# General S.A.M. Setup For Light Baking (Updated June 1, 2025)

Disclaimer: The previous Unity Light Baker has been replaced by the Unity Light Baker v2, which will allow for better APV support in the future. If you have a Light Baker Action already created, please re-create it using the Unity Light Baker (v2) Automated Action creation command!

- 1) S.A.M's Light Baking system is not compatible with Unity's Incremental/Auto Generate light baking functionality, so you should disable these in the Lighting window before continuing.
- S.A.M. currently supports baking light maps, light probes, and reflection probes, however all Asset Chunks that you want to bake **must be in scene format**, and you **must be using a** Scene based Chunk Streamer to load those scene based Asset Chunks.
- 3) Because the scene files are needed when baking, it is IMPERATIVE that you disable the Edit Prefab Roots When Possible option on your Scene based Asset Manager, so that the scenes are loaded while using the World Designer Tool.

The Asset Manager(s) can be found in the Editor Default Resources/Deep Space Labs/SAM/AssetManagers folder.

If using a Scene Chunk Streamer, this will be the **SceneChunkStreamer\_AssetManager**.

If using an Addressable Scene Chunk Streamer, the Asset Manager will be the AddressableSceneChunkStreamer\_AssetManager).

- 4) You must also enable the Keep Scenes Intact option in all Scene based Chunk Streamers loading scenes with lighting data, to ensure the scenes remain loaded while the Asset Chunks are loaded. Note that you can still use a Hierarchy Organizer with these Asset Chunks, as it is not necessary to keep the Asset Chunks in the scenes!
- 5) If you have Light Probes in your main scene, you can now split/slice them into individual cells which can be streamed in and out at runtime. To do so, simply assign game objects with Light Probe Group components to Zone Groupings via the World Designer Tool. The Zone Grouping you assign Probes to should generally be either Grouping 1, or a Zone Grouping that is part of a new World Grouping setup solely for baking lighting data (refer to Step 1 in **Setting Up Your First Bake**).

6) Note that changing probe positions at runtime is only possible on Unity 2023.2 and later versions. Since changing the positions is required for floating origin based worlds and endless/repeating worlds, this means these features can only be used with Light Probes on 2023.2 or later.

On earlier versions, you can still stream Light Probes in/out during the game, but the positions of those probes must remain fixed, meaning at runtime they will load in and remain at the position they were baked at.

- 7) If using Light Probes with a Floating Origin based World and/or Endless World on Unity 2023.2 or later, any LOD Groups with scenes that have baked lighting data must be paired with a Non Pooling Chunk Manager whose Chunk Reuse Possible setting is disabled.
- 8) You can and should create a Lighting Settings Asset and assign it to your main scene (where the SAM game objects are) via the Lighting window (just make sure this main scene is set as the Active Scene before assigning the asset to the Lighting Settings Asset field). This asset will be copied to each baked scene to ensure that all scenes contain the same lighting settings.

# Setting Up Your First Bake

1) First you must decide which World Grouping will serve as your Primary Baking Grouping.

Usually Grouping 1 is the best choice however that is not strictly required. Generally speaking, the most important thing is that **the Cells of this Grouping be as large as or larger than the Cells of other Groupings**.

When baking lighting data, each Cell on a Zone Grouping that is chosen for each **Automate Operation** will serve as a Primary Cell. Lighting data will be baked for each Primary Cell's Asset Chunks, along with Asset Chunks associated with the Secondary Cells loaded alongside each Primary Cell. Asset Chunks belonging to Tertiary Cells will be loaded as well for each Primary Cell, however they will only contribute to the lighting calculations.

For more information on this process, please refer to the preview video on Automation: <u>https://www.youtube.com/watch?v=qyPJXeJOVHY</u> 2) Create a new Unity Light Baker asset in whatever folder you wish (right click a folder and choose Create -> Deep Space Labs -> SAM -> Automated Actions -> Unity Light Baker (v2)).

Assets GameObject Component Jobs Window	Help									
Deep Space Labs					► II	M				
Create	MicroVerse	>								
Show in Explorer	Deep Space Labs	>	Core	>		_	🔾 🔻 2D 💡			
Open	Folder		SAM	>	Assignment Selectors	>				
Delete					Automated Actions	>	Chunk Cleaner			
Rename	C# Script				About SAM		Find Disabled Behaviours			
Conv Path Alt+Ctrl+C	2D	>			Default Asset Creator		Find Disabled Renderers			
copyrum Airean c	Visual Scripting	>	_		Asset Chunk Group Renamer		Find Mesh Renderers With Material			
Open Scene Additive	Shader	Shader > Shader Variant Collection			Loading Blueprint Repository		Modify Static Editor Flags			
View in Package Manager	Shader Variant Collection				Miniature Asset Generator		Null Tree Replacer			
	Testing	>			Scene Formatter		Print Cell Info			
Import New Asset	Playables	>			Scene Generator		Serialized Property Modifier			
Import Package	Assembly Definition				Streamable Grid		Terrain Detail Replacer			
Export Package	Assembly Definition Reference		d Designer To	ol		Units Patent Light Clearer				
Find References In Scene	Text	>	Error Pause E				Unity Light Baker (V2)			

3) If you know are not using Light Probes, you can disable the **Store Light Bake Data** setting on your **Automated Unity Light Baker** asset. This will stop the baker from adding a special component to each baked Asset Chunk, which will slightly reduce the memory size of each scene.

Inspector Automated Unity Light Baker (Automated Unity Light Baker)									
Addressable									
Script	AutomatedUnityLightBaker								
Copy Lighting Settings									
Asset From Main Scene*	~								
Copy Render Settings									
From Main Scene*	<b>~</b>								
Store Light Bake Data*	~								

Do note this component could be used for other custom systems if you wish, as it stores the position of each Asset Chunk at the time of baking, which may be useful to you. In that case, leave the setting enabled.

4) Open the World Designer Tool and make sure the Grouping you selected in step 1 is set as the **Grouping To Edit**:



5) Navigate to the **Automate** tab in the **Operation Settings** Control Area.

•	<ul> <li>Operation Settings*</li> </ul>									
	General	Chai	ning	Assignment						
	Transfer	Auto	mate	Evaluate						
Settings for World Grouping 1: Terrains										
Samplin	ig Method*	Activ	Active Grid Query							
Active 0	Grid*	None	None (Active Grid)							
Automa	te Every Cell*									
Adjust C	Drigin Cell*	~								
► Auto	mated Actions*				0					
Load Secondary/Tertiary Cells*										
	Enable All									
1: Terrai	ins	~	2: MV_H	lierarchies	~					

6) **If using Light Probes in Unity 2023.1 or earlier**, where probe position changes are not possible, you must disable the **Adjust Origin Cell** setting, which ensures the Asset Chunks are loaded at the correct position when baking the lighting data.

You should also confirm that the Origin Cell set in the World Designer Tool is the one you want to use for your game. If you need to change this later, you will need to re-bake the light probe data!

If using Unity 2023.2 or greater with Light Probes, or an earlier version without Light Probes, and you are planning on using a Floating Origin based World, you can enable the **Adjust Origin Cell** setting, which will keep all loaded assets close to the World's Origin (more important for larger game worlds).

7) Set the **Sampling Method** to **Active Grid Query**.

- 8) Set the **Automated Actions** array to a size of 1, then drag and drop the Automated Unity Light Baker v2 asset you created in **Step 2** to the Element 0 field.
- 9) Decide which World Groupings should have Secondary and Tertiary Cells loaded alongside each Primary Cell that is baked. Effectively, these are the cells whose Asset Chunks you want baked (in addition to the Asset Chunks belonging to the Primary Baking Grouping's cells).

For each Grouping you want cells loaded for, enable the toggle under the Load Secondary/Tertiary Cells section.

Note that any Cells (other than the Primary Cell) from the same Grouping as the Primary Baking Grouping/Grouping To Edit will always be loaded as Tertiary Cells, meaning they will contribute Global Illumination but not be baked. If the Primary Baking Grouping's asset chunks actually contain assets that can affect the lighting, you should leave this Grouping enabled. If the Grouping is only being used to store lighting data, disable this Grouping.

- [Optional Step] Duplicate your existing Active Grid and deactivate the new game object. This will allow you to perform Step 11 without effecting the loadout of your cells at runtime (i.e., when you enter Play Mode).
- 11) Decide which Loading Blueprints to use on your Active Grid, which will dictate which cells are loaded alongside the Primary Cell as Secondary and Tertiary Cells.

At this time, you only need to be concerned with the Blueprints assigned to Active Grid Groupings that correspond to the Groupings enabled in **Step 9**, since these are the only Groupings for which cells will be loaded.

In most cases, a simple 3x3 (2D or 3D depending on your World Groupings) Uniform Ring Blueprint should suffice, though this will vary depending on how much of an influence your assets have on surrounding assets. For instance, taller assets will cast longer shadows and have a larger sphere of influence.

A good way to think about it is like this: How many cells do I need to load around whatever cell I am baking lighting data for, so that the lighting data for that cell is as accurate as possible?

Another thing to consider is that Terrains have an option to automatically generate Light Probes for Trees. If you enable this option, then all Terrain loaded for a baking operation, including Terrain associated with Tertiary Cells, will generate Light Probes which are then saved with the Primary and Secondary Cell scenes. In some cases this may be undesirable, as it can result in Light Probes that fall outside the bounds of the Primary and Secondary Cells. In these scenarios, you should configure the blueprint to only load a single LOD 1 cell (assuming these cells use the Unity Terrain Component), and then all other cells should be LOD Cells which load imposter terrains (no Unity Terrain component).

- 12) Assign your Active Grid (either the original or the duplicate if you created one in step 9) to the **Active Grid** field in the **Automate Settings** tab of the World Designer Tool.
- 13) Do a test run of the light baking by selecting a single Cell (perhaps towards the middle of your Grid), and pressing the Execute Automate Operation button (found in the Editing Controls tab of the Main Controls area, below the Cell Color Key).
- 14) Examine the results of the light baking by loading all Primary and Secondary Cells that would have been loaded during the Automate Operation.

In this case, the Primary Cell is the cell you selected before pressing **Execute Automate Operation**.

Secondary Cells are cells from any other Groupings (which were enabled in the **Load Secondary/Tertiary Cells** section) whose positions (min position of Cell) fall inside of the bounds of the Primary Cell. 15) Set the Scene View's Shade Mode to **Baked Lightmap** to see the lightmaps.



If using Light Probes, you can also enable Light Probe visualization to see the baked light probes, however note that light probe positions are not adjusted by the World Designer Tool, so may appear out of position.

- 16) If the lighting results are not to your satisfaction, adjust them and re-bake the same Primary Cell. Also pay attention to the time it took to perform the Automate Operation (printed in the console log), as the total bake time will be at least as long as this value times the remaining number of Primary Cells.
- 17) Once the results are to your satisfaction, bake the rest of the Primary Cells, or alternatively you can proceed with just a few more Cells first to try and get a feel for a small portion of the World before committing to a larger bake.

If you wish to bake all cells but not all cells are selectable (because your grid is very large), you can enable the **Automate Every Cell** setting before pressing **Execute Automate Operation** (in this case, selected Cells are ignored).

#### Final Note

You will notice that the main scene (the one that contains the S.A.M. related game objects/components) has a lighting data asset generated for it (in a folder with the same name as the scene, within the folder that the scene is in). This is because S.A.M. performs a final bake

containing just this scene after baking one or more Primary Cells. This is done to ensure the scene does not reference any of the Lighting Data assets associated with the Primary Cells, and is also necessary to ensure adding Light Probes at runtime works properly.

In addition, note that you will likely see some warning messages in the Console during/after the bake. These may say something like "Cross scene references are not supported . . ." or "Light Name in scene Scene Name (click to ping it)", and finally "Your current multi-scene setup has inconsistent Lighting settings . . ." All of these warnings can be ignored.

# **Clearing Baked Lighting Data**

 To clear baked lighting data, create a Unity Baked Light Clearer asset (right click a folder and choose Create -> Deep Space Labs -> SAM -> Automated Actions -> Unity Baked Light Clearer).

nux - l	Inity 2022.3.0 <dx11></dx11>		<b>A</b>										
	Deep Space Labs	>	FlatKit	>									
	Create	>	Addressables	>									
=			MicroSplat	>	iahtina								
	Show in Explorer		Deep Space Labs	>	Core	>	▼ Scale ●— 1x	Play F		澎	• =		Gizm
	Open		Eolder		Terra Slicer	>							
	Delete		Toldel		SAM	>	Assignment Selectors	>					
/	Rename		C# Script			ы.	Designer Grouping Listeners	>					
	Copy Path	Alt+Ctrl+C	2D	>			Automated Actions	>	Unity Ba	ked Li	ight Cle	arer	
	Open Scene Additive		Shader Graph	>			About SAM		Unity Lie		NEI		
-	· View in Package Manager		Shader	>	> Default Asset Creator				Print Action				

- 2) Open the World Designer Tool (if it is not already open), then set **Grouping To Edit** to the Primary Baking Grouping.
- 3) Assign the AutomatedUnityBakedLightClearer asset to **Element 0** of the **Automated Actions** array.
- 4) [Optional Step] Change the Sampling Method to Cell Ownership. This will ensure that only each Primary Cell processed and Cells it owns (Secondary Cells) have their baked lighting data cleared.

If you do not perform this step, any Tertiary Cells loaded will be unloaded before clearing the data, to ensure baked lighting data is not improperly cleared. While this means there is no danger if you decide to skip this step, it will add some extra processing which will slow down the **Automate Operation**. 5) Select any Cells you wish to clear baked lighting data for and press **Execute Automate Operation**. If you wish to clear all cells, simply enable the **Automate Every Cell** setting before pressing the **Execute Automate Operation** button (any selected Cells will be ignored).

### **How Automated Light Baking Works**

When you execute an Automate Operation on a Primary Cell, the World Designer tool loads the Asset Chunks associated with that Primary Cell, along with additional Cells (and their Asset Chunks) that is controlled by the **Sampling Method** and **Load Secondary/Tertiary Cells** toggles.

**Secondary Cells** are cells whose positions (as defined as the min point on the cell) fall within the bounds of the Primary Cell, while **Tertiary Cells** are cells whose positions fall outside of the bounds.

When baking a Primary Cell, the Asset Chunks associated with the Primary Cell and Secondary Cells are baked. Asset Chunks associated with Tertiary Cells are loaded but not baked, which is achieved by moving the asset chunks into the main scene, unloading the scenes they were formerly in (without saving them, which preserves the asset chunks in their original state) and setting any Mesh Renderers or Terrain components' Scale In Lightmap values to 0.

Because each Secondary Cell can only be loaded one time as a Secondary Cell (i.e., it is tied to a single Primary Cell), and each Primary Cell is only loaded once as a Primary Cell, this strategy ensures that lighting data will only be baked once for each Asset Chunk (which ensures the data is never overwritten, unless you re-bake the Primary Cell).

#### <u>Zones</u>

If you are using multiple Zones, the light baking system will fully support them. This means cells from other Zones will still be loaded as Secondary or Tertiary Cells so long as they are within range of the Primary Cell.

#### **Baking Times**

If doing a comparison between baking a single giant scene and baking all Primary Cells (which should result in all assets being baked), baking with S.A.M. will likely be faster, however this is not guaranteed.

The real advantage of S.A.M.'s automated baking is the ability to perform partial bakes, where only a small subset of assets in your game world are baked at a time. This has the following advantages:

- It allows you to fine tune your baking and lighting settings before committing to a full world bake. This way you can experiment with different options in order to see how a small portion of your world will look, which should be representative of how the entire world will large after performing a full bake (a bake of all Primary Cells).
- 2) When you make changes to an asset, or multiple assets in a certain area of your game world, you do not need to re-bake everything. Just re-bake the Primary Cells that those assets are associated with!

Do note that this will require you to track which assets have changed, and manually determine which Primary Cell owns the cells those assets belong to (assuming the assets do not belong to the Primary Cell itself).

The advantage here may be minimal if you are already using Unity's Progressive Light Baking.

3) If you want to perform a full bake, you can do so and then cancel the operation mid-way if needed. All Primary Cells (and secondary cells loaded alongside those cells) that have already been baked will retain their baked settings, and you can come back and continue the bake with the Primary Cell that was being processed when the operation was cancelled, along with any other Primary Cell's that had not been baked yet.