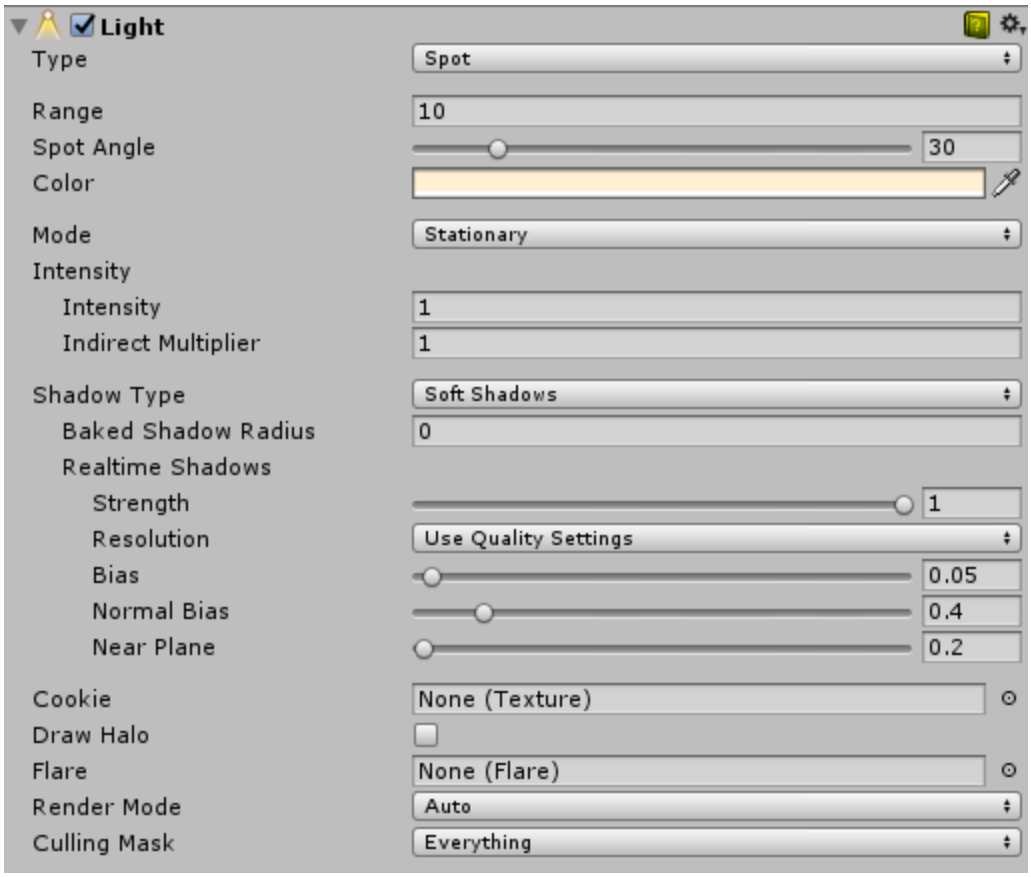
2017-06-17 THIS DOCUMENTATION IS SUPERSEDED BY THE UNITY USER MANUAL LIGHTING SECTION. Please note these draft documents will be deleted in the near future.

5.6 DRAFT DOCUMENTATION: This is first draft documentation for a feature that is new in Unity 5.6. As such, the information in this document may be subject to change before final release.

The Light Inspector

The light inspector main section should be updated as below :

Lights are a fundamental part of graphical rendering since they determine the shading of an object and the shadows it casts. See the Lighting and Global Illumination sections of the manual for further details about lighting concepts in Unity.



Properties

Property:	Function:

Type	The current type of light. Possible values are *Directional*, *Point*, *Spot* and *Area* (see the Lighting Overview for details of these types).
Range	How far light is emitted from the center of the object (Point and Spot lights only).
Spot Angle	Determines the angle (in degrees) at the base of a spot light's cone (Spot light only).
Color	The color of the light emitted.
Mode	Specifies the light mode used to determine if and how a light will be baked. Possible modes are Realtime, Mixed and Baked. See the *Realtime Lighting* page, *Mixed Lighting* page, and *Baked Lighting* page in the *[5.6 Beta] Light Modes* index page for further information about modes.
Intensity	Brightness of the light. The default value for a *Point*, *Spot* or *Area* light is 1 but for a *Directional* light, it is 0.5.
Indirect Multiplier	This allows you to vary the intensity of indirect light, i.e. light that is bounced from one object to another. The value is a multiple of the default brightness calculated by the GI system; if you set Bounce Intensity to a value greater than one then bounced light will become brighter with every bounce, while a value less than one will make it dimmer. This is useful, for example, when a dark surface in shadow (such as the interior of a cave) needs to be rendered brighter in order to make detail visible. Or alternatively, if you want to use Precomputed Realtime GI in general, but want to limit a single dynamic light to direct light only, you can set its Bounce Intensity to 0. See the Global Illumination section of the manual for further information.
Shadow Type	Determines whether *Hard Shadows Soft Shadows* or no shadows at all will be cast by this light.
Baked Shadow Radius	If shadows are enabled then this property adds some artificial softening to the edges of shadows cast by point or spot lights (in theory, light originating from a point casts perfectly sharp shadows but this situation rarely occurs in nature).

Baked Shadow Angle	If shadows are enabled then this property adds some artificial softening to the edges of shadows cast by directional lights (in theory, parallel light rays coming from a truly “directional” source cast perfectly sharp shadows but natural light sources don’t strictly behave like this).
Shadow Strength	Controls how dark the shadows cast by the light will be.
Shadow resolution	Controls the rendered resolution of the shadow maps. A higher resolution will increase the fidelity of shadows at the cost of GPU performance and memory usage.
Shadow Bias	Controls the distance at which the shadows will be pushed away from the light. Useful for avoiding false self-shadowing artifacts.
Shadow normal bias	Controls distance at which the shadow casting surfaces will be shrunk along the surface normal. Useful for avoiding false self-shadowing artifacts.
Shadow Near plane	Controls the value for the near clip plane when rendering shadows. Currently clamped to 0.1 units or 1% of the lights range property, whichever is lower.
Cookie	Specifies the Texture mask to cast shadows, create silhouettes, or patterned illumination for the light.
Draw Halo	If checked, a spherical halo of light will be drawn with a radius equal to Range. See also the page about the Halo component.
Flare	Optional reference to the Flare that will be rendered at the light’s position.
Render Mode	Importance of this light. This can affect lighting fidelity and performance, see *Performance Considerations* below. The options are *Auto* (the rendering method is determined at runtime depending on the brightness of nearby lights and current Quality Settings), *Important* (the light is always rendered at per-pixel quality and *Not Important*(the light is always rendered in a faster, vertex/object light mode). Use *Important* mode only for the most noticeable visual effects (eg, headlights of a player’s car).

Culling Mask	Use to selectively exclude groups of objects from being affected by the light; see Layers.