

REALITY2

User's Guide

VERSION 2.00

Table of Contents

<u>Welcome.....</u>	<u>4</u>
<u>What's new in version 2.0.....</u>	<u>5</u>
<u>Installing Reality.....</u>	<u>7</u>
<u>Important!.....</u>	<u>7</u>
<u>Installation.....</u>	<u>10</u>
<u>Configuring Reality to use Lux.....</u>	<u>12</u>
<u>Migrating the ACSEL Database.....</u>	<u>13</u>
<u>Using Reality.....</u>	<u>13</u>
<u>Sun at midday.....</u>	<u>15</u>
<u>Sunrise/Sunset sun.....</u>	<u>15</u>
<u>The render never stops.....</u>	<u>16</u>
<u>When is the right time to stop Lux?.....</u>	<u>17</u>
<u>The exported scene files.....</u>	<u>17</u>
<u>Key concepts.....</u>	<u>18</u>
<u>Unbiased vs. biased renderers.....</u>	<u>18</u>
<u>Creating soft shadows with Reality.....</u>	<u>18</u>
<u>Indirect lighting is always on.....</u>	<u>19</u>
<u>Reality's materials.....</u>	<u>19</u>
<u>Editing materials.....</u>	<u>20</u>
<u>Changing multiple materials.....</u>	<u>23</u>
<u>Multi-edit.....</u>	<u>24</u>
<u>Save your Scene.....</u>	<u>24</u>
<u>Merging Scene.....</u>	<u>24</u>
<u>A hidden gem: The UV Map Viewer.....</u>	<u>25</u>
<u>Lux materials are simpler.....</u>	<u>26</u>
<u>See before you render: The Material Preview.....</u>	<u>26</u>
<u>The material snapshot.....</u>	<u>28</u>
<u>Resetting materials to the original state.....</u>	<u>29</u>
<u>Glossy.....</u>	<u>31</u>
<u>Glossiness/Specular.....</u>	<u>31</u>
<u>Alpha Channel.....</u>	<u>33</u>
<u>Modifiers.....</u>	<u>33</u>
<u>Matte.....</u>	<u>36</u>
<u>Matte Translucent.....</u>	<u>36</u>
<u>Glass.....</u>	<u>39</u>
<u>Metal.....</u>	<u>42</u>
<u>Metal.....</u>	<u>42</u>
<u>Mirror.....</u>	<u>43</u>
<u>Velvet.....</u>	<u>43</u>
<u>Water.....</u>	<u>43</u>

Null.....	44
DAZ Studio's SubD and Reality.....	44
ACSEL	45
Quick ACSEL tutorials.....	46
Save with textures.....	51
Deleting a shader.....	51
ACSEL Database maintenance.....	53
ACSEL Share.....	53
Installing a shader set.....	54
Sharing a shader set.....	54
A message for all content creators (PAs).....	55
Light control.....	56
Light types.....	58
Converting materials to lights.....	61
Light Editor.....	62
IES Lights.....	65
Color Temperature.....	66
Image Based Lighting (IBL).....	67
Which IBL format?.....	67
Trying out IBL in Reality.....	68
Setting your own IBL map.....	68
Using images with IBL: HDR vs. LDR.....	69
IBL without images.....	70
Light portals.....	70
House lights.....	72
The Lights tab.....	73
Light groups.....	73
Custom light groups.....	73
Camera control.....	74
Selecting the rendering camera.....	74
Shallow depth of field.....	74
Exposure controls.....	75
Film Response.....	76
Output options.....	77
Output path.....	78
File type.....	78
Alpha Channel.....	78
Preserve Light hue.....	79
Overwrite Warning.....	79
Collect Textures.....	79
Texture Size.....	79
Geometry Format.....	80
GPU Acceleration.....	80

<u>Use SLG for Rendering.....</u>	<u>80</u>
<u>Moving around.....</u>	<u>83</u>
<u>Scene type.....</u>	<u>84</u>
<u>Scene Parameters Editor.....</u>	<u>85</u>
<u>Memory conservative.....</u>	<u>86</u>
<u>Preview.....</u>	<u>87</u>
<u>Max samples.....</u>	<u>87</u>
<u>FF Removal.....</u>	<u>87</u>
<u>Gamma.....</u>	<u>87</u>
<u>Start Frame/End Frame.....</u>	<u>88</u>
<u>Animation rendering with SLG.....</u>	<u>89</u>
<u>LuxRender options.....</u>	<u>89</u>
<u>Advanced Options.....</u>	<u>89</u>
<u>The About tab.....</u>	<u>92</u>
<u>Keeping Reality updated.....</u>	<u>92</u>
<u>Best practices for exporting your scene.....</u>	<u>93</u>
<u>Hide what you don't see.....</u>	<u>93</u>
<u>Start simple.....</u>	<u>93</u>
<u>Fog.....</u>	<u>94</u>
<u>Saving your Reality settings.....</u>	<u>97</u>
<u>Tonemapping.....</u>	<u>97</u>
<u>Reinhard.....</u>	<u>98</u>
<u>Linear.....</u>	<u>98</u>
<u>Converting from Reinhard to Linear.....</u>	<u>99</u>
<u>Render as Statue.....</u>	<u>102</u>
<u>Stopping and resuming renders.....</u>	<u>102</u>
<u>Known issues.....</u>	<u>103</u>
<u>Acknowledgements.....</u>	<u>105</u>
<u>Stay updated with Reality!.....</u>	<u>105</u>
<u>Appendix A - Converting the ACSEL Database</u>	<u>106</u>
<u>Appendix B - How to use the Softbox.....</u>	<u>107</u>
<u>Appendix C - Network rendering.....</u>	<u>108</u>
<u>Appendix D - sIBL License.....</u>	<u>109</u>
<u>Appendix E - Reality plug-in software license.....</u>	<u>110</u>

Welcome

Thank you for purchasing Reality, we hope that you will have fun using it.

Reality is a plug-in for DAZ Studio that connects Studio with LuxRender, the Open Source physically-based, unbiased rendering engine. This makes it possible for Studio artists to use an incredible array of features from Lux that can bring very high levels of realism to your scenes. Some of the features provided to you by Reality are:

- ◆ A wide selection of physically-based materials like: “Water,” “Glass,” “Velvet,” “Glossy,” “Metal,” “Fog,” “Mirror,” etc. These materials are adjustable within the Reality User Interface (UI) to create an infinite variety of materials.
- ◆ Single-light, physically-based Sun light with indirect lighting, caustics, etc.
- ◆ Spotlight, mesh-based lights, point lights that can be used to simulate any real-life lighting fixture.
- ◆ Image Based Lighting that uses real HDR images.
- ◆ GPU Acceleration.
- ◆ Selection of two rendering engines: LuxRender or SLG.
- ◆ Lighting can be changed while the image is rendering. Render status is saved automatically and can be stopped and resumed at a later time.
- ◆ Material preview right inside the Reality UI. For some materials, such as V4’s Lips and Eyes, a special “shape” preview is provided that gives a more accurate version of the final result.
- ◆ Automatic conversion of Studio’s materials to Reality / Lux.
- ◆ Volumetric fog.
- ◆ Cloud-based service for sharing and installing custom Reality shaders.
- ◆ Network rendering. Render nodes can be running Mac OS, Windows or Linux, making it possible to create low-cost renderfarms.
- ◆ Rendering runs independently from Studio. You can continue working on your scene while the rendering runs.
- ◆ Multi-core support, 32 and 64 bit support for Mac OS (Lux), 32 and 64 bit version of the plug-in for Windows.
- ◆ Presets for dozens of materials from real-life measured data. For example, Reality includes all the Index Of Refraction presets from real glass, water at different temperatures, the human cornea, gemstones, gasses, and others. These presets can be used to create realistic effects of light passing through, for example, a ruby or a diamond.

What's new in version 2.0

Version 2.0 introduces many new features and improvements. Notably, this is the first version of Reality that runs on DAZ Studio 4. Studio 3 compatibility is also maintained.

Here are all the major improvements:

1. DAZ Studio 4 and Studio 3 compatibility.
2. Mac OS 64 version for DS 4.
3. LuxRender GPU support.
4. Support for Lux 0.9
5. Animation support for Lux.
6. Blazing fast, 100% GPU-base rendering via SmallLux GPU (SLG) renderer.
7. Animation support for SLG.
8. Up to 100% speed improvement in exporting.
9. Multi-edit. Select multiple materials and edit the common parameters.
10. New light editor provides all the parameters to customize your lights, including the size of the sun.
11. Mesh lights now support textures, including transparency.
12. Support for texture with Spotlights.
13. Support for IES lights.
14. Support for Lux light presets (spectrum data).
15. Support for binary PLY files in export.
16. Built-in texture collection with optional downscaling.
17. Automatic, non-blocking material preview.
18. ACSEL shaders load 10 times faster than before.
19. Improved material preview for Glass materials.
20. Material Snapshot: do a "before and after" material comparison.
21. Material preview is cached up to 20 materials. No need to recalculate the preview when switching from material to material.
22. Much faster material converter.
23. Support for flat shading.

24. Advanced tab for fine tuning your preferences.
25. Start-up time is much faster.
26. Improved ACSEL Database.
27. The material editor now shows the name of the ACSEL shader loaded.
28. Now you can delete an installed ACSEL shader.
29. The material editor now shows which materials have been edited.
30. Selecting a material in Reality shows the same material in the Studio's Surfaces tab.
31. Added editable "Fireflies removal" parameter.
32. Support for new Studio 4 LinearPointLight.
33. Added support for level of roughness of the Matte material.
34. Metal material now supports textures.
35. New, more informative About box includes reference to the location of the ACSEL database, the location of the "Reality User's Guide" and links to video tutorials.
36. Added support for Lux's PhotonMap (ExPhotonMap).
37. Added support for Lux's SPPM.
38. Added an advanced editor for Lux's rendering parameters.
39. Material preview type is now persistent.
40. New Face shape for the material preview.
41. Now reality automatically selects the best material shape for the preview.
42. Tripled the number of metal presets.
43. Added support for normal maps.
44. Added "Save scene" button to save your Studio scene directly from inside Reality.
45. Added support for Lux's new Metal material.
46. Added selection of Film response directly in the Camera tab.
47. Selecting a file now remember the last folder visited.
48. Detection of the Studio lighting type: metallic will cause a material to be converted to metal. Similarly a Matte shading will create a Matte material.

49. Added detection of Studio “Brick” shaders. They are not used but now they don’t crash Reality.
50. Automatic notification of Reality updates.
51. Better, easier installation procedure. Now the path for the add-ons is added automatically to your configuration.
52. Better, easier, much improved UI.
53. Added f32 aperture for both camera parameters and DOF controls.
54. Support for Studio 4 SubD.
55. New light: “Curved Light”.
56. Added support for Studio’s Scene merge.
57. Added support for creation of Reality Light sets.
58. Improved Installers.
59. Control on Lux’s write and refresh intervals.
60. Scene overwrite prevention.
61. Glass reflection map support.
62. Studio’s second specular support.
63. Improved accuracy of Studio’s material conversion.
64. Simpler image selection option.
65. “SMASH” (Smart Material Analysis and Scene Harmonization) technology adapts the Lux render options based on the materials used.

Installing Reality

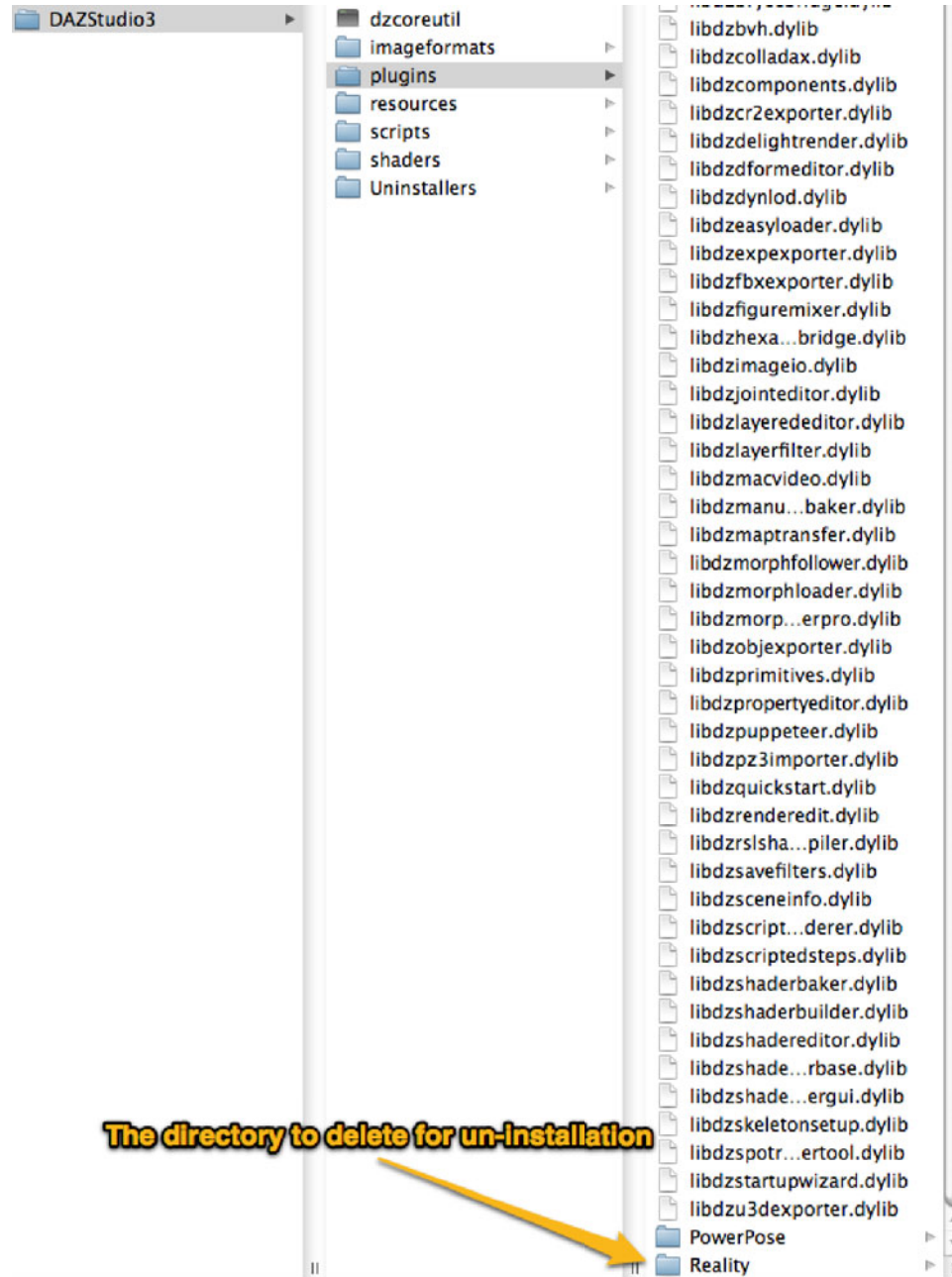
Important!

If you used Reality 1.0, 1.2 or 1.25 you **MUST** uninstall it before installing Reality 2.0. The uninstallation is very easy:

1. Navigate to the Studio directory:
 - For Mac OS: /Applications/DAZ 3D/DAZStudio3

- For Windows: either C:\Program Files\DAZ 3D\DAZStudio3 or C:\Program Files (x86)\DAZ 3D\DAZStudio3
2. Locate the directory named plugins.
 3. Inside that directory there should be a directory named Reality. Delete it. Please be sure that you delete the Reality directory, and not the plugins directory.
 4. Navigate to the Studio directory:
 - For Mac OS: / Applications/DAZ 3D/DAZStudio3
 - For Windows: either C:\Program Files\DAZ 3D\DAZStudio3 or C:\Program Files (x86)\DAZ 3D\DAZStudio3
 5. locate the directory named scripts.
 6. Inside that directory there should be a directory named Reality. Delete it. Please be sure that you delete the Reality directory, and not the scripts directory.

Here is an example of the directory in the plugins folder:



Now you can run the installer for Reality 2.0.

Installation

Reality runs on both Mac OS and Windows (32 and 64 bit versions). When you purchase the plug-in, either directly from Pret-a-3D or from DAZ 3D, you will receive an installer program and a serial number in your email. The installers come as either a .dmg file, for Mac OS, or as a zip file, for Windows.

If you use Mac OS just double-click on the .dmg file. It will be mounted and shown in the Finder. Inside you will find the installer itself and a Readme document. Please read the document before performing the installation.

If you use Windows double-click on the zip file. Windows explorer will show you the content of the archive which should include the installer itself (an executable program), and the Readme document. Please read the document before performing the installation.

1. Run the installer program that is appropriate for your OS (Mac OS or Windows) and your version of Studio (32 or 64 bit). Please note that for DS3 on Windows and DS4 on Mac OS you can use either the 32 bit or 64 bit version of Studio regardless on the “bitness” of your OS. For example, you could install and run Studio 32 bit on the 64 bit version of your OS. Only Studio 3 Advanced provides a 64 bit version and Studio 4 provides a 64-bit version for the Advanced and Pro versions. When installing Reality you must match the version of the plug-in with Studio. So, use 32 bit Reality for 32 bit Studio and 64 bit Reality for 64 bit Studio.
2. After you finish installing Reality you will need to install LuxRender, which is provided in another installer and is included with your copy of Reality. LuxRender is the external program that renders your scenes exported by Reality.
To avoid duplications and to minimize your download time the LuxRender installers are provided separately from Reality. You should have received an email with the links for downloading Lux, but here are the links again:

- Mac OS:
www.preta3d.com/files/luxrender/0.9/LuxRender_0.9dev_universal_install.zip
- Windows 32-bit without OpenCL:
www.preta3d.com/files/luxrender/0.9/LuxRender_0.9dev_32_NoOpenCL_install.exe
- Windows 32-bit with OpenCL:
www.preta3d.com/files/luxrender/0.9/LuxRender_0.9dev_32_OpenCL_install.exe
- Windows 64-bit without OpenCL:
www.preta3d.com/files/luxrender/0.9/LuxRender_0.9dev_64_NoOpenCL_install.exe
- Windows 64-bit with OpenCL:
www.preta3d.com/files/luxrender/0.9/LuxRender_0.9dev_64_OpenCL_install.exe

Windows versions of Lux are available for 32 or 64 bit variants and for graphic cards that have OpenCL support or not (please note that OpenCL is different from OpenGL).

Mac OS 10.6 and above has built-in OpenCL support so this is not an issue for Macintosh users.

Please note: the LuxRender installer might mention some installable scripts. Unless you use Blender 3D, a 3D modeling application, you must **uncheck** the installation of these scripts. These files are not related to Reality or DAZ Studio.

3. If this is the first time that you're installing any version of Reality, you will need to register this plug-in. If this is an upgrade from version 1.x you will need to register as well since the license data have changed.

To register the Reality plug-in you just need to start Studio. The registration window will be shown after Studio finishes loading and there you will be able to enter your information. Please verify all the data before submitting. A real name and a valid email address are required to successfully register Reality.

Now you should also be able to see the Reality menu option in the Render menu of Studio:



Configuring Reality to use Lux

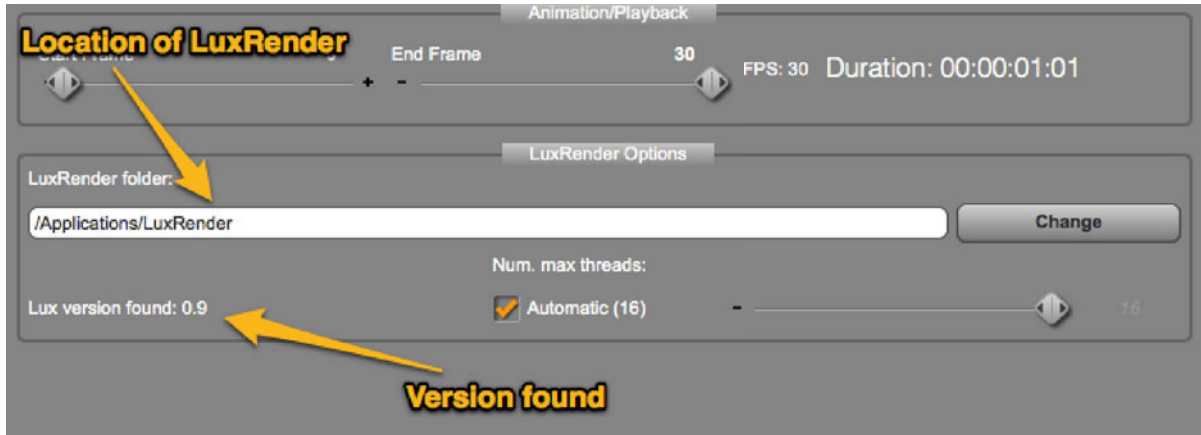
Since Reality has to call LuxRender to do the actual rendering it's important that the location for Lux is configured correctly. Lux is included in the installer for Reality but if you have it already installed on your system, or you installed it in a custom location, you need to tell Reality where the renderer is.

First, check that LuxRender runs correctly on your system. To do this, locate in which directory it is installed and run the program called LuxRender (on Mac OS) or luxrender.exe (on Windows). If the program starts, close it and take note of the location. If there is a problem, check your installation and possibly re-install Lux.

Important! Note that Reality 2.0 **requires** LuxRender 0.9. Older versions are not supported and will cause problems.

Call Reality and in the Output tab set the location of LuxRender by clicking on the "Change" button and navigating to the directory where LuxRender is installed. Once done, Reality will record the location in the DAZ Studio application-wide con-

figuration. The version of Lux is verified and displayed below the path to the executable, in the Output tab.

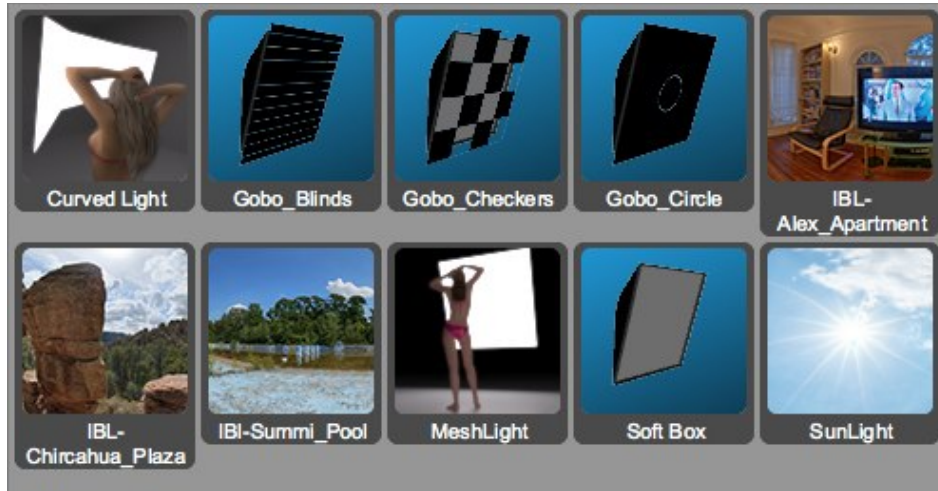


Migrating the ACSEL Database

If you are upgrading from Reality 1.x you will need to update the ACSEL Database since the format for the ACSEL shaders has changed. Please see Appendix A for how to perform this operation.

Using Reality

Reality comes with a set of Add-ons, lights, props and scripts, that are automatically installed in your System and added to the Studio Content Manager. In Studio 4 they are located under the "DAZ Studio Formats" container, with the name "Reality-Add-ons". In there you will find, for example, a pre-made Sun light that is very convenient for exterior scenes. Also, the Mesh light is generally an excellent 1-click solution for beautiful portrait lighting. Here is the Light subfolder.

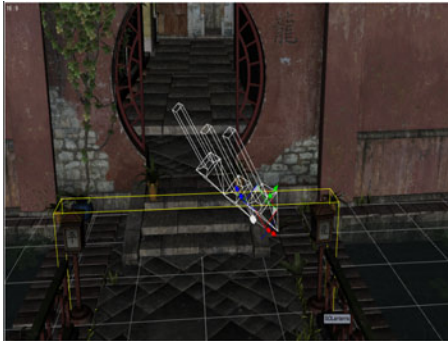


The workflow for using Reality is simple. The basic steps are:

- **Create the scene.** First you create your scene in Studio as you would normally. Don't spend too much time setting the materials in Studio as you will be doing all the fine-tuning in Reality. Setting the basic textures for characters and props is generally all that you have to do.
- **Add your lighting.** While you become familiar with the Reality way of working, use one or more Mesh Lights. The Mesh Light is installed in the Reality-Add-ons folder that you find in Studio's Content Manager. Mesh Lights are geometry objects that emit light. A mesh light generally provides a very nice, diffuse light that projects soft shadows. It produces that most pleasant result with the least amount of work. Please note that Studio will not recognize the Reality mesh light as a light so you will not see the effect of it in the Studio preview. Simply orient the front of the Mesh light to point in the direction you want the light to shine. If you need sunlight simply add the Sun light provided with Reality.
- **Launch Reality.** After you're done setting up the lights, launch Reality. You can launch Reality by either:
 - ▶ Clicking on the "Reality Render" option in Studio's Render menu:
 - ▶ Or by pressing the keyboard shortcut Cmd+Shift+L (Mac OS) or Ctrl+Shift+L (Windows).

When invoked, Reality will recognize the lighting you added. For example if you added the Sun light, it will configure a lighting system for Lux that includes the Sun light and the sky. Keep in mind that the angle of the sunlight will determine its look. For example, if the sun is pointing down at a sharp angle, between 20 to 45 degrees, it will be interpreted as midday sun. On the other hand, if you make the sunlight horizontal Lux will interpret it as the sun at sunrise or sunset, and therefore the light will have a distinctive orange shift.

Sun at midday



Sunrise/Sunset sun



- **Modify.** The Reality UI provides full control over the final result of the render. You can modify the type of materials used for each material zone defined in Studio and fine-tune all the parameters involved.

- **Render.** For your first render simply click on the “Render frame” button. When you do that Reality exports the Studio scene to a series of files on disk using the LuxRender format (see the Output tab for setting your names for the exported scene and for the image file) and then calls Lux to render that scene. In just a few seconds you will see the LuxRender window appear. Reality has been always very fast in exporting the scenes but for version 2.0 we have optimized the exporter even more and it's now 100% faster than in version 1.25. In addition it also takes less memory.

Lux renders a scene in a different fashion than the Studio built-in renderer. Instead of starting from the first line at the top and then progressing toward the bottom of the image, Lux renders the whole image in one pass, which usually takes a few seconds. This first render is very coarse but it gives an overall idea of the result. In a few seconds Lux will refine the image and the cycle continues until you quit Lux. Lux **saves your file automatically to disk**, using the PNG format by default. You can change the location and name of the file in the Reality “Output” tab. Lux saves the file at intervals of three minutes and when you quit the program the latest version of the render is saved automatically.

You can see Reality in action in this video tutorial:

http://www.youtube.com/watch?v=2xX-_nUjla0&list=UUSiSigiy-SC2XK8IR187Ru2Q&feature=plcp

The tutorial is for Reality 1.0 but most of the concepts still apply.

The render never stops

If you are not familiar with Lux, the above heading might seem a bit odd. Lux works by collecting light samples in an endless cycle. The renderer doesn't make the decision for you about when to stop; you stop the renderer when you are satisfied with the image. Many different factors can determine the time required for a satisfactory render. When you close Lux, the rendered image is saved on disk, in the image file that you have specified in the Output tab of Reality. In addition, Lux automatically saves the image every three minutes while rendering.

When is the right time to stop Lux?

Given that Lux doesn't stop automatically at an arbitrary point, you might wonder when is the right time to stop the render. The short answer is "when it's good enough for you." Until you gain more practice with Lux it's impossible to give an answer to what is the acceptable level of quality, expressed in time. Each pass in the rendering process doesn't just refine the "grain" of the image but it adds quality to the light that is simulated. For example, glass reflections will look more realistic, and subtle sheens in the materials will catch elements from the surrounding objects.

The status bar of Lux provides some good metrics to judge the render. The section marked "Statistics" lists a series of numbers. Look at the one labeled "S/px", as in "2422 S/px." That is the number of "samples per pixel" and it's a good indicator of the overall quality of the image. Depending on the scene the image might be considered finished at 500 spx or at higher values, like around 2,000-3,000 samples per pixel. Experience will show you the right value. Keep in mind that even when the image is free of noise the longer you let it render the better it becomes, because Lux adds subtle reflections and light effects that add realism beyond the simple clarity of image.

The exported scene files

Reality performs its task by reading all the information about the Studio scene and then generating a scene file using the Lux language. After the scene file is created, Reality then calls Lux and points the program to the scene file generated. Rendering starts as soon as the scene file is read, the rendering image will appear on screen, and this image will be automatically saved to disk.

The location and name of the scene file, ending in extension `.lxs`, is specified in the Output tab. You can change the name and the location to anything you want. By default Reality will place the file in your Desktop folder and will call the generated scene file "reality_scene.lxs". The image rendered by Lux is saved automatically to a PNG file, by default, and the name and location of that file is also specified in the Output tab, just below the name of the scene file. By default the image file is also placed in the Desktop folder and it's called "reality_scene.png". Please note that the

extension is not specified in the name because it is defined by the file type. You can save a rendered image in PNG, OpenEXR (.exr) or Targa Format (.tga).

Key concepts

Unbiased vs. biased renderers

LuxRender is a “physically-based,” “unbiased” renderer. If you are not familiar with these terms, it’s worth spending a few seconds learning how they affect your work. Most rendering engines, including 3Delight (the renderer included in Studio) and Firefly (the renderer of Poser), are biased. What that means, in simple terms, is that you can drive the renderer to create effects that are not possible in real life. For example, in Studio you can arbitrarily decide if a light projects shadows and if the shadow is soft or hard. In real life, if you turn on a light, that light will cause shadows. You cannot change that. And if that light is small, in comparison to an object, the shadow will be sharp. If the light is large, like in the case of a big photographer’s softbox, the shadows will be soft. This is a law of physics that you can use to your advantage, and these “physically based” behaviors are the basis of “unbiased” renderers like LuxRender.

Creating soft shadows with Reality

Remember, sharpness or softness of the shadows is not based on the brightness of the light but on the **size**. For example, the Sun is a very small light source in reference to people and even buildings. Of course we know that the Sun is several times the size of the Earth, but because of the distance it appears as a point in the sky. During a bright sunny day shadows are sharp. If you were to create a Studio scene and decided to set the sun with soft shadows, while visibly showing an outdoor, sunny day, that scene would look unrealistic. People might not know immediately what was the problem but they would perceive that something was not quite right.

During an overcast day shadows are very faint, to the point of not being visible. That is because the layer of clouds between the Earth and the Sun diffuses the light and acts as a light source; a huge, all-covering, light source. Large light sources create soft shadows. The larger the light, the softer the shadow. If you want to create soft

shadows in your Reality scene stay away from sunlight and spotlights and use a Mesh Light or a Soft box, both provided in the Reality add-ons. Any polygonal mesh can be turned into a mesh light but simple planes are usually the best option because they only have one face. When using a mesh light each face/polygon is a light source and each light source will add more time to your render. Unless you need very special shapes for the light source, the Mesh Light add-on will work perfectly. If you need more control you can use the Softbox add-on. Appendix A explains how to use the Softbox.

If you need to use the Sun and want to have softer shadows, like during an overcast day, you can simulate that situation in Reality. Contrary to what many believe, “dimming” the sun doesn’t give soft shadows. Remember, the brightness of a light doesn’t determine the sharpness of the shadows, the size does.

Reality includes a Diffuser prop. The Diffuser is just a pre-made plane that is configured to use the Matte Translucent material. This material lets light go through it practically acting as a diffuser and becoming a large light source. Place the Diffuser between the Sun and your subject and you will obtain softer shadows.

Indirect lighting is always on

Light, in real life, bounces off objects and “touches” surrounding objects, picking up colors and other characteristics. For example, light hitting a metal surface will be reflected quite strongly and will light nearby objects. This is called indirect lighting. In Lux indirect lighting is always active and so your scene will require generally fewer lights than when using Studio’s built-in renderer.

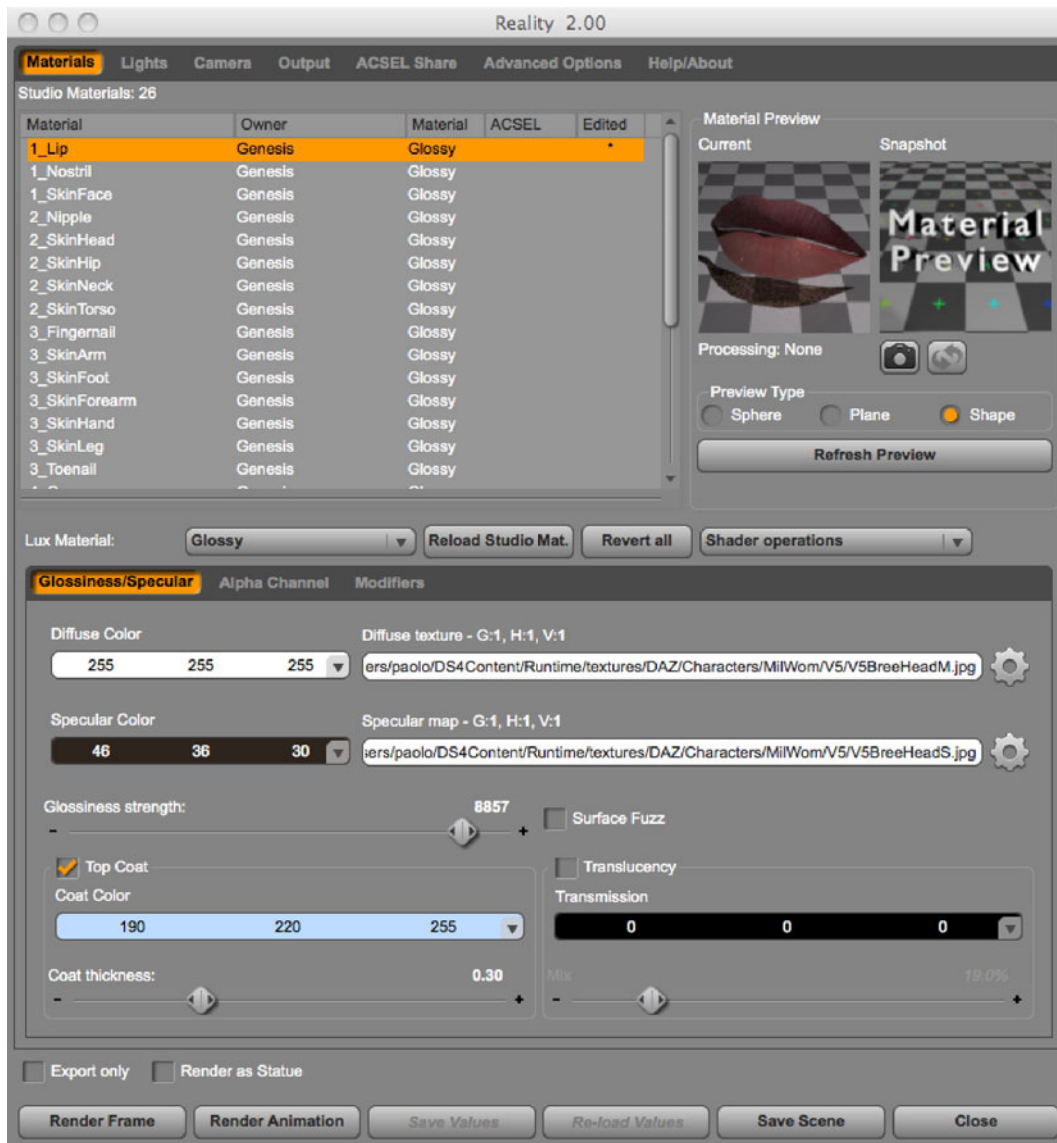
Reality’s materials

The main component of the Reality UI is the material editor. When you launch Reality the plug-in scans your Studio scene and converts all the materials in it to the closest equivalent in Lux. For example, a shiny object is converted to the Lux “Glossy” material, others are converted to “Matte” and so on. Reality recognizes automatically the skin materials for DAZ’s Generation 3, 4 and Genesis figures and has a special configuration for those. For example, the cornea is converted to a Glass

material with the Index Of Refraction (IOR) of a human cornea. Reality also uses the Coat and “Surface Fuzziness” options to create realistic-looking skin.

Editing materials

The following picture shows Reality’s Material editor after the plug-in has been invoked on the “Victoria 5” model:

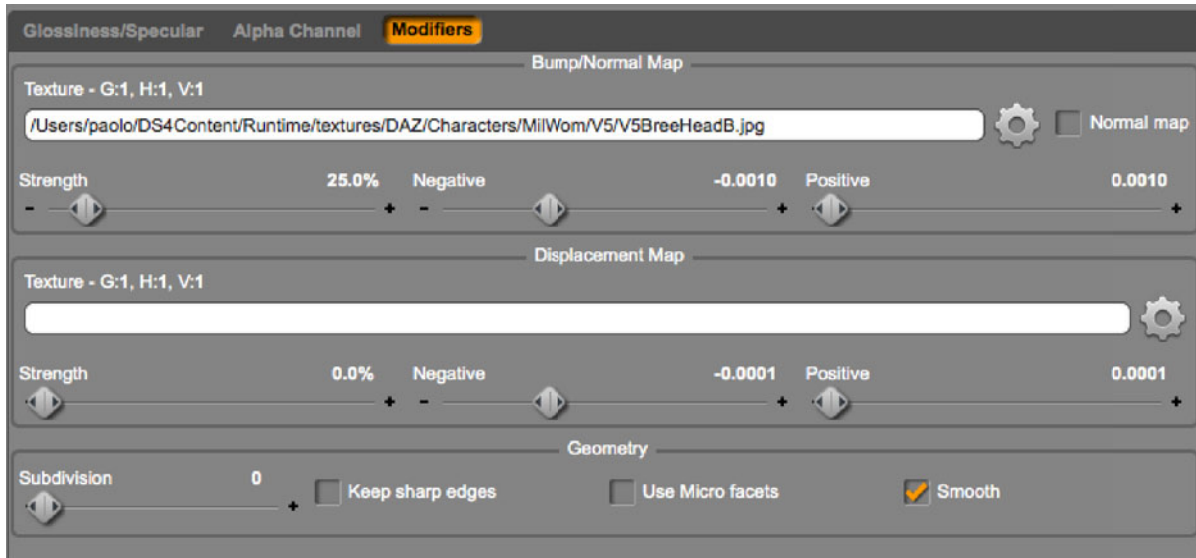


The Material list is ordered alphabetically by name, by default. You can reorder the columns to your liking and you can sort by material name, owner, the model that

holds the material, and by material type. This kind of organization makes it easy to find materials of the same kind and to operate some transformations on them.

When you select a material, the lower part of the window shows the properties of that material grouped in several tabs. For example, the Glossy material has three tabs: Glossiness/Specular, Alpha Channel, and Modifiers.

The Modifiers tab is common for most materials and contains the settings for bump/normal map, displacement map, subdivision and other geometry modification flags.



If you don't agree with Reality's selection for a material (for example, if you want to change from Glossy to Matte), you can simply select a material and select a new type from the "Lux Material" drop-down list:

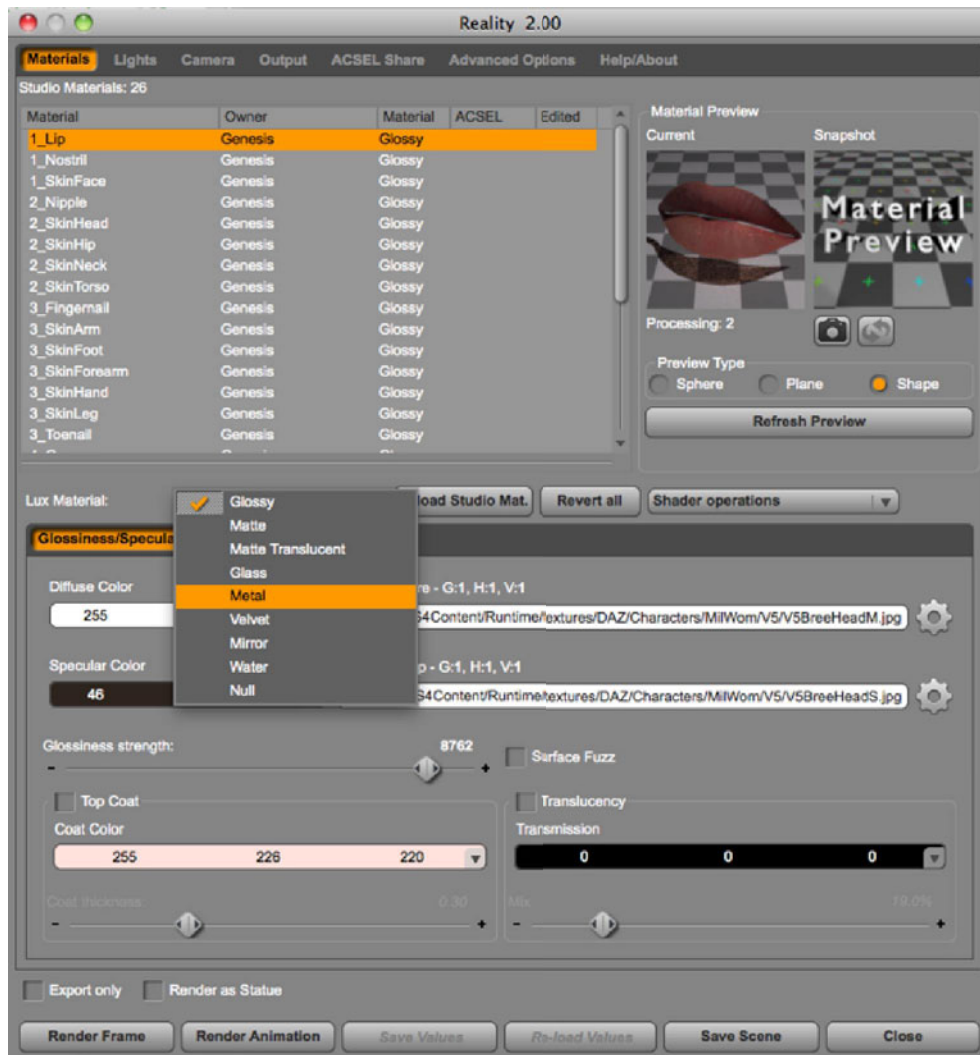
When converting from one material type to another, Reality tries, where possible, to retain the characteristics of the material. For example, moving from Glossy to Matte will keep the diffuse texture and color, among other things.

New for Reality 2.0 is the "Edited" column which shows an asterisk when the material has been edited from the original values calculated by Reality. There are also a few new features in the conversion process.

- The accuracy of it has been improved to take into account many different Studio variations.
- The support for Studio's second specular has been added. This plugs into the "Coat" option of the Glossy material.
- The interpretation of various textures and maps has also been made more flexible which leads to more accurate conversion of the Studio material into the Reality/Lux equivalent.
- In addition, Reality now takes in consideration the "Lighting model" of the Studio material. If the model is set to "Metallic" then a Metal material will be created . If the model is set to "Matte" then a Matte material will be created in Reality.
- Finally, the conversion process itself is much faster than with the previous version of Reality.

Changing multiple materials

Reality allows you to quickly change multiple materials with a single action. If you need to change the type of multiple materials at once you can simply select all the materials affected and then select a new type from the drop-down list.



All selected materials will be changed to this new type. If you edited one material and you want to apply those changes to a set of similar materials:

- Right-click on the source material you've changed.

- Select “Copy” from the pop-up menu.
- Select all the target materials.
- Right-click and select “Paste.”

All the settings from the source material will be copied to the targets except for the textures. This allows you, for example, to copy the skin settings from V4’s torso to V4’s limbs without losing the textures in the target materials.

If you also need to apply the textures, use the “Copy with texture” option instead.

Multi-edit

New for Reality 2.0 there is support for Multi-edit, the ability to edit one or more parameters for multiple selected materials. Simply select multiple materials either via Shift or Cmd/Ctrl, and then change whatever parameter you want. For example change the glossiness level or the Positive/Negative values for bump mapping and all selected materials will take that new value.

Save your Scene

Reality saves its own data inside the Studio scene, so when you save it, all the Reality materials and configuration data are saved as well. This operation is very easy thanks to Reality 2.0's new “Save Scene” button. Click on it and Studio will save the whole scene. No need to exit Reality for that.

Merging Scene

An exciting new feature of Reality 2.0 is the ability to detect and handle Studio's merging of scenes. Artists often use the File/Merge option to bring two Studio scenes together. In the past this created an issue because Reality's settings are stored in the Studio scene and the merge would cause one set of values to be overwritten with the other. No more. Reality 2.0 has a smart-merge system that detects the upcoming merge and combines the two data sets together.

Not only does this preserve long hours of work but it opens new possibilities for content creators. Because the scene holds all the parameters for Reality's lights, including the intensity, color temperature, RGB color, IES file etc, you can now use this

feature to provide creative light sets for Reality. For example you could generate a special set for Stonemason's Urban Future or Jack Tomalin's West Park. This is a great opportunity for content providers to provide commercial products that appeal to the Reality artists.

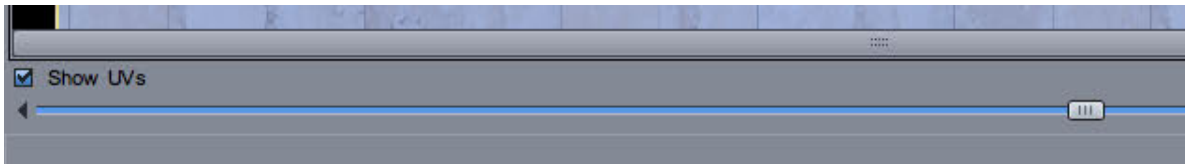
A hidden gem: The UV Map Viewer

Sometimes it's not very easy to understand what part of an object a material affects. Sometimes you would like to view the bitmap used in a texture channel. For all these times you have a solution in Reality's built-in UV Map Viewer. Wherever you have a texture in a material channel (Diffuse, Specular Map, Bump Map or others), you have a button next to the file name labelled "View/UV".

Press this button and the texture will be shown on screen with the UV Map of the selected material superimposed and partially transparent.

Here is a screenshot of the UV Map Viewer in action:

The UV Map Viewer allows you to see exactly which part of the geometry uses which part of the texture for any given material. You can change the opacity of the mesh grid from barely visible to completely opaque.



The "Show UV" checkbox allows you to toggle visibility of the grid on and off.

The UV Map Viewer is a great tool for regular users and material designers alike.

Lux materials are simpler

Don't be alarmed that the rather long list of properties that you see in the Studio Surface window is not showing in the Reality Material Editor. Lux materials are defined by their characteristics, like being Velvet, Metal or Glass. Those characteristics in many cases include multiple parameters in a traditional renderer that may not be necessary for an unbiased renderer. For example, a Glass material needs opacity control in Studio, while that is not necessary in Reality since Glass is by definition transparent. The alpha channel for Glass in Reality is only used to create irregular edges or holes in the material. So, the list of properties in Reality is in many cases different and generally simpler than what you see with a biased renderer. It's also easier to read because it maps properties of real-life materials.

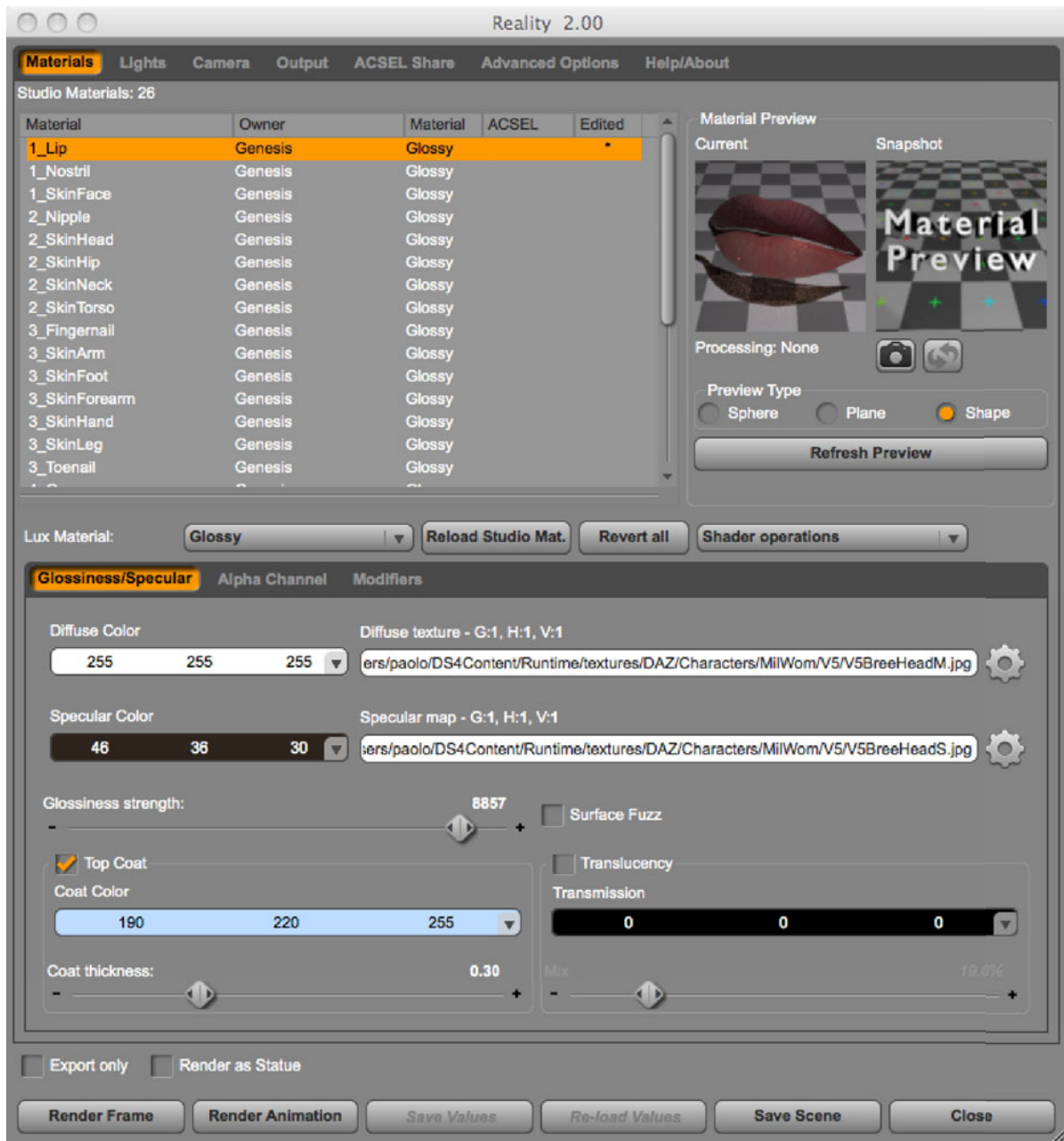
See before you render: The Material Preview

When tweaking and adjusting a material it's nice to be able to see how it will look without having to wait for the whole scene to render. For this task there is the Material Preview.

Reality works in the background and renders a preview of the material whenever you make a change -- automatically and without waiting for the preview to finish. This is a quick approximation of your material mapped either to a sphere or a plane. The preview is close to the final result but it's not completely accurate, for reasons of speed. It will give you a good understanding of the material's colors, textures, level of glossiness, and so forth. Nevertheless we constantly improve it and for Reality 2.0 the glass preview is closer to the final result than before.

When making materials for such important areas of human models like lips, face and eyes, you have a third option for the preview: Shape. Shape gives you a stylized pair of lips, a simplified face and a stylized eye to help you see more precisely the effect of your material. These shapes are all mapped to reflect the real model so that textures project as expected. Reality 2.0 introduces the Face shape for the first time.

The following figure shows the Shape preview in action:

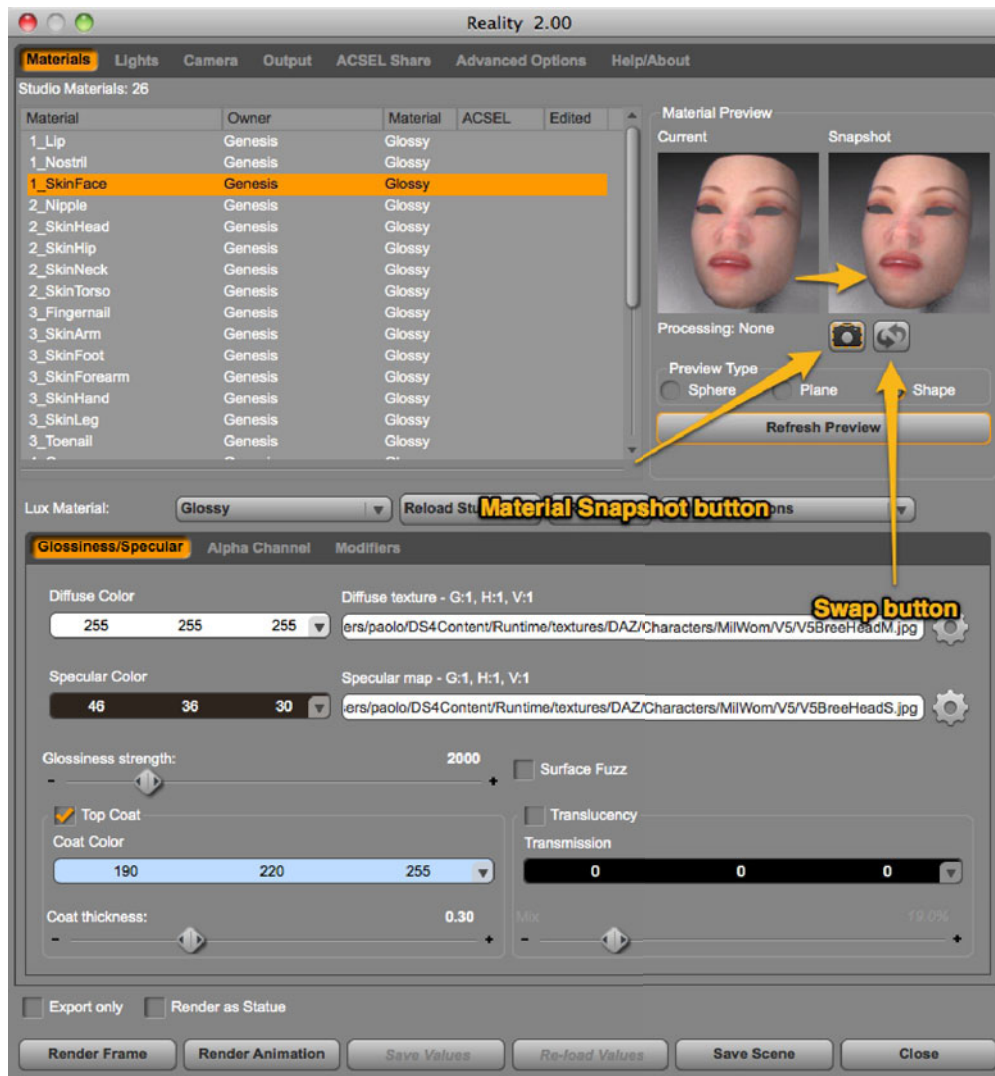


Another great feature of the Material Preview, new for Reality 2.0, is that the previews are persistent: if you select another material and then come back, the preview will be shown immediately without re-rendering it. The number of previews cached is limited. By default Reality will cache five previews. You can change this number,

setting it up to twenty, in the Advanced tab. Also new for Reality 2.0 is the automatic detection and switch to the Shape preview for certain materials. When previewing the lips or face materials of DAZ figures Reality will automatically select the “Shape” preview. It almost reads your mind! Scary stuff.

The material snapshot

When editing a material you might get to a point where it looks good, almost perfect. Maybe you could improve the material even further but you don't want top risk upsetting something that is already good. What if you can't go back and you lose your carefully crafted settings? What a dilemma. Reality 2 gives you the way to solve this situation. Simply click on the “Snapshot” button and the material is copied to the snapshot. Now you can change your material as much as you want. If you want to revert back to the snapshot simply press the swap button. This swaps the current material with the snapshot. You can, of course swap back. Please note, if you switch to another material the snapshot is erased.



Resetting materials to the original state

Sometimes things don't go in the direction that we want. After tweaking a material for a few minutes you might find yourself in a situation where you would rather start again from the beginning, from the original material that was imported by Reality from Studio. For those times your friend is the "Reload Studio Mat." button. Click it and the current material will revert to Reality's original version. You can also select multiple materials and perform this operation on all of them at once. The "Revert all" button reverts all Reality parameters, including lights, camera and output

options to the original state. This operation will cause Reality to close and restart again automatically.

Reality material reference

Reality provides access to most of LuxRender materials. Reality materials are designed to provide all the advantages of using Lux's physics-based materials but without the complexity generally found when using renderers of this kind. To make things easier to approach, and to avoid clutter in the UI, all texture entries have a "gear" menu that give you access to the common properties for all textures. Please note that for version 2.0 almost all material panels have been substantially re-arranged for better layout, more consistent presentation and better use of the screen space.



When clicking on the gear button a menu will be shown providing the following options:

- **Change.** This option allows you to select a different texture stored on disk. Reality 2.0 remembers the last folder visited and will run the File/Open dialog box position in that folder instead of some random location on your disk.

- **View Texture+UV Map.** This option gives you access to the UV Map viewer.
- **Reload original Texture.** This option reloads the original texture used in the Studio material. Useful to revert the texture without changing the other material properties.
- **More options.** This choice give you access to the tiling and gain parameters. Unlike Studio, where the texture tiling is set at the material level and applied to all the texture used by the material, the tiling parameters in Reality are set at the texture level. So, for example, you can set the bump-map tiling at a different value than the one used for the diffuse channel. The **Gain** parameter can be used to brighten or darken the texture.

What follows is a list of the materials and their properties:

Glossy

One of the most widely used materials, it can be used for things like plastic, glossy paper, semi-glossy stone, leaves, and human skin. In fact Reality uses this material for all the skin materials recognized in DAZ's human figures. Glossy has four categories of values: Glossiness/Specular, Alpha Channel, and Modifiers. The parameters are:

Glossiness/Specular

- **Diffuse color.** The same color that is found in Studio. The color is mixed (multiplied) with the diffuse texture, if present.
- **Diffuse Texture.** This is taken from Studio's Diffuse channel, if present, and can be changed in Reality. This gives you the ability to override what Studio provides. If you change this texture the original Surface settings in Studio will not change. All textures in Reality include control for the gain (brightness) and Horizontal and Vertical tiling. Tiling can be specified for up to 100 repetitions and is applied to each texture individually.

Reality supports the format used by Studio's **Multi-Layer Image Editor**. These files, which have the extension `.dsi`, can define images that are the result of multiple layers blended together with several mixing parameters. See the documentation for the MLIE in Studio. Whenever you can use a texture in Reality you can use a `.dsi` file.

- **Gain.** Set at 1 by default. It's a value that can be used to change how bright a texture is. For example, you can make the sclera (white of the eye) of Victoria whiter by increasing the gain for that texture.
- **Specular color.** The color used to show a reflected object on the surface of the glossy object. A color has two main components: hue and brightness. The hue is determining the "tint" of the light reflected on the surface of the object. The brightness determines how much light is reflected by the surface. Brighter colors will generate a very high amount of light reflected. If you want to make a material with a certain tint for the reflection but want it to be less shiny, make the color darker. Many times the best results are obtained by using a shade of gray. Generally this color works best when is set at around 1/4 brightness of the scale, but there are exceptions, of course. Reality calculates this color from the specular value found in Studio and by applying a correction for lowering the brightness, since the Studio version is much too bright for Lux. The Specular Color and Glossiness strength work together to create the final glossiness effect.
- **Glossiness.** The amount of glossiness of the material. Below 5,000 the material is fairly matte-looking. At 10,000 it's almost mirror-like.
- **Specular map.** Taken from Studio, if present. A specular map is a bitmap that determines the amount of glossiness of different areas of the material. When the map is present, the glossiness slider determines the maximum glossiness of pure white in the specular map.
- **Coat.** This option allows you to control the color and thickness of the Glossy coat. Think of the Glossy material as a diffuse layer with a clear coat of "varnish" on top. The Coat parameter allows you to control that coat and will result in a "glaze" on top of the material. The color parameter controls the translucent color of the coat. The thickness parameter, by default at 0.3, determines how visible the coat is. Higher values will make the coat more prominent.
- **Translucency.** Some materials need to have their glossiness scattered and softened a bit. This property does exactly that. It mixes, at the amount determined by the **Mix** parameter, a Matte Translucent material with the Glossy one. The color control in this section determines the color of the translucent part.

Important: LuxRender 0.8 introduces a new material called “**Glossy Translucent**”. Reality supports this material directly. If you set the Mix to 0, then Reality will select the “Glossy Translucent” material. If you set the mix to any value higher than zero then the mixing of “Glossy” and “Matte Translucent” will be used. The two approaches lead to different results and this solution allows you to select the more appropriate material. Setting up a test scene with two objects using the two different materials side by side is probably the best way of seeing the difference. “Glossy Translucent” (Mix set to zero) is very good for leaves, balloons, soap, and hair.

- **Surface Fuzz.** This property is used to access the new “Asperity” option of Lux 0.7.1 and 0.8. It adds a surface scattering of the glossiness. This effect works best with high values of glossiness and can be used to create velvet-like materials. It’s extremely effective for human skin when combined with a specular map.

Alpha Channel

- **Strength.** How strong the effect of the alpha map on the texture is. Usually the default of 1 is what you want. Making this number smaller will reduce the effect of the alpha map. If you don’t use an alpha map the slider will control the opacity level of the material. The value is taken directly from Studio but it can be changed to your liking.
- **Texture.** A bitmap that defines the alpha map. White is solid and black is transparent. This value is taken from Studio if present.
- **Gain.** This control can be used to make the specular map brighter.

Modifiers

This tab is named in this way because the values in here have the potential to modify the geometry of the model.

- **Bump Map.** You can load a bump map or a normal map (new for Reality 2.0) in this part of the UI. Reality will take the bump map from Studio, if one is used in the material’s shader. If you enter a normal map you will

have to notify Reality about this by enabling the Normal Map checkbox:



Reality recognizes several standard suffixes for the file name. If the file ends with “NRM”, “normal” or “nmap” (this is case insensitive) then the flag will be set automatically. When a normal map is used the Strength, Negative and Positive values are disabled. Normal maps provide also a much more realistic result and can be an excellent replacement for displacement maps resulting in renders that take a fraction of the time and the memory to finish, compared to displacement maps. There are several programs, free or inexpensive, that can generate normal maps from the diffuse or bump map. It's definitely worth spending some time getting familiar with this great technique.

- ▶ **Strength.** The strength of the bump map.
 - ▶ **Gain.** The gain of the image used for the bump map. Some Studio/Poser models use the trick of adding the diffuse map in place of the bump map. While this “technique” does not produce good results, you can increase the quality of your render by setting the gain to anything between 3 and 5, depending on the texture.
 - ▶ **Negative.** The maximum value for the negative bump.
 - ▶ **Positive.** The maximum value for positive bump.
- **Displacement map.** The parameters are similar to the ones used for bump mapping, and can be used in conjunction with those. Displacement mapping creates a real alteration of the rendered geometry. It depends on the subdivision level, which has to be increased to 1 or more. Higher values of subdivision produce finer detail in the resulting geometry but require a much larger amount of memory. This could be an issue with computers running with a limited amount of memory, like the ones using a 32 bit version of XP. Keep in mind that each level of subdivision, when not using

the “microfaceted” option, **quadruples** the amount of polygons in the mesh (for quadrilateral polygons). The microfaceted option activates a higher-detail type of displacement. See below.

Please note: in Lux/Reality a displacement map has a Negative and Positive value. The negative value indicates how recessed the geometry should become when the lowest value in the map, pure black, is found. The positive value indicates how raised the surface should be when the highest value in the map, pure white, is found. It derives that to indicate no displacement, the map should use a mid-grey values, 127,127,127. This is the same logic used by DAZ Studio so maps designed to work in Studio should work very well with Reality and Lux. If your displacement map has been made for Poser be aware that Poser assumes that black is no displacement. In that case you should set the Negative value to 0.

- **Subdivision.** Reality/Lux can subdivide the material to make it appear smoother. The Reality subdivision option works well for water planes and other single-material objects.
- **Keep sharp edges.** This option determines if the subdivision will “round” the edges of the subdivided material or not. This only makes sense when using open figures like floor planes. For human and organic figures this checkbox should be unchecked.
- **Use Micro facets.** New in Lux 0.8 and firstly used by Reality 1.25, this option allows you to displace a geometric surface with very high resolution, even when the original geometry is coarse. Enabling this option raises the maximum number of subdivisions from 8 to 500. Microfacet displacement is closer to Studio’s built-in displacement with the exception that the facets are flat shaded. This means that the displacement is in fact faceted and not smooth. Depending on the resolution of the original geometry this might not be an issue. Microfacet displacement is ideal for simulating brick, stone surfaces, metal engraving and so on. As a general guideline, if you need to displace organic shapes, like muscles and veins, do not use the microfacet displacement. If you need to displace walls, floors, weapons etc. then try the microfacet variant. If your base geometry is fairly detailed you can use low values of subdivision. If the geometry is very coarse, like when a floor is defined as a single polygon, then very high values of subdivision, 100, 300 or higher might be used. Beware that

higher values will require more memory and will take longer to render. Be conservative in increasing the subdivision level.

- **Smooth.** New in Reality 2.0. This options allows you to change how the model is rendered: smooth or faceted (aka flat shading). Flat shading is good when you want to create something like the effect of diamond or a Swarovski crystal where the faces of the model must remain flat and sharp. This updates the Studio preview.

Matte

As the name implies, the Matte material has the purpose of helping define surfaces that have no specularly. Only the Diffuse tab is unique to the Matte material. In it you can find a diffuse color and texture. New for Reality 2.0 you can adjust the roughness of this material.

Matte Translucent

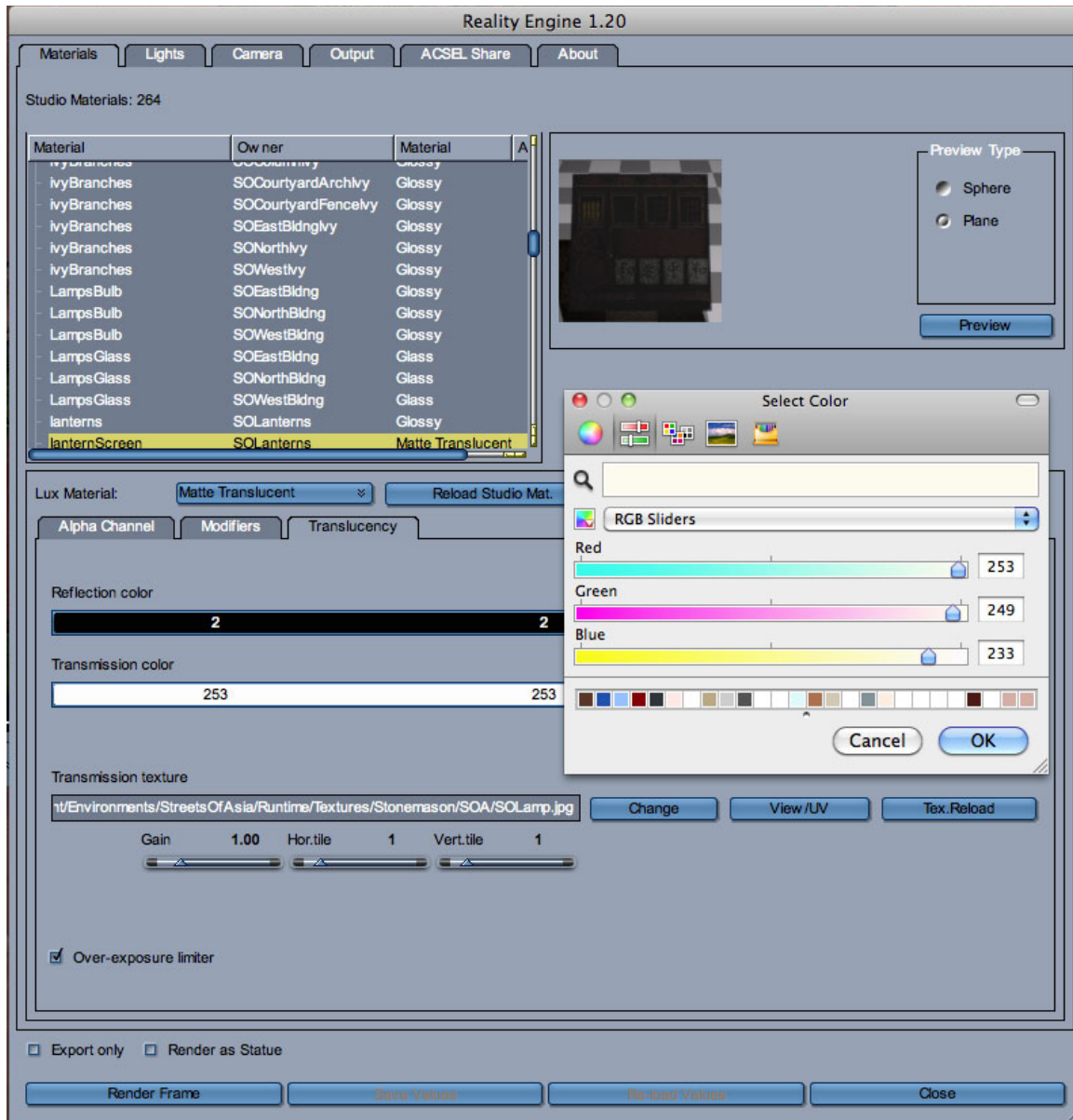
This is one of the trickiest materials to use so be sure to start using it after you have gained some experience with the others. Matte translucent is ideal for things like lamp shaders, light diffusers or shoji doors.

Here is an example using Stonemason's "Streets Of Asia:"



The screen for the lamp has been converted to be Matte Translucent, which means that it lets light filter through.

These are the Matte Translucent settings in Reality that were set for this scene:



The color picker shows the color for the Transmission parameter, which is replacing the previous value of 253,253,253. If you use Windows the color picker will appear different. Reality transferred the texture for the lamp from the Studio

shader, which is then mixed with the Reflection and Transmission colors. To achieve the effect in the rendered image you can place a cube primitive and turn it into a Mesh Light. You can also use a point light. The soft shadows are caused by the diffusion given by the lantern screen, which makes the light source a large one.

The “Over-exposure limiter” flag helps in avoiding having the texture for the translucent material being “blown out” when a light is placed behind the translucent object. Also note that you can use the Alpha channel to control the opacity of the translucent material to make it partially transparent and let it pass more light through.

Glass

Glass is a simulation of real glass and it can create near-photographic results. Glass is mainly defined by the Reflection and Transmission parameters and the Index Of Refraction (IOR). Reality includes dozens of IOR presets measured from real materials. For example, if you want to render an emerald you can find the exact IOR for it, 1.56, under the Gemstones preset group. The IOR does not change the color of the material, which needs to be modified manually. Glass has three tabs: Glass, which defines the glass characteristics, and the usual Alpha Channel and Modifiers tabs.

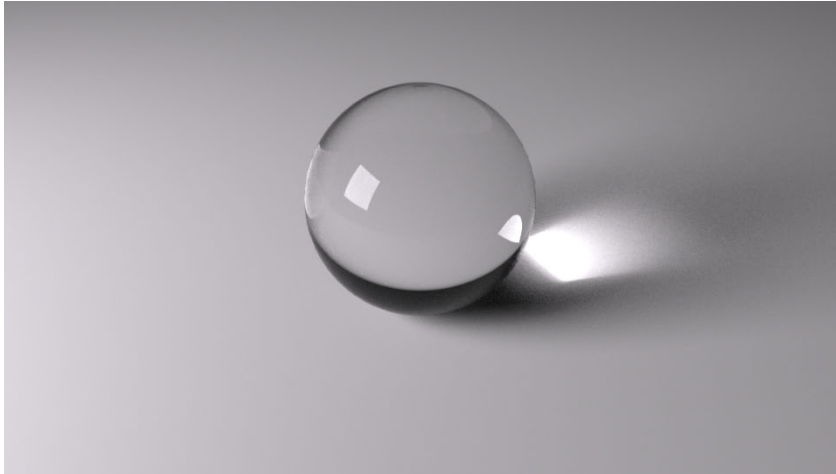
The parameters in the Glass tab are:

- **Reflection.** The color of the reflections on the surface of the material. The specular color from Studio is imported in this channel but it’s usually best to change it to an off-white (250,250,250 for example). The higher this number is the more reflective the glass is. If you want to create the effects of a glass pane that has been coated with an anti-glare treatment, use a low Reflection value, like 42,42,42.
- **Reflection map.** New for Reality 2.0 you can have a texture to mix with the reflection color.
- **Transmission.** The color that the glass will have and consequently the amount of light that will pass through the material. Studio’s diffuse color is applied to this channel. Please note that this is different from Reality 1.01, where the specular color was used instead. If you want to create a

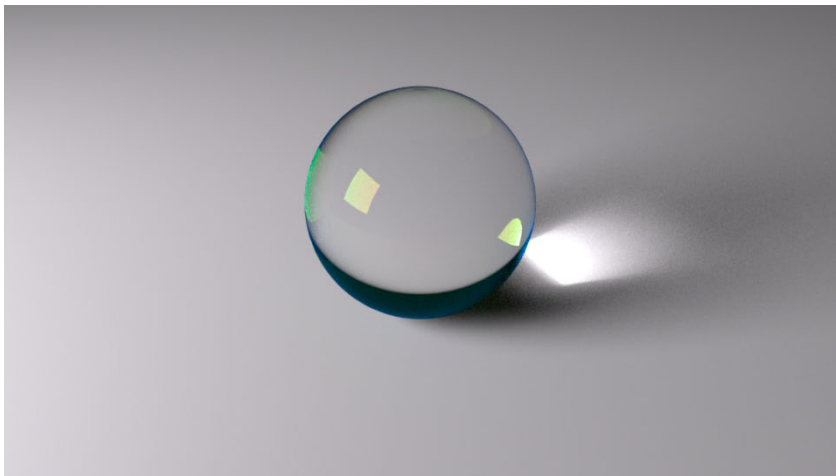
green glass bottle this is where the green will be configured. This is the color that the light will pick up when traversing the glass. For neutral glass, like windows or car windshields, you should set it to white. For Lara Croft's sunglasses a nice shade of orange will do.

- **Transmission map.** In addition to the transmission color you can also mix in a transmission texture. This allows you to create a plethora of stunning effects, including stained glass and light gels. If the texture washes out because of the light going through it, mix in a dark gray for the transmission color, like 20,20,20.
- **Architectural.** Glass is computationally expensive so if the object is small and at a distance, or if you don't need absolute accuracy for it, enabling this parameter will speed things up while maintaining the basic characteristics of glass like being reflective and transparent. The refraction deformation will be less visible.
- **Hyper-Realistic.** Lux 0.7 introduced volumetric glass, a material that transforms light based on its volume. What this means is that the thickness of the glass in the model affects the final result of how it's painted on the image. Also, light is affected when passing through the glass and it creates colored shadows. Hyper-Realistic is basically the opposite of "Architectural" and that's why the two options are mutually exclusive.
- **Frosted.** This flag creates Frosted Glass, or rough glass. When enabled, the Roughness slider will be activated. Higher values of roughness make the glass appear more and more as if it was "sand blasted."
- **Thin film thickness.** This value allows you to define a glass material with a film coat that is expressed in nanometers. Optical instruments, like a camera lens, are often coated with a thin film to reduce the glare caused by direct bright light like the sun. This coat can show as an iridescent "film" when seen at a certain angle. A good starting value for this parameter can be 200nm.

The following figures show what a glass sphere looks like without and with thin-film coating.



A glass sphere without thin-film coating



Glass sphere with thin film coating. Thickness at 200, IOR at 1.5

- **Thin film IOR.** This is the Index Of Refraction of the thin film. Different IORs will create different colors in the film. For this IOR you can try 1.0 and move from there to see the shift of iridescence in the coat.

When converting glass from a Studio material you need to keep an eye on a couple of issues. Studio simulates glass via a transparency map or opacity value. Both these options should be avoided in Reality. If you convert a material to Glass check immediately if it has an alpha map. If so, erase the file name or the resulting material will be too “thin” and transparent. Also check that the Opacity value should be 1.0. Also check the texture in the Transmission channel of the

Glass. Some textures in Studio are designed to create a certain coloration in the simulated material. Unless the transmission texture has a real image, you are better off by removing it and using the transmission color instead. If the texture has a sort of painting, like the images found on stained glass, then it will be OK to leave it in place.

Metal

This material defines metallic objects that can have anisotropic reflections (see the Polish parameter, below, for a definition). There are four presets: Gold, Silver, Copper and Aluminum. These presets can be configured to have variable levels of shininess and they can be used to create other types of metal. For example, very shiny silver can be used in place of stainless steel. Besides the usual “Modifiers” tab, Metal has also an Alpha Channel. New for Reality 2.0 is the use of Lux’s “Metal2” material. This provides an improved, more realistic type of metal that is also much simpler to customize. In fact, now you can select “Custom metal” and use a simple color channel to create materials like anodized aluminum in seconds. Also new for Reality 2.0 is the ability to assign a texture for the metal. The properties for this material are defined in the Metal tab.

Metal

- **Metal Preset.** Drop-down list of available presets. These change the color and the light response of the material. New for Reality 2.0, eight new presets have been added: Chrome, Cobalt, Lead, Mercury, Nickel, Platinum, Titanium, Zinc.
These are in addition to the pre-existing Gold, Silver, Aluminum and Copper.
- **Polish.** How shiny the metal is. Higher values bring more shine. The Horizontal and Vertical sliders are linked together by default to give uniform reflection. When the “Synchronized” option is deselected the two sliders can move independently, allowing you to create a polish that has different values for the vertical and horizontal directions, called “anisotropic reflection.”
- **Custom Metal.** This option allows you to define your own metallic materials by controlling the color and texture for the surface . When using this

option the presets are not valid anymore. The Polish value is used in the usual way.

Mirror

This material simply defines a mirror surface. The parameters include a color for the reflectivity. The brighter the color, the brighter the mirror. For example, if you define the reflectivity to be 50,50,50 (a dark shade of gray), the mirror will act as if the glass was tinted dark. Generally shades of gray work best for this parameter. The other parameters define the optional thin film (zero thickness will disable it), and the IOR for the film. These work exactly like their Glass counterparts.

Velvet

This is a great material for clothes and other textile objects like drapes. In addition to the usual diffuse color and texture, similar to Matte, velvet includes a “Thickness” parameter. Higher values accentuate the velvet effect.

Water

This material is one of the most exciting features of Reality. Water allows you to create very realistic-looking water like you see in high-end 3D programs. The parameters for water definition are:

- **Water type.** There are two types: “Clear Water” and “Sea/Pool Water.” The difference between the two types is that the “Sea/Pool water” works with scientifically measured data for the light absorption at different depths. Clear water instead has a Clarity parameter that expresses in meters how far in depth the water is clear. Clear water can be used for anything, from ponds to Mai Tais. When Clear Water is select the Tint and Clarity parameters are enabled, otherwise they are disabled.
- **Tint.** The color for the water. Use it to create orange juice, wine, beer or any other liquid. Please render responsibly.
- **Clarity.** The measure of how far you can see into the water. Expressed in meters.

The easiest way to add water to your scene is to use the Water Plane prop provided with Reality. Just drop it in your scene, subdivide it a couple of times right inside Reality and render. Alternatively you can import any OBJ into Studio

and convert it to water. If the OBJ already has a single material called “Reality-Water” then Reality will do the assignment of the water material automatically.

Water in Reality 1.2 gained an alpha channel to help define irregular shapes for the water prop by using a transparency map. Please note that the same issues for alpha maps explained in the Glass material apply here.

Null

Null is perfect for when you need to hide something in the scene. Please note that if you make a model or a part of a model invisible in the Studio scene tab, that object will not be exported. Using the Null material is different, in that it allows you to selectively hide materials while keeping the geometry.

DAZ Studio's SubD and Reality

Up to version 3 DAZ Studio's SubD (Subdivision) feature was almost completely undocumented and unsupported by the plug-in system. This changed with Studio 4 and now you can use the SubD option and have the model rendered via Reality without problems. We'd like to add a word of caution about this. SubD in Studio works at the model level. This means that if you apply SubD to V5 you will subdivide the whole model. Each subdivision level quadruples the number of polygons. V5 starts at 18,872 polygons with one level of subdivision for a total of 75,488 polygons. This is the default, unless you move the subdivision level to zero, in which case it goes down to the original 18,872. If you apply a second level of subdivision the whole geometry ramps up to a staggering 301,952 polygons. This is for one naked V5 and nothing else in the scene.

Subdivision is rarely needed at high levels for the entire figure. The parts that are close to the camera usually need to have more detail but other body parts less prominent might be just wasting memory and slow down your renders. Remember that all those 301,952 polygons need to be processed by Lux. So, the best approach is to be very conservative with SubD. If you need to smooth out a certain element, like the face, that is closer to the camera, instead use the Subdivision slider in Reality on a per-material, instead of a per-model, approach. For example apply Subdivision in

Reality for the 1_SkinFace material. You will get excellent results in shorter time and with a smaller memory footprint.

Either way, Reality will render your scene via Lux regardless of the approach that you use.

ACSEL

Converting a shader from one program to another, from Studio to Lux for example, is a tricky job. Because 3Delight (the built-in renderer of Studio) and LuxRender use completely different philosophies, it can be difficult to automatically create a Lux material that corresponds exactly to the Studio version.

3Delight also doesn't have the concept of physically-accurate materials so it's your responsibility to configure a material to take advantage of what Lux has to offer. For example, you might want to convert Studio's simulated glass to Reality's Glass material, which in turn uses Lux to render glass that is physically accurate, including the absorption of light based on the thickness of the object. This kind of decision cannot be taken by a program because software doesn't have eyes, common sense or artistic taste.

In other words, the human touch is irreplaceable.

Having said so, it doesn't mean that we have to "trapped" into an endless cycle of changing and adjusting materials. We can find ways of optimize this activity. For example, if you decided that a certain material of a certain model should be red Velvet instead of brown Matte (as suggested by Studio), then you don't want to repeat the same conversion operation every time that you use that model in a Reality scene. There has to be a way of telling Reality: "This is how I want this material to be!"

There is and it's called ACSEL: the Automatic Custom ShadEr Loader (pronounced AXEL).

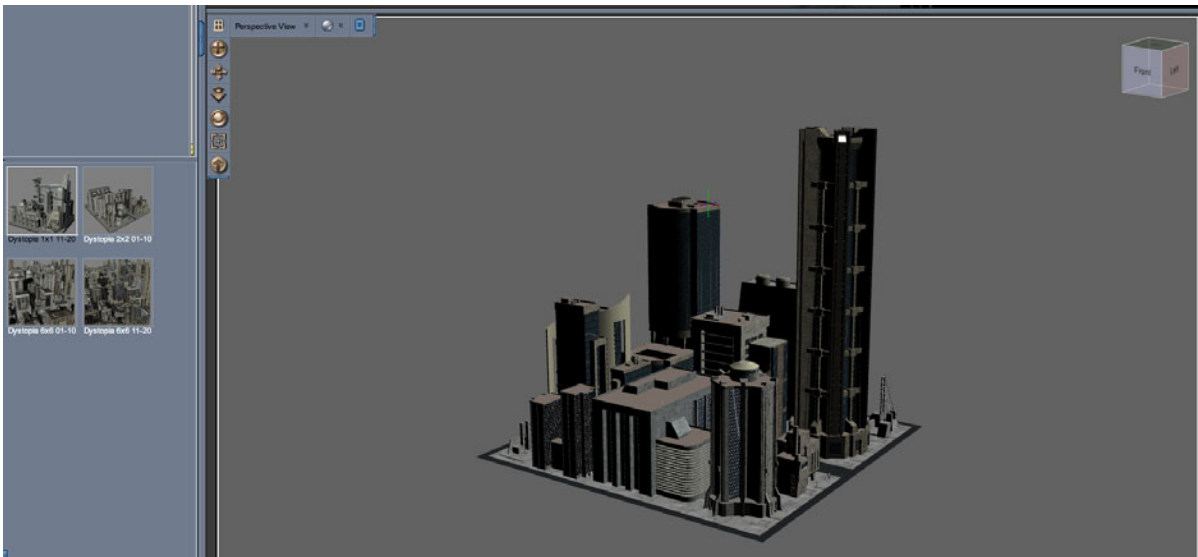
Look at the Reality window when the "Materials" tab is active. In the middle, to the right, of it there is a drop-down menu that says "Shader operations":

Click on it and you will find five options. We are going to see how they work with a few examples.

Quick ACSEL tutorials

For this exercise we will use the free “Dystopia blocks” available at DAZ in the Free models section. You can use a different model if you prefer. The idea is to see how to change a series of materials in Reality and store those settings as the default by using ACSEL.

1. Start by loading Dystopia block 11-20:



2. Go to the menu and select “Tools/Surface Selection”.
3. In the viewport find the tall cylindrical building with the sides made out of glass. Click on the glass material. Studio should report it being called “window_c”.
4. Call Reality, and window_c should be automatically selected in the material editor.
5. The material type selected by Reality is “Glossy.” Change it to “Glass.”

6. Click on the “Glass” tab of the material and notice that the Diffuse texture that is used by the Studio material is now in the Transmission texture.
7. Click on the “Architectural” checkbox.
8. Lower the transmission color to 200,200,200, just to change it from the default.
9. Add a Sun light to the scene and orient it as you prefer, frame the shot in the camera and start a render. Be sure to use the “Scale city up to 12000” pose before rendering. Since Lux is physically-based, accurate scale is important in determining the effect of the Sun light.
10. Notice the difference in glass effect with the other, unchanged, buildings which have window_a or window_b materials. The real Glass material reflects the surrounding buildings and the sky.
11. Close Lux, go back to Reality and select the window_c material again.
12. Click on the “Shader operations” drop-down list and select “Save.” This action saves the selected shaders using the ACSEL format and creates a shader file on your disk.
13. Reality will show you the usual “File save” dialog box. In there you need to select a folder where to save the shader file. You will not supply a file name since that is set by Reality based on the name of the material. Go to your Desktop folder, create a new subfolder called Dystopia, select that folder and click on the “Choose” button. Your shader is saved.
14. Create a new Studio scene. You don’t need to save what you have done so far.
15. Add the same Dystopia city block as before.
16. Call Reality and select the window_c material. It’s back to “Glossy”
17. With that material selected click on the “Shader operation” list, and select “Load shaders.”
18. Reality shows you a dialog box titled “Select the base directory for the shader files.”

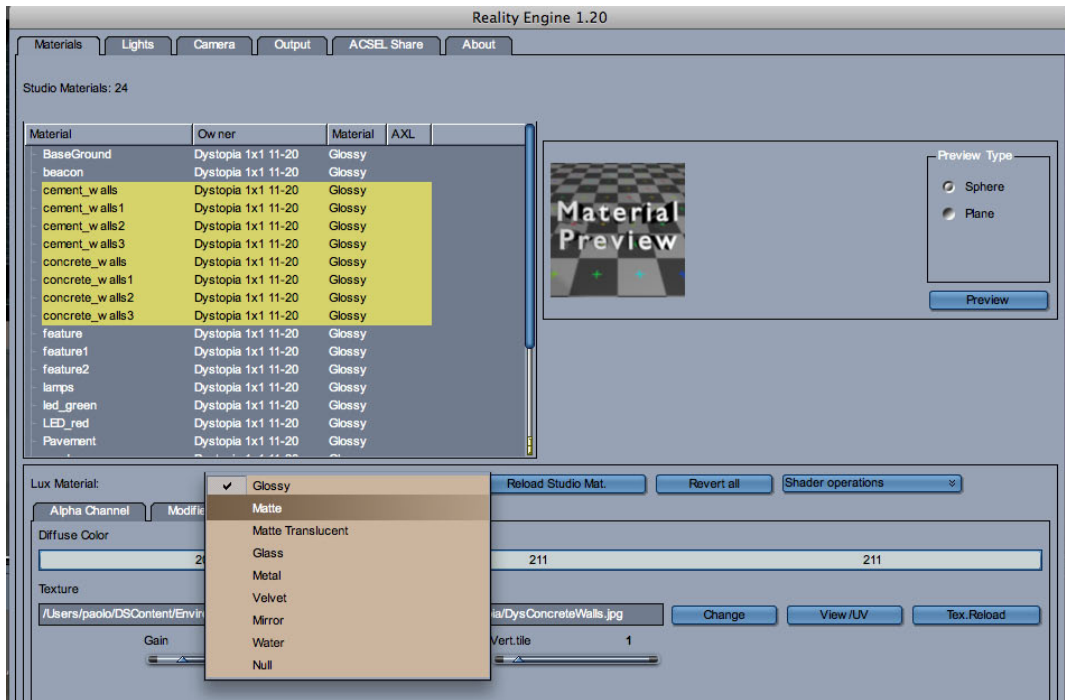
19. Navigate to your Desktop folder, select the Dystopia folder and click on “Choose.”
20. The window_c material should now be Glass, with the architectural flag on and the transmission color set to 200,200,200. Victory!
21. All these steps can be performed also on multiple shaders by selecting several materials in the list and then using the same “Save for ACSEL” option in the drop-down list. In this case Reality will save the whole set of shaders in one single operation. Each shader is saved into a separate file.

While all this is nice and convenient and can be very helpful during the fine-tuning of custom shaders, what we are looking for is a system to define the material for window_c and have Reality use that definition automatically every time we use the “Dystopia 1x1 11-20” prop. To make this happen simply save the shader using the “Save for ACSEL” option. This command will not ask you for a location (that is pre-defined by Reality), but it will ask you to choose if this is the default shader or a customization. A default shader is what is used all the time for a given model. If you would like a model to use these shaders every time, then choose “Default.”

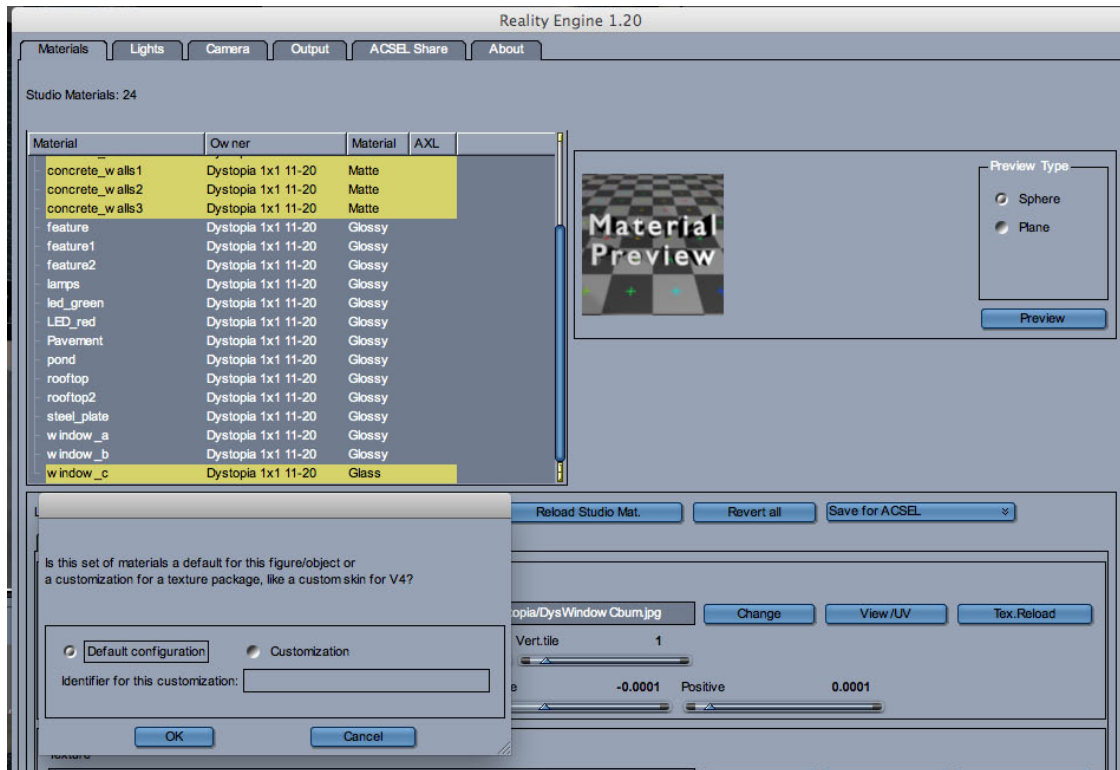
If a model uses multiple texture packs, like in the case of Victoria 4, where the skin can be changed often, then it’s better to select “Customization.” For example, if you are adjusting the shaders when using the Elite Lana or Reby Sky packages, you would need to select “Customization.” Otherwise, the shader configurations will be used every time you use Victoria, regardless of the skin selected. When selecting a customization you will be asked to provide a name for the shader set. For example “Reby_Sky.”

Let’s see all these ideas in action.

1. Select all the materials from “cement_walls” to “concrete_walls3” as shown below.



2. Change the materials to Matte.
3. Re-select all the materials converted and also select window_c.
4. With the materials still selected click on "Shader operations" and select "Save for ACSEL".



5. Leave the “Default configuration” option active and press OK.
6. You will see a progress bar on the screen updating as Reality saves the ACSEL shaders.
7. Close the Reality window, erase the scene again and re-add the Dystopia block used before.
8. Call Reality again.
9. The materials that have been changed are now showing your modifications. ACSEL loaded the custom shaders automatically. Fantastico!

You will also notice that the materials modified have a “Y” in the “ACSEL” column. That mark is simply to confirm that the material definition has been loaded from the ACSEL database.

When you use a named set, the name of the set will appear in the ACSEL column. For example, if you have installed the Elite Lana ACSEL shaders for V4, when you use that “skin” the ACSEL column will show “Elite_Lana”. This is a new feature of Reality 2.0 that allows you to know exactly what shader is used.

Save with textures

An additional powerful ACSEL option is the “Save with textures” variant. This command doesn’t save the texture files with the shader but it allows you to remap where textures are used. There are a couple of reasons for this. Sometimes materials are set up in Studio in a way that doesn’t translate appropriately to Reality. For example, glass in Studio is generally defined as a standard material with a transparency (alpha) map to simulate the “see through” property of glass. This system doesn’t account for other more subtle properties of glass, like the refraction or the way it tints light exiting the glass object. When you convert that material to Glass in Reality, the alpha map is taken as well. This might be the right thing or not, but Reality has no way of knowing. In general you should not use an alpha map with Glass except for defining irregular shapes or holes in the object. If the alpha map is used just to create a “see through” effect then it will be redundant with Glass and should be removed.

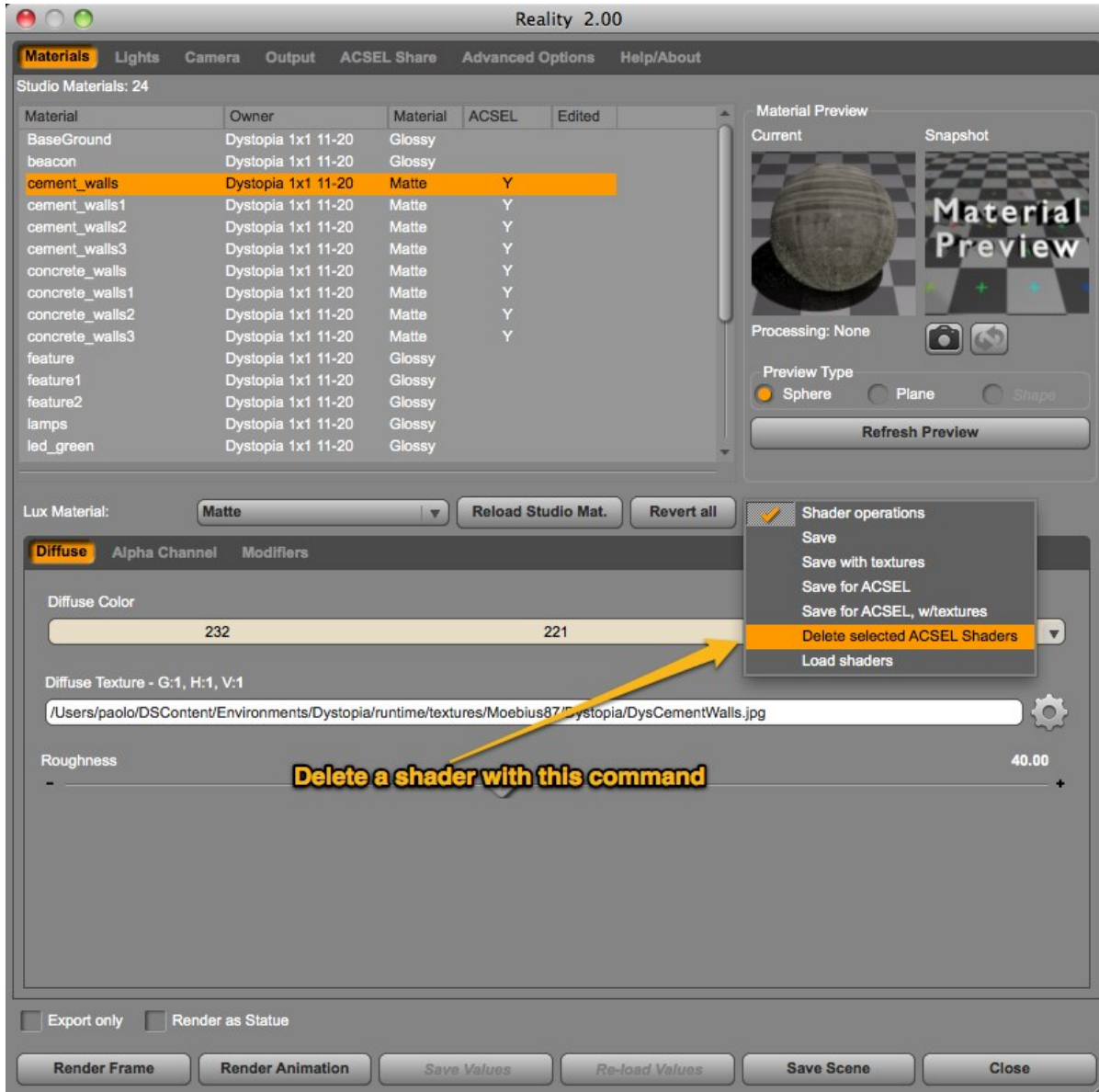
In these kind of cases you want to save your ACSEL shader with texture because that command will create a shader that redefines the texture usage as well, including the possible removal of a texture from one channel.

In a similar way there are situations in which materials don’t get transferred correctly and the textures are not placed in the right channels. This happens often when Studio shaders have been converted automatically (via software) from Poser shaders, instead of being hand made. Some software converters place the textures in non-standard channels and that makes it impossible to recognize their usage. In those cases you might need to redefine the correct placement and then save the shader using the “Save with textures” option.

Deleting a shader

It might happen that, from time to time, you want to delete a shader that you made or that you have installed with ACSEL Share (see below). New for Reality 2.0 is the

delete command in the Shader Operations menu. Simply select the shaders that you want to delete and click on the delete command. They will be removed from the ACSEL database and will never be used again:



ACSEL Database maintenance

ACSEL has a major overhaul with Reality 2.0. While the previous versions stored the shaders as files on disk (.rsf) and the ACSEL “database” was a designated directory, with Reality 2.0 we have a real database that is capable of growing without significant performance penalties. The database is contained in a single file named “ACSEL2.db”. Its exact location changes based on the Operating System and you can find where it's located by looking in the About tab of Reality. The single file makes it simple to back up and move your ACSEL database. For example, if you upgrade your computer all you have to do is to copy the ACSEL2.db file from to old machine to the new one.

The combination of the new database and the the internal optimizations made to handle it also lead to another nice improvement: ACSEL loading is much, much faster. For example, calling Reality 1.25 with the Stonemason's “Village Courtyard” model, with ACSEL shaders installed, took about 35 seconds on our development machine. With Reality 2.0 it takes only 3.8 seconds. It's basically 10 times faster than before.

ACSEL Share

The world of DAZ Studio has a wonderful online community of dedicated and passionate users who help each other. When ACSEL was finished it became obvious that, at some point, Reality artists would want to share shaders with each other. ACSEL Share is a service, free for all Reality users, that allows you to share shaders with the rest of the world and to immediately install any available shader that other Reality users have shared. The whole catalog of shader sets is hosted by Pret-a-3D and backed up daily. The entire process works inside Reality and is point-and-click easy.

Once you have a shader set that you want to share simply click on the “ACSEL Share” tab and click on the “Refresh” button. The list of the shaders available to be installed on your computer will appear in the top pane, and the list of your shaders ready to be shared will appear in the lower pane:

Please note, in order to use this service you first need to create a free account in the Pret-a-3D forums by visiting <http://www.preta3d.com/forum>. Once you have your user ID, enter it in the ACSEL Share panel including your email address exactly as it's registered in the forums.

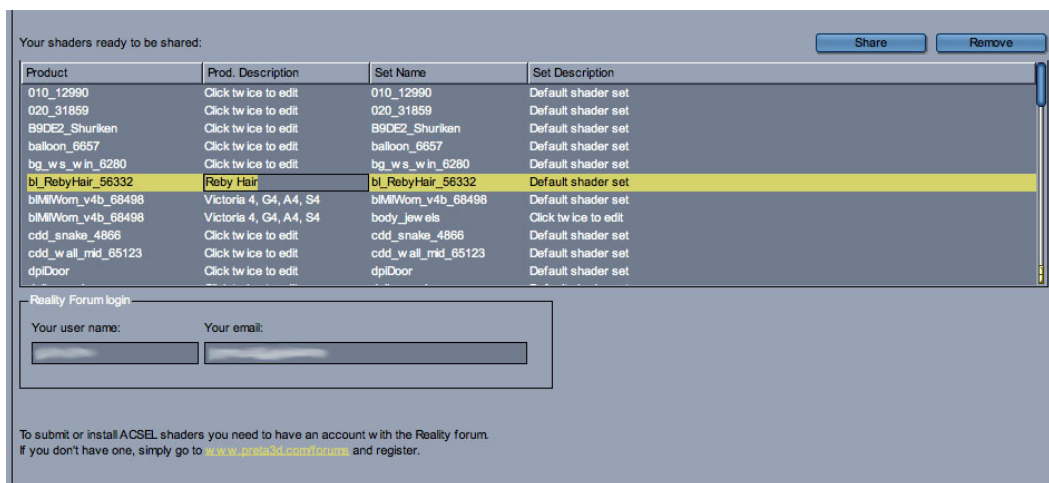
Installing a shader set

If you find a shader set for one of the models that you own, you can install it by simply selecting it in the top list and then pressing the “Install” button. Reality will connect to the ACSEL Share catalog, retrieve the shader set, install it on your disk, and update the ACSEL catalog to link that shader set to the model. To use the new shader set you just need to exit Reality and restart it. If the model linked to the shader set is in the scene, the ACSEL shaders will now be used automatically.

Sharing a shader set

All shader sets that you have created and saved for ACSEL are listed in the lower list, labelled “Your shaders ready to be shared”. To share them you need to do the following:

- If a shader set doesn't have a product description, enter one by selecting the shader set and then clicking on the “Prod. Description” column (see the following figure).



- This field must be used to identify the name of the model, for example “Victoria 4,” “Michael 4,” “Kingdom of Marrakesh,” and so on. The product description should not identify the texture used. Texture sets, like “Elite Lana” or “The Overlook” (texture package for “West Park”), must be described in the “Set Description” column.
- Look in the “Set Description” column. If you see the words “Click twice to edit” then you need to provide a description for the shader set. If you see the words “Default shader set” you don’t need to provide a description. You can, but it’s not required.
- Once you’re done with the setup simply select the shader set that you want to share and click the “Share” button. If everything goes well you will see a message thanking you for your submission.

A message for all content creators (PAs)

If you are someone who creates content for DAZ Studio (what DAZ calls a “Published Artist”), or you create content for Poser that is also used in Studio, chances are that someone will use your products with Reality. In that case you can make sure that your user will have the most accurate materials that match your vision for your products.

This is achieved via ACSEL Share. Making life easier for your customers and providing the best, most realistic, and faithful results for your products reflects directly on your sales and the satisfaction with your products. Well-supported products tend to appear in renders shared on the Web and word of mouth is an extremely powerful vehicle, especially in the era of Social Networking.

In designing ACSEL Share, the needs of PAs have been carefully considered and the result is a system that can be used easily to retrofit existing products without having to issue updates via DAZ, Renderosity or other content brokers. This lightens the load to the customers, who don’t need to re-download and re-install your product, and puts the PA in control of publishing the shader sets and doing it in a timely manner. ACSEL Share will soon provide a notification system so that user will be able to select to be notified when a shader set is shared or updated.

Light control

Realism is achieved by careful use of lights. What Reality and Lux allow you to do is to avoid using the usual “cheats” of traditional 3D lighting and start applying lighting solutions that work in real life. If you have experience in photography then the way Lux handles lights will be immediately familiar. For example, if you need to fill a part of your scene with some soft diffuse light you can set a plane on the opposite side of the subject, facing the light source, and you have a beautiful “fill” without using an additional light.

The following pictures show this principle in action. In the first picture we see a car lit by a single large mesh light suspended above it at an angle. Notice that the front of the car is dark and we can't quite see the detail of the grille.



Let's add a simple plane in front of the car to reflect the light from the top fixture:



Now the exact same scene, with no added lights, shows more details in the front of the car:



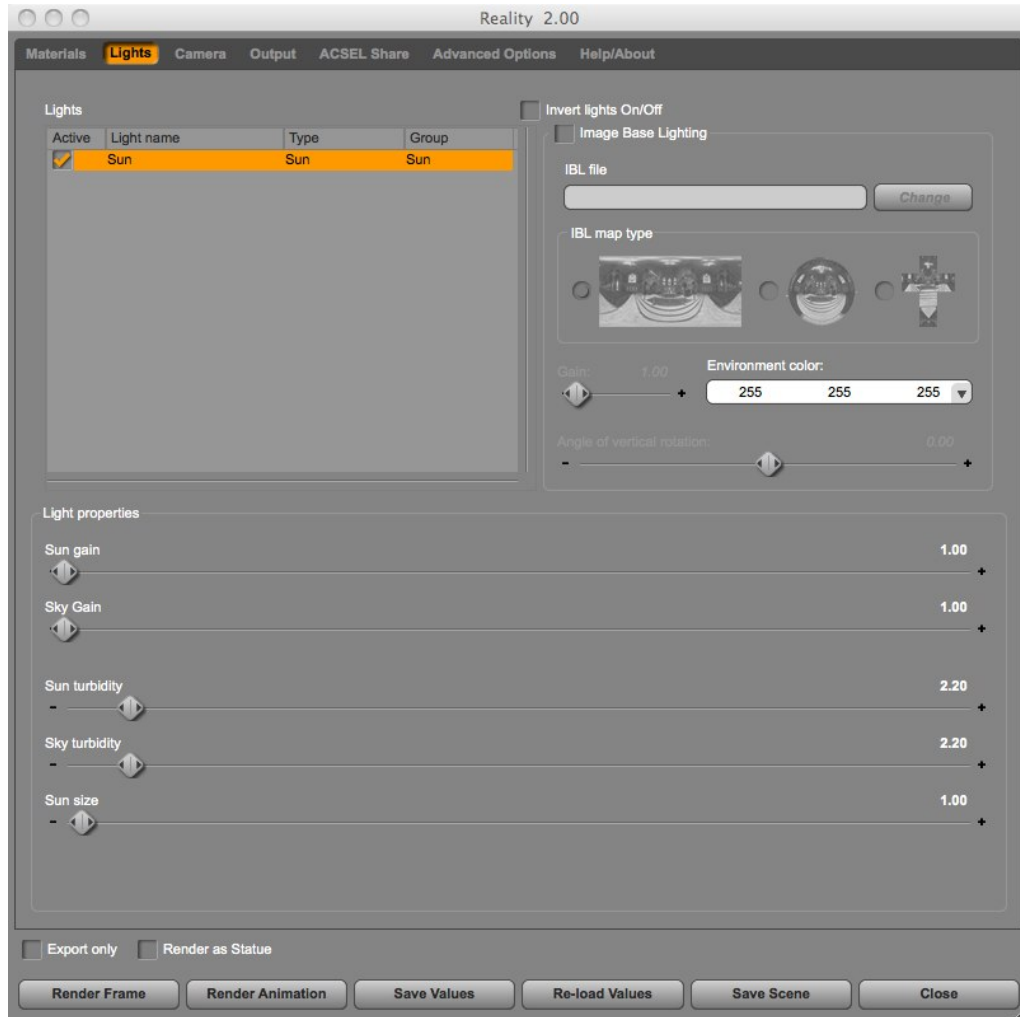
On a similar note, if the Sun light is too harsh you can place a plane between the Sun and your subject, set the plane as “Matte translucent,” and in doing so you created a diffuser in two easy steps. Remember, think like a photographer and the software will react as expected. The sooner you stop using some of the “trickery” acquired by using biased renderers, the sooner your images will look as you want.

Light types

Reality supports the following types of lights:

- **Sun.** This is a Studio distant light that is named Sun. When used, Reality will export a Sun plus atmosphere which will be rendered very believably in Lux. The angle of the Sun will determine the type of light. Horizontal Sun will look yellow-orange and Vertical sun will look cooler and more on the blue side of the spectrum. The position of the Sun in the scene is not relevant, only the angle is. You can place the Sun wherever it is convenient for you.

New in Reality 2.0 , the Sun and Atmosphere light controls are now editable: .



Of particular use is the “Sun size” control. With that you can make the sun, if it's visible by the camera, appear larger than the default size determined by Lux. The turbidity parameters can be used to create a more visible atmosphere. The default of 2.2 is what is normally used by Reality. Reality generates a separate light group for both the Sun and the Sky so that you can adjust their balance in Lux while rendering. Look in the “Light groups” tab of Lux and you will find a “sun” and “sky” group that

you can adjust independently from each other. For this effect to work you should use the “Linear” tonemapper.

- **Spotlight.** A Studio spotlight is converted to use the exact same orientation, position, and cone angle in Lux. The color of the spotlight is also preserved. If the light’s color is any shade of gray then the standard color temperature of 6,500 Kelvin is used. This can be changed dynamically in Lux when the scene renders. Spotlights usually have a soft area around the edges, where the light is “feathered.” Studio doesn’t have that control and so Reality sets the edge feather at 5 degrees. It can be adjusted to other values in order to make the edge as soft or as sharp as you need. New for Reality 2.0 is the ability to adjust the cone angle of the spotlight. If you were used having to close Reality simply to adjust the width of the spot, you don't need to do that anymore. Another new feature is the ability to set a texture for the Spotlight. This turns the spotlight into a projector.
- **Point Lights.** Reality supports Studio’s point lights which are light source that emit light in all directions. New for Reality 2.0 is the support of Studio 4 “LinearPointLight”. This light is converted automatically to the Reality pointlight. Pointlights in Lux always worked in the same fashion of the new Studio pointlight. In fact, Lux pointlights were much improved in behavior than the standard Studio 3 pointlight. You can convert existing Studio light sets that use the LinearPointLight without any issue, using Reality 2.0.
Pointlights are also enabled to use IES profiles. See the description about IES lights below.
- **Mesh Light.** These are also called Area Lights and they are geometric meshes that emit lights. These are generally the best choice for creating good lighting and for achieving realism. To create a mesh light you can simply add a plane or cube or any kind of primitive, rename it “RealityLight” and orient it in the direction you want it to spread light. A cube will emit light from all six sides, and a plane will emit light from the direction pointed by Studio’s Universal manipulator. The rule is that light is emitted in the direction of the object’s normals and the Universal tool of Studio is aligned with the normals. If you need more light sources simply add a suffix: RealityLight_1, RealityLight_fill, RealityLight_hair, etc. The important thing is that the object starts with “Real-

ityLight.” Capitalization has to be exactly as in the examples given. To make this task easier we have included three meshlights in the Reality Add-ons:

- MeshLight: this is a simple plane that is preconfigured to work as a light-emitting object. It provided beautiful lighting and it's very light-weight
- Curved Light: new for Reality 2.0 this is a “wrap-around” light that provides instant “portrait” lighting with just one fixture.
- Soft Box: emulation of a photographer's light it provides a “gobo” and a “snoot” for additional light control. See the appendix for more information.

Converting materials to lights

You can also turn any material into a light by right-clicking on its name in the material list and selecting “Convert to light.” When this option is used the material will be removed from the list and instead a light with the same name will be added in the Light tab. This option can be used effectively to convert a large amount of scene elements into LEDs and other light sources.

The following is a picture of Stonemason’s “Level 19” in which the LED materials and other parts of the mesh have been converted to lights.

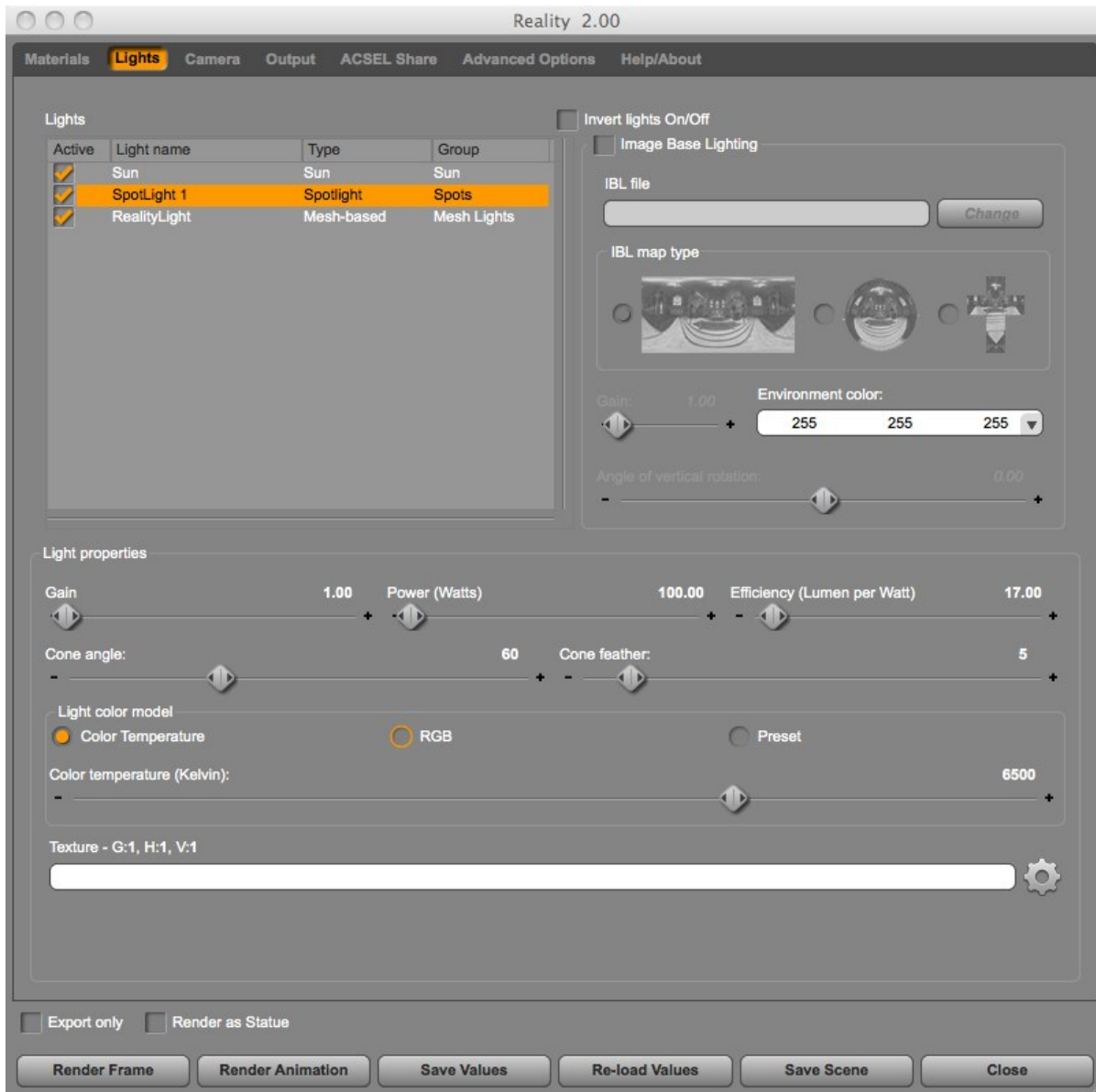


This works very nicely and can create dozens of light sources in seconds. To revert a light back to a material, right-click on it and select “Convert back to material.”

- **Distant light.** Works as in Studio. While this is technically the same kind of light that you use for the Sun, when not named Sun a distant light depends on its position and angle, like any other light. If you have cinematography experience, think of this as your 10K-20K HDMI.

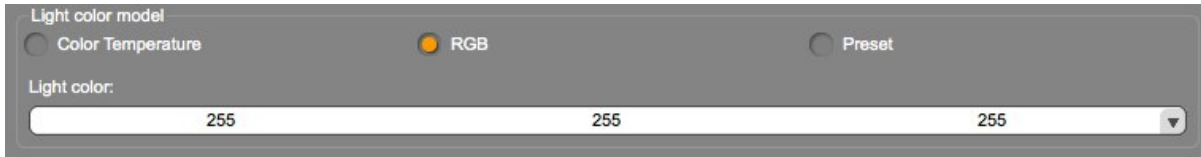
Light Editor

New in Reality 2.0 is the extended light editor that allows you to control all the aspects of each light. In addition to expanding the parameters available, the new light editor provides presets for lights and allows you to assign textures to some types of lights. Look in the “Light properties” group box below the list of lights. There you will find controls for changing the characteristics of each light. The light editor automatically shows only the controls that are relevant for each type of light. This is a new feature of Reality 2.0 and one of the many UI enhancements designed to make the program artist-friendly.

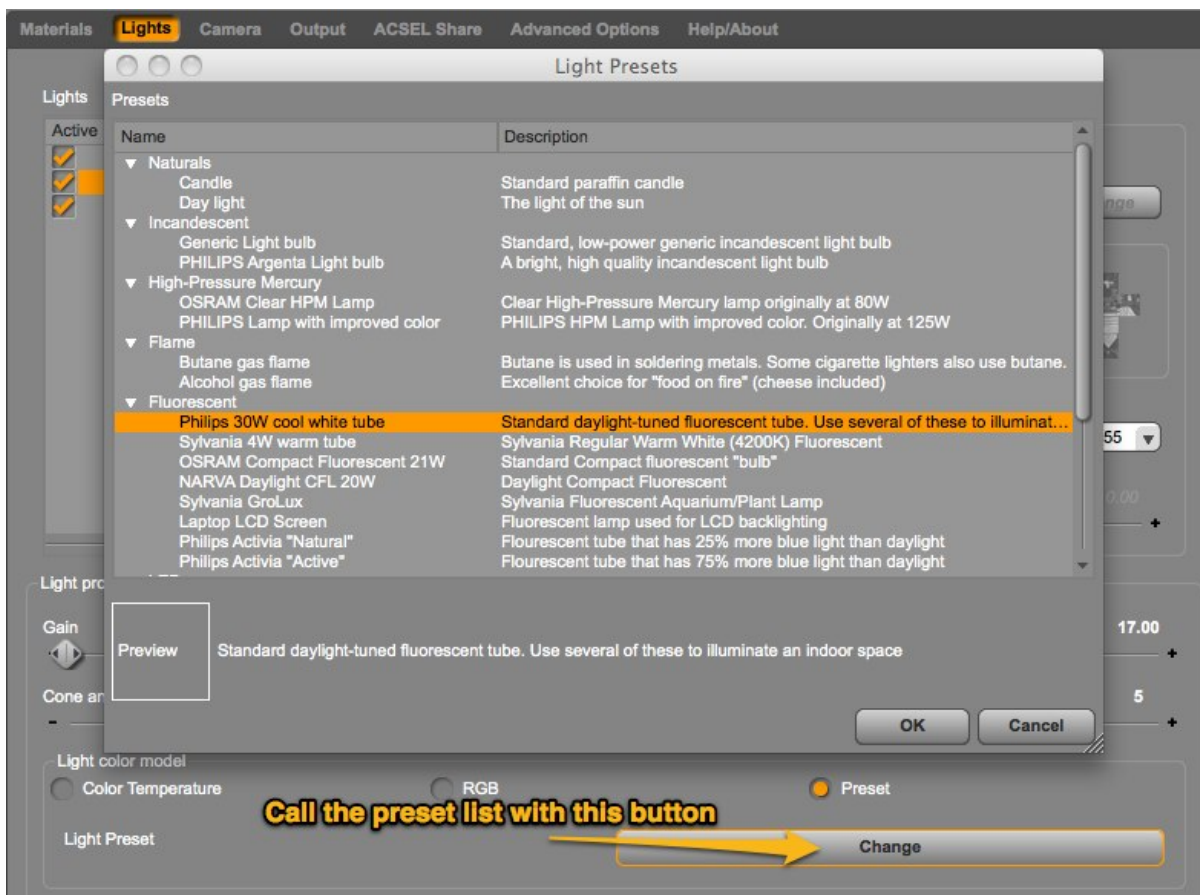


As we can see in the figure above you can set the color of a light based on the desired "model." If you are a photographer you are probably comfortable with color temperature. This is the physical model of light and the default one. You can simply move the slider and go from a very warm light, 1,800K for example, to a cold one, 8,000K and set your light from all ranges from candle light to bright sun light.

If you prefer to dial in a specific color, simply select the RGB model and you will see the familiar color picker:



Finally, you can select one of the light presets defined in Lux based on real spectral data. These presets are useful if you want, for example, to accurately reproduce the light of a neon tube. Create a cylinder in Studio, convert its material to a light and then select the preset for neon:



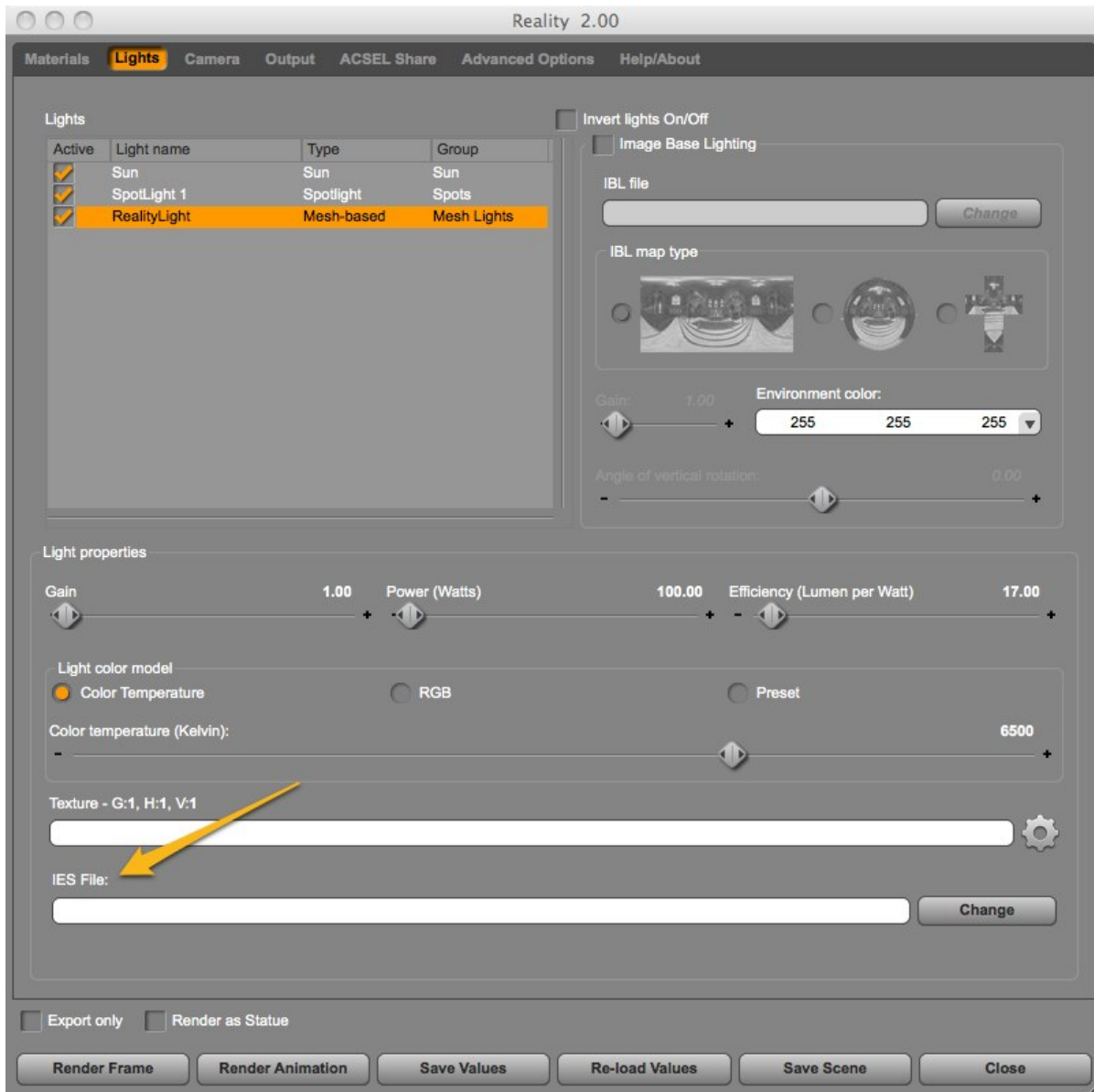
IES Lights

New for Reality 2.0 is the ability to use an IES file to make a MeshLight or PointLight behave like a specific, real, light.

IES stands for Illuminating Engineering Society. The IES standard file format was created for the electronic transfer of photometric data. An IES file contains the measurement of distribution of light. It's a digital profile of a real world light. Lux supports the use of IES files and will reproduce the physical characteristics of the light described in the IES file.

IES light files are provided by several lighting manufacturers and can be downloaded freely from their sites. One such lighting manufacturer is Lithonia Lighting (<http://www.lithonia.com>) which has an extensive library of IES files in different categories.

You can simply enter the IES file in the light editor and the rest is done automatically:



Color Temperature

Color temperature controls the spectrum shift of each light. Natural light has a color that is determined by the source. For example, incandescent light bulbs have a more yellow color than neon tubes or sunlight. The sun shifts color temperature during the day, being more yellow/orange during sunrise and sundown and being more

blueish during the day. These shifts of tints are defined as “color temperatures” and are measured in Kelvin. The following table gives a few reference values:

LIGHT TYPE	KELVIN VALUE
Candle flame	1,930
Common Tungsten bulb	2,500-2,900
Sunrise/Sunset	3,000
Quartz Light	3,200
Fluorescent	3,200-7,500 (cool white)
Daylight	5,500-6,500

Image Based Lighting (IBL)

Reality provides a great IBL system that supports full HDR (High-Dynamic Range) images both in HDR (.hdr) and OpenEXR (.exr) format. With IBL you can create incredibly realistic renders in which your 3D objects are bathed by light that is coming from all directions and that is obtained from photos of real places.

IBL in Lux uses one single image to generate both the light and the background. This means that, with the right type of image, you can create spectacular backgrounds for your scene with the same image used to generate the lighting.

In addition, Reality includes a “preview sphere” to help you adjust the rotation of the background so that it works best for the scene that you want to create.

Which IBL format?

In addition to the format used to encode an IBL file (HDR, EXR, Tiff, etc) an IBL map can be created in three different layouts:

- LatLong, from “Latitude” and “Longitude.” This layout works best with Reality and Lux for creating backgrounds.

- Angular, or “Light probe.”
- Vertical cross.

All three formats are supported but LatLong is generally your best choice.

Trying out IBL in Reality

Follow these steps to get a feel for how IBL works.

- Look in your Reality Add-on folder in the Content Manager and you will find a sub-folder called “Lights.”
- In there you will find three IBL setups ready to be used. Double-click on any of them and the Preview sphere will appear in your Studio scene. This sphere will not be rendered by Reality so you don’t have to worry about hiding it.
- Select the Sphere in the Scene tab and then look in the Parameters tab.
- Use the “Y-Rotate” slider to rotate the sphere and position the background as you prefer. Be sure that you are looking at the scene from the designated camera and that the “Show Aspect Frame” option is enabled.
- Now call Reality and look in the Lights tab. The IBL option should be enabled and the file name for the IBL map should be set. Also, the rotation angle for the sphere should be reflected in the “Angle of vertical rotation” slider.
- Without adding any other light to the scene press “Render Frame” and watch the scene being rendered with the same light that was present when the photo was taken. Amazing!

The included IBL maps are courtesy of Christian Bloch of HDR Labs and are provided under the terms of the license listed in [Appendix C](#). Please visit <http://www.hdrlabs.com> for more information about IBL and to download many more free IBL sets. Look in the “Smart IBL” page and from there in the “sIBL Archive” section. Smart IBL maps come as a package that includes several files. Read below to see how to use each file.

Setting your own IBL map

If you want to use your own IBL map and position the background using the Reality Preview Sphere, here is how to proceed.

- Download a new IBL map. For example, the “Grand Canyon C” from HDR Labs.
- Unzip the sIBL package; you will find a few files in it.
- One of the files is a JPG version of the image that is ideal for using with the Preview Sphere. In Studio, select the RealityIBLSphere and click on the “Surface” tab. Be sure to be in Advanced mode.
- In the Diffuse channel click on the name of the file just below the diffuse color and select “Browse” from the popup menu.
- Browse to the folder where you have uncompressed the sIBL archive and select the GCanyon_C_YumaPoint_8k.jpg file. After a few seconds it will load in the sphere and you will see the different preview. This file is not used for the rendering, just for the preview.
- Call Reality and click on the Lights tab. Click on the “Change” button next to the “IBL file:” input field.
- Navigate to the previous location and select the GCanyon_C_YumaPoint_3k.hdr file. The other files are not used for now.

That’s all that it takes. Now you can adjust the rotation of the sphere and the Lux render will reflect the new setup.

Please note that IBL generally creates a fairly low-intensity light. Increasing the gain to 4-6 is not uncommon. Higher values might be required if using IBL in conjunction with other lights. Reality creates a “IBL” group for this kind of light so you can adjust the intensity and other parameters directly in Lux during the render.

Using images with IBL: HDR vs. LDR

Aren’t we having fun with TLAs (Three Letter Acronyms)?

When using IBL, you will need to provide an image of some kind. One characteristic of images is whether or not they provide high ranges of light information. HDR (High Dynamic Range) refers to the ability of an image to provide a high amount of light information, higher than conventional photographs and closer to the range perceived by the human eye. LDR (Low Dynamic Range) refers to the limited range of light information handled by conventional photographs.

The human eye can detect a wide range of light with finesse, and this "HDR" ability becomes part of our experience with the real world. In contrast, film and digital cameras can only capture a small subset of light (in the low to medium ranges), resulting in LDR images. For example, if you want to take a picture of a friend facing you in a room, with her back to a bright window, your camera will be unable to take a good picture. You will either need to expose the image to have your friend's face well lit, which will overexpose the window behind her, or, you can expose for the bright window but then the face will be just a dark oval. This problem demonstrates the limitations of cameras.

When implementing IBL, you can use LDR images, but the result will be a very low-contrast type of lighting. In that case you can improve your render by adding one or more lights positioned to reflect the light into dark areas of the scene.

The ideal solution is to use HDR images for IBL. Photographers create HDR images by taking multiple exposures of the same image and then combining all those exposures inside a single file that contains light information that a program like Lux can use. You can often find these images for purchase on the Internet.

IBL without images

You probably noticed that there is a color control in the IBL editor. You can use that color to add a tint to the picture, in case you want to shift the tone of it. Or, you can use it by itself without any picture. As long as you don't use the pure white (255,255,255), Reality will export the color to Lux. If you have a picture the color will be combined, multiplied, with the picture. If you don't use any picture you will still get an excellent even lighting of the whole scene with the selected color. This can be an excellent way of adding some dim, blueish light to a night scene lit by a spot to simulate the moon. There are many other uses. Experiment with this to find creative ways of adding diffuse light to your scenes.

Light portals

Picture this scenario: you want to render an indoor scene that is lit by the sun filtering through some windows. You set the sun in the right direction, place the camera

and start the render. It's all good but now Lux, which has Indirect Lighting on all the time, is also wasting time rendering and calculating the materials of the exterior of the building that is not visible -- all those walls that are facing the sun and that are never going to be seen by the camera. There should be a way of stopping all that waste of good CPU cycles and memory!

There is a way, and it's called a "Light Portal." A portal is, as the name implies, an object that lets the light go through. A light portal **is not a light source**; you will need to have the Sun in the scene as usual. The portal gives hints to Lux about what parts of the scene are important or irrelevant. It helps optimizing the thousands of decisions that Lux needs to take about tracing the path of light.

A portal has a direction. In this example, it should face the "indoor" part of the scene and it should have its "back" to the outside part of the scene that doesn't need to be rendered. The direction of the portal is determined by the surface normals¹ of the shape.

It works in this way:

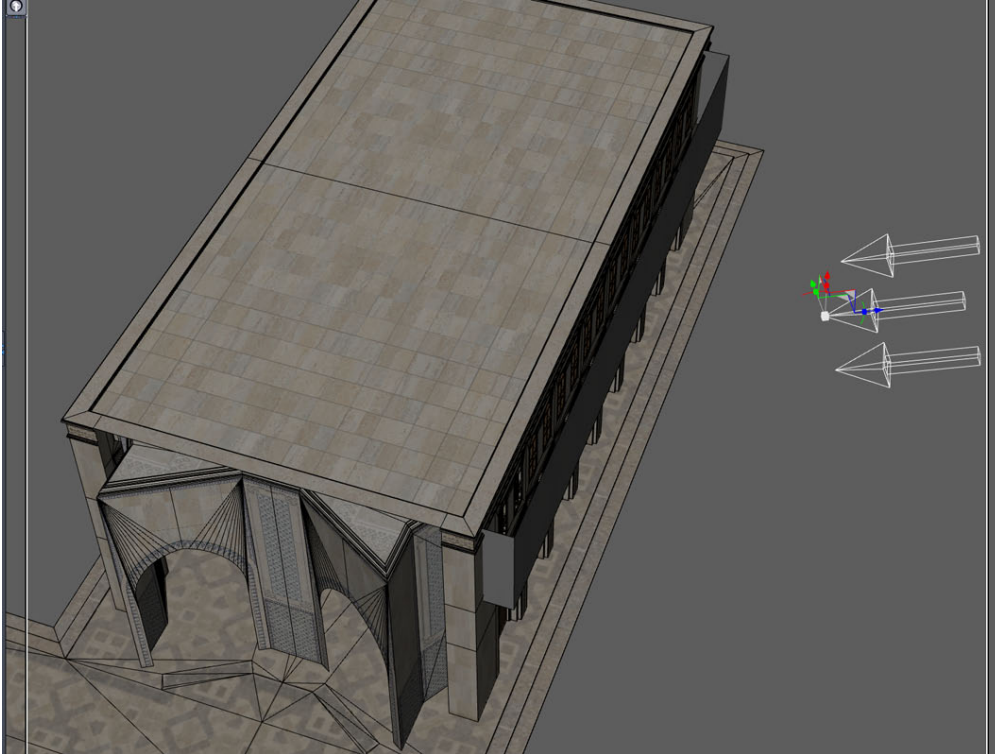
- In your scene place a portal on the outside of the building to cover the windows.
- Size the portal to be slightly larger than the area of the window and move it close to the wall of the building.
- Render as usual. The portal will be completely invisible in the final image but it will speed up the process considerably and generate cleaner images.

Reality includes a handy "Light Portal" prop. Place it in the scene as described above and use the "flaps" to orient it correctly. Lux needs to have the portal facing in the right direction. To determine the right direction Lux uses the object normals.

Those have to point toward the windows.

¹ A surface normal, or simply normal, to a flat surface is a vector that is perpendicular to that surface. In 3D modeling normals are used to describe the direction to which a shape is facing.

Because Studio doesn't have a way of showing the normals, Reality's portal has convenient bent sides. Just point them toward the window, as shown in the following figure:



Light Portals only work with the Sun, distant lights and with IBL. They have no effect with mesh, spot or point lights.

House lights

In movie production a lot of scenes are shot inside what is called a "sound stage," basically a large building that can be configured to host many different stages. Inside the sound stage are local light fixtures known as "house lights," which are not meant to be part of the shot but are turned on to help people move about the set before filming begins. When cameras start rolling the house lights are turned off and the movie lights are turned on to light the scene as the cinematographer designed. In Reality you can use a similar approach. If you need more light in a part of the scene to see better what's in the frame, you can simply create a spotlight, call it "House Lights", and that light will be turned off automatically by Reality, before rendering.

If you need more lights of this kind you can simply add a suffix of your choice. For example: “House Lights 1,” “House Lights balcony,” “House Lights basement” (don’t use quotes in the names). As long as a light’s name begins with “House Lights” it will be automatically turned off by Reality.

The Lights tab

Before hitting the “Render” button it’s a good idea to check your lights. They are listed in the “Lights” tab. Lights can be turned on or off in this page directly, by checking or unchecking the “Active” box. The status of the light is remembered. For mesh lights you can adjust the brightness up to 5,000 watts. The “Invert Lights” button swaps the active state of each light so that you can switch which ones are on or off.

Light groups

Lux has a powerful feature that allows you to adjust the brightness and color of light **while** the render is running. To use this feature you need to create light groups. When one or more groups have been defined then you can use the “Light Groups” tab in Lux to adjust the brightness, color temperature or tint.

Reality automatically sets up some groups.

- The sun is set in its own group called “Sun” and the sky has its own “Sky” group.
- Spotlights are grouped in “Spots”
- Mesh lights are grouped in “Mesh Lights”
- Point lights are grouped in “Point Lights”
- Distant lights are grouped in “Misc.”

Custom light groups

Reality allows you to define your own groups for finer control. In the light editor click on the light for which you want to redefine the group name. Then click on the group name. You will be able to edit the name to your liking. Using the same group name for multiple lights will group them together. Please remember that this feature can take more memory during rendering so it’s best to use the minimum amount of light groups necessary.

Camera control

The Camera tab is used to control the use of the camera, including the use of DOF (Depth Of Field), the exposure controls and, new for Reality 2.0, the Film Response.

Selecting the rendering camera

Reality will use whatever camera is active for rendering. If you invoked the plug-in while looking at the scene from a different camera than the one meant to render, you don't have to close Reality and switch. Just use the drop-down list of available cameras, select a camera, and Reality will switch the view for you.

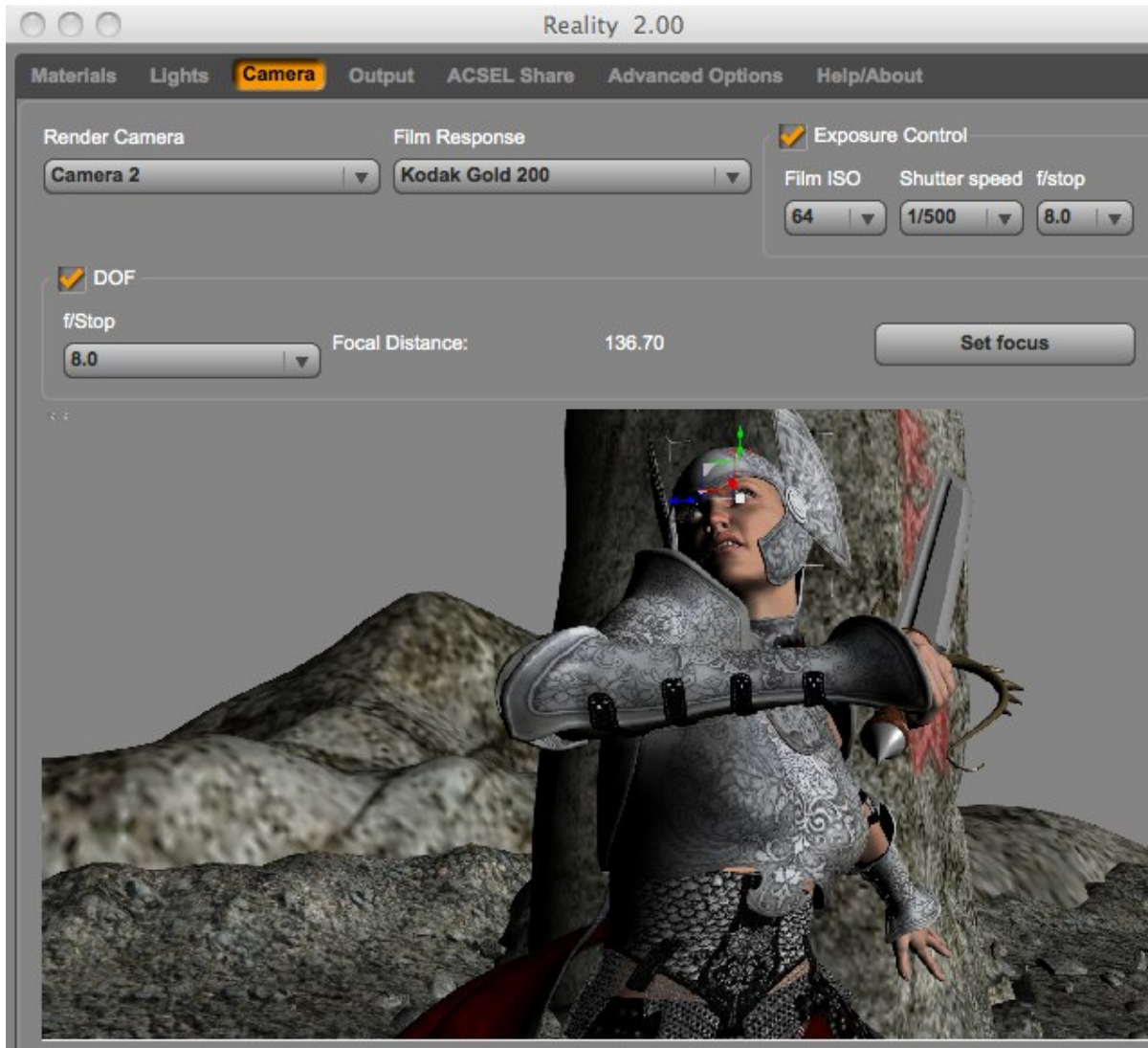
Shallow depth of field

You can get beautifully shallow depth of field with Lux. Reality makes this operation as easy as point-and-click.

- If you plan on using this feature, before launching Reality select the object that you want to be in focus. You can use Nulls for this too.
- Once in Reality, click on the Camera tab.
- Check the "Enable DOF" box.
- Select an f/stop (lower values make the out of focus part blurrier).
- Press the "Focus on selected object" button.

This operation will calculate the distance from the camera to the selected object. The result is shown in the "Focal distance" label in the window. Now, when you render the scene, the objects closer or further away from the focal point will be increasingly blurred, like the effect of using a 35mm camera in similar conditions.

The following picture shows the DOF controls used to focus on the left eye, selected, of Victoria 5:



Exposure controls

Instead of starting a render and then having to adjust the values for Film speed, Shutter speed, and f/stop all the time, you can set those values directly in Reality and they will be saved with your scene. New for Reality 2.0 is the addition of f32 for both the exposure and the DOF.

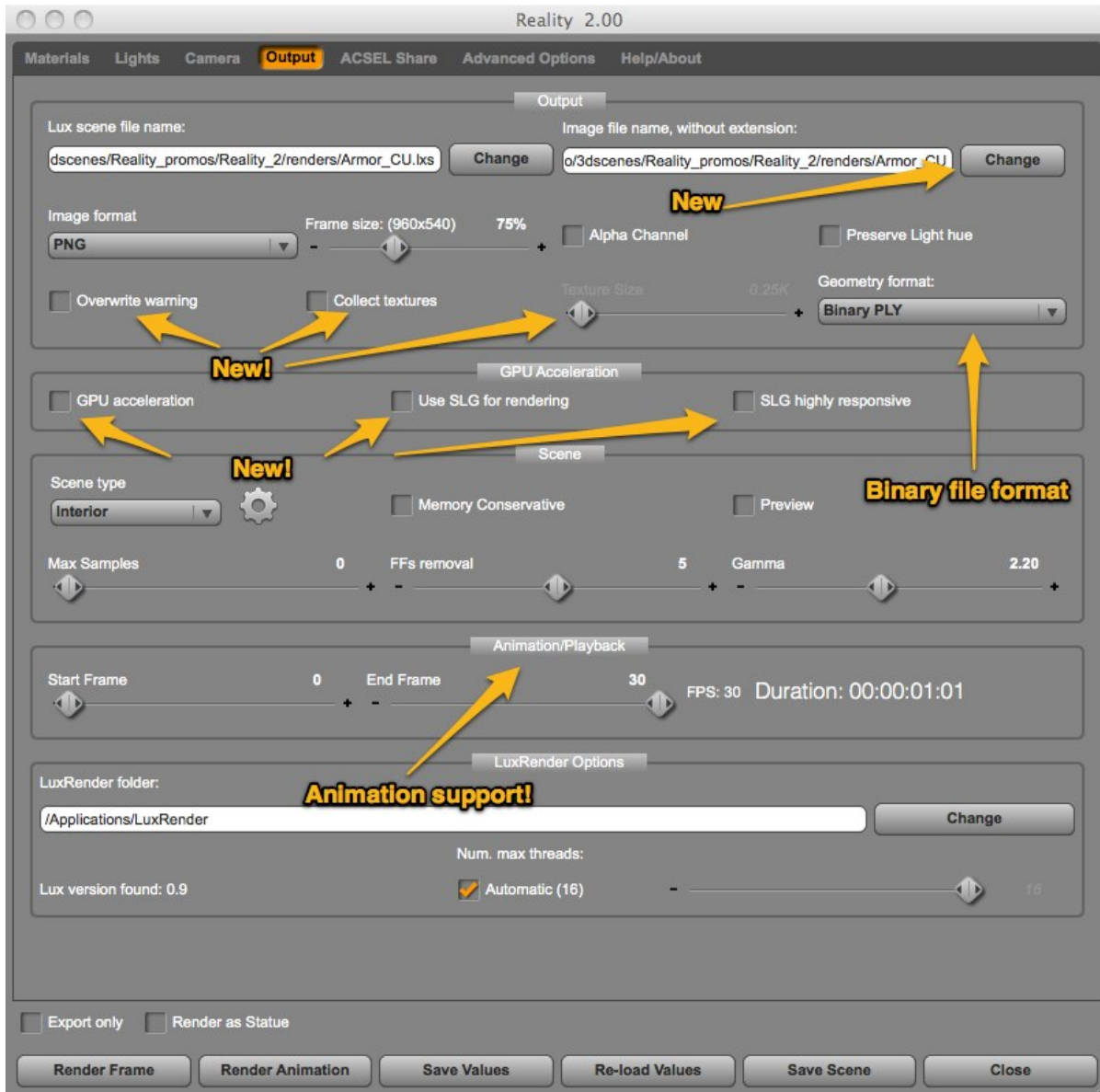
Film Response

This feature allows you to use one of the many film emulations of Lux. Film response profiles are descriptions of the way several real film stocks react to light and color. For example, the Kodachrome series of films have been used in the past by photographers to achieve a certain look because of the very specific color response of that film. Reality provides access to the whole set of Lux film response files with a twist: the Reality descriptions are a bit easier to read. We have done extensive research with the manufacturers to get the right name for each film type. As soon as possible we will update the LuxRender code to reflect the same naming scheme.

Some Film response names have the “B&W” suffix to indicate that its an emulation of “Black and White” film. Some names have the “Color Rev.” suffix to indicate diapositives or film for “slides.” This is called “color reversal” film because the final result is expected to be projected and not printed on paper.

Output options

The Output panel has changed considerably for version 2.0, as some new options have been added and existing options have been re-arranged to make this panel easier to use:



A LuxRender scene is generally composed of a few files. The way Reality exports your Studio scene to Lux is via 3 files:

- **reality_scene.lxs** This base file contains the general scene definition, including the resolution, format, and name used for the rendered image. This file references the following two “include” files.
- **reality_scene.lxm** This file contains the definition for the materials used in the scene.
- **reality_scene.lxo** This file contains the exported geometry of the objects used in the scene.

The Output tab allows you to change the name of the base file, which will determine the name for the .lxm and .lxo include files. These three files and their purpose are mentioned here purely to give you all the information necessary to back up and transfer your scene files. Copying the .lxs file is not enough. You have to get the corresponding .lxm and .lxo files as well.

Output path

When you select a new destination folder for your scene Reality will set that folder as the default destination for the new scenes created afterward.

File type

The output file formats supported for your render are PNG, EXR, and Targa. The size of the render is taken directly from Studio but you can also select to have the final render at 100%, 75%, 50% or 25% of the size, or at 200%, 300% and 400% of the size. This is a great way of making test renders that take a fraction of the time of the final scene. The “% of frame rendered” parameter is persistent, and it’s remembered across invocations of Reality for the same scene.

Alpha Channel

The Alpha Channel checkbox allows you to create an alpha channel for your scene in case you want to do background replacement in graphic packages like Adobe Photoshop. When this option is selected, any part of the scene that is not covered by some object will be set to transparent.

Preserve Light hue

When the camera faces a light, and that light has a certain color, or hue, the objects lit by that light will take the hue but the light source itself will render almost pure white. If you enable this option then the light hue will be preserved.

Overwrite Warning

This is new for Reality 2.0. When this checkbox is enabled Reality will warn you if the scene file (.lxs) that you are creating for the export already exists on disk. You can then decide to overwrite it or not. This flag is saved for each scene. The default for new scenes, on or off, is decided by the setting in the Advanced tab.

Collect Textures

Another Reality 2.0 feature. Collecting textures allows you to store all the textures for the scene in a location where they can easily be copied. For example when you need to move the scene to a different computer. Reality created a subdirectory named “name of the scene file”-resources. For example, if the scene file is called “reality-scene.lxs” then a directory named “reality-scene-resources” will be created at the same subdirectory where the scene file is stored. In that directory Reality copies all the files necessary to render the scenes. There is where the collected textures are stored.

Texture Size

Together with “Collect Textures” this option allows you to greatly reduce the amount of memory required to render your animation by resizing the textures. For example you can set each texture to be no wider than 1024 pixels.

During the texture collection Reality will check the width of each texture and compare it against the maximum value selected. If the value is higher, then the texture is resized. Reality saved the scaled texture as a PNG file to avoid loss of data due to transcoding. The minimum size of the texture is 256 pixels wide (0.25K). The maximum is 2048 pixels wide (or 2K). The last value for this slider is “Full size.” This allows you to collect the textures without resizing them.

Geometry Format

New for Reality 2.0. Lux supports the use of the Stanford PLY file format. PLY is a simple format for the storage of geometry, similar but simpler than the OBJ file format. It's widely used in 3D applications and the popular Open Source program Meshlab reads it.

PLY files can be written as text or binary files. Binary files take much less time to be written and to be read by Lux, resulting in less disk space used for the storage of the scene and faster turn-around time.

Reality defaults now to binary PLY for the export but it supports also the text PLY and the legacy Lux-native text format. PLY is mandatory for rendering with SLG (see "Use SLG for Rendering").

GPU Acceleration

If your computer has a OpenCL-compatible video card (note, different from OpenGL) you can now take advantage of that additional computing power by selecting this option. Another new feature of Reality 2.0, it enables the rendering via GPU of LuxRender. GPU acceleration is limited to the Interior and Exterior scene types but you don't have to worry. If you select a scene type that doesn't support GPU rendering and you have selected GPU acceleration, Reality will warn you and it will disable the GPU option.

Use SLG for Rendering

SLG, or SmallLux GPU, is a very special renderer that is blazingly fast and physically accurate. SLG uses a subset of the Lux rendering engine and utilizes the GPU almost exclusively. It has several limitations compared to Lux. The materials are not nearly as sophisticated, for example there is no support for specular maps, but the program can do some very impressive renders in a matter of seconds. This is analogous to Blender's Cycles or other commercial GPU-only renderers. SLG is installed together with Reality. To use it you must be sure that you have an OpenCL-capable video card and that the OpenCL drivers are installed and working properly. At the cost of being pedantic, please note that OpenCL is not OpenGL and that the two technologies are not related. OpenCL is also different from CUDA so, if you have a

CUDA-supported card, you still need to have proper OpenCL drivers installed if you are on Windows. On Mac OS OpenCL is built-in into the system since Mac OS 10.6. You only need to be sure that you have a card that is providing the computational power for it. Please note that even low-end nVidia cards and some old MacBookPros officially support OpenCL but the effective processing power is very limited. Be sure to verify your graphic hardware before you try this feature. Your card manufacturer's website is the best resource for this kind of information. At the time of this writing only AMD/ATI and nVidia cards support OpenCL.

When using SLG Reality automatically enables texture collection. This is a requirement because SLG loads all the textures inside the GPU's memory, which is usually much less than your computer's RAM. The normal size for textures when using SLG should be 0.5K. In rare cases it might be necessary to use the 0.25K option. When rendering with SLG Reality will open the "SLG monitor," a window that shows you the output messages from the renderer. SLG is not as robust as Lux and it doesn't have much of a User Interface so, if it crashes, you will find the error messages in the SLG Monitor:

```

[SLG] [PathOCLRenderThread::0] Compiling AdvancePaths Kernel
[SLG] [PathOCLRenderThread::0] Kernels compilation time: 2680ms
[SLG] [PathOCLRenderThread::1] FrameBuffer buffer size: 14462Kbytes
[SLG] [PathOCLRenderThread::1] Camera buffer size: 0Kbytes
[SLG] [PathOCLRenderThread::1] MeshIDs buffer size: 551Kbytes
[SLG] [PathOCLRenderThread::1] Normals buffer size: 1557Kbytes
[SLG] [PathOCLRenderThread::1] UVs buffer size: 1038Kbytes
[SLG] [PathOCLRenderThread::1] Vertices buffer size: 1557Kbytes
[SLG] [PathOCLRenderThread::1] Triangles buffer size: 1654Kbytes
[SLG] [PathOCLRenderThread::1] Materials buffer size: 4Kbytes
[SLG] [PathOCLRenderThread::1] Mesh material index buffer size: 0Kbytes
[SLG] [PathOCLRenderThread::1] SunLight buffer size: 0Kbytes
[SLG] [PathOCLRenderThread::1] SkyLight buffer size: 0Kbytes
[SLG] [PathOCLRenderThread::1] TexMaps buffer size: 98304Kbytes
[SLG] [PathOCLRenderThread::1] TexMaps Alpha Channel buffer size: 6144Kbytes
[SLG] [PathOCLRenderThread::1] TexMaps description buffer size: 0Kbytes
[SLG] [PathOCLRenderThread::1] Mesh TexMaps index buffer size: 0Kbytes
[SLG] [PathOCLRenderThread::1] Mesh BumpMaps index buffer size: 0Kbytes
[SLG] [PathOCLRenderThread::1] Mesh BumpMaps scales buffer size: 0Kbytes
[SLG] [PathOCLRenderThread::1] Ray buffer size: 2048Kbytes
[SLG] [PathOCLRenderThread::1] RayHit buffer size: 1024Kbytes
[SLG] [PathOCLRenderThread::1] OpenCL buffer creation time: 0ms
[SLG] [PathOCLRenderThread::1] Size of a GPU Task: 552bytes (12 + 452 + 88)
[SLG] [PathOCLRenderThread::1] Tasks buffer size: 35328Kbytes
[SLG] [PathOCLRenderThread::1] Task Stats buffer size: 256Kbytes
[SLG] [PathOCLRenderThread::1] Defined symbols: -D PARAM_TASK_COUNT=65536 -D PARAM_IMAGE_WIDTH=1280
-D PARAM_IMAGE_HEIGHT=720 -D PARAM_RAY_EPSILON=1.000000e-04f -D PARAM_SEED=1389299 -D
PARAM_MAX_PATH_DEPTH=6 -D PARAM_MAX_DIFFUSE_PATH_VERTEX_COUNT=5 -D PARAM_RR_DEPTH=5 -D
PARAM_RR_CAP=2.500000e-01f -D PARAM_ACCEL_QBVH -D PARAM_ENABLE_MAT_MATTE -D
PARAM_ENABLE_MAT_ALLOY -D PARAM_HAS_SKYLIGHT -D PARAM_HAS_SUNLIGHT -D
PARAM_DIRECT_LIGHT_SAMPLING -D PARAM_DL_LIGHT_COUNT=0 -D PARAM_HAS_TEXTUREMAPS -D
PARAM_HAS_ALPHA_TEXTUREMAPS -D PARAM_HAS_BUMPMAPS -D PARAM_IMAGE_FILTER_TYPE=0 -D
PARAM_SAMPLER_TYPE=2 -D PARAM_SAMPLER_METROPOLIS_LARGE_STEP_RATE=4.000000e-01f -D
PARAM_SAMPLER_METROPOLIS_MAX_CONSECUTIVE_REJECT=512 -D
PARAM_SAMPLER_METROPOLIS_IMAGE_MUTATION_RANGE=1.000000e-01f -D __APPLE__
[SLG] [PathOCLRenderThread::1] Compiling kernels
[SLG] [PathOCLRenderThread::1] Kernels not cached
[SLG] [PathOCLRenderThread::1] Compiling Init Kernel
[SLG] [PathOCLRenderThread::1] Compiling Sampler Kernel
[SLG] [PathOCLRenderThread::1] Compiling AdvancePaths Kernel
[SLG] [PathOCLRenderThread::1] Kernels compilation time: 152ms
[SLG] [Telnet server] Thread started
[SLG] [Telnet server] Server started on port: 18081

```

Close when finished

Stop Close

You can see a good example of the power of SLG in the YouTube video at <http://www.youtube.com/watch?v=s0qAZmGxTzU>

SLG is extremely fast. For example the following scene was rendered in 90 seconds:



The GPU used was an AMD Radeon HD 5870 and the rendering happened with several applications opened, including 12 tabs in the Chrome browser. This is a real-life scenario.

Moving around

As remarkable as SLG is, things get even better. If you click and drag on the picture while it's rendering you will be able to move the camera around. When you release the mouse the picture will be re-rendered using the new camera angle. There are several operations that can be performed with the mouse. For the following description I will assume a right-hand mouse; if you use your left hand please make the necessary adjustments:

- Left click and drag: rotates the camera
- Right-click and drag: left and right will move the camera, up or down will push the camera backward or forward, respectively.

You can get a very terse but useful help screen by pressing the letter “h” in SLG. By default Reality sets the reaction time for SLG to be fair but not too “snappy.” That’s because fast response to mouse actions require to poll the mouse frequently and that slows down the rendering process. If you want SLG to be as responsive as possible then enable the checkbox titled “SLG highly responsive” in the GPU acceleration section of the Output tab.

Scene type

Lux can use a myriad of different rendering options based on the type of settings. The parameters are really complex and can make a grown man cry. To make things simpler Reality groups all those settings in a series of “Scene types.” Most of the time you will need to use just the Interior type. This provides the highest quality of rendering for both interior and exterior scenes. Sometimes you might get some speed advantage by using the Exterior type while having an outdoor scene lit by the sun.

New for Reality 2.0 are the “Photon Map” and “Smart Photon Map” scene types. These are provided for the more adventurous types who like to test and experiment. The plain “Photon Map” corresponds to Lux’s “ExPhotonMap” and it’s a fast rendering option that brings results close to the Interior type but possibly in much shorter time. This is a description of it from the LuxRender Wiki:

“Exphotonmap is physically based, and consistent. It can simulate more or less all the optical phenomena supported by LuxRender, and given precise enough settings, should look basically correct. With that said, exphotonmap is biased, so it intentionally takes some shortcuts; how much this affects the rendered result depends on which render settings are being used. Exphotonmap is not going to render your scene in a blink of an eye. Although when configured properly it will outrun path tracing (Exterior) and bidirectional (Interior) path tracing by a long shot, it is still spectral progressive rendering.”

Words between parentheses are to indicate the corresponding Reality settings. You can find the full technical description of this configuration at http://www.luxrender.net/wiki/Intro_to_ExPhotonMap.

In some cases “Photon Map” can be twice as fast as Interior. Please note that Photon Map will take a long time up front during which it calculates the photon map. During this time you will not see any image being rendered. This is normal. You can see the activity progress in Lux's Log tab. After the photons are calculated the rendering will happen quite quickly.

The Photon Map scene type has several parameters that are exposed by the Scene Parameters Editor (see below). Particularly, the amount of direct, indirect and caustic photons can play a large role in the final result. If you want to try the Photon Map type but you are intimidated by the complexity of the parameters, let Reality do the job for you. Reality 2.0 performs a quick analysis of the scene when it's exported and it will adjust the photon map parameters for you to obtain the right result in the shortest amount of time possible.

This technology, called “SMASH” (Smart Material Analysis and Scene Harmonization) is another way Reality helps you tackle the power of Lux without getting overwhelmed by details that take away your artistic inspiration.

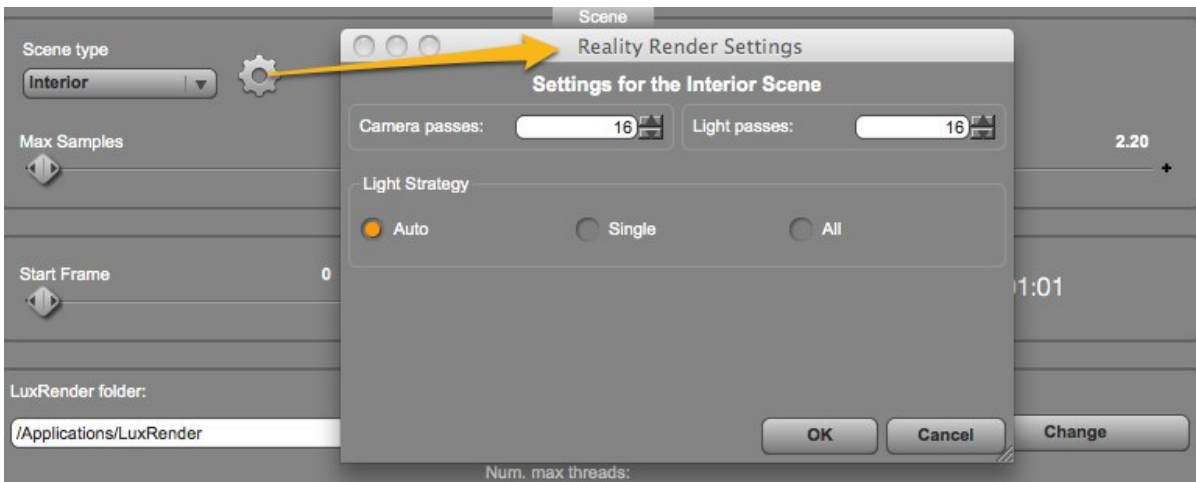
Smart Photon Map, what Lux calls SPPM, is a variation on the Photon Map theme in which the number of photons used is variable and dynamically adjusted as the rendering proceeds. It has the characteristic to start rendering almost immediately and produce a very “splotchy” image with green color tones. Progressively the image becomes more detailed and the green color cast goes away. This scene type has provided very good results but it's still a work in progress in Lux 0.9.

Scene Parameters Editor

New for Reality 2.0, there is the option to edit the internal rendering parameters passed to Lux for each scene type. This allows you to optimize Lux for the specific scene that you are rendering. This option is best used by people who are quite familiar with Lux's internals. Otherwise you can simply continue with the defaults. To call the editor click on the “gear menu” next to the Scene type drop-down menu:



When invoked the first time the editor presents the defaults used by Reality. Switching from one Scene type to the other erases the previously customized parameters.



After you close the editor you'll see a new icon that indicates that the values have been edited, looks like two engaged gears.



Memory conservative

This option uses a feature of Lux that allows to you render a scene with a less memory intensive configuration. This system is a bit slower but it can save a sizable

amount of RAM and can come in handy for machines that use a 32 bit version of XP, which limits the amount of RAM available to Lux to 2GB (32 bit Mac applications can access 4GB of RAM). This option could also come in handy with machines using 64 bit OSes but with 4GB or less of RAM.

Preview

This option enables a faster rendering setup in Lux that can be up to 200% times faster than the standard render. It's an excellent option to check parts of the scene quickly, although not as accurate as with the standard mode.

Max samples

The "Max Samples" slider is used to force Lux to stop rendering after a given number of samples per pixel is reached. Leave it at zero for no limit. This slider is also essential for the rendering of animations, both with Lux and with SLG. Setting it, for example, at a value of 500 will stop the rendering of a frame when that number of samples has been reached, and will cause Lux to start rendering the following frame. Otherwise, if left at zero, Lux would never move past the first frame in the series.

FF Removal

To help in removing certain types of rendering noise, nicknamed "Fireflies," Lux has a special parameter that works as a "sentinel." This rendering parameter was used by Reality 1.2 with a fixed value of 5. In Reality 2.0 it can be adjusted to your liking. Higher values can be more effective but will introduce bias. Please note that the normal rendering noise that you see at the beginning of the rendering is NOT an indication of the presence of Fireflies. Fireflies are very large "specks" of white pixels that either stay or get worse over time. They don't appear often and they are frequently caused by over-lighting very shiny surfaces. In any case the parameter is used by default by Reality. You can set the value to zero to marginally speed up the rendering.

Gamma

This value can be used to alter the gamma correction to the scene, if necessary. The gamma value is set up automatically by Lux to be correct for your system. If you are not familiar with this value it's better to leave it at the default.

Start Frame/End Frame

Reality 2.0 has full support for animation rendering. If your Studio scene is animated, you can render any range of frames. Simply select the start and end frame and Reality will show you the duration of your animation based on the FPS value and the length of the range. The format used is the usual timeline/SMPTE format found on many non linear video editors like Adobe Premiere or Final Cut Pro. The values are represented as:

HH:MM:SS:FF

Where HH is hours, MM is minutes, SS is seconds and FF is frames.

If you want to automatically create a file name that includes a number corresponding to the frame, simply add a certain number of “#” characters at the end of the file name. This feature works both for the scene file name (.lxs) and the image file name. For example, if you change the scene file name from the default to “reality_scene_#.lxs” and you render frame 42 of your animation, Reality will export the scene to a file named “reality_scene_42.lxs” (and “reality_scene_42.lxm” and “reality_scene_42.lxo”). If you use more # signs the file name will be padded with zeros if necessary. For example, if the pattern used is “reality_scene_####.lxs” the generated scene file will be named “reality_scene_0042.lxs”. If you want to use the same feature for the image file name you can add the “#” to that name too. Simply using the “#” on the scene file will not automatically apply the auto-numbering to the image file as well. If you render an animation and you forget to add the “#” symbols Reality will add automatically a number of those symbols sufficient to represent the highest frame number in the animation.

Once you're done setting your animation range all you need to do is to click on the “Render Animation” button. Reality will export each scene file and then call LuxRender, adding the full animation set to the Render Queue. You will be able to see the full set of scene files passed by clicking on the “Queue” tab in Lux.

The result of this rendering is a set of single frames, usually PNG files. To create the final animation, as it's standard practice, you will need to load the frames into an editing program like Adobe Premiere. There are also free alternatives like iMovie (Mac OS) and VirtualDub (Windows).

Animation rendering with SLG

You can, of course, render animations with SLG as well. The procedure is slightly different because SLG doesn't have a render queue. Therefore each frame has to be passed individually to SLG to be rendered before moving to the next one. When rendering an animation with SLG the renderer actually doesn't show any visible screen element. It runs in the background in "batch mode." This is completely controlled by SLG. For this reason the best way to check the progress is to look at the SLG monitor window. Under Mac OS the SLG monitor will also display the current status of the frame with a refresh every 30 seconds. Unfortunately there are some issues about file sharing under Windows and so this feature is not available on that OS. We are working on fixing it and will provide a fix if a solution is found.

LuxRender options

In this section you select the path to where LuxRender is installed and how many threads should be assigned to the render. Next to the selected path for Lux, Reality confirms the version of the program.

Advanced Options

Realty 2.0 provides a new panel to set up some advanced parameters and default values:



- **Max Material Previews Cached.** The Material Preview caches a certain number of previews in memory. This makes it possible to show the preview for a number of materials without re-rendering them. The number of previews kept in memory is determined by this parameter. By default it is set at 5 and it can be changed to any number up to 20. Change this parameter only if you know that you have enough memory. Each material preview requires about 45 kilobytes of RAM.
- **Overwrite warning default.** This option sets how the overwriting of scene files is set for each new scene. If it's off then the Reality parameters for the new scene will be set to not warn you. If it's on then by default Reality will ask you confirmation before overwriting the scene file set in the output tab (the .lxs file). After the initial value is set using this default you can change it in the Output tab.
- **Lux Display Interval.** This allows you to change the frequency of the screen refresh of Lux when rendering an image. Longer refresh times can result in slightly faster rendering because less CPU time is spent in refreshing the image on screen.
- **Show Surfaces tab when Reality starts.** This setting, if enabled, causes Reality to show the Studio's Surfaces tab and to show in the Surface editor the ma-

terial that is selected in Reality. If you feel that this is distracting or that it slows down your work you should disable it. This is an application-wide configuration.

- **Lux write Interval.** This parameter sets how often Lux should save the rendered image to disk. The default is to save the image every two minutes. Sometimes it might be beneficial to specify longer intervals to minimize the time spent saving to disk. On the other hand you risk to losing a significant amount of rendering if you extend this interval.
- **Use GPU by default.** If this parameter is enabled new Studio scenes will have the “Use GPU” flag in the Output tab enabled by default.
- **Memory Conservative.** If this parameter is enabled new Studio scenes will have the “Memory conservative” flag in the Output tab enabled by default.
- **Use Studio lighting model.** This flag causes a behavior change in the material converter of Reality. When this flag is enabled (the default) Reality selects which Lux material to use (Metal or Matte) not by its internal heuristics but by looking at the “Lighting model” of the Studio shader. For example, if the model is set to “Metallic” then a Lux metal will be selected to be the converted material. The same is for Matte. Be aware that this can have some dramatic consequences. For example, the “West Park” environment by Jack Tomalin uses the metallic model for almost all the shaders, resulting in an “aluminum foil” version of it when using Reality. As shown by the following image.



The effect was quite stunning and that's the reason why we decided to keep this image, and in fact feature it in the promotional material for Reality. Nevertheless this may not be what you expected. So, remember, if everything starts looking too metallic, check the Advanced Options.

The About tab

Up until Reality 2.0 the About tab was just the average credit box. With version 2.0 it's actually a place where you can find useful information. There you can find the exact location of your ACSEL database file, useful for backing up your shader data. You also find the location of this guide, links to our support pages and video tutorials.

Keeping Reality updated

Reality 2.0 includes a new automatic updater. Whenever you start Studio, and if you are connected to the Internet, Reality will check if there are updates. If a newer version is found, a small window will be shown giving you the list of changes for the new build and the opportunity to upgrade. By clicking on the "Upgrade..." button your Web browser will be called and passed a link for the download of the update. You will then just need to run the Reality installer as usual. In this way you can stay on top of the latest updates, receive bug fixes and improvements. It's easy, fast and convenient.

For your peace of mind we'd like to disclose what happens during this process. Whenever a program connects to the Net you have the right to know what kind of connection is established. Reality sends a single, non-blocking, very fast request to the update server. This request doesn't block Studio and takes probably just a few milliseconds to be executed. If the computer doesn't have an Internet connection, for example when you are traveling with a laptop, the process simply aborts silently. There are no failures and no problems.

What Reality sends to the update server is simply its own build number plus the Studio version (3 or 4) and the OS type (Mac OS or Windows). These few details are necessary to identify the exact version to download in case there is an update. Lastly Reality sends the order number and serial number used for the registration in order to identify this as a legitimate request.

No personal information is sent. Not even an email address. Your confidentiality is completely assured and preserved.

We hope that, in honestly disclosing these details, we can provide the excellent service that you require with the protection of privacy that you deserve.

Best practices for exporting your scene

Hide what you don't see

LuxRender performs very demanding operations because it uses real-world mathematical models for calculating the effects of light and the behavior of materials. While often it's a good idea to render off-camera objects, many times you will save a considerable amount of time and memory by not exporting part of your scene.

For example, many scenes that use the DAZ human figures, like Victoria or Michael, pose the models with their mouths closed. In that situation hiding the teeth, tongue and gums saves you hundreds of polygons during the export, polygons that will not be rendered but that will require memory for the geometry, the materials, the UV maps, etc. Save yourself some render time by hiding them before rendering.

Start simple

After you use Reality and Lux for a while you will get a sense of how things work and what to expect. Until you get there, start simple. Load a simple scene, a few props, and a human figure. Add one light only. Render the scene at a low resolution, something around 800x450. This will likely give you a very quick result and a way of checking that your system works with Reality and Lux. From there increase the complexity gradually. In a short amount of time you will develop the skills to recognize the different scenarios and how to optimize Reality for them

Fog

You can create dramatic volumetric light effects by using Fog. Reality includes a Fog Prop to make this operation simple to configure.

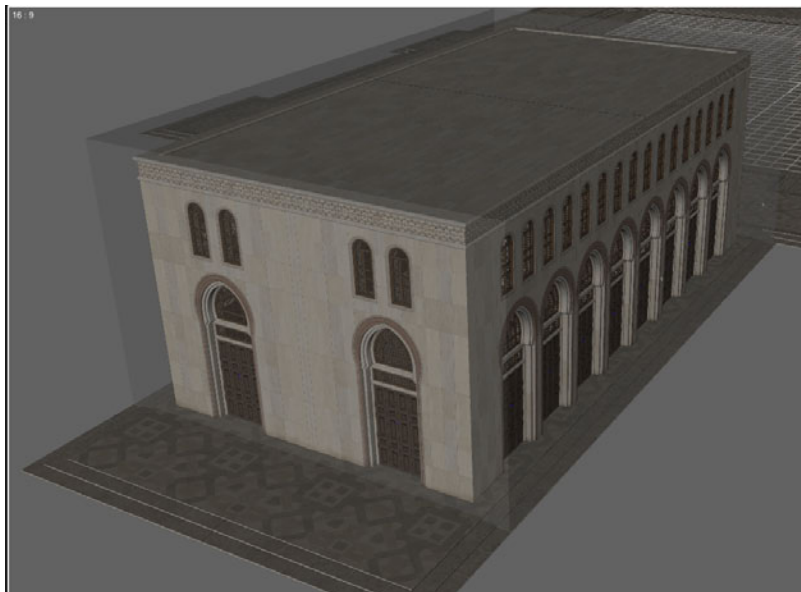
- Navigate in the Studio content manager and find the Reality props folder.
- Add the Fog prop to your scene.
- Scale the fog cube to completely enclose your scene.
- Add one or more spotlights pointing from outside the fog cube toward a spot that is visible with the render camera. Set the spotlights to be very narrow, typically between 5 to 8 degrees, unless the spotlight goes through narrow openings in the scene, like windows or cracks in the walls.
- Render.

You should see beautiful volumetric light in your scene.

The following scene has been created with the Fog cube enclosing the ! Pre_PR_High_N building from the “Kingdom of Marrakesh” set and then shining two small mesh light from outside, pointing inside:



This is a screengrab from the Perspective camera looking at the scene from outside:



You can see the Fog cube surrounding the building.

The following parameters are used for the fog position and size:

- Y: 469.55
- Z: -4339.52
- Scale: 505.9
- X Scale: 302.5
- Y Scale: 192.9
- Z Scale: 501.2

The fog material was set as:

- Amount: 5, 5, 5
- Rays: 20, 20, 20

The two mesh lights were set with the following parameters:

Light 1:

- X: -1032.54
- Y: 604.07
- Z: -4226.57
- X Rot: 175.65
- Y Rot: -1.99
- Z Rot: 65.97
- Scale: 10%
- Power: 500W

Light 2:

- X: -967.01
- Y: 604.66
- Z: -5010.04
- X Rot: 175.65
- Y Rot: -1.99
- Z Rot: 65.97
- Scale: 10%

- Power: 100W

The Tonemapper in Lux was set to “Linear” with exposure parameters sets to:

- Sensitivity: 640
- Shutter: 1/4
- F/Stop: 2.0

The color temperature for the lights was set to 8893.

Saving your Reality settings

Reality saves all your material configurations and other settings inside your Studio scene automatically. Once you save the Studio file your Reality configuration is saved.

Tonemapping

After you play with Lux a little bit you will notice some controls in the main render window. The topmost is called “Tonemapping.” Tonemapping is very important in making things work the way you want, and in achieving the look that you desire.

Unlike 3Dlight and many other renderers, Lux calculates the light values using full spectra. This means that when it calculates the light amount for a given pixel Lux has the real value and not some number that has been clamped down to stay in the RGB limits of what our monitors can show. Light in the real world has wavelengths that go beyond what the human eye can see. Photographic film has a more limited ability to capture light values than what our eyes can see. Computer monitors have an even more limited range of light values that they can display. If Lux was going to render a light value that was based on what your monitor can “see” then the fidelity of the information would be seriously compromised. The tonemapper is an algorithm that converts the internal light values to what we can see on the computer. As it turns out there are a few different possibilities, but for Reality we will focus on just two: Reinhard and Linear.

Reinhard

Think of your scene as something that you can capture with a photographic camera. As you decide what camera you should use you have two choices: point-and-shoot “all automatic” or state-of-the-art DSLR (Digital Single-Lens Reflex) with full manual control.

Point-and-shoot cameras nowadays are pretty good and deliver nice pictures. You point the camera at the scene, click the button, and you get a nice picture that is correctly exposed, in focus, and with good colors. However, if you tried to take a picture of somebody in front of a window you know that there are limitations to what those cameras can do. Nevertheless they get the job done in a very good way and without your involvement in things like shutter speed, film ISO and aperture.

Reinhard is your point-and-shoot camera. It works well for many situations, including scenes that have sun and spotlights, but it averages the light intensity in the scene in order to find a good balance, and consequently things are not always exposed correctly.

While you might be tempted to use Reinhard all the time, when using mesh lights Lux needs to use the Linear tonemapper. Reality detects this situation automatically and will set the scene to use Linear.

Linear

Linear is your most realistic tonemapper and it gives you full control over the final aspect of the image. You can dynamically change the exposure to achieve overexposed/underexposed effects without re-rendering your scene, and the result is usually shown in a second or less.

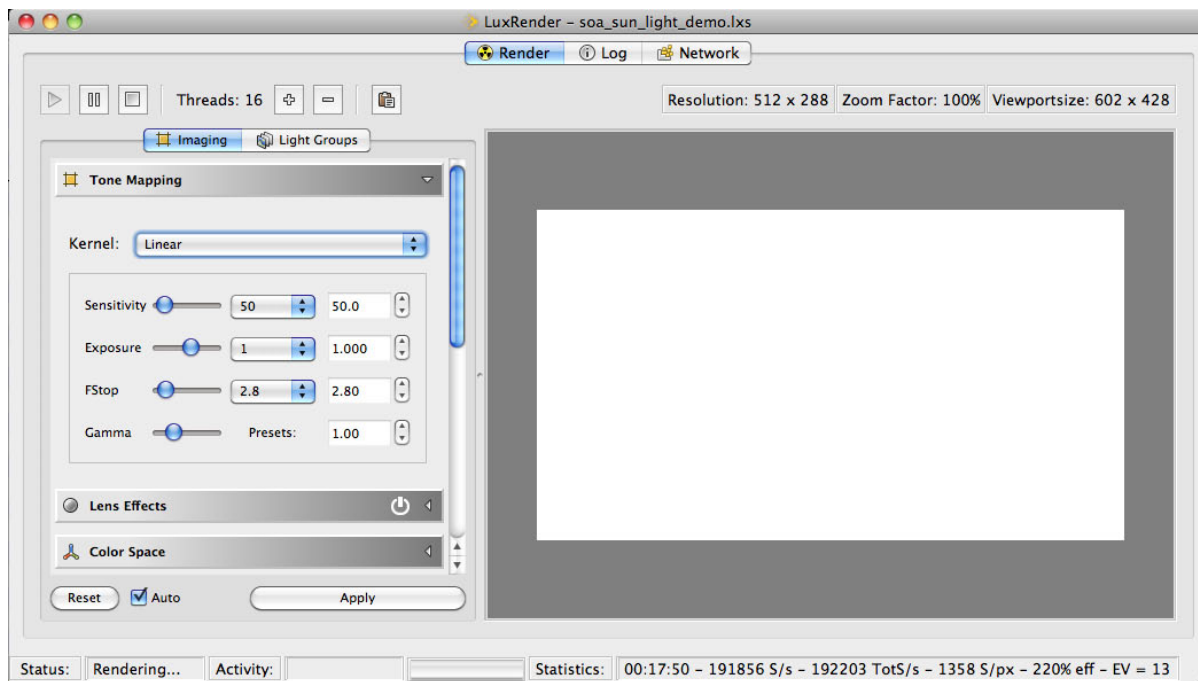
Linear works like a manual camera. The parameters to adjust are:

- **Sensitivity.** This is also known as film speed. Higher values mean faster film, faster film means that the “film” is more sensitive to light. If your scene looks dark then change this parameter first and see if you gain some definition. Lux sets Sensitivity to 50 by default while Reality changes it to 200 when using mesh lights.

- **Exposure.** This is equivalent to the speed of the “shutter” of a camera, and is usually expressed in a fraction of a second. 1/60, 1/125 or 1/250 of a second are normal values. For a night scene with very faint light you might want to try an exposure of 1 full second.
- **f/stop.** This is the aperture of the “iris” in your camera. Lower numbers mean wider aperture and more light hitting the “film.” Generally each number is half the light of the previous one. For example f4.0 gives you half the light of f2.8.

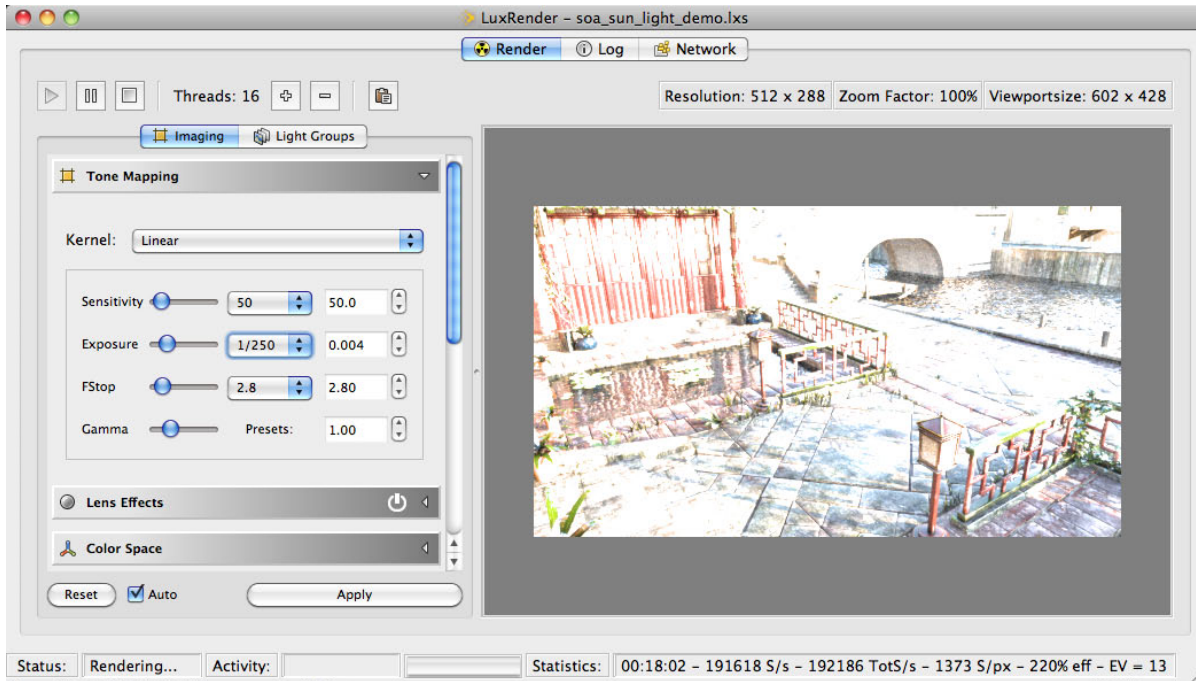
Converting from Reinhard to Linear

When you change a tonemapper from Reinhard to Linear, it can sometimes seem that you have done something wrong. Where did the image go?

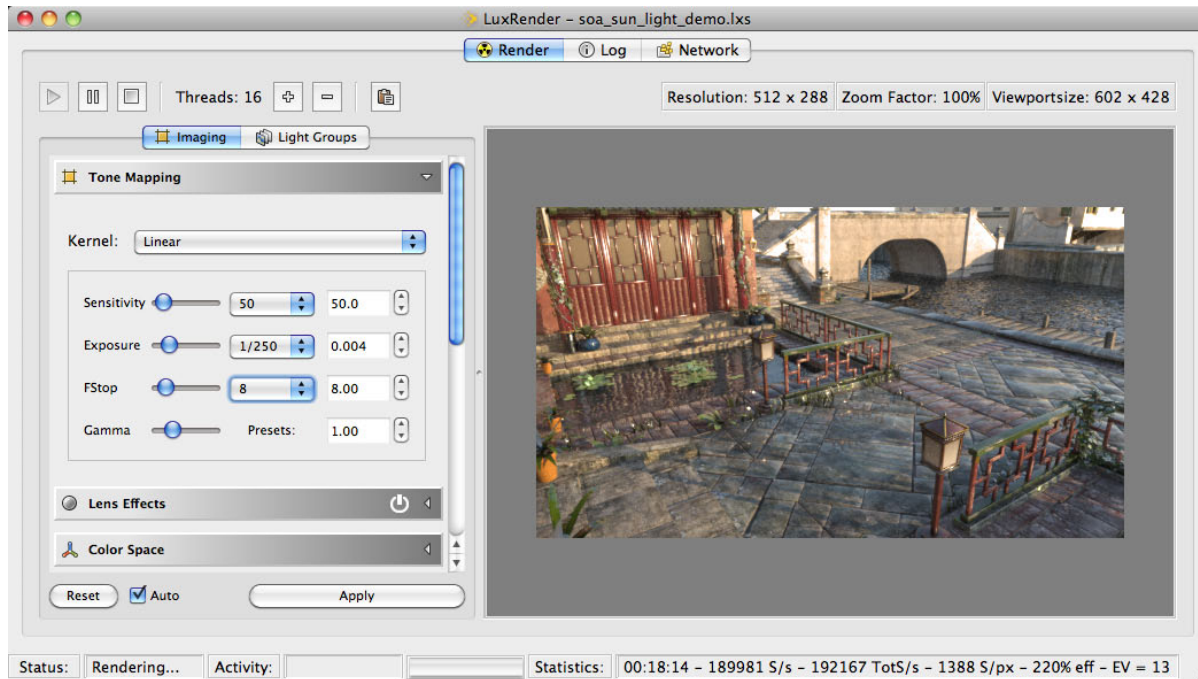


If you look at the parameters, in this example, you’ll see that we have a full exposure of 1 second and the iris wide open at f2.8. The scene that I’m trying to render is a daylight scene and if you used that kind of aperture and exposure with a real camera you would end up with something overexposed like that as well.

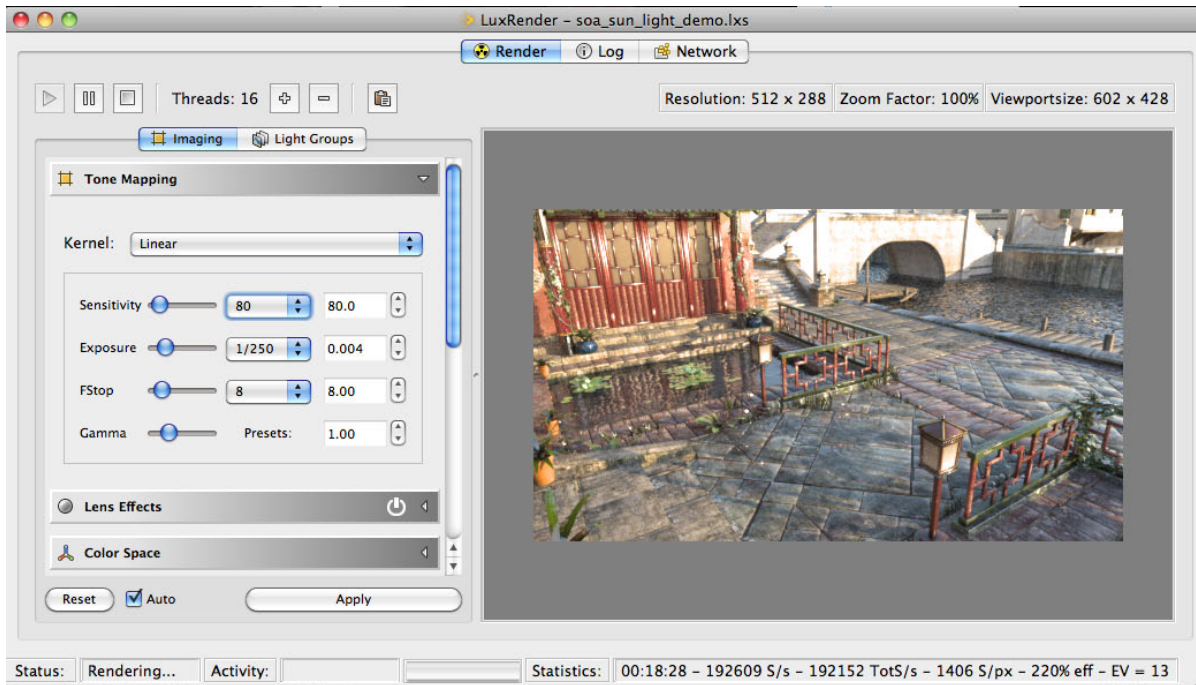
So, lets first change the exposure to something more reasonable like 1/250 of a



second. Even with the aperture at f2.8 we can immediately see the difference:
OK, we are on the right path. Next let's change the aperture (f/stop) to something a little more appropriate:



At f8.0 we have $1/8$ of the light that we were getting before and now almost everything is exposed correctly. Just for the heck of it, let's see what happens when we increase the sensitivity to 80:



That looks good to me. Even though a couple of spots might be a tad overexposed, I can adjust the textures to compensate for that.

Render as Statue

Next to the “Export Only” checkbox there is the “Render as Statue” option. This checkbox renders your scene by using a single material and ignoring transparency maps, as if the whole scene was carved out of Carrara marble. It can generate renders that are just fun to see or that can be used for extraction of mask. It depends on the subject and the light. Try it, it’s fun!

Stopping and resuming renders

Lux allows you to stop and resume your renders at any time. This is an optional feature which is disabled by default in Lux but Reality enables it for all scenes that it generates. The feature is based on the creation of a “film” file, a file that keeps track

of the status of Lux at any point in the process. The file has the same name of your image file but with the extension “.flm”.

You can stop a Reality-created render at any time, and close Lux.

To restart the render from where it stopped:

- Launch Lux directly, outside Reality.
- Select the **F i l e / R e s u m e F L M** menu option.
- Lux will first ask you for the .lxs file. Select the scene file.
- Lux then will ask you for the .flm file, and again make your selection.
- In a few seconds you will see the render resume from where it was stopped and continue the refining process.

Known issues

There is really one single known issue for Reality 2.0 and it relates to an undocumented feature of Studio 3. Switching to Studio 4 resolves the issue completely. Here is the original note from Reality 1.2:

- **Converted ACSEL shaders (from DS3) don't load in Reality for DS4.** DS4 seems to be handling some of the internal data about models in a different way than DS3. As a result some models receive an internal name that is different from what DS3 used. Reality uses the model's internal name to match the ACSEL shaders and so it can happen that a shader that you converted from DS3 does not load when using the same exact model in DS4. We are investigating this issue and we hope to provide a solution via the auto-updater.
- **SubD figures in DS3.** SubD in DAZ Studio 3 is a feature that is mostly undocumented in Studio, a situation that makes it hard to support it in plugins. Regardless, it is a way of achieving smoother surfaces that is a bit of an overkill. When you use “Edit/Convert to SubD...” in Studio your figure’s geometry is altered from the original. The whole figure. This means that Victoria 4, for example, quadruples the amount of polygons in the figure, from around 70,000 to around 280,000. As you can imagine, that many polygons can pose a serious load on a rendering software.

If you don't need this level of detail, the standard V4 and M4 figures have plenty of resolution, so it would be a waste to use a subdivided figure. It would at least quadruple the memory requirement for the render and at least quadruple the rendering time.

The truth is that many times you don't need to subdivide the whole figure but only a few parts that are closer to the camera. Reality provides a more efficient solution by allowing you to subdivide a material, instead of acting on the whole figure. Also, this subdivision only happens during the export to Lux, leaving your original V4 in Studio unchanged and much lighter to move in the scene. It's a great solution.

•

Remember that you can set V4 or other figures to be SubDivided in the Power Loader. If you find that your model is subdivided even though you didn't use the SubD option in the edit menu, look in the defaults for the Power Loader and disable SubDivision there. To do this:

- ▶ Press and hold the Cmd (Mac) or Ctrl (Windows) key on your keyboard while double clicking on the models icon, for example V4.
- ▶ You will see the PowerLoader dialog box on the screen. Click on the "Advanced" tab.
- ▶ Click on the "Mesh Resolution" tab and uncheck the "Convert to Sub-Division Surface" checkbox.
- ▶ Click on the Accept button.

If you try to render a scene with a SubD figure (from Studio) Reality will show you a warning about it and will stop the render. You can then fix the issue and re-start the render.

Studio 4's SubD feature is completely supported by Reality 2.0.

Acknowledgements

I personally want to acknowledge and thank all the developers and dedicated individuals who made LuxRender a reality, no pun intended. LuxRender is a real gem of a program. Through the genius and dedication of the developers we now have a state-of-the-art rendering engine that is available to everybody, on all major Operating Systems, and that has been designed to last a long time. If you can, please go to <http://www.luxrender.net> and thank the generous people who made this possible.

I'd like also to thank the invaluable beta testers who patiently put up with the lack of documentation and with the lack of a proper installer for the beta. They did an astounding job at testing every part of Reality with as many models as possible and provided great feedback throughout the process. A big thanks to: Randy Anderson, Mario Bush, Charley S., Robert Dick, Matthew Doucette, Jeffrey Felt, Paul Fields, Martin J. Frost, Christopher Jones, Kevin James, Alex Lewis, Roman Pittroff, , John Reilly, Emily Smirle, Jack Tomalin, Anton Van Tonder. If I forgot anybody, my deepest apologies.

Lastly I want to thank my wife Holly, for her saint-like patience in reviewing and editing this manual ... again. This new revision added more than 40 pages and it is probably the largest manual for any Studio plug-in, If this guide is at all readable it's all because of her attention, amazing editorial skills, and great wisdom.

Stay updated with Reality!

Be sure to get the latest about Reality by visiting our website at <http://www.preta3d.com>

There you'll find news, tips on how to get the most out of your scenes, video tutorials and updates to this documentation.

For the latest about Reality and upcoming products be sure to "Follow," "Like" or "Circle" us on:

- Twitter: <http://twitter.com/preta3d>

- Facebook: <http://www.facebook.com/pages/Reality-Plugin/143936782331542>
- Google Plus: <https://plus.google.com/10662581615330416311>

All content in this document is copyright © 2010-2011 by Pret-A-3D. Reality plug-in, Reality Engine and ACSEL (Automatic Custom ShadEr Loader) are all copyright © 2010-2011 Pret-A-3D. All rights reserved.

Reality: See the light!

Appendix A – Converting the ACSEL Database

If you used Reality 1.0, 1.2 or 1.25 you must convert the ACSEL database because its format has changed. This operation needs to be performed only once but it needs to be done for each installation of DAZ Studio. In other words, if you use both DS3 and DS4 you need to run the conversion utility (described below) once for DS3 and once for DS4 because each version of Studio keeps a separate copy of the the ACSEL Database.

To do the conversion you simply need to use a supplied script named:

```
convertACSELDatabase.dse
```

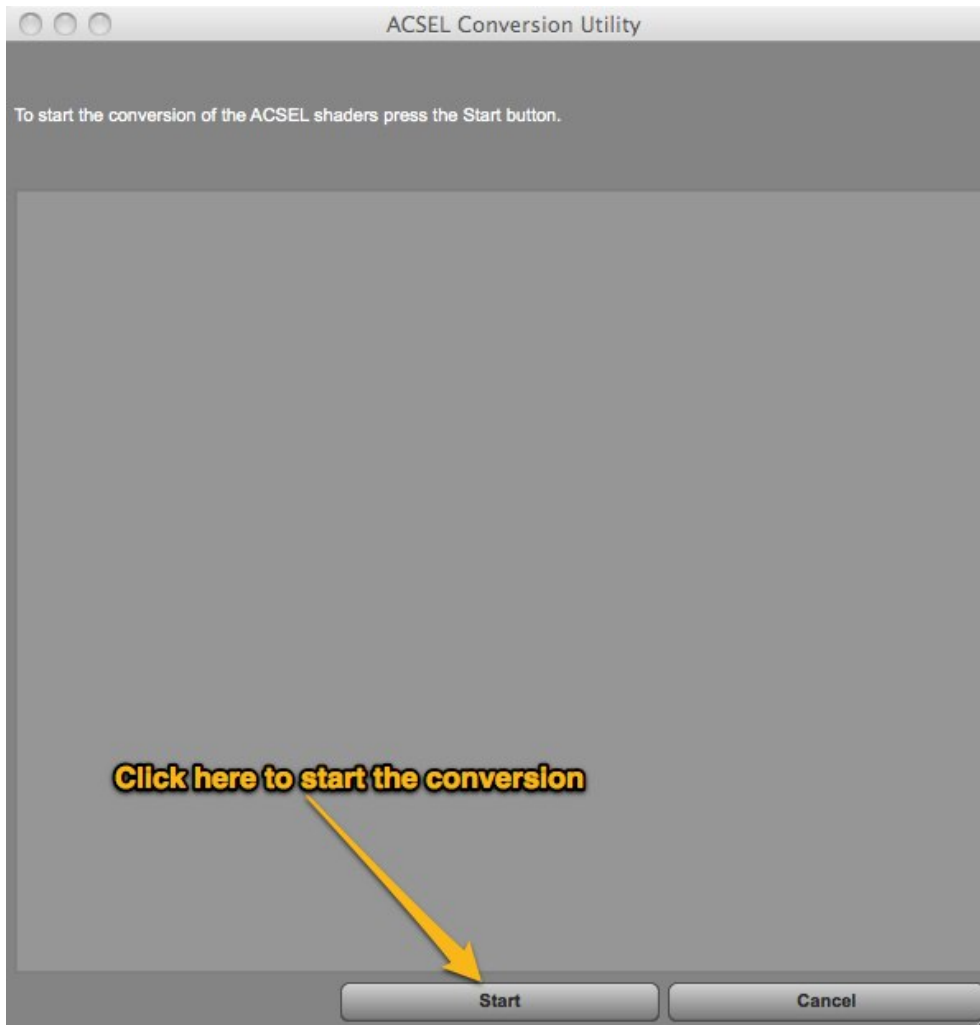
This script is located in the scripts/Reality folder under Studio. So, for example, for an installation of Studio 4 on 64-bit Windows it will be in:

```
C:\Program Files\DAZ 3D\DAZStudio4\scripts\Reality
```

On Mac OS it will be in:

```
/Applications/DAZ 3D/DAZStudio4/scripts/Reality
```

To execute the script simply click on the File/Merge menu of Studio and navigate to where the script is. Select the script and click “OK.” The script window will appear:



The conversion might take a few minutes, depending on how many ACSEL shaders you have. After that you can simply close the window.

Appendix B - How to use the Softbox

See the videos on how to use this light at:

http://www.youtube.com/watch?v=IVch_4d8C3w

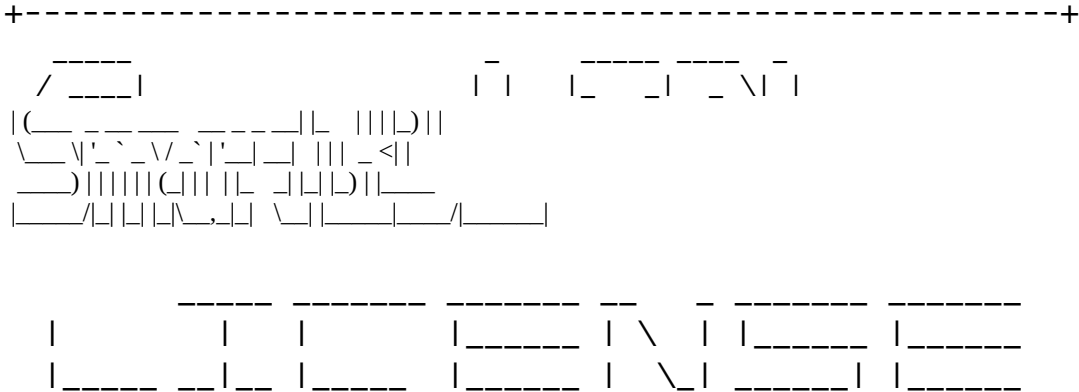
<http://www.youtube.com/watch?v=v0R8lGZVQmE>

Appendix C - Network rendering

With LuxRender you have an unlimited license to use as many CPUs and as many render nodes you want. Network rendering is accomplished by running the command-line version of Lux on each render node and then connecting the main render process to the nodes. Render nodes can be a mixture of Mac OS, Windows and Linux machines.

- Install exactly the same version of LuxRender on every machine that will participate in the render process. The exact version means the revision number; you can mix Mac OS, Windows and Linux nodes without issue. For the sake of this tutorial we assume Luxrender 0.8RC1
- Start the command-line processor on each node. On Mac OS you need to use Terminal.app, located in / Applications/Utilities. On Linux you can use any of the many terminal emulators available. On Windows you need to start cmd.exe.
- Run the luxconsole command in server mode by typing the command:
luxconsole -s
- Take note of the IP address of the render node. For this example we assume that it is 192.168.0.2.
- Go back to your machine where you have Reality installed, and start the render as usual.
- When Lux starts rendering your scene click on the Network tab and add 192.168.0.2 in the Server input field. Repeat the process for every render node that you have.
- Lux will seem “frozen” while it sends the scene and all the textures to the render node. If you look at the render node screen you will see the activity about the sent files. After a while the nodes will start sending samples and contributing to the render process.

Appendix D - sIBL License



These sIBL-sets and all images therein are licensed under the Creative Commons Attribution 3.0 License.

You are free:

- to Share: to copy, distribute and transmit the work
- to Remix: to adapt the work

Under the following condition:

- Attribution: You must attribute the work as "Lit using Smart IBL from HDR Labs.com/sibl".

Further license details here:

<http://creativecommons.org/licenses/by/3.0/>

We are actively looking for volunteer developers, who have the time and abilities to adapt the Smart IBL plugins to Blender, Houdini, Rhino... Join the club!

+-----+

Get more sIBL sets like these at:
www.hdrlabs.com/sibl/archive.html

+-----+

If you want to learn how to shoot your own HDRI panos
you should get my book:

www.hdri-handbook.com

Christian Bloch
www.HDR Labs.com

Appendix E - Reality plug-in software license

END-USER SOFTWARE LICENSE AGREEMENT

The Pret-a-3D Software delivered is intended for individual personal use only. Redistribution is not allowed.

1. GRANT OF LICENSE:

Copy and Use - Application Software. Pret-a3D grants to Recipient a limited, non-exclusive License to install, use, access, display, run, or otherwise interact with ("RUN") this copy of Reality plug-in (hereinafter the "SOFTWARE PRODUCT"), or any prior version, on a single computer, workstation, terminal, handheld PC, or other digital electronic device ("COMPUTER"). The primary user of the COMPUTER on which the SOFTWARE PRODUCT is installed may make copies for his or her exclusive use on other computers.

Pret-a-3D retain all ownership and title to the Software at all times.

Support and Updates. This License does not grant Recipient any right to technical or Software product support of any kind, bug fixes, or any enhancement or update, by Pret-a-3D.

This License applies to any updates or extensions of time-protected-software which may be supplied to Recipient by Pret-a-3D or any of its distributors.

Redistribution and Content Use. This License prohibits the Recipient from redistributing of any part of the Software Product, including images, models, applets, documents, icons and programs, both for commercial or non-commercial purposes.

This License grants to the Recipient the right to use images and models created through using the Software Product for either commercial or non-commercial purposes. At no time may the Recipient redistribute images or 3D models that are delivered with the Software Product. The Recipient may not re-license or re-distribute the images, models or clip art that are delivered with the Software Product without first receiving written permission from Pret-a-3D. For the purposes of this agreement, e-mail is not considered a valid form of written communication.

2. RESTRICTIONS:

Copyright, Title, ownership rights, and intellectual property rights in and for the Software Product shall remain with Pret-a-3D and its suppliers. The Software Product is protected by the copyright laws of the United States and international copyright treaties. Title, ownership rights, and intellectual property rights in and to the content accessed through the Software Product is the property of the applicable content owner and may be protected by applicable copyright or other law. This License does not provide any rights to Recipient for such content.

Recipient must maintain all copyright notices on all copies of the Software Product.

Recipient shall not attempt to reverse-engineer, decompile or take actions to disassemble the Software Product.

3. DISCLAIMER OF WARRANTY:

The Software Product is provided on an "AS IS" basis, without warranty of any kind, including without limitation the warranties of merchantability, fitness for a particular purpose and non-infringement. The entire risk as to the quality and performance of the Software Product is borne by Recipient.

Should the Software Product prove defective, Recipient, and not Pret-a-3D, assumes the entire cost of any service and repair. In addition, the security mechanism implemented by the Software Product has inherent limitations, and Recipient must determine that the Software Product sufficiently meets Recipient requirements. This disclaimer of warranty constitutes an essential part of the agreement.

SOME STATES DO NOT ALLOW EXCLUSIONS OF AN IMPLIED WARRANTY, SO THIS DISCLAIMER MAY NOT APPLY TO Recipient AND Recipient MAY HAVE OTHER LEGAL RIGHTS THAT VARY FROM STATE TO STATE OR BY JURISDICTION.

4. LIMITATION OF LIABILITY:

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, TORT, CONTRACT, OR OTHERWISE, SHALL Pret-a-3D OR ITS SUPPLIERS OR RESELLERS BE LIABLE TO Recipient OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES. IN NO EVENT WILL Pret-a-3D BE LIABLE FOR ANY DAMAGES IN EXCESS OF Pret-a-3D'S LIST PRICE FOR A License TO THE SOFTWARE PRODUCT, EVEN IF Pret-a-3D SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. FURTHERMORE, SOME

STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION AND EXCLUSION MAY NOT APPLY TO RECIPIENT.

5. TERMINATION:

This License will terminate automatically if Recipient fails to comply with the limitations described within this Agreement. On termination, Recipient must remove the Software Product from all computers onto which it is installed, and must destroy or return all copies of the Software Product.

6. EXPORT CONTROLS:

None of the Software Product or underlying information or technology may be downloaded or otherwise exported or re-exported (i) into (or to a national or resident of) Cuba, Iraq, Libya, Yugoslavia (Serbia and Montenegro), North Korea, Iran, Syria or any other country to which the U.S. has embargoed goods; or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nationals or the U.S. Commerce Department's Table of Deny Orders. By downloading or using the Software Product, Recipient agrees to the foregoing and Recipient is representing and warranting that Recipient is not located in, under the control of, or a national or resident of any such country or on any such list.

7. MISCELLANEOUS:

This Agreement represents the complete agreement concerning this License between the parties and supersedes all prior agreements and representations between them. It may be amended only by a writing executed by both parties. If any provision of this Agreement is held to be unenforceable for any reason, such provision shall be reformed only to the extent necessary to make it enforceable.

This Agreement shall be governed by and construed under California law as such law applies to agreements between California residents entered into and to be performed within California, except as governed by Federal law. The application the United Nations Convention of Contracts for the International Sale of Goods is expressly excluded.

NOTWITHSTANDING ANYTHING TO THE CONTRARY IN THE GENERAL LICENSE AGREEMENT, THE USER MAY NOT TRANSFER OR ASSIGN THE PRODUCT OR ANY COPY THEREOF, OR THE USER'S LICENSE TO USE THE PRODUCT TO A THIRD PARTY.

Manufacturer is Pret-a-3D, Aptos, CA 95003 USA.

<http://www.preta3d.com>, reality@pret3d.com

© Copyright 2010-2011 Pret-a-3D All rights reserved.