Setting up the Sun and Sky Light

In production, unless you are setting up a test, you would already have a basic lighting set up in the map as well as a .bsp.

In this document we will cover setting up Sun & Sky Lights in a map that already has a lighting prefab.

Explanation of Sun Volume and Sky Light

What is a Sun Volume?

The Sun Volume is a Global Reflection Probe.

A Sun Volume is an object you place in the scene and use the shape to define where the Global Lighting is contained.

A Sun Volume will contain the sun and sky lights that you define in APE.

More explanation about the Global Probe can be found in the Reflection Probes Document.

global_fill_color is be used to tint the color of the global probe to help it match the local probe lighting better if needed.

global_fill_intensity is used to adjust the amount of **global_fill_color**

grid_density is used to adjust the resolution of the global probe voxels. You can go up to 64x64. This resolution is based on the size of the Sun Volume not world units.

Lighting States

You can have up to 4 lighting states (additional Global Lighting scenarios) that can be accessed & switched through script during run time.

Each lighting state has runtime lighting and an optional baked lighting override.

If you want to use bake override, assign the baked lighting to the first field. You can then use another ssi setup in the run time override.

ssi_runtime_override is what will be displayed in game and **ssi** is what will bake into the reflection probes.

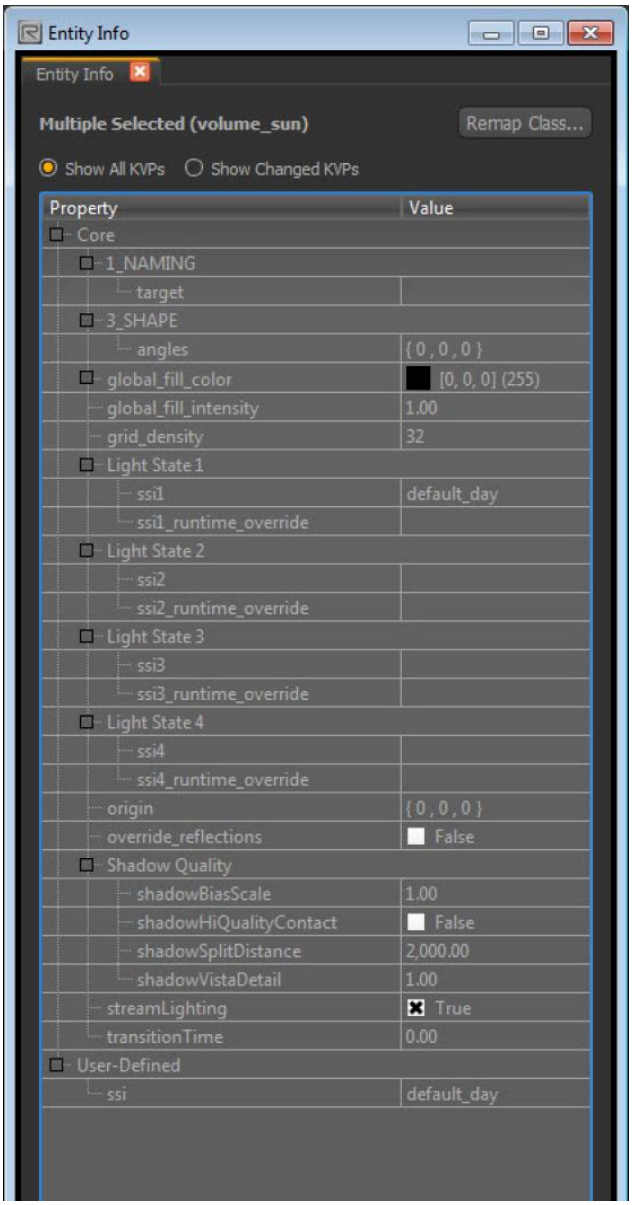
Further explanation on how to use Run Time Override & Override Reflections can be found in the **Advanced** section of this document below.

streamLighting can be toggled to keep this Sun Volume always in memory. This can be helpful to avoid popping when the camera is moving across Sun Volumes.

Shadow Quality

shadowSplitDistance is used to tune the quality of the sun shadows.

Lower numbers = higher quality and a larger memory footprint. Be careful! Further explanation below.



What is a Sky Light?

A sky light is defined in APE as a material.

A sky light is a spherical light projected into the scene using the sky texture for value and color.

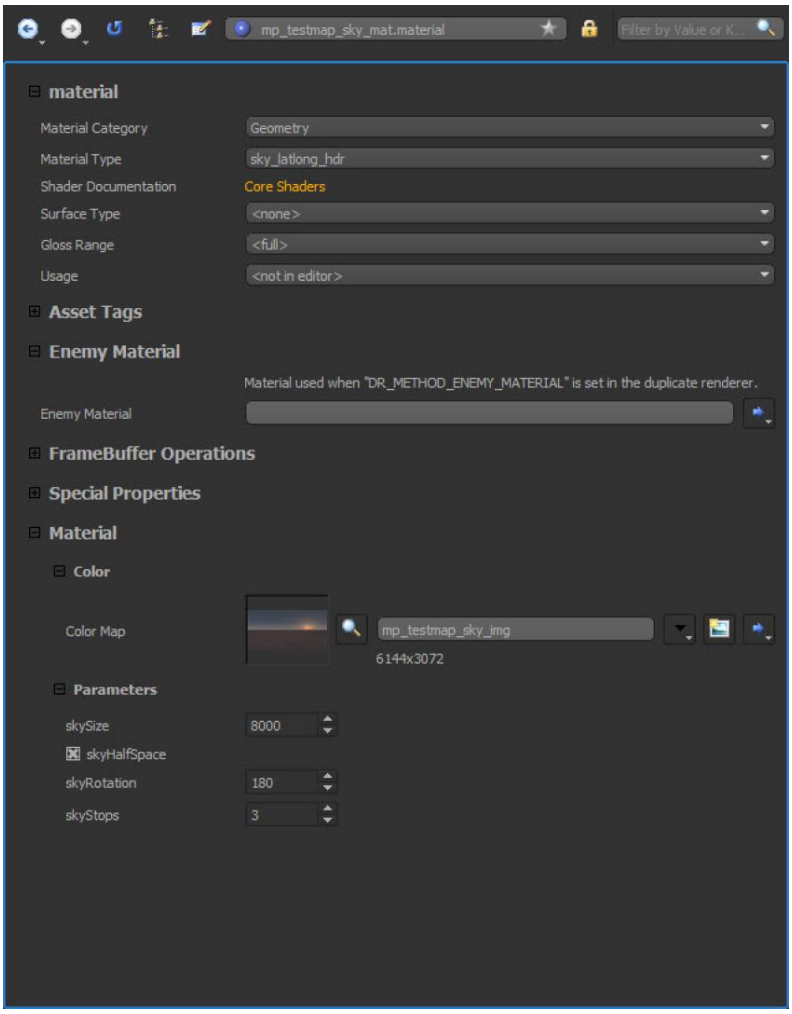
There are only two parameters to adjust for the sky light.

skyRotation is used to rotate the sky arount the up axis to move the sky into a desired position.

For example, the sun direction may not be lined up to the elements in the sky but you want to keep the sun direction where it is.

Use **skyRotation** to line the sky up to the sun direction.

skyStops is used to adjust the brightness of the sky light.



Shape the Sun Volume

This document assumes there is already a Sun Volume in the scene.
In case you need to add one you can find the volume in the Entity Browser under Volumes and drag it into the scene.

Tip: Add the Sun Volume to your right click menu by choosing Toggle Favorite after right click on the sun entity in the Entity Browser.

Manipulating a Sun Volume in 3d

Shape the Sun Volume the same way you would a brush. Adjust the size until the Sun Volume is covering the entire space you wish to light.

Limitations:

Sun Volume cannot be rotated.

Sun Volume cannot be cut or split. It must be a box with six sides.

Sun Volume cannot overlap other Sun Volumes.

Create Sun and Sky Light assets in APE

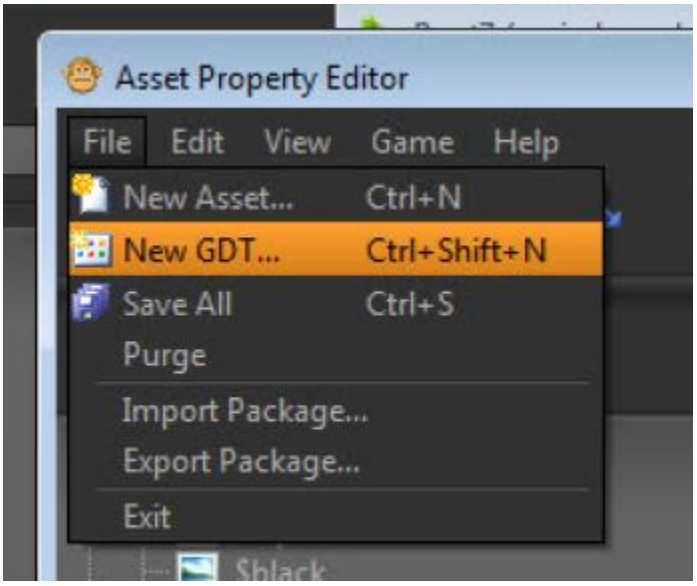
The Asset Property Editor or "APE" is used to create every asset in the game.

This includes Sun, Fog, Sky, light gobos, or anything else you want to use in the game or in radiant.

Create .gdt

It is recommended to create a new .gdt for lighting for each map. This is to streamline the debugging process and to avoid clutter.

Once APE is open, choose New GDT under the File Menu at the top left.

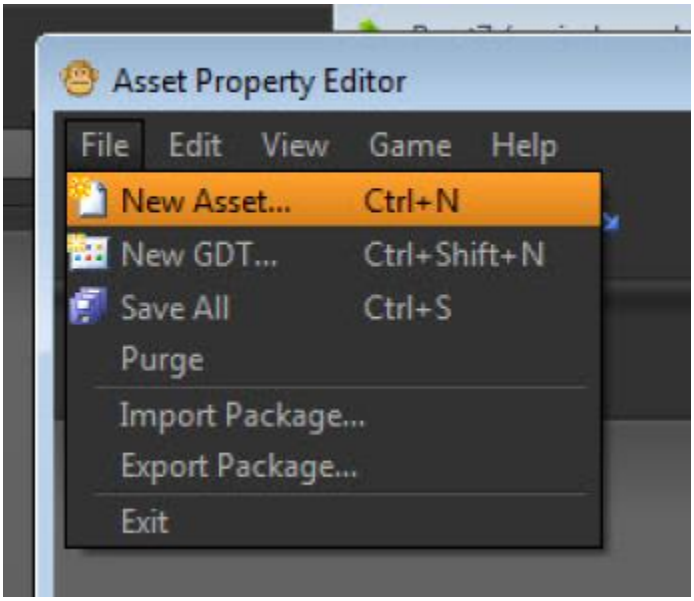


You will now have a new empty .gdt document to work with.

Sun Setup

Create SSI

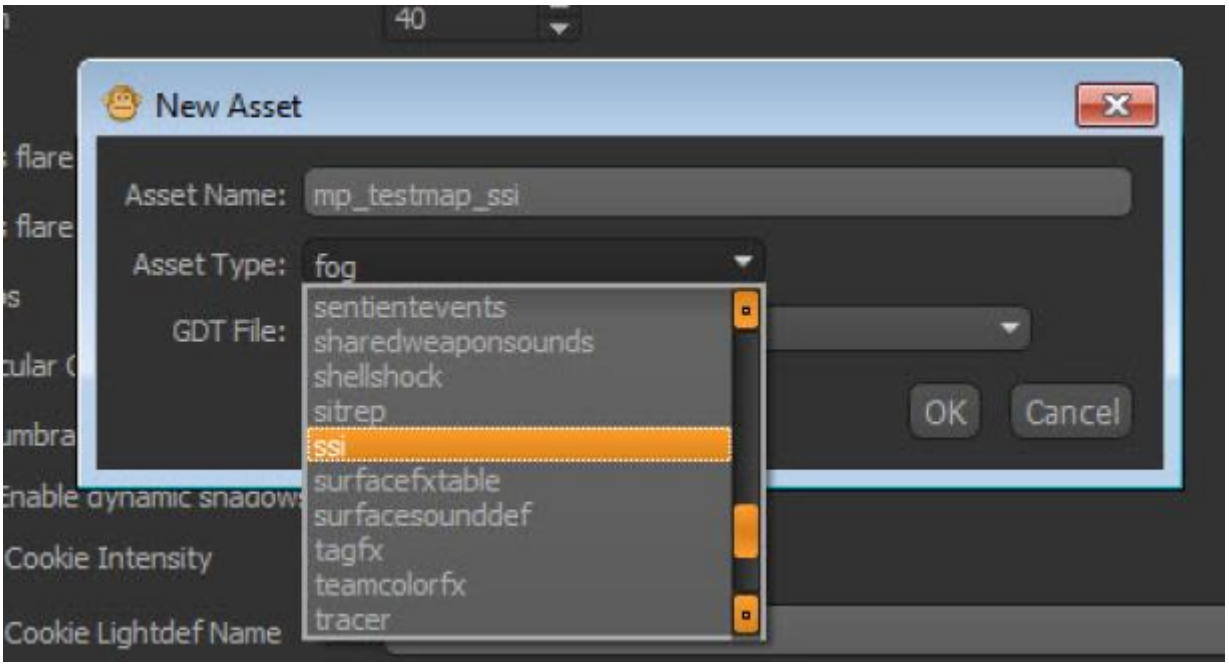
Choose New Asset under the File Menu at the top left.

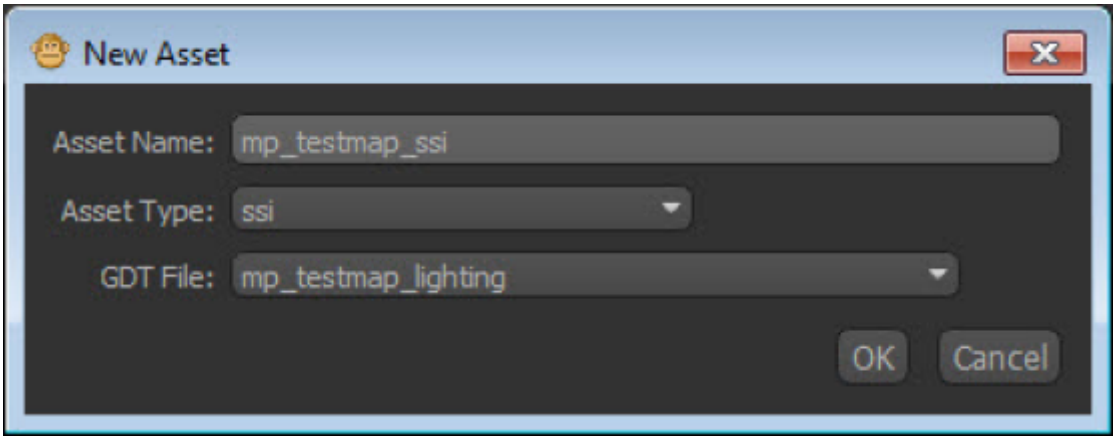


A dialogue will pop up that is used to define the new asset.

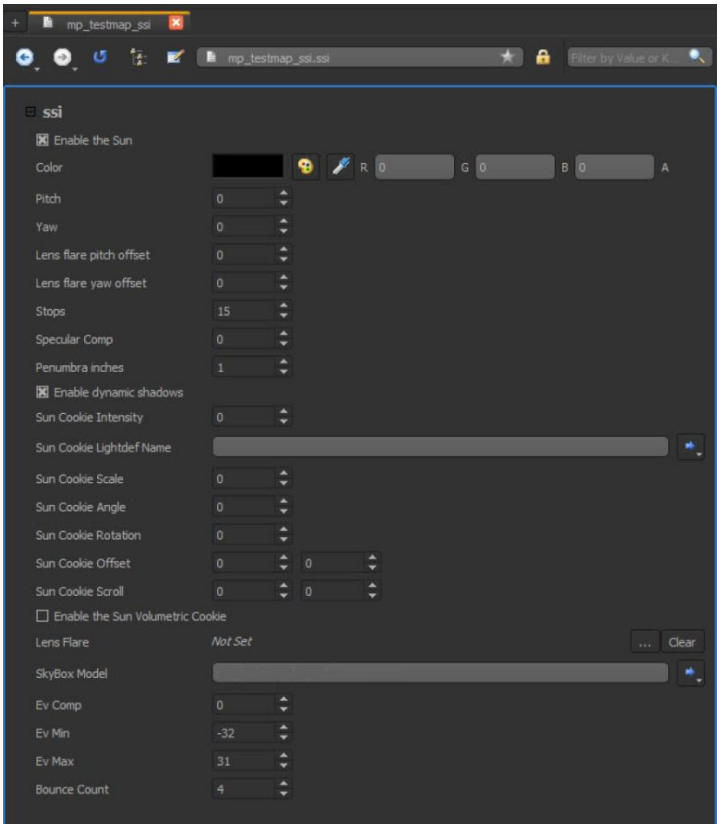
Name the asset and set the asset type to ssi. By default the open GDT will be selected in the GDT File dropdown. Alternatively you can choose a different GDT to add the asset to. For this tutorial keep the current GDT selected.

Tip: Follow a naming convention.



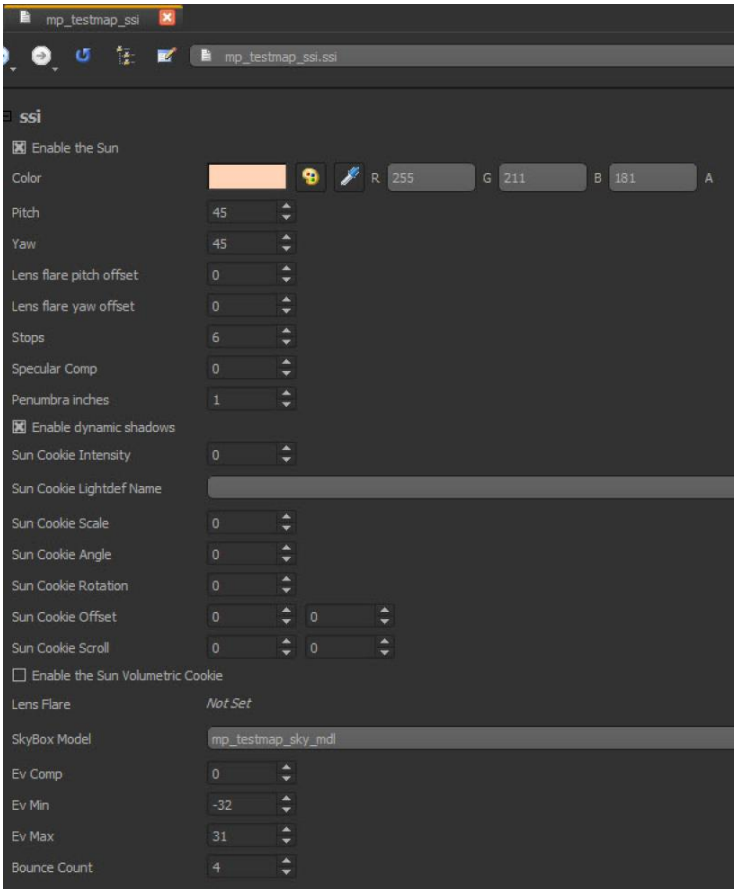


You will now have a new SSI (Sun) defined.



You can adjust the settings however you like.

These are the settings I am using:



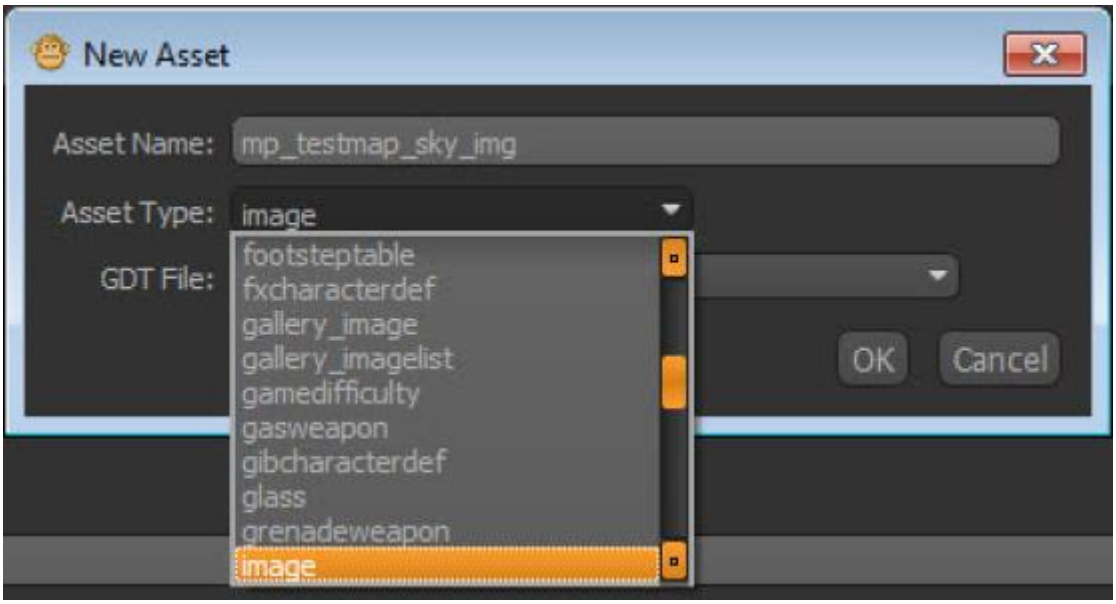
Sky Setup

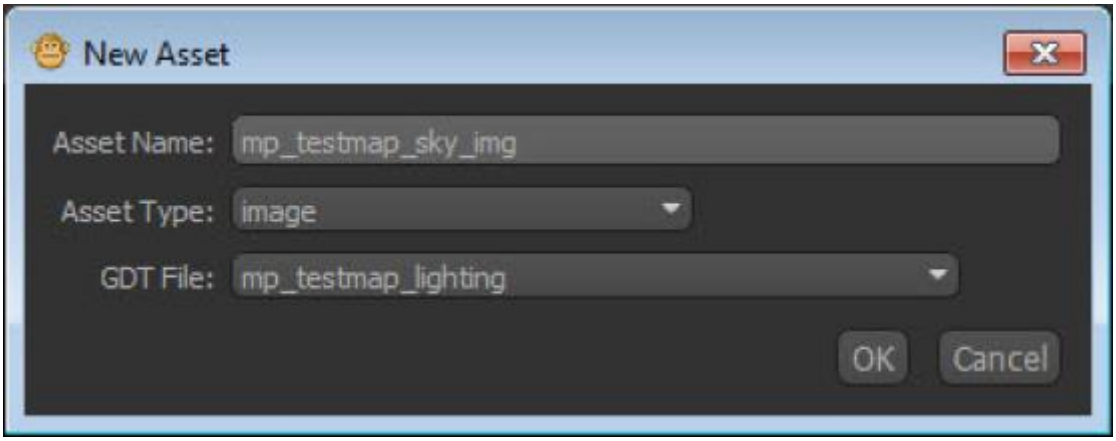
In order to use a sky three assets need to be defined.

1. Sky Image
2. Sky Material
3. Sky Model

Once these three assets are made they will then be added to the SSI asset.

Create Sky Image



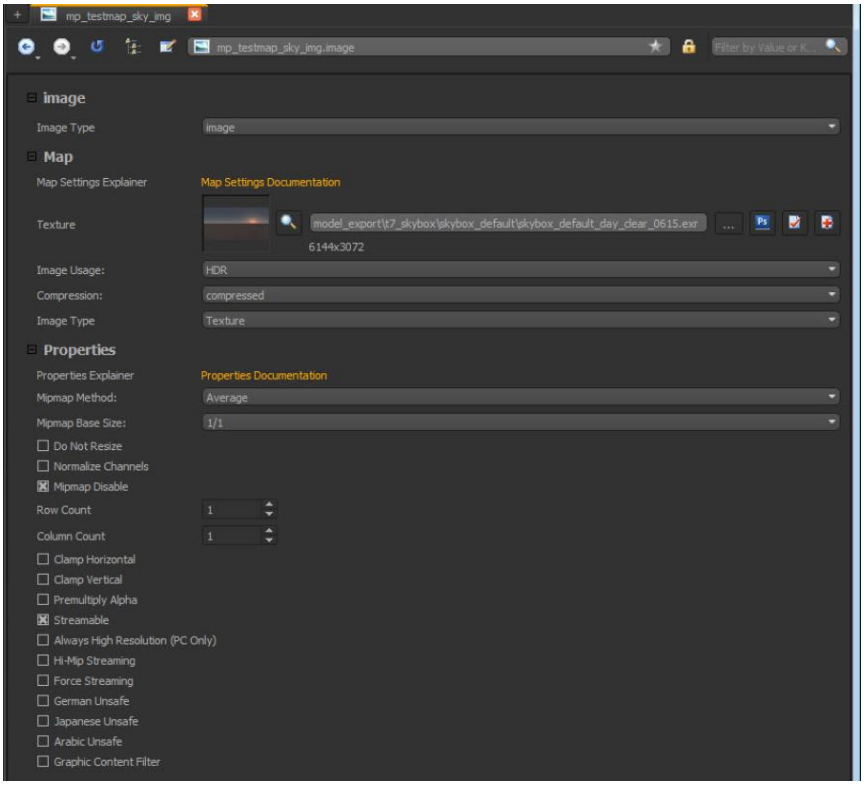


Use the following settings on your new image asset. Choose whatever sky .exr you like.

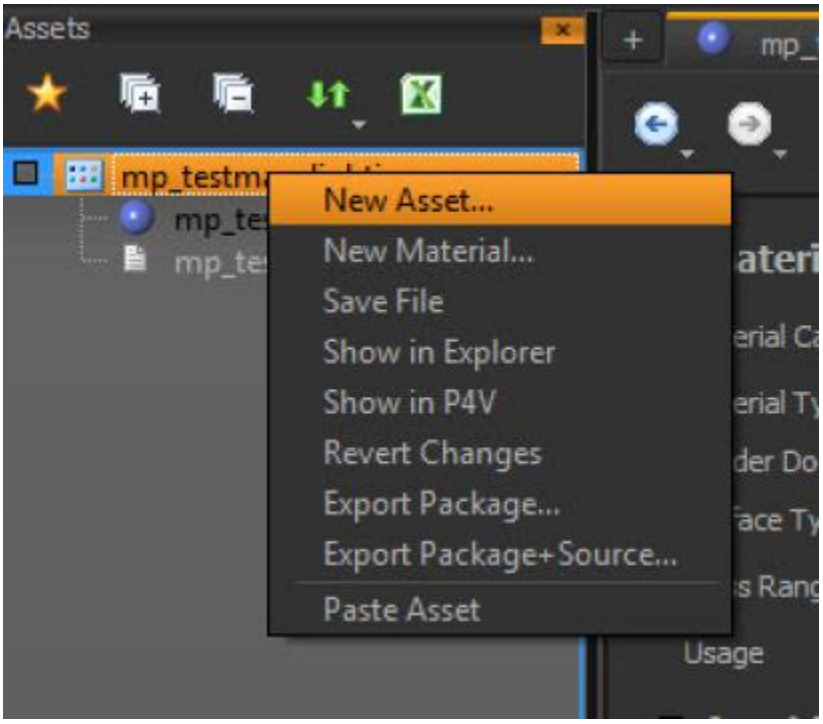
In a real production you MUST MAKE A UNIQUE .exr FILE!

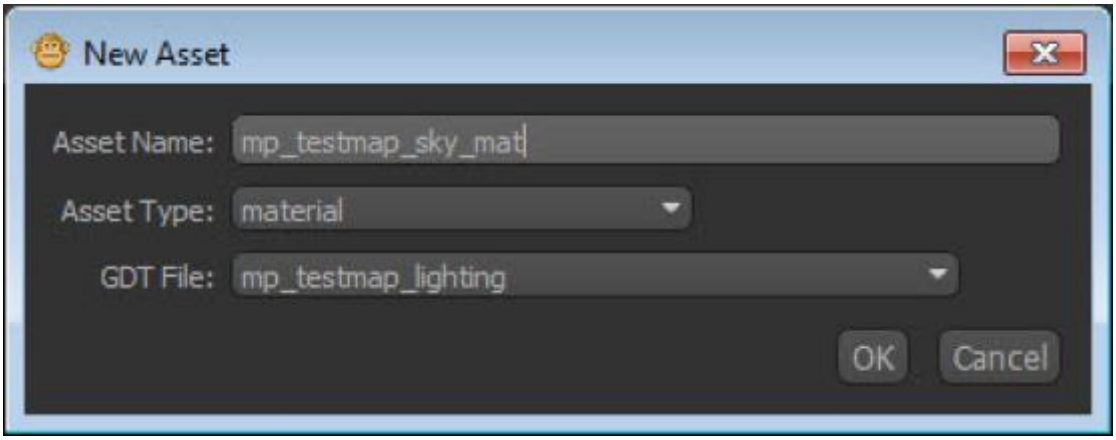
If you don't make a SPECIFIC .exr for your map bad things will happen when the .exr is edited for the map it was intended for.

In fact, do not directly reference any other asset that is used in another map.



Create sky material

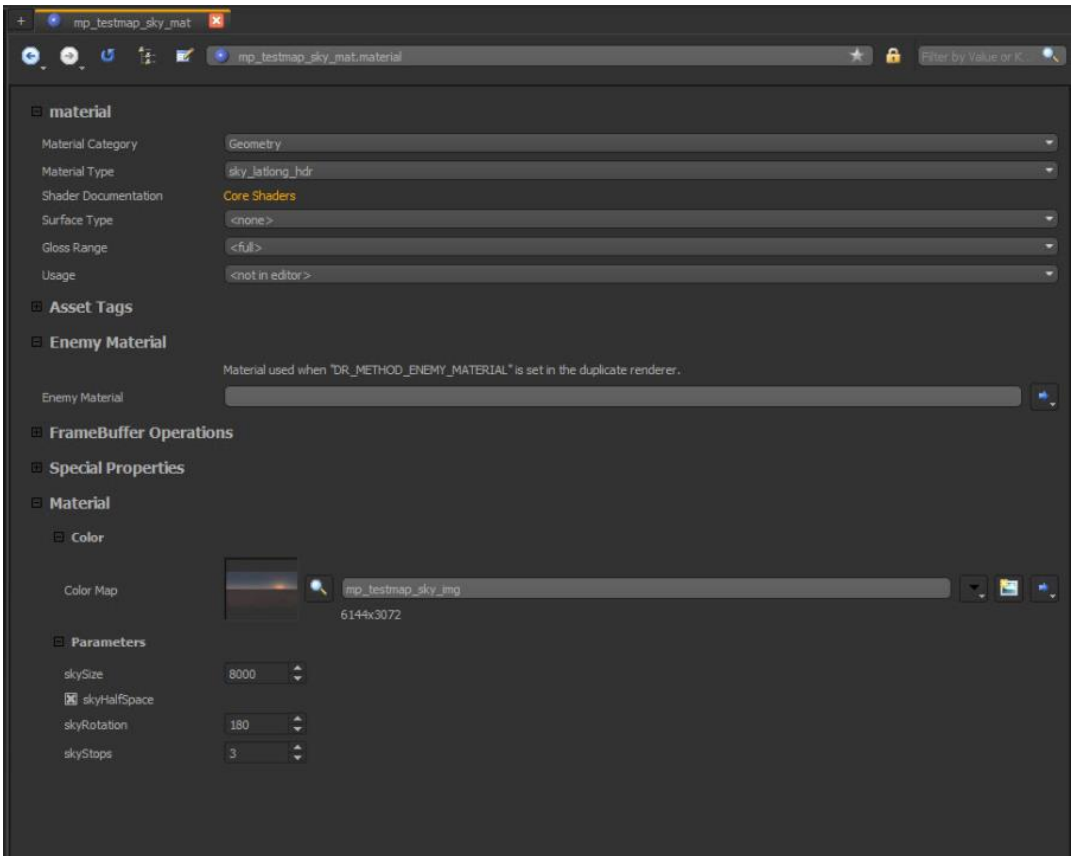




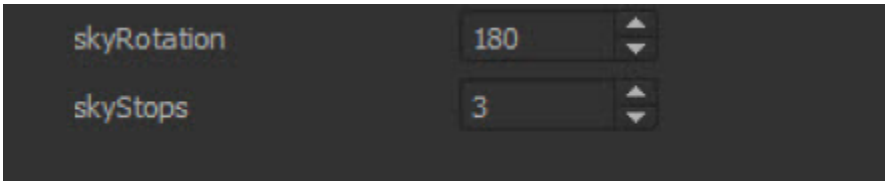
Use the following settings on your new material asset.

Add the image asset you previously made to the **Color Map** slot.

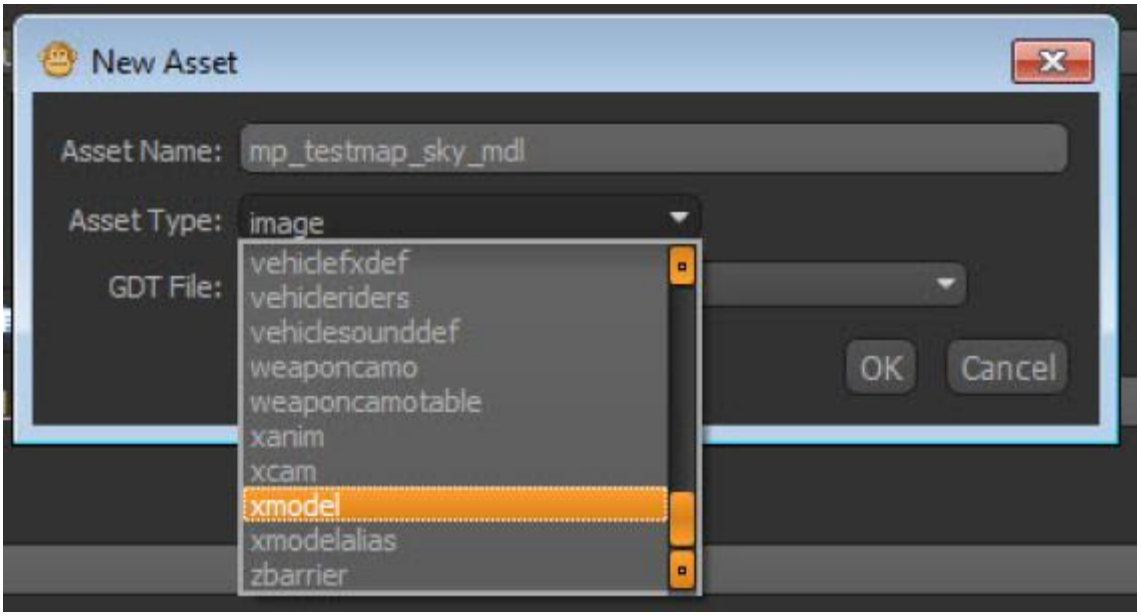
If you get linker errors recheck your settings.

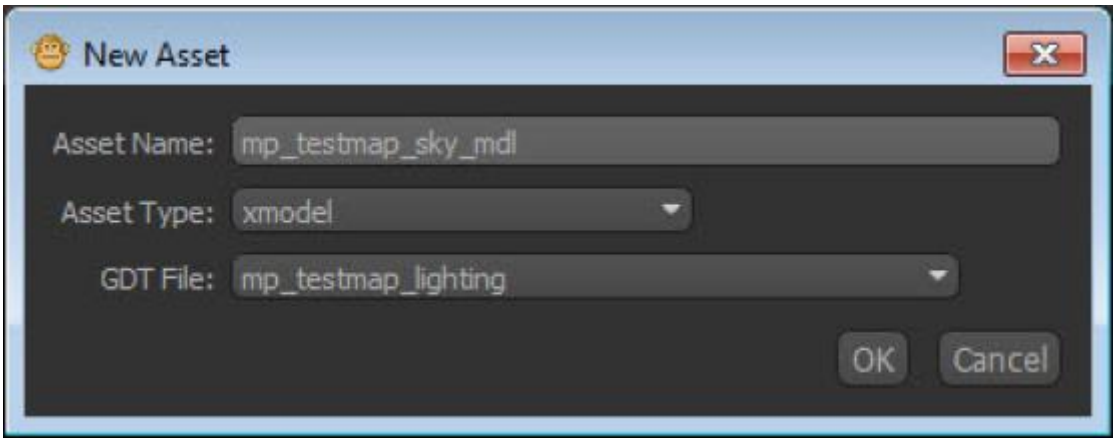


Here are the setting I used:



Create Sky Model



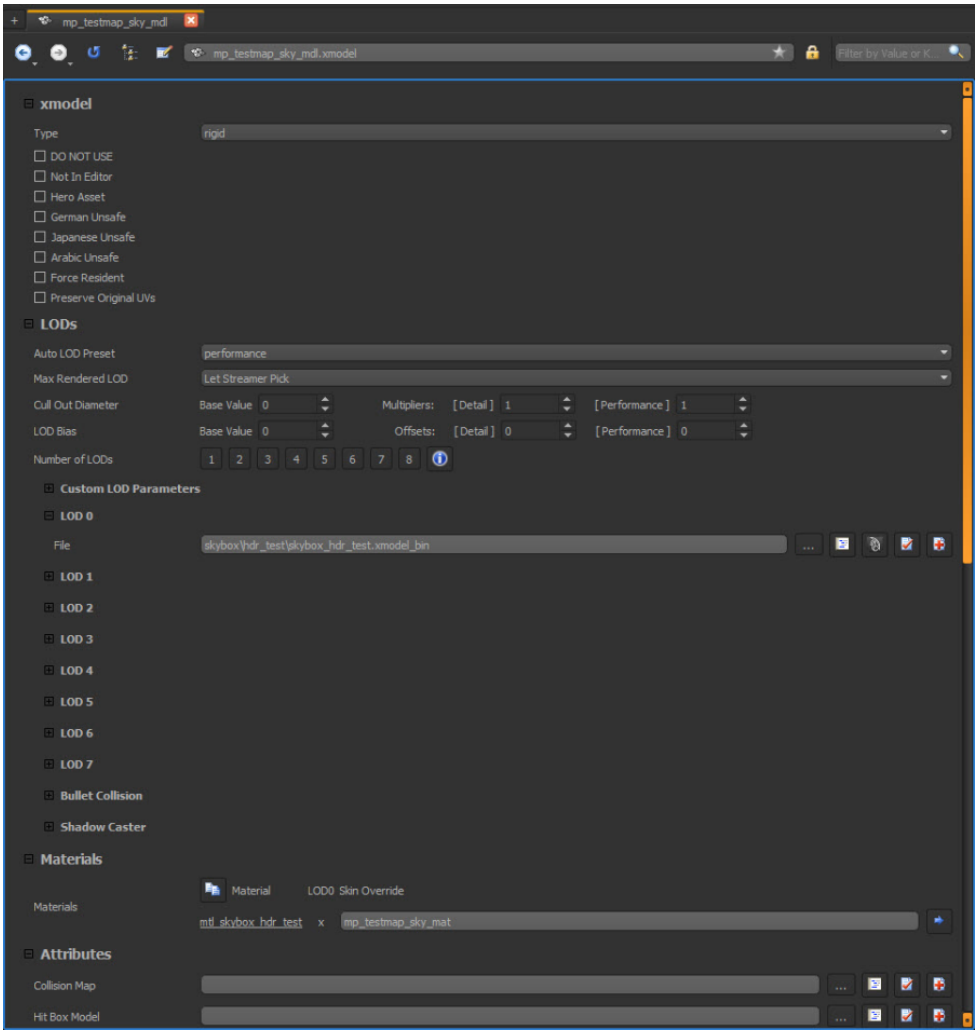


Use the following setting on your new model asset.

Under **LOD 0** use the model indicated in the **File** slot.

Under **Materials** use the sky material you previously made in the **Material** slot.

Again, if you get linker errors recheck your settings.



Assign Sky to SSI

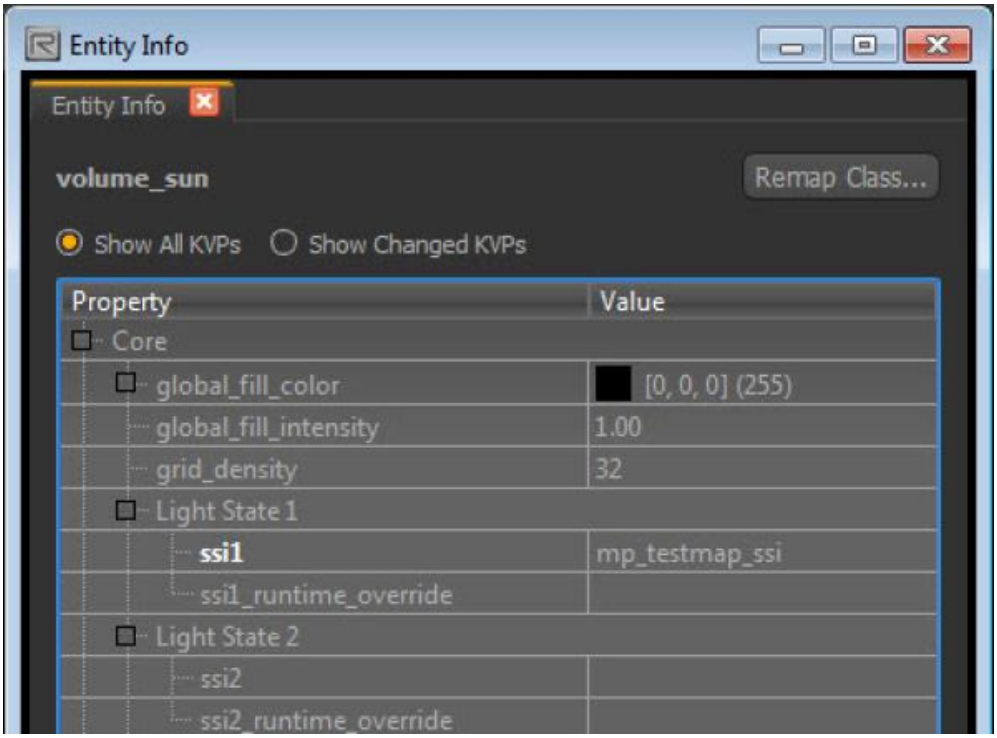
Select the ssi you are using and assign the sky model you previously created to the **SkyBox Model** slot.

You are now ready to go to Radiant!

Assign assets in Radiant

Now that you have an ssi with a sky defined you can plug it into the Sun Volume in Radiant.

Under **Light State 1** type or paste the same of the SSI you made in APE.

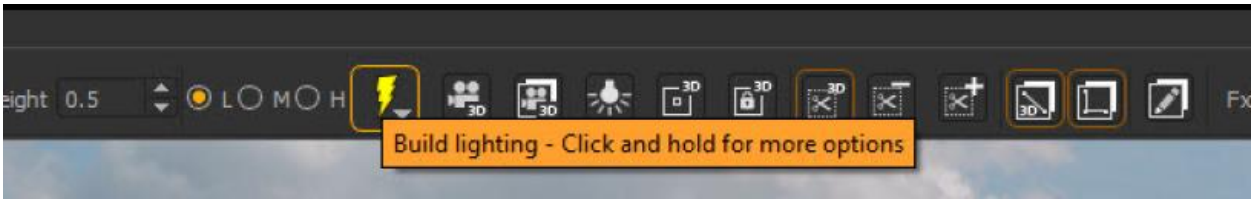


You will now see something happen in the 3d view but it doesn't look like good lighting.

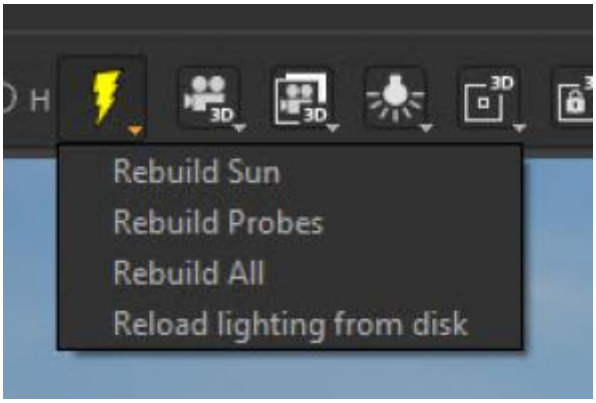
You need to bake to see your lighting.

Bake lighting in Radiant

Left Click the lightning bolt Icon.



Tip: Alternatively you can bake certain aspects of lighting by Left Click and hold on the Lighting Bolt Icon.



Once the Baking is done and if you used my settings; you should see something like the image below. *I have added some Geo so that I can see the Shadows better.*

Now that you have everything hooked up you may have some bad looking shadows. You may want to adjust the **shadow split distance.**

shadowSplitDistance set to Default @ 2000

shadowSplitDistance set to 250

Workflow & Live Update to Radiant

Now try adjusting all the different variables and see what kind of results you get.

Most variables will live update to radiant.

Some variables require that you rebake to see the result.

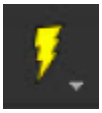
These include:

Pitch

Yaw

skyStops

Changing the Stops on the ssi & skyStops on the Sky Material will live update to Radiant but to see the effect of the bounced (GI) light you need to rebake.

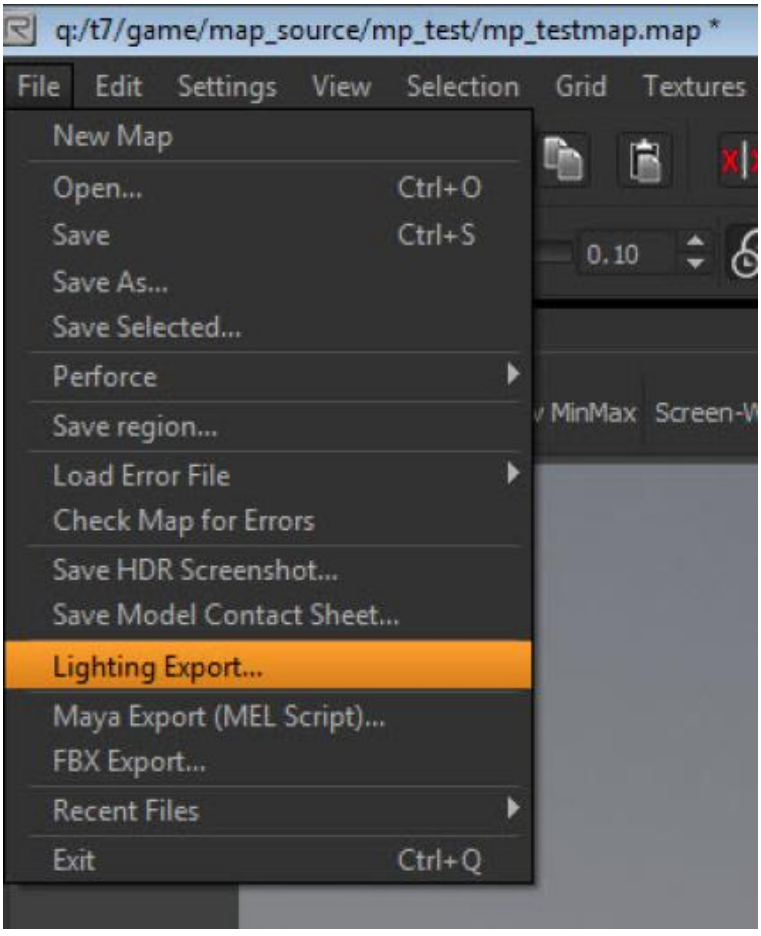


Export and run in game

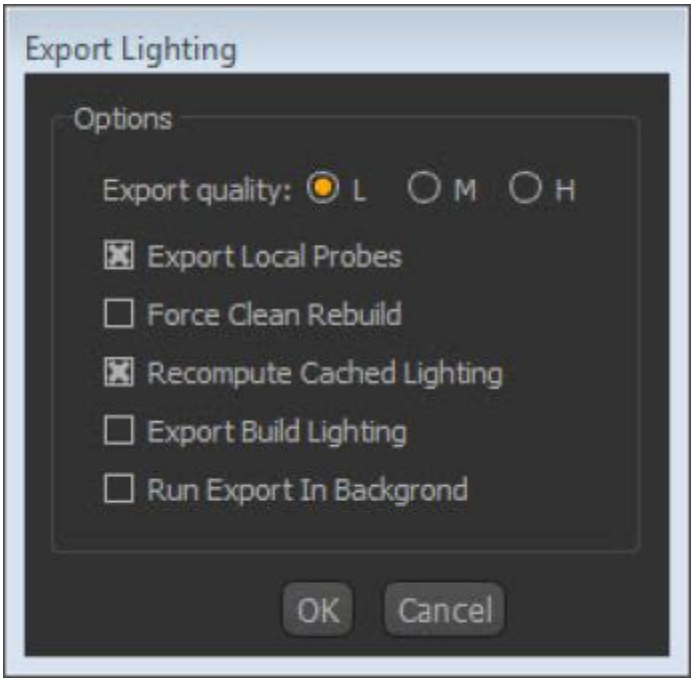
If you are happy with what you see in Radiant you are ready to export and see your work in Game.

Export an LED

Under the top left file menu select Lighting Export.



This Dialogue will pop up



You can choose your quality level. Higher level = longer bake time.

Usually you want to Force Clean Rebuild. This basically deletes the LED and creates a new one from scratch. Sometimes random bad things happen and this ensures a clean Build.

If want to see updated lights or need to see an exploder and know you don't need to update the reflection probes, you can save a lot of time by only checking Export Local Probes.

Compile, link and run your map

Once you have an LED, you are ready to Compile, Link and run the Game.

That's it!

Play around and try different setting, brightness, color etc.

You will be surprised how much you can do with just this simple set up.

Advanced

Using Override bake lighting

If you would like to use a different set up for baked lighting (GI) you can set up a new Sun and Sky light set up and use that.

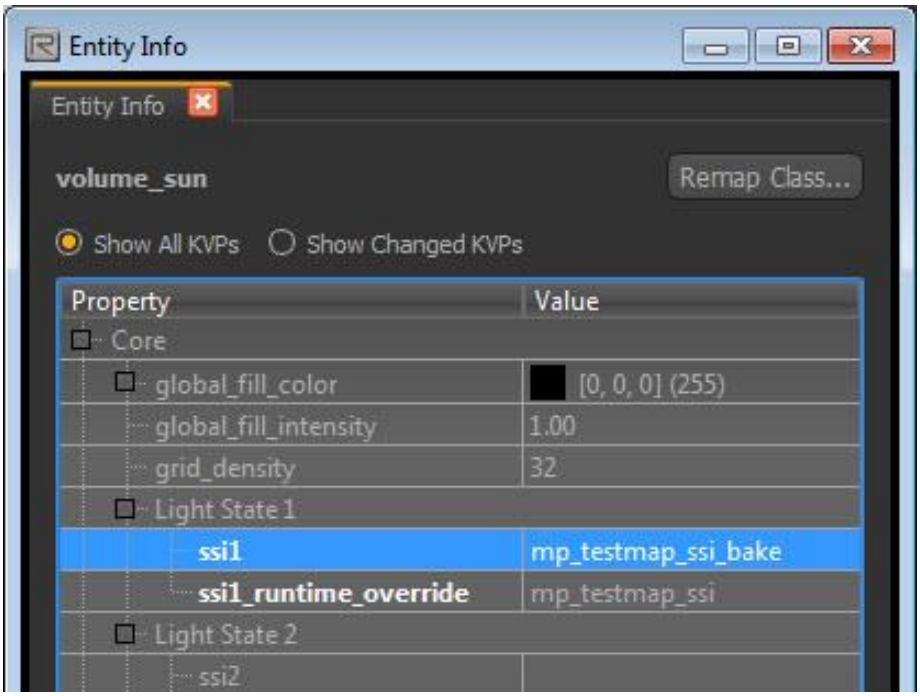
For example, art direction would like a very saturated blue sky in game. This saturated sky will cause the level to look very blue when the GI is baked.

This is where you should consider using a bake over ride set up.

Refer to section “**Create Sun and Sky Light assets in APE**“ For asset Creation.

Make a second setup and use the extention _bake. For this testmap i will call it "mp_testmap_ssi_bake"

On the Sun Volume under Light State 1 assign "mp_testmap_ssi_bake" to the ssi slot and move "mp_testmap_ssi" to the ssi_runtime_override slot.



ssi1_runtime_override is what will show up in game.

ssi1 is what is used to bake the GI.