# Bake AO Documentation

---

Version: 1.0.1

**Asset Store:** https://assetstore.unity.com/packages/slug/263743

**Discord Community**: https://discord.gg/NT2pyQ28Jx

**Roadmap**: https://trello.com/b/tyx5eewn/bake-ao-roadmap-after-release

**Website**: https://proceduralpixels.com

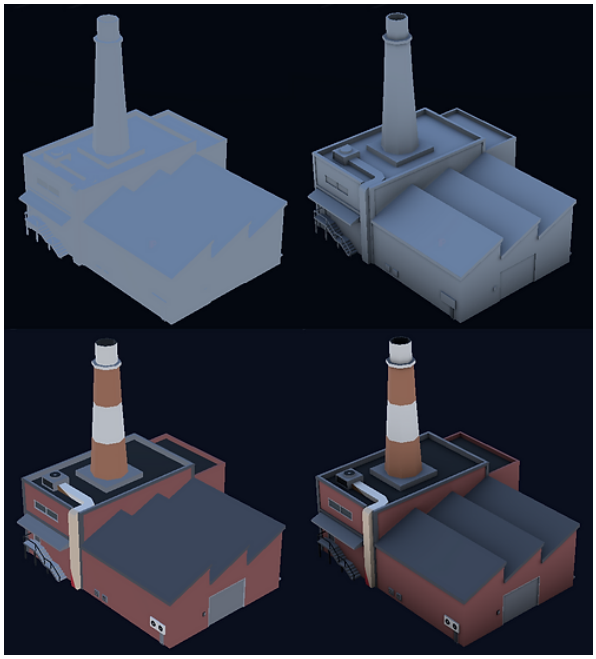
## 1. Introduction

### About Bake AO Plugin

Bake AO is a plugin designed to enhance your Unity projects with high-quality ambient occlusion textures. Bake AO streamlines your workflow by enabling you to bake ambient occlusion textures directly within the Unity Editor, saving you time and effort in the creation process.

With Bake AO, you can generate ambient occlusion textures for any mesh in your project to add visual depth and realism. Whether working on a small indie project or a large-scale production, Bake AO is the perfect tool to elevate your game environments.

Why even care about AO? Under certain light conditions (ambient lighting, shadows), models with AO textures seem to be more detailed and easier for players to read. The example below is a bit exaggerated, but shows the importance AO.



Model with and without ambient occlusion texture

The ambient occlusion texture used in the model above (128x128, format BC7 Unorm, 21.4KB in the build)

## 2. Installation

## Requirements

Before installing Bake AO in your project, ensure that your system meets the following requirements:

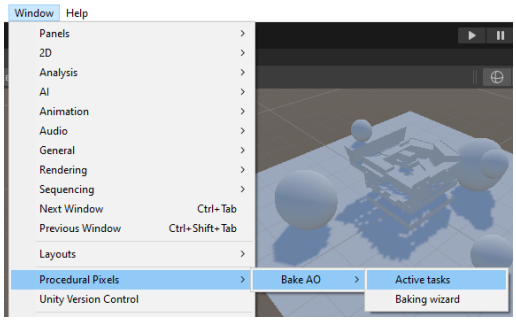
- **Supported OS:** Windows, MacOS
- **Supported GPUs:** NVidia, AMD, Intel, and Apple GPUs with at least 2GB of VRAM and Shader Model 5.0, such as GTX 580 or Radeon HD 6970. Integrated GPUs also work fine, but baking times will be longer.
- **Supported Unity Versions:** 2021.3 or newer
- **Supported Render Pipelines:** Built-in, URP

  Textures can also be baked in HDRP, but the BakeAO component for automatic texture setting is not supported on HDRP, no shaders for HDRP are included.
- **Project adjustments:** To fully utilize the features of the BakeAO component, you need to use shaders included with BakeAO or adjust custom shaders (more details here).

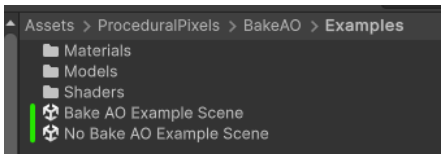
Textures baked with Bake AO are compatible with all platforms.

## Downloading and importing Bake AO from the Unity Asset Store

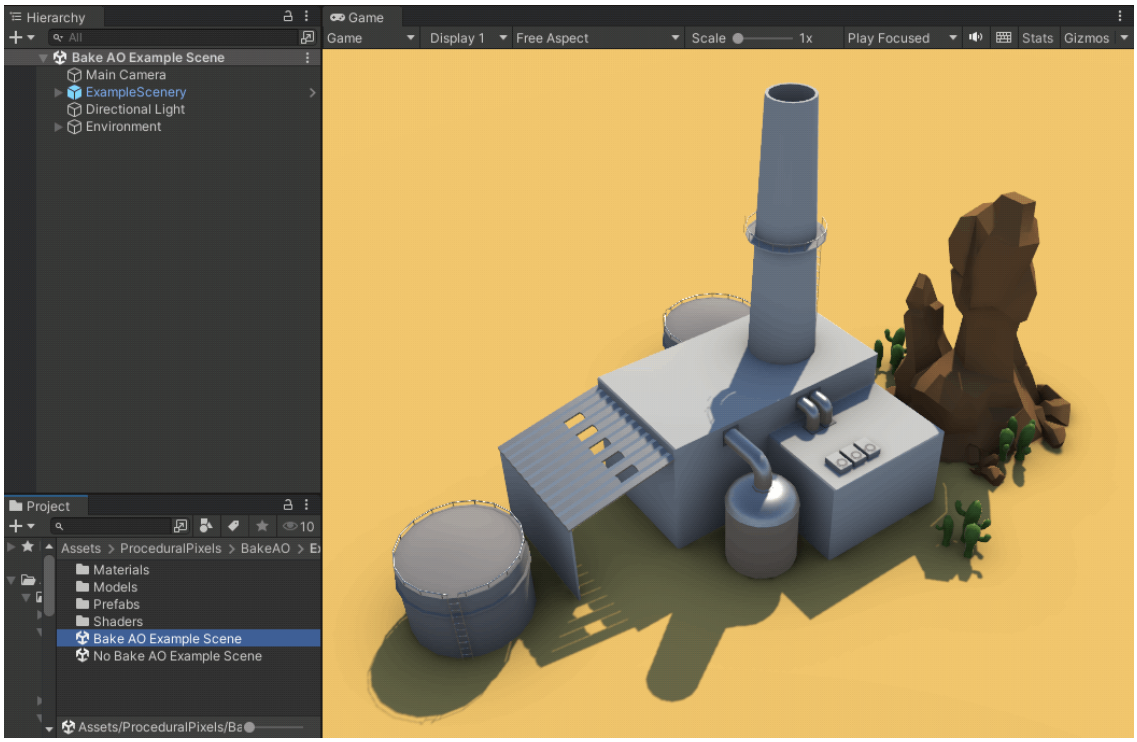
1. Download the Bake AO plugin from the Unity Asset Store.
2. When importing, Unity will display a list of files. First-time users should click "Import." The files will be in the "Assets/Procedural Pixels/Bake AO" directory but can be moved later.
3. Once the import process is finished, you will find the "Window/Procedural Pixels/Bake AO" menu in your project.




Installing the Examples will add two simple scenes for experimenting with Bake AO options.




Switching between the scenes shows the visual differences. Feel free to experiment with them.

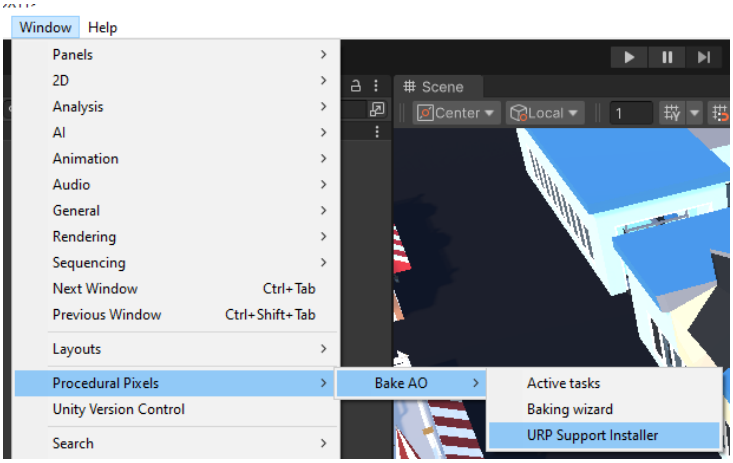



### Installation for URP - Universal Render Pipeline

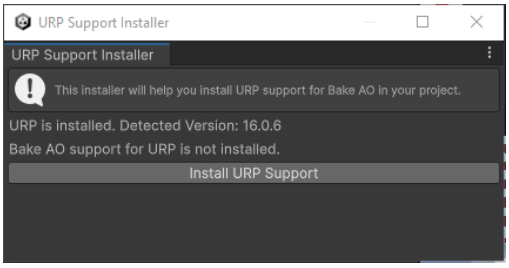
Bake AO integrates with Universal Render Pipeline, making Bake AO Component work with URP.

If your project uses URP and you want to use Bake AO component features, do the following steps:

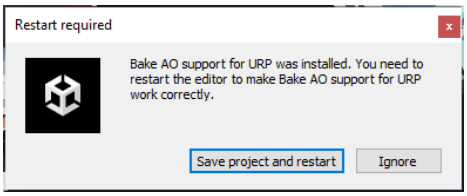
1. Open BakeAO URP integration installer using `Window->Procedural Pixels->Bake AO->URP Support Installer`




2. The Installer window will check if the project currently imports the URP. It will display an "Install URP Support" button. You can click the button to install Bake AO Integration for URP.




3. After successful installation, the URP Support Installer window will ask you to save and reopen a project. Reopening the project is needed for integration to work correctly.




After installing URP integration, you will find new shaders in the project.

- `Procedural Pixels/Bake AO - URP/Lit`
- `Procedural Pixels/Bake AO - URP/Simple Lit`
- `Procedural Pixels/Bake AO - URP/Complex Lit`
- `Procedural Pixels/Bake AO - URP/Unlit`

Those shaders work the same as URP shaders but have an additional Bake AO section and work with the Bake AO component.

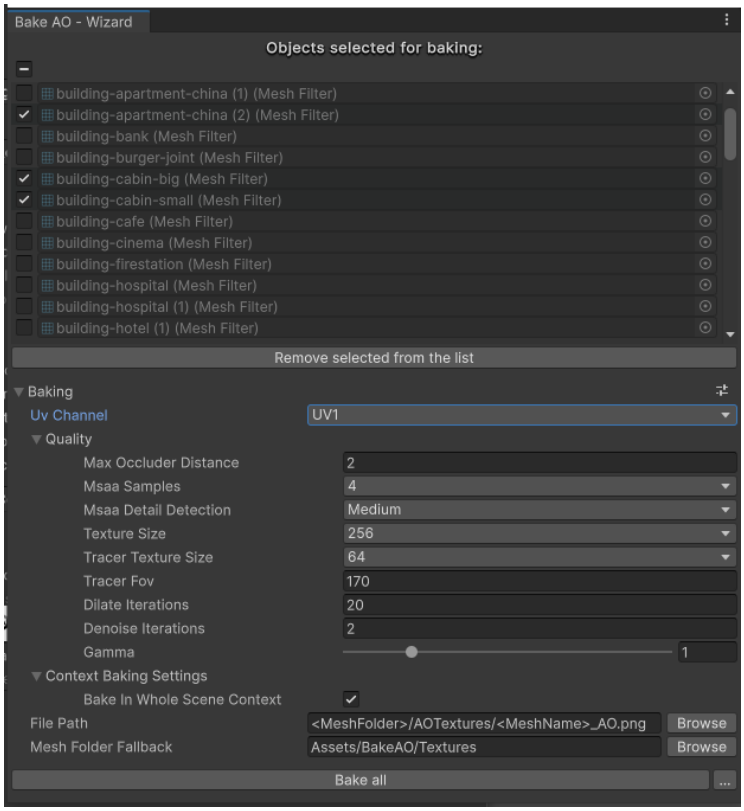
**You need to use those shaders instead of URP shaders to make BakeAO work.**

**All custom shaders also need to be modified appropriately to work with BakeAO.**

See Custom Shaders section to find out more about shader requirements.

## 3. Baking Wizard

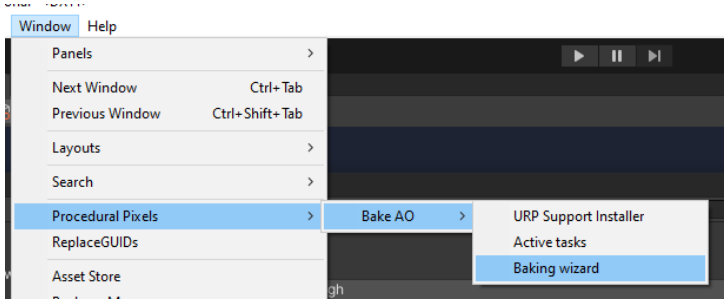
The Baking Wizard simplifies batch baking for multiple models. Customize baking parameters and efficiently queue multiple models for baking at once.




### Opening the Baking Wizard

To open the Baking Wizard, navigate to the Unity Editor's menu:

**Window -> Procedural Pixels -> Bake AO -> Baking Wizard**

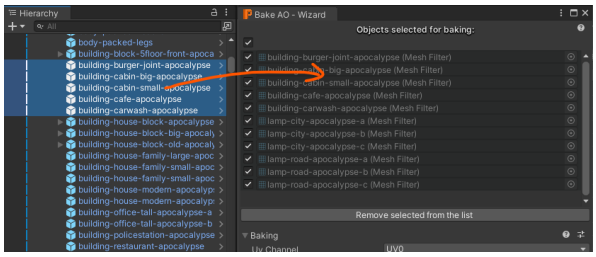
1. 

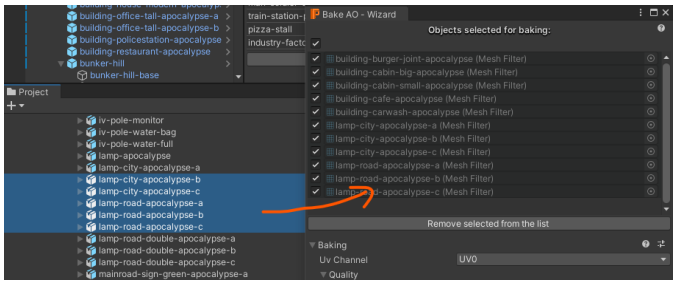

Upon opening the Baking Wizard, you'll see an empty window where you can drag and drop objects from hierarchy and project files to enqueue them for baking.

### Managing Objects in the Baking List

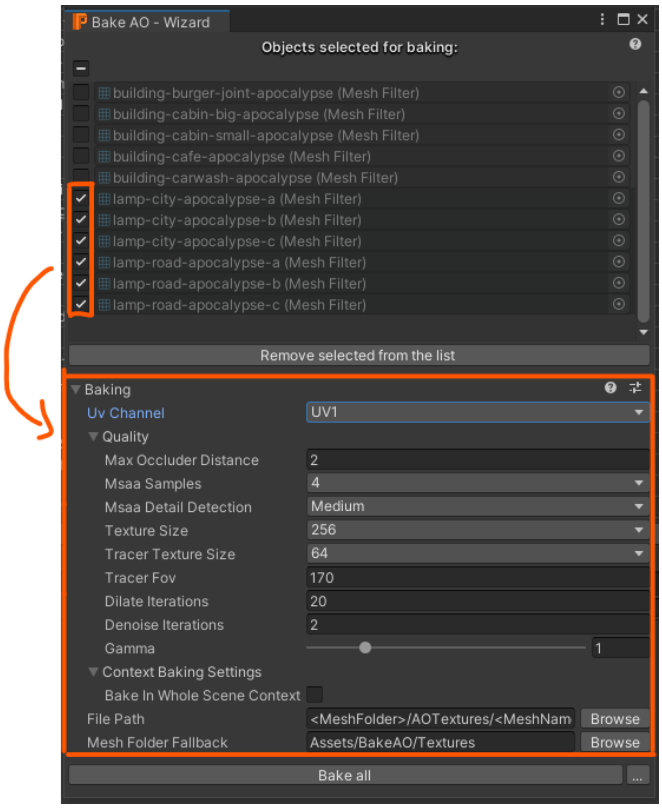
You can add models to the baking list by dragging and dropping them into the "Drag files here" field. The Baking Wizard supports these object types:

1. Model assets from the project assets.
2. Folder from the project assets - all models inside the folder will be added to the list. Bake AO searches for models recursively.
3. GameObject from the scene hierarchy - the plugin will recursively search for all the MeshRenderers and SkinnedMeshRenderers in this object and its children.
4. Components from the scene hierarchy - works similarly to dragging a GameObject.




Each list element includes a toggle on the left for selecting the object. Below you can see the baking parameters for the selected objects. Bottom section of the window works like an "inspector" for baking parameters.




### Baking

The baking parameters section is below the list. The settings here are specific to the selected models, allowing adjustments for each model individually. Use the Shift key to select a range of models from the previously selected to the current one.

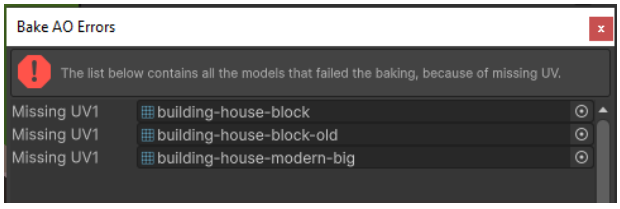
The baking parameters are common to the Baking Wizard and the BakeAO components. Detailed explanations of each parameter are provided here.

Once you have selected the models and customized the baking parameters, **click the "Bake all" button to start the baking process for all models in the list**.

**Context Baking Settings are relevant only for the objects dragged from the scene.** Objects from the assets are always baked with no context.

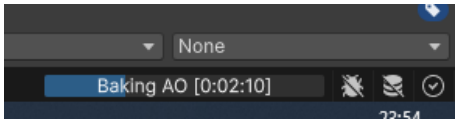
### Handling Baking Errors

If there is an issue with some meshes during baking, a separate window will display the problematic meshes and the reasons for the failure. Resolve these errors and re-enqueue the baking process. Meshes with errors will stay in the Baking Wizard list, allowing you to retry after fixing the issues (often due to missing UV sets).




### Monitoring Baking Progress

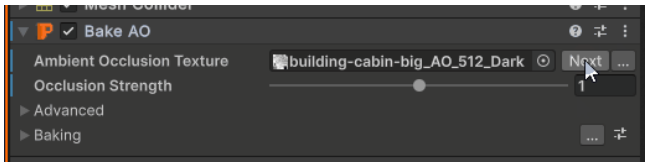
During baking, monitor the progress via the progress bar in the bottom-left corner of the Unity Editor window.




For detailed control over active baking tasks, use the "**Window->Procedural Pixels->Bake AO->Active Tasks**" menu. This window shows all ongoing tasks, allowing efficient management. You can find more information here.

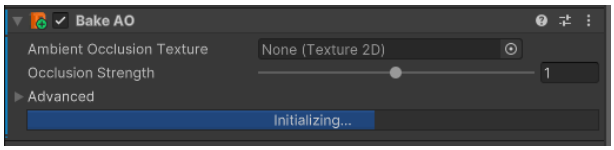
## 4. Bake AO Component

The Bake AO Component lets you customize ambient occlusion textures for individual renderers on the scene. Please attach it below MeshRenderer or SkinnedMeshRenderer to quickly apply AO texture for a specific renderer. You can also bake textures for models using this component.




**Initialization**

You can see the initialization progress bar in BakeAO component, when you use it first time since launching the project. During this time, certain features of BakeAO are not available. Initialization may take some time when your Unity project has many Bake AO textures.Once initialization is complete, all features of the Bake AO Component will be fully accessible.



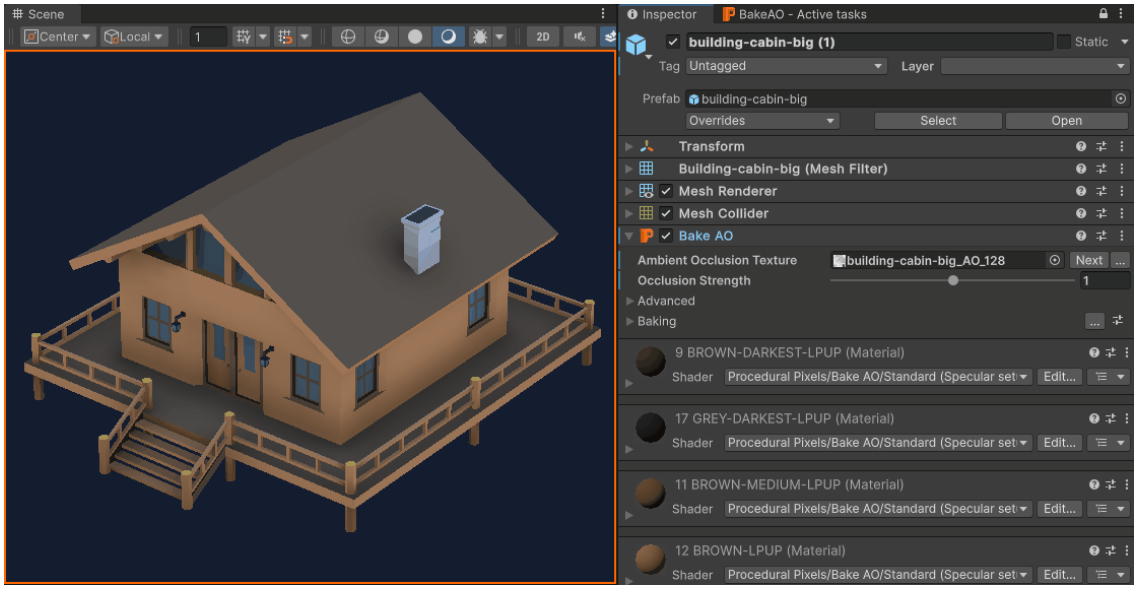

## Basic properties

The Bake AO Component has two primary fields that can be easily set up by using the inspector:

1. **Ambient Occlusion Texture:** This texture2D field references the ambient occlusion source texture. When set, it will override the ambient occlusion texture used by the shaders for the attached model.
2. **Occlusion Strength:** A float value slider ranging from 0.0 to 2.0, the Occlusion Strength parameter controls the ambient occlusion application strength. It will override the occlusion strength parameter in the shaders.

The "Ambient Occlusion Texture" field also includes two essential buttons:

- **Next** - Clicking this button will cycle through all the textures baked for the mesh displayed by the attached MeshRenderer or SkinnedMeshRenderer.
- **...** - Clicking this button will display a list of textures baked for this model, allowing you to select any of them.




## Advanced Settings

The Bake AO Component also features an "Advanced" foldout that offers additional customization options:

- **Occlusion UV Set**: The user can select the UV set that should be used to sample the ambient occlusion texture.
- **Apply Occlusion Into Diffuse**: A boolean toggle that, when enabled, multiplies albedo texture colors with ambient occlusion, enhancing the AO effect for light sources. This setting provides a more stylized look that can benefit specific game aesthetics.

## Baking

There is a "**Baking**" foldout. When expanded, this section displays all the baking parameters. Foldout concludes with the **"Bake AO" button**. Clicking this button will enqueue the baking of the ambient occlusion texture for the model displayed by the attached MeshRenderer or SkinnedMeshRenderer.

The Baking foldout properties doesn't work like usual component properties. Those properties are not saved in the component; it always shows the latest used baking properties (or default properties if you haven't baked anything since theproject launched).

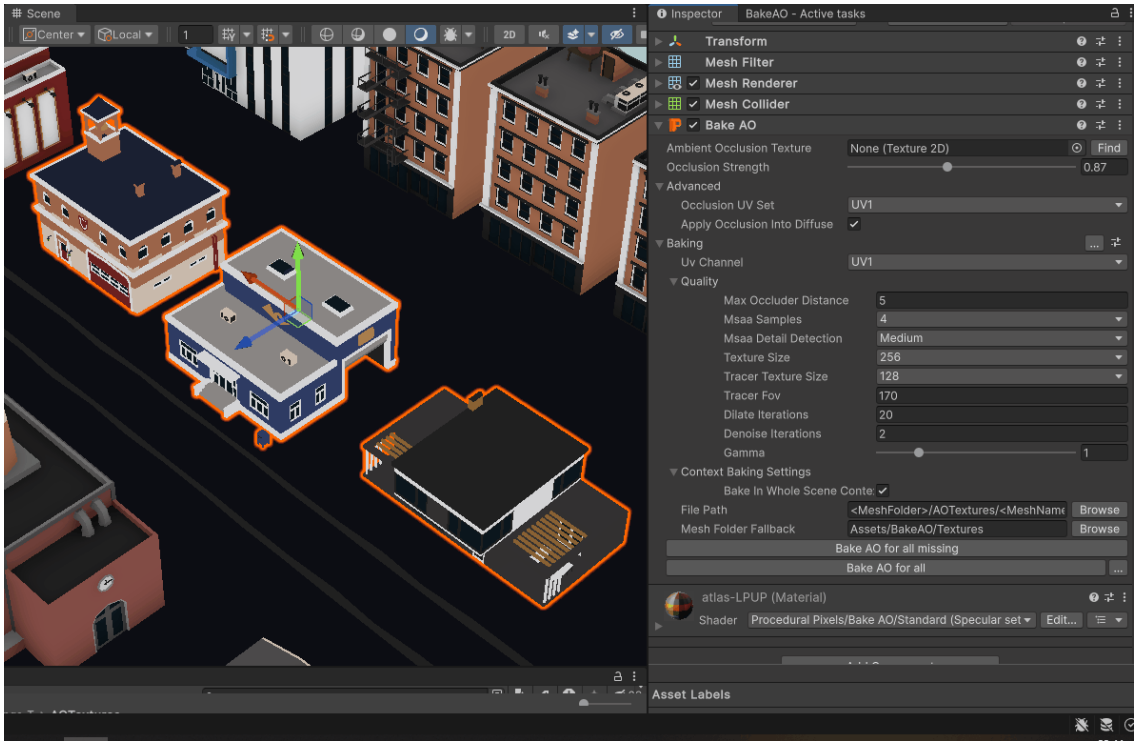
**Multi-Selection Support**

Bake AO provides multi-selection support. When multiple objects with Bake AO Components are selected, the component allows you to simultaneously initiate baking for all selected objects. Two buttons for baking are provided:

- **Bake AO for all missing**: This button enqueues baking only for the components that don't have an AO texture set.
- **Bake AO for all**: Clicking this button enqueues baking for all selected Bake AO components.

Additionally, when multiple Bake AO components are selected, the "Ambient Occlusion Texture" field displays a single button:

- **Find:** This button lets you find AO textures for the missing components. There is no option to cycle through existing textures.




### Custom Shaders

Bake AO uses MaterialPropertyBlock to set up parameters for shading. These parameters include:

- **_OcclusionMap**: This shader property holds the assigned texture that will be bound to the ambient occlusion map.
- **_OcclusionStrength**: A shader uniform that contains the occlusion strength value.
- **_AOTextureUV**: An advanced setting containing the UV set's index to sample the AO texture.
- **_MultiplyAlbedoAndOcclusion**: This parameter will have a value of 1 if albedo multiplication is enabled and 0 otherwise.

Developers can utilize these parameters to create custom shaders for unique visual effects. For more information about custom shaders, see this section.

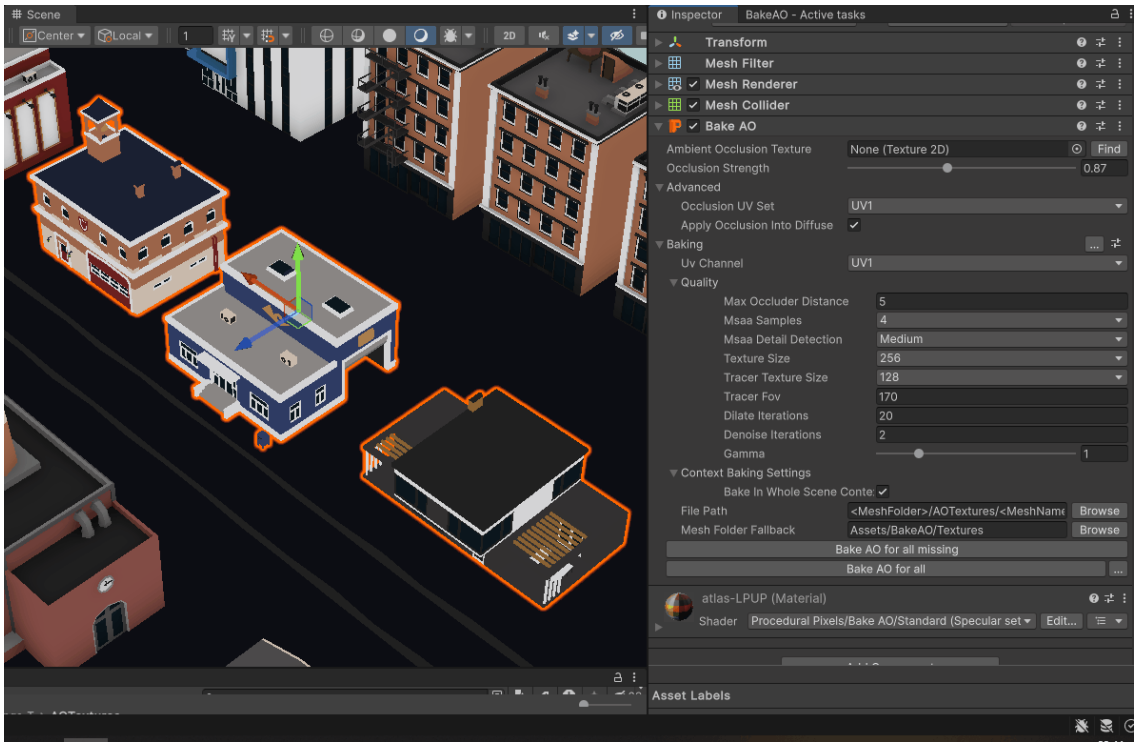
### Inherit from the "GenericBakeAO" Class

Developers can inherit from the "GenericBakeAO" class to provide custom behavior for Bake AO components, adding further flexibility to the plugin's functionality.

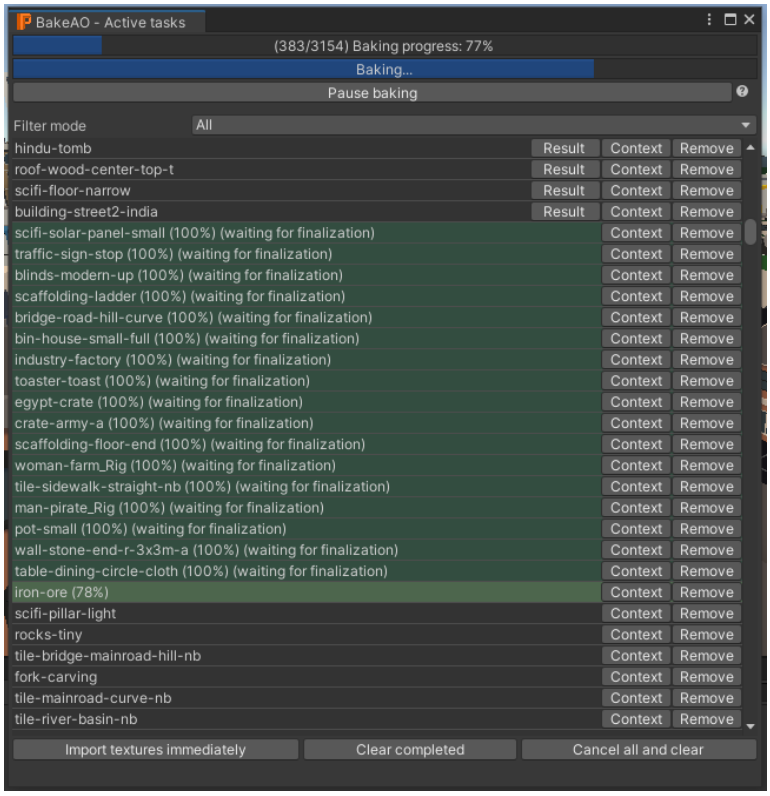
To customize the property block setup, developers should override the `protected SetupPropertyBlock(MaterialPropertyBlock propertyBlock)` similarly as the BakeAO class does and other methods if needed.

## 5. Managing Baking Tasks

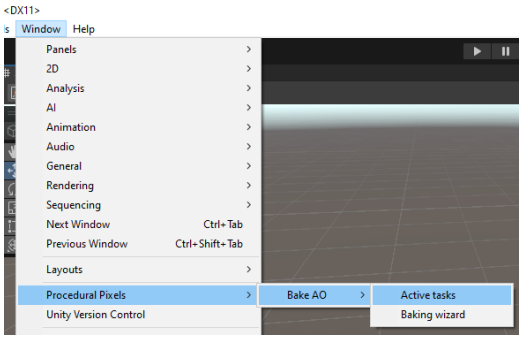
The Bake AO plugin provides a system for managing baking tasks, allowing users to monitor and control the queue of ongoing tasks. This chapter outlines the functionalities that assist users in overseeing and reviewing baking processes.




## Monitoring and Controlling the Baking Queue




To keep track of the active baking tasks, you can open the "Active Tasks" window by navigating to "W**indow -> Procedural Pixels -> Bake AO -> Active Tasks**" in the Unity Editor.




The "Active Tasks" window displays the list of objects currently enqueued for baking. You can monitor the progress of each baking task and see information about the models being processed.

At the upper part of the window, two progress bars provide an overview of the overall baking progress and the progress of the current baking texture.

Each element in the list includes the name of the baking task, which typically corresponds to the name of the object or mesh that initiated the baking. Additionally, each baking task has two buttons: "Context" (to focus Unity Editor selection on the object associated with the baking task) and "Remove" (to remove the baking task from the queue). An additional "Result" button is available for completed tasks, allowing you to select and view the texture baked by the plugin.

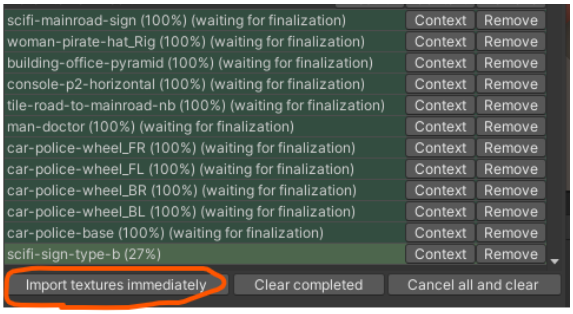
You can pause and resume baking using a button below the progress bar.

### Task Progress and Completion

Each task in the list displays its progress. Completed tasks will have a "**(completed)**" suffix. Tasks that are currently being processed will show the percentage of progress.

Some tasks may show "**(wait for finalization)**" as their status. This indicates that the baking process is complete for these tasks, but the textures have not been imported yet. **Texture import is deferred to prevent interruptions during your workflow. Instead, the textures are imported only when the baking memory exceeds the specified memory limit, before switching into play mode or script recompilation.**

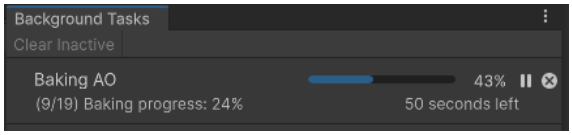
You can force immediate texture import by using a dedicated button.




### Built-in Progress Bar

You can track the baking progress using a progress bar. As soon as you start baking tasks, the progress bar appears at the bottom-right corner of the Unity Editor. It keeps you informed about how the baking is coming along.

For more details, click on the progress bar. You can pause or resume the baking using the pause icon or cancel all baking tasks with the 'x' icon.




## 6. Baking Parameters

The baking parameters determine the texture resolution, quality, and style during the ambient occlusion texture baking process. They can be accessed in both the Baking Wizard window and the Bake AO Component inspector.

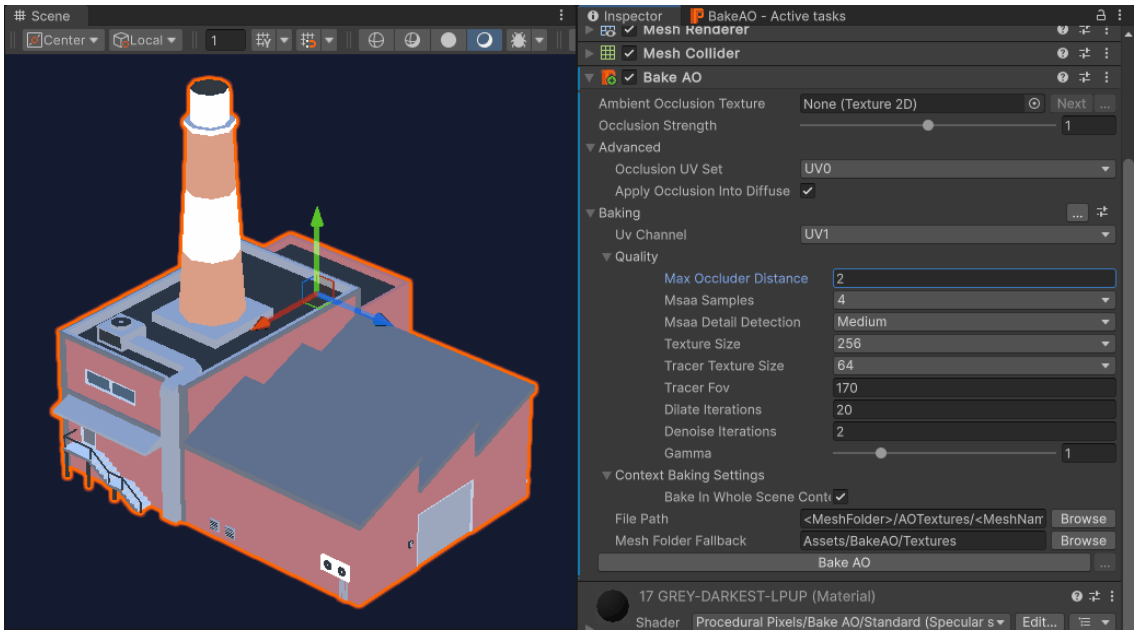
To view the baking parameters, navigate to the "Baking" foldout in either the Baking Wizard window or the Bake AO Component inspector. Click on the arrow to expand the foldout and reveal all the parameters.

### UV Channel

The "UV Channel" parameter determines which UV set from the mesh will be used to bake the ambient occlusion texture. It is crucial to use the same UV set later to apply the baked texture to the model accurately.

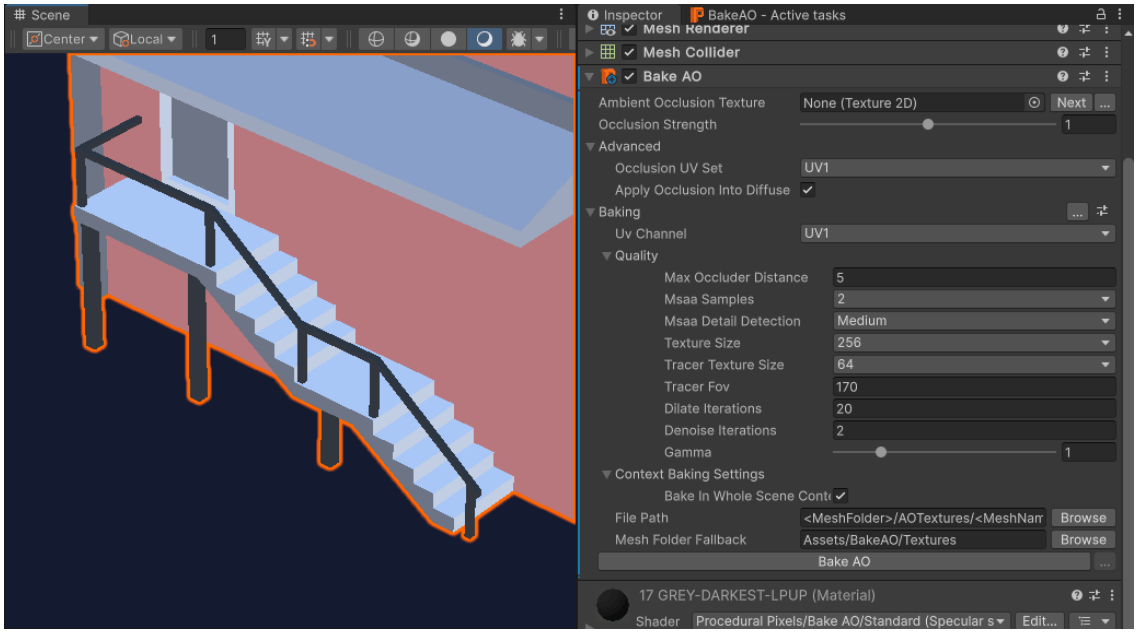
### Max Occluder Distance

This parameter sets the maximum distance to the occluder that will affect the occlusion. Any occluders further than this distance will not influence the texture.




### MSAA Samples

The "MSAA Samples" parameter defines how many more samples the plugin should generate for more detailed parts of the model. For example, a value of 8 means that the plugin will bake the texture at 8 times larger resolution in more detailed areas and then average the results for the final pixel.'



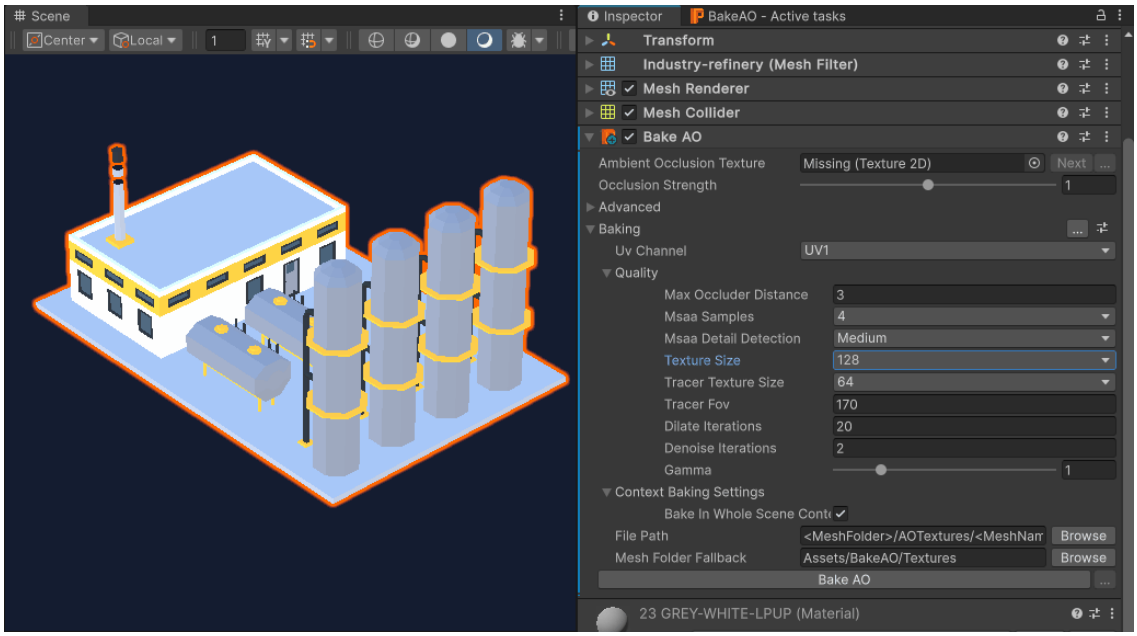

### MSAA Detail Detection

The "MSAA Detail Detection" parameter controls the sensitivity of the detail detection. This parameter is more useful when baking for custom UV, not Unity-generated. It offers three options:

- Soft: Faster baking times, but may produce more visual artifacts.
- Medium: Balances baking speed and quality with rare visual artifacts.
- Hard: Longest baking times, providing the highest baking quality.

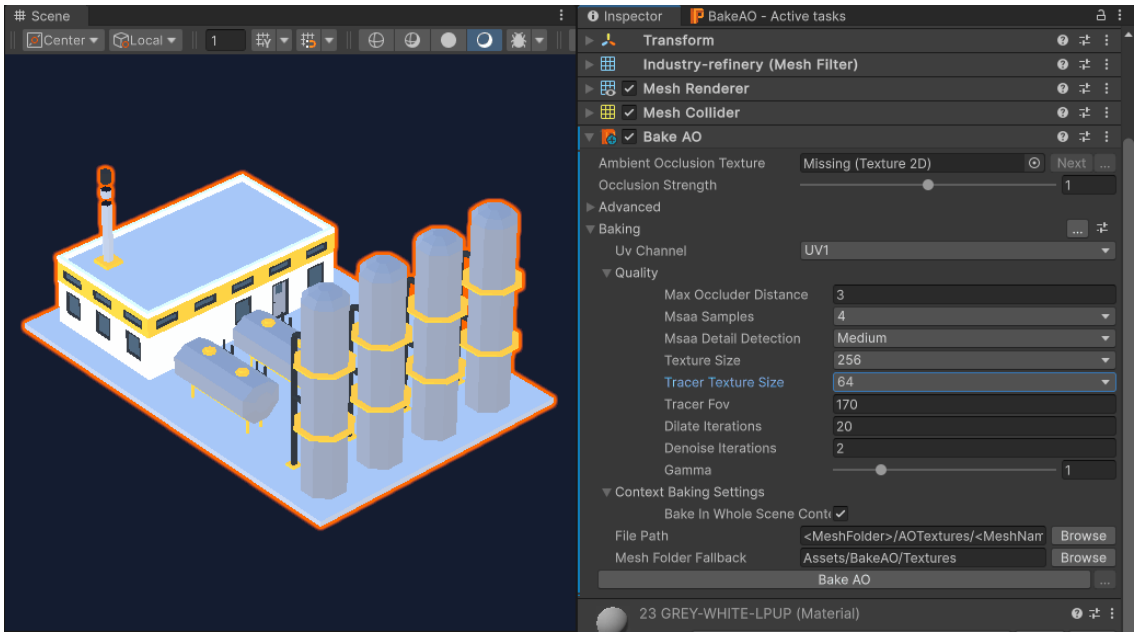

## Texture Size

The "Texture Size" parameter determines the resolution of the baked ambient occlusion texture.



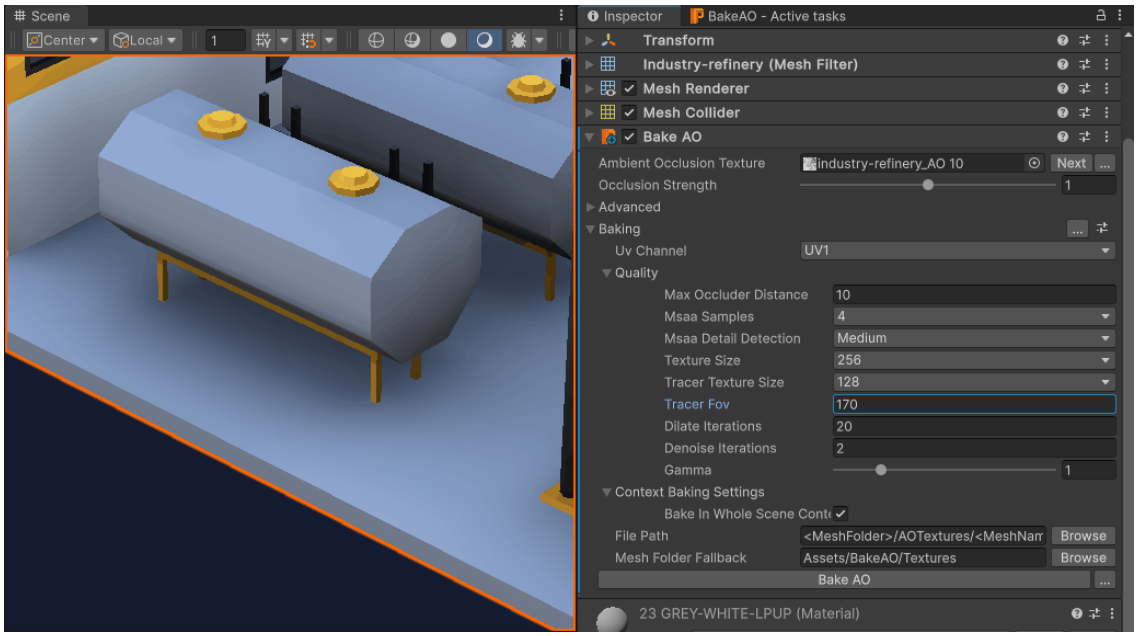

## Tracer Texture Size

The "Tracer Texture Size" parameter specifies how many samples should be collected for each pixel of the resulting texture. Typically, 64x64 or 128x128 provides the best trade-off between bake time and quality. This parameter usually only affects baking times on older or weaker GPUs. Often, there is no point in using settings other than 128x128 or 64x64.



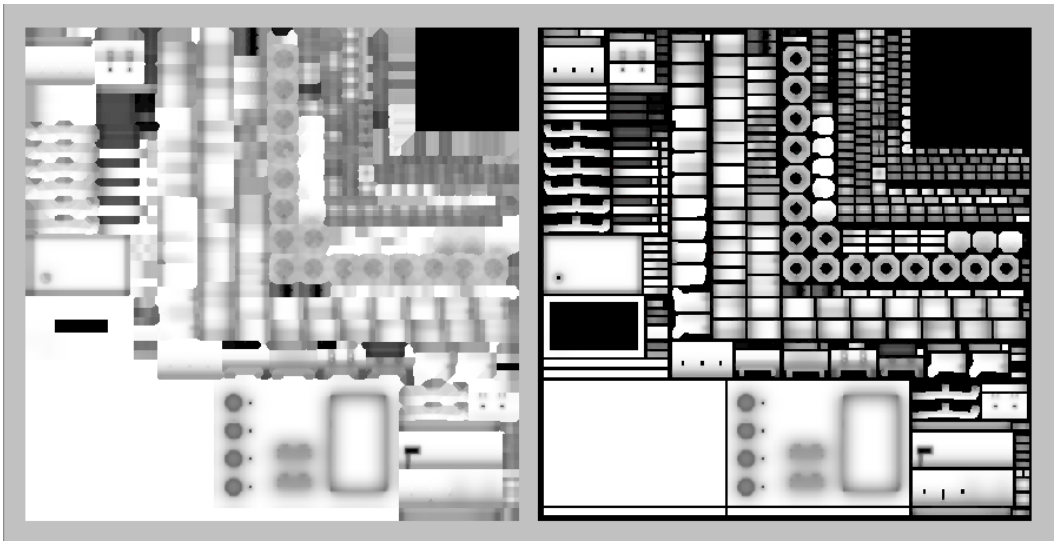

### Tracer FOV

Lower values of the "Tracer FOV" parameter will limit the angle of ray generation, producing more stylized results.




### Dilate Iterations

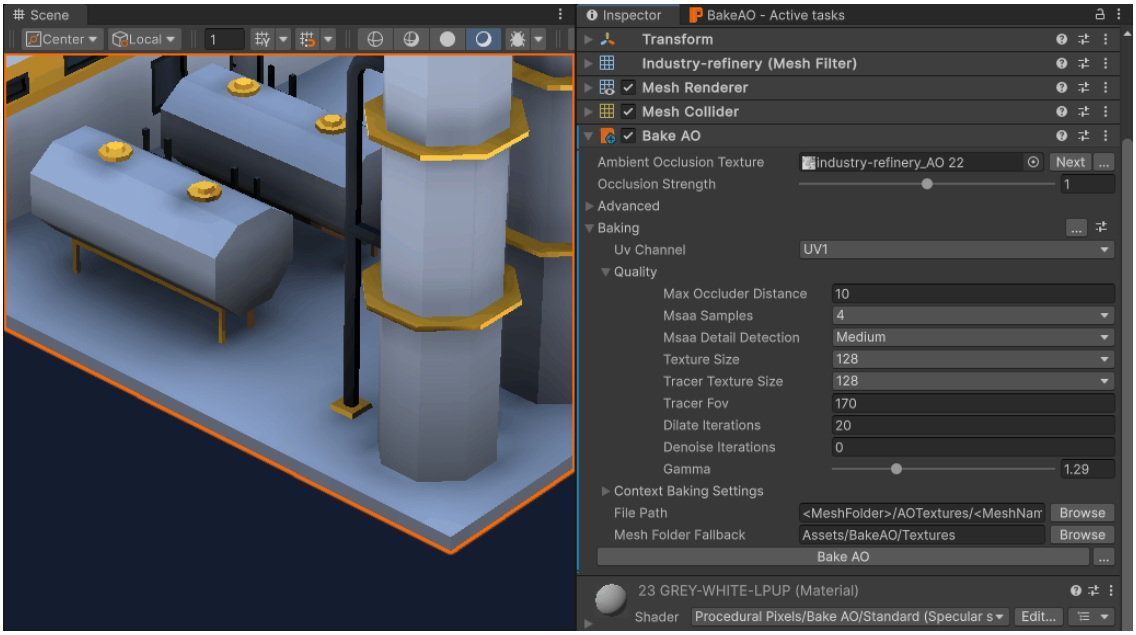
The "Dilate Iterations" parameter defines how often the texture content should be dilated. Dilating the texture content expands it to the pixels that were not directly baked, which can reduce some artifacts.



Comparison of 20 iterations of dilatation (left side) and two iterations (right side)

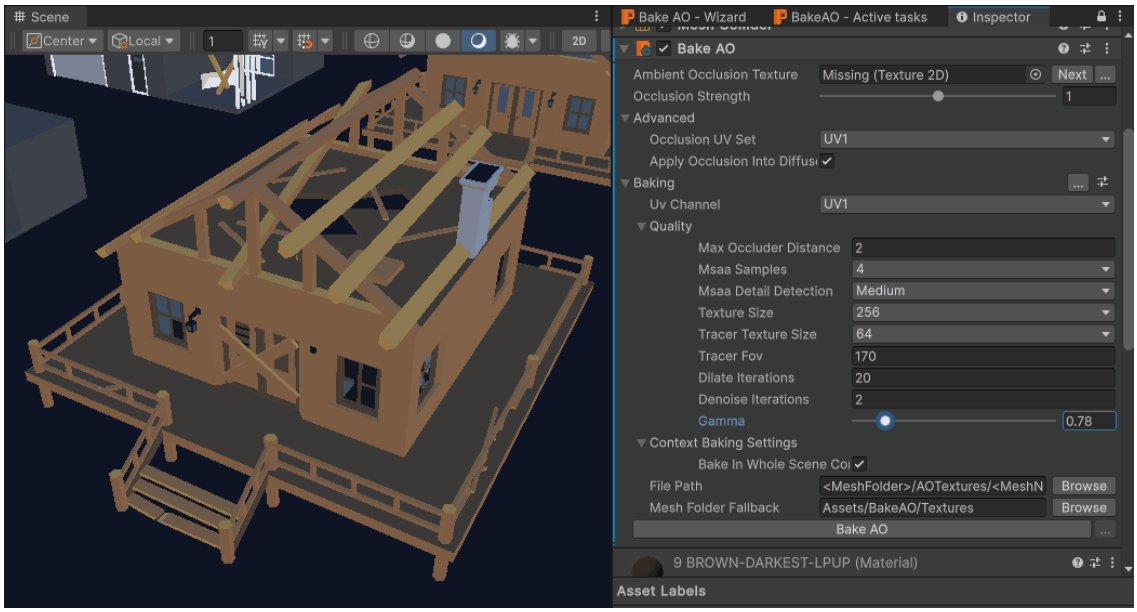
### Denoise iterations

Denoising makes the baking results smoother. Usually, 1 or 2 denoising passes are enough.



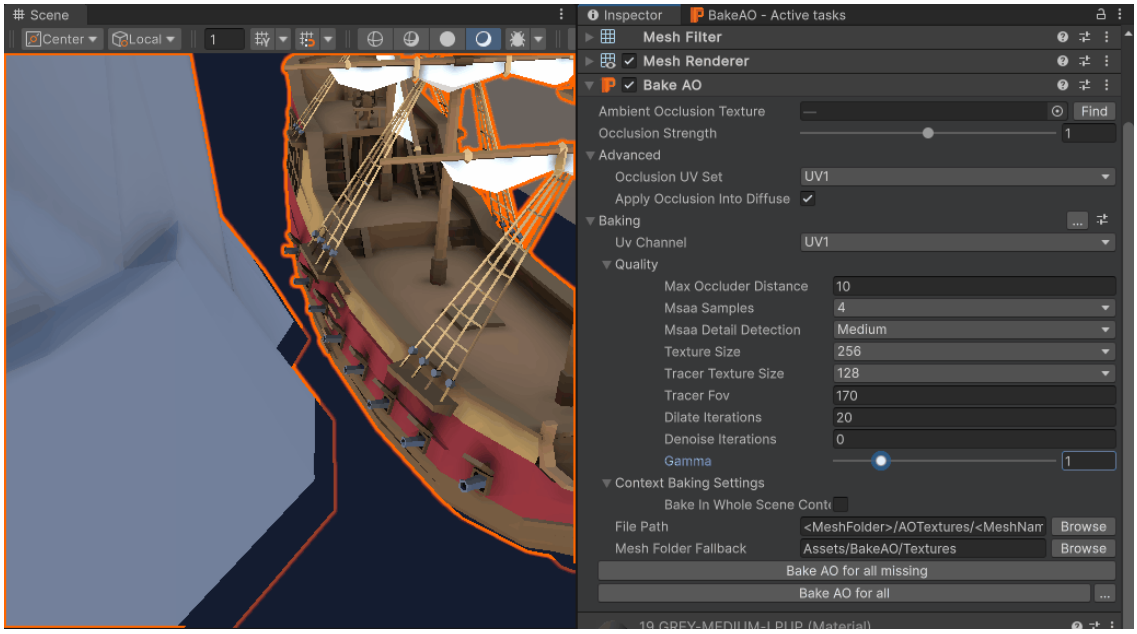

### Gamma

The "Gamma" parameter affects the grayscale curve of the final texture. Larger values make the baked texture darker, while lower values make it brighter.

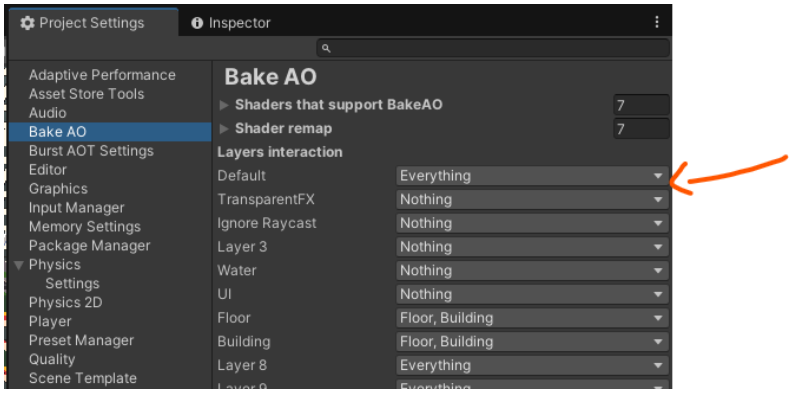



### Bake In the Whole Scene Context

When enabled, the entire scene around the baked model will act as an occluder during baking. Turning off this option makes the baked object itself the only occluder.




You can modify which object layers should be used as occluders in **Project Settings / Bake AO** menu. Here you can set occluder set for each layer separately.




In example setup above:

- For objects in "Default" layer, every other renderer will act as an occluder.

- For objects on "TransparentFX" layer, there will be no occluders (except the baked object itself)
- For objects on "Building" layer, only renderers on "Floor" and "Building" layers will act as an occluder.

### File Path

The "File Path" parameter allows you to specify where the texture will be saved. You can use the "Browse" button to select a custom location. The file path can contain wildcards:

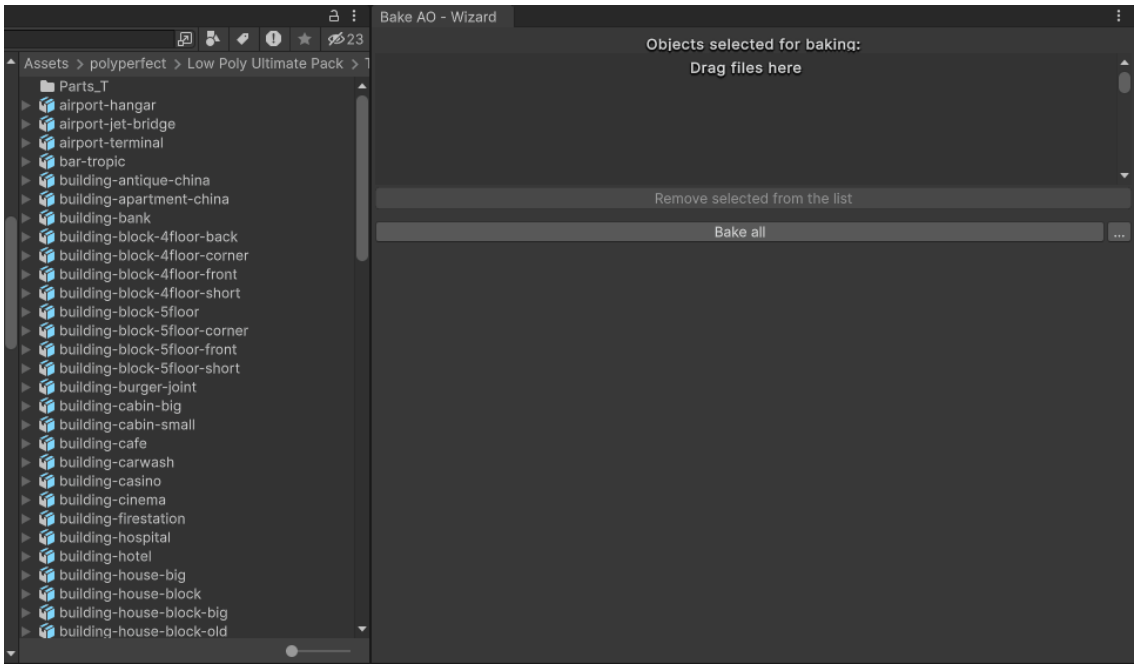
'**<MeshFolder>**': Replaced by the path to the folder where the model asset is located.

'**<MeshName>**': Replaced with the name of the Mesh asset.

'**<AssetName>**': Replaced with the asset name containing the mesh. This can differ from the mesh name, if it is a subasset.

'**<GameObjectName>**': Replaced by the name of the GameObject that initiated the baking. This only works when triggered from the Bake AO Component.

Example: *"/AOTextures/_AO.png"* - In a folder when the baked model is saved, create a folder "AOTexture".




### Fallback Mesh Folder

The "Fallback Mesh Folder" parameter allows you to specify the mesh folder path when the baked model is not a project Asset (e.g., procedurally generated or from an external package).

### Other baking parameters properties

1. **Baking Parameter Persistence** - Bake AO remembers the last-used baking parameters, so when you open different Bake AO Component inspectors, **it will display the last used values**. However, these parameters are **not saved within the component but stored inside a texture asset**. When you open a Bake AO Component inspector for the first time, the baking parameters will show default values.
2. **Fetch from Texture (Bake AO Component only)**. You can fetch the baking parameters for the currently assigned texture. Click the "..." button and select "Fetch from texture" to apply the parameters from the assigned texture.
3. **Unity Presets** - The baking parameters support Unity presets, allowing you to use predefined parameter presets or create your own for different baking scenarios. How to use presets: https://docs.unity3d.com/Manual/Presets.html

## 7. Custom Shaders

Bake AO Component requires to use shaders prepared to work with this component. This means that you need to use shaders included with Bake AO, or adjust shaders you're already using in your project. This section of the documentation will explain how to apply the modifications.

Shaders included with BakeAO work the same way like the ones provided by Unity, but have additionall properties, allowing you to apply AO texture to diffuse color, and select UV set for sampling occlusion texture.

Usage of included shaders or modification of custom shaders are not needed if you don't use BakeAO component.

## Included Shaders

**Bake AO comes with those shaders for built-in shader replacement:**

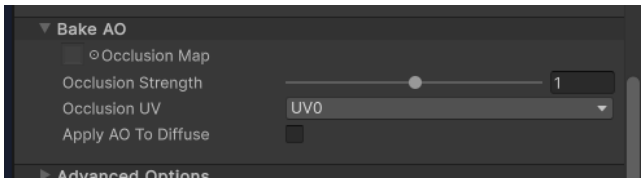
- `Procedural Pixels/Bake AO/Standard`
- `Procedural Pixels/Bake AO/Standard (Specular setup)`

Those shaders add two fields to the material.


**And those shaders for URP shader replacements:**

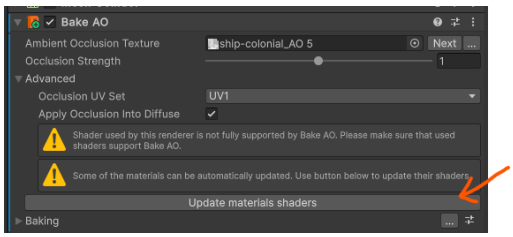
- `Procedural Pixels/Bake AO - URP/Lit`
- `Procedural Pixels/Bake AO - URP/Simple Lit`
- `Procedural Pixels/Bake AO - URP/Complex Lit`
- `Procedural Pixels/Bake AO - URP/Unlit`

Those shaders work the same as URP shaders but have an additional Bake AO section.




## Automatic shader change in the materials

Bake AO can suggest and replace shaders used in your materials to make it's all features available. If the object is rendered by a shader not supporting Bake AO features, the Bake AO Component will display the message in the "Advanced" section. If the material can be updated with the supported shader, you can use a button to update it.




## Bake AO shader properties

Bake AO uses MaterialPropertyBlock to set shading parameters. These parameters include:

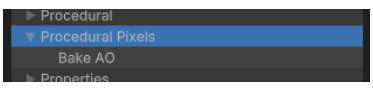
- **Texture2D _OcclusionMap**: This shader property holds the assigned ambient occlusion texture.
- **float _OcclusionStrength**: A shader uniform that contains the occlusion strength value.
- **float _AOTextureUV**: UV set's index to sample the AO texture.
- **float _MultiplyAlbedoAndOcclusion**: This parameter will have a value of 1 if albedo-occlusion multiplication should be performed, 0 otherwise.

Renders that use Bake AO need to use shaders that contain those properties.

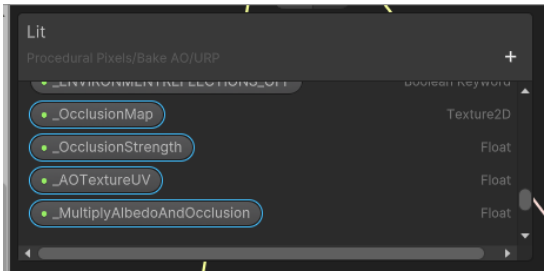
## Adjusting shader graph

To make custom shaders made with shader graph support BakeAO, follow those steps:

1. Find the **Procedural Pixels / Bake AO** node in the shader graph nodes, and add it into the graph.

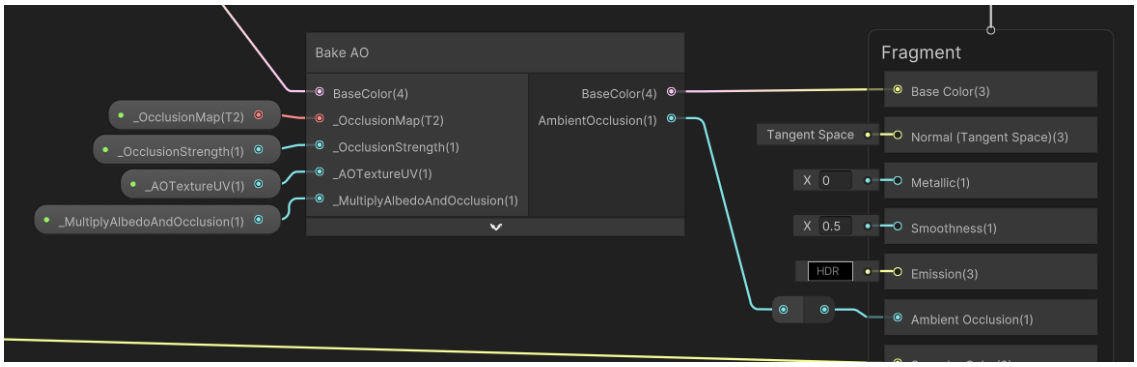



2. The node requires the setup of properties in the Blackboard. Copy those properties from the example shader (**Bake AO Lit Shader Example. shadergraph**) and paste them into your shader. As in the example above, you must connect those properties to the Bake AO node.




3. Connect inputs of the node with the created properties.
4. To the **BaseColor** input, connect the value you would usually output into the **Base Color** of the master node.
5. Connect **BaseColor** and **AmbientOcclusion** output of Bake AO node into the respective shader graph outputs.

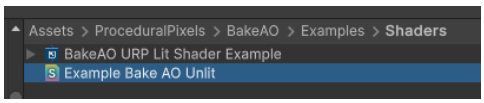
Final setup should look like this.




Remember to add the shader to shaders supported by Bake AO in **Project Settings / Bake AO**.

### Adjusting text shaders

The Bake AO plugin contains an example of a text shader that supports the rendering with the Bake AO component. The shader code includes comments that explain how to make your shader compatible with Bake AO.




If you want to modify the existing shader in code to support Bake AO features, follow the steps below:

1. Include BakeAO_CG.cginc (if your shader is written in CG) or BakeAO_HLSL.hlsl (if your shader is written in HLSL) library into your shader. The include should point to the exact file (relative or absolute file path), so if you moved the package after it was installed, please modify this path accordingly:

   #include "Assets/ProceduralPixels/BakeAO/Runtime/Shaders/ShaderLibrary/BakeAO_CG.cginc"

2. Add the UV sets you want to use to the vertex attributes. Usually, you want to use 2 or 3 UV sets; more of that is often unnecessary.

```
struct appdata
{
    float4 vertex : POSITION;
    // Include uv0, uv1, uv2 and uv3 in vertex attributes.
    float2 uv0 : TEXCOORD0;
    float2 uv1 : TEXCOORD1;
    float2 uv2 : TEXCOORD2;
    float2 uv3 : TEXCOORD3;
};
```

3. Add fragment shader interpolator for the UV that will be used for sampling ambient occlusion texture:

```
struct v2f
{
    float2 aoTextureUV : TEXCOORD0; // Include uv for ambient occlusion texture in
```

```
the interpolator
    float4 vertex : SV_POSITION;
};
```

4. You need to get the AO texture UV in the vertex shader and pass it to the interpolator for the fragment shader.

   Add this line in vertex shader code. `BakeAO_FilterAOTextureUV` will pick up the UV set for AO texture.

   ```
   o.aoTextureUV = BakeAO_FilterAOTextureUV(v.uv0, v.uv1, v.uv2, v.uv3, _AOTextureUV);
   ```

5. Use `_OcclusionMap` texture and uv set added in interpolator to sample the occlusion in UV shader. Use `BakeAO_ApplyOcclusionStrengtht`function to apply the occlusion strength parameter to the occlusion.

   ```
   // sample the ambient occlusion texture.
   float occlusion = tex2D(_OcclusionMap, i.aoTextureUV).r;

   // Apply the occlusion strength
   occlusion = BakeAO_ApplyOcclusionStrength(occlusion, _OcclusionStrength);
   ```

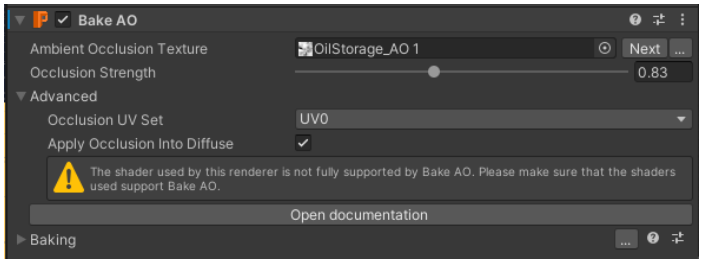
6. Use the occlusion value according to your needs. If you're using some lighting method, you can probably pass the occlusion value to the lighting. If you're not using any lighting function that can take occlusion as an argument, environment lighting color should be multiplied by this occlusion value. Your shader can return the color multiplied by the occlusion value.

7. To use a feature to apply occlusion into a diffuse color, use BakeAO_ModifyBaseColor, the value in your shader code.

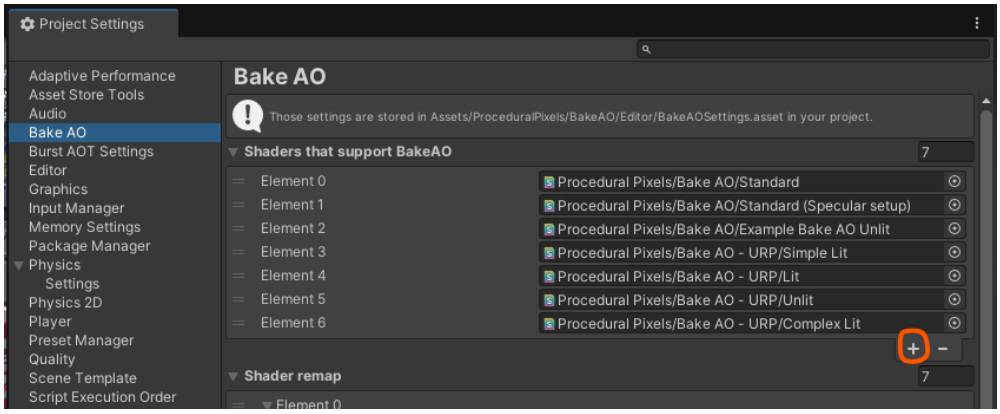
   ```
   surfaceAlbedo = BakeAO_ModifyBaseColor(surfaceAlbedo, occlusion,
   _MultiplyAlbedoAndOcclusion);
   ```

### Marking shader as supported by BakeAO

Custom shaders that support BakeAO properties need to be additionally configured for BakeAO, otherwise, BakeAO will display a warning in the inspector.

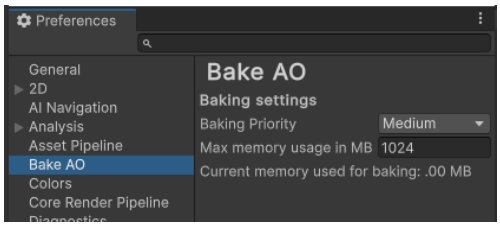



All supported shaders should be added to the **Project Settings / Bake AO**, which are stored with appropriate scriptable object in the project. This will mark the shader as supported by the plugin, and the Bake AO component will not display any warning when your shader is used.




## 8. Bake AO Preferences

Bake AO Preferences allow you to modify specific settings that impact the baking process and editor experience. To access the Bake AO Preferences, click "Edit" in the Unity Editor menu and select "Preferences." Find and click on the "Bake AO" tab in the Preferences window.




### Memory Limit

This parameter allows you to specify the maximum amount of memory (in megabytes) the plugin can use during baking. By adjusting this value, you can control the frequency of texture imports in the Unity Editor. During baking, Bake AO will import the textures only if the memory limit is exceeded.

A lower memory limit results in more frequent texture imports and reduced memory usage.

A higher memory limit will reduce the frequency of texture imports, resulting in a smoother editor experience during baking tasks.

### Setting Up Baking Priority

This setting determines the resources allocated to the baking process, directly influencing the time to complete baking tasks.

Like modifying memory limits, you can find the "Baking Priority" setting within the Bake AO Preferences tab. The baking priority options include:

- **Very Low -** slowest baking times.
- **Low**
- **Medium -** you can expect the editor to operate around 60FPS
- **High -** you can expect the editor to operate at 30FPS
- **Very High -** fastest baking times, but the editor will be very laggy during baking.

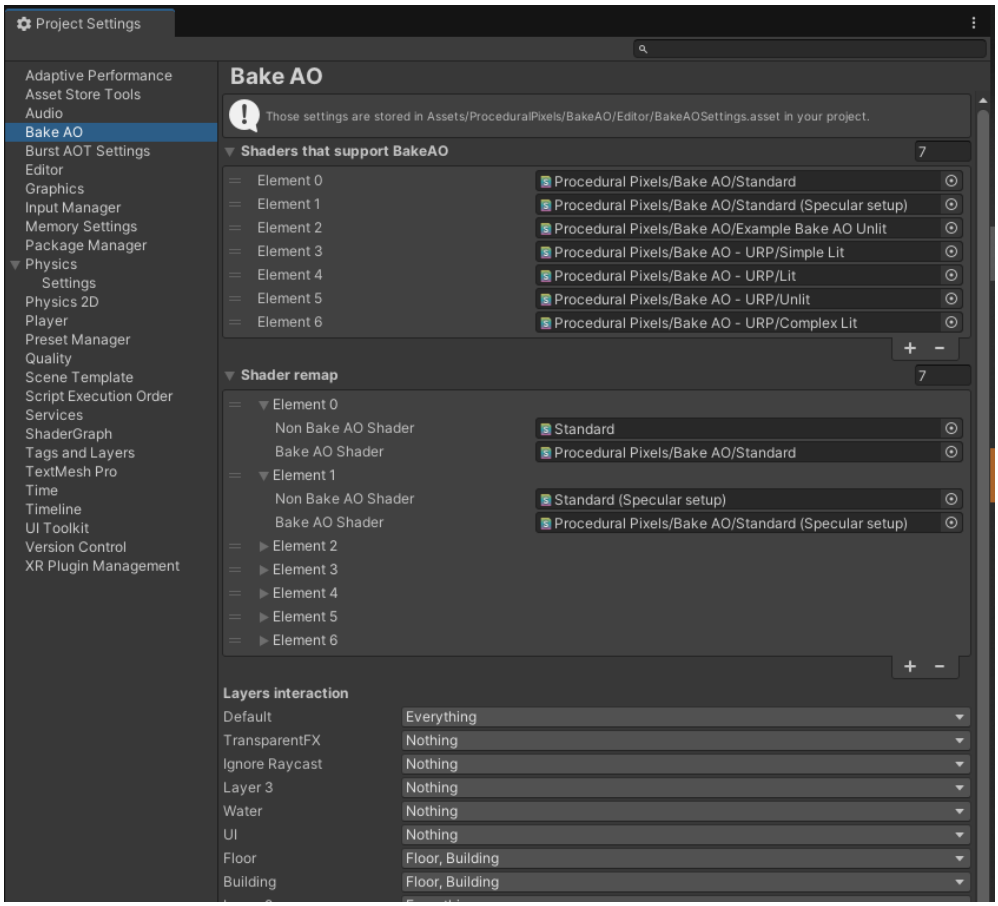
The choice of baking priority depends on your specific requirements and preferences. If you want to continue working on your project with minimal disruption during baking, consider selecting "Low Priority." If you prefer quicker baking results and don't mind a less responsive editor during baking tasks, opt for "High Priority."

## 9. Bake AO Project Settings

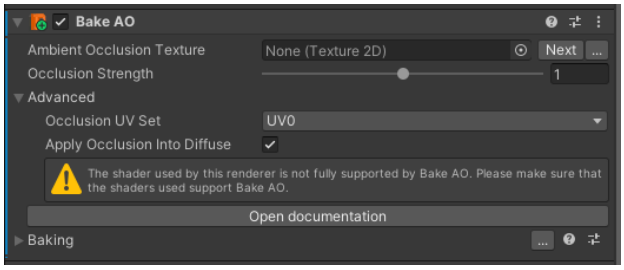
**Menu:** Edit -> Project Settings -> Bake AO

In the **Project Settings** you can find **Bake AO** section that contains per-project settings for Bake AO.

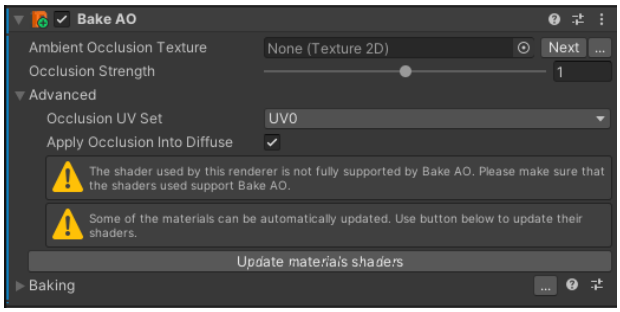
- **Shaders that support BakeAO -** those are the shaders that are explicitly marked as supported by Bake AO. Bake AO will not display any warning in the component inspector, when those shaders are used for rendering. All shaders added here must support Bake AO shader properties, see custom shaders.
- **Shader Remap -** The Bake AO component can update the shaders used in the materials. If you want to instruct BakeAO that some of the shaders can be replaced, put them here.
- **Layers interaction -** defines the interaction between different GameObject layers when selecting occluders from the scene. Here you can set which layers acts as an occluder for a different layer.



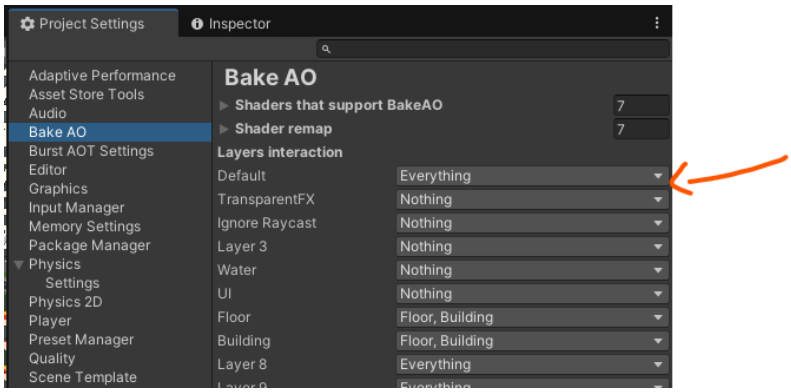

This is a warning you will get when using shader that is not included in the **Shaders that support BakeAO:**




And this is the information when one of the shaders used can be replaced in one of the materials (is included in **Shader remap**):




You can specify which object layers should be used as occluders when baking in scene context. Here you can set occluder set for each layer separately.




## 10. Performance considerations

Bake AO components use MaterialPropertyBlock to setup renderers, and this breaks SRP batching. This is the main performance concern. Basically renderers with BakeAO attached can't be batched by SRP batcher. The performance impact is minimal if you use baked textures in the materials instead of using Bake AO Component.

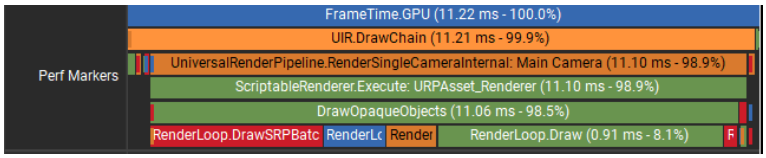
You can see some measures (profiled using NVidia Nsight and Unity Profile Analyzer) related to BakeAO component below.

Scene with 2151 draw calls executed by the graphics API (~2000 models in camera view), URP:

Profiled on i5-10400F, RTX 3060, 16GB RAM

Those are values measured **without** BakeAO.

| API Call | Count | Avg CPU ms | Σ CPU ms | Avg GPU ms | Σ GPU ms |
|---|---|---|---|---|---|
| Map() | 487 | <0.01 | 0.25 | 0.00 | 0.00 |
| IASetVertexBuffers() | 936 | <0.01 | 0.14 | 0.00 | 0.00 |
| PSSetConstantBuffers1() | 2,018 | <0.01 | 0.11 | 0.00 | 0.00 |
| Present() | 1 | 0.10 | 0.10 | 0.00 | 0.00 |
| VSSetConstantBuffers1() | 2,024 | <0.01 | 0.09 | 0.00 | 0.00 |
| IASetIndexBuffer() | 936 | <0.01 | 0.08 | 0.00 | 0.00 |
| Unmap() | 487 | <0.01 | 0.06 | 0.00 | 0.00 |
| PSSetShaderResources() | 331 | <0.01 | 0.06 | 0.00 | 0.00 |
| DrawIndexedInstanced() | 2,024 | <0.01 | 0.05 | <0.01 | 0.98 |
| BeginEvent() | 71 | <0.01 | 0.03 | 0.00 | 0.00 |
| EndEvent() | 71 | <0.01 | 0.03 | 0.00 | 0.00 |

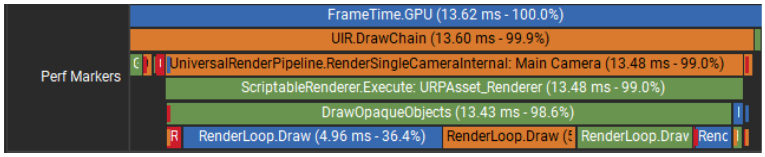



**Marker Details for currently selected range**

| Marker Name | Depth | Media | Media | Mean | Min | Max | Range | Count | Count Fra | At Median F |
|---|---|---|---|---|---|---|---|---|---|---|
| DrawOpaqueObjects | 4 | 0.96 | | 0.98 | 0.91 | 1.47 | 0.56 | 308 | 1 | 1.29 |
| Gfx.WaitForGfxCommandsFromMainThread | 4 | 0.23 | | 0.24 | 0.16 | 0.38 | 0.22 | 313 | 1 | 0.33 |
| DrawTransparentObjects | 4 | 0.03 | | 0.04 | 0.03 | 0.19 | 0.16 | 308 | 1 | 0.05 |
| MainLightShadow | 4 | 0.03 | | 0.03 | 0.02 | 0.08 | 0.06 | 308 | 1 | 0.02 |
| Gfx.SetRenderTarget | 4 | 0.01 | | 0.01 | 0.01 | 0.05 | 0.04 | 1848 | 6 | 0.01 |
| LightCookies | 4 | 0.01 | | 0.01 | 0.00 | 0.05 | 0.04 | 308 | 1 | 0.01 |
| DrawScreenSpaceUI | 4 | 0.00 | | 0.00 | 0.00 | 0.01 | 0.00 | 308 | 1 | 0.00 |

**And the same scene where each model has BakeAO component attached with 512x512 texture** (it is an overkill a bit)**:**

| API Call | Count | Avg CPU ms | Σ CPU ms | Avg GPU ms | Σ GPU ms |
|---|---|---|---|---|---|
| Map() | 4,825 | <0.01 | 0.93 | 0.00 | 0.00 |
| Unmap() | 4,825 | <0.01 | 0.62 | 0.00 | 0.00 |
| PSSetShaderResources() | 2,702 | <0.01 | 0.53 | 0.00 | 0.00 |
| IASetVertexBuffers() | 2,124 | <0.01 | 0.36 | 0.00 | 0.00 |
| IASetIndexBuffer() | 2,124 | <0.01 | 0.23 | 0.00 | 0.00 |
| DrawIndexed() | 2,110 | <0.01 | 0.19 | <0.01 | 1.45 |
| PSSetSamplers() | 927 | <0.01 | 0.11 | 0.00 | 0.00 |
| IASetInputLayout() | 319 | <0.01 | 0.05 | 0.00 | 0.00 |



Marker Details for currently selected range

| Marker Name | Depth | Media | Media | Mean | Min | Max | Range | Count | Count Fra | At Median F |
|---|---|---|---|---|---|---|---|---|---|---|
| DrawOpaqueObjects | 4 | 8.06 | | 8.13 | 7.57 | 10.66 | 3.09 | 422 | 1 | 7.88 |
| Gfx.WaitForGfxCommandsFromMainThread | 4 | 0.24 | | 0.25 | 0.11 | 0.47 | 0.37 | 430 | 1 | 0.25 |
| DrawTransparentObjects | 4 | 0.05 | | 0.05 | 0.04 | 0.10 | 0.06 | 422 | 1 | 0.04 |
| MainLightShadow | 4 | 0.02 | | 0.02 | 0.02 | 0.04 | 0.03 | 422 | 1 | 0.03 |
| Gfx.SetRenderTarget | 4 | 0.01 | | 0.01 | 0.01 | 0.02 | 0.01 | 2532 | 6 | 0.01 |
| LightCookies | 4 | 0.01 | | 0.01 | 0.00 | 0.02 | 0.02 | 422 | 1 | 0.01 |
| DrawScreenSpaceUI | 4 | 0.00 | | 0.00 | 0.00 | 0.01 | 0.01 | 422 | 1 | 0.00 |

You can see that **much more** time was spent on the CPU on the Render Thread. This is because BakeAO uses MaterialPropertyBlock to set textures, which breaks SRP batching.

Of course, games and hardware are different, and I can't predict how BakeAO will affect your game's performance. It's better to use it if you have some performance budget to spare on CPU and GPU.

For best performance, use BakeAO component only if needed, if possible set textures in the materials directly.

I advise using baked lighting for static models (scene lightmaps), and utilizing BakeAO component only on the modelcd qcd s that can't be baked as static models in the scene.

**Using baked AO textures in materials directly will be more performant than using BakeAO component.**

# 11. Troubleshooting and FAQs

## Troubleshooting:

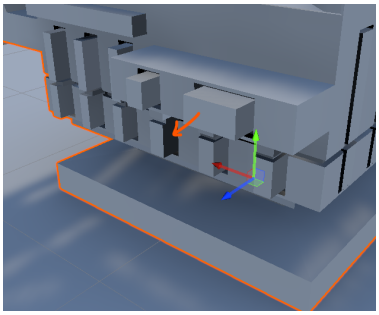
### Error: Missing UV Set

**Issue:** During baking, you might encounter an error related to missing UV sets in the mesh.

**Solution:** Ensure that your mesh has proper UV sets defined. If not, you can generate them using external modeling software like Blender or enable "Generate Lightmap UVs" in Unity's mesh import settings.

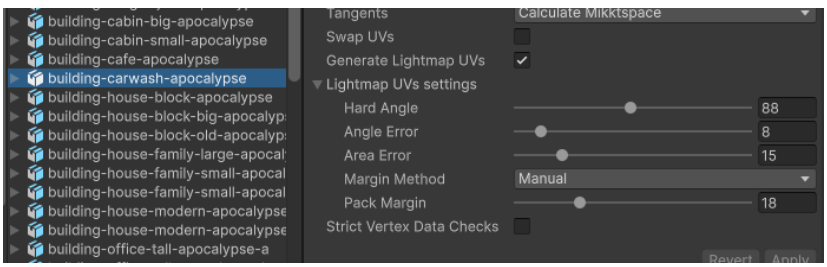
### Black faces after baking the model

**Issue:** Some faces appear as black after baking the model.

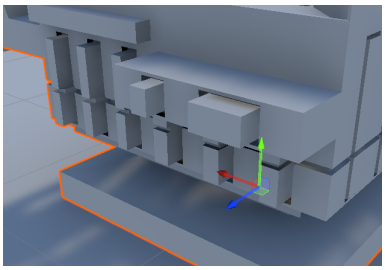


**Solution:** This issue appears when you bake low-resolution textures or use low MSAA samples. Try increasing the MSAA samples in the baking quality settings or increasing the texture resolution.

If increasing texture resolution does not help, try baking the texture for different UV set. If UV1 is not present, you can find the model asset in your project and select the "Generate Lightmap UVs" option. A higher pack margin will allow you to bake lower-resolution textures.



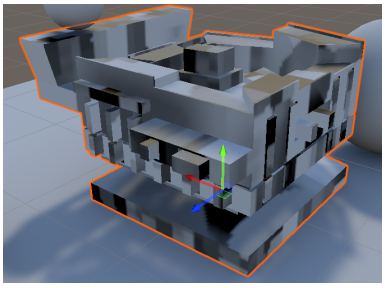

This is how this model looks after increasing Msaa samples in baking quality from 4 to 16:



Baking textures with lower MSAA settings is recommended because larger values can vastly increase baking times.

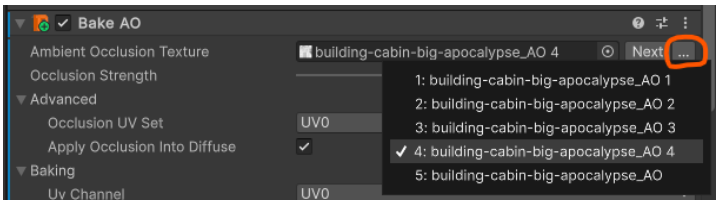
**AO looks glitchy**

**Issue:** AO texture looks very buggy after being applied to the model



**Solution 1:** Change **the Occlusion UV Set property in the advanced foldout of the Bake AO component**. There is a chance that the texture was baked for different UV set.

**Solution 2:** There is a chance that none of the UV sets will look fine. In this case, check if the texture you're using was baked for this model. You can check if the texture appears on the list after clicking the "..." button in the Bake AO component. If the correct texture is used, you will see a checkmark on the list.

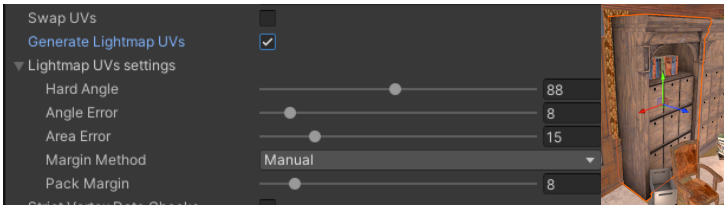



**Solution 3**: If you're sure that the texture was baked for this model, and no other texture works, probably the UVs of the model changed. In this case, you must bake the texture again or revert the model to the previous version.

**Some parts of the model does not look right.**

**Issue:** Some parts of the model looks like they are occluded, but you can't see the occluder.

**Solution 1:** This can happen when model have overlapping UVs. Make sure your model UVs are in 0-1 range and does not overlap. You can also generate UV set in Unity by using **Generate Lightmap UVs** in model import settings. If you use generated lightmap UV, bake AO texture for UV1.

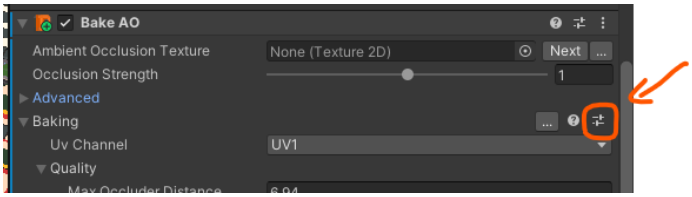


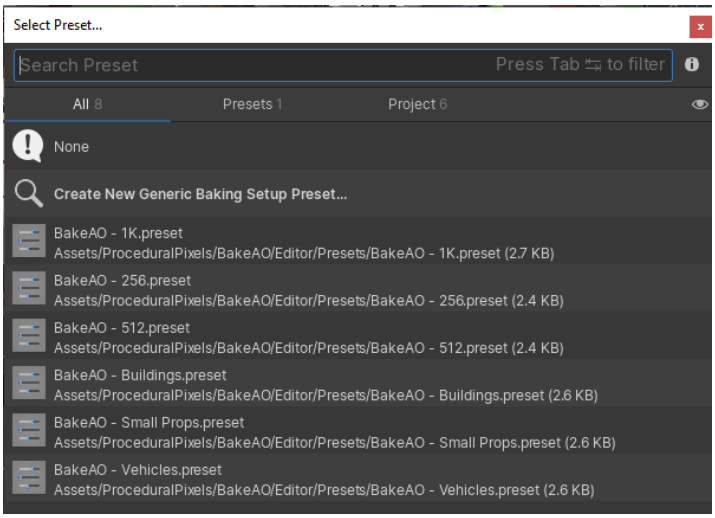
**Solution 2:** Those artifacts can also appear when another renderer acts as an occluder, but it is not visible in a scene view. Ensure you have correctly set layer interactions in **Project Settings / Bake AO.**

## FAQs

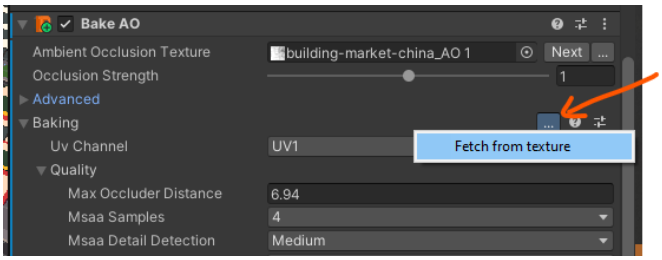
**Is it possible to save baking parameters for some models?**

Yes, Bake AO is integrated with Unity Presets. You can use a button to see and manage all the presets using the built-in preset system.




Additionally, if you forgot the parameters for already baked texture, you can fetch its parameters using this menu:




**How does BakeAO affect game performance?**

See: Performance considerations

**Does it have real-time ambient occlusion?**

No. Bake AO is a tool for generating textures that can be used in materials or Bake AO components. The Bake AO component only assigns baked texture to a renderer; it does not make the renderer react to AO in real time.

**Does Bake AO support baking AO for transparent or cutout objects?**

Unfortunately, the Bake AO does not support transparent or cutout renderers; those will always be baked as opaque. I have a plan to support transparency and cutouts and those will probably appear in the first quarter of 2025.

**Can Bake AO bake lighting for the whole scene?**

No. Bake AO is a tool for generating textures that can be used in materials or Bake AO components. It is not a lightmapper. But you can use Baking Wizard and enqueue baking for all the models in the scene, although it is advised to use a built-in lightmapper for baking lights for static meshes.

**Which platforms are supported?**

Textures produced by Bake AO can be shipped on any platform. Bake AO component can also be used on any platform. But for baking textures, make sure that your hardware meets those requirements.

**Do I need a license for BakeAO to use textures created with BakeAO?**

You are the owner and have full rights to the textures you baked with BakeAO. BakeAO is just a tool for creating content. You can do whatever you want with your textures, and I don't care about that. You don't need a license if you don't use BakeAO features. But if you use the BakeAO component in the project you work on, or click the "Bake" button, youS should have a proper license for BakeAO.

See *Extension Asset license*: https://unity.com/legal/as-terms

## 12. Known issues, limitations and work in progress

**Known issues**

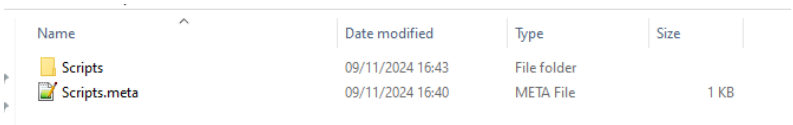
**Import issue in Version 1.0.0, fixed in 1.0.1 and later.**

There is an issue in some projects where the plugin imports some files into your "Scripts" folder during installation. While the fix for that awaits approval from the Unity Curation Team, the workaround is to change the GUID of the folder in your project that conflicts with BakeAO.

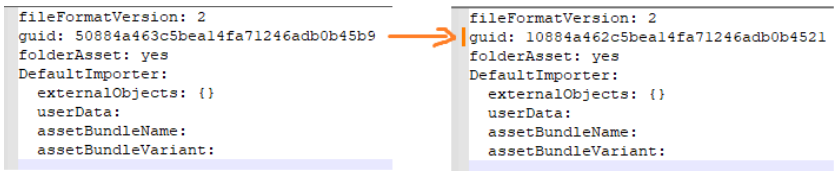
For example, if BakeAO wants to add files to your project, instead of `ProceduralPixels/BakeAO`, cancel the plugin import and change the GUID of the problematic folder in your project.

For example:

1. Find the problematic folder in your project:




2. Open the .meta file of this folder in a text editor and edit the GUID by hand. It is just enough to change one digit to a different one.

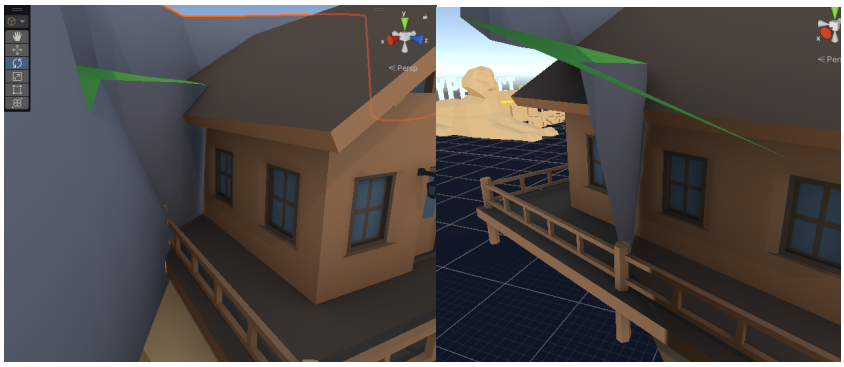



**When updating BakeAO from 1.0.0 to a later version, you need to remove BakeAO completely from your project before the update.**

**When baking skinned meshes, they are always baked in bind pose.** This can cause some artifacts when baking in a scene context or when a skinned mesh works as an occluder. I will be fixing this in the next updates. See the roadmap here.

**BakeAO does not support double-sided meshes, transparent meshes and cutout transparency.** Only opaque meshes are supported.

**Some artifacts can appear when occluder triangles intersect with the baked model.** Brighter, or darker contact points. The issue is not present when objects are not intersecting. Usually, you need to know what to look for to see the issue.



**Extensive usage of Bake AO Component can decrease the CPU performance,** see Performance Considerations

**When using OpenGLES in Editor, baking times can be insanely long, and during baking editor is very laggy. Please use Vulcan, DirectX11 or DirectX12 instead. BakeAO Component and textures will work without issue when using OpenGLES, only baking performance is affected.** When using Windows, Unity Editor usually uses DirectX API, so the issue is not present there. It is present only if the project has changed the rendering API to OpenGLES in **Player Settings.**

**Limitations:**

**You need to use shaders included with BakeAO to make BakeAO Component work.** This is by design.

**All custom shaders need to be modified appropriately to work with BakeAO Component.** This is by design. See Custom Shaders

**Textures can be baked only for MeshRenderers, SkinnedMeshRenderers, MeshFilters and models (mesh assets) in the project. No other assets are supported.**

**Track my work:**

You can track my work in the roadmap. Here I keep information about current issues, limitations, planned work, and my current focus regarding BakeAO: See the roadmap here.

## 13. Reporting issues and suggesting improvements

If you encounter any issues while using the Bake AO plugin or have suggestions for improvements, I'm here to help. Your feedback is crucial in improving the plugin.

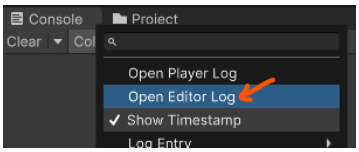
**Feel free to report any issues or provide feedback:**

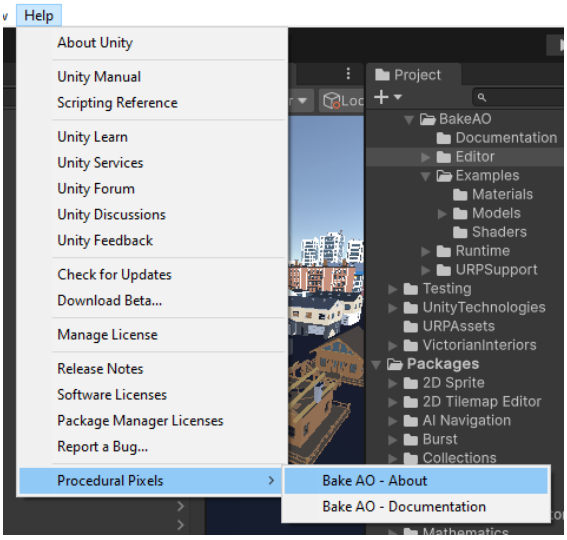
- **Discord:** Join **Procedural Pixels Discord Community** and ask directly in the discord channel (or in a private message): https://discord.gg/NT2pyQ28Jx
- **Email:** You can send your issues and suggestions to my email address at dev@proceduralpixels.com

To report an issue or provide feedback, follow these guidelines for creating a helpful and informative report:

1. **Invoice number:** This is needed only if this is your first reported issue. I don't directly support developers who didn't buy the product. If I can't verify that you bought the product, I will ask you to provide an invoice number.
2. **Detailed Description:** Provide a clear and concise description of your problem or the suggested improvement. Include information about the specific feature or functionality involved.
    1. **Reproduction Steps:** List step-by-step instructions to reproduce the issue. This will help me understand the context and conditions of the problem.
    2. **Expected Behavior:** Describe what you expected to happen when using the plugin under normal circumstances.
    3. **Reproduction Chance:** If possible, estimate the likelihood of reproducing the issue. Does it happen consistently, or is it sporadic?
    4. **Screenshots and Videos:** Include screenshots or short video clips to demonstrate the issue visually. These visuals can provide valuable context and help me identify the problem faster.
3. **Assets and Files:** Whenever possible, attach relevant model and baked texture assets with their .meta files (if the problem is about the texture, models etc.) This will allow me to analyze the problem more effectively.
4. **Additional Information:** Any other relevant information, such as **Unity console logs, Unity version, plugin version, specific settings you use, or hardware specification (GPU used and operating system)**, will help resolve the issue.

To get the editor to log file, right-click on the console tab, and select "Open Editor Log"





**I aim to quickly respond to all reports and feedback, usually within 1-3 days. I check the Discord server most often. Please understand that my availability might be limited as I work on the plugin during my free time (usually after 7PM CET). Nonetheless, I'm committed to providing you with the best possible support and continuously improving the plugin based on your valuable input.**

**My goal is to provide you a highest quality and intuitive product.**

**Thank you for being a part of the Bake AO community and helping improve this plugin.**