

Hidden Light Objects

This is a walkthrough of using a Hidden light object with GI and other rendering features in v8. This should help "connect the dots" between all of the settings necessary for setting up a Hidden Light Object.

I know you are thinking a couple of things, right away....

What is a "hidden" light object ... and why would I want one?

A hidden light object is a special use of geometry which is "linked" to an Area Light for photon emission. It does not create any illumination of it's own like a normal Area Light, it is only used for reverse illumination and shading of secondary points.

The reason for this feature is that skylight in Electric Image does not "bounce", and generally, brute force rendering methods which do allow skylight to bounce also tend to be inefficient. EI skylight provides direct illumination (and shadows) but no reverse illumination. This is true for the built in skylight (color only), textures used as skymaps, and high dynamic range images (HDRI) used as adaptive skymaps.

Photons emitted from an area light, however, bounce ALL OVER THE PLACE!

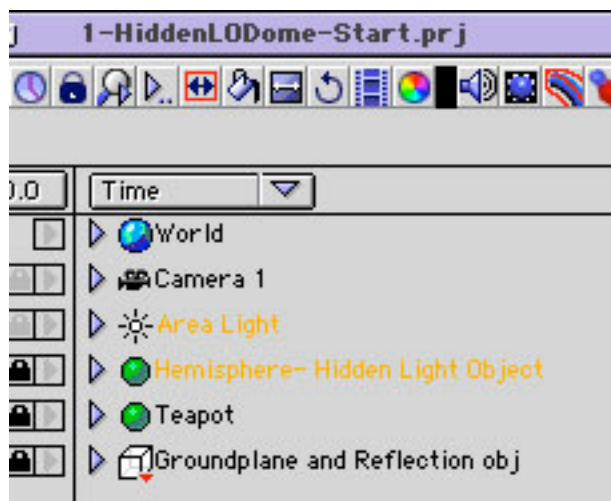
This is a good thing, since it allows for very realistic reverse illumination.

Another benefit is that these photons can be used for illumination of secondary points in a scene. This eliminates the need to calculate time costly secondary rays in the GI engine, and gives much more realistic results than using lights in "secondary" GI mode for faster reflections.

The sample project illustrates the use of a hidden light object to create reverse illumination, and secondary (reflected) points in combination with direct illumination from an HDRI adaptive skymap. Open 1-HiddenLODome-Start.prj

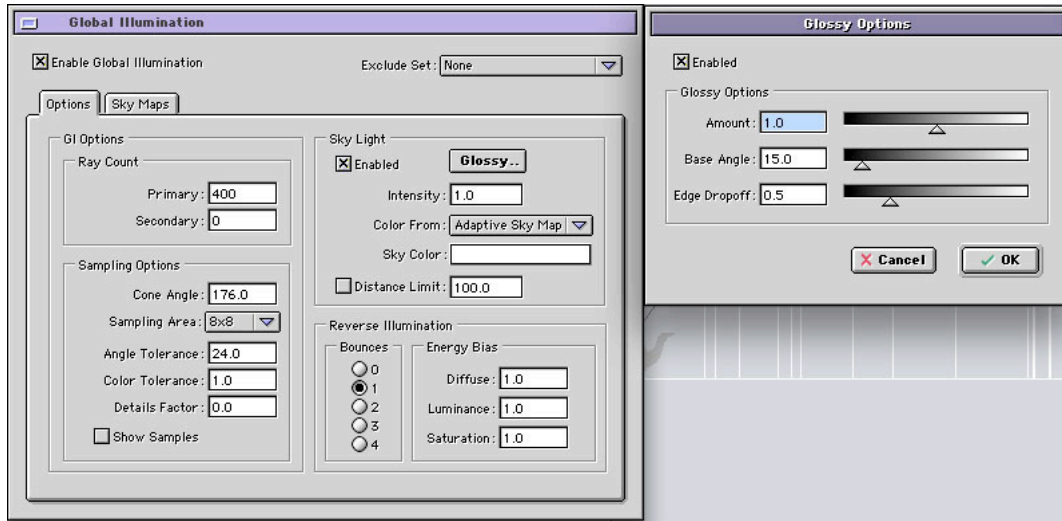
In this scene there are quite a few objects in a hierarchy below an effector.

These are the ground plane and some tall rectangular towers which should reflect on the teapot in the center of the camera view.

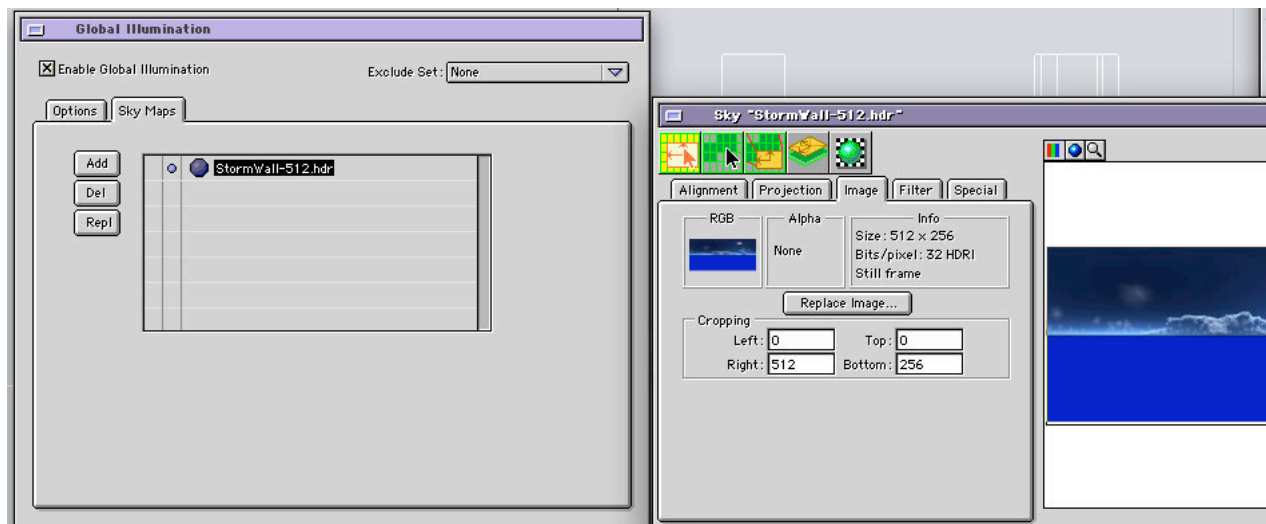


The Area Light and Light Object hemisphere are currently OFF in the project window. They are not contributing to the scene at all, at this point.

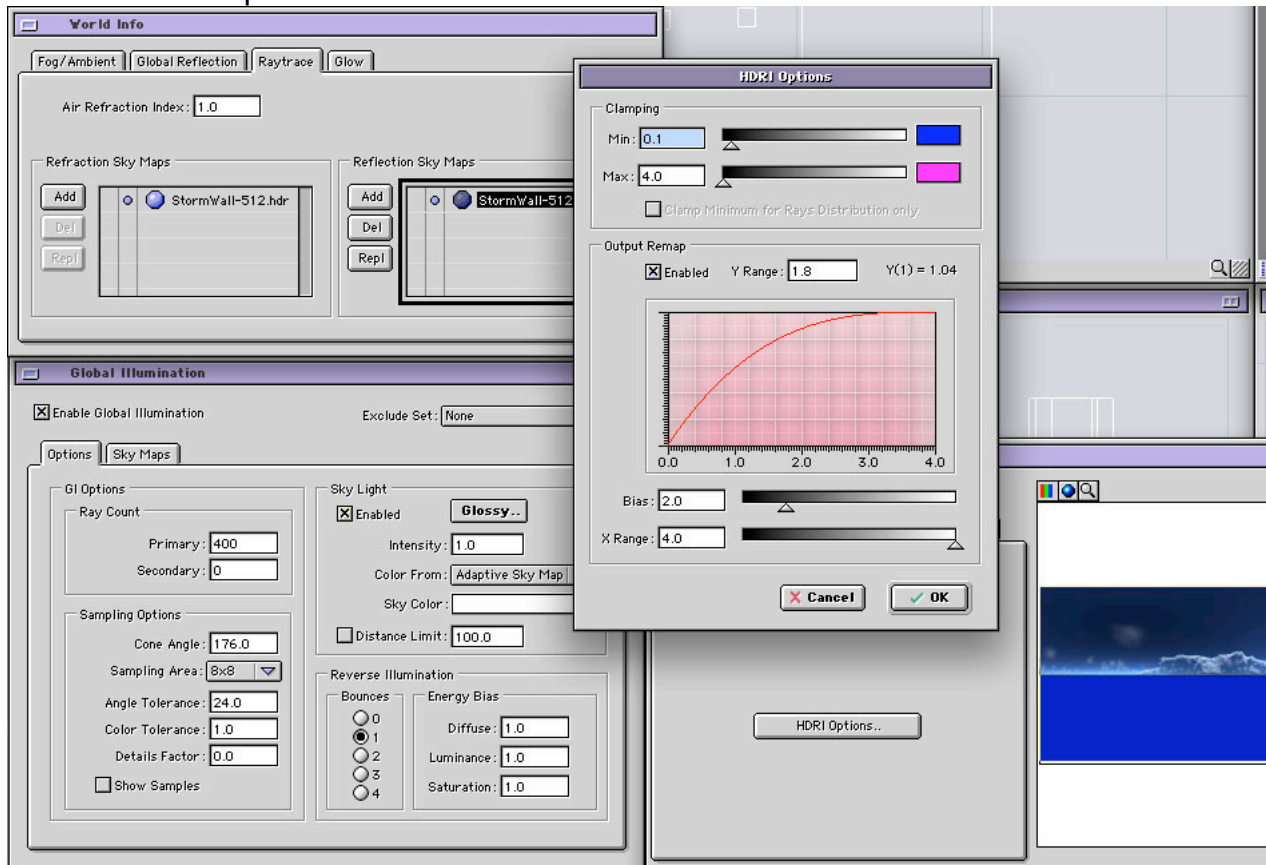
Let's review the current GI setup... notice GI Glossy is ON (for specular reflections from the sky) We have ZERO secondary rays, and reverse illumination is set to 1 bounce.



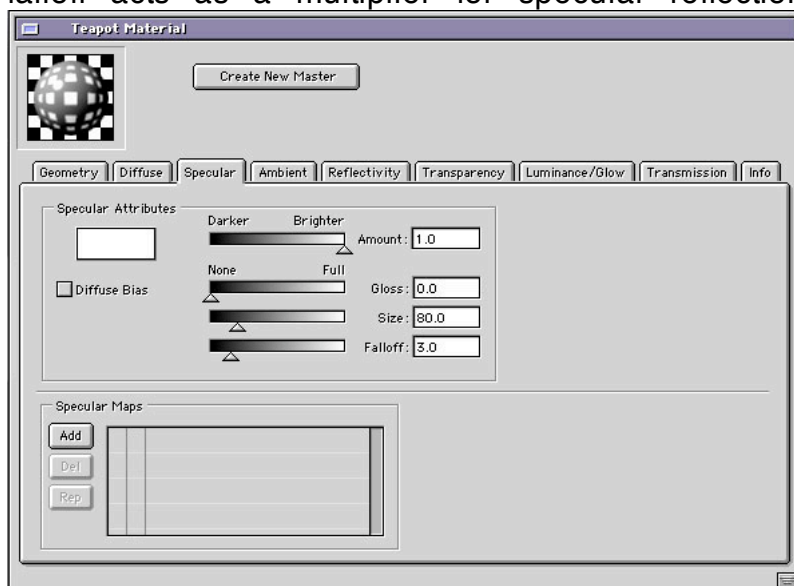
We have an HDRI adaptive skymap loaded...



In the world object we also have the same map loaded as Raytrace Refraction and Reflection maps. The re-mapping has been adjusted a bit here to tweak the HDR values of the map.



The Teapot itself has raytraced reflections enabled in the raytrace tab of the material editor, and the specular falloff has been set to 3.0. Any value above 1.0 in specular falloff acts as a multiplier for specular reflections coming from "GI Glossy".



If we render the scene, we can see the direct illumination of the skylight colored by the HDRI texture. We can also see the reflection map reflected in the teapot.



Notice that the rest of the scene reflected in the teapot is dark.

This is because we are not calculating any secondary rays with the GI engine, so the reflections of the objects are not being illuminated at all.

So let's enable our area light and hidden light object.

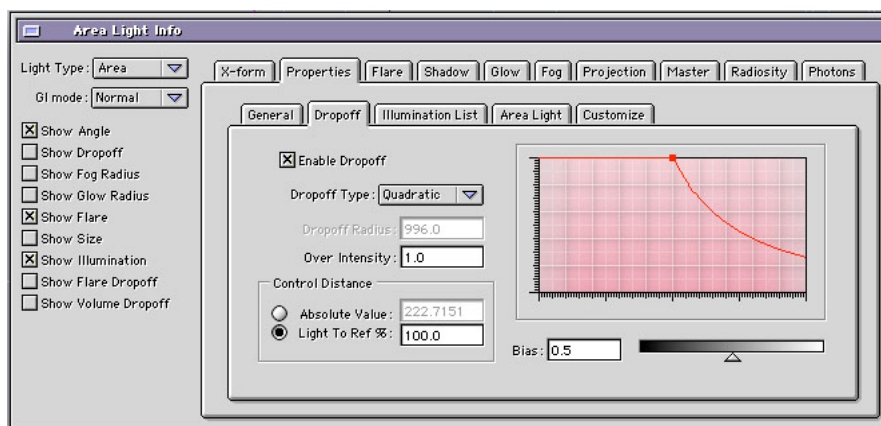
Check the visibility box for both in the project window.

The area light is positioned at approximately the radius of the hemisphere light object.

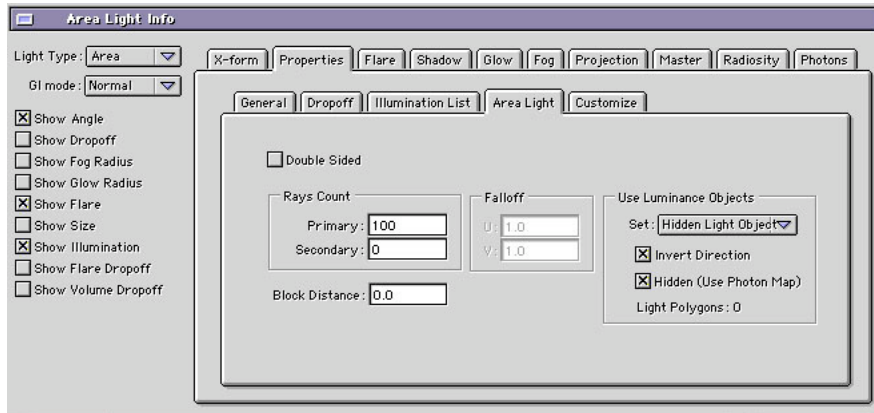
Looking at the Area Light settings- Properties Tab: Dropoff tab shows

Quadratic Dropoff enabled, with the distance set to "light reference distance", and the over intensity value set to 1.0 (essentially, no over intensity).

This will set the dropoff value to approximately the radius of the hemisphere (based on where the area light was placed).



Next we look at the Area Light tab...



On the right side is a box "Use Luminant Objects".

This is where the Area Light is "linked" to objects that are in a selection set.

In this case the selection set is "Hidden Light Objects".

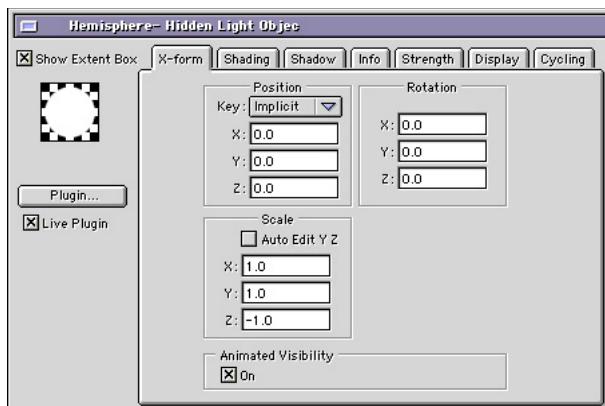
The Invert Direction checkbox is ON, which just reverses the normals of the light object to emit photons inwards, instead of out.

The "hidden" checkbox is also ON. This makes the object emit photons, but NOT contribute to the direct illumination of the scene. For all practical purposes this makes our our light object invisible, and only leaves behind the photons it emits.

Let's look more closely at our light object.

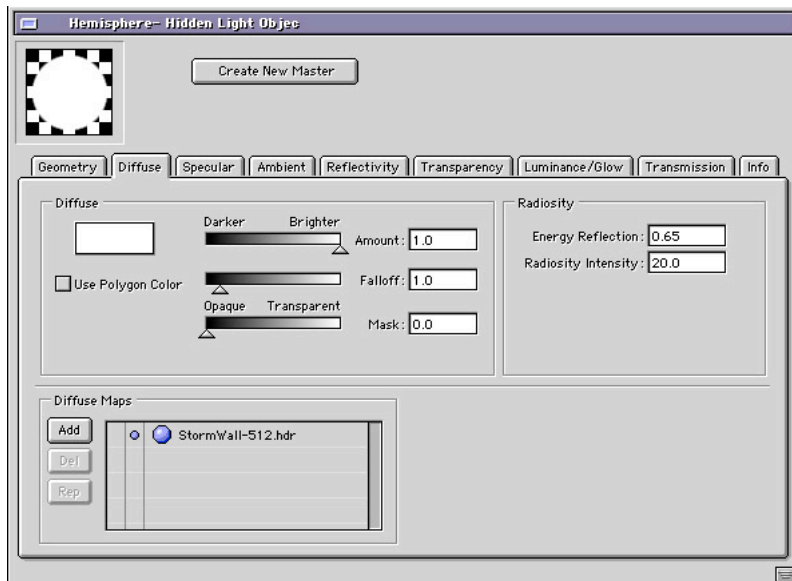
Double click on the Hemisphere Light Object and open it's material editor.

The object is set to be 100% luminant in the luminance/glow tab, and it has the skymap texture applied as a spherical diffuse map. One trick to this is that the object's Z scale must be set to negative. This changes the orientation of the map so that the mapping matches the skydome. The reason for this difference is that most spherical maps are oriented "outside" the object, while skymaps are oriented "inside".

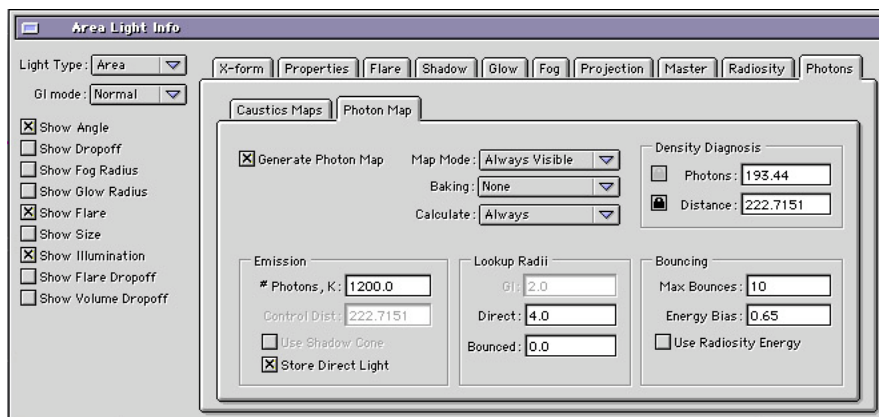


This texture will cause the photons to be colored to closely match the direct illumination coming from the GI Skymap.

It is important to understand that the color of the photons are based on the color of the individual polygons of the light object, NOT the individual pixels in the texture map. Each polygon takes an average of the texture's pixels that cover the polygon's surface and this average becomes the color of the photons emitted from that polygon.



Next we look at the light's Photons tab...



First- the "Generate Photon Map" is enabled, and 1200 is entered into the Emission # Photons box. This number is in "thousands" (K). So, in this case, there are 1.2 million photons.

First we need to visualize our photon map and tweak the settings. The Map Mode is set to "always visible" which means that Camera will render the actual photon map itself. "Baking" is set to none and "Calculate" is set to always.

During this photon map visualization, "Enable Global Illumination" must be DISABLED in the GI dialog box! Below is our first photon map rendering.

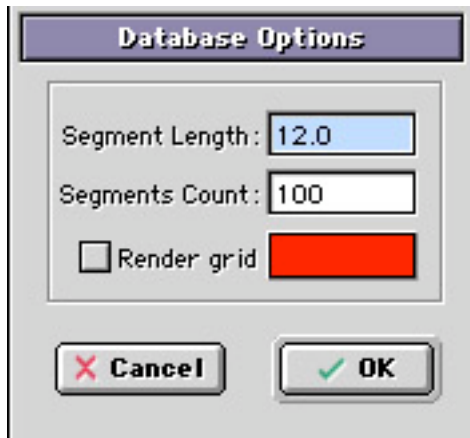


Notice that the towers reflected in the teapot are no longer completely dark! Our photons are visible as "pre-calculated" secondary points in reflections and refractions. However, the photon map is very spotty, so let's increase our photon radius just a bit. With our "Direct" Radius set to 8 world units our map is much smoother. There are no gaps between the photons, although the map is not completely smooth.

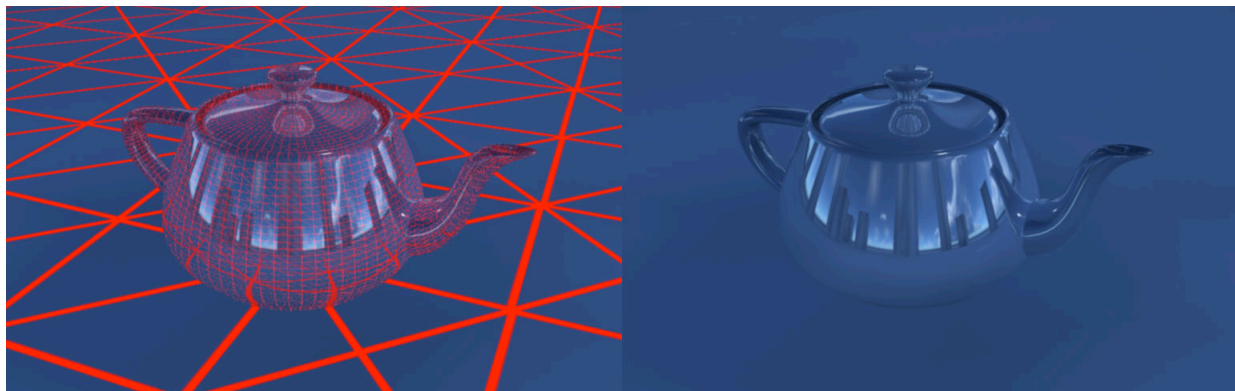


At this point we can use the "baking" feature with "database" to completely smooth the map. This feature creates a subdivision mesh which overlays the existing model. This "database" of points is then sampled and used as the final Photon Map. This mesh exists separately from the model mesh- so there is no actual change to the geometry in the scene and no render time penalty for raytracing due to the increased mesh density in the database.

Choosing "Database..." from the Baking menu brings up the Database Options Window.



Segment length is the size in world units of each subdivision segment.
Segment count is the maximum # of subdivisions allowed.
Checking the "render grid" box allows visualization of the database mesh.
Below are renderings with and without the database grid.



With a greater number of photons (perhaps many millions!) the photon radius and database segment length can both be reduced, and the Photon map can have a great deal more precision. Details in the shading will appear in the photon map (like detailed ground shadows). But for many scenes this very smooth map is very useable, renders quickly, and produces very pleasing results.

Since we now have a nice smooth photon map- it's time to re-enable GI and render! Check the "Enable" box in the GI dialog, then return to the Area Light's photon tab and set the map mode to "GI & Secondary". This tells Camera to feed the photon map to the GI engine and also to use the photon map for secondary points.



Now we can see the reflected towers. Using the hemisphere as a hidden light object has allowed for better reverse illumination in the scene as well as very fast and realistic shading of secondary points.