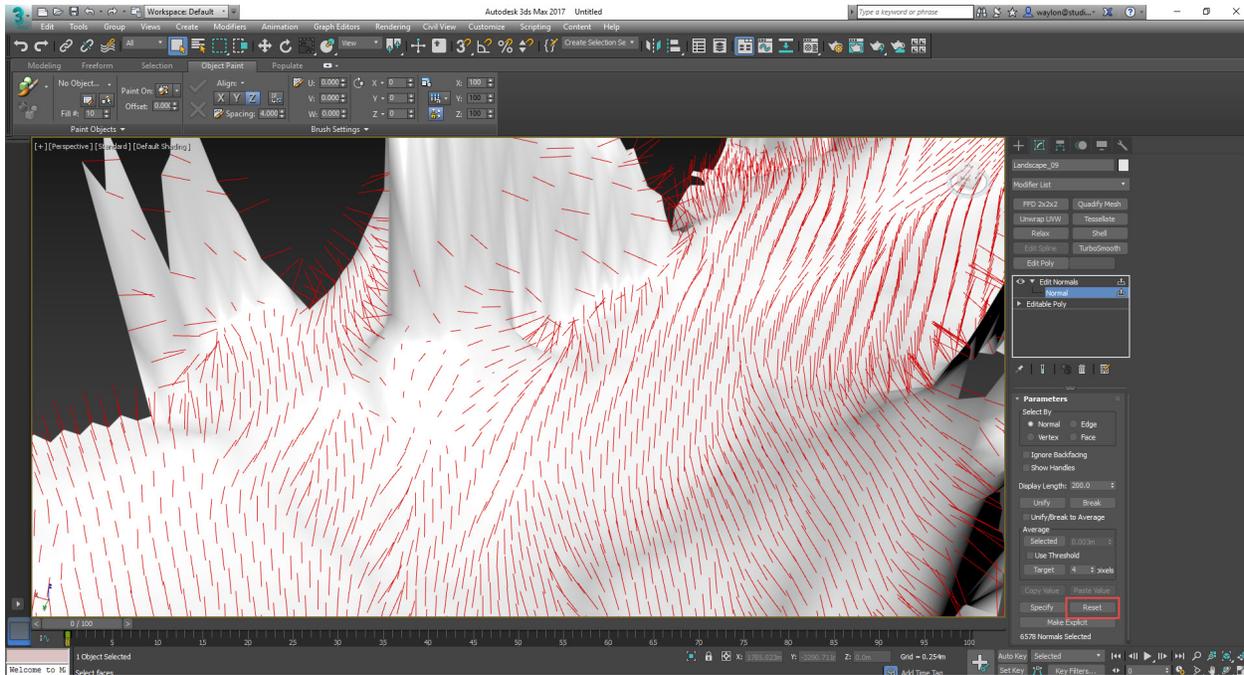
Place the landscape(s) static mesh into the level and copy the location values from the Landscape actor to the new static mesh. This will make sure that both the old landscape and the static mesh are overlapping exactly.



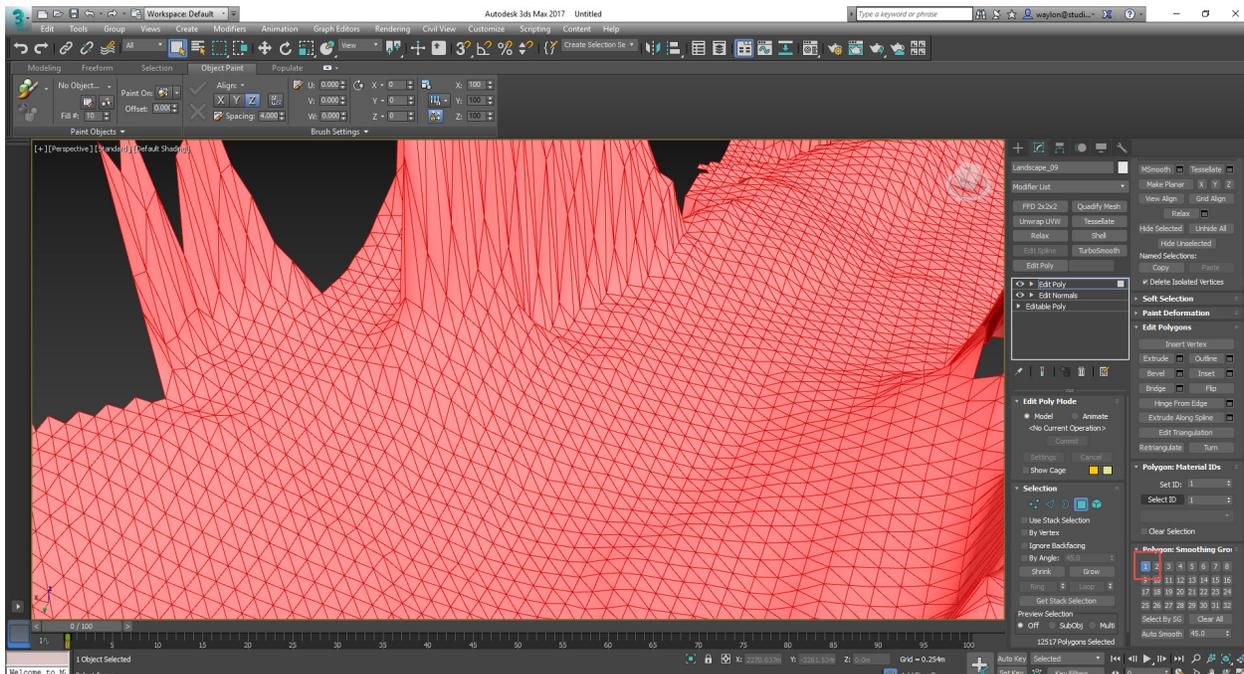
Under the landscape static mesh actor's details make sure you set "Forced LOD Model" to 1. This is so that the landscape static mesh will not use the collision LOD visually.

### 3.9.4 Normals

Before exporting the FBX it is good to reset the normals and to apply one smoothing group to all of the faces of the landscape. This can be done in 3ds Max or Maya but the example below shows the process in 3ds Max. You can usually tell there is an issue with the normals if the vert count is much higher than the triangle count in Unreal. Usually the vert count is less than the triangle count if everything is imported correctly.



Select the mesh geometry and apply an Edit Normals modifier. Press shift + a to select all of the normals and then click on the reset button. This will unlock the normals in case the mesh was imported with custom normals.



After that you can add an Edit Poly modifier, or collapse the stack, to select all of the faces. Just go to face selection

mode and press shift + a to select all of the faces. Then make sure that there is only one smoothing group applied to all of the faces, like in the screenshot. You might have to click on “Clear All” down in Polygon: Smoothing Groups if there are already multiple smoothing groups applied.

### 3.9.5 Material Setup

Now you need to make a new material for the landscape static mesh. Select the material that is being used on the old landscape and create a material instance of it. The new material should use the old landscape material as its parent. Apply the new material to the landscape static mesh. Open the new material and enable `isConvertedLandscape`. This will make sure that the UV scaling is the same as before.

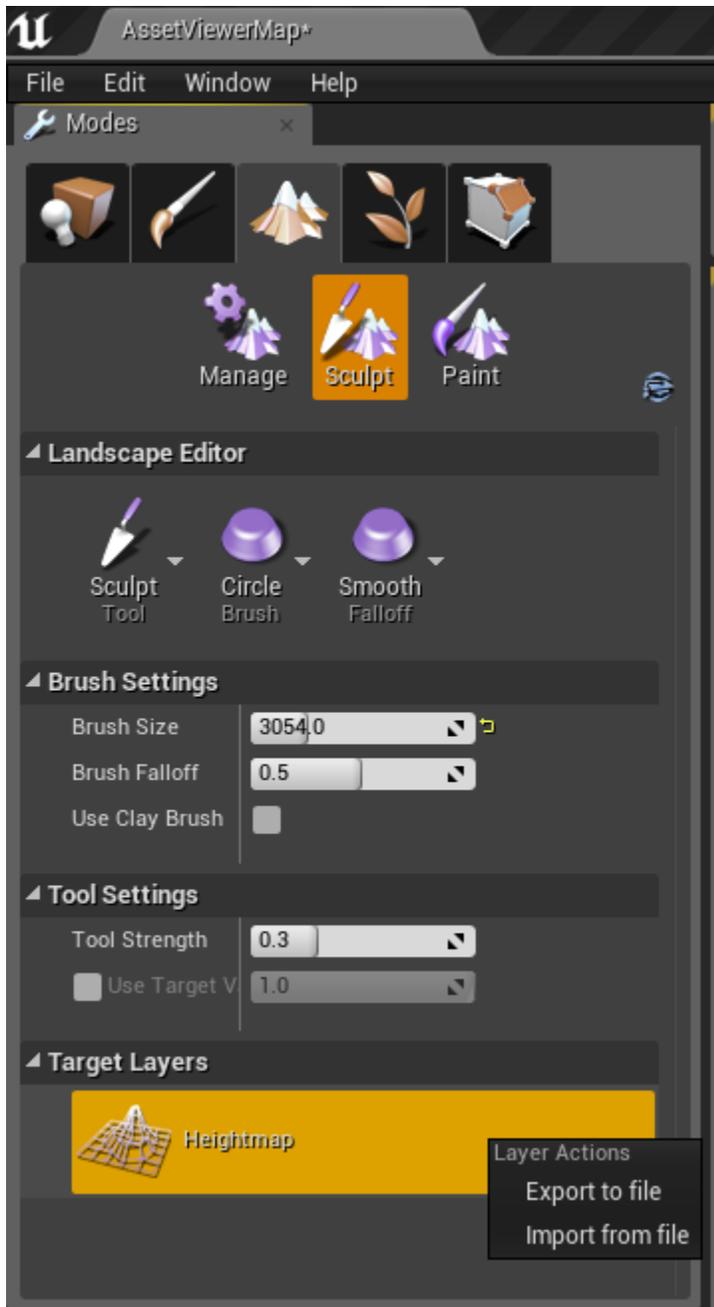
## 3.10 Terrain Heightmap

### 3.10.1 Overview

The terrain heightmap is used to drive a number of effects in Unreal, such as faked global illumination from the landscape. It can also be used to generate useful textures, like a riverbed mask for flow maps or a depth map for caustics. This page outlines the generation and some common uses of the terrain heightmap.

### 3.10.2 Generating the Heightmap

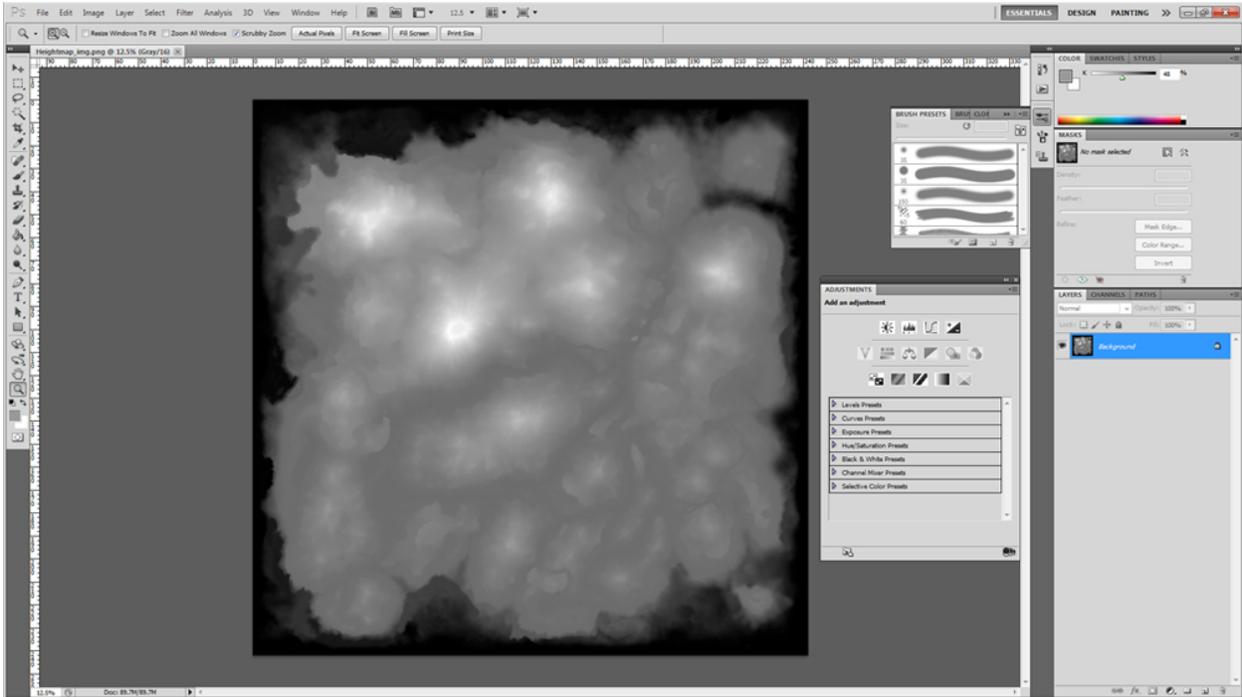
The terrain heightmap should be regenerated whenever major edits to terrain are made. To do so, select the terrain and go into Unreal’s Landscape mode by clicking the mountain icon in the Modes toolbar. Select the “Sculpt” submode and look at the “Target Layers” tab. Right click the Heightmap layer and select Export.



You may export the heightmap as either a .png or .raw; a .png is easier to work with.

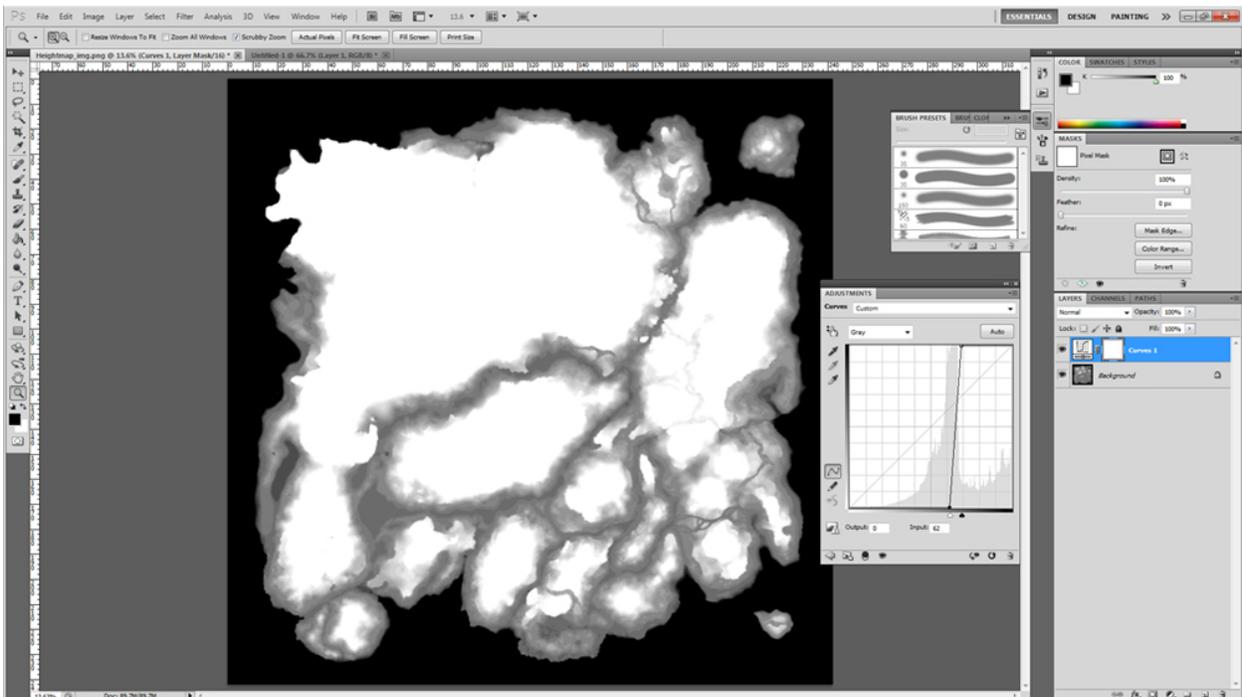
### 3.10.3 Editing the Heightmap

Unreal exports the heightmap as a 16-bit grayscale image. You should leave it as 16-bit when editing and only switch to 8-bit when exporting a texture to minimize artifacting.



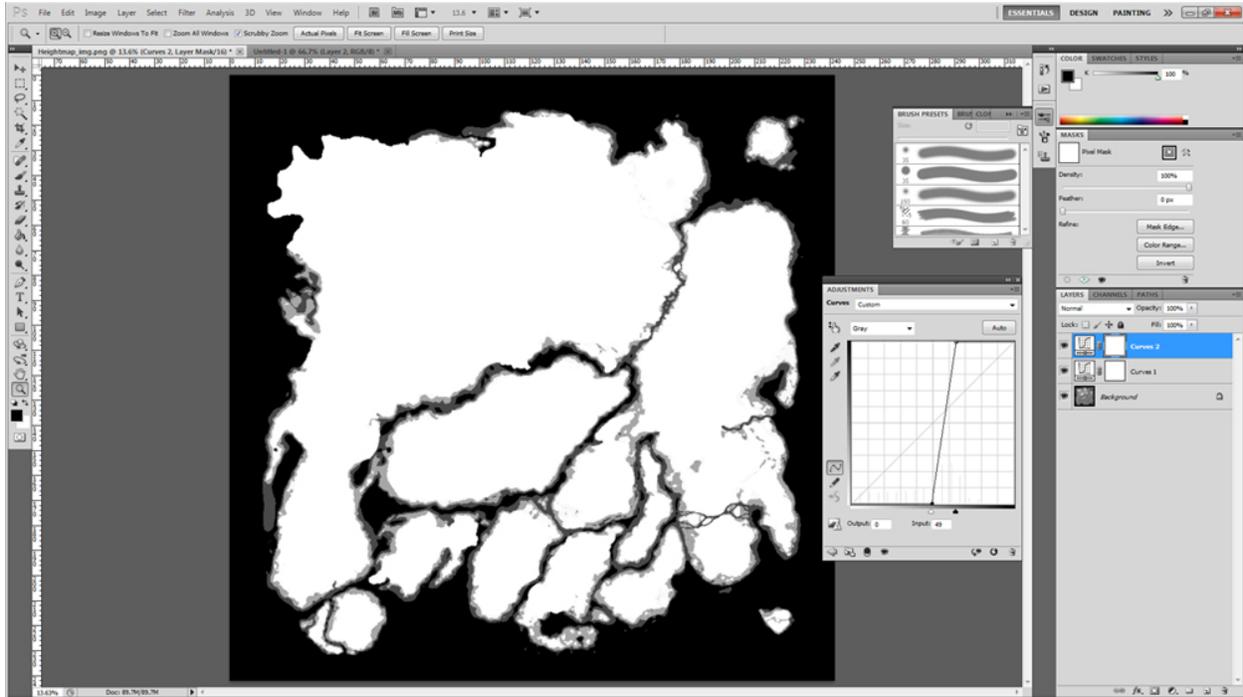
Generally, you'll want to reduce the heightmap to a specific range of values (flowmap masks, for instance, will be most accurate if you rerange the heightmap between sea level and the highest body of water). To do so in Photoshop, use a Curves Adjustment Layer and pull the black- and white-points inward.

You may see banding in gradients after doing a curves adjustment- as long as you're still in 16-bit mode, this is an artifact of how Photoshop presents the image and things should look smooth if you zoom in.



(A rough mask of sea-level rivers)

You can stack multiple Curves layers to hone in on a range of values, as below.



(A more accurate river mask. Two Curves layers should be all you need to get a usable map.)

### 3.10.4 Retaining Accuracy As An 8-bit Texture

As of writing this, the terrain height runs between about 90,000 worldspace units. The 16-bit map Unreal generates can only express 65,535 units of accuracy, and this is reduced to a mere 256 units of accuracy when converted to 8-bit. We generally want to retain as much accuracy in the heightmap as possible, and there are a few methods for doing so.

For our purposes, the best method is the “Modulo” method: we divide the 65,535 units of height into 256 chunks (each 256 units in length), storing the chunk index in the alpha channel and the sub-chunk value in a color channel. Decoding this information in Unreal is relatively easy and cheap on shader instructions (see Using the Heightmap in Unreal, below).

The following Filter Forge filter will take a 32-bit .EXR image and compress its red channel losslessly into an 8-bit .tga. You’ll need to open the Unreal-generated heightmap in Photoshop, set its mode to 32-bit RGB, and then save it out as .EXR before processing it.

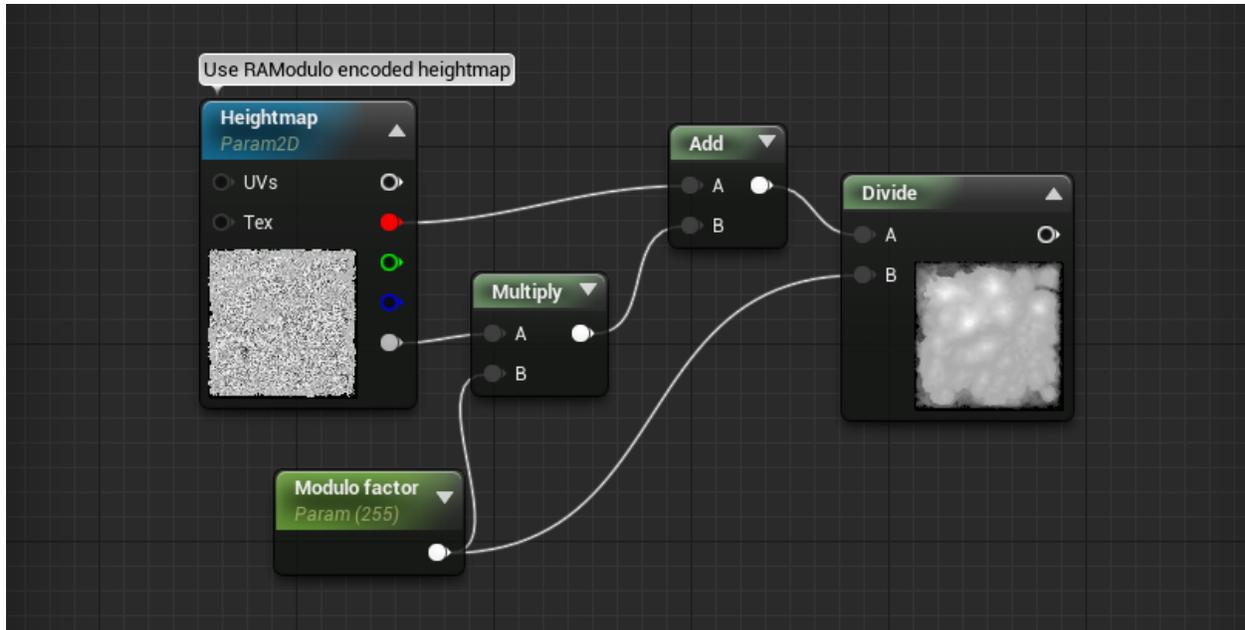
[https://drive.google.com/a/studiowildcard.com/file/d/0Bx9uxHQ19G\\_tekxydk14VmcxNjg/view?usp=sharing](https://drive.google.com/a/studiowildcard.com/file/d/0Bx9uxHQ19G_tekxydk14VmcxNjg/view?usp=sharing)

Depending on how you plan to use the heightmap, you’ll need to resize it after processing. For debugging purposes feel free to use a 4k image, but for subtle effects like landscape GI or caustics, a 2K (or even a 1k) will probably be fine.

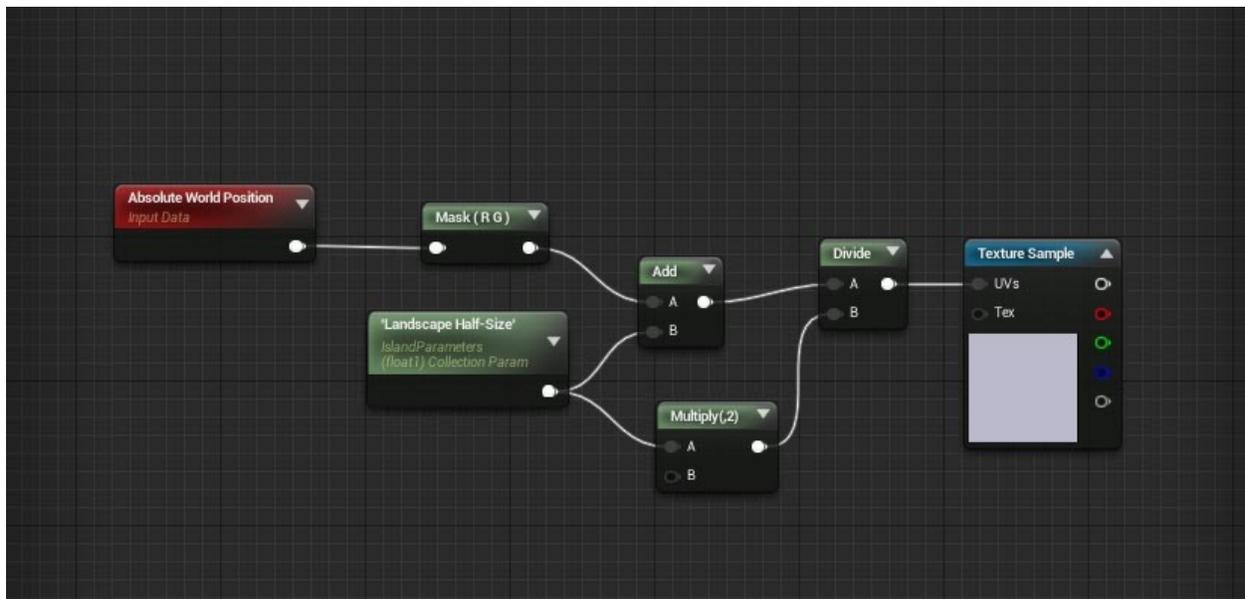
### 3.10.5 Using the Heightmap in Unreal

After importing the heightmap, you’ll need to change a few texture settings. Uncheck “sRGB” since we don’t want to gamma-correct the height values. If you’re using it at 4k resolution, set the LOD Group to “Cinematic”. If you compressed it as described above, set its compression to “TC Vector Displacementmap”, otherwise “TC Greyscale” should be fine (as long you only need one color channel).

If you used Modulo compression, you’ll need to do a bit of math to recombine the color and alpha channel into an HDR-range of values:



Typically, you will want to use the heightmap (and any similar maps, i.e. a flowmap) as worldspace maps on a per-pixel basis, querying the value of the map at a particular XY world position. To do so, use the following UV setup:



## 3.11 TrueSky

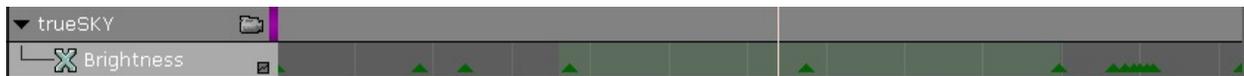
### 3.11.1 Overview

This tutorial covers how to setup new True Sky Sequences as well as some of the many options/features you need to be aware of when doing so. This tutorial is assuming you have already setup a persistent level with a TrueSky-Sequence Actor and have the matinee/day manager stuff setup properly in your level. If not it is probably easiest to simply duplicate and rename our Ocean persistent map as a good starting point which can be found here - World'/Game/Maps/SeamlessTest/Ocean.Ocean'

### 3.11.2 TrueSky Actor Properties:

1 - Find and select your TrueSkySequenceActor in the persistent level and look at the properties, you should see the below settings.

**Active Sequences:** This array contains the different truesky cloud sequences that we switch between based on the different weather events. If you are making custom versions of these be sure that they remain in the same order. Regular cloud sequences go in 0, coldfront in 1, heatwave in 2, rain in 3. If the order of these is changed you will get strange irregularities such as rain during a heatwave, or hot weather during a coldfront etc... **Blend Weights:** This array is changed dynamically during gameplay when the different weather events are triggered but it is also a useful way for designers to temporarily visualize their environments with the different weather events, simply change the values here from 0 to 1 to visualize a different weather event, blend weight array numbers correspond to the above active sequences. Only one blend weight should be set to 1 at a time with all others set to 0 to properly visualize the different weather events. **Moon Texture:** Only change this if you want to use your own custom moon textures. **Cosmic Background Texture:** We don't use this in Atlas but it can still be set if you want to get some cool night time sky effects. it is used mostly to mimic the appearance of the milkyway in the sky at night, if you want to see the effect we have a texture that you can plug in and use found here - Texture2D'/Game/PrimalEarth/Environment/Sky/MilkyWay.MilkyWay' **External Scatter RT:** No need to touch this, just needs to be set as shown in the picture. **External Loss RT:** No need to touch this, just needs to be set as shown in the picture. **Brightness:** This is a global brightness setting for the Truesky system, it will brighten all parts of it from the clouds, to sun, to atmospheric. If you want to edit this option you will need to do so in the Matinee\_MasterDayCycle Upon opening the masterdaycycle matinee look for this section. The Brightness for our truesky sequences are set to lerp between different values through out the day so if you want to edit these settings it must be done here in the matinee instead of through the actor.



**Meters Per Unit:** By default this is set to 0.1. To be physically accurate this setting would normally be set to 0.01 as it is setting up how many meters per unreal unit the system uses but on Atlas we use 0.1 to sort of shrink the whole system to make the overall clouds/environment feel a bit more epic. **Cloud Shadow Settings:** The various cloud shadow settings control the distant cloud shadows that appear, in Atlas we mostly keep them more dim and only to appear in the distance but you can change that with these settings if you want! **Visible:** Pretty straight forward, will basically enable/disable the rendering of the truesky system. **Max Sun Radiance:** This setting will basically control how much the sun blooms, we keep this number much lower than default at 500 to keep the sun from taking up too much of the sky and becoming one giant bloomy fireball. **Adjust Sun Radius:** Adjusts the sun radius to be consistent with Radiance, we keep this disabled on Atlas mostly because we get more consistent results with it disabled.

**True Sky**

Active Sequences 4 elements + - 🗑️ 📄

- 0 Atlas\_Temperate\_Clouds ⏪ 🔍 ⏩ 📄
- 1 Atlas\_ColdFront\_Manual ⏪ 🔍 ⏩ 📄
- 2 Atlas\_HeatWave\_Manual ⏪ 🔍 ⏩ 📄
- 3 Atlas\_RainClouds ⏪ 🔍 ⏩ 📄

Blend Weights 4 elements + - 🗑️ 📄

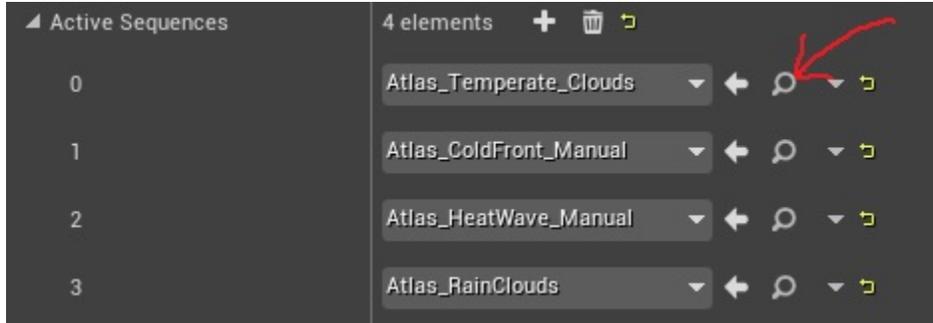
- Moon Texture  Atlas\_Moon ⏪ 🔍 📄
- Cosmic Background Texture  None ⏪ 🔍 📄
- External Scatter RT  TrueSkyScatter ⏪ 🔍 📄
- External Loss RT  TrueSkyLoss ⏪ 🔍 📄
- Brightness 1.942368 📄
- Metres Per Unit 0.1 📄
- Min Simple Cloud Shadowing 0.1 📄
- Max Simple Cloud Shadowing 0.8 📄
- Min Simple Cloud Shadowing Dis 0.01 📄
- Max Simple Cloud Shadowing Di 0.02 📄
- Simple Cloud Shadow Sharpness 1.0 📄
- Visible

**True Sky Rendering| Celestial**

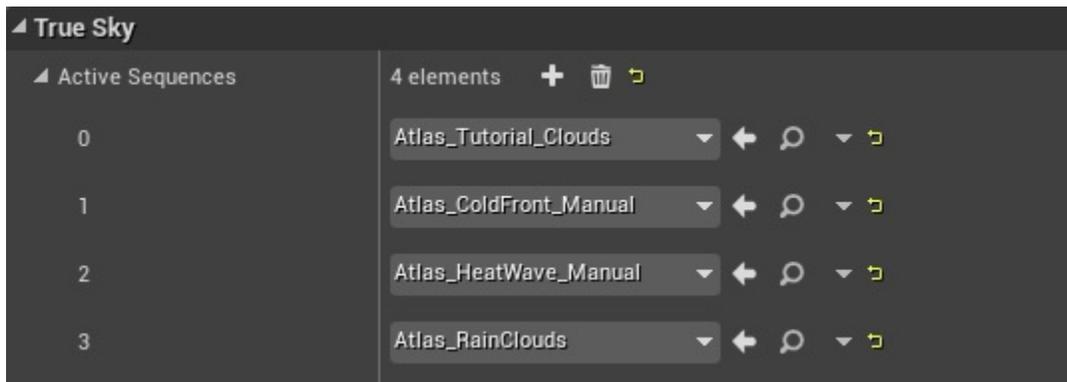
- Max Sun Radiance 500.0 📄
- Adjust Sun Radius

### 3.11.3 Setting Up New Sequences:

1 - With your TrueSkySequenceActor selected go under the Active Sequences. For this tutorial we are just going to create a copy of the regular cloud sequence, so just go to the top cloud sequence here and click the icon in the pic below to take you to the cloud sequences location in the content browser.



2 - In the content browser you are going to want to make a copy of this TrueSky Sequence Asset and put it into whichever folder your project lives in, for the sake of the tutorial I'll be duplicating this and calling it "Atlas\_Tutorial\_Clouds" but you can name it whatever you want. After duplicating it go back to the properties of the TrueSkySequenceActor and plug the new cloud sequence you just made into the 0 position of the array to replace the "Atlas\_Temperate\_Clouds" (If you are making duplicates for any of the specific weather ones you'd want to replace them in the correct positions as they still must remain in the same order to trigger properly) So after plugging this in you should have something similar to the below image and you should see the truesky clouds in your level update.

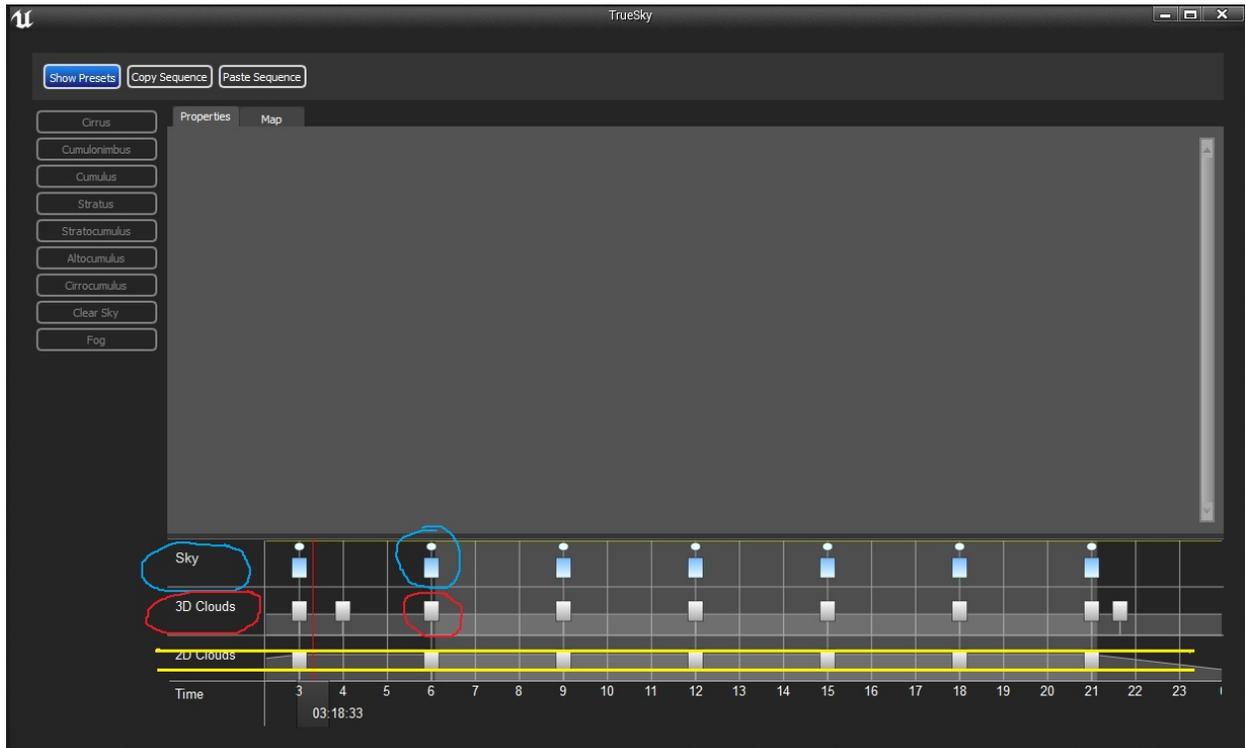


3 - Next you will want to find an actor in the level called "DayCycleManager\_ATLAS" and look in the properties for an option called "Current True Sky Sequence Asset" and also set this to the TrueSky Sequence Asset you created, otherwise the Active Sequence setting you set before will automatically reset itself to the previous cloud sequences everytime you update the time of day in editor/game. You may want to change the time of day in your matinee after doing this so that it is day time again so you can see whats going on in the next few steps!

4 - Now that we have that hooked into the level itself you will want to go back to the content browser and actually open up the new truesky sequence asset that you created earlier and you should get a window that pops up exactly like the below image. This is by far the most complex part of the TrueSky system, it is where all of the customizing of the various cloud sequences happens whether it is rain, heat/cold etc, all of the unique looks are created and maintained in this editor. Overall this system works similarly to matinee in the sense that it uses keyframes based on time of day to determine the look of the clouds and atmospherics. The system is pretty tough to cover so we will have to split it into multiple areas. In the screenshot below you will see words on the left side, Sky circled in blue, and 3DClouds circled in Red. Clicking on these words will pull up one set of menus that cover overall the global settings for both the sky and clouds. These are basically settings that stay persistent no matter what time of day it is.

Clicking on the various keyframes to the right will bring up a different set of menus, ones circled in blue will bring up various keyframe information for the sky, ones circled in red will bring up the keyframe information for the 3DClouds.

I have crossed out the 2DClouds section as these are not used in Atlas and I believe they are hard coded to be disabled even if you try and enable them.



5 - Sky: First click on the word Sky circled in Blue above. You should get a bunch of settings that appear in the large area that was previously grey above. This tutorial won't cover what every single setting in the sequence editor does as most of it is covered in the official TrueSky documentation already, which you can find here - <http://docs.simul.co/unrealengine/Sky.html> Keep in mind however that the documentation will not match up exactly with what you are seeing in our version as we have customized some things and use an older version. You will not need to do any of the custom blueprint setup mentioned on their page.

**Mode:** There is no need to touch any of these settings. **Atmospheric Scattering Tables:** These settings are specifically to handle the overall atmospheric, how foggy things look, color of fog, etc. . . . **The Atmosphere:** These settings may be hard coded to be disabled as they do not appear to do anything. **Sun:** The only thing you will really need to control here is the sun diameter if you desire. The other settings should remain default. **Moon:** Similar to Sun, keep all settings default but feel free to adjust the moon Diameter. **Stars:** Feel free to change any of these settings. The Background setting will only work if you have something plugged into the "Cosmic Background Texture" location on your True Sky Actor.

6 - 3D Clouds: Similar to the Sky settings above, to get these settings you just click the words 3DClouds circled Red in the above picture. To get a bit more information on what a lot of these settings do check the TrueSky documentation again - <http://docs.simul.co/unrealengine/Clouds.html> The settings here are much more technical than the Sky settings and setting wrong values can have a worse effect on performance or cause major issues when transitioning between sequences. Regardless of this though many of the settings found in this window are important in getting nice unique looking clouds. Keep in mind though that whatever settings you use here must remain the same between all of your cloud sequences your level is using or you will get issues with clouds shooting through the sky at high speeds when you try and transition to another weather event.

**Mode:** Similar to the Sky setting these should remain the default settings and do not need to be touched. **Rendering:** The settings found here are more technical ones that deal more with performance stuff, lowering some of these settings may make things look worse but perform better, raise them to get things looking slightly better with worse performance. **Edge Noise/Grid/Generation:** These settings more directly correlate to the actual look/shape of your clouds at a global level and must remain the same between all of your truesky sequences. **Loop Wind Offset:** No

need to touch this, may cause issues!

7 - Sky KeyFrames: Now that the global settings are set you can start getting more into the meat of the TrueSky system. The Slider at the bottom of this will allow you to cycle through the different times of day, each keyframe will set what the sky/clouds are meant to look like at that time of day. To the right of these Sky/3D Clouds buttons you will see several keyframes, to start off we are going to cover the Sky Keyframes in blue. (Very important! If you want to add or remove any keyframes keep in mind that every truesky sequence you make must have the same amount of keyframes at the exact same time of day, if the times between sequences are different or there aren't as many keyframes you will get issues with your clouds racing through the sky when you transition between weather events! For instance, if your regular truesky has keyframes at 2AM, 6AM, and 3PM you must put keyframes at those exact times on all sequences you plan to make.) You can get a little more information about what each setting here does on this page - <http://docs.simul.co/unrealengine/sky.html> Haze/Fog/Mist/Dust: These settings are mostly to control the fog/atmospheric settings, making things more foggy at night/morning etc... Sun and Moon: These settings control where the sun and moons position are in the sky, I wouldn't mess with these unless you need to do something unique with it as it is very easy to break some of this setup and not easy to fix.

8 - Cloud Keyframes: Similar to the Sky Keyframes all of these settings simply control what the clouds look like based on the time of day. These settings are the most important ones in the entirety of the truesky system as far as determining the overall look of your clouds. For a more in depth look at what the different settings do specifically you can find those here - <http://docs.simul.co/unrealengine/Clouds.html>

**Main:** Quick note on the settings found here. Outside of the "Cloudiness" value these values should also remain the same between all of your sequences to avoid getting the racing cloud issue on transitioning cloud sequences. The Cloudiness value can and should still be changed though without issue. **Shape/Lighting/Edge Noise:** The settings here are what will really control the overall look of your clouds. Feel free to change these values to anything you feel looks good, the only exception being the "Wavelength" setting under Edge Noise. This setting specifically will cause the racing cloud effect on cloud transitions. You can still change it to whatever you want but it must remain the same between all of your cloud sequences!

Weather Effects: These ones are rather awesome and will basically allow you to simulate the effect of heavy rain in the distance. Check our Atlas\_RainClouds sequence for an example of this! TrueSkySequenceAs-set'/Game/Atlas/Environment/TrueSky/Atlas\_RainClouds.Atlas\_RainClouds'

### Few hints/things to keep in mind:

**Night lighting:** If you look at any of our sequences you will notice that we raise the direct lighting drastically at night. The reason for this is in our matinee system we are setting the global brightness of the truesky system to be extremely dark, with this the clouds will become almost black otherwise so we artificially increase the cloud lighting to very high values at night to give them a nice rim lighting feeling. **Limit Keyframe time:** If you are adding or moving the keyframe times around be sure to keep them between 0-24 as far as Time goes. Our system basically replays the same day over and over again so any keyframes placed after 24 or before 0 will not be used and will probably cause strange issues in transitioning into the next day. **Setting multiple Keyframes:** When your setting up the look for your clouds it is usually easiest to drag and select all cloud keyframes at once to adjust them instead of doing them one by one. **Using the Time Slider:** If your adjusting the time of day using the time slider it will unfortunately only adjust your truesky time and not the actual in world time/sun position etc so you may get wrong visuals such as the sun being out at night. You will still need to open up the matinee and adjust the time of day there to properly visualize your stuff at different times of day.

## 3.12 Trunk Detail Material

### 3.12.1 Overview

This page describes the general workflow in using the trunk detail material for Atlas. The material will create a mask based on the U dimension so we can isolate the trunks of a particular tree. This isolated section will then use a series of tiled textures blended together using a mask texture that is unique for each tree.

### 3.12.2 Reference

MaterialInstanceConstant'/Game/Atlas/CoreMaterials/MasterMaterials/MIC/MM\_Foliage\_Leaves\_DetailBlend\_MIC.MM\_Foliage\_Leaves\_DetailBlend\_MIC

### 3.12.3 MIC Parameters

#### Global Detail

**Detail Weight:** Controls the amount the CNR textures influence the unique texture. This affects the normal map as well. Generally this should be left at 3 and should not be changed. **DetailMask:** The grayscale texture that will be used for blending the three CNR textures. Black will be layer-0, grey will be layer-1, and white will be layer-2. **DetailMask Debug:** Enables a visualizer with RGB colors that shows you how the CNR textures are being used. Useful to also see when the original composite texture lerp in. **LightVectorMask Debug:** Another debug visualizer that shows a mask based on sun vector. This is used for reducing the CNR normal intensity on areas facing the sun. **Sunside Detail Normal Intensity:** This will reduce the CNR normal intensity on parts of the trunk that are facing the direction of the sun light. Creates a more pleasing result. In general you should not have to change the default float of 0.5. **Trunk Distance Factor-Far:** Controls when the CNR texture contribution fades out and the material just uses the composited texture from SpeedTree. Makes sure that it blends well with the billboards. **Trunk Distance Factor-Near:** Controls when the CNR texture contribution fades out and the material just uses the composited texture from SpeedTree. Makes sure that it blends well with the billboards. **Trunk U Max:** This controls the lower U value for the trunk UVs. It's also used for masking the trunk off so that the CNR textures are only applied to the trunk. **Trunk U Min:** This controls the lower U value for the trunk UVs. It's also used for masking the trunk off so that the CNR textures are only applied to the trunk. **Use Custom Trunk Mask:** If you need to extend the CNR texture contribution to branches, etc set this to True. It will then allow you to specify a custom mask. Do not change Trunk U Max and Trunk U Min. Those will still be used to handle seamless texture tiling along the trunk, **UV Tiling Override:** When set to False the CNR texturing will all be done in the material. Probably Will be best to leave it as True for now.

#### Detail-[0-2]

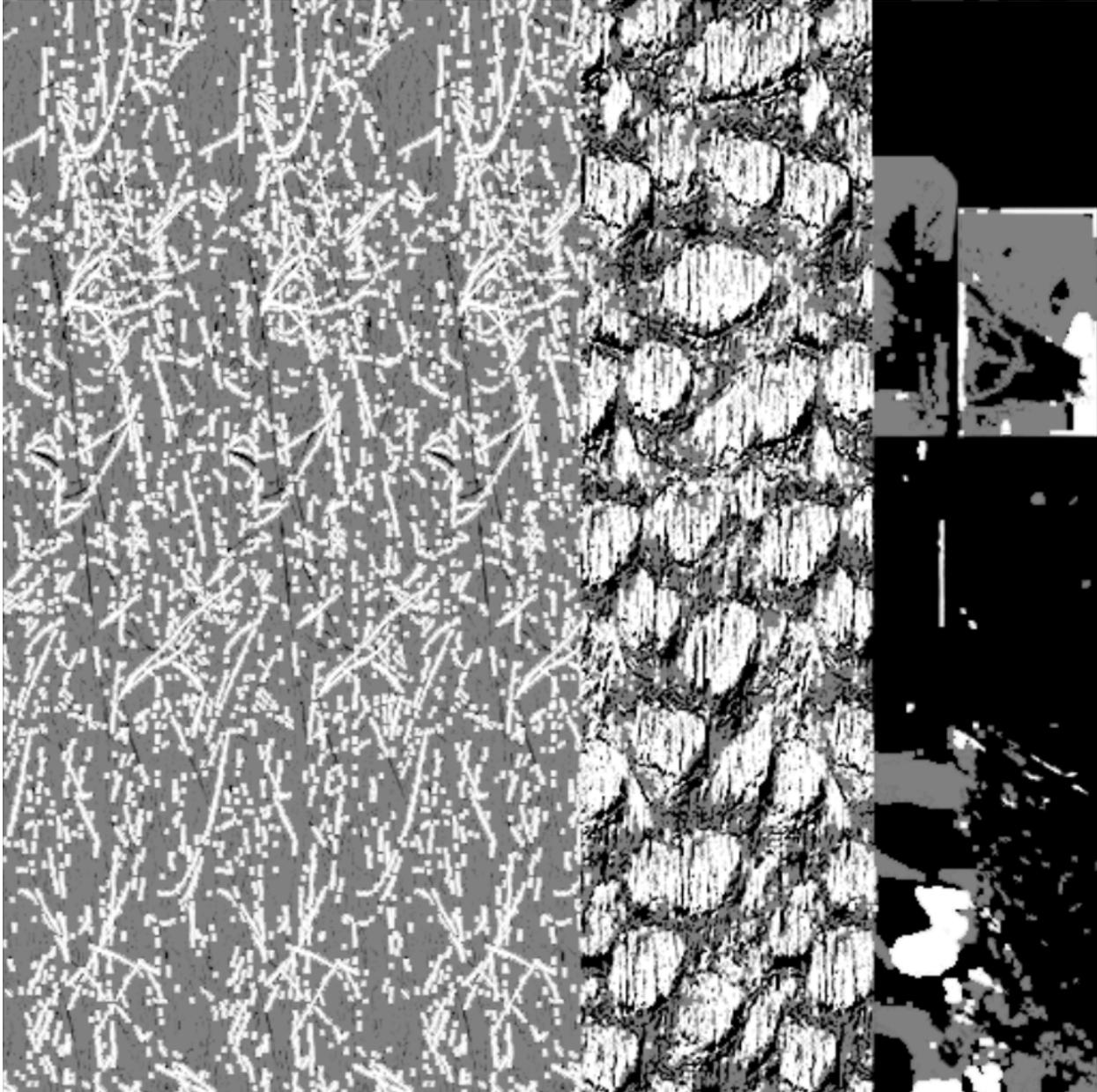
**Detail-[0-2]:** CNR texture for detail. Info on creating them can be found here. **Detail-[0-2] UV Tiling:** Controls the UV tiling of the CNR textures. Defaults to 5 for both U and V. U always rounds up to the nearest integer to preserve texture tiling across the trunk. **Detail-[0-2] Tile Y Ratio:** Added in case needed for non square textures. **Detail-[0-2] Angel:** Allows you to rotate the CNR texture. **Detail-[0-2] Normal Intensity:** Controls the intensity of the CNR texture. **Detail-[0-2] Roughness - Max:** Max roughness for CNR texture. Defaults to 0.5. This blends with the composite texture from SpeedTree. Usually you will want to leave the default value. **Detail-[0-2] Roughness - Min:** Min roughness for CNR texture. Defaults to 0. This blends with the composite texture from SpeedTree. Usually you will want to leave the default value.

#### Asset Requirements

There are essentially three things that will be needed for an asset that uses this material. You will need a texture mask which controls where each CNR texture is blended, appropriate CNR textures (These can be shared so as time goes

on we should need less of these), and the precise U coordinates for the extreme edges of the trunk UVs. These are the parameters Trunk U Max and Trunk U Min under Global Detail.

### DetailMask



Texture mask Texture mask built off of the composite texture from SpeedTree. Depending on the Trunk U Max and Trunk U Min settings not all of the mask might be used. You will also want to make sure it tiles across the trunk.

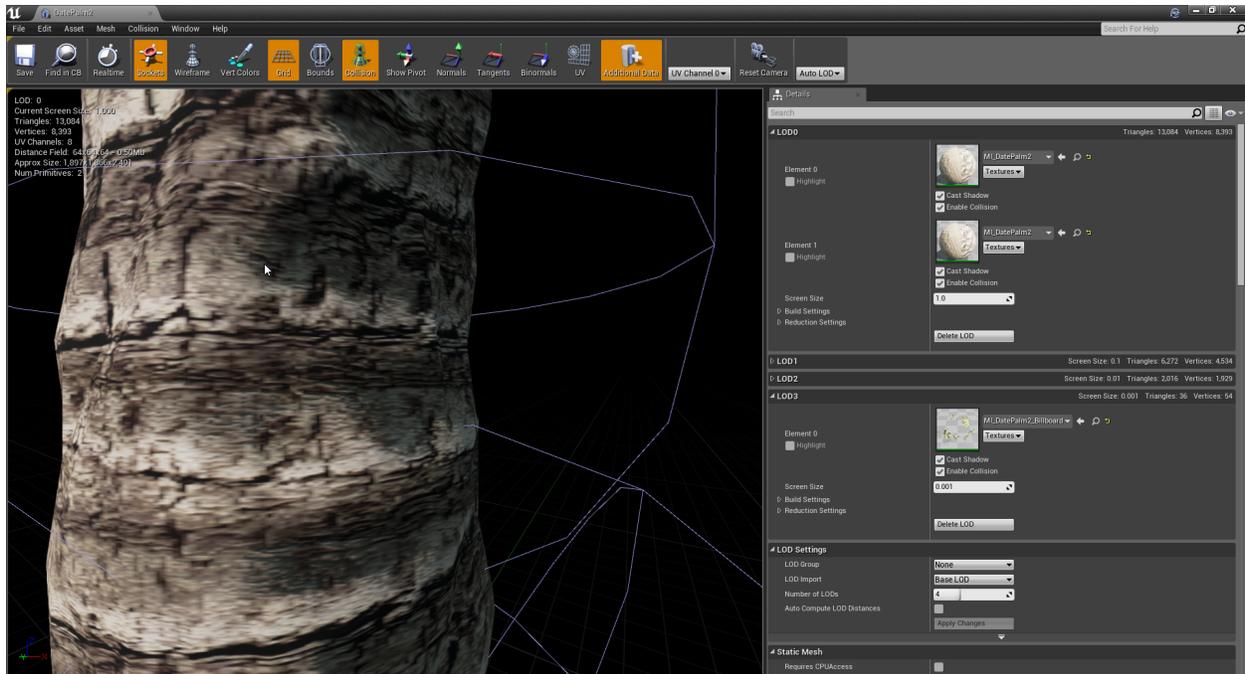
### CNR Textures

Shared CNR textures should be imported here in engine: `Game/Atlas/Environment/Shared/TilingTextures/Detail/` Info on creating CNR textures can be found here. They usually work best when the red channel (Color ) has a lot of contrast.

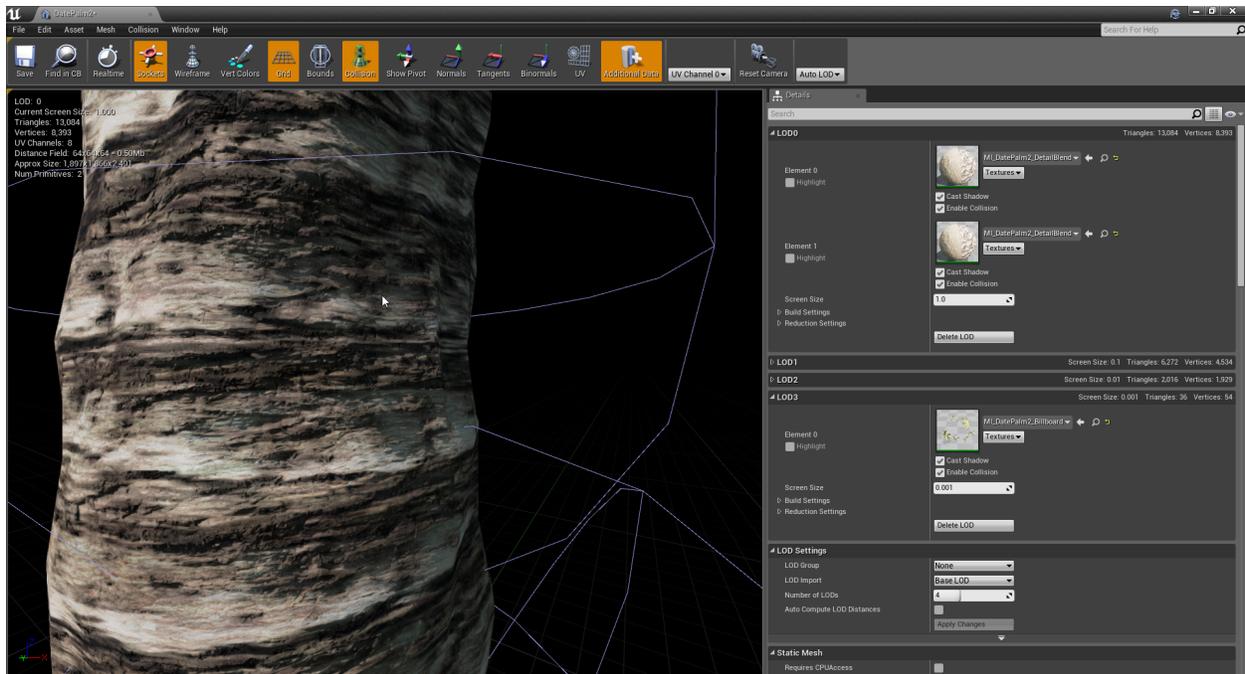
## Trunk U Min/Max

You will need to export a tree out of Unreal and import in Maya. Once in Maya open the UV editor, select one of the far left vertices for the trunk and enter the following mel command: `polyEditUV -q`. This will output the UV coordinates for that vert to the log window. You only need the U coordinate. Enter this float as Trunk U Min. Select one of the far right verts of the trunk and run the command again. Enter this U value for Trunk U Max. Make sure to enter the full number. You will want the precision!

## Result



Before



After

## 3.13 Landscape

### 3.13.1 Overview

In general you enter the constants below for the landscapes. The only thing that potentially changes is the size of the landscape which we control by the number of components. Non square landscapes are fine. Landscapes are scaled 50% on x and y to get more verts per surface area.

### 3.13.2 Unreal Engine

#### Constants

1. Location = {0, 0, 0}
2. Rotation = {0, 0, 0}
3. Scale = {50, 50, 100}
4. Section Size = 255x255
5. Sections Per Component = 2x2 Sections
6. If importing procedural layer weights: Use MI\_Landscape\_Tropical for material.
7. Material will be changed out depending on island.
8. Should use the same landscape layer infos in Atlas/Environment/Landscape so clutter settings match across multiple islands.

### 3.13.3 World Machine

We typically use World Machine and Geoglyph 2 for landscapes. We might use more Houdini in this area in the future but the erosion in WM seems superior. The bottom of the landscapes need to reach -20,000 uu while in engine to meet up with the procedural ocean floor. In WM you will want to have the edges of the landscape to go down to 287 m (When using default world settings). If you do this the landscape will go a touch below the -20,000 uu requirement.

### 3.13.4 Layer Painting

Will fill in later. After importing layer weights the following console command will need to be run so the landscape has correct physical materials: `FixupLandscapeCollisions`.

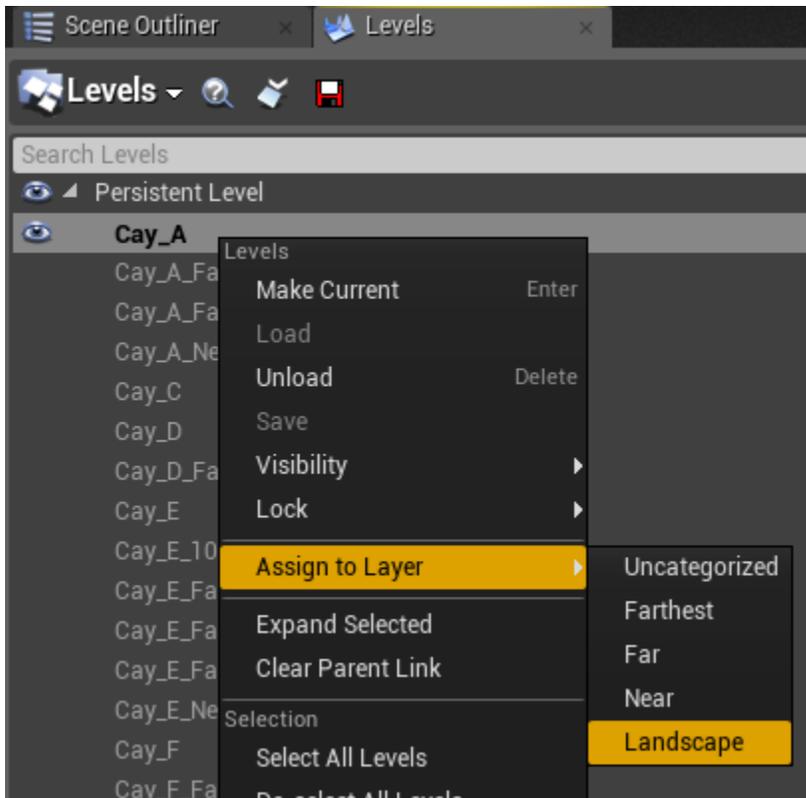
## 3.14 Level LOD

### 3.14.1 Overview

Landscapes take up a lot of memory and are expensive to render. In ATLAS, every Landscape sub-level should switch to a staticmesh version via the Level LOD system. This page covers how to generate and configure Landscape Level LODs using the built-in Simplygon functionality available in World Composition.

### 3.14.2 Steps

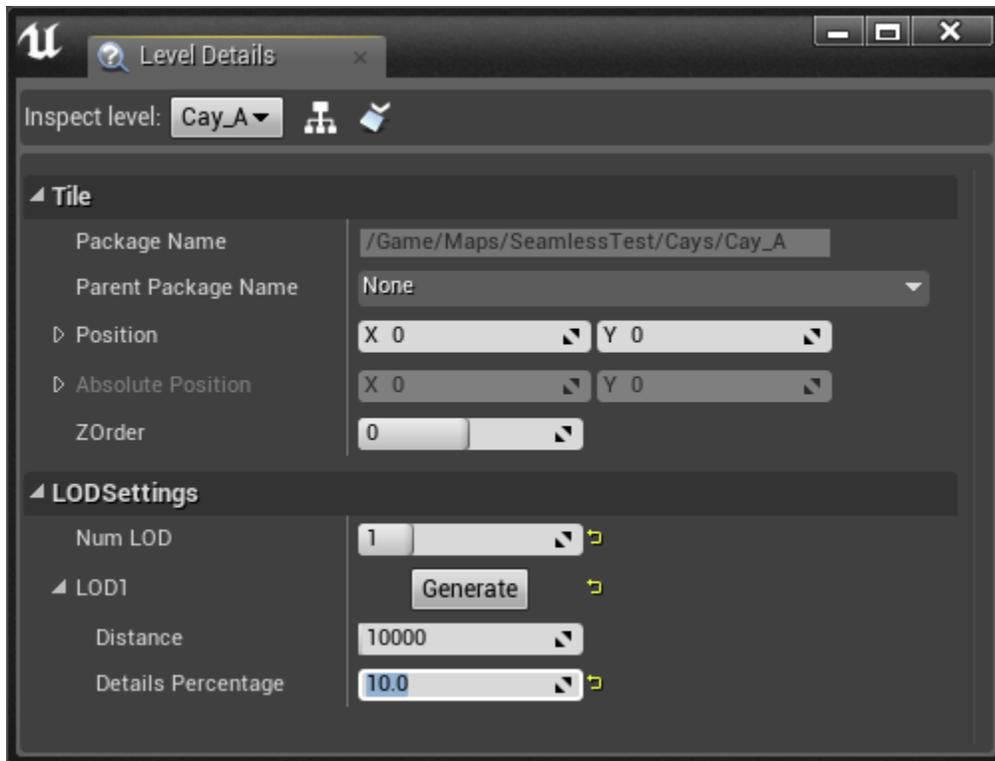
- **Temporarily delete BP\_Ocean, as it will be baked into your StaticMesh landscape textures if left visible.**
- Load the island Landscape sub-level you are interested in. For example, *Cay\_A*.
- Right-click on the name of your level in the **Level Panel** and assign it to the Landscape layer.



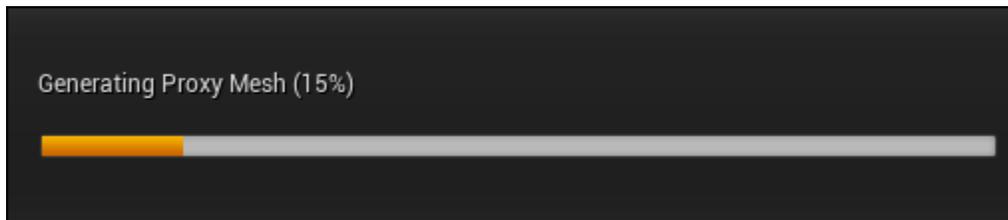
- Open **Level Details** from the **Level Panel**.



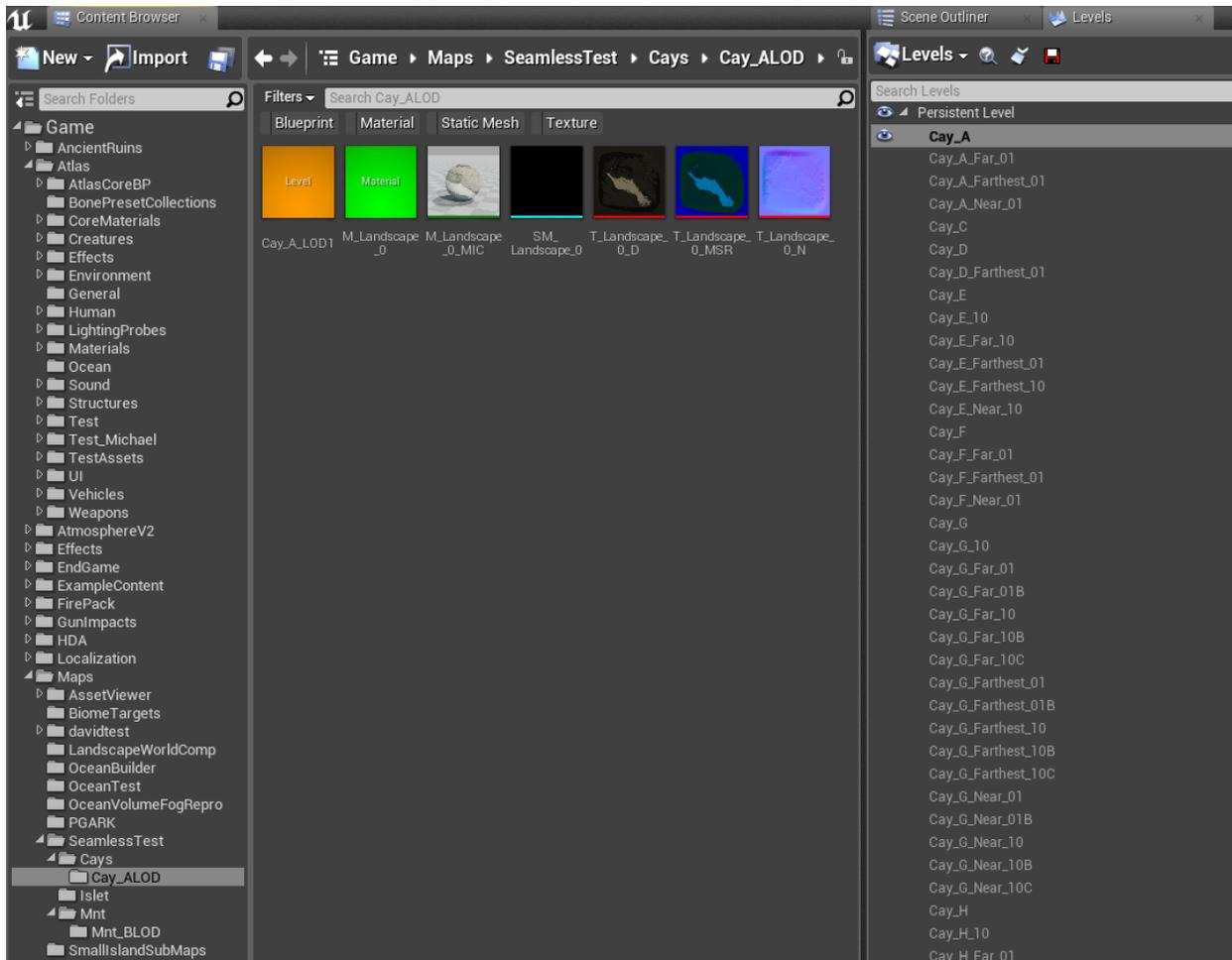
- Set **Num LOD = 1** and **Details Percentage = 10**. ( *Note: Distance is automatically calculated and the number here is meaningless* ).



- Click **Generate** and grab some coffee or take a healthy walk break.



- You should now see a bunch of level-named assets in **Content Browser**.



- Save your sub-level.

## 3.15 Map Check

### 3.15.1 Overview

Below is an overview of mapcheck errors and what to do when you spot them.

### 3.15.2 Errors

**Warning** [Asset\_Name] **Instance has** [Number above 25,000] **in** [Name of map] **elements and is networked.**

- The above warning displays when you have a mesh used as foliage more than 25,000 times. Foliage can hold an arbitrary number of foliage meshes but each one of those meshes cannot be used more than 25,000 times for performance reasons. To fix this issue run the `reducefoliage reduceby=[x]` console command in the editor where x is a float value between 0-1 (1 would be no change as it equals 100%).

**Info** [Asset\_Name] **Per tri collision on Static mesh has alot of triangles** [Number above 2,000].

- If there is a static in the level that has per triangle collision above 2,000 triangles this map check error will appear. This is a bit of a soft rule now and we do go over 2,000 if the asset truly needs it. To fix you simply open the static mesh and lower the resolution of the last collision lod.

**Warning** [Asset\_Name] **Static mesh actor has NULL StaticMesh property**

- This happens when an actor placed in a level no longer references a static mesh. The vast majority of the time these actors can just be deleted to solve the problem.

**Warning** [Asset\_Name] **\_settings Large Cull Distance for Foliage.**

- This warning appears when an asset used as foliage has a very large cull distance in its settings. If this asset is intended to have these values (Trees, large rocks, etc) you can disable this warning from popping up for a particular asset by setting the bool Map Check Ignore Large Cull Distance to true in its foliage settings.

**Info** [Asset\_Name] **Proxy Mesh is Dirty!**

- This warning displays when a proxy mesh parent no longer matches its children. This can happen when a parameter is changed on a proxy mesh child or it has been moved. To fix you can simply right click the asset and regenerate the proxy mesh.

**Info** [Asset\_Name] **Child Proxy with no parent found. (possibly level not loaded)**

- This warning appears when the editor sees a proxy child in a level but cannot find the parent. This typically happens when the level that has the proxy parent actor is not loaded. Simply loading the level and running map check again should resolve the issue. The proxy parent accidentally getting deleted can cause the issue as well. If this is the case you will need to clear the proxy mesh child data and regenerate.

**Info** [Asset\_Name] **Foliage Mesh with lowest LOD tri count above 500** [Triangle count of last lod] . ([x] instances)

- The last lod for foliage should not be higher than 500 triangles. If it is higher this warning will display in map check. To fix simply optimize the last asset. Keep in mind that this can sometimes be a false positive. If the asset uses expensive per triangle collision for the last lod this will show up. Perhaps a setting can be added in the future that disables this warning for assets that have per triangle collision enabled.

**Info** [Asset\_Name] **Foliage Mesh with lowest LOD at tiny screen size** [Some small number] . ([x] instances)

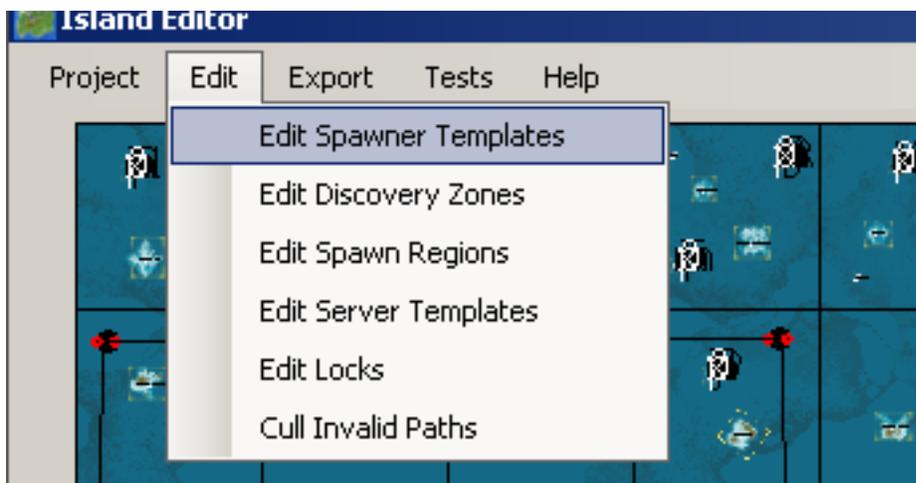
- If a mesh has a very small screen size for the last lod this warning will appear. Sometimes it's hard to avoid this though so we usually have many of these on ATLAS.

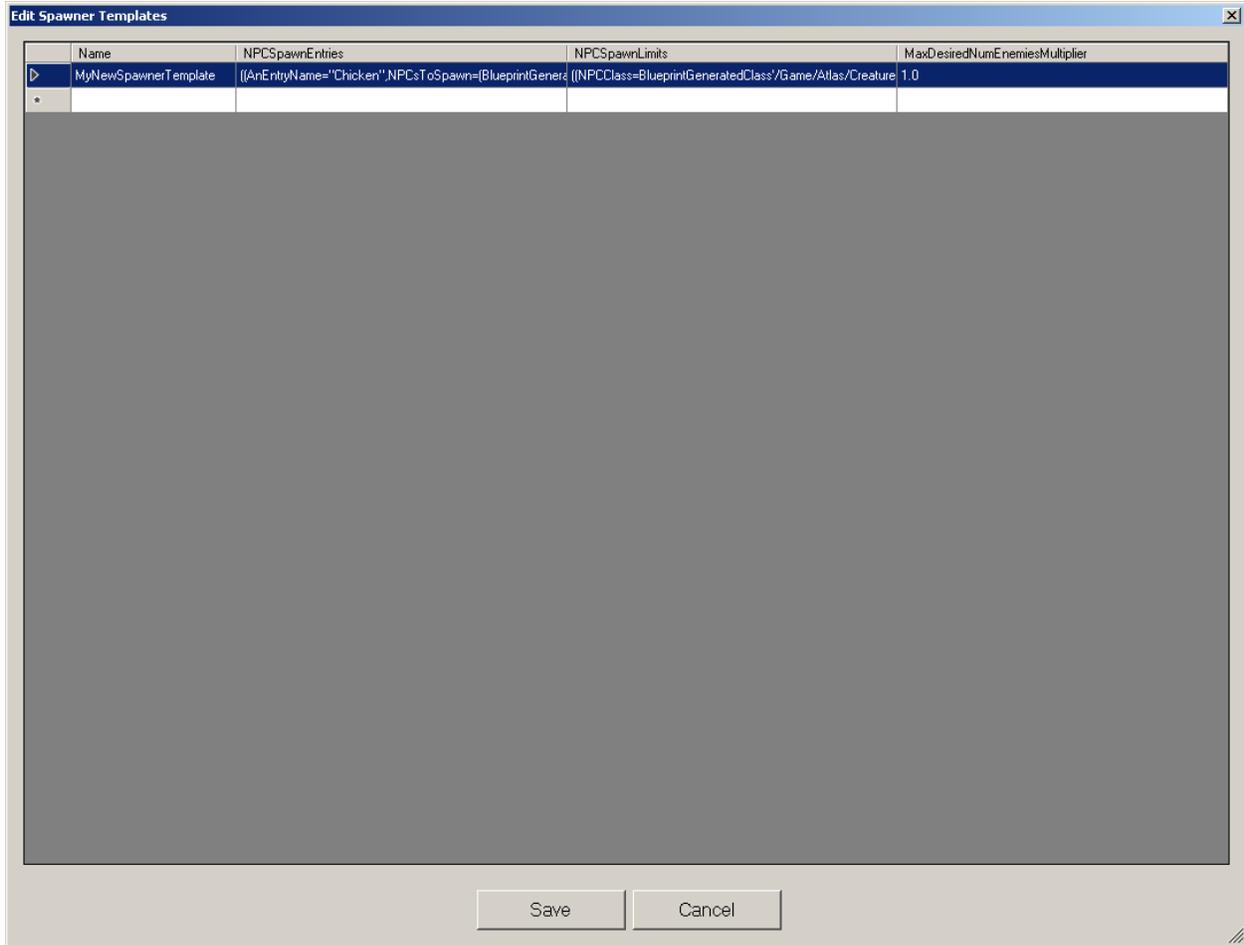


### 4.1 Overriding Spawns

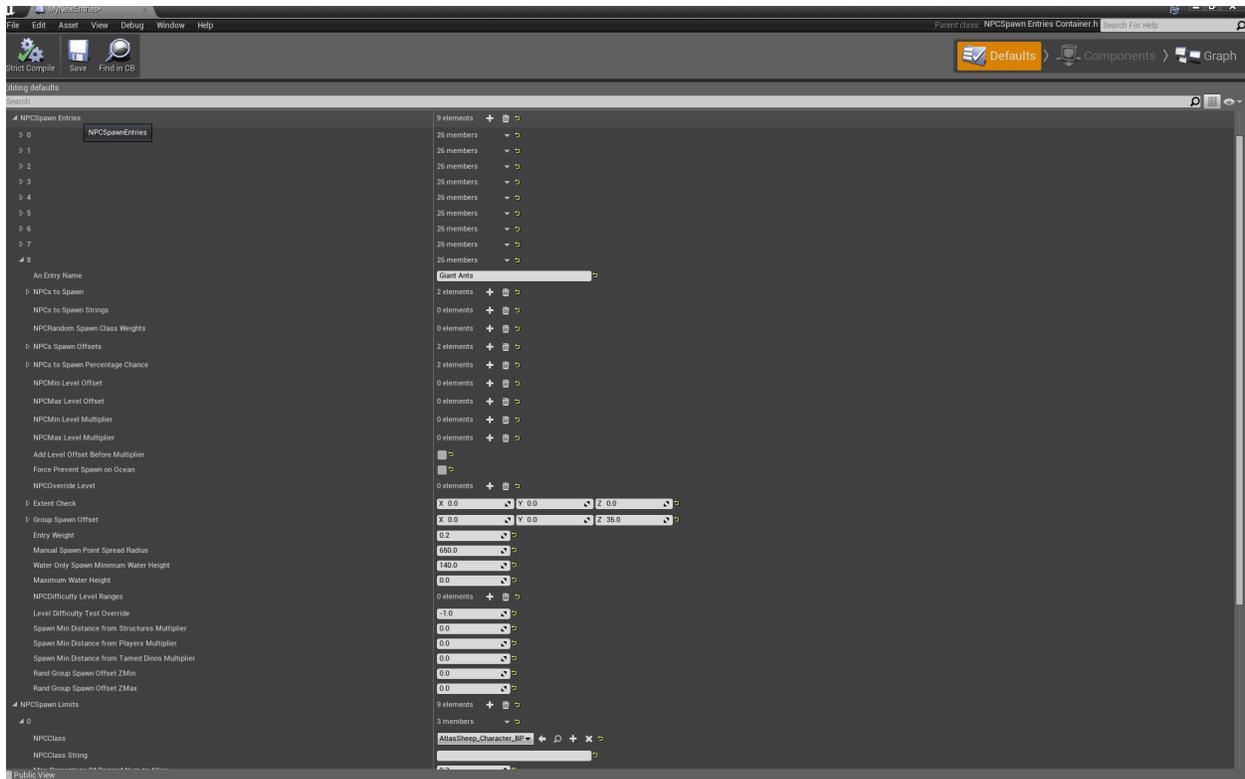
Having the DevKit is VERY helpful to this (to copy/paste the spawn entries arrays and lookup the Blueprint paths), but not explicitly required:

In the severgrid editor you can add new Spawner Templates.



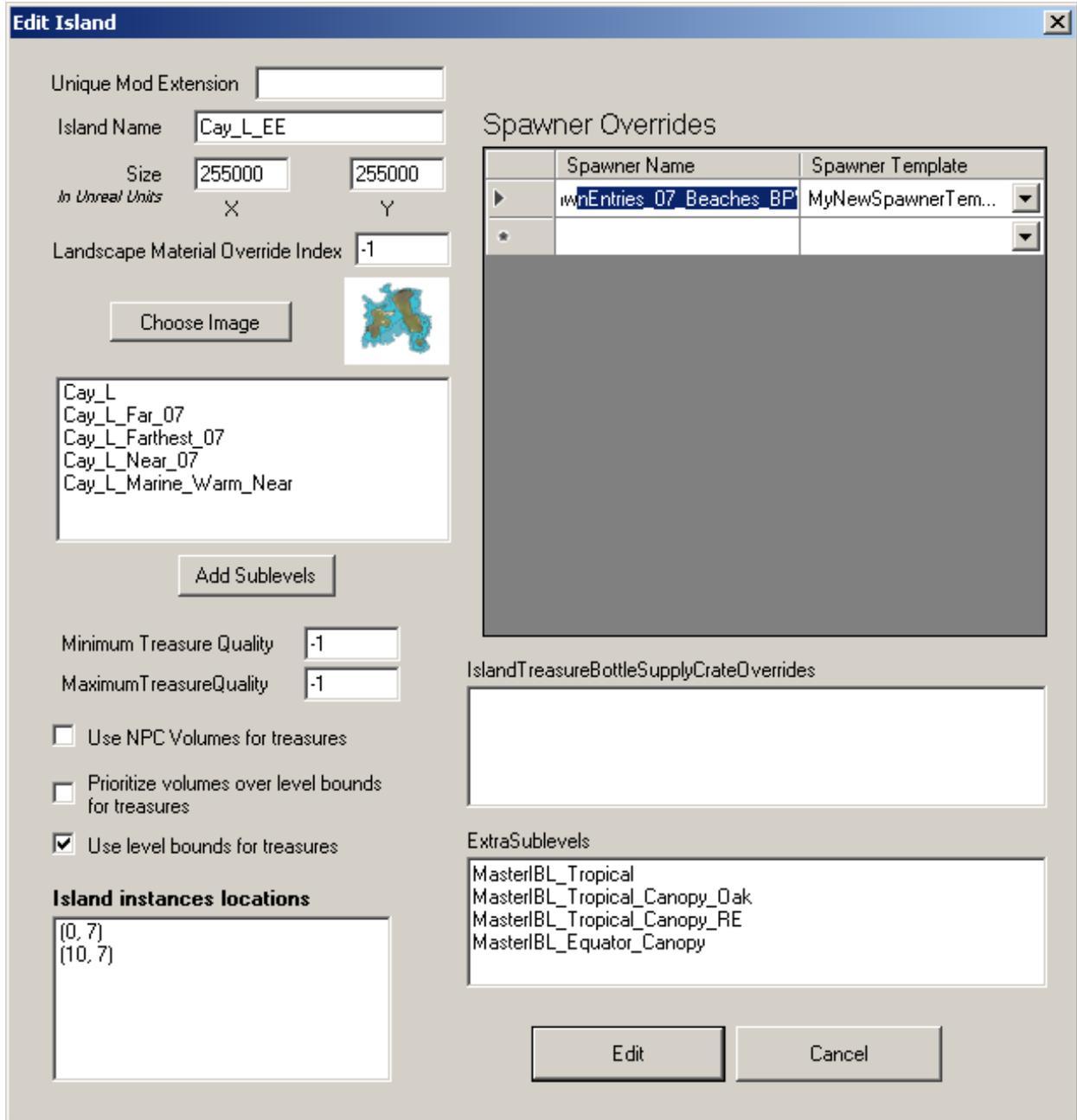


The names of your new Spawner Templates should have no spaces or special characters, but is otherwise arbitrary. The NPCSpawnEntries and NPCSpawnLimits should be copy/pasted values from these two arrays within the DevKit (of an NPCSpawnEntires Blueprint), set them up to have whatever values you want to use for your override; **however**, since they're just plain-text, a careful end-user could theoretically do those without the DevKit provided they know and use the correct Blueprint paths for the creatures. MaxDesiredNumEnemiesMultiplier should be 1.0 or whatever scale of the number of enemies you want (scales by however many the spawner you're going to override would already have).



Now, to actually override the Spawners on an Island, using our new Spawner Template.

In the island template, input (copy/paste) from the editor the Blueprint names of the spawn entries you wish to override (must be spawn entries in use on that island), and select your custom Spawner Template that you wish to override it with, like so:



The challenge, of course, is knowing which Spawn Entries Blueprints are in use on which islands. That's most easily done from the DevKit by doing "Show References" on the Spawn Entries to see which actual island maps they're used on. Or experiment. Alternatively, you can override the spawners on a specific instance of the Island within the Server Grid, not just the Island Template (the specific instance overrides take precedence), like so: "Ctrl-LeftClick" on an island instance in the Server Grid Editor (must have Unlocked Islands in the Grid Editor), and then input the Spawn Entries BP name and select the override name same as can be done on the Island Template:

**Edit Island Instance**
✕

spawnPointRegionOverride:

finalNPCLevelMultiplier:

finalNPCLevelOffset:

instanceTreasureQualityMultiplier:

instanceTreasureQualityAddition:

IslandInstanceCustomData1:

IslandInstanceCustomData2:

IslandInstanceClientCustomData1:

IslandInstanceClientCustomData2:

### Spawner Overrides

	Spawner Name	Spawner Template
▶	SpawnEntries_07_Beaches	MyNewSpawnerTem... ▼
*		▼

Save

Cancel

That's all there is to it! For reference, here's all the Creature BP's and Spawn Entries BP's currently in ATLAS (though as mentioned, determining which Spawn Entries are used in which Island is easiest done via the DevKit, and easiest to use the DevKit to generate the NPCSpawnEntries and NPCSpawnLimits strings via copy/paste):

```
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/AoD/AoDSpawnEntries_Cave.'
↳AoDSpawnEntries_Cave'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/AoD/AoDSpawnEntries_Inland.'
↳AoDSpawnEntries_Inland'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/AoD/AoDSpawnEntries_NearBeach.'
↳AoDSpawnEntries_NearBeach'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Caves/_01/CaveSpawnEntries_01_PVE.'
↳CaveSpawnEntries_01_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Caves/_02/CaveSpawnEntries_02_PVE.'
↳CaveSpawnEntries_02_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Caves/_04/CaveSpawnEntries_04_PVE.'
↳CaveSpawnEntries_04_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Caves/_06/CaveSpawnEntries_06_PVE.'
↳CaveSpawnEntries_06_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Caves/_07/CaveSpawnEntries_07.'
↳CaveSpawnEntries_07'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Caves/_09/CaveSpawnEntries_09.'
↳CaveSpawnEntries_09'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Caves/_10/CaveSpawnEntries_10_PVE.'
↳CaveSpawnEntries_10_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Caves/CaveSpawnManager_10.'
↳CaveSpawnManager_10'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/01_NE_Tropical/CreatureSpawnEntries_01_
↳Beaches.CreatureSpawnEntries_01_Beaches'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/01_NE_Tropical/CreatureSpawnEntries_01_
↳Beaches_Mnt.CreatureSpawnEntries_01_Beaches_Mnt'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_01/CreatureSpawnEntries_01_
↳Beaches_NoCarnos.CreatureSpawnEntries_01_Beaches_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/01_NE_Tropical/_PVE/
↳CreatureSpawnEntries_01_Beaches_PVE.CreatureSpawnEntries_01_Beaches_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/01_NE_Tropical/CreatureSpawnEntries_01_
↳Forest.CreatureSpawnEntries_01_Forest'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/01_NE_Tropical/CreatureSpawnEntries_01_
↳Forest_Mnt.CreatureSpawnEntries_01_Forest_Mnt'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_01/CreatureSpawnEntries_01_
↳Forest_NoCarnos.CreatureSpawnEntries_01_Forest_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/01_NE_Tropical/_PVE/
↳CreatureSpawnEntries_01_Forest_PVE.CreatureSpawnEntries_01_Forest_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/01_NE_Tropical/_PVE/
↳CreatureSpawnEntries_01_FreshWater_PVE.CreatureSpawnEntries_01_FreshWater_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/01_NE_Tropical/CreatureSpawnEntries_01_
↳Inland.CreatureSpawnEntries_01_Inland'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_01/CreatureSpawnEntries_01_
↳Inland_NoCarnos.CreatureSpawnEntries_01_Inland_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/01_NE_Tropical/_PVE/
↳CreatureSpawnEntries_01_Inland_PVE.CreatureSpawnEntries_01_Inland_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/01_NE_Tropical/CreatureSpawnEntries_01_
↳IslandWater.CreatureSpawnEntries_01_IslandWater'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_01/CreatureSpawnEntries_01_
↳IslandWater_NoCarnos.CreatureSpawnEntries_01_IslandWater_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/01_NE_Tropical/CreatureSpawnEntries_01_
↳Mountain.CreatureSpawnEntries_01_Mountain'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/01_NE_Tropical/CreatureSpawnEntries_01_
↳Mountain_Mnt.CreatureSpawnEntries_01_Mountain_Mnt'
```

(continues on next page)

(continued from previous page)

```

Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_01/CreatureSpawnEntries_01_
↳Mountain_NoCarnos.CreatureSpawnEntries_01_Mountain_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/01_NE_Tropical/_PVE/
↳CreatureSpawnEntries_01_Mountain_PVE.CreatureSpawnEntries_01_Mountain_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/02_NW_Temperate/CreatureSpawnEntries_
↳02_Beaches.CreatureSpawnEntries_02_Beaches'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_02/CreatureSpawnEntries_02_
↳Beaches_NoCarnos.CreatureSpawnEntries_02_Beaches_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/02_NW_Temperate/_PVE/
↳CreatureSpawnEntries_02_Beaches_PVE.CreatureSpawnEntries_02_Beaches_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/02_NW_Temperate/CreatureSpawnEntries_
↳02_Forest.CreatureSpawnEntries_02_Forest'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_02/CreatureSpawnEntries_02_
↳Forest_NoCarnos.CreatureSpawnEntries_02_Forest_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/02_NW_Temperate/_PVE/
↳CreatureSpawnEntries_02_Forest_PVE.CreatureSpawnEntries_02_Forest_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/02_NW_Temperate/CreatureSpawnEntries_
↳02_FreshWater.CreatureSpawnEntries_02_FreshWater'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_02/CreatureSpawnEntries_02_
↳FreshWater_NoCarnos.CreatureSpawnEntries_02_FreshWater_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/02_NW_Temperate/_PVE/
↳CreatureSpawnEntries_02_FreshWater_PVE.CreatureSpawnEntries_02_FreshWater_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/02_NW_Temperate/CreatureSpawnEntries_
↳02_InlandPlains.CreatureSpawnEntries_02_InlandPlains'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_02/CreatureSpawnEntries_02_
↳InlandPlains_NoCarnos.CreatureSpawnEntries_02_InlandPlains_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/02_NW_Temperate/_PVE/
↳CreatureSpawnEntries_02_InlandPlains_PVE.CreatureSpawnEntries_02_InlandPlains_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/02_NW_Temperate/CreatureSpawnEntries_
↳02_Mountains.CreatureSpawnEntries_02_Mountains'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_02/CreatureSpawnEntries_02_
↳Mountains_NoCarnos.CreatureSpawnEntries_02_Mountains_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/02_NW_Temperate/_PVE/
↳CreatureSpawnEntries_02_Mountains_PVE.CreatureSpawnEntries_02_Mountains_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/03_E_Temperate/CreatureSpawnEntries_03_
↳Beaches.CreatureSpawnEntries_03_Beaches'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_03/CreatureSpawnEntries_03_
↳Beaches_NoCarnos.CreatureSpawnEntries_03_Beaches_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/03_E_Temperate/_PVE/
↳CreatureSpawnEntries_03_Beaches_PVE.CreatureSpawnEntries_03_Beaches_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/03_E_Temperate/CreatureSpawnEntries_03_
↳Forest.CreatureSpawnEntries_03_Forest'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_03/CreatureSpawnEntries_03_
↳Forest_NoCarnos.CreatureSpawnEntries_03_Forest_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/03_E_Temperate/_PVE/
↳CreatureSpawnEntries_03_Forest_PVE.CreatureSpawnEntries_03_Forest_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/03_E_Temperate/CreatureSpawnEntries_03_
↳FreshWater.CreatureSpawnEntries_03_FreshWater'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_03/CreatureSpawnEntries_03_
↳FreshWater_NoCarnos.CreatureSpawnEntries_03_FreshWater_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/03_E_Temperate/_PVE/
↳CreatureSpawnEntries_03_FreshWater_PVE.CreatureSpawnEntries_03_FreshWater_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/03_E_Temperate/CreatureSpawnEntries_03_
↳Inland.CreatureSpawnEntries_03_Inland'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_03/CreatureSpawnEntries_03_
↳Inland_NoCarnos.CreatureSpawnEntries_03_Inland_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/03_E_Temperate/_PVE/
↳CreatureSpawnEntries_03_InlandPlains_PVE.CreatureSpawnEntries_03_InlandPlains_PVE'

```

(continues on next page)

(continued from previous page)

```

Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/03_E_Temperate/CreatureSpawnEntries_03_
↳Mountains.CreatureSpawnEntries_03_Mountains'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_03/CreatureSpawnEntries_03_
↳Mountains_NoCarnos.CreatureSpawnEntries_03_Mountains_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/03_E_Temperate/_PVE/
↳CreatureSpawnEntries_03_Mountains_PVE.CreatureSpawnEntries_03_Mountains_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/04_W_Tropical/CreatureSpawnEntries_04_
↳Beaches.CreatureSpawnEntries_04_Beaches'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_04/CreatureSpawnEntries_04_
↳Beaches_NoCarnos.CreatureSpawnEntries_04_Beaches_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/04_W_Tropical/_PVE/
↳CreatureSpawnEntries_04_Beaches_PVE.CreatureSpawnEntries_04_Beaches_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/04_W_Tropical/CreatureSpawnEntries_04_
↳Forest.CreatureSpawnEntries_04_Forest'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/04_W_Tropical/_PVE/
↳CreatureSpawnEntries_04_Forest_PVE.CreatureSpawnEntries_04_Forest_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/04_W_Tropical/CreatureSpawnEntries_04_
↳Inland.CreatureSpawnEntries_04_Inland'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_04/CreatureSpawnEntries_04_
↳Inland_NoCarnos.CreatureSpawnEntries_04_Inland_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/04_W_Tropical/_PVE/
↳CreatureSpawnEntries_04_Inland_PVE.CreatureSpawnEntries_04_Inland_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/04_W_Tropical/CreatureSpawnEntries_04_
↳IslandWater.CreatureSpawnEntries_04_IslandWater'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/04_W_Tropical/_PVE/
↳CreatureSpawnEntries_04_IslandWater_PVE.CreatureSpawnEntries_04_IslandWater_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/04_W_Tropical/CreatureSpawnEntries_04_
↳Mountain.CreatureSpawnEntries_04_Mountain'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/04_W_Tropical/_PVE/
↳CreatureSpawnEntries_04_Mountain_PVE.CreatureSpawnEntries_04_Mountain_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/06_W_Tundra/CreatureSpawnEntries_06_
↳Beaches.CreatureSpawnEntries_06_Beaches'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/06_W_Tundra/_PVE/CreatureSpawnEntries_
↳06_Beaches_PVE.CreatureSpawnEntries_06_Beaches_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/06_W_Tundra/CreatureSpawnEntries_06_
↳Forest.CreatureSpawnEntries_06_Forest'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/06_W_Tundra/_PVE/CreatureSpawnEntries_
↳06_Forest_PVE.CreatureSpawnEntries_06_Forest_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/06_W_Tundra/CreatureSpawnEntries_06_
↳FreshWater.CreatureSpawnEntries_06_FreshWater'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/06_W_Tundra/_PVE/CreatureSpawnEntries_
↳06_FreshWater_PVE.CreatureSpawnEntries_06_FreshWater_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/06_W_Tundra/CreatureSpawnEntries_06_
↳Inland.CreatureSpawnEntries_06_Inland'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/06_W_Tundra/_PVE/CreatureSpawnEntries_
↳06_Inland_PVE.CreatureSpawnEntries_06_Inland_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/06_W_Tundra/CreatureSpawnEntries_06_
↳Mountains.CreatureSpawnEntries_06_Mountains'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/06_W_Tundra/_PVE/CreatureSpawnEntries_
↳06_Mountains_PVE.CreatureSpawnEntries_06_Mountains_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/07_Equator/CreatureSpawnEntries_07_
↳Beaches.CreatureSpawnEntries_07_Beaches'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_07/CreatureSpawnEntries_07_
↳Beaches_NoCarnos.CreatureSpawnEntries_07_Beaches_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/07_Equator/_PVE/CreatureSpawnEntries_
↳07_Beaches_PVE.CreatureSpawnEntries_07_Beaches_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/07_Equator/CreatureSpawnEntries_07_
↳Forest.CreatureSpawnEntries_07_Forest'

```

(continues on next page)

(continued from previous page)

```

Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_07/CreatureSpawnEntries_07_
↳Forest_NoCarnos.CreatureSpawnEntries_07_Forest_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/07_Equator/_PVE/CreatureSpawnEntries_
↳07_Forest_PVE.CreatureSpawnEntries_07_Forest_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/07_Equator/CreatureSpawnEntries_07_
↳FreshWater.CreatureSpawnEntries_07_FreshWater'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_07/CreatureSpawnEntries_07_
↳FreshWater_NoCarnos.CreatureSpawnEntries_07_FreshWater_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/07_Equator/_PVE/CreatureSpawnEntries_
↳07_FreshWater_PVE.CreatureSpawnEntries_07_FreshWater_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/07_Equator/CreatureSpawnEntries_07_
↳Inland.CreatureSpawnEntries_07_Inland'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_07/CreatureSpawnEntries_07_
↳Inland_NoCarnos.CreatureSpawnEntries_07_Inland_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/07_Equator/_PVE/CreatureSpawnEntries_
↳07_Inland_PVE.CreatureSpawnEntries_07_Inland_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/07_Equator/CreatureSpawnEntries_07_
↳Mountain.CreatureSpawnEntries_07_Mountain'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_07/CreatureSpawnEntries_07_
↳Mountain_NoCarnos.CreatureSpawnEntries_07_Mountain_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/07_Equator/_PVE/CreatureSpawnEntries_
↳07_Mountain_PVE.CreatureSpawnEntries_07_Mountain_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/07_Equator/CreatureSpawnEntries_07_
↳ShallowWaterForest.CreatureSpawnEntries_07_ShallowWaterForest'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_07/CreatureSpawnEntries_07_
↳ShallowWaterForest_NoCarnos.CreatureSpawnEntries_07_ShallowWaterForest_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/07_Equator/_PVE/CreatureSpawnEntries_
↳07_ShallowWaterForest_PVE.CreatureSpawnEntries_07_ShallowWaterForest_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/08_Polar/CreatureSpawnEntries_08_
↳BarrenLand.CreatureSpawnEntries_08_BarrenLand'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/08_Polar/CreatureSpawnEntries_08_
↳Beaches.CreatureSpawnEntries_08_Beaches'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/08_Polar/CreatureSpawnEntries_08_
↳Forest.CreatureSpawnEntries_08_Forest'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/08_Polar/CreatureSpawnEntries_08_
↳FreshWater.CreatureSpawnEntries_08_FreshWater'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/08_Polar/CreatureSpawnEntries_08_
↳Inland.CreatureSpawnEntries_08_Inland'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/08_Polar/CreatureSpawnEntries_08_
↳Mountains.CreatureSpawnEntries_08_Mountains'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/09_C_HighDesert/CreatureSpawnEntries_
↳09_Beaches.CreatureSpawnEntries_09_Beaches'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_09/CreatureSpawnEntries_09_
↳Beaches_NoCarnos.CreatureSpawnEntries_09_Beaches_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/09_C_HighDesert/_PVE/
↳CreatureSpawnEntries_09_Beaches_PVE.CreatureSpawnEntries_09_Beaches_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/09_C_HighDesert/CreatureSpawnEntries_
↳09_Forest.CreatureSpawnEntries_09_Forest'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_09/CreatureSpawnEntries_09_
↳Forest_NoCarnos.CreatureSpawnEntries_09_Forest_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/09_C_HighDesert/_PVE/
↳CreatureSpawnEntries_09_Forest_PVE.CreatureSpawnEntries_09_Forest_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/09_C_HighDesert/CreatureSpawnEntries_
↳09_FreshWater.CreatureSpawnEntries_09_FreshWater'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/09_C_HighDesert/CreatureSpawnEntries_
↳09_FreshWater_Carnos.CreatureSpawnEntries_09_FreshWater_Carnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_09/CreatureSpawnEntries_09_
↳FreshWater_NoCarnos.CreatureSpawnEntries_09_FreshWater_NoCarnos'

```

(continues on next page)

(continued from previous page)

```

Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/09_C_HighDesert/_PVE/
↳CreatureSpawnEntries_09_FreshWater_PVE.CreatureSpawnEntries_09_FreshWater_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/09_C_HighDesert/CreatureSpawnEntries_
↳09_Inland.CreatureSpawnEntries_09_Inland'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_09/CreatureSpawnEntries_09_
↳Inland_NoCarnos.CreatureSpawnEntries_09_Inland_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/09_C_HighDesert/_PVE/
↳CreatureSpawnEntries_09_Inland_PVE.CreatureSpawnEntries_09_Inland_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/09_C_HighDesert/CreatureSpawnEntries_
↳09_Mountain.CreatureSpawnEntries_09_Mountain'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_09/CreatureSpawnEntries_09_
↳Mountain_NoCarnos.CreatureSpawnEntries_09_Mountain_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/09_C_HighDesert/_PVE/
↳CreatureSpawnEntries_09_Mountain_PVE.CreatureSpawnEntries_09_Mountain_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/10_C_LowDesert/CreatureSpawnEntries_10_
↳Beaches.CreatureSpawnEntries_10_Beaches'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_10/CreatureSpawnEntries_10_
↳Beaches_NoCarnos.CreatureSpawnEntries_10_Beaches_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/10_C_LowDesert/_PVE/
↳CreatureSpawnEntries_10_Beaches_PVE.CreatureSpawnEntries_10_Beaches_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Caves/CreatureSpawnEntries_10_Cave.
↳CreatureSpawnEntries_10_Cave_PvE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/10_C_LowDesert/CreatureSpawnEntries_10_
↳Forest.CreatureSpawnEntries_10_Forest'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_10/CreatureSpawnEntries_10_
↳Forest_NoCarnos.CreatureSpawnEntries_10_Forest_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/10_C_LowDesert/_PVE/
↳CreatureSpawnEntries_10_Forest_PVE.CreatureSpawnEntries_10_Forest_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/10_C_LowDesert/CreatureSpawnEntries_10_
↳FreshWater.CreatureSpawnEntries_10_FreshWater'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_10/CreatureSpawnEntries_10_
↳FreshWater_NoCarnos.CreatureSpawnEntries_10_FreshWater_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/10_C_LowDesert/_PVE/
↳CreatureSpawnEntries_10_FreshWater_PVE.CreatureSpawnEntries_10_FreshWater_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/10_C_LowDesert/CreatureSpawnEntries_10_
↳Inland.CreatureSpawnEntries_10_Inland'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_10/CreatureSpawnEntries_10_
↳Inland_NoCarnos.CreatureSpawnEntries_10_Inland_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/10_C_LowDesert/_PVE/
↳CreatureSpawnEntries_10_Inland_PVE.CreatureSpawnEntries_10_Inland_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/10_C_LowDesert/CreatureSpawnEntries_10_
↳Mountain.CreatureSpawnEntries_10_Mountain'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_10/CreatureSpawnEntries_10_
↳Mountain_NoCarnos.CreatureSpawnEntries_10_Mountain_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/10_C_LowDesert/_PVE/
↳CreatureSpawnEntries_10_Mountain_PVE.CreatureSpawnEntries_10_Mountain_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/10_C_LowDesert/CreatureSpawnEntries_10_
↳ShallowWater.CreatureSpawnEntries_10_ShallowWater'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_10/CreatureSpawnEntries_10_
↳ShallowWater_NoCarnos.CreatureSpawnEntries_10_ShallowWater_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/10_C_LowDesert/_PVE/
↳CreatureSpawnEntries_10_ShallowWater_PVE.CreatureSpawnEntries_10_ShallowWater_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/CreatureSpawnEntries_Boss_Dragon.
↳CreatureSpawnEntries_Boss_Dragon'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/CreatureSpawnEntries_Boss_Hydra.
↳CreatureSpawnEntries_Boss_Hydra'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/CreatureSpawnEntries_Cattle.
↳CreatureSpawnEntries_Cattle'

```

(continues on next page)

(continued from previous page)

```

Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/CreatureSpawnEntries_Dinos.
↳CreatureSpawnEntries_Dinos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/CreatureSpawnEntries_Dinos_
↳Leatherwings.CreatureSpawnEntries_Dinos_Leatherwings'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/CreatureSpawnEntries_Farm.
↳CreatureSpawnEntries_Farm'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/CreatureSpawnEntries_Flying.
↳CreatureSpawnEntries_Flying'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/10_C_LowDesert/_PVE/
↳CreatureSpawnEntries_Flying_PVE.CreatureSpawnEntries_Flying_PVE'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/CreatureSpawnEntries_Penguins.
↳CreatureSpawnEntries_Penguins'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/CreatureSpawnEntries_Sheep.
↳CreatureSpawnEntries_Sheep'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/CreatureSpawnEntries_Town_
↳Desert.CreatureSpawnEntries_Town_Desert'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/CreatureSpawnEntries_Town_
↳Temperate.CreatureSpawnEntries_Town_Temperate'
Blueprint '/Game/Atlas/Test/CreatureSpawnEntriesTest.CreatureSpawnEntriesTest'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_BeachesTemperate.
↳DinoSpawnEntries_BeachesTemperate'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_BeachesTropical.
↳DinoSpawnEntries_BeachesTropical'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_BeachesTropical_
↳NoCarnos.DinoSpawnEntries_BeachesTropical_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_Carnos.
↳DinoSpawnEntries_Carnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_Flying.
↳DinoSpawnEntries_Flying'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_ForestsTemperate.
↳DinoSpawnEntries_ForestsTemperate'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_ForestsTropical.
↳DinoSpawnEntries_ForestsTropical'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_ForestsTropical_
↳NoCarnos.DinoSpawnEntries_ForestsTropical_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_FreshWater.
↳DinoSpawnEntries_FreshWater'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_Herbi.
↳DinoSpawnEntries_Herbi'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_InlandTemperate.
↳DinoSpawnEntries_InlandTemperate'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_InlandTropical.
↳DinoSpawnEntries_InlandTropical'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_InlandTropical_
↳NoCarnos.DinoSpawnEntries_InlandTropical_NoCarnos'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_Jungles.
↳DinoSpawnEntries_Jungles'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_MountainsTemperate.
↳DinoSpawnEntries_MountainsTemperate'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_MountainsTropical.
↳DinoSpawnEntries_MountainsTropical'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_NPCShips.
↳DinoSpawnEntries_NPCShips'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_OceanEpic.
↳DinoSpawnEntries_OceanEpic'
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_PlayerSpawnTemperate.
↳DinoSpawnEntries_PlayerSpawnTemperate'

```

(continues on next page)

(continued from previous page)

```
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_TreasureEnemies1.  
↳DinoSpawnEntries_TreasureEnemies1'  
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_TreasureEnemies2.  
↳DinoSpawnEntries_TreasureEnemies2'  
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_TreasureEnemies3.  
↳DinoSpawnEntries_TreasureEnemies3'  
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/DinoSpawnEntries_TreasureEnemies4.  
↳DinoSpawnEntries_TreasureEnemies4'  
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_04/FreeportSpawnEntries_04_  
↳Tropical.FreeportSpawnEntries_04_Tropical'  
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_06/FreeportSpawnEntries_06_  
↳Beaches_NoCarnos.FreeportSpawnEntries_06_Beaches_NoCarnos'  
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_06/FreeportSpawnEntries_06_  
↳Forest_NoCarnos.FreeportSpawnEntries_06_Forest_NoCarnos'  
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_06/FreeportSpawnEntries_06_  
↳FreshWater.FreeportSpawnEntries_06_FreshWater'  
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_06/FreeportSpawnEntries_06_  
↳Inland_NoCarnos.FreeportSpawnEntries_06_Inland_NoCarnos'  
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_06/FreeportSpawnEntries_06_  
↳Mountains_NoCarnos.FreeportSpawnEntries_06_Mountains_NoCarnos'  
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/FreeportSpawnEntries_Desert.  
↳FreeportSpawnEntries_Desert'  
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/FreeportSpawnEntries_Equator.  
↳FreeportSpawnEntries_Equator'  
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/FreeportSpawnEntries_Polar.  
↳FreeportSpawnEntries_Polar'  
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/FreeportSpawnEntries_  
↳Temperate.FreeportSpawnEntries_Temperate'  
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/FreeportSpawnEntries_Tropical.  
↳FreeportSpawnEntries_Tropical'  
Blueprint '/Game/Atlas/AtlasCoreBP/SpawnEntries/Freeport/_06/FreeportSpawnEntries_  
↳Tundra.FreeportSpawnEntries_Tundra'  
Blueprint '/Game/Atlas/Creatures/Cod/SaltwaterFish_Cod.SaltwaterFish_Cod'  
Blueprint '/Game/Atlas/Creatures/Marlin/SaltwaterFish_Marlin.SaltwaterFish_Marlin'  
Blueprint '/Game/Atlas/Creatures/Tuna/SaltwaterFish_Tuna.SaltwaterFish_Tuna'
```