AOVs

AOVs (Arbitrary Output Variables) provide a way to render any arbitrary shading network component into different images. For example, an artist might find it convenient to separate direct and indirect lighting contributions and later recombine them during compositing. Arnold provides built-in AOVs for outputting depth, position, and motion vectors.

There are some limitations:

- Using closures as AOVs is not currently supported.
- A maximum of 16 different light AOVs is supported, although a given AOV can contain a bundle of any number of lights.

> - Multichannel EXRs are possible using Merge AOVs.
> - MtoA supports Cryptomatte AOV shaders.
> - Light path expressions can be used to output light into specific AOVs.
> - An introduction to AOVs for compositing tutorial can be found here.

Legacy

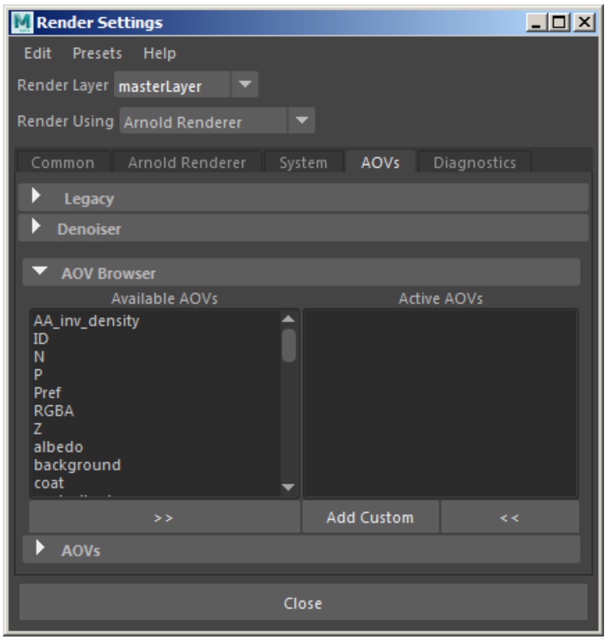
Denoiser

AOV Browser

AOVs

Per-Light AOVs

To the right of the Arnold Renderer tab in the Render Settings dialog, you should see the AOVs tab. The available AOVs for each group are shown in the left list. When you select one, it will move across to the 'Active AOVs' list, which is the set of AOVs you want to use for your render.



Render Layer masterLayer

Render Using Arnold Renderer

Legacy

Maya Render View

Mode

This allows you to enable AOVs, disable them, or specify that AOVs are only to be used in batch renders.

Render View AOV

This allows you to select an AOV channel to preview in the render view. Remember to switch back to 'beauty' afterward.

AOV Shaders

A list of shaders can be defined that will be evaluated after the regular surface shader. With this, it's possible to add shaders to set specific AOVs without modifying the original shader tree. Shaders intended for this purpose should add a booleanmetadata named `aov_shader` on the node itself, as a user-interface hint. If `options.atmosphere` or `options.background` are set, these global AOV shaders will also be run for atmosphere and background contexts.

> Only shaders with AOV-writing capabilities should be added to 'AOV Shaders' in AOVs.

Denoiser

Optix Denoiser

Denoise Beauty AOV

Use the fast, GPU-powered Nvidia OptiX AI denoiser. See the System tab for the GPU device selection options. When enabled a new AOV with the _**denoise** postfix is available in the AOV list.

Arnold Denoiser

Output Denoising AOVs

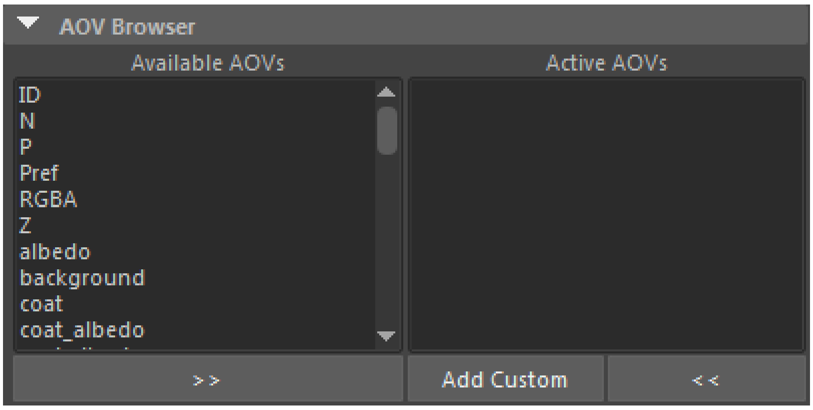
Automatically outputs the optional AOVs (diffuse_albedo, Z, N). This should be used for denoising using Noice (Arnold denoiser), (multilayer EXR) or Kick.

> A current limitation means that you must render in EXR and Merge AOVs need to be enabled.

> This should also be enabled when using batch rendering with Noice.

AOV Browser

The AOV Browser allows you to select the AOVs you want to be active for your render. You can also choose to add a custom AOV.



Either double-click or select and use the arrow buttons to move AOVs from 'Available' to 'Active'.

Arnold provides the following 'built-in' system AOVs. These AOVs are always available, no matter what shader(s) you are using.

Click here to show the full list of AOVs.

- **A:** Alpha.

- **AA_inv_density:** Visualizes the sample density with *Adaptive Sampling.* Use it with a *Heatmap* filter.
- **ID:** Random number value derived from the name of the shape. You can also add specific ID numbers via the user options string field for an object. ie 'id 1'.
- **N:** Smooth normal at the shading point (in world space).
- **P:** Position of the shading point (in world space).
- **Pref:** Reference position of the shading point.
- **RGBA:** Beauty AOV, containing the full rendered image**.**
- **Z:** Depth of the shading points as seen from the camera.
- **albedo:** Reflectivity, the surface or volume color without lighting or shadowing.
- **background:** Emission from the background and skydome lights visible to the camera.
- **coat:** Coat reflection.
- **coat_albedo:** Coat color without lighting or shadowing.
- **coat_direct:** Coat direct lighting.
- **coat_indirect:** Coat indirect lighting.
- **cputime:** This layer contains the CPU time (measured in “ticks”) to evaluate the samples in the pixel.
- **diffuse:** Diffuse reflection.
- **diffuse_albedo:** Diffuse color without lighting or shadowing.
- **diffuse_direct:** Diffuse direct lighting.
- **diffuse_indirect:** Diffuse indirect light.
- **direct:** Direct lighting from all surfaces and volumes.
- **emission:** Lights and emissive objects directly visible from the camera.
- **indirect:** Indirect light from all surfaces and volumes.
- **motionvector:** 2D vector representing the motion in screen space of the shading point during the given time interval. If output to an RGB format, the vector is contained in the R and G channels. You must set an instantaneous shutter for the camera. The reason being is that we don't want motion blur in the render, but we still want the motion velocity information in our motion vector AOV. This can be found under *Motion Blur-> Instantaneous Shutter.*
- **opacity:** RGB AOV with full three-channel opacity (as opposed to single channel alpha).
- **raycount:** Total number of rays traced for samples in the pixel.
- **shadow_matte:** Shadows in the scene, computed as the ratio of occluded direct lighting over unoccluded direct lighting.
- **sheen:** Sheen weight.
- **sheen_albedo:** Sheen color without lighting or shadowing.
- **sheen_direct:** Sheen direct lighting.
- **sheen_indirect:** Sheen indirect lighting.
- **specular:** Specular reflection.
- **specular_albedo:** Specular color without lighting or shadowing.
- **specular_direct:** Diffuse direct lighting.
- **specular_indirect:** Diffuse indirect lighting.
- **sss:** Subsurface scattering and diffuse transmission.
- **sss_albedo:** SSS and diffuse transmission color without lighting or shadowing.
- **sss_direct:** SSS and diffuse transmission direct lighting.

- **sss_indirect:** SSS and diffuse transmission indirect lighting.
- **transmission:** Specular transmission (refraction).
- **transmission_albedo:** Specular transmission color without lighting or shadowing.
- **transmission_direct:** Specular transmission direct lighting.
- **transmission_indirect:** Specular transmission of indirect lighting.
- **volume:** Volume scattering.
- **volume_z:** The Z depth for the first volume contribution is output in a flat AOV.
- **volume_albedo:** Volume color without lighting or shadowing.
- **volume direct:** Volume scatter direct lighting.
- **volume indirect:** Volume scattering indirect lighting.
- **volume opacity:** RGB AOV with the full three-channel opacity for volumes only.

The other AOV groups correspond to the shader nodes being used (assuming those shader nodes support AOV). For example, *Shadow Matte* provides:

- **shadow:** Direct light shadow.
- **shadow diff:** A difference AOV which can be used to eliminate the shadow from the direct component.
- **shadow mask:** This AOV can be used in comp to localize and tweak the shadow.

Other shaders used in your scene will support various other AOVs. Multiple shaders can contribute to the same AOV (for example a *Standard Surface* and a *Lambert* shader both write to the **diffuse_direct** AOV).

Composing the Beauty AOV

The RGBA beauty AOV can be split into smaller AOVs where each contains part of the lighting. In compositing, these AOVs can then be individually modified and added together to get the full beauty AOV.

More AOVs give more control in compositing, but also extra work to handle, and they take up more memory and disk space, especially combined with light groups.

Some example sets of additive AOVs for the full beauty AOV are:

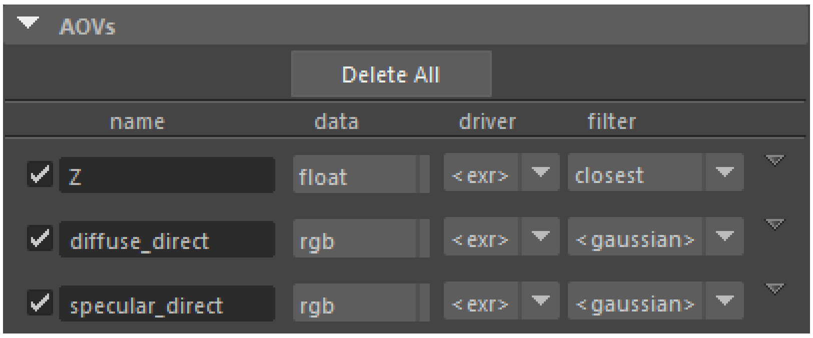
- direct, indirect, emission, background.
- diffuse, specular, coat, transmission, sss, volume, emission, background.
- diffuse_direct, diffuse_indirect, specular_direct, specular_indirect, coat, transmission, sss, volume, emission, background.

Simply adding together such AOVs is all that is needed for the beauty AOV. The albedo AOVs are not needed to reconstruct the beauty AOV but may be used for example to get just the lighting without the surface texture, by dividing diffuse by diffuse_albedo, or for denoising just the lighting while keeping the texture detail intact.

A tutorial about compositing AOVs can be found here.

AOVs

Beneath the AOV browser, the active AOVs that you have selected for output are listed in more detail:



	name	data	driver	filter
☑	Z	float	<exr>	closest
☑	diffuse_direct	rgb	<exr>	<gaussian>
☑	specular_direct	rgb	<exr>	<gaussian>

Each AOV is actually represented by three nodes - the AOV node itself, plus an associated driver node and filter node.

This dialog shows you the type, driver, and filter of the chosen AOVs in drop-down menus, and the checkbox to the left allows you to control which are active. If you click on the triangle at the far right, a context menu is displayed which in addition to providing another way to remove or make an AOV active/inactive, also allows you to add an alternative output driver for each AOV (and to select the driver and filter).

Because the driver and filter nodes are separate to the AOV node, you can add multiple outputs for each AOV by adding extra driver nodes to that AOV node, for example allowing both EXR and JPG to be written out by the same AOV. This extra flexibility in AOV output can be really useful in some situations (maybe you want to output your beauty pass in multiple formats or using different filters – or you might have a custom output driver that you want to output to at the same time as the normal render view).

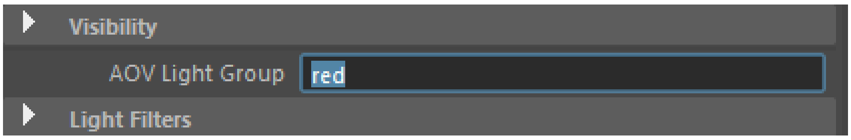
Note that the driver drop-downs will show the driver which is currently the selected file output type in the Common tab of Render Settings in brackets, e.g., <exr> as in the above screenshot. So, you can choose a specific driver for each AOV, or if you choose the top one from each drop-down menu, shown in angle brackets, then you are choosing to use the current default output driver. So, in the above illustration where the driver is shown as <exr>, if you went to the Common tab and altered the file output to be png, when you return to the AOV tab you'll see that all the drivers are now shown as <png>. If you don't want them to change to reflect the current default driver, select a driver name without angle brackets. Similarly, the filter selection is also shown with angle brackets if it is set to use the current default filter.

Each AOV has a node associated with it. You can use the attribute editor to examine these nodes in more detail.

Per-Light AOVs

Each light object has an 'AOV Light Group' attribute which can be used to write out the light contribution to a separate AOV with a corresponding name. To create a per-light AOV, you must do the following:

1. Enter a name for the per-light AOV in the 'AOV Light Group' of the light. In this case, we have a red and a blue light in the scene and so the light groups have been named red and blue.

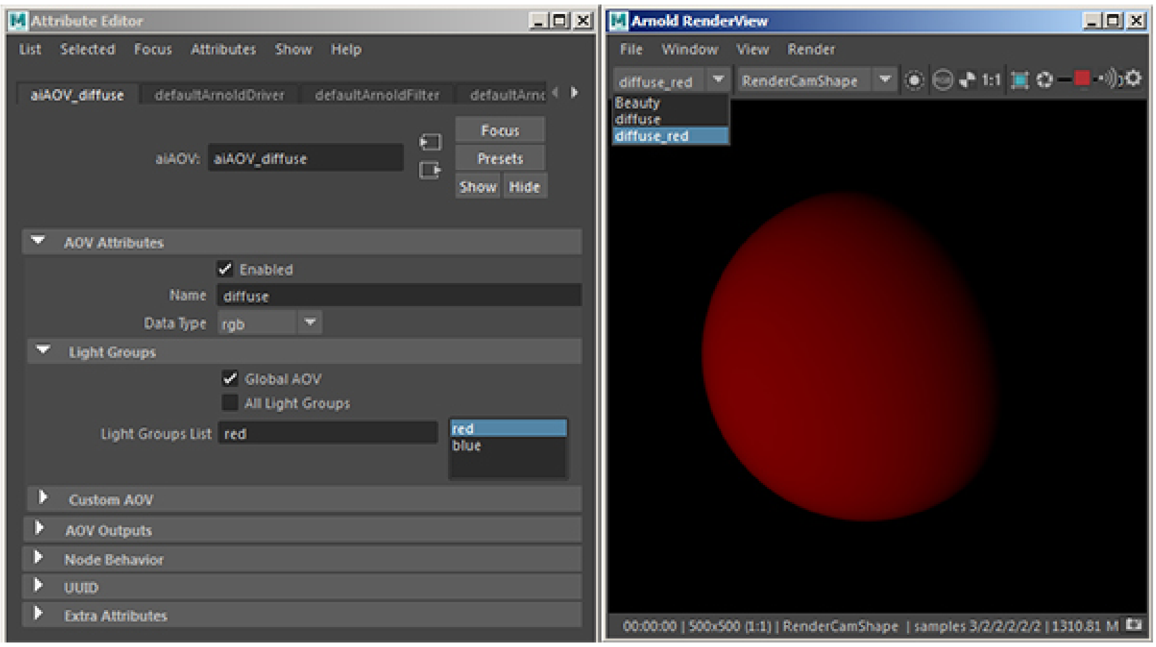


AOV Light Group: red

AOV Light Group in Arnold attributes of light

2. Create an AOV and select it to show it in the attribute editor (in this case we have used a *diffuse* AOV).

3. Render the scene. The AOVs should now be visible in the RenderView by selecting them in the Light Groups List.



aiAOV: aiAOV_diffuse

☑ Enabled

Name: diffuse

Data Type: rgb

☑ Global AOV

□ All Light Groups

Light Groups List: red

red
blue

diffuse_red | RenderCamShape

Beauty
diffuse
diffuse_red

00:00:00 | 500x500 (1:1) | RenderCamShape | samples 3/2/2/2/2/2 | 1310.81 M

Rollover image for blue Light Group

The <LightGroup> token can be used in the output filename so that Light Group AOVs tokens can be inserted in custom ways.

A scene file that demonstrates light group AOVs (with denoising) can be downloaded here.