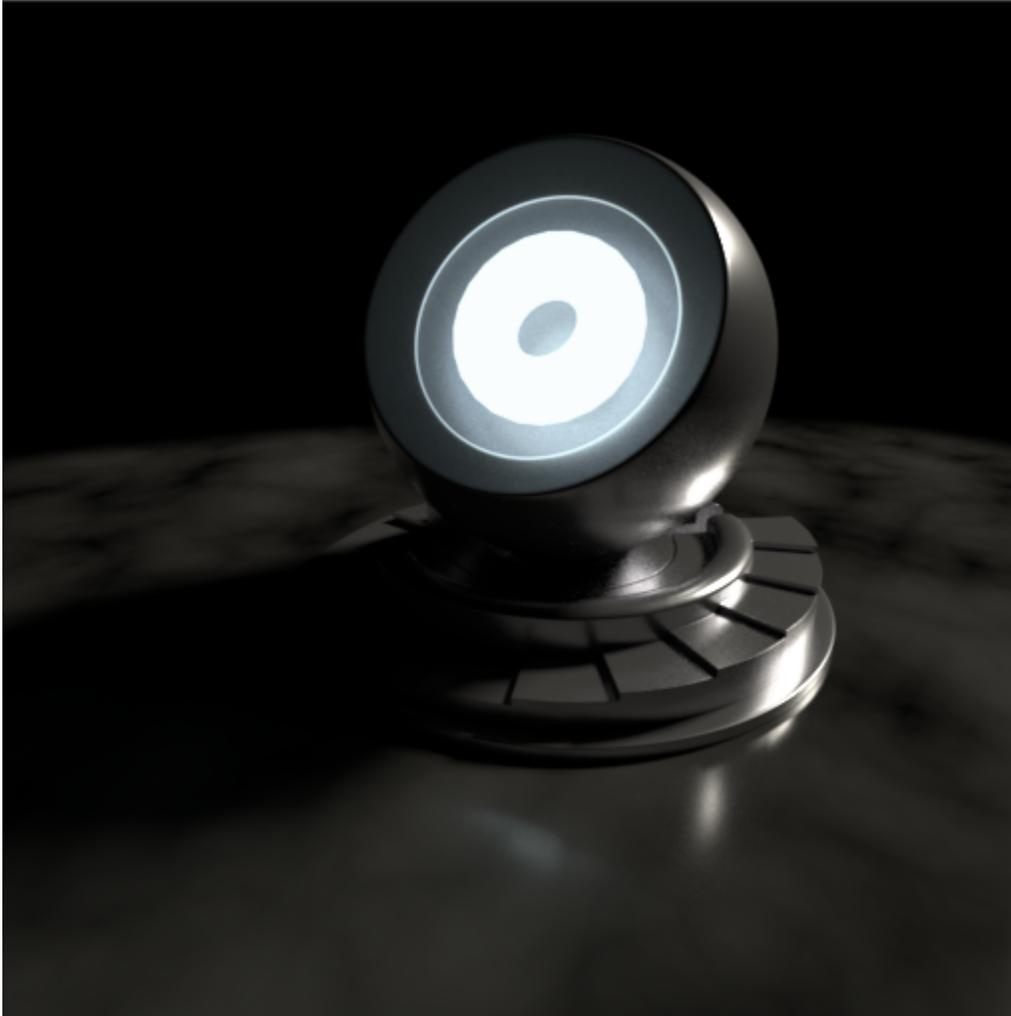


Mesh Lights in Katana

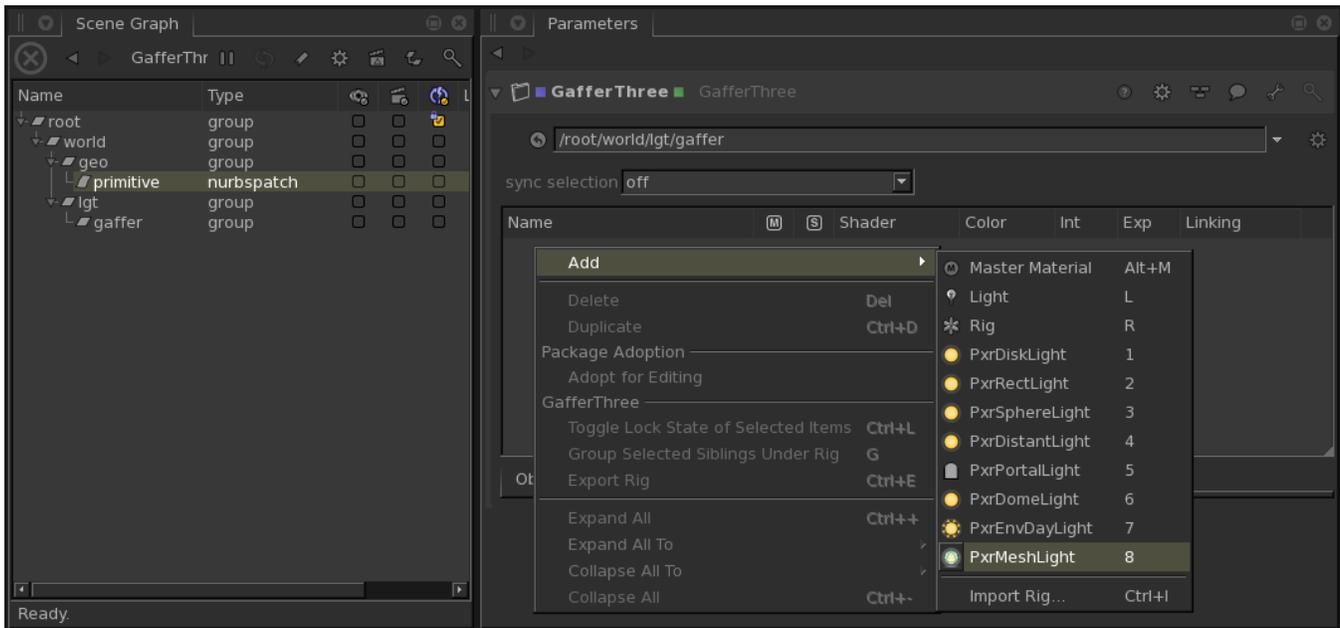
Creating Mesh Lights

Mesh Lights in RenderMan are created with a [PxrMeshLight](#). Any piece of geometry can be used as a mesh light (also referred to as a "geometric area light"). The two main workflows for creating mesh lights in RfK are described below.



Method 1: PxrMeshLight GafferThree Package

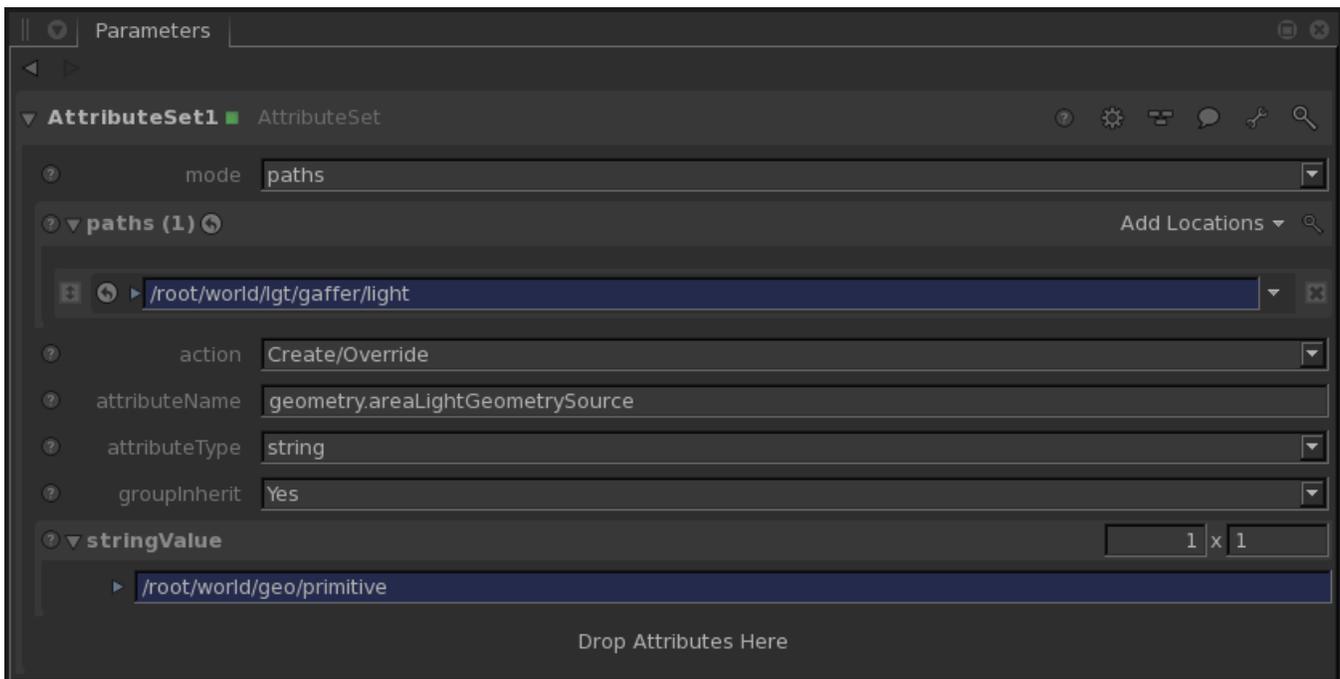
Creating mesh lights is simple with this method. First, select the geometry that you wish to turn into a mesh light in the Scenegraph. Then in GafferThree, add a PxrMeshLight.



The Mesh Light Geometry parameter can also reference a node with a Scene Graph location. Node Graph references will automatically update the parameter if the geometry's name or location changes.

Method 2: The `geometry.areaLightGeometrySource` Attribute

Mesh light geometry needs to be attached to the light using a specific attribute that RfK looks for when processing the lights. That attribute is named "geometry.areaLightGeometrySource". The attribute is a string attribute on the light and it's value should be the geometry location. For convenience, the `PxrMeshLightCreate` macro, a part of the RfK distribution, demonstrates this set up.



Additional Notes

The geometry can be rendered with or without a Bxdf. RfK assigns `PxrBlack` to mesh lights if no other Bxdf is assigned. If a Bxdf is assigned to the geometry then the geometry will be rendered according to the characteristics of that shader.

It is recommended that you increase the `micropolygonlength` on the mesh light for improved performance. This attribute is available on the `PxrMeshLight` GafferThree package in the Object Tab.

Unlike in previous versions of RenderMan, Mesh Lights now have a specific light shader. Where you could previously use any geometric area light as the light shader, you must now use `PxrMeshLight`.

If you want to use the Texture Color parameter of PxrMeshLight, you can create a shading network with PrmanShadingNodes and a NetworkMaterial node. Then assign the NetworkMaterial to the light location with a MaterialAssign node, downstream of the GafferThree where the light was created.