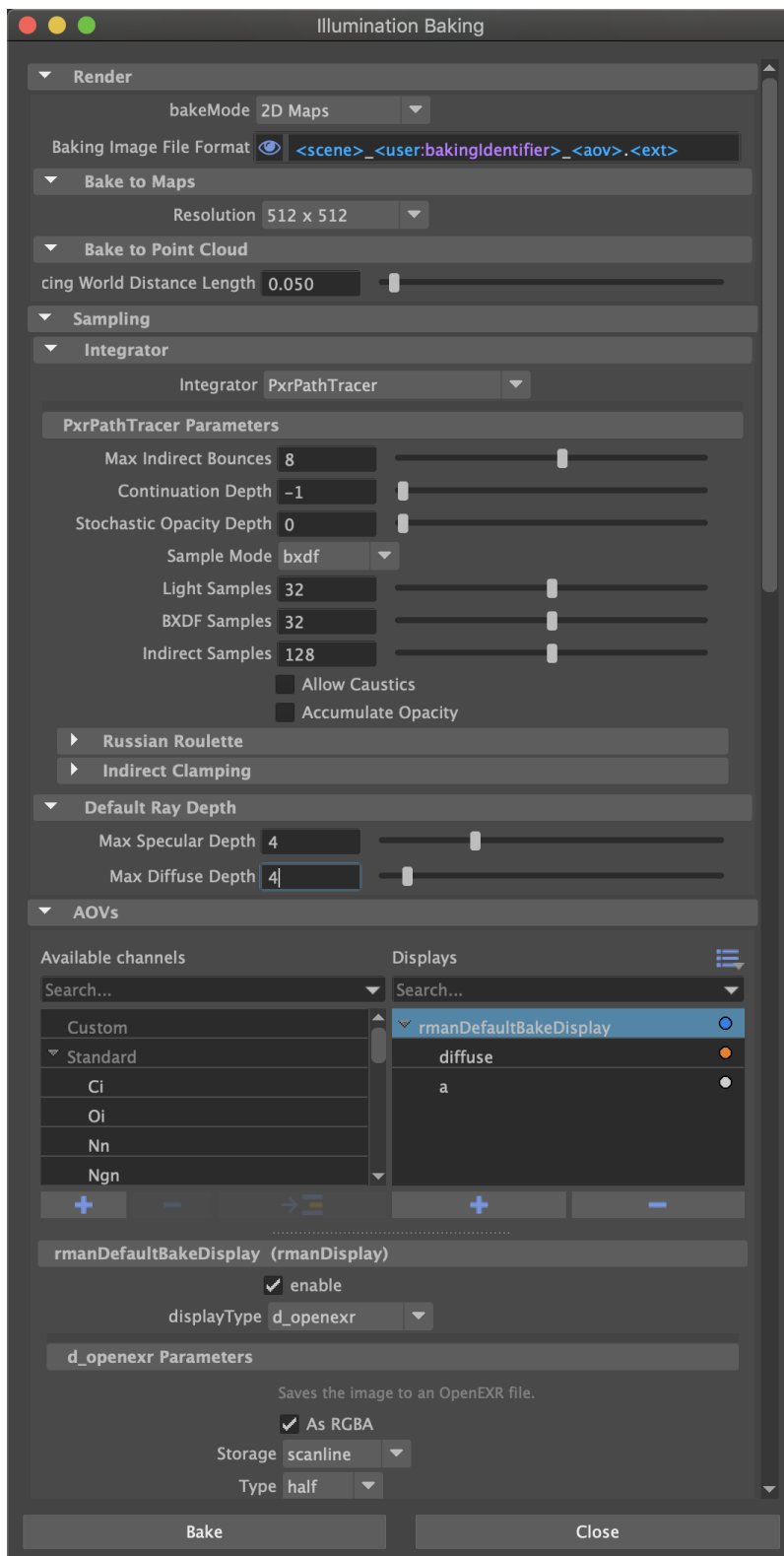# Baking illumination

Overview

This mode actually bakes outputs of an integrator. It will be mainly used to bake diffuse global illumination but it can bake any light path expression too.

How does it work ?

- Scene objects are baked one by one.
- The integrator run in a single increment on each point (Max Samples = 1).
- The result is output to one or more AOVs, each containing a single display channel.

Settings

The illumination baking settings overrides only a subset of parameters of your scene's render globals. If your current scene renders as expected, these are the only settings you need to bake.

Bake Mode

The main purpose of this menu is to validate your setup before rendering and warn you if an incorrect setup has been detected.

2D Maps

- Generates 1 image per object.

⚠️ All baked objects MUST have valid uvs for baking

3D Point Cloud

- Generates one or more point clouds, depending on file naming format.
- Bakes in world space

Baking Image File Format

As usual, you create a pattern with tokens to create named AOVs, but with a twist. The default file pattern is:

```
<scene>_<user:bakingIdentifier>_<aov>.<ext>
```

`user:bakingIdentifier` is a new attribute you can find on all bake-able shapes in the RenderMan section of the Attribute Editor:
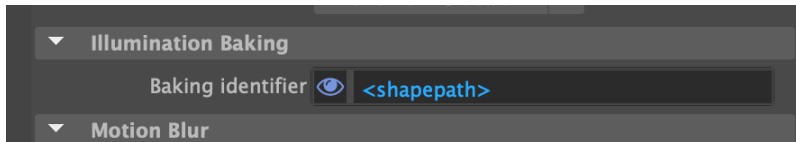
It defaults to `<shapepath>` so that each object gets a unique file name where the "|" character is replaced by "_", which is what you want when you bake to 2D images.

If you want to bake to a single 3D point cloud, you should remove `<user:bakeIdentifier>` from the file name pattern.

Bake to Maps (2D baking)

Select the global baking resolution. Larger resolutions take longer to render.
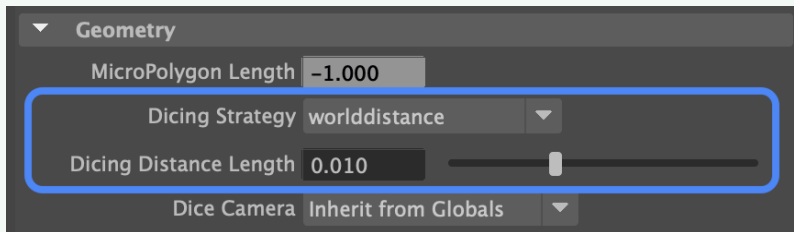
⚠️ **Limitation**

All objects will be baked at the same resolution. This is a limitation of the current implementation

Bake to Point Cloud (3D baking)

Point clouds are baked in world space and their density depends on the shape's dicing density. Select here the global dicing density for the scene.

✅ You can override the dicing density per shape by setting the Dicing Strategy to worlddistance and entering a different Dicing Distance Length. Smaller values creates more points.

Integrator

Select an integrator and its settings. We recommend PxrPathTracer.

⚠️ **Limitation**

Some integrators like PxrVCM and PxrUnified are **incompatible** because they rely entirely on incremental rendering to accumulate samples and resolve illumination over time.

Baking runs the integrator **once per point** (Max Samples = 1) and it must be setup to **deliver good results in a single increment**.

⚠️

For best results, you need to increase the number of direct (light samples and BXDFs samples) and indirect samples enough to get reasonably clean results. Best settings are very scene-dependant and will require experimentation.

> ⚠️ **Limitation**
>
> Baking is **not adaptive** (i.e. brute-force) and all samples will be computed, regardless of convergence.
>
> Illumination baking can be very slow: make sure you **choose a minimal number of samples** and remember that a bit of noise is nearly invisible in baked illumination.

Note that it makes no sense to bake view-dependant illumination, like specular, and you should alway try to bake view-independent signals like diffuse, albedo, etc.

> ⚠️ **Limitation**
>
> It is currently impossible to run the denoiser on baked 2D images.

Default Ray Depth

As usual, you may set the max diffuse and specular depth. This will override the usual scene settings during baking.

AOVs

The AOV section works exactly like in the render globals.

We default to baking diffuse illumination, both direct and indirect.

> ✅ Should you need to bake raw illumination, you will need to bake the albedo as well and divide the illumination by the albedo (in a compositing package or a shader) to obtain raw illumination (aka irradiance).

> ⚠️ **Limitation**
>
> Each AOV (aka Display) can only contain a single display channel and output at most 4 channels (RGBA).
>
> Add more displays to bake more channels.

Bake 2D

Any non-deep image format can be used to bake to images. OpenEXR is recommended to avoid clamped results.

Bake 3D

Make sure to select the `d_pointcloud` display driver to bake point clouds.

If you forget to do so, the preflight checks should warn you and cancel the bake render.

Launching a bake render

A bake render is essentially a batch render with one different setting. It uses the bake Hider instead of the raytrace Hider.

To start a bake render simply choose **Bake Render: Illumination** instead of Batch Render from the RenderMan menu. All the usual batch render preferences apply to the bake render.

IPR Render

Preview Render
Batch Preview Render

Batch Render
Bake Render: Patterns
Bake Render: Illumination   ☐

Render Settings
Preferences

Tools   ▶
Archive   ▶
Diagnose   ▶
Utilities   ▶
License   ▶

Help   ▶
About

OSL Tools   ▶
rmanAssets Logging Level   ▶