Lights

You can use standard Maya lights when rendering with MtoA. If you select a light and then inspect the Maya attribute editor, as well as the regular light attributes, you will also see a new group of Arnold attributes for the light, which is where any additional settings used by Arnold can be accessed.

Arnold does not support **constant light decay**. However, Arnold's **Quad** and **Disk** area lights have a **Spread** parameter, that when set to a low value, will give you something similar to a constant fall-off. Similarly, the **Spotlight** in Arnold has a **Lens Radius** parameter that, when set to a non-zero value, and coupled with a low angle, has the apparent effect of flattening the decay, like in a Hollywood-style searchlight. Another workaround is to use distant/directional lighting.

- **Area lights** (including support for disk, cylinder, quad).
- **Sky Dome light**
- **Mesh light**
- **Photometric light**
- **Maya Directional light**
- Maya Point light
- Maya Spot light
- Light filters

Ensure that Light Linking is set to none when instancing lights, otherwise the instanced light will not render.

**Common Light Attributes**

As well as honoring the standard Maya light attributes, the Attribute Editor will also show the following attributes under the Arnold group:

![Arnold Area Light Attributes](image.png)

**Use Color Temperature**

The temperature of an ideal black-body radiator, in kelvin units, that is used to determine the color for a light source. The default color is set to 6500 K, which is considered as the white point by the Commission Internationale de l'Eclairage (CIE). The color ranges from red, through to white and then to blue. Values
above 6500 K will give a cool color, whilst values below will show a warm color.

The image above shows the effect of color temperatures on a scene in kelvin units.

A range of color temperature values in kelvin (in this case a cylinder light has been used).

Note that **Use Color Temperature** will override the default color for the light. That includes any textures that are assigned to the color attribute.
Exposure

Exposure is an f-stop value which multiplies the intensity by 2 to the power of the f-stop. Increasing the exposure by 1 results in double the amount of light.

In Arnold, the total intensity of the light is computed with the following formula:

\[
\text{color} \times \text{intensity} \times 2^\text{exposure}
\]

You can get the same output by modifying either the intensity or the exposure. For example, intensity=1, exposure=4 is the same as intensity=16, exposure=0. **Note:** \(2^0 = 1\), not 0.

\[
1 \times 1 \times 2^4 = 16 \\
1 \times 16 \times 2^0 = 16
\]

The reasoning behind this apparent redundancy is that, for some people, f-stops are a much more intuitive way of describing light brightness than raw intensity values, especially when you're directly matching values to a plate. You may be asked by the director of photography (who is used to working with camera f-stop values) to increase or decrease a certain light by 'one stop'. Other than that, this light parameter has nothing to do with a real camera's f-stop control. Also, working with exposure means you won't have to type in huge values like 10,000 in the intensity input if your lights have quadratic falloff (which they should).

If you are not used to working with exposure in the lights, you can simply leave the exposure parameter at its default value of 0 (since \(2^0 = 1\), the formula then simplifies to: color \times intensity \times 1).
Samples

Controls the quality of the noise in the soft shadows and direct specular highlight. The higher the number of samples, the lower the noise, and the longer it takes to render. The exact number of shadow rays sent to the light is the square of this value multiplied by the AA samples. A schematic of how light noise occurs in Arnold can be found here.

Note that setting light samples to 0 disables the light.

Noise from lights can sometimes be difficult to diagnose, particularly if the light source is broad in comparison to the scene and the shadows have an extremely wide penumbra. In these cases, it can sometimes be mistaken for indirect diffuse noise. It highlights the necessity for testing noise raytype. This diagram shows how a light is traced in Arnold.

The example below shows specular highlights from area light sources. There are four spherical mesh light sources of varying size and color temperature. Underneath are four cubes with Standard shaders assigned to them with varying degrees of specular roughness. Note that more noise is apparent with smaller light
sources. Increasing the number of light samples resolves the noise.

If the issue is noise in a specular highlight, you will need to confirm that the source is the direct light and not a secondary ray type (such as specular). This is easy to achieve by turning off global illumination; set the GI_diffuse_depth, GI_specular_depth, and GI_reflection_depth to zero (this essentially turns off all global illumination). If the noise is still there, we know it is the specular component of the illumination model. If the issue is shadow noise, then we can simply toggle ignore shadows in the Arnold render settings, and the noise will completely resolve.

The key, again, is to modify the sampling and observe the changes. Increasing the number of light samples should have an immediate and quantitative effect on the smoothness of the specular highlight and the shadow. If there is no change, light samples are not responsible for the noise.

Multiple importance sampling (MIS) is enabled by default in Arnold. The images below show the difference when rendering with and without multiple importance sampling.
Multiple importance sampling: Off  Multiple importance sampling: On

More information about light samples and removing noise can be found [here](#).

**Normalize**

If enabled, you will be able to tweak the shadow softness by changing the size (i.e., radius) of the light, without affecting the amount of emitted light. This is very handy for artistic control. Otherwise, if not enabled, the amount of emitted light is proportional to the light's surface area.

Enabled (default). Radius 2.5.  Disabled. Radius 2.5.

**Cast Shadows**
Enables the computation of shadows cast from the light.

**Shadow Density**

Sets the shadow density, or strength. This controls how the shadow blends with the material on which the shadow is cast: a value of 1.0 produces an opaque, black shadow, and a value of 0.0 gives no shadow. Normally this would be 1.0.

**Shadow Color**

Sets the intensity of each color channel for shadows. Normally this would be black.
Cast Volumetric Shadows
Determines if volumetric shadows will be computed. This option is not in Distant or Skydome Lights.

Enabled (default)  Disabled

Volume Samples
The volume samples parameter handles the number of samples used to integrate the in-scattering from direct light. Like the "samples" parameter for surfaces, it is also a squared number.
'Volume Samples' does not apply to the 'Volume Scattering' atmosphere shader. To improve the quality of the 'Volume Scattering' shader, you must increase the 'Volume Scattering' samples.

**Visibility**

Per-light scaling for Diffuse, Specular, SSS, and Volume. Weights scaling the light contribution to each of those components independently. Should be left at 1 to produce physically accurate results.

**Indirect**

The relative energy loss (or gain) at each bounce. This should be left at its default value of 1.0 for physically meaningful results. Values bigger than 1 make it impossible for GI algorithms to converge to a stable solution.
Max bounces

The maximum number of times the energy from this light will be allowed to bounce in the scene. A Max Bounces value of 0 means that the light will only be part of the direct lighting computations, effectively disabling GI for this light. Note that this value works along with the global ray depth controls, so the default value of 999 bounces per light is just a theoretical maximum; in practice, the global ray depth limits are set much lower.

AOV Light Group

Per-light AOVs are available via a string parameter. Each light has an AOV parameter which writes out the
light contribution to a separate AOV with a corresponding name. See also Light Path Expression AOVs.

Information about creating per-light AOVs can be found [here](#).