

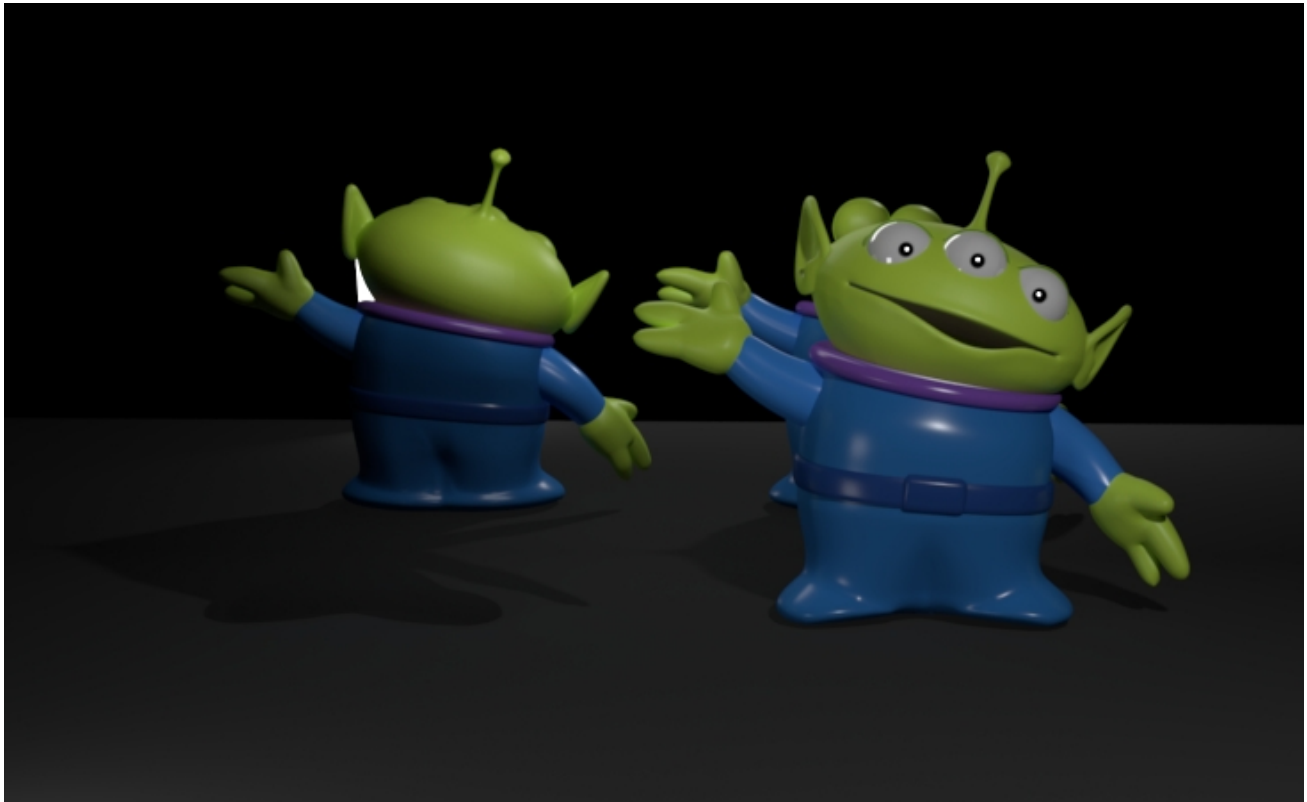
Visibility

Object Visibility

Camera Visibility

Camera Visibility is an *object attribute* that sets the camera visibility. It is used to show or hide an object from the camera.

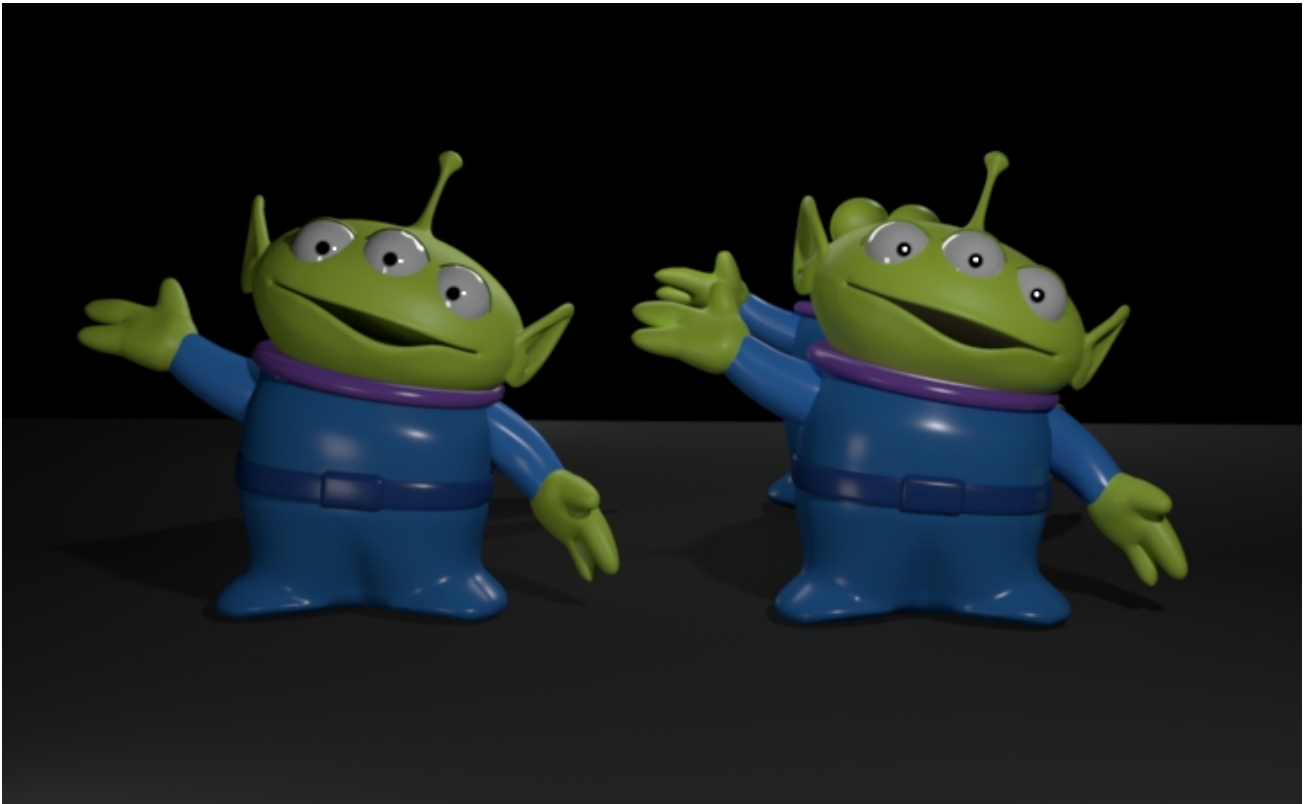
By default, camera visibility is set to 1. Setting camera visibility to 0 is the most optimized way to hide an object from the camera but still have it cast shadows, be visible in reflections, etc. Below, the left Alien is not visible to the camera but he casts a shadow and is visible in the mirrored wall behind him.



Indirect Visibility

Indirect Visibility is an *object attribute* that sets the indirect visibility. It is used to show or hide an object from reflections, refractions, etc.

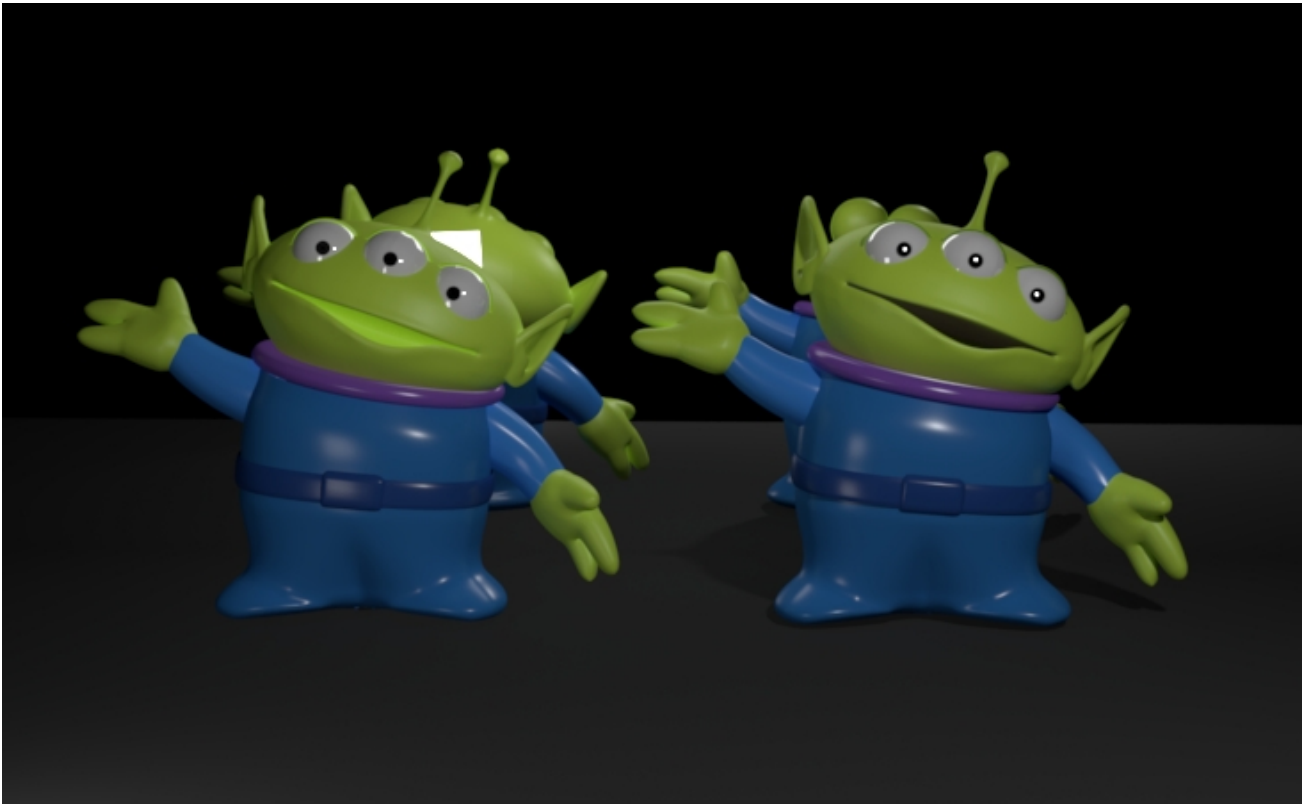
By default, indirect visibility is set to 1. Setting indirect visibility to 0 is the most optimized way to hide an object from other objects but still have it cast shadows and be visible in camera. Imagine rendering a vampire where it shouldn't (according to legend) be visible in a mirror. Below, the left Alien is not visible to the mirrored wall but he casts a shadow and is visible to the camera.



Transmission Visibility

Transmission Visibility is an *object attribute* that sets the shadow visibility. It is used to disable shadow casting.

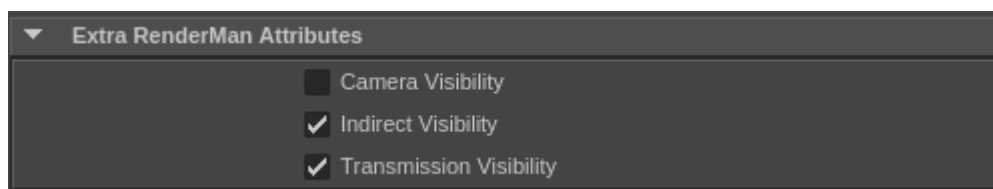
By default, transmission visibility is set to 1. Setting transmission visibility to 0 is the most optimized way to disable shadowing. This may be useful on scenes like an interior where a translucent curtain can darken a room too much by blocking some portal lighting. Or where a shadow from an object may create an unwanted shape. Below, the left Alien is not casting shadows (even onto himself) but is visible in the mirrored wall behind him and to the camera. He also fails to block the light source.



Below is a simple example of how to change these parameters in two common applications.

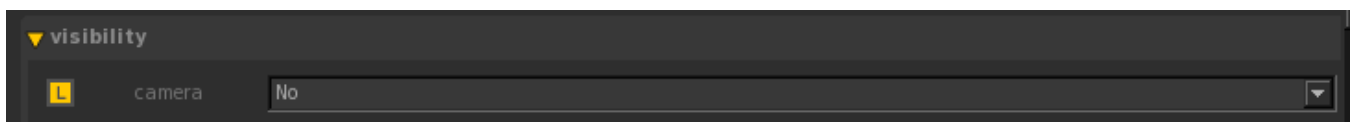
Maya

Camera visibility in RenderMan for Maya is an object attribute. To set it to 0, uncheck it in Maya Attribute Editor's *Attributes > RenderMan > Visibility Settings*:



Katana

Camera visibility in RenderMan for Katana is an object attribute. To set it to 0, set it to No in *PrmanObjectSettings*' Visibility for camera:



 Note that in Katana, another option of hiding the object is by pruning the object. This removes it completely.

Presence

Presence is a *Bxdf* parameter that sets the *probability* of surface hit. It is a scalar value of either 0 or 1. By default, it is set to 1 which means that it has a 100% chance of surface hit and thus the surface is present to the renderer. If it is set to 0, that means the surface is not present to the renderer. This is useful for making parts of geometry invisible to the renderer. Unlike above where we specify when an entire object is seen, here we can create things like leaves made of cards, removing the area outside the leaf texture. This can be combined with the above visibility settings.

Whether it is present or not present, the light scattering properties do not change.

Presence should be connected to a black and white mask. We can see in this example below where it is 0 (black in the mask), there is no surface. The alpha channel is black.

Example

Presence [1.0]
Checked mask connected to Presence parameter
Resulting alpha

Why Do We Need This?

For render optimization. Presence is for camera rays *and* indirect rays. The renderer only needs to shade the surface and execute the Bxdf when the surface is present.

When To Use

- Connect the Presence parameter to a texture that defines a "cut out" in the surface. For instance, we model a leaf as a cheap bilinear patch and paint a black and white mask for the shape of the leaf.
- To hide particles.
- When you need to control the opacity of an object. Keep in mind the production integrators may have this feature off (set to 0) for efficiency reasons. You can enable this opacity calculation to a certain depth by making the 0 an integer with a reasonable low number.

When Not to Use

- Not for translucency, transparency, or refraction.

If we want to shade a translucent surface, we can set the Double-Sided Transmit shading parameter under the Diffuse controls of [PxrSurface](#) instead.