



MAXWELL RENDER

Maxwell for Rhino

Plugin Manual

Maxwell for Rhino

The Maxwell for Rhino plugin has been designed to make using Maxwell with Rhino as seamless and efficient a process as possible. To this end, the plugin provides fully-integrated Maxwell materials, along with a native-to-Rhino Material Editor which is built to work in a way which is very similar to Maxwell's own MXED MXM editor.

Supported Platforms

The Maxwell for Rhino plugin runs on all versions of Rhino equal or greater than Rhino 4 SR5. This includes Rhino 5 WIP releases, both x86 and x64. Two versions of the installer are available; *Maxwell for Rhino x86* and *Maxwell for Rhino x64*, and the architecture here refers to the target version of Rhino, not the operating system architecture. Therefore, Rhino 4 and x86 versions of Rhino 5 should use the x86 plugin installer, while x64 versions of Rhino 5 should use the x64 installer. This version of the plugin is designed for use with Maxwell Render 2.6.0.

Installation

Installation of the plugin is accomplished as follows:

1. Download and unzip the plugin. When doing so, you should not use any special 3rd-party downloaders, and you should disable any anti-virus software which may interfere with the download. Once downloaded, the standard Windows unzip function should be used to unzip the downloaded folder to a location at which you hold permission to install programs. Using 3rd-party unzip software or having anti-virus software running during download, unzip, or installation may corrupt the installer files and cause a failure to install correctly.
2. Open the unzipped folder and locate the file *setup.exe*; double click this file to start the installer (do not use the file with the .msi extension to install; it will be called by *setup.exe* after certain preliminary checks have been performed). Follow through the installation wizard until the plugin has been installed. Rhino should not be running when you do this.
3. When you start Rhino next time, you should find a *Maxwell for Rhino* item in the *Render > Current Renderer* menu. Choosing this item will load the Maxwell plugin and set it as the current renderer. If this item does not appear in the Current Renderer menu, you should attempt to load the plugin manually using *File > Properties > Plug-ins > Install* and locating the file *Maxwell.Rhino.rhp*, which will be located at the location you specified during installation. The folder containing this file should be located somewhere similar to *C:\Program Files (x86)\MaxwellDotNET\Maxwell.Rhino*, unless you specified a custom installation location when installing.

The installer requires a minimum of Windows Installer version 3.1, and the plugin itself requires the Microsoft .NET Framework 2.0. The installer, *setup.exe*, will check for these prerequisites and prompt you to download and install them if required. Additionally, the installer will install the Microsoft Visual Studio 2008 SP1 C++ Runtime redistributable. The file which is used for this, *vcredist_x86.exe* (*vcredist_x64.exe* for the x64 plugin), is included in the download, and in the event of a failure to install automatically, you should try executing this file manually and then running the installer again.

What is covered in this plugin manual?

This manual includes all the information you need to know to use the Maxwell for Rhino plugin. It goes into the details of some general Maxwell concepts, but it is not intended to replace the regular Maxwell Render manual, which may be found in your Maxwell installation directory. If any non-plugin-related parameters are not explained sufficiently here, please consult the regular Maxwell Render manual for further direction.

Special Note

This version incorporates an updated user-preference system. By default, it will have a color scheme similar to that of Rhino. If you prefer the the previous color scheme, just go to *Scene Manager > Options* and click the *Load UI Colors File* button; a dialog will be opened, in which you should find a file named *classic.xml* -- this is the default color scheme used in previous plugin versions.

Table of Contents

topic	page
What do I need to know to use the plugin?	4
Plugin Quick-Start	5
Frequently Asked Questions.....	13
Main Menu.....	18
Toolbars.....	20
UI Controls.....	24
Scene Manager.....	26
Materials.....	30
MXM Gallery Search	31
Camera	32
Location & Time.....	36
Environment.....	38
Output	43
Options.....	50
Materials.....	52
Material Editor	55
Database Manager.....	66
Object Properties.....	69
Maxwell Fire	70
Tutorials	
Real Scale.....	72
Motion Blur.....	80
HDRI.....	84
Maxwell Script for .NET Plugins.....	89
RhinoScript.....	97
EV Chart.....	101

What do I need to know to use the plugin?

The Maxwell for Rhino plugin has been designed to work in as simple a manner as possible. Wherever appropriate, unnecessary items have been optimized away, leaving what is intended to be a very simple and straightforward, yet complete, interface to Maxwell from within Rhino. Most of the functionality contained in the plugin is accessed via one of several plugin windows, all of which may be opened or closed using the Maxwell entry in Rhino's main menu, or by using the buttons located in the plugin's *Tools* toolbar:

Scene Manager

The Scene Manager window holds the parameters used to define the Maxwell scene; that is, it contains the values which will be written into the MXS files the plugin produces. It also contains parameters used to determine how and where the scene is to be written. The Scene Manager may be used either as a floating window, or as a Rhino dockbar using the plugin option *Dockable Scene Manager*.

Database Manager

The Database Manager window provides an MXM Browser, as well as several object-specific catalogs for storing individual material components (i.e. BSDF, Emitter, etc.). You can drag MXM files into the scene, drag materials from the scene into the MXM Browser to save them as MXM files, or drag components back and forth between the Material Editor's layers tree and the applicable pages in the Database Manager to build new materials using saved components. The Database Manager may also be used either as a floating window, or as a Rhino Dockbar using the plugin option *Dockable Database Manager*.

Material Editor

The plugin implements a native-to-Rhino Material Editor which is designed to work in a way very similar to Maxwell's MXED MXM editor.

Camera LCD

This tool is provided for convenience in managing the most common camera parameters; it is made to resemble the LCD back-panel of a digital camera.

Log Viewer

As you use the plugin, it keeps track of various actions by printing messages to this window. Some of these messages can be useful for diagnosing any possible issues you may encounter. Should you run into an issue you can't resolve, you can email a copy of the current log directly to support using the Log Viewer's right-click menu.

Help Viewer

If you open the Help Viewer, the cursor will change to a 'help' cursor when the mouse hovers over many of the parameter-labels in the various plugin windows. Clicking (left or right) on any of these parameters will show help specific to the selected parameter in the Help Viewer. The Help Viewer may be used either as a floating window or a Rhino Dockbar using the plugin option *Dockable Help Viewer*.

Maxwell Toolbars & Commands

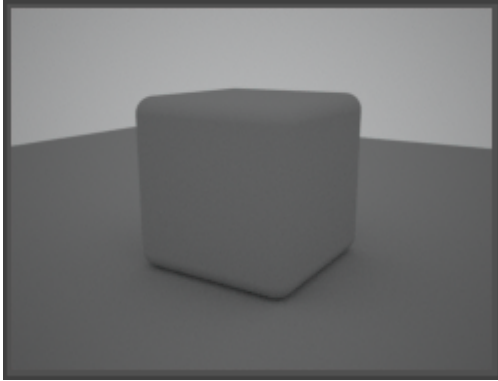
In addition to the various windows mentioned above, the plugin also provides several toolbars and commands. The toolbars may be shown by clicking *main menu > Maxwell > Plugin Windows > Show All Toolbars*. If you have modified the toolbars and wish to return them to stock configuration, you can use the command *Maxwell_ToolbarReset*. In addition to the commands used by the plugin's toolbars, there are several other Maxwell-specific commands provided and you can get a complete list of these by typing *Maxwell_GetCommandList* into the Rhino command-line. If you show the built-in Rhino help-viewer (use the command *CommandHelp*) then context-sensitive help will be shown for the plugin's commands when they are invoked.

Maxwell for Rhino :: Plugin Quick-Start

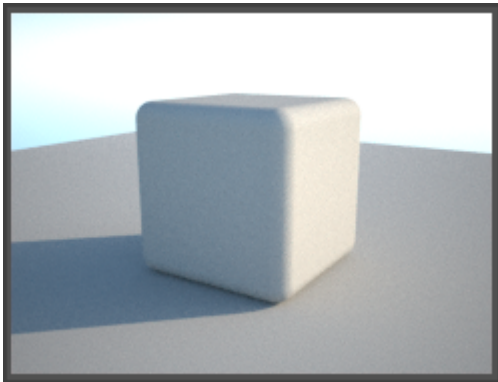
It is possible to use the plugin without assigning explicit Maxwell materials; the plugin will automatically convert native Rhino materials into Maxwell materials at export time, if there are no explicit Maxwell materials assigned. This is not recommended, of course, as automatically-created materials are not likely to look as good or perform as well as custom Maxwell materials.

Even if no Maxwell materials are assigned, it will probably be necessary to adjust the Maxwell camera; to do so, open the Scene Manager and click on the camera tab. The plugin creates and manages one Maxwell camera for each Rhino viewport; at any given time, the camera shown in the Scene Manager (or in the camera LCD) is the one which corresponds to the selected Rhino viewport.

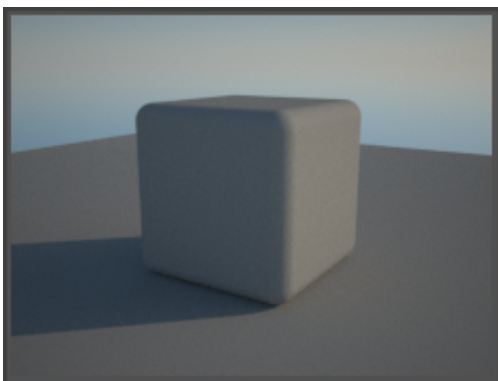
So, to begin with, let's just render a simple one-meter cube using all default values:



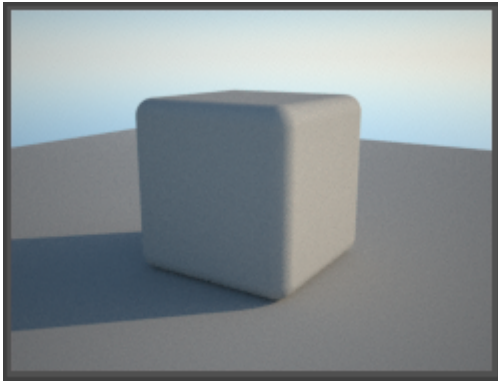
As you can see, the defaults use Maxwell's Sky Dome, since Rhino tends to be a product-development tool. Now let's go to the Environment tab in the Scene Manager, choose Physical Sky, check Use Sun, and render again:



This looks to be a bit bright, so let's switch to the camera tab. Once there, adjust the camera's EV from the default of 13 to 15, since the scene has gotten much brighter with the Sun enabled. Rendering again, we get this result:

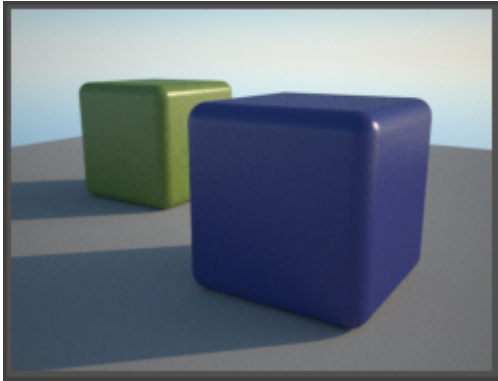


Too dark now, let's try EV=14:

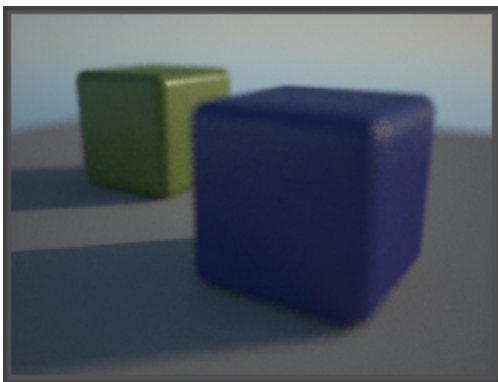


As you can see, the EV parameter makes it very easy to adjust the exposure – higher numbers are required for brighter environments, and vice-versa. There is a chart at the end of this manual with suggested EVs for different scenarios.

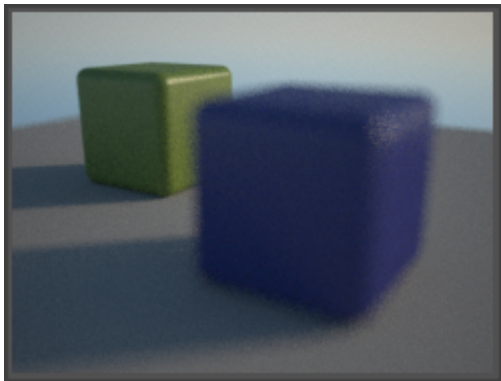
Now, let's apply some Maxwell materials. The easiest way to do this is to open the Database Manager and navigate to the MXM Browser page. By default, the main Maxwell materials database folder should be selected, and you can view the MXM files in each of the directories inside that using the folder tree. To assign MXMs to geometry, simply drag them onto objects in the viewport:



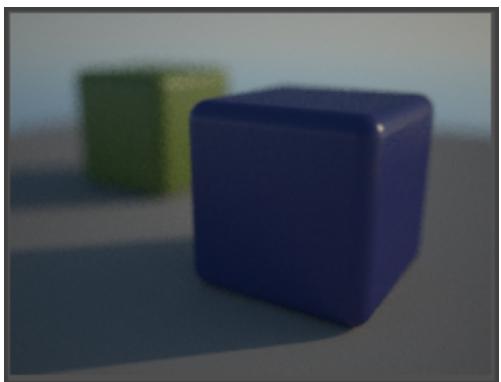
Moving back to the camera side of things, let's create some DOF (depth-of-field) in this image. First, what is needed to do that? Probably, the scene will need to be smaller; change File > Properties > Units to inches (the cubes are one inch in size now). Also, we will want to open up the Maxwell camera's aperture a bit, so change the fStop (it's located directly next to EV in the camera's Optics group) to 4.0. When 'Lock exposure to EV' is checked, it is possible to adjust fStop without also affecting the exposure of the image; the plugin will maintain a consistent EV, regardless of the chosen fStop. Here's what we get after making these changes:



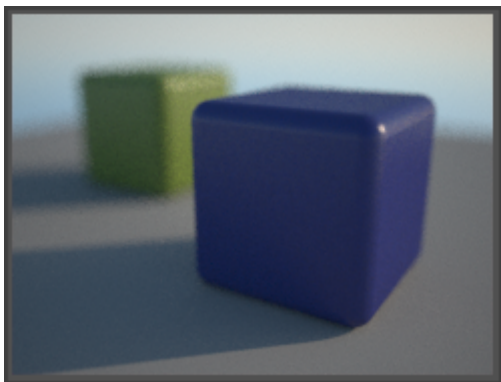
Okay, so we are seeing some DOF here, but unfortunately, none of our subjects is in focus. Let's use a plugin tool to adjust the camera's focal distance: in the plugin's camera toolbar, click on the *Pick camera Focal Distance* button to set the focal point of the camera. First I'll pick a point on the cube in the background:



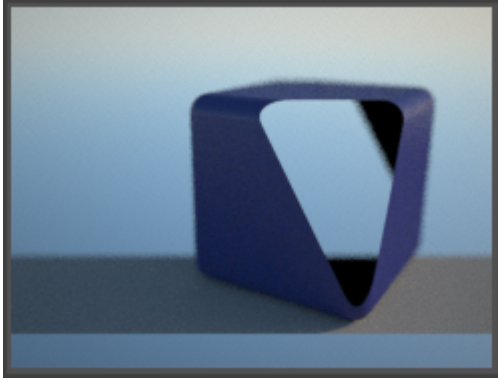
Moving the camera's focus to the cube in the foreground, we get this:



Since we have reduced the size of the scene the exposure has darkened up somewhat; to remedy this, let's adjust the camera's EV back down to 13. So, with EV=13, fStop=4, and the focus on the foreground cube, we end up with this image:



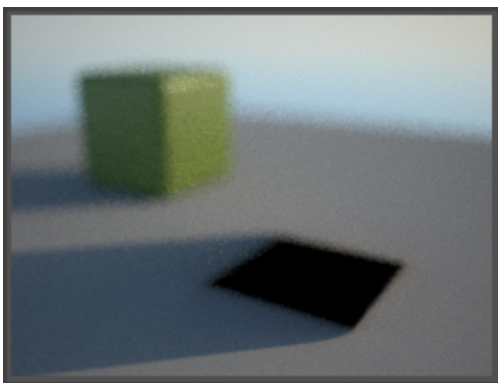
So far, we have only adjusted the camera, changed the render environment, and assigned two materials. Let's try using the camera's Z-Clip planes to section the geometry; here is the result using 0.06m for the near plane and 0.08 for the far plane:



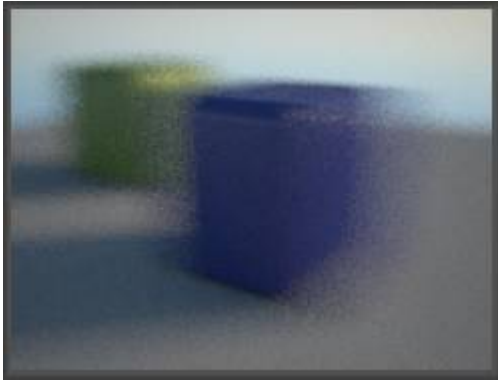
Now, let's try rendering using a Region; click the Region Render button in the plugin's Rendering toolbar and click-drag to select the rendered rectangle:



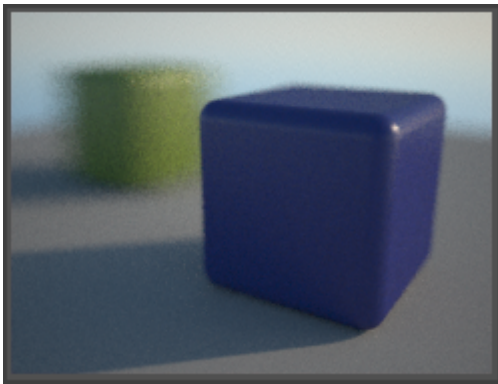
Moving on, let's disable Z-Clipping and try using some of Maxwell's Object Properties flags. Open Rhino's Object Properties window, select the foreground cube, select *Maxwell Object Properties* from the pull-down, and enable the 'Hide from camera' checkbox:



All of the 'Hide from X' settings in the Object Properties group are fairly self-explanatory; just consult the regular Maxwell Render manual for more details on what they are used for. Besides for those properties, the Object Properties window also has options for controlling Motion Blur. There are two types of motion-blur: Linear and Full-Range. For objects which will only move in a linear path, you should choose Linear motion-blur, since it takes much less memory than Full-Range (Full-Range has to store a duplicate set of mesh coordinates for two object-positions, for all objects in question). To create blur, move both cubes to where they would be located in the frame before the one you will be rendering. Then, select the cubes and, in the Object Properties window, enable *Linear Motion Only* and click *Reset Initial Positions* button to store the positions. Then, move the cubes to the position where they should be in the frame you are rendering and click Render:



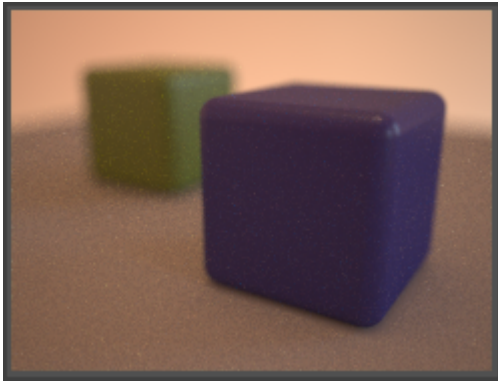
Now, select the foreground cube, in Object Properties, set motion-blur to *None*, and render the scene again:



Now, to switch gears a little, let's change from using Physical Sky and Sun, to using an Image Based Environment. Go to the Scene Manager's Environment tab, un-check the 'Sun' checkbox, and choose Image Based from the Type drop-down. The interface will change to show you settings for Image Based Environment. By default, the scene will have the 'Use Background for all Channels' option enabled – you only need to un-check this if you want to use different HDR images in the Background, Illumination, Reflection, and Refraction Channels. Many times that is not necessary though, and you can just let the plugin copy the settings from the Background Channel into the others at MXS-export time.

So, click on the browse button in the Background Channel group to select a HDR image to use. Once you do, the Background Channel group will expand, showing the Scale, Offset, and Intensity parameters. Usually, you may not need to change these values. What you may want to do though is to click the *Enable Background Channel* button in the plugin's *Environments* toolbar – this will allow you to preview the HDR image in the Rhino viewport.

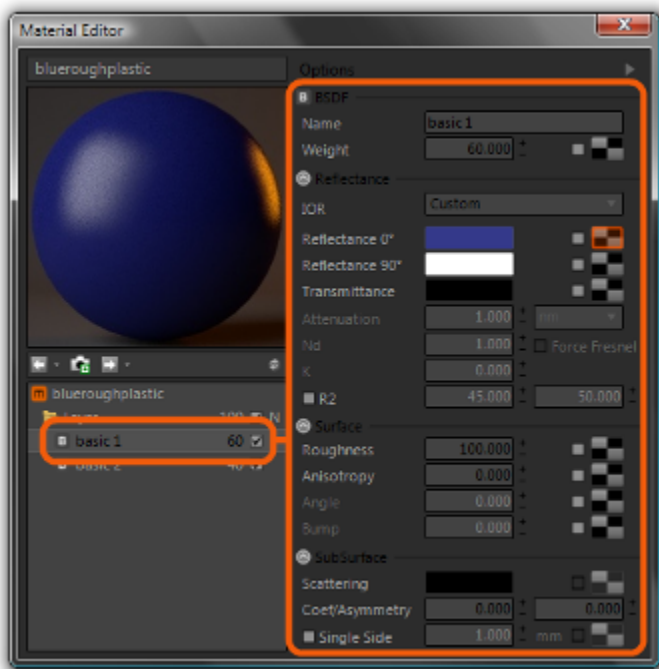
Here is how the scene renders (after disabling motion blur for the background cube) using the Image Based Environment:



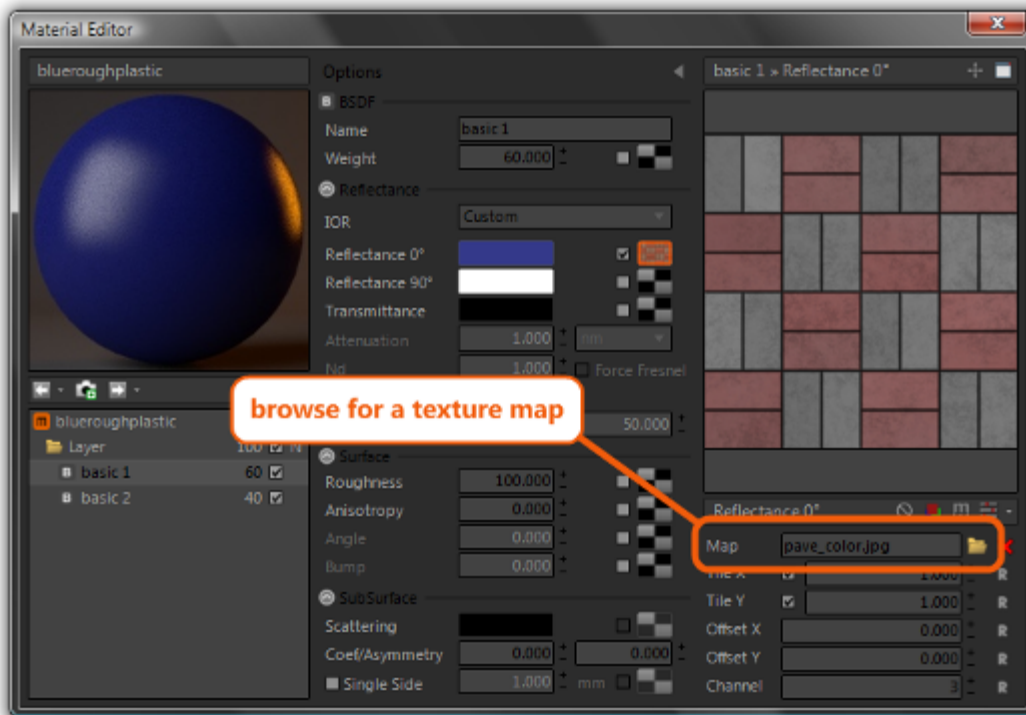
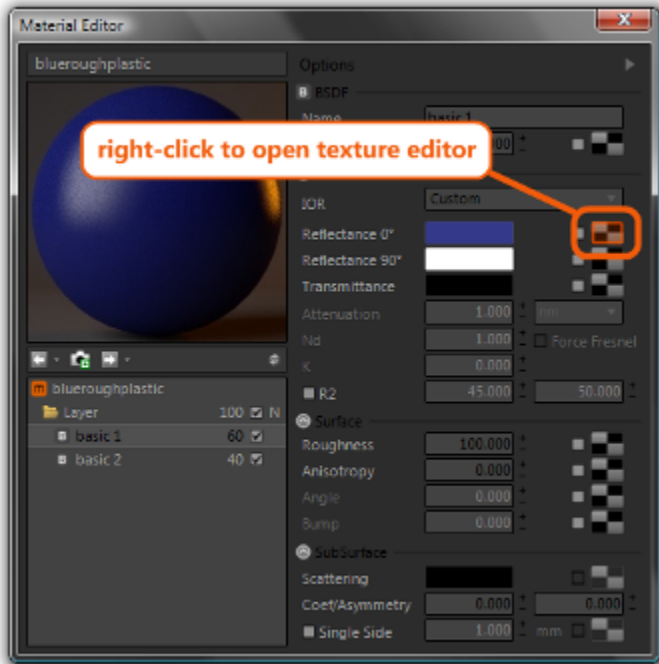
It was also necessary to adjust the EV down to something more like 9, since there is less light in this scenario. To wrap things up, let's texture one of these objects using the Maxwell material that is applied to it.

First, we should open the Material Editor, select the material, select the top node in the layers tree, and switch the material's MXM Linking to Embedded, if it is not already. The reason for doing this is because when a material is created by dragging it from the MXM Browser, it may be set to Link the MXM file which was dragged – this depends on the value of the plugin option *MXM Linking*, under *material Defaults*. As such, any changes we make to that material in Rhino will be lost next time we open the document, because the material will automatically update itself using the MXM file it links to. We are going to add a texture, so we do not want this.

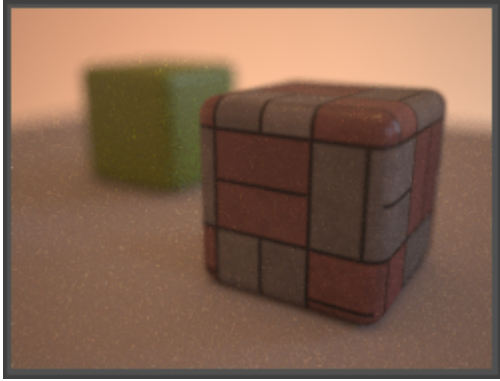
After making sure the material uses Embedded MXM Linking, select one of the BSDF layers in the material:



Now, right-click the texture button (the button with the gray/black checker pattern) for the Reflectance 0 channel and in the texture editor panel which is shown, browse for a map to use in this channel.



Next, left-click the Bump texture button (left-click it because the texture editor is already open), choose a map, and then set the Bump value to 5. Click the 'Refresh Preview' button, located to the lower-right of the material's preview image to refresh the preview. Rendering with the newly-textured material results in this image:



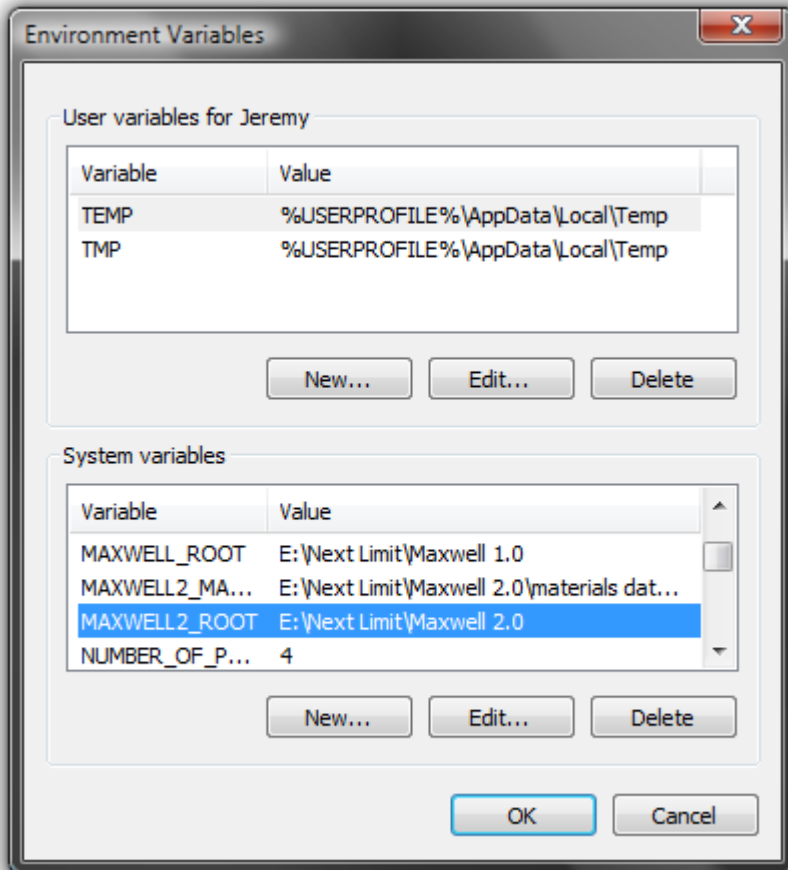
This wraps up the plugin Quick-Start. Other feature-specific tutorials may be found near the end of this manual, as well as on the Maxwell THINK! site (<http://think.maxwellrender.com>).

Maxwell for Rhino :: Frequently Asked Questions

Question: *Why does the plugin tell me it can't find a necessary directory?*

On Windows, the Maxwell installer (that is, the Maxwell Render installer, not the plugin installer) creates an environment variable named MAXWELL2_ROOT, and sets its value to point to the Maxwell installation directory on your machine. This allows plugins to find Maxwell. If the plugin has told you that you do not have a MAXWELL2_ROOT, it probably means that you either have not installed Maxwell Render, or that the Maxwell installer was unable to set the variable correctly.

To resolve this, go to Start > right-click My Computer > choose Properties > go to the Advanced tab > and click the Environment variables button. This should get you to a window like this:



Make sure that there is a System variable named MAXWELL2_ROOT, and that it points to the Maxwell folder on your machine. Also, make sure if there are MAXWELL2_ROOT variables in both the User and System sections that they both point to the same directory, otherwise the plugin may not be able to find your real Maxwell directory.

Question: *How can I make the rendered image match what is shown in the Rhino viewport?*

By default, the rendered image will match what is shown in the viewport. If however, you have modified the Camera's film size, this will no longer be the case. The way to get back to the starting point is to switch the Camera back to 'Viewport' resolution mode in the Scene Manager's Camera Settings page.

It is also useful to enable the plugin's Camera Heads-up Display. This tool shows the relationship of the Camera's films size to that of the Rhino viewport, which is always 24mm x Nmm or Nmm x 24mm. If you change the Camera's film size, which puts it into 'Manual' mode, then the Camera's resolution will no longer be linked to the viewport size. This is quite advantageous, since you can set up a shot which will not be destroyed by the simple act of re-sizing the related viewport.

Question: *Why aren't my Emitters working?*

Maxwell Emitters behave according to real-world parameters. Therefore, it is necessary to:

- make sure your scene scale is realistic - 100W spread out over 100 sq. meters is not very bright
- make sure your emitter plane normals are facing the correct direction
- make sure your camera settings are correct
- make sure that the Emitter power is defined correctly (don't forget the Efficacy parameter for watt-based emitters)

Question: *Textures work differently than I expected, is this by design?*

Most definitely. The way to work with Textures in the plugin may seem unconventional upon first use, but it has been purposefully designed to provide a quick workflow. The first concept to understand is that there is always only one 'current' texture for each material, if any. It is this texture which is shown in the viewport. Here are the basics of operation:

- **LMB:** Clicking a Texture button with the left mouse-button makes it the 'current' texture. Only one Texture is ever 'selected' for each material. It is this Texture which will be shown in the Texture Editor, and on objects in the viewport. Selecting an empty Texture shows the untextured material in the viewport. Double-clicking the button will bring up a file browser.
- **RMB:** Clicking with the right mouse-button opens or closes the Texture Editor, and also sets the Texture for that button current.
- **Drag-and-Drop:** Textures may be dragged from a Texture button onto another Texture button, or to the Texture preview window, assigning the path of the source Texture to that of the target Texture. If CTRL is held down during the operation, not only the path will be copied, but all of the source Texture's parameters as well. Textures may be dragged also over the material Layers Tree, making it possible to drag a Texture from one layer to another (it is the same for colors as well)
- **Path Text-box:** The text-box in the Texture Editor shows only the filename, rather than the entire path (which is shown in a tooltip), of the file which is assigned to it. It is not possible to directly edit the path using the keyboard, this is always done using a dialog-box shown when the 'browse' button is clicked. Because directly editing the path is not possible, paths may be cleared in one of two ways: 1) dragging an empty Texture on to the Texture you want to clear, or 2) right-clicking the path text-box and choosing 'Clear' from the context-menu that is shown

While the Material Editor does emulate Maxwell MXED in many respects, Texture-handling is one area where it makes somewhat of a departure. This aspect has been designed with the plugin's goal of providing completely-modeless operation in mind, and to accommodate real-time editing of Textures. Typical first-time out Q & A might include:

Q: *Why doesn't left-clicking a Texture button open the Texture editor, or show a file-dialog?*

A: Texture buttons are used to set the selected Texture, whether or not the Texture Editor is visible. It is designed to allow selecting different Textures in the viewport without having the Texture Editor open. Double-clicking shows a file dialog.

Q: *Why isn't it that a Texture is selected, and the Texture Editor is shown (if hidden) when left-clicking a Texture button?*

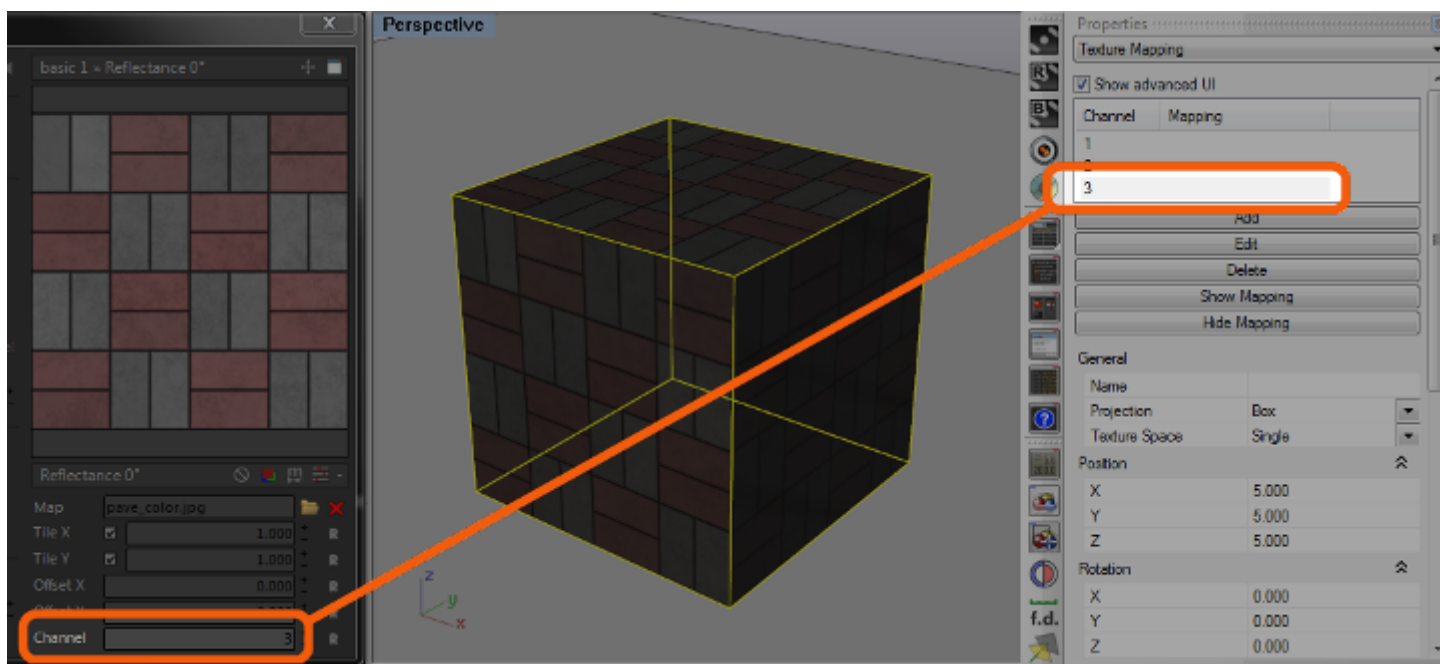
A: Similar to above, selecting a Texture is a separate action from opening the Editor. In fact, this exact sequence is accomplished by right-clicking a Texture button

Question: What are Channels, and how do I use them in Rhino?

Using the new texture-mapping tool in Rhino 4, an object may be given multiple sets of UV mapping coordinates. A set of UV coordinates (UV Set in Maxwell) may be one of several types, and is used to determine how a texture map is 'projected' onto the object it is associated with. The projector-types available in Rhino include:

- **Surface:** This is technically not a projection. The texture-map rectangle is basically stretched out across the object's surface
- **Planar:** A theoretical plane is created, sized to the bounding-box dimension of the surface which it is applied to. The texture-map rectangle is scaled to match the size of the plane, and the image pixels are projected through the plane in a direction perpendicular to it
- **Box:** similar to Planar, but here there are 6 planes surrounding the object, all 6 of which are mapped as with the Planar projection. An individual mesh-face's orientation determines which side of the mapping box it takes gets its coordinates from
- **Spherical:** The texture-map image is stretched/squished onto a theoretical sphere, that is, pixels at the center (figuring top-to-bottom) are mapped around the sphere's equator, and moving toward the top or bottom of the image, each row of pixels is compressed to fit the radius of the sphere at that cross-section height. At the top and bottom of the sphere, the last line of pixels is theoretically squished to infinity. Now, each pixel is projected onto the mapped object, using the theoretical surface normal at its location on the sphere
- **Cylindrical:** The texture-map is stretched to fit a theoretical cylinder, sized to enclose the object to which it is applied. Each pixel is projected onto to the mapped object using the theoretical surface normal at its location on the cylinder
- **Capped Cylindrical:** Identical to Cylindrical, but here the ends of the cylinder are capped and mapped using a Planar projection

A Texture's Channel number is used to determine which UV Set the Texture uses, and should be set to correspond with the index of the desired texture-mapping in Rhino's texture-mapping list:



This allows the various Textures that any material may use to be mapped individually onto the objects which the material is applied to, with visual adjustments possible using either the global Tile/Offset parameters in the Maxwell Texture Editor, or the per-object UV mapping controls provided by the Rhino texture-mapping tool.

Note that Rhino starts numbering mappings at 1, and prohibits the re-numbering of a mapping to 0. This is because channel 0 is reserved for the 'current' mapping shown in the viewport. The Maxwell plugin follows the same convention, so to use a Texture with a certain Rhino mapping, simply set the Texture's Channel number to the same channel number used by the desired Rhino texture-mapping. Also, because it is possible to 'skip' projector numbers (e.g. a material with one Texture, for which the Channel is set to 4), Maxwell will give objects the channel 0 (viewport) UV coordinates for any channels which have no explicit Rhino texture-mapping channel.

Additionally, since there is never an explicit Rhino mapping on Channel 0, the plugin uses this Channel to implement the [Real Scale](#) feature. If a Texture uses Channel 0 and has Real Scale enabled, the plugin will create and maintain a special cubic mapping, which will always be sized based on [Tile X/Y x 1 meter], and will remain oriented to the object it is assigned to. This plugin-maintained orientation may be adjusted on a per-object basis using the Real Scale Texture Control section of the Maxwell Object Properties window.

Question: Why do projections show up as 'locked' when I open my MXS in Maxwell Studio?

The MXS format does not hold data about 'projectors'; UV mapping is saved on the objects themselves in the form of UV data attached to each mesh triangle. So in short, it is not projectors which are written into an MXS, it is the UV coordinates that they create.

Question: What about scene Scale?

You don't need to worry about scene Scale; the plugin translates the current Rhino drawing units into meters for Maxwell. Of course, as noted above, Emitters depend on scene Scale being correct. Also note, if you have enabled 'Cached Meshes', and switch Rhino drawing units after the meshes have already been cached, an incorrectly-scaled Camera may be written into the MXS. So, if you change drawing units mid-session, make sure to invalidate any cached meshes by either disabling/enabling 'Cached Meshes' (in Plug-in Options), or by moving/adding/deleting some geometry.

Question: I want to use an MXI/HDR background image in the Rhino viewport. How do I do this?

From the 'Maxwell' menu in Rhino's main menu, choose 'Show Maxwell Toolbars'. When they are shown, locate the 'Environments' toolbar:

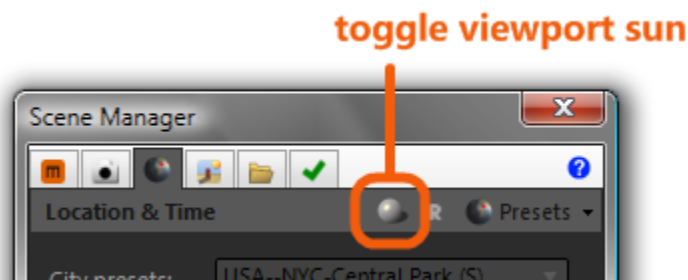


The first four buttons, 'Background', 'Illumination', 'Reflection', and 'Refraction', may be used to set their respective environment as visible in the viewport, while the fifth ('None') disables any active channel. If no MXI/HDR has been assigned to the channel (*Scene Manager > Environment Settings > Image Based*), the viewport will show a pale red background color. The path for the channel may be set either using the Scene Manager, or by right-clicking the toolbar button, which will show a file-browsing dialog. Note: display of Image Based Environments is not linked to their actual 'enabled' status in the scene; it is necessary to use the Scene Manager to make sure that the Sky Type is set to 'Image Based', and that the desired channels are enabled.

Changes to the active channel's Scale/Offset parameters will be made interactively in the viewport as they are adjusted in the Scene Manager.

Question: I can't see the Sun in the viewport. How do I turn it on?

In the Scene Manager's Location Settings page, there is a button in the toolbar next to the 'Presets' dropdown:



Use this button to toggle the Sun display in the viewport on and off.

Question: *The plugin does not seem to be working. What is wrong?*

The Maxwell for Rhino plugin requires Rhinoceros 4.0 SR5, and Maxwell Render 2.0. The plugin consists of one Rhino plugin, and five .NET libraries:

- Maxwell.Rhino.rhp
- Maxwell.NativeRMATypes.dll
- Maxwell.dll
- Maxwell.Common.dll
- Maxwell.UI.dll
- Maxwell.Script.dll

By default, Maxwell.Rhino.rhp is installed to *[ProgramFilesX86]\MaxwellDotNET\Maxwell.Rhino*, though you may have chosen a different location during installation. The remaining .dll files are installed in the .Net Framework's Global Assembly Cache, which may be viewed by going to *[system root]\Windows\Assembly*. You may view and confirm that these libraries are installed, but you should not attempt to alter the contents of this directory manually.

If a problem is encountered, the first thing to do is to confirm that Maxwell.Rhino.rhp is actually loaded. If it is not, the plugin will not be present in Rhino's Renderers list. The plugin installer sets up the registry in such a way that Maxwell should be available immediately after installation. However, if any errors occur, or if the configuration is not set up correctly, the plugin may not be automatically loaded, and it would be necessary to perform a manual load:

- go to Rhino main menu > File > Properties > Plug-ins
- if 'Maxwell for Rhino' is not found, Click 'Install', and browse to *[ProgramFilesX86]\MaxwellDotNET\Maxwell.Rhino*, choose Maxwell.Rhino.rhp, and click 'OK'
- verify that 'Maxwell for Rhino' is now in the list of plugins
- click 'OK' to exit the Document Properties dialog

Maxwell should now be installed correctly, and available in the Renderers menu

If you encounter an error during this process, it may be necessary to manually install the Microsoft Visual Studio 2008 SP1 C++ Runtime redistributable, which should normally be done by the plugin's installer, *setup.exe*. The runtime installer, which is included in the download, is a file named *vcredist_x86.exe* (or *vcredist_x64.exe*), and it is located in a folder of the same name in the plugin download. To run this, make sure it is located in a directory you have permission to install programs from, double-click the .exe file, and follow the installer's instructions.

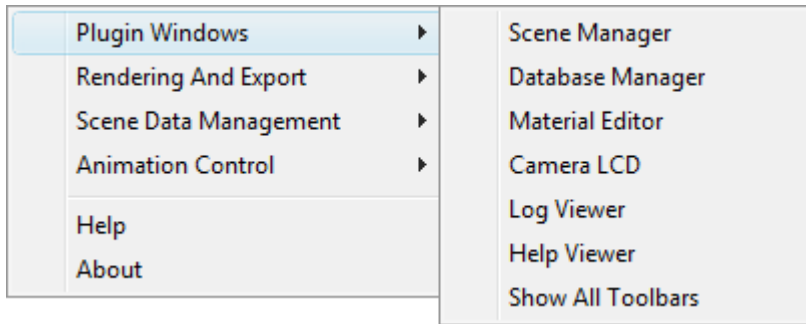
If these steps do not help, please note:

- your Maxwell Render version
- your Rhinoceros version (SR0, SR1, etc.)
- a list of other installed, non-standard, plugins
- confirmation that the plugin libraries are present, and located in the correct directories
- screen-shots of any error dialogs
- any log entries that may be present in Windows Event Viewer

Once you have gathered this information, send it, along with any other information available, to Next Limit tech support.

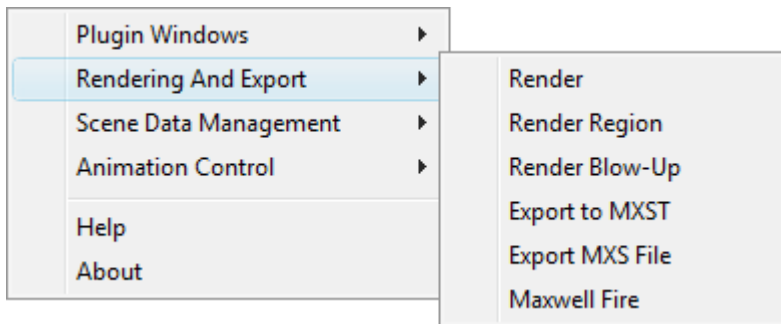
Maxwell for Rhino :: Main Menu

Plugin Windows



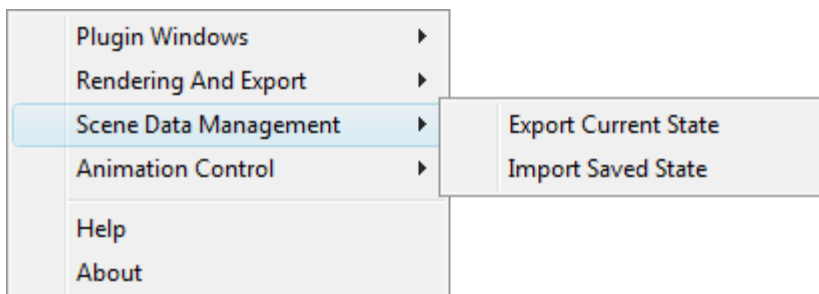
The visibility of the various plugin windows may be toggled using these menu items.

Rendering and Export



This menu item provides access to the various ways of exporting and rendering with Maxwell.

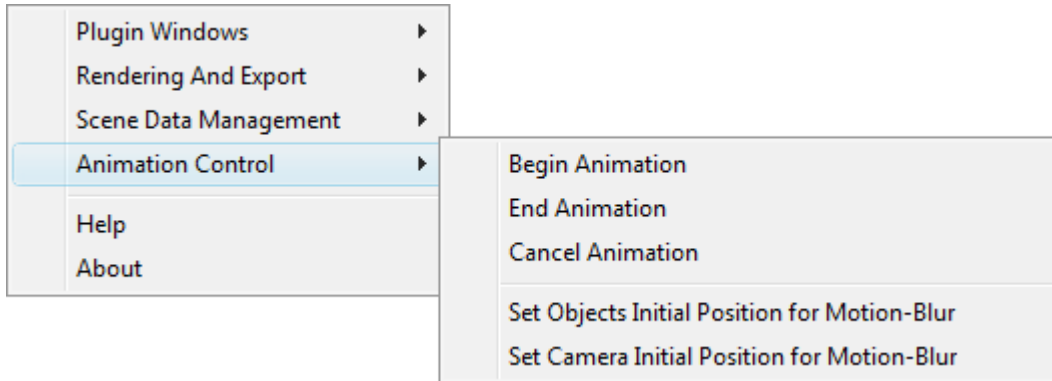
scene Data Management



This menu item provides access to commands used to import and export scene data and materials.

- Export Current State: saves all Scene Manager parameters to an .sxl file for sharing or for later use
- Import Saved State: prompts for the location of an .sxl file to import

Animation Control



This menu item provides access to commands used to begin, end, and cancel animations, as well as for setting Motion Blur positions (see Motion Blur) for Cameras and objects.

Maxwell for Rhino :: Toolbars

The plugin's toolbars may be shown using *main menu > Maxwell > Plugin Windows > Show All Toolbars*. If the toolbars become corrupted, run the command: *Maxwell_ToolbarReset*.

Maxwell Shortcuts



Each button shows one of the plugin's other toolbars as a flyout.

Tools



- **Scene Manager:**

- (LMB) toggle visibility of the Scene Manager
 - (RMB) show the Scene Manager Shortcuts flyout

- **Database Manager:**

- (LMB) toggle visibility of the Database Manager

- **Material Editor:**

- (LMB) toggle visibility of the Material Editor

- **Wizards:**

- (LMB) run the material Wizard

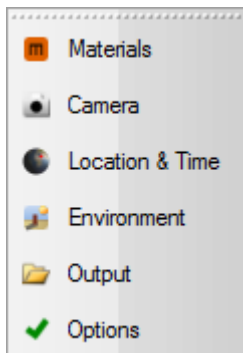
- **Log Viewer:**

- (LMB) toggle visibility of the Log Viewer

- **Help Viewer:**

- (LMB) toggle visibility of the Help Viewer

Scene Manager Shortcuts



Each button opens its related page in the Scene Manager.

Camera



- **Camera LCD Readout:**

(LMB) toggle the Camera LCD readout

- **Camera Heads-Up Display:**

(LMB) toggle the Camera heads-up display

- **Auto-Focus to Objects:**

(LMB) prompts for a selection of objects, which are then used to set the current Camera's focal distance and fStop. It is recommended that the current Camera have Auto-Exposure enabled when using this command

- **Pick Camera Focal Distance**

(LMB) start the Camera focal-distance-setting routine

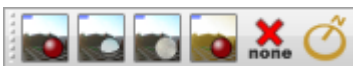
- **Pick Z-Clip Near Plane Distance:**

(LMB) start the Camera Z-Clip near-plane-setting routine

- **Pick Z-Clip Far Plane Distance:**

(LMB) start the Camera Z-Clip far-plane-setting routine

Environments



- **Background:**

(LMB) enables the Image Based Background channel (set a valid path first)
(RMB) browse for an MXI/HDR map to be applied to the Background channel

- **Illumination:**

(LMB) enables Image Based Illumination channel (set a valid path first)
(RMB) browse for an MXI/HDR map to be applied to the Illumination channel

- **Reflection:**

(LMB) enables Image Based Reflection channel (set a valid path first)
(RMB) browse for an MXI/HDR map to be applied to the Reflection channel

- **Refraction:**

(LMB) enables Image Based Refraction channel (set a valid path first)

(RMB) browse for an MXI/HDR map to be applied to the Refraction channel

- **None:**

(LMB) disables Image Based Environment display

- **Viewport Compass:**

(LMB) enables or disables display of a compass which indicates North (see Scene Rotation)

(RMB) picks the radius of the viewport compass, which must be enabled before using this command

Functions



- **Select Objects and Apply the Current Maxwell material:**

(LMB) choose objects to apply the current material to

- **Apply the Current Maxwell material to Selected Objects:**

(LMB) applies the current material to any pre-selected objects

- **Select Objects with the Current Maxwell material:**

(LMB) select all objects which have the current material applied

- **Remove material from selected Objects:**

(LMB) remove any Maxwell material from the selected objects

- **Remove material from Layers:**

(LMB) remove Maxwell material from the current Layer, if one is assigned

(RMB) remove Maxwell materials from all Layers

- **Restore Viewport materials:**

(LMB) refreshes the state of the viewport materials

Rendering



- **Render in Maxwell:**

(LMB) export the scene to an .mxs file and begin rendering in maxwell.exe

(RMB) export all selected objects to an .mxs file and begin rendering in maxwell.exe

- **Render Region:**

(LMB) select a region to render, export the scene to an .mxs, and begin rendering in Maxwell

(RMB) select a region to render, export all selected objects to an .mxs file, and begin rendering in maxwell.exe

- **Render Blow-up:**

(LMB) select a region to render in Blow-up mode, export the scene to an .mxs, and begin rendering in maxwell.exe

(RMB) select a region to render in Blow-up mode, export all selected objects to an .mxs file, and begin rendering in maxwell.exe

- **Write MXS and Open in Maxwell Studio:**

(LMB) export the scene to an .mxs file, and open in MXST

(RMB) export all selected objects to an .mxs file, and open in MXST

- **Write MXS:**

(LMB) export the scene to an .mxs file

(RMB) export all selected objects to an .mxs file

- **Maxwell Fire:**

(LMB) toggle the Maxwell Fire rendering window.

(RMB) open the Maxwell Fire window and start rendering.

Animation



- **Begin Animation:**

(LMB) sets up the animation loop, creates animation-output folders, and resets the frame counter. Until the animation loop is ended or cancelled, MXS output will be redirected to the animation folders, and maxwell.exe will not be started

- **End Animation:**

(LMB) ends the current animation loop (if there is one), and starts maxwell.exe rendering any queued frames

- **Cancel Animation:**

(LMB) cancels the animation loop, if there is one in effect

- **Motion Blur:**

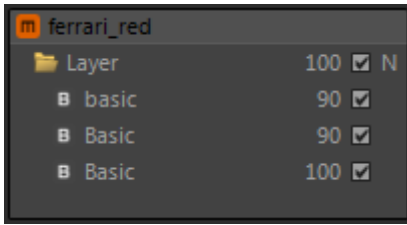
(LMB) Stores the current location of motion-blur-enabled objects. The stored location will be used to generate motion-blur when the scene is rendered.

(RMB) Stores the location of the current Camera, which will be used in the calculation of Camera-motion-blur when the scene is rendered.

Maxwell for Rhino :: UI Controls

The plugin uses several custom UI controls which are designed to provide for convenient usage.

Material Layers Tree (Material Editor)

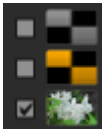


Materials are built from Layer, BSDF, Coating, Emitter, and Displacement components. Adding and removing of these components is accomplished by right-clicking the appropriate parent component and choosing from the options presented in the context-menu. It is possible to drag Texture and Colors between different layers by dragging them over the desired layer. When this is done, the layer currently under the mouse will be shown in the editor panel. Layers may also be renamed in the material Layers tree directly by clicking the layer name, followed by a short pause.

It is possible to re-order the structure using drag/drop; by default the component will be moved to the new location, or if CTRL is held down, a copy will be made.

Additionally, individual components may be dragged back and forth from the Layers tree to the various pages in the Database Manager. In this way, it is possible to build new materials from previously-saved components. Components stored in the Database Manager may be edited in the Material Editor by double-clicking them in the Database Manager; the current material (if any) will be unloaded and the chosen component will be loaded.

Texture Control



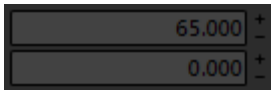
The Texture control consists of a Texture button and related checkbox. The checkbox enables or disables its associated Texture. Left-clicking the Texture button causes the associated Texture to become the material's 'Selected Texture'. It is this Texture that will be shown in the Texture Editor, as well as in the Viewport. Right-clicking a Texture button toggles the visibility of the Texture Editor, which is located on the right side of the Material Editor. Double-clicking (using the left mouse-button) a texture button will cause a file-browser to be shown which may be used to set the texture's path directly.

Color Button



Clicking a Color button brings up the Color Picker. Colors may also be dragged from one button to another, and between layers using the method described under the material Layers Tree topic. Colors may also be picked from anywhere on the screen by holding RMB down on the desired Color button, and dragging elsewhere. The cursor will then become a color eyedropper, and a tooltip will be shown detailing the RGB values currently under the cursor.

Numeric Textbox



The Numeric textbox allows a few different ways to adjust its value. Besides standard keyboard input, it is also possible to adjust the value by moving the mouse up or down while holding down LMB. The value may also be incremented using the small +/- buttons at the right of the textbox.

Color Picker

The plugin provides a custom color-picker. Colors may be added to the *presets* panel by drag-drop from the 'Current Color' and 'Starting Color' panels. As a small shortcut, right-clicking any color-picking surface is equivalent to clicking OK.

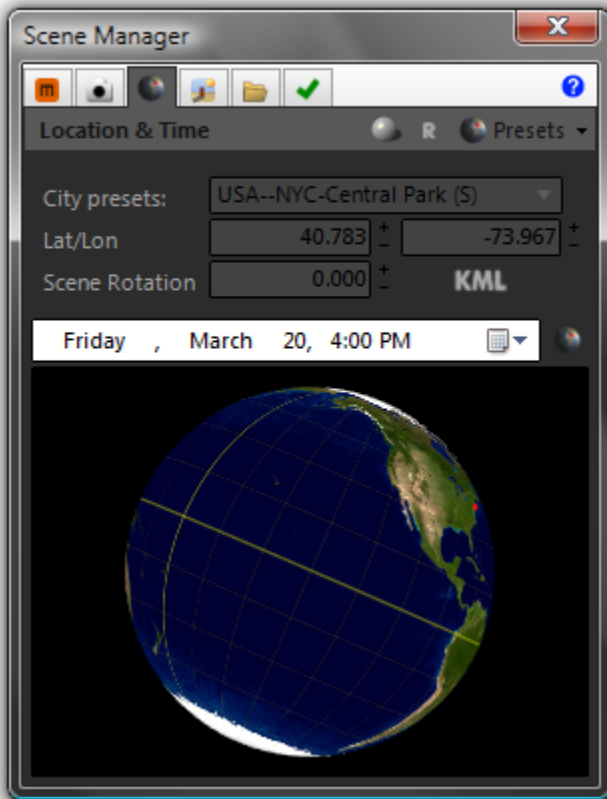


The plugin allows you to choose whether to use this custom color picker, the standard Windows color picker, or the modeling application's custom color picker, if it offers one. If you use either the plugin's custom color picker, or the Windows color picker, the color presets will be available; if you use the application's color picker, they will not.

As technical issues have been observed in some rare cases, the plugin default is to use the Windows color picker; to use one of the others use *Scene Manager > Options > User Interface > Color Picker*.

Maxwell for Rhino :: Scene Manager

The process of exporting and rendering of MXS files with the Maxwell for Rhino plugin is managed using the Scene Manager window. To open the Scene Manager, either click the appropriate button in the plugin's Tools toolbar, or click *main menu > Maxwell > Plugin Windows > Scene Manager*.

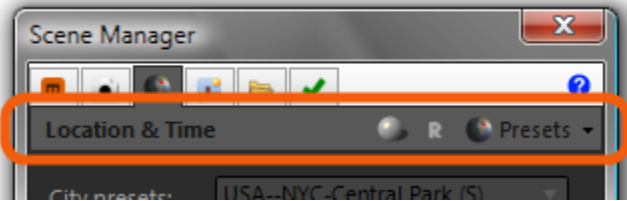


The Scene Manager has several tabs, each of which deals with a particular aspect of the Maxwell scene:

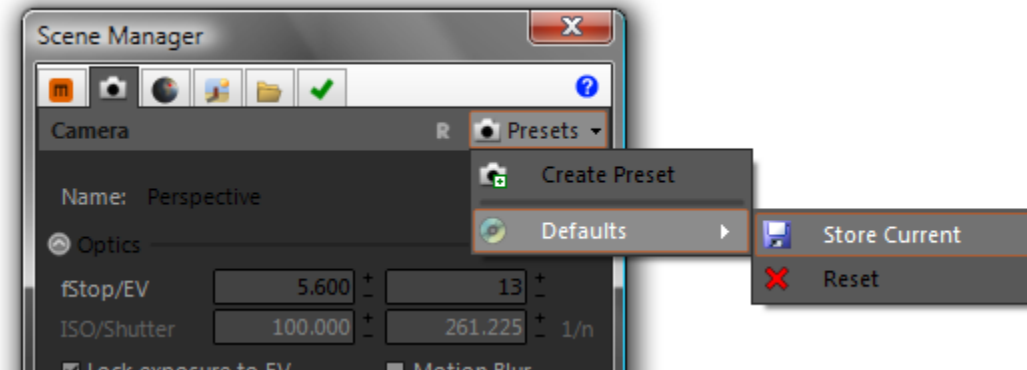
- Materials:** this is the list of materials used in the current scene; these materials are saved with the document. New materials may be created from scratch, by using the plugin's material Wizard, or by dragging MXM files into the list from the Database Manager's MXM Browser page, or from Explorer.
- Camera:** contains parameters used to control the Maxwell camera. Each Rhino viewport gets its own camera, including Named Views. At MXS-export time, all Rhino views and Named Views will be exported, unless the *Active camera only* option is set in *Scene Manager > Options > Export*.
- Location & Time:** contains parameters which control the scene's location, time, and date. Time and date may be set interactively by dragging the SunView globe, which provides a Sun's-eye view of the Earth. The scene's location may be set explicitly using the Latitude and Longitude values, by choosing a city preset from the drop-down list, or by browsing for a .kml file. The Sun may be shown in the viewport by toggling-on the *Show Sun in Viewport* button in the toolbar.
- Environment:** determines what render environment to use; the choices are: Sky Dome, Physical Sky, Image Based, or None.
- Output:** contains parameters which control where the output files will be written, how the image will be rendered, and which export options to use.
- Options:** contains various plugin user-Preferences which are used to determine various plugin behaviors.

Scene Manager Tabs & Toolbar

The toolbar located below the Scene Manager's tabs strip is context-sensitive; it holds slightly-different controls, depending on which tab page is currently being shown:

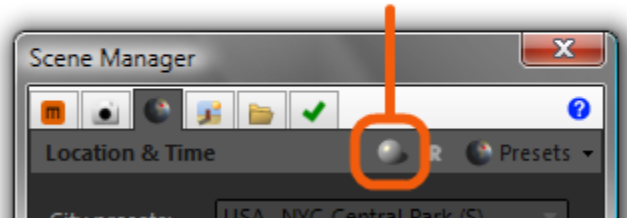


Most of the pages contain a 'Reset' button and 'Presets' drop-down. The Reset button simply sets all parameters in the current page back to default. Default means one of two things: the default settings used by the plugin as installed, or the user-defined default settings. The way you define your own defaults is to use the 'Defaults' item found at the bottom of the Presets menu:



Clicking 'Store Current' will set the current state of the page as your user-defined default, while clicking 'Reset' will return the factory settings. Similarly, clicking 'Create Preset' stores the current state of the page into a user-defined preset which may be recalled any time in the future. The materials page has no controls in the toolbar, while the Camera and Output pages have only the Reset button and Presets drop-down. The Location & Time page has, in addition to Reset and Presets, a toggle-button which enables the plugin's viewport Sun display:

toggle viewport sun



Toggling-on this button will also enable the Sun in the exported MXS file. However, since it is not always desirable to have the Sun enabled in the viewport, toggling this button off does not also disable the Sun in the exported MXS file; to turn the Sun off in the render, un-check the Use Sun checkbox in the Environment page.

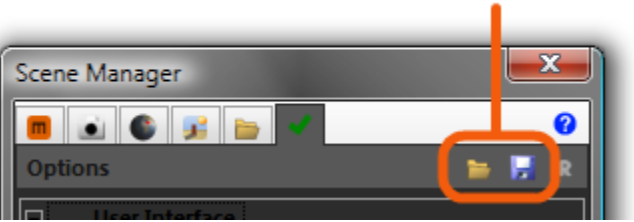
The Environment page has two extra buttons which are used to import and export Maxwell .sky preset files, which hold all of the information required to define Maxwell's atmosphere, location, date, and time:

import/export Maxwell .sky files



The Options page, which does not have a Presets drop-down, adds two buttons to the toolbar which are used to import/export files used to define plugin color-schemes. These are simple .xml files, and they allow easy sharing of color schemes.

import/export UI preset files



In addition to these functions, the Scene Manager's tabs:

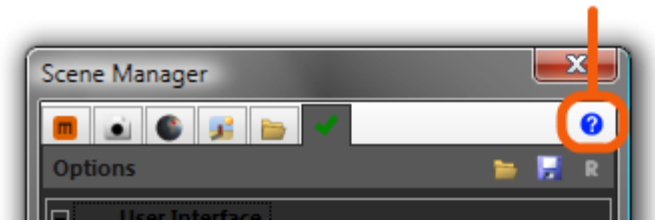
- remember which tab was selected at the close of the last session
- can be re-ordered by holding down ALT and using drag/drop to move tabs

Due to the auto-GMT feature of the Location & Time page's SunView globe, it is recommended not to leave this page visible at shutdown; this is because each page is loaded on-demand and loading the Location & Time tab takes more time than the other lighter-weight tab pages.

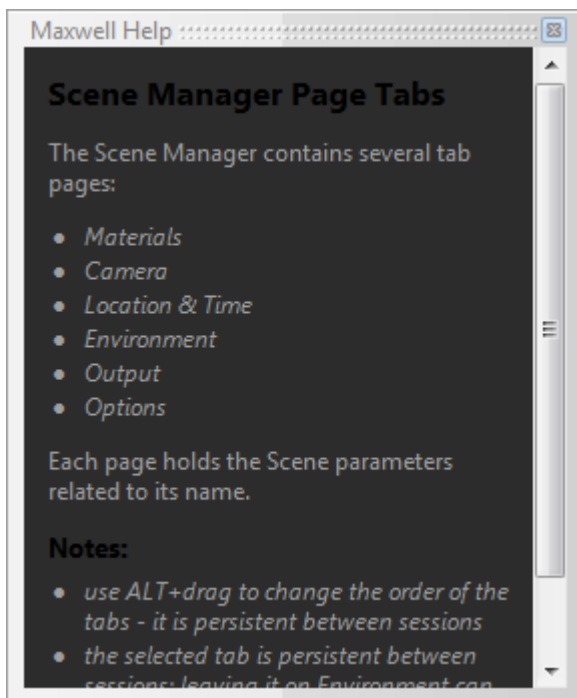
Other General Scene Manager Info

The Scene Manager can be used as either a standalone window, or as a Rhino dockbar; this is controlled using the plugin option: *Options > User Interface > Dockable Scene Manager*. Additionally, the Scene Manager also has in its toolbar a button which toggles the visibility of the plugin's real-time help viewer window:

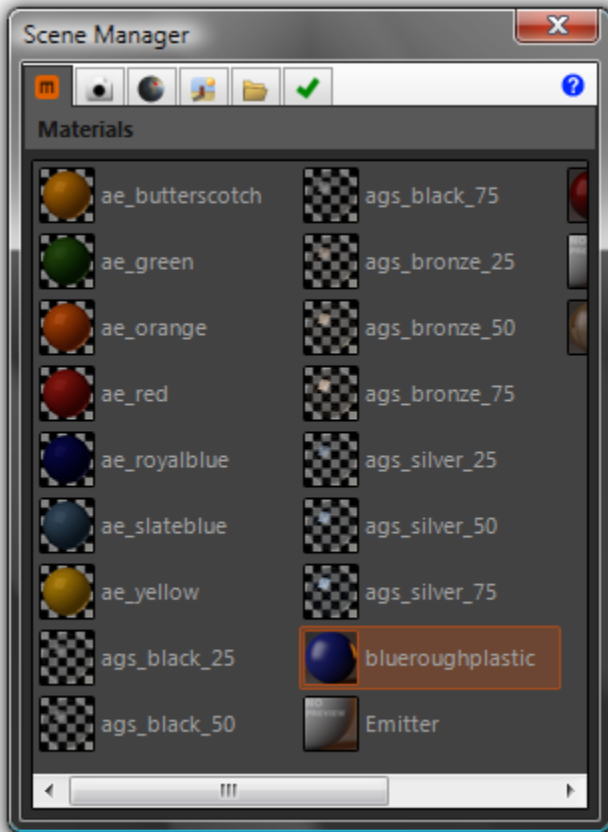
toggle help viewer



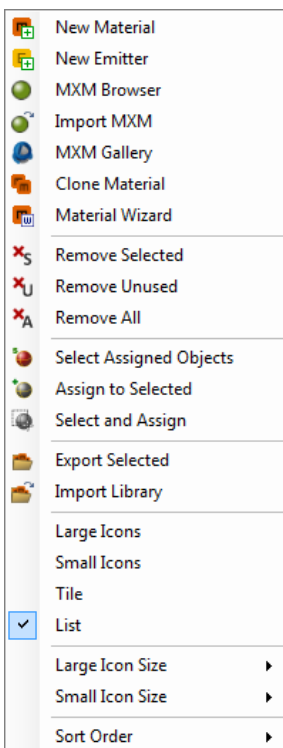
When the help viewer is open, the cursor will change to a 'help' cursor whenever it is over an item which has help associated with it; left or right-clicking at this location will cause this help to be shown in the help viewer:



Scene Manager :: Materials



The Materials tab simply holds a list of all of the materials contained in the current scene. These materials are all saved in the document and may be edited in the Material Editor by double-clicking on their thumbnail. Many actions are available in this window's context menu:

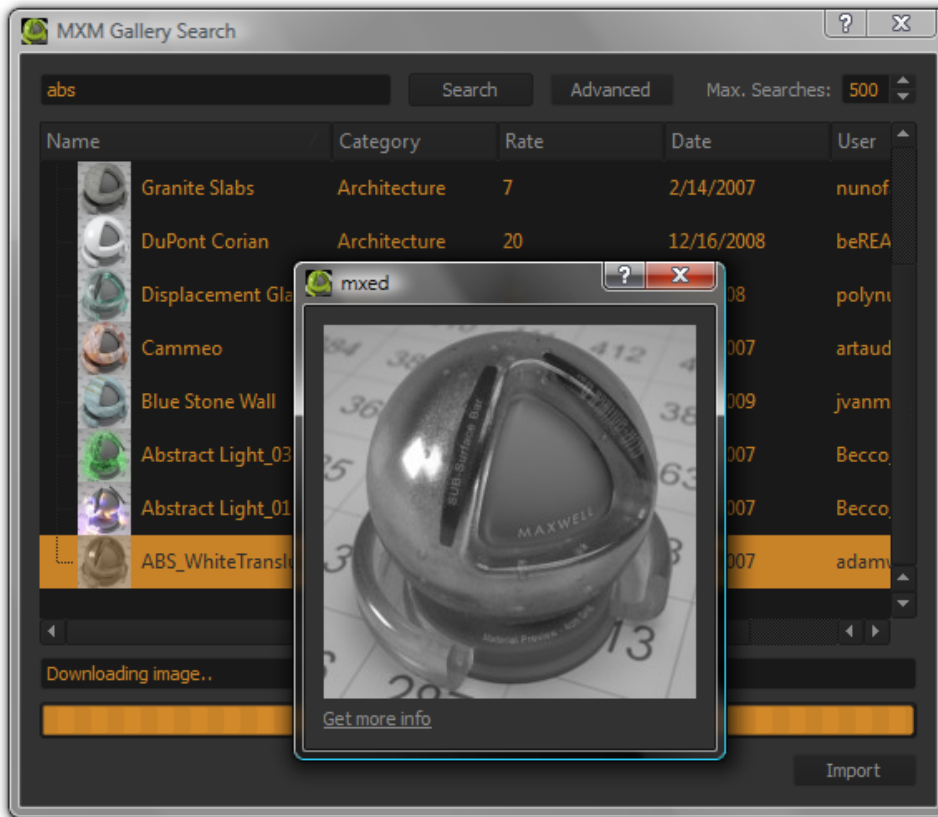


Materials Context Menu:

- the first group deals with creation of materials. *New Material* and *New Emitter* are self-explanatory; *Import MXM* prompts you to browse to an MXM file for import; *MXM Gallery* shows the Maxwell MXM Gallery Search dialog; *Clone Material* creates a copy of the selected material (as does using drag/drop while holding down the CTRL key), and *Material Wizard* brings up the plugin's Material Wizard.
- the second group deals with removing materials from the Scene.
- the third group deals with material selection and assignment.
- the fourth group allows you to work with plugin-specific material library files, which are very convenient for packing up groups of materials into a single easy-to-transport file which is independent of the Scene.
- the remaining items allow you to control how material thumbnails are drawn and organized.

MXM Gallery Browser

In Maxwell 2.0, the Maxwell MXED material browser includes a new 'MXM Gallery Search' dialog. Choosing 'MXM Gallery' from the *Scene Manager > Materials* context menu shows this dialog, allowing you to search the Maxwell MXM Gallery database without leaving the plugin environment. Here is what the MXM Gallery Search window looks like:

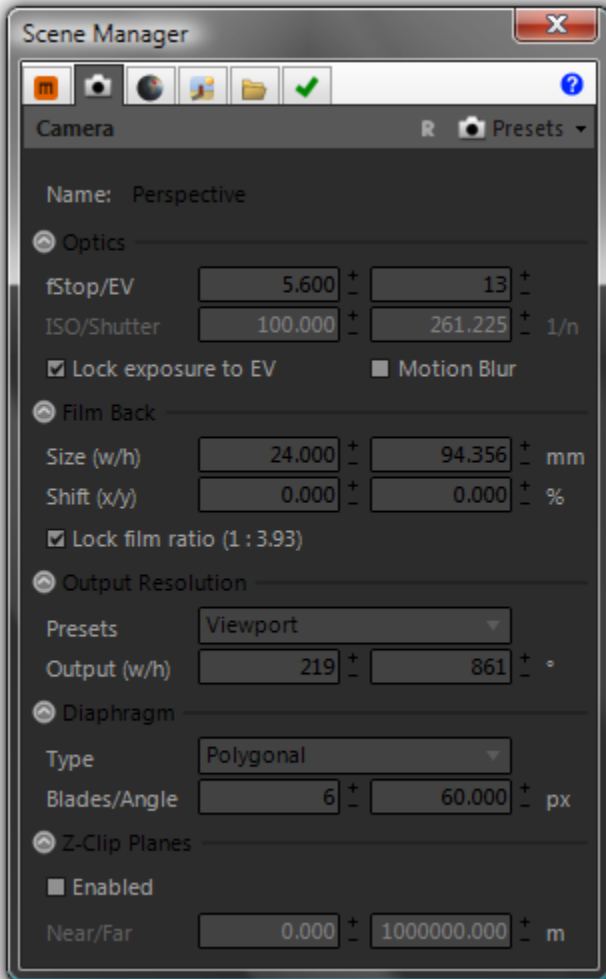


When you have located the MXM you would like to download, simply click the 'Import' button in the search window to import the selected MXM file into the Scene. The file will be downloaded and unzipped to one of two places:

- if the current document has not been saved, the MXM and related files will be saved to `[documents]\Maxwell\mxmgallery`
- if the current document has been saved, the MXM and related files will be saved to a new folder named `mxmgallery`, which will be created in the same directory as the current document

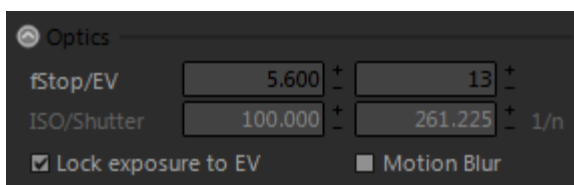
During this process, all files referenced by the newly-downloaded MXM will be placed in a `textures` folder under the `mxmgallery` folder. The paths inside of the MXM file will be altered to point to the new location, since many MXMs from the MXM Gallery contain paths which will not exist on your machine.

Scene Manager :: Camera



The Camera page contains parameters which are used to control those parts of the Maxwell camera which are not inferred from the Rhino view to which it is associated (i.e. position, target, etc.). These parameters relate to the exposure, image output resolution, and some other special camera-related Maxwell features. Each view gets its own Maxwell camera.

Optics



The parameters in this group determine the camera's exposure.

EV

EV, or Exposure Value is a number used to express the basic exposure which will be seen in the exported MXS. EV may be used to set the exposure directly, where brighter scenes will require a higher EV, or it may be used as an exposure readout when setting fStop, Shutter, and ISO manually. For example, an outdoor photo during the daytime may require an EV somewhere around 12-16, while a dark candle-lit scene may require a value much lower, maybe around 4-6. Charts containing recommended EV for various lighting conditions can be found online (see the end of this manual for a chart from wikipedia.org).

EV is not actually a parameter found on Maxwell cameras – it is a theoretical value which is continuously figured by the plugin itself. The 'Lock exposure to EV' option is provided for one basic reason: while exposure may be adjusted manually using different combinations of fStop, Shutter, and ISO, it is possible to adjust the DOF (depth of field) on the camera using the fStop without also affecting the overall exposure of the shot when the plugin is allowed to maintain a given EV. This allows you to adjust the camera's exposure and DOF on a totally independent basis, and this makes setting up exposure much easier and more predictable.

fStop

A camera's fStop refers to the size of its aperture. Lower numbers denote a larger aperture, which exposes the camera's sensor to more light. Additionally, a larger aperture reduces the camera's DOF (i.e. 'depth of field'), or the distance between the nearest and furthest points which will appear in focus. Smaller apertures (higher fStop numbers) result in a greater DOF, but expose the camera's sensor to less light. Therefore, as the fStop is increased, the exposure must be compensated. This can be done manually, by adjusting the camera's shutter or ISO values, or it can be handled automatically by the plugin by enabling the 'Lock exposure to EV' option. When the exposure is locked to the EV, changes to fStop will not affect the overall exposure of the output image, since the plugin will adjust the camera's shutter as necessary to maintain a consistent exposure.

ISO

As with a physical camera, the ISO parameter affects the camera sensor's light-sensitivity. It is not necessary to adjust ISO directly when 'Lock exposure to EV' is enabled.

Shutter

Specified in units of 1/second, this parameter controls how long the camera's Shutter is to remain open. As with the ISO parameter, it is not necessary (or even possible, in the interest of simplifying the camera's operation) to adjust the camera's Shutter directly when 'Lock exposure to EV' is enabled.

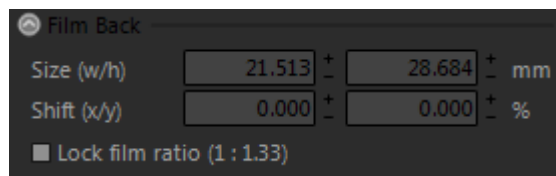
Lock Exposure to EV

Enabling this option disables the camera's ISO and Shutter inputs, since with it enabled, it is not necessary to adjust them directly.

Motion Blur

Enabling this option will cause the plugin to export two different positions for the camera: a 'previous' and 'current' position. It is the difference between these two physical positions which will determine how Maxwell calculates motion blur in the rendered output. While the 'previous' frame will be calculated automatically during export of animations, it is also possible to set it explicitly by positioning the camera as desired and right-clicking the last button (*Set camera Initial Position for Motion Blur*) in the plugin's Animation toolbar. Doing so will cause the camera's position to be stored in order that it may later be used during export.

Film Back



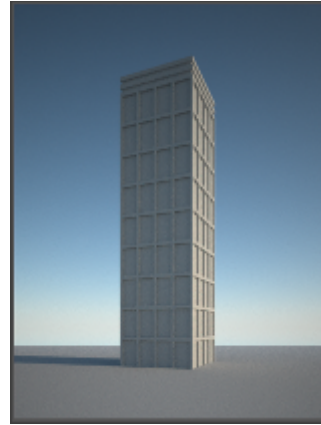
The parameters in this group are used to control the size and position of the camera's film.

Size (w/h)

Specified in millimeters, these inputs simply determine the size of the film. When the plugin is set to use 'Viewport' Resolution Mode (more on this later), the plugin will manage the film size in such a way as to maintain a perspective match between the Rhino viewport and the rendered image.

Shift (w/h)

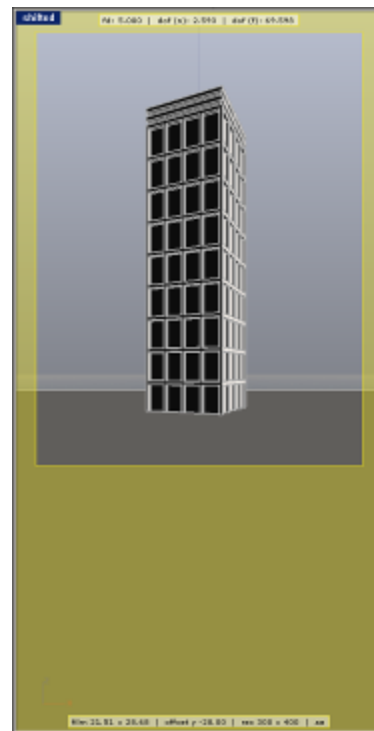
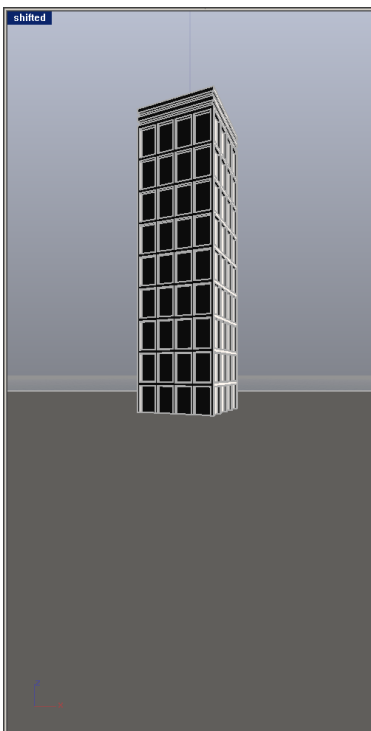
Specified in percentage, these inputs shift the film up/down and side-to-side in relation to the lens. This can be useful for setting up perspective-corrected shots of objects such as tall buildings where the camera should be located near the ground, but the entire building should be in the frame. Normally, this would require tilting the camera up, but by using lens-shift, the camera can be kept horizontal, and the film shifted downward in relation to the lens, thereby bringing the entire building into the frame without introducing perspective distortion. Here is a quick comparison of the two methods (camera-tilt vs. film-shift):



Since the Rhino viewport does not have the ability to show the effects of film-shift, or even of non-35mm film types, the plugin provides a specialized tool, the Heads-up Display, to assist you in setting up shots like this. To activate the Heads-up Display, click the 'Toggle Heads-up Display' button in the plugin's camera toolbar:



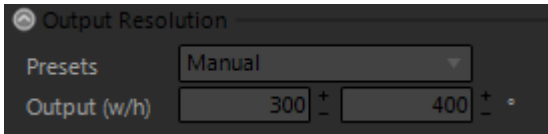
The Heads-up Display is an overlay that the plugin draws in the Rhino viewport, which shows how the current view will look once the film size and shift have been applied. Using the preceding example, here is how the Heads-up Display helps you to visualize the effect of your Maxwell camera's film settings (the color of the overlay can be adjusted using *Options > User Interface > UI Preferences > HeadsUpHatchColor*):



Lock Film Ratio

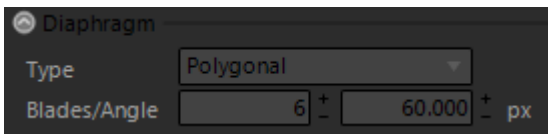
When this is checked, then the plugin will maintain the current width-to-height ratio of the film when either one is adjusted. As the Output Resolution is tied directly to the size of the film being used, enabling this option also has the effect of constraining changes made to the Output width or height in such a way as to maintain the film ratio.

Output Resolution



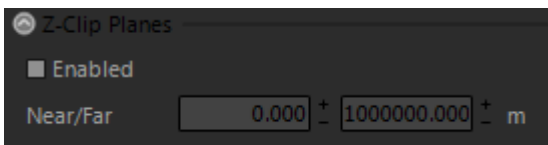
By default, the camera uses what is called the 'Viewport' preset. When this is the chosen preset, the plugin will manage the output resolution based on the physical size of the associated viewport. If, at any time, manual changes are made which affect the image or film size, the camera will switch to using the 'Manual' preset. When this is the case, the plugin will not actively change the image or film sizes; what you enter will be what is rendered. The remaining presets in the drop-down operate the same as the 'Manual' preset and are just provided for convenience.

Diaphragm

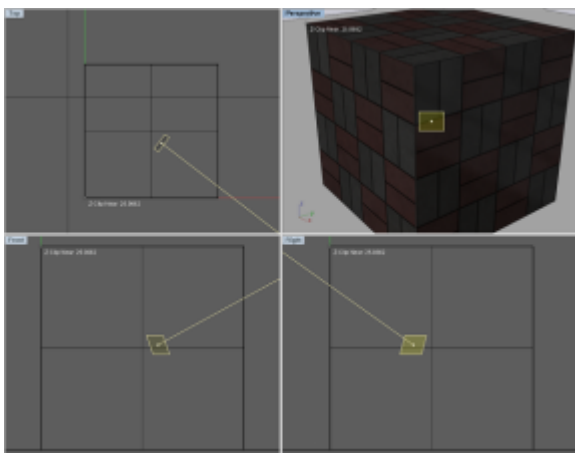


The parameters in this section affect the angle and number of blades used in this camera's aperture, when the Diaphragm type is set to 'Polygonal'. These properties affect the 'bokeh' effect seen in highlights located in out-of-focus portions of the image.

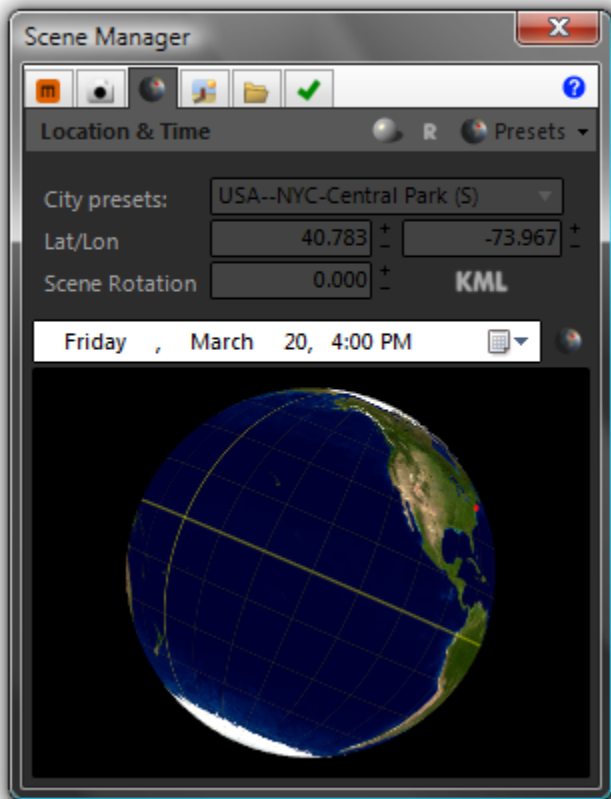
Z-Clip Planes



When enabled, Maxwell Render will clip any geometry nearer or further than the Near and Far distances defined here. These planes may be set interactively using the commands provided in the Cameras toolbar:



Scene Manager :: Location & Time



The Location & Time page controls the where and the when of the scene; this affects the Physical Sky environment when it is enabled, and the position of the Sun.

City Presets

The plugin uses a file, cities.txt (located in the main Maxwell directory, i.e. C:\Program Files\Next Limit\Maxwell), which contains a list of cities and their respective latitude and longitude values.

Latitude & Longitude

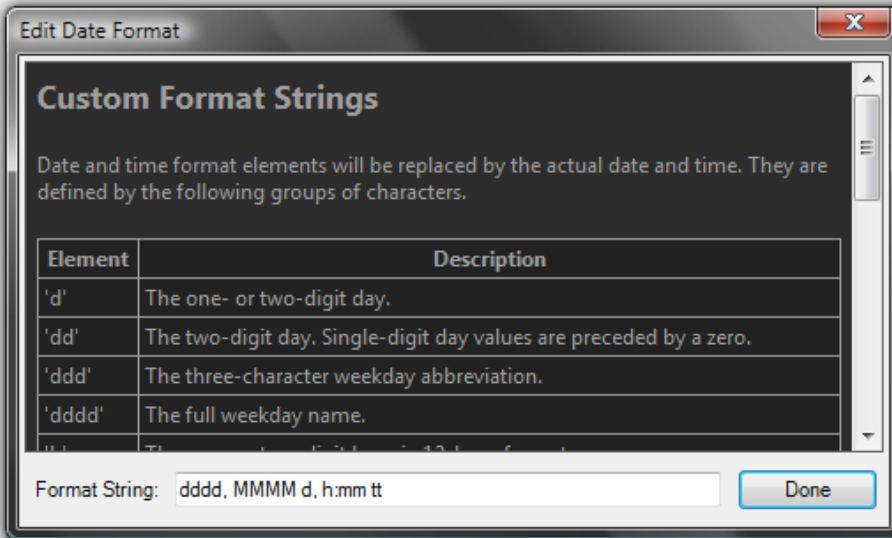
These parameters control the physical location of the scene. Though Maxwell also requires the accurate specification of the location's particular GMT offset, the plugin contains an internal map of valid GMT values for all locations, so it is not necessary for you to manually specify this information.

Scene Rotation

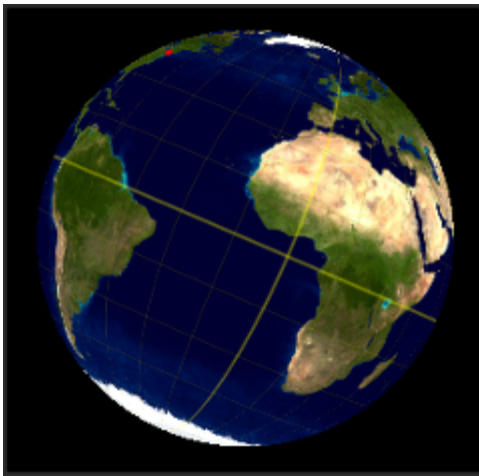
While North lies along the positive y-axis by default, you can use this parameter to quickly change it to point in another direction. This is much easier than physically rotating the actual model.

Date & Time Control

This control provides convenient methods for setting a specific date/time for the scene. Clicking on the down-arrow will show a graphic calendar, while selecting a part of the date/time and scrolling the mouse-wheel up or down will increment the selected value. The format shown in the control may be customized by right-clicking it and choosing 'Customize Format'. Doing so brings up a window which includes instructions on how to define the date/time format:



SunView Control

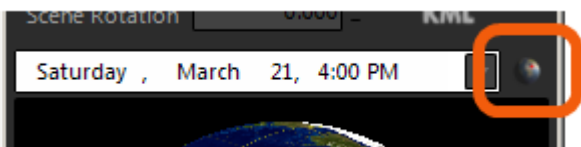


The SunView globe provides an easy and intuitive way of setting the current date and time of day. Conceptually, it shows you the Earth from the Sun's point of view, allowing you to intuitively perceive how the changes you are making will affect the Sun's position in relation to your scene location. Operation is accomplished using the mouse:

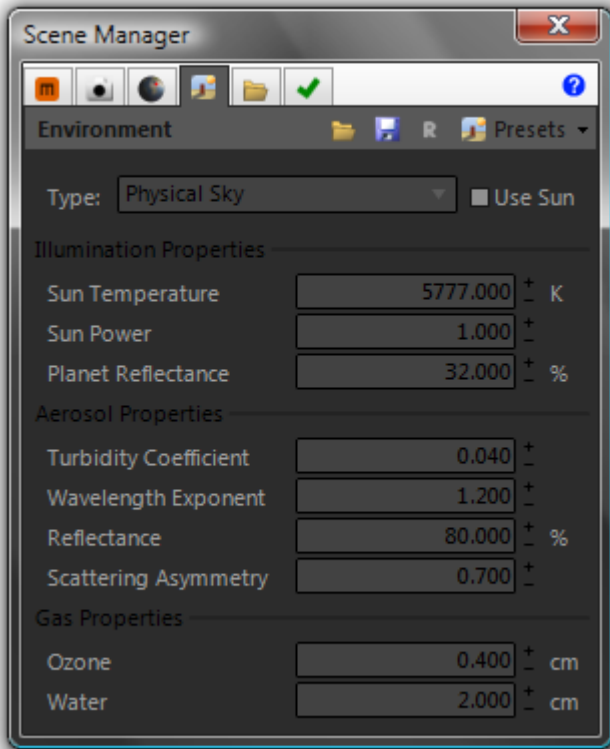
- left mouse-button down, drag left/right: the globe will spin on its axis, setting the time of day
- right mouse-button down, drag up/down: the globe's axis will be adjusted, setting the day of the year

Rolling the mouse-wheel will advance the time of day in one-minute increments, while doing the same with CTRL pressed will increment the day of the year.

To provide an alternative means of control in the event that the plugin is unable to obtain the necessary OpenGL rendering context on your video card & monitor, the plugin also provides an alternate slider-type user-interface for adjusting date and time. This is enabled using the toggle-button located next to the date and time picker:

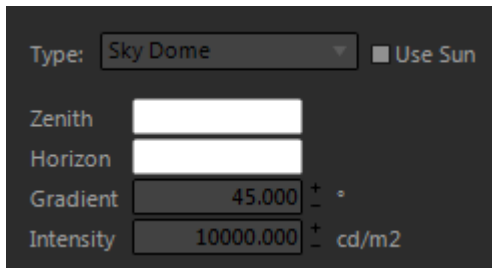


Scene Manager :: Environment



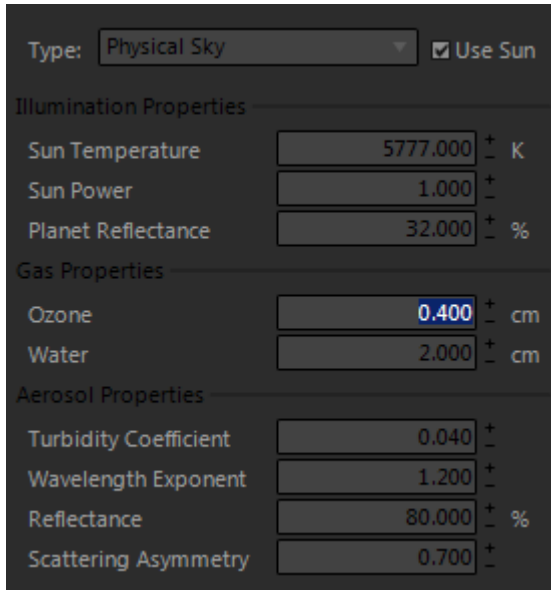
The Environment tab controls the environment which will be used in the exported MXS, and contains all of the parameters necessary to control each Environment Type. There are four types of environment available; Sky Dome, Physical Sky, Image Based, and None. The environment for a scene is set by using the drop-down list contained in the main Environment group. The Sun may be enabled regardless of which environment type is chosen.

Sky Dome Parameters



When Sky Dome is chosen as the Environment Type, the scene will be lit by a sphere of light, the intensity of which is controlled by the Intensity parameter. The angle specified by the Gradient parameter determines the point at which the Zenith and Horizon colors are evenly mixed.

Physical Sky Parameters



When Physical Sky is chosen as the Environment Type, the scene will be lit by an all-encompassing dome of light generated by Maxwell to simulate the appearance of a natural sky. How this sky appears is controlled by a set of parameters divided into three basic groups:

- **Sun Parameters:** control the power and color of the Sun and the basic reflectivity of the planet
- **Gas Parameters:** control the amount of ozone and water vapor in the air
- **Aerosol Parameters:** control the characteristics and concentration of particulates in the air

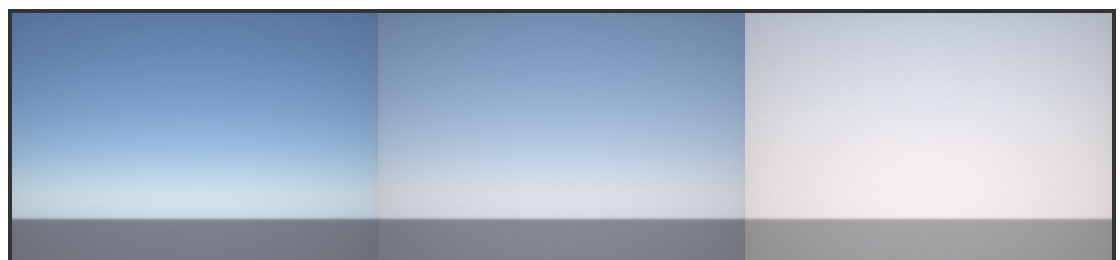
Physical Sky parameters may also be loaded or saved using Maxwell .sky files, just click the 'Load' or 'Save' buttons. Note, .sky files hold more than just atmospheric data; they include the date, time, location, and whether or not the Sun is enabled.

Sun temperature: Temperature of the sun's spectral radiation. The default value of 5777 is the most common value measured outside the Earth's atmosphere.

Sun power: A multiplier that controls the amount of light emitted from the sun. Higher values than the default 1.0 will make the sun emit more light, lower values will emit less.

Planet reflectance: Controls the percentage of light reflected from the planet surface back into the atmosphere.

Turbidity Coefficient: Defines the concentration and amount of particles in the atmosphere. A value of 0.0 means a perfectly clear sky (in which case the next three parameters won't have any effect). 0.01 is a good value for low turbidity, e.g. a sky containing very few aerosols, which will appear almost clear. 0.040 is a good medium value and 0.3 is a fairly high value.



Turbidity Coef. 0.0,

Turbidity Coef. 0.04

Turbidity Coef. 0.3

Raising the turbidity coefficient in sunset situations will darken the sky. Remember that you can also alter the EV of the camera in these cases, which can create an interesting sky:



Turbidity Coef. 0.0, EV=15

Turbidity Coef. 0.1, EV=15

Turbidity Coef. 0.1, EV=12

Wavelength Exponent:

Defines the average size of the particles in the atmosphere. The particle size influences which wavelengths of light are absorbed and which are scattered. Higher values than the default will increase the saturation of the sky, until gradually it turns first green and then orange:



Wavelength Exp. 1.2

Wavelength Exp. 5.0

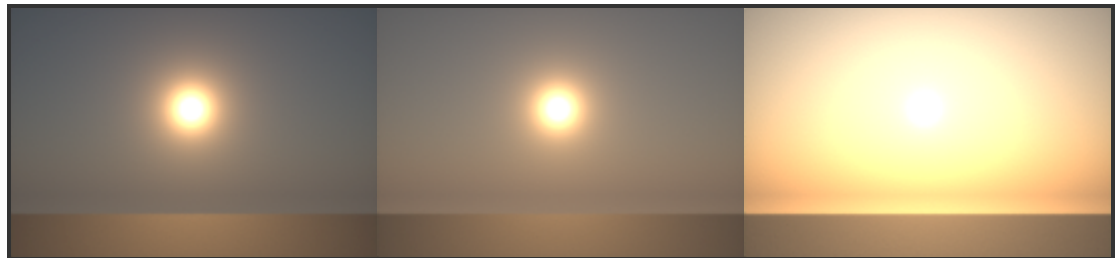
Wavelength Exp. 10.0

Reflectance:

Refers to the albedo of the aerosols, or the rate of energy scattered and absorbed by the aerosols.

Scattering Asymmetry:

This factor controls the anisotropy of the particles, that is, in which direction most of the light will be scattered. Positive values will produce a halo effect around the sun, brightening the area around it:



Scattering Asym. -0.7

Scattering Asym. 0.0

Scattering Asym. 0.7

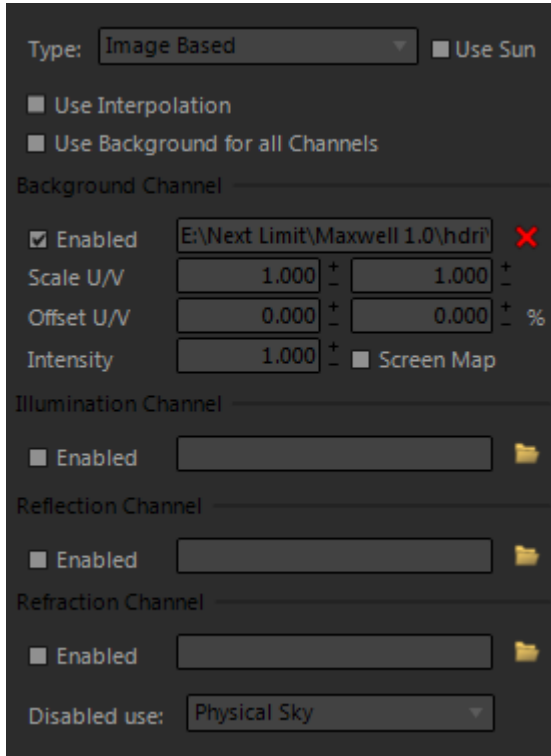
Ozone:

the amount of ozone gas in the atmosphere, measured in the centimeter-height of a column of atmosphere. The default value is 0.4.

Water:

The amount of water vapor in the atmosphere. It is measured in centimeters, in the same way as ozone.

Image Based Parameters



The Image Based Environment Type allows you to light the scene using HDR images. There are four Image Based Channels provided:

- **Background:** this image will be seen where the background is directly visible to the camera
- **Illumination:** this image will be used to illuminate the scene
- **Reflection:** this image will be seen in reflections in the scene
- **Refraction:** this image will be seen where light is refracted in the scene

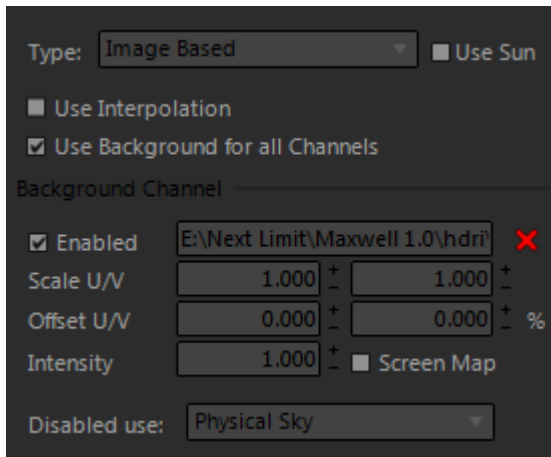
Each channel shares a few similar parameters and a similar operation:

- **Path:** this is the path to an .mxi, .hdr, or .exr file which will be used in the channel
- **Enabled:** this checkbox is only shown when it is possible to enable the channel
- **Scale U/V:** scales the input image in width and height such that n-number of tiles will be fit into the mapped area
- **Offset U/V:** shifts the image up or down by n-percent
- **Intensity:** alters the basic output power given to this channel's image
- **Screen Mapped:** this parameter is only present in the Background channel. It causes the Background channel to be mapped flat against the image plane, rather than spherically.

Parameters which have no context, given the state of other parameters, are not shown. For example, until a channel's path has been set to point to a valid file, nothing is shown for that channel except for the path chooser. When a path has been specified, but the channel is not enabled, all of the parameters, except for the path and the 'Enabled' checkbox, will not be shown.

As a shortcut, holding down CTRL while adjusting parameters in one channel will cause the change to be effected in all four channels.

Besides these per-channel parameters there are also three other main parameters available in the Image Based Environment Type:



Use Interpolation

This option turns on render-time pixel-interpolation for environment HDR images.

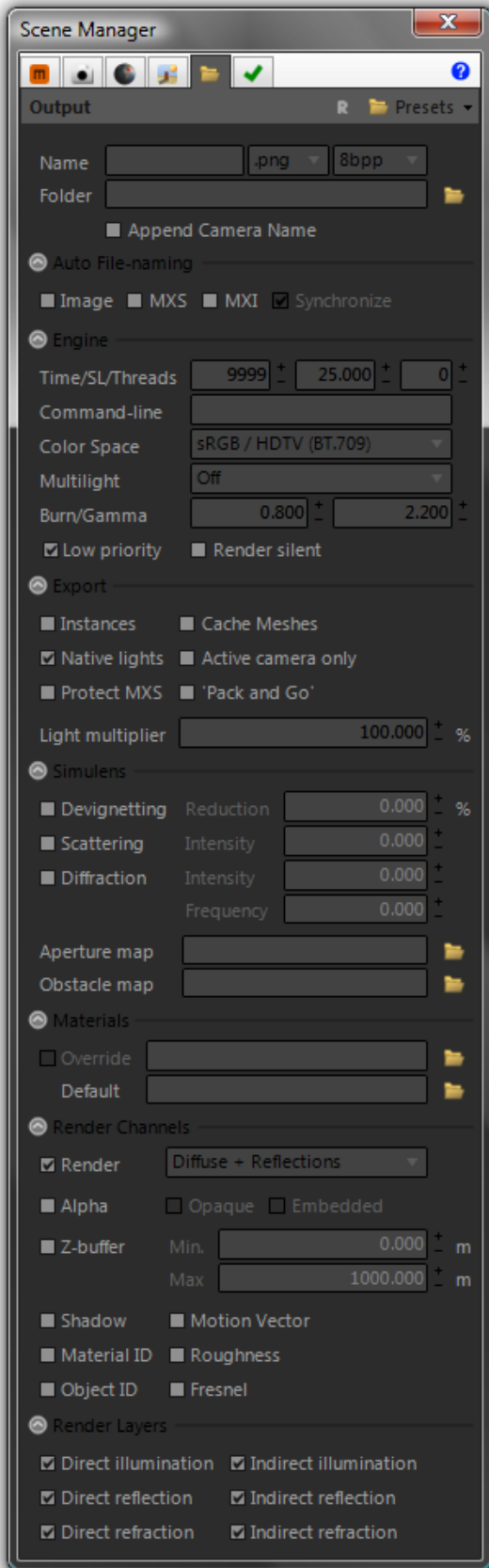
Use Background for all Channels

Most often, it is not necessary or desired to use four different HDR images in the Image Based environment. When this is the case, enable the 'Use Background for all Channels' option to have the plugin write all channels using the values defined in the Background Channel. When this is enabled (this is the default), the user-interface will not be cluttered by the unnecessary channel groups and parameters.

Disabled Channels Use

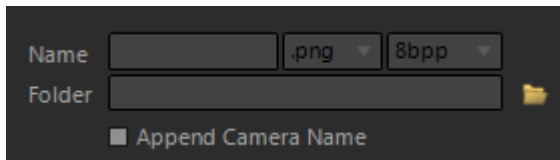
When there are channels which are disabled, or which have invalid paths set, the 'Disabled Channels use' drop-down will be shown, allowing you to choose which environment type to use in those channels.

Scene Manager :: Output



The Output tab contains several groups which control how MXS files are written and exported

Output (main group)



This group contains parameters which control where the files created by the plugin and Maxwell will be written.

Name

This is the name which will be used to generate the various filenames required to write MXS files. This includes the file name of the MXS file, the image-output file, and the MXI file. This should be specified as a name with no extension; the appropriate extensions will be inserted as necessary by the plugin. The Name should therefore also not contain characters which are illegal for use in file paths. If no name is entered here, the plugin will first try to use the name specified in *Options > Default Output > scene Name*, or else it will try to use the current document name.

Image Format/Output Bit-depth

This determines the format of the image output path written into the exported MXS; this is the file which Maxwell Render will write the primary (RGB) render output to. Depending on the image format selected, output may be written in 8-, 16-, or 32bpp.

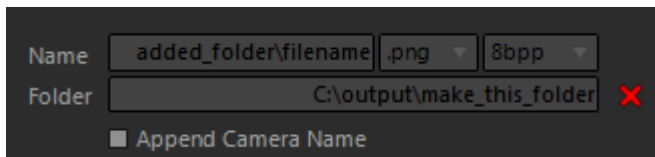
Folder

This is the root folder used for the plugin's output. MXS, image, and MXI files will be written to this location. If no folder is specified here, the plugin will try to use the location specified in *Options > Default Output > Working Folder*, or else it will try to use the current document location. If it is not possible to determine the current document location (for example, when the document has not yet been saved), the plugin will try to write output to *[documents]\Maxwell\output*.

Append Camera Name

Sometimes it is useful to keep track of output files using the name of the view being rendered; enabling this option accomplishes this by appending the name of the current view to whatever is entered in the Name field. So, rendering to a file named **output** with a current view named **perspective** would result in files named **output_perspective.mxs**, etc.

Both the Name and Folder parameters accept relative folders as input. So, if values were set like this:

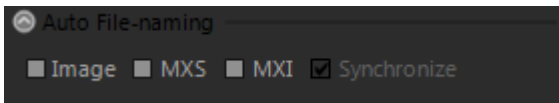


If **added_folder** and **make_this_folder** did not currently exist, the plugin would create the necessary intermediate folders, and the output files written would be:

- C:\output_folder\make_this_folder\added_folder\filename.mxs
- C:\output_folder\make_this_folder\added_folder\filename.mxi
- C:\output_folder\make_this_folder\added_folder\filename.png

Additionally, rather than setting Folder to an absolute path like **C:\output_folder**, it would also be possible to simply specify **output_folder**. In this case, the plugin would create a folder named **output_folder** at the current document location and write the output there. Furthermore, it is possible to set these values using environment variables; to do so, an environment variable must exist and point to a valid location. To use it, enclose the name of the variable in percent symbols: **%MAXWELL_OUTPUT_PATH%**; the plugin will retrieve the value of this environment variable and use it for the rendered output.

Auto File-naming



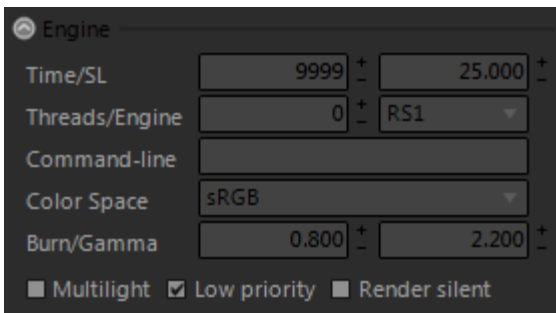
This group contains parameters which determine whether or not the plugin will automatically name output files, and how that naming will be done.

Image, MXS, MXI, Synchronize

For each of these that is checked when the export is done, the plugin will scan through the output directory and determine the highest-numbered existing file; it will then increase the number by one and use that as the new output filename. So for example, if the MXS option is checked, the output Name is **scene**, and a file named **scene_0006.mxs** is found in the output directory, the MXS file written by the plugin will be named **scene_0007.mxs**. This is done individually for the image, MXS, and MXI outputs, unless the **Synchronize** option is checked; in that case, all three outputs will be numbered using the same number, which will be the highest number found in the output directory for any of the three formats.

The reason that the **Synchronize** option is provided is that it can be confusing to open an MXS file named **scene_0007.mxs** and find that it contains an image output of **scene_0001.png** and an MXI output of **scene_0001.mxi**. Such a scenario would be possible if all three auto-name options were enabled, but no rendering was done; in that case, seven MXS files would exist, but no image or MXI files would exist; using Synchronize makes sure that all three would be named using the same **_0007** sequence number.

Engine



This group contains parameters which affect various aspects of the rendering engine (i.e. Maxwell Render) itself, which is a standalone application that runs in a separate process from the host application and plugin.

Time/SL

Time is the maximum time Maxwell will render before stopping. As it is possible to stop Maxwell at any time, it is generally best to leave this set to a higher number of minutes than will be necessary to obtain a clear image. Rendering will also stop if it reaches the specified SL (Sampling Level), or the Time value, if it is unable to reach the specified SL before the Time has been surpassed.

Threads/Engine

This parameter specifies how many threads will be spawned by the Maxwell render process. Setting this to zero will cause Maxwell to auto-detect and render using all available CPU resources. Setting the number of threads to [CPU count - 1] represents a trade-off between rendering speed and machine responsiveness, in case the machine must be used for other purposes while rendering.

Maxwell contains two rendering engines: RS1, which is the production engine, and RS0, which is the quicker 'preview' engine. RS0 can reach a higher SL in less time, but not all Maxwell features are activated, and the noise patterns are different than those seen when using the RS1 engine. Final images should always use the RS1 render engine.

Command line

The plugin (like all Maxwell plugins) does not directly contain any rendering functionality. Rather, it is an MXS file exporter. To render the MXS files it produces, it uses the Maxwell Render application. Maxwell Render runs from a command line, and different command-line flags are supplied by the plugin to start it rendering on an exported MXS file. In addition to the flags passed by the plugin, you can also enter any other Maxwell Render command-line flags you wish in this text box, and they will be added to the flags that the plugin gives Maxwell.

Color Space

Determines which color space maxwell.exe will use to render the output image.

Burn/Gamma

These parameters correspond to those of the same name as seen in the Maxwell Render user-interface. Burn is used to simulate over-exposure, while Gamma is used to compensate for monitor gamma.

Multilight

If this is enabled, the Multilight feature will be enabled. When Multilight is enabled, each emitter will get its own controls in the Maxwell Render's Multilight panel, allowing you to adjust the output power emitters on an individual basis, in real-time, while rendering.

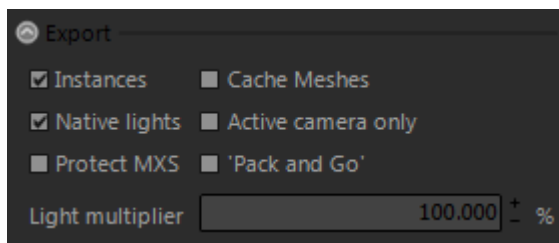
Low Priority

If this is enabled, Maxwell Render will be started on a low-priority process. This can help to keep the machine responsive, since on Maxwell is able to use nearly all available CPU power for rendering.

Render Silent

If this is enabled, Maxwell Render will be run with no GUI. This allows it to use fewer resources.

Export



This group contains parameters which control various aspects relating to the actual export process.

Instances

Enabling this option will cause the plugin to write Block Links out as Maxwell instances. This can drastically reduce both the size of the output files and the amount of memory required during export and rendering. Export speed can also be reduced greatly since the actual mesh-geometry only needs to be written once.

Native Lights

Enabling this option will cause the plugin to translate native spot, cone, and rectangular lights into real geometry for use by Maxwell. Emitter materials will be generated based on the native parameters of the light, or an explicit Maxwell material may be directly assigned. In the specific case of a spot light which has an IES-based Maxwell emitter assigned, the global transformation of the native light will be used to determine the directionality of the IES representation.

Protect MXS

Enabling this option will set a flag in the exported MXS file which will prevent the geometry it contains from being exported again from Maxwell Studio. This is provided as a way to protect intellectual property which may be contained in the file, while still allowing it to be distributed as necessary.

Cache Meshes

The process of exporting an MXS file consists of the plugin reading the mesh data contained in the Rhino model and transferring it into an in-memory representation of an MXS file. Once this has been accomplished, the in-memory MXS data is written to disk as an MXS file, which may then be processed in Maxwell Render. Enabling this option will cause the plugin to keep the in-memory data instead of freeing it immediately after export. While this means that more memory will be used, it also means that as long as no physical changes are made to the model, then the mesh data which was previously exported does not need to be exported again.

Therefore, when this option is enabled, changes to materials and camera position may be made after an MXS export, and the export repeated without the actual mesh-data being exported again. Since complex models may contain millions of points of mesh data, this data-caching can save a substantial amount of time. It has limitations, however, and these include: objects may not be added/removed/alterd/moved, texture-coordinates may not be altered, and the optics group of the camera may not be changed. For the most part, these changes are detected automatically, and any cached mesh data is invalidated. These things may not be detected in every case though, so keep it in mind when using this option that it may be necessary to disable and re-enable the option, thereby manually invalidating cached data, in some rare cases.

Active Camera Only

Normally, the plugin exports a camera for each view (including Named Views), whether it is the current camera or not. If you do not wish to export these extra Cameras, enable this option; only the currently-active camera will be exported.

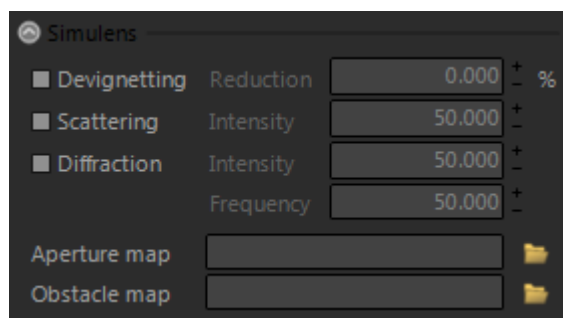
Pack and Go

When this option is activated, the plugin will scan through all files referenced in the scene (i.e. texture, ior, hdr, etc.) and copy them to the output directory during export. The paths contained in the exported mxs will point to the new copies, rather than the original files. This makes it much easier to transport an entire scene to another location.

Light Multiplier

When the plugin exports native lights, it attempts to approximate the power values used in the native application. This is not an exact science, depending on the size of the scene. If the output is unsatisfactory, it may be adjusted on a global basis using this value.

SimuLens



This group contains parameters which control the image-enhancing features known as Maxwell SimuLens.

Devignetting

By default, Maxwell Render simulates realistic vignetting (i.e. the darkening around edges of images); to disable this, check the Devignetting option and set the reduction value between zero and one-hundred percent to vary the amount of vignetting seen in the final output image.

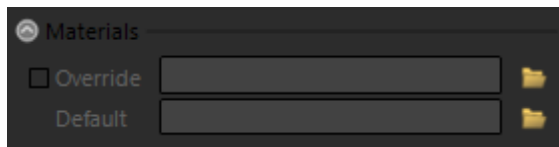
Scattering

This parameter simulates light-scattering caused by the camera's lens. Since this scattering is figured by maxwell.exe at each image update, it is recommended (in the interest of reducing render-time) to leave this disabled, then apply it to the final image via maxwell.exe's user-interface when the image has finished rendering.

Diffraction & Aperture/Obstacle Maps

SimuLens can also simulate camera lens-diffraction effects. There are aperture and obstacle maps included in the Maxwell Render installer, which may be found in the 'SimuLens maps' folder inside the Maxwell folder on your machine. As with the Scattering feature, it is not recommended to enable Diffraction from the plugin, since maxwell.exe will apply it at each image update – rather, wait until the image is finished and apply Diffraction using maxwell.exe.

Materials



This section is deals with materials on a global basis

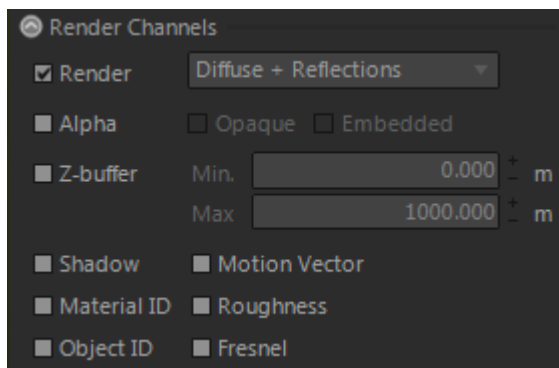
Override

If the path to a valid MXM file is placed here, then the file will be used to override all material-assignments (with the exception of emitters). This can be useful for light-studies and similar.

Default

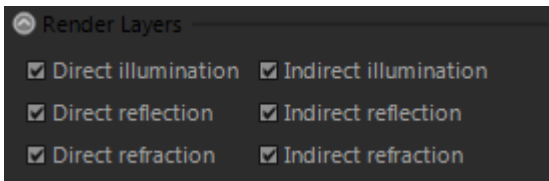
All triangles in a Maxwell MXS file must have a material assigned. For those which do not have a material explicitly-assigned, one will be assigned automatically, and this can be controlled by specifying the path to an MXM file in this field. If no MXM path is specified here, a default gray (153,153,153) diffuse MXM will be created and given to maxwell.exe to use.

Render Channels



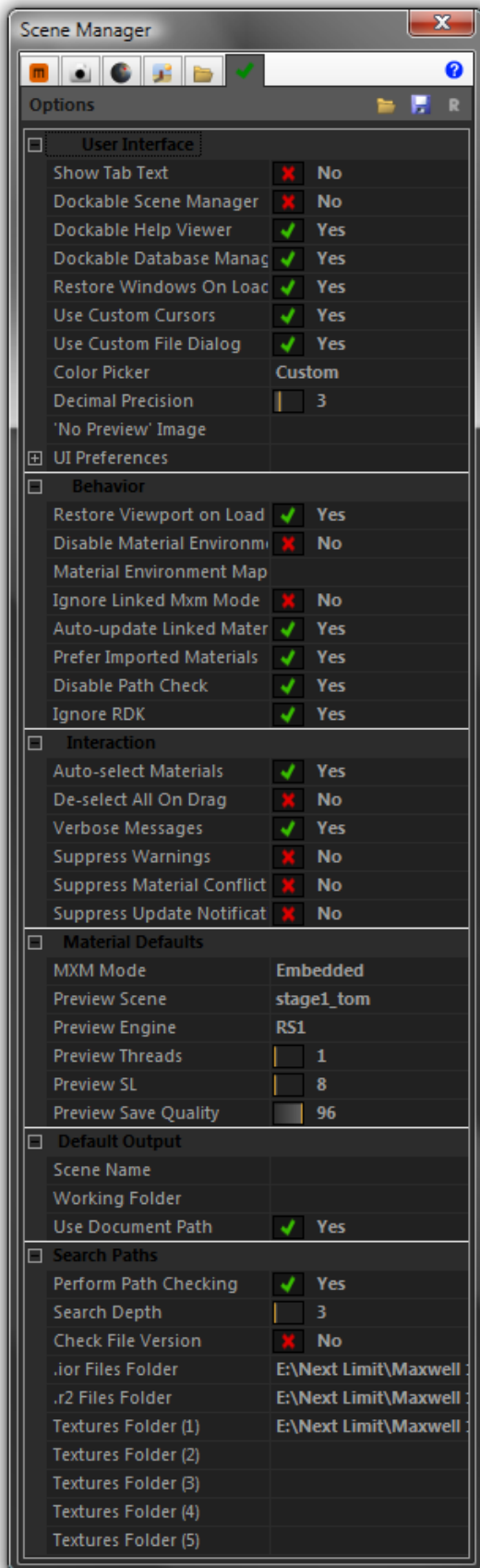
Options in this group control the various different image outputs which may be rendered by Maxwell Render. 'Render' is the main output image. The other channels are used for compositing in a third-party application like Photoshop – see the main Maxwell Render manual for more details on the various channels.

Render Layers



Though Maxwell naturally calculates all possible light interactions in a scene, it is possible to inhibit certain types of calculation using the toggles found in this section. The specific light-interactions which each of these options affect should be fairly self-explanatory, for more details, see the regular Maxwell Render manual.

Scene Manager :: Options



User Interface:

- **Show Tab Text:** shows or hides text on the Scene Manager tabs
- **Dockable (window):** sets whether the window in question is used as a standalone window or a Rhino dockbar
- **Restore Windows On Load:** sets whether or not the plugin should re-open windows which were opened at the end of the last session
- **Use Custom Cursors:** enabling this may cause cursor issues on some video cards; it may be fixed by disabling cursor acceleration
- **Use Custom File Dialog:** enabling this causes the plugin to provide a file dialog which is able to show previews for MXM, MXI, etc., files. It may be useful to disable this if the files you browse are large
- **Color Picker:** sets whether to use the Windows, Rhino, or the plugin's custom color picker
- **Decimal Precision:** sets the precision for the plugin's numeric inputs
- **'No Preview' Image:** this is the image used for Materials with no preview
- **UI PLinks:** various settings related to the UI color theme

Behavior:

- **Restore Viewport on Load:** when enabled, the plugin will refresh viewport materials when documents are loaded
- **Disable Material Environment:** disables the environment-mapping the plugin creates on viewport materials; it may cause problems on some video cards
- **Material Environment Map:** the bitmap used for viewport material environment mapping
- **Ignore Linked MXM Mode:** prevents any Linked Materials from reloading their linked MXMs at document-load
- **Auto-update Linked Materials:** causes Linked Materials to actively monitor their linked MXM files
- **Prefer Imported Materials:** when geometry is imported, there may be material conflicts; this determines what action to take when Suppress Material Conflict Dialog option is enabled
- **Disable Path Check:** at MXS export time, the plugin checks all texture paths; this takes some time, and it may be disabled using this option
- **Ignore RDK:** if the RDK is installed, it will be used. If you wish to forego its use temporarily, enable this option; it is quicker than load-protecting the RDK and restarting Rhino

Interaction:

- **Auto-select Materials:** enabling this will cause the materials of objects selected in the viewport to be selected in the plugin
- **De-select All On Drag:** when drag-dropping materials, you may prefer that all selected objects are de-selected first; if so, enable this option
- **Verbose Messages:** enables more detailed command-line reporting
- **Suppress Warnings:** suppresses 'overwrite?' types of messages, answering them in the affirmative instead of asking
- **Suppress Material Conflict Dialog:** causes the plugin to use the 'Prefer Imported Materials' option instead of asking which Materials to keep when there are conflicts during import
- **Suppress Update Notification:** the plugin checks for newer versions and notifies you when there are; enabling this option suppresses this notification

Material Defaults:

- **MXM Linking:** sets whether or not newly-created materials will have linking enabled or not
- **Preview scene:** sets which MXS preview scene should be used in new materials
- **Preview Engine:** sets which engine, RS1 or RS0, new materials will use
- **Preview Threads:** sets how many threads to use during preview-refresh
- **Preview SL:** sets what SL (sampling level) to use in newly-created materials
- **Preview Save Quality:** sets the save quality used in newly-created materials

Default Output:

- **scene Name:** this is the name which will be used when *Output > Name* is empty
- **Working Folder:** this is the directory which will be used when *Output > Folder* is empty
- **Use Document Path:** when *Output > Folder* and *Default Output > Working Folder* are empty, then this option sets whether the plugin should try to write output to the current document's directory, or to the default plugin output directory (located in your Documents folder). When this is enabled, the plugin will prefer to write to the current document's directory, though it may still write to the default folder when the document has not been saved yet

Search Paths:

- **Perform Path Checking:** sets whether or not the plugin should try to fix broken paths (i.e. textures, etc.) automatically or not
- **Search Depth:** sets how many folders deep the plugin should search when attempting to locate missing files
- **Check File Version:** thumbnails and preview images for textures are cached by the plugin; this means that once a large bitmap has been loaded, it need not be loaded again, and this saves much processing time. It is a problem, however, if you edit a file which has already been cached during the session – the changes will not be seen in the plugin. Enabling this option will cause the plugin to physically check whether the file version has changed each time a particular cached bitmap is accessed, updating it if the current file is newer than what is cached
- **.ior Files Folder:** this is the folder which will be searched first when looking for missing IOR files
- **.r2 Files Folder:** this is the folder which will be searched first when looking for missing R2 files
- **Textures Folder (1-5):** these are folders which will be searched when looking for missing texture files

Maxwell for Rhino :: Materials

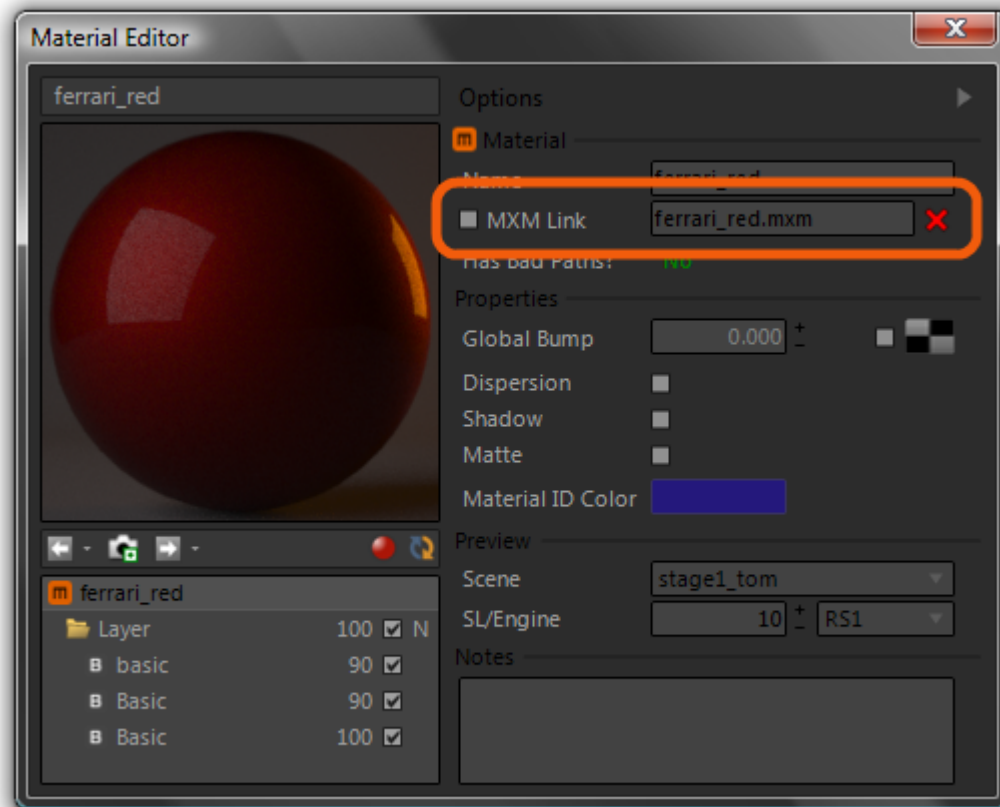
To create a new Maxwell material:

- in the Material Editor's *Options* menu, choose *New material*, *New Emitter*, or *material Wizard*
- in *Scene Manager > materials*, right-click and choose *New material*, *New Emitter*, *Import MXM*, or *material Wizard*
- in *Database Manager > MXM Browser*, choose a folder and drag an MXM file into the scene

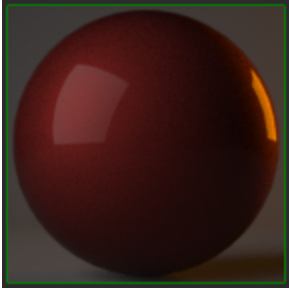
Plugin materials and Maxwell MXM Files

Firstly, it is important to recognize that there is a difference between an MXM file on your hard disk and a Maxwell material contained in a Rhino file – they are not the same thing, though they are very similar. It is necessary for the plugin to have its own 'version' of Maxwell MXM materials, so that it can save them directly inside of Rhino documents. If it did not do so, Maxwell materials could only exist as file paths stored in the document which link to MXM files on your disk. While this is entirely possible to do, and even preferable for some workflows, but it means that in addition to a Rhino file, you also need to bring along any number of physical MXM files with you when you want to transfer a document from one machine to another. Furthermore, it is prone to error, since directory structures most likely will not match from one machine to the next, and this often results in broken MXM links, and material assignments which do not produce any valid Maxwell material during export.

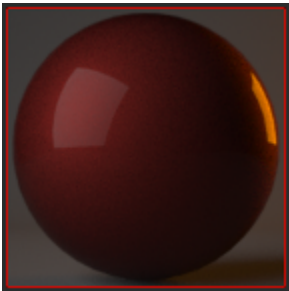
As such, a Maxwell material in the new Maxwell for Rhino plugin may operate in one of two ways: either linking to an MXM file on your machine, or simply embedded in the Rhino document. To control whether a material behaves as Linked or as Embedded, use the MXM Linking section in the material's top node in the Material Editor:



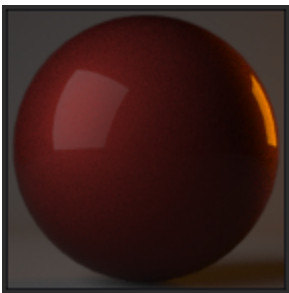
In order for a material to use Linked mode, it is necessary that its source MXM path points to a valid MXM file. When this is true, and the MXM is Linked, then material's preview will appear in the Material Editor and the Scene Manager's materials page with a green border:



If a material is set to use Linked mode, but the specified MXM file is not able to be found, it will appear with a red border:



If a material is set to use Embedded mode, then the border will have the standard border-color:



How does MXM Linking work?

MXM Linking is completely optional; that is, the full definition of the material is always stored (i.e. embedded) in the Rhino document. This ensures that no matter where you open a Rhino document, it will still be able to be rendered in Maxwell the same as it did the last time it was saved. How it works in practice is this:

- a Rhino document is opened
- the plugin reads the materials stored inside
- for each material which was saved with MXM Linking enabled, the plugin reads its source MXM path, and checks whether an MXM file is found at this location
- if the MXM file is found, the material is immediately updated with the contents of that MXM file. If however, the MXM file is not found, nothing further happens – the definition of the material which was found in the document will be used at render-time

Therefore, the MXM Linking feature of the plugin offers the ability to link central MXM files in individual Rhino documents, thereby allowing you to change how materials in multiple documents render just by editing one MXM file, while simultaneously handling scenarios where a specified MXM file is not found, or where the material is not set to link to any MXM file at all.

The plugin also provides a special mode which may be used to defeat this process entirely, in case you have a document which you know has linked materials contained inside, but you do not want them to update themselves. This is found at *Options > Behavior > Ignore Linked MXM Mode*. When this option is enabled, all materials will behave as though they are using Embedded MXM Linking.

The other plugin options which are concerned with MXM Linking are:

- Options > material Defaults > MXM Mode
- Options > Behavior > Auto-update Linked materials

The *material Defaults > MXM Mode* option simply determines whether new materials will have MXM Linking enabled. The *Behavior > Auto-update Linked materials* option determines whether or not linked materials should actively listen for changes to their linked MXM files. When this is enabled, then any changes made to an MXM file will immediately be reflected in any materials which link to it. When it is disabled, the only time a linked material updates itself is when the document it is contained in is opened – it may be modified during the session, but any changes will be lost the next time its document is opened (unless *Options > Ignore Linked MXM Mode* is enabled, of course).

By now, you may be asking, what is the real purpose of all of these options? Why are they necessary? The answer is simple: some workflows require materials to behave in an embedded manner, and not to hold any external relation to an MXM file, while other workflows require materials to behave in a way such that they always use the current state of an MXM file. Generally, where you have multiple people working on a project, and you want them all to use the same MXM file for certain materials, you would want to adopt a linked model of working. That way, it becomes possible to use Maxwell's MXED Material Editor to edit a single MXM and have the changes take effect in all documents which link to it. To establish this workflow, you would want to set your options as such:

- Options > material Defaults > MXM Mode = Linked
- Options > Behavior > Ignore Linked MXM Mode = No
- Options > Behavior > Auto-update Linked materials = Yes

Doing so ensures that any new materials created from MXM files will use linked mode, that they will actively listen for changes being made to those MXM files, and that they will update themselves whenever the documents they are contained in are opened.

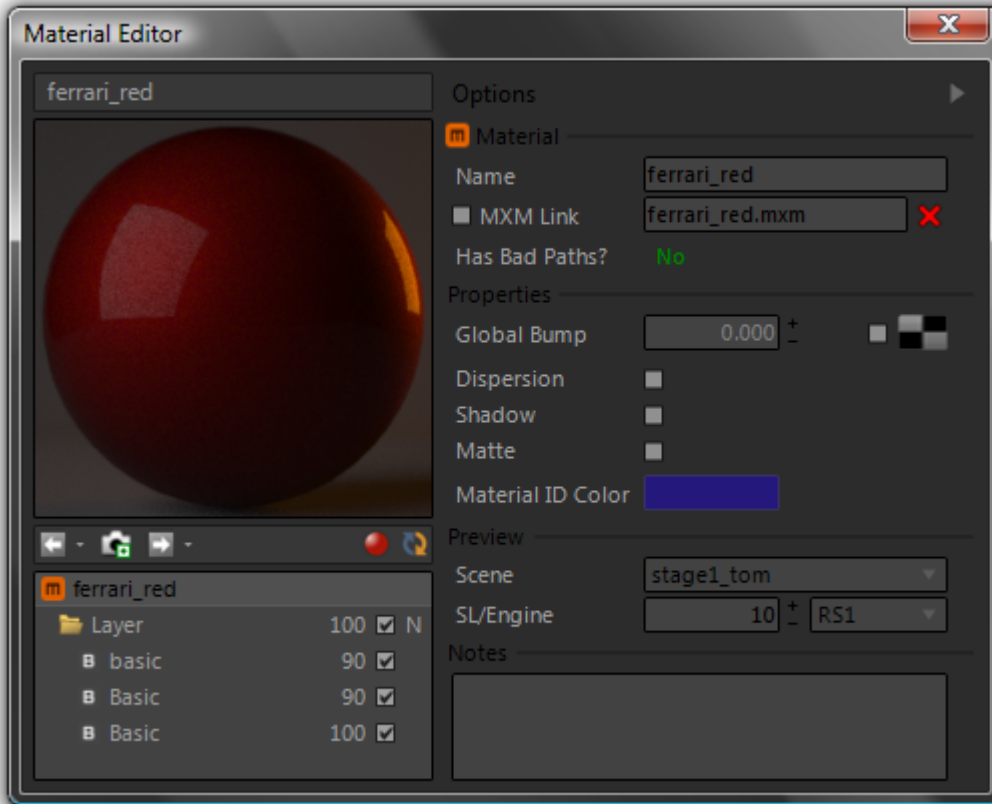
If, on the other hand, you prefer that each material always maintain its as-saved state in the document, then you would want to set those options such that:

- Options > material Defaults > MXM Mode = Embedded
- Options > Behavior > Ignore Linked MXM Mode = Yes
- Options > Behavior > Auto-update Linked materials = No

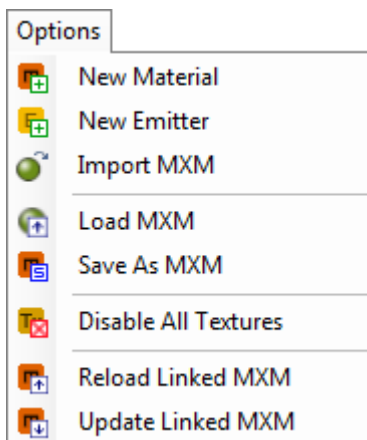
Doing so ensures that all new materials will be created using embedded mode, and that any which were created by importing MXM files will never attempt to update themselves. At any time, it is possible to manually re-load a material's source MXM file, either by using *Material Editor > Options menu > Reload Linked MXM*, or by checking the 'Linked' checkbox.

Maxwell for Rhino :: Material Editor

To open the plugin's integrated Maxwell Material Editor, either click Rhino main menu > Maxwell > Plugin Windows > Material Editor, or by clicking the appropriate button in the plugin's Tools toolbar. This will bring up the plugin's integrated Material Editor, which is very similar to Maxwell MXED:



Options Menu



New Material / New Emitter

These simply create a new material and add it to the scene's material List. New material creates a default grey diffuse material, while New Emitter creates a 40W white emitter material.

Import MXM

This prompts you to select an MXM file from your disk. If you do, a new material is created in the scene, and the state of the chosen MXM file is imported into it.

Load MXM

This item will prompt you to choose an MXM file. Once chosen, the file will be read into this material. Any objects this material is assigned to will now be using the newly-loaded MXM data. If *Options > material Defaults > MXM Linking* is set to Linked, then this material will be set to Link the chosen MXM file.

Save As MXM

This item exports the current material as an MXM file for use in Studio or other plugins. If the material currently has no Source MXM path, then it will be set to use the path chosen here (the material will not be set to use Linked MXM mode, but it will be easy to Link the exported MXM later, if desired).

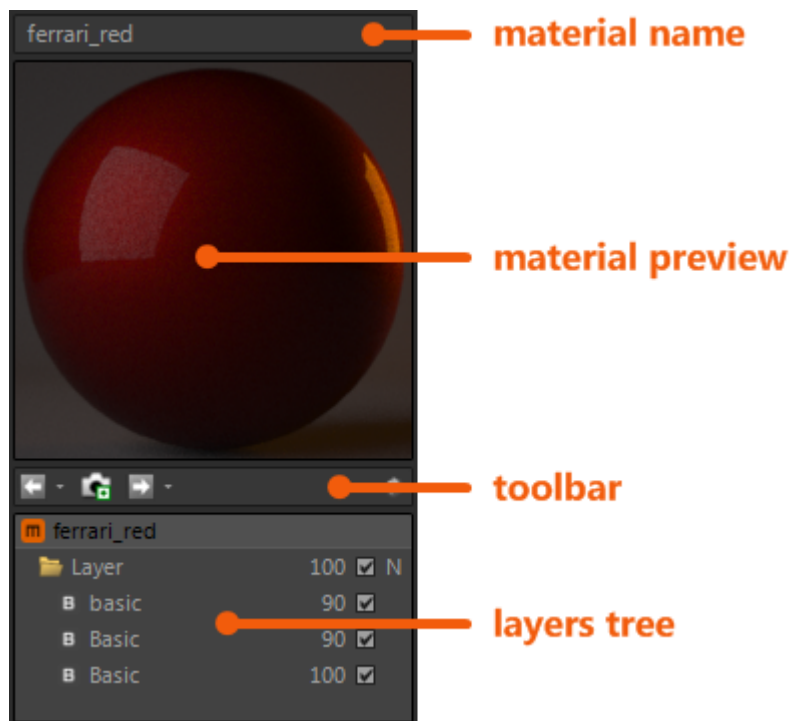
Enable/Disable Textures

These items are context-sensitive, that is, they will only appear when it makes sense for them to. As you can guess, what they do is to enable or disable all textures in the current material.

Reload/Update Linked MXM

These items are also context-sensitive, meaning that they will only appear when the material has a valid linked MXM file. When this is true, that MXM file can either be re-loaded into the material, or the current state of the material can be exported out to the MXM file using these commands.

Material Overview Pane



This portion of the Material Editor window is used to control the global state of the material.

Material Name

This simply shows the name of the current material. When the material's preview is being refreshed, it shows the current sampling level of the preview-refresh render.

Material Preview

This shows the preview image for the current material. This may be re-rendered by double-clicking on the image, or by clicking the 'Refresh Preview' button in the toolbar below.

Enable/Disable Real-time Preview

This toggles whether or not the material editor automatically refreshes the material preview whenever a parameter changes in the material. It can be desirable to disable this, for example, when Fire is currently rendering; material changes will be instantly viewable in Maxwell Fire, so you may not care so much about seeing them in real-time in the material editor as well, seeing that it will slow down Maxwell Fire to some degree.

Refresh Preview

This button either starts a preview-refresh, or quits the preview-refresh routine, if it is running in the current material.

Material Snapshots



By clicking the 'Take a Snapshot' button, you can quickly store the current state of the material. This copy then is accessible by clicking the 'Back' button. To move back to where you were, click the 'Forward' button; this is designed to work in a way similar to a standard web browser. As such, adding a new snapshot will clear any which are currently shown in the 'Forward' drop-down.

The snapshots do not operate as static copies; when you have switched to a certain snapshot, and you make edits to the material, these edits will not be lost again when moving to a different snapshot. That is to say, snapshots are 'live' materials.

material Snapshots is a runtime-only feature, meaning that these versions are not stored in the document. If you have made edits to a particular snapshot, and you wish both to move to another, but not to lose the edits you've made, simply make that snapshot current and then, in the Scene Manager's materials page, right-click the material and choose 'Clone material'. The clone will not have any snapshots, but it will be a duplicate of the current state of the material it was cloned from. After this clone has been created, go back to the original material and return its state to the snapshot you wish it to use.

It should be noted, the snapshots feature is not designed to be used when a material is using Linked MXM mode. You are not prevented from taking snapshots of such a material, but the feature is rendered somewhat useless, since moving between snapshots causes a re-load of the material, and this in turn causes a re-load of the Source MXM.

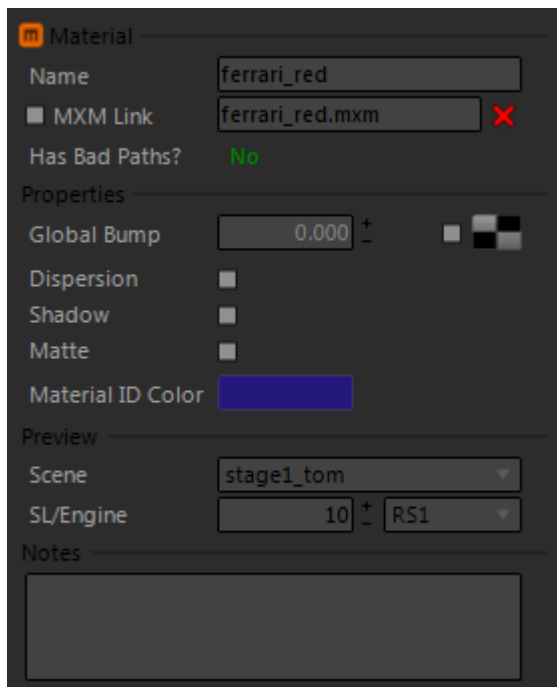
Layers Tree

Similar to the tree found in MXED, this control allows you to define the structure of the material. New components are added by right-clicking the desired parent component and choosing what to add (i.e. right-click a Layer to add a BSDF). Child components may be re-ordered under their parent by dragging them from their current position and dropping them on their parent node; this will put them at the top of the list.

To re-order components in the tree, drag an item and drop it on its parent node; it will be inserted at the top of the list. Components may also be copied by holding down CTRL while executing a drag-drop action; rather than moving the dragged component, a copy of it will be added to the target node.

Additionally, individual components may be dragged back and forth between the Layers tree and the various pages in the Database Manager; this makes it easy to build new materials from pre-defined components.

Material Page



The parameters in this page apply to the material as a whole. Each material saves its own preview options, so that different MXS preview scenes may be used for different types of materials. The MXS preview scenes listed in the scene drop-down are those found in the *Maxwell/preview* directory.

As explained in the 'How does MXM Linking work' section several pages above, a material which has MXM Linking enabled (and has a valid linked MXM path) will re-load the linked MXM when the document it is contained in is opened. When the plugin option *Behavior > Auto-update Linked materials* is enabled, such a material will also actively listen for changes made to its linked MXM.

In addition to specifying individual bump maps in each BSDF layer, it is also possible to supply a bump (or normal) map here which will be used to affect the material as a whole.

Dispersion

This switch enables the calculation of dispersion. Dispersion is the 'rainbow' effect seen when light is split based on its different wavelengths as it passes through refractive materials. Enabling this option can raise render times because of the extra math involved in calculating these wavelength-based refractions.

Matte

Materials which have this switch enabled will cause the background (i.e. Physical Sky, etc.) to show through all geometry where the objects they are assigned to are seen by the camera. This can be useful for creating composite images.

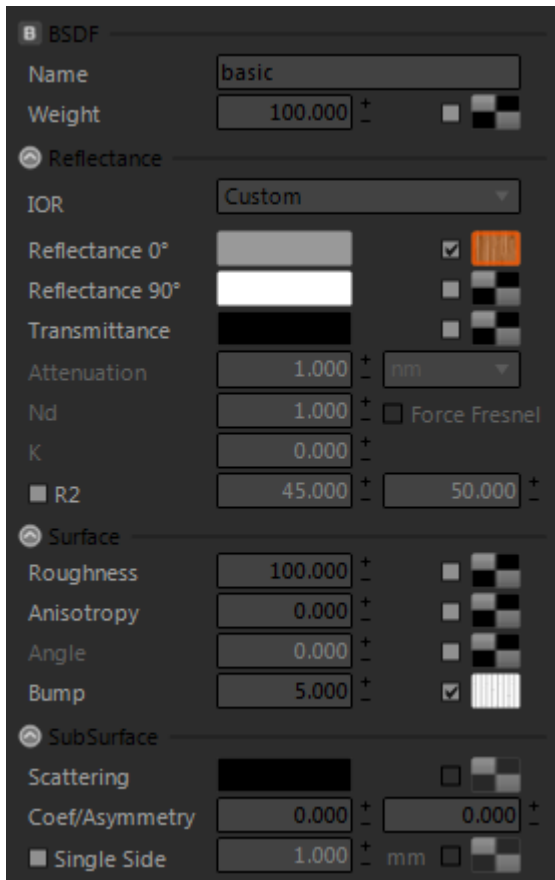
Shadow

Materials which have this switch enabled will have the objects they are assigned to casting shadows when the Shadow channel is rendered (see *Output > Render Channels > Shadow*).

Material ID Color

This is the color which will be used for this material when the Material ID channel is rendered.

BSDF Page

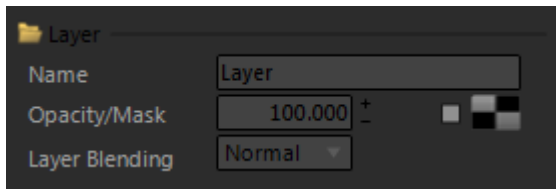


The operation of this page is similar to that of the BSDF page in MXED. When certain parameters in the page are grayed-out, it means that they would have no effect; they will become enabled when other parameters have been changed in such a way as to make them effective.

As the BSDF page is taller than the minimum-height of the Material Editor, it includes collapsible Reflectance, Surface, and SubSurface groups; their operation is the same as that of collapsible groups in the Scene Manager, and the plugin remembers the expanded/collapsed state of each group across sessions. When the Material Editor is sized in such a way as to hide part of the BSDF page, you can 'grab' the page by clicking down the left mouse button and then drag it up and down. By design, it does not show a scrollbar, due to the limited space available, and also because if it did, the mouse-wheel adjustments made to numeric textboxes would scroll the page up and down rather than adjusting the value.

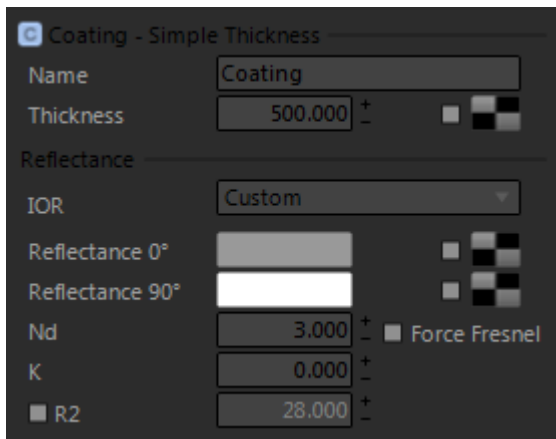
When *Reflectance* > *IOR* is set to 'Measured Data', all of the parameters in the Reflectance group will be hidden, since they have no effect in this case. Similarly, the min/max Thickness controls in the SubSurface group are only shown when (a) Single Side SSS is enabled, and (b) an SSS Thickness map is assigned and enabled.

Layer Page



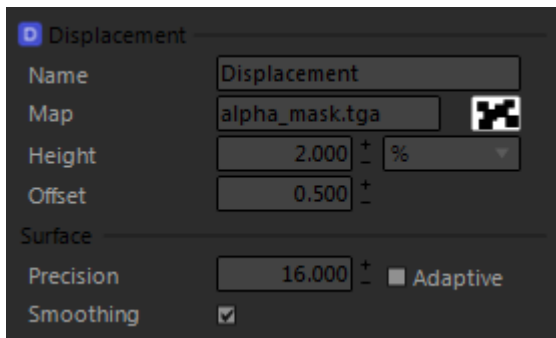
The operation of this page is similar to that of the Layer page in MXED. As a shortcut, the Layer Blending parameter may be set either by using the drop-down list in this page, or by clicking the small *A/D* toggle in the Material Editor Layers Tree.

Coating Page



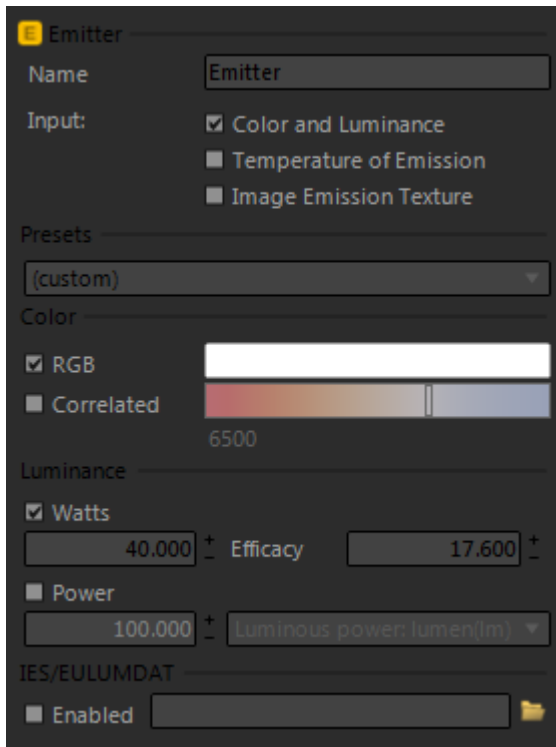
The operation of this page is similar to that of the Coating page in MXED. Similar to the BSDF page, the parameters in the Reflectance group will be hidden when *Reflectance* > *IOR* is set to 'Measured Data', since they have no effect in this case.

Displacement Page

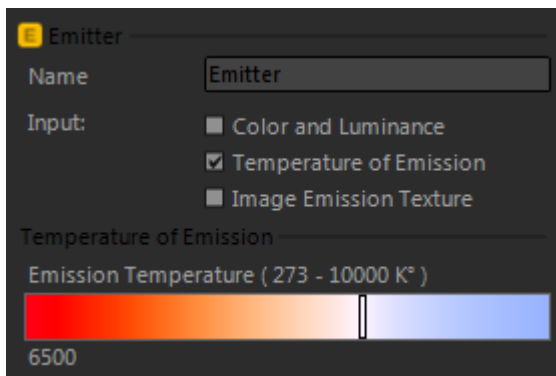


The operation of this page is similar to that of the Displacement page in MXED. Note that while MXED allows a separate Displacement layer to be added to each Layer, the plugin does not, instead placing Displacement at the Material-level. Functionality is similar, because in MXED, one of the several possible Displacement layers must be chosen – in other words, only one Displacement may be specified for the material. The plugin therefore chooses to display this relationship as it is in the layers tree.

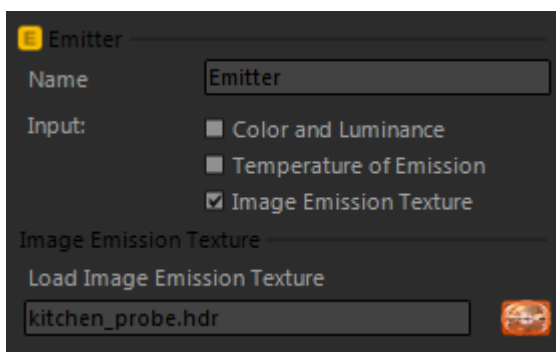
Emitter Page



The operation of this page is similar to that of the Emitter page in MXED. The contents of the page change based on which Input type is currently selected. When *Color and Luminance* is selected, the page will appear as shown above. When *Temperature of Emission* is selected the interface will show a single slider used to set the emitter output in Kelvins:

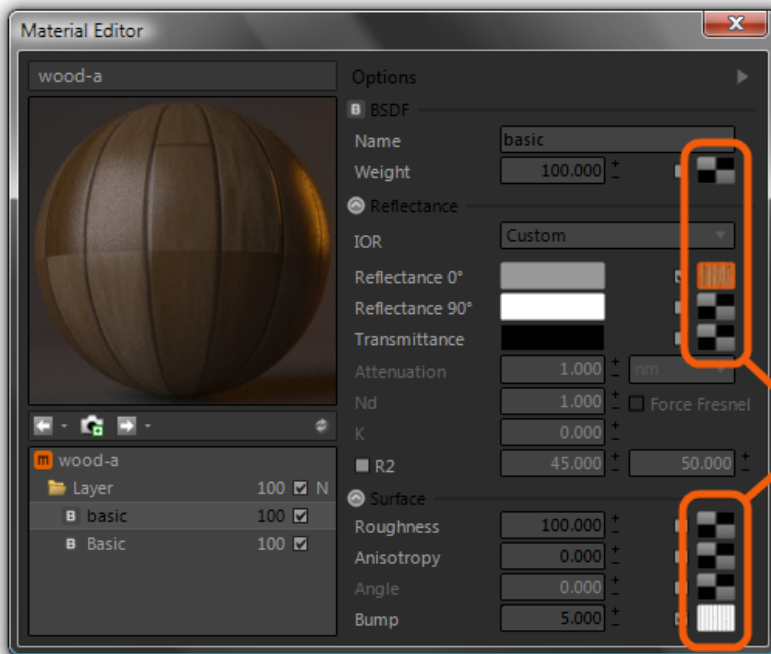


When *Image Emission Texture* is selected, a path/texture button will be shown; as with other textures, this texture may shown in the viewport by left-clicking the texture button, and it may be edited using the Material Editor's texture-editing pane.



Texture Editor

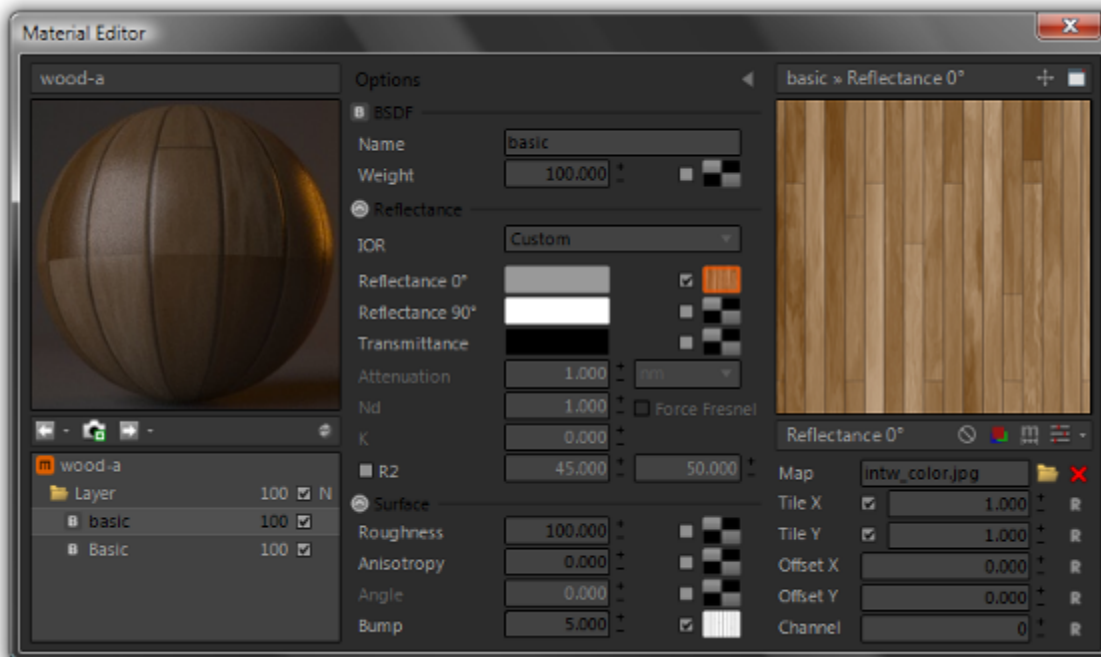
To open the Material Editor's texture-editing panel, click the arrow button at the Material Editor's top-right corner, or right-click on a texture button:



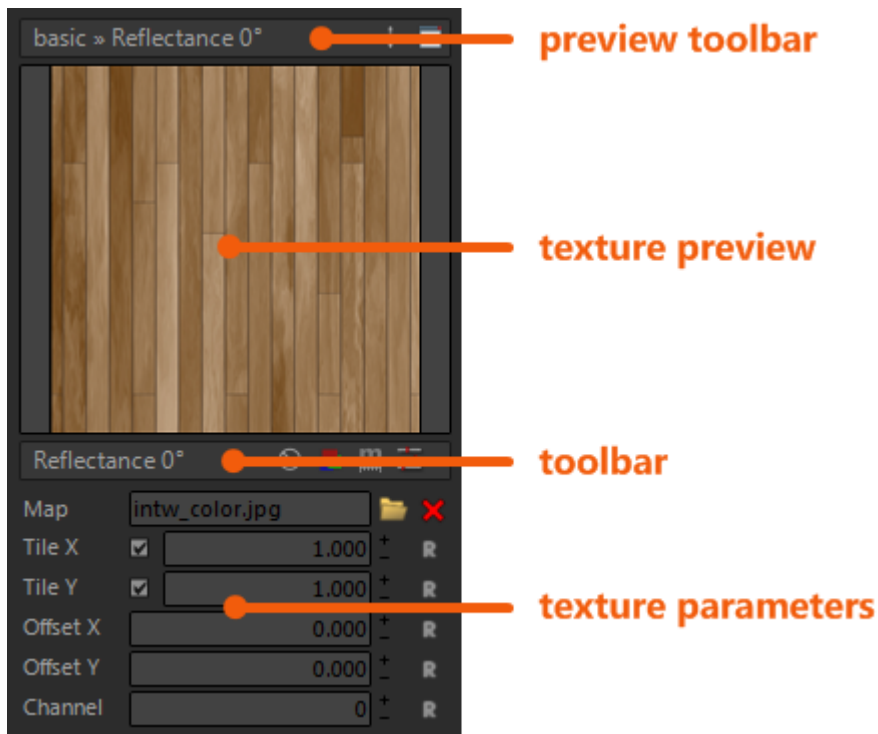
left-click to
select texture

right-click to open
texture editor

A material has, at most, one texture which is currently selected – which one is selected is changed by left-clicking on the corresponding texture-button (the button will take on a highlighted border). The selected texture will both be shown in the viewport, and in the texture editor. For example, in the material shown above, the first BSDF's Reflectance 0 texture is the material's selected texture, and right-clicking its button opens the texture editor panel, showing this texture and its parameters:



The texture editor has four main elements:



The preview toolbar shows the location of the current texture, and provides buttons to fit the texture into the available preview area, and to open the texture in your system's default editor for the texture's file type (an error may occur if there is no default editor for this file type).

The texture preview shows the selected texture, including the effects of tile, offset, RGB-inversion, and the plugin's saturation/contrast/brightness/RGB clamp controls. You can zoom & pan around the image using the mouse-wheel and left-mouse button.

The texture toolbar contains toggles for several texture options:

- **Interpolation:** enables Maxwell's cubic pixel-interpolation in the render
- **Invert:** inverts the RGB values of the specified bitmap in the render
- **Real Scale:** changes the meaning of the Tile X/Y parameters so that they refer to absolute meter-based size
- **Imaging Controls:** provides a drop-down which contains saturation/contrast/brightness/RGB clamp controls

In addition, when the selected texture is a bump map, another toggle appears which enables you to indicate that the specified map is a normal-map. When using a normal-map, three toggles are supplied: Flip X, Flip Y, and Wide Z. These are used to change how Maxwell interprets the color values encoded in the normal map, since different software uses different conventions regarding direction and bit-depth.

The remaining parameters control how the texture is scaled and translated in texture-space. Tile X & Y can be enabled or disabled using the appropriate checkboxes, and their values are set using the numeric input box. When Real Scale is not enabled, the meaning of Tile X & Y is basically, 'squeeze this many copies of the texture into the given space'. When Real Scale is enabled, the meaning becomes, 'make each tile x-number of meters wide'.

Offset X & Y simply translate the texture along the x & y axes by either the given percentage, or by the absolute meter-size when Real Scale is enabled.

The Channel parameter allows you to associate different textures with the different texture-projections which may have been defined for the objects to which this texture is assigned. For example, some textures may use a cubic projector on channel 0, and a planar projection on channel 1. Each texture mapping, channel 1 & channel 2, represents a different set of UV coordinates. Setting the Channel parameter in each texture individually allows each texture to use whichever mapping is appropriate. If an object has fewer texture-mappings assigned than are specified in the various textures its material has, then copies of the channel 0 mapping will be created

during export in order to accommodate those textures. This takes extra processing power, so it is always best to make sure that textures only specify as many different Channels as texture-mappings exist for objects which use them.

As a convenience, holding down the CTRL and/or SHIFT keys while adjusting texture parameters will affect where the changes take effect:

- **CTRL:** the changes will be applied to all textures in this texture's parent layer
- **CTRL+SHIFT:** the changes will be applied to all textures in this texture's parent material

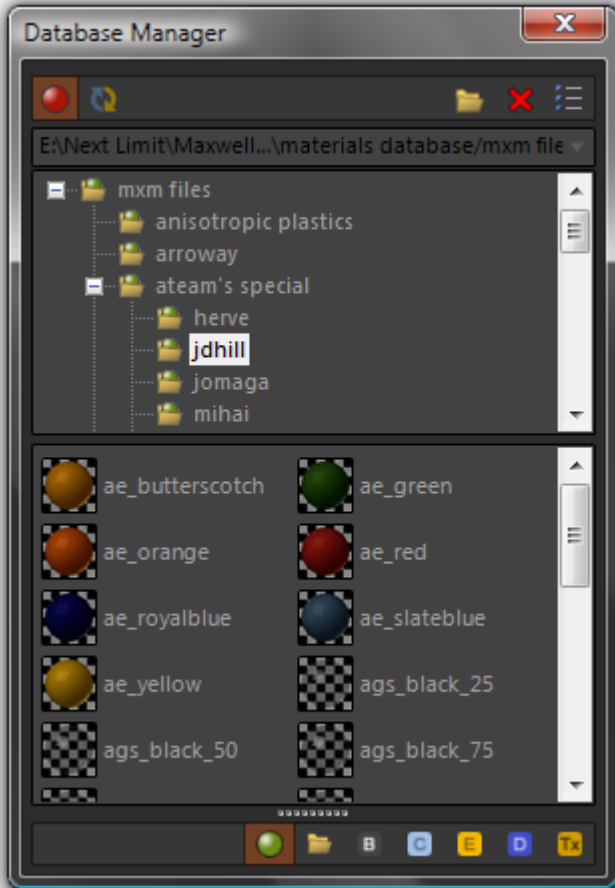
To adjust numeric inputs, you can either:

- type in a new value and move focus to another input
- click the small +/- buttons next to the input box to adjust the value
- click in the desired input box and roll the mouse-wheel up or down to adjust the value
- click down on the desired input box and move the mouse forward & backward to adjust the value

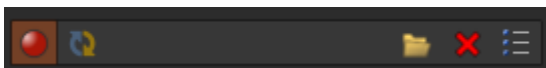
In the first case, the value will be changed as soon as focus moves elsewhere; this allows you to type in a new value, then hold down CTRL and/or SHIFT to execute the change in multiple textures as described above.

In addition to these conveniences, textures may also be drag/dropped from one texture to another. To do this, simply click down on the desired texture button and begin dragging it. This texture may now be dropped on any other texture button in this layer, or in another layer. To select another layer during the drag, drag the texture over the Material Editor's layers tree to select a different layer; once the desired layer is visible in the Material Editor, drag the texture over and drop it on the desired target. If you simply drop it, only the texture's path will be copied to the target; if you hold down CTRL while dropping the texture, then all of the parameters will be copied from the source texture to the target.

Maxwell for Rhino :: Database Manager



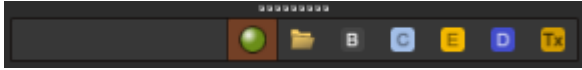
The Database Manager provides access to MXM files stored on your machine, as well as collections of individual material components (i.e. BSDFs, Emitters, etc.). Similar to the Scene Manager, the Database Manager may be used either as a standalone window, or as a Rhino dockbar, by using *Options > User Interface > Dockable Database Manager*. The top toolbar in the Database Manager is context-sensitive, meaning that the buttons it has take on different meanings depending on which page is currently selected:



The buttons on the left do not change, and they have the following functions:

- **Enable Real-time materials:** when this is checked, the plugin will adjust the viewport appearance of materials in real-time as they are modified in the Material Editor. If it is not enabled, the plugin will not alter the appearance of objects in the viewport
- **Refresh Viewport materials:** for various reasons, it may occur that the viewport appearance of Maxwell materials may become unsynchronized. When this happens, or when 'Enable Real-time materials' is not enabled, using this button will bring the viewport up to date

The Database Manager window contains seven pages, which are accessed using the buttons contained in the bottom toolbar. These seven pages are as follows:



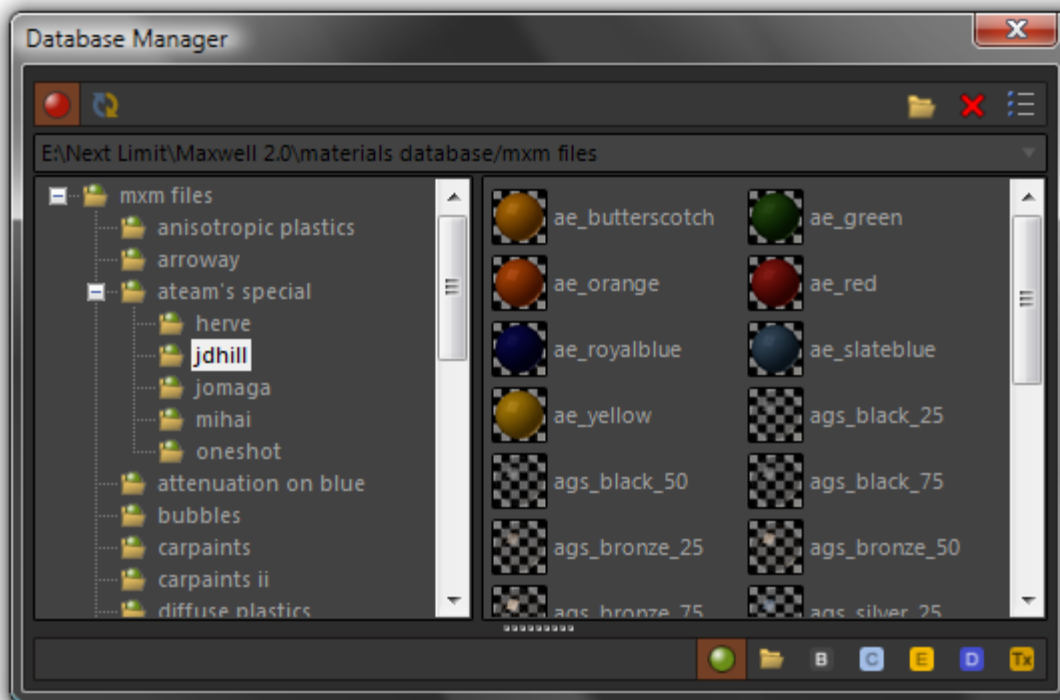
- **MXM Browser**
- **Layers**
- **BSDFs**
- **Coatings**
- **Emitters**
- **Displacements**
- **Textures**

MXM Browser

The MXM Browser, shown above, is unique among these pages. It contains:

- a drop-down which holds user-defined directories which contain MXM files
- an Explorer-style folder tree which shows the folder-structure of the selected directory
- an Explorer-style thumbnail-viewer which shows the MXM contents of the directory selected in the folder tree

To add a new directory to the MXM Folders drop-down, click on the Browse button in the top toolbar. You will be prompted to select a directory, which will then be added to the drop-down list. Selecting an MXM directory from the drop-down list will cause it to be shown in the tree. Selecting a folder from the tree will cause the MXM files contained in it to be shown in the thumbnail-view below the tree. In the tree, folders which contain MXM files are shown using a different folder-icon than those which contain none. When the Database Manager is wider than it is tall, then the separator bar between the folder-tree and the thumbnail-viewer will switch to a vertical orientation:

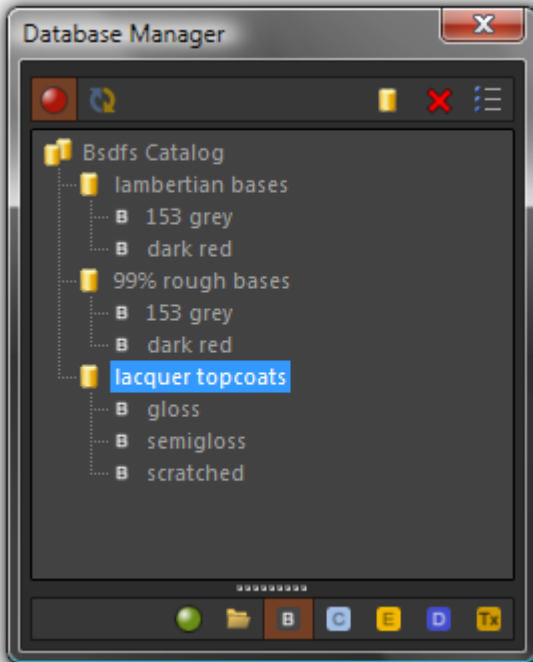


MXM files may be imported by dragging them from the thumbnail-view into either the Rhino viewport, or the Scene Manager's materials page. Additionally, multiple MXM files may be imported by windowing them in the thumbnail view, or by dragging an entire folder from the folder-tree into the Scene Manager's materials page. materials may also be exported from the scene as MXM files by dragging them from the Scene Manager's materials page into the thumbnail-view of the MXM Browser.

Layers/BSDFs/Coatings/ Emitters/Displacements/Textures

These pages all work similarly. At the top of the hierarchy in each page is a 'Catalog' node; as such, each page represents a catalog of items. There are only two other types of nodes used in these trees:

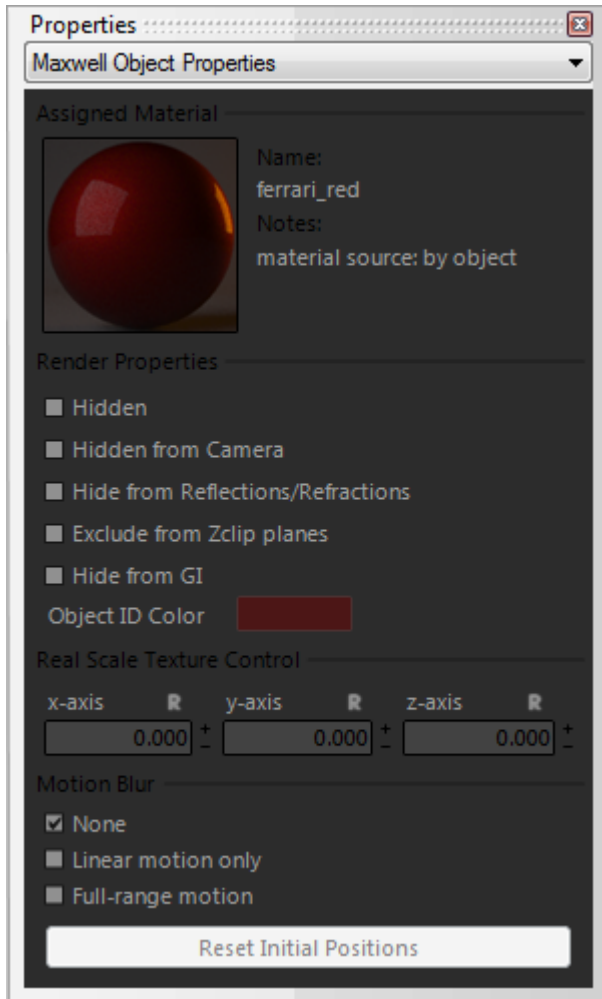
- **Volumes:** these contain either other volumes, or components
- **Components:** an individual BSDF, Emitter, Coating, etc.



Components can be dragged back and forth between the Material Editor's layers tree and their respective Database Manager page. They can also be edited in-place while stored in the database by double clicking them in the Database Manager; the Material Editor will be emptied, and the selected component will be loaded into the appropriate Material Editor page.

The components stored in the database are saved automatically whenever a document is saved, and there is no need to be concerned with the files they are stored in; these files are plugin-specific and located somewhere in the app-data folder under your user profile. This location is used to make sure the plugin has permission to write these files. In addition to being saved when the current document is saved, there is also a 'Save Changes' button which will appear in the Database Manager's top toolbar whenever changes have been made which would be lost were the document to be closed without saving – clicking the button will commit the current state of all database components to disk, so it is possible to save them without also saving the document.

Maxwell for Rhino :: Object Properties



The Maxwell page of Rhino's Object Properties window shows information about currently-selected objects:

Assigned material

This section shows which material, if any, is assigned to the selected object or objects. It also indicates whether the material is assigned by object, parent, or layer. Clicking on the preview image opens or closes the plugin's Material Editor.

Render Properties

This section contains checkboxes for enabling the various Maxwell object properties:

- **Hide:** completely hide this object in the render
- **Hide from camera:** hide this object from being seen directly by the camera
- **Hide from Reflections and Refractions:** hide this object from reflection/refraction calculations
- **Hide from GI:** hide this object from the global-illumination calculation
- **Hide from ZClip:** hide this object from Maxwell's Z-Clip feature
- **Object ID Color:** the color of this object when the Object ID channel is rendered

Real Scale Texture Control

When the assigned material's selected texture has Real Scale enabled, and that texture uses Channel 0, then these x/y/z inputs may be used to adjust the attitude of that texture's mapping. Real Scale mappings always use a cubic projection, so adjusting these values has the effect of rotating the theoretical mapping cube. To learn more, see the Real Scale tutorial.

Note that there is one very special case in which the Real Scale texture control's Z-Axis value is used for a very different purpose, and it has to do with experimental support for IES emitter directionality in this version of the plugin. When a Rhino spotlight is selected in the object properties page, and an IES-enabled emitter material is assigned to it, the Z-Axis value here may be used to rotate the IES lobe around its axis. This is useful because many IES profiles are asymmetrical. In the future, this functionality will be augmented and moved out of this Real Scale texture control (it is only used now since the feature is still experimental) and placed into an emitter-specific object properties page.

Motion Blur

The plugin supports Maxwell motion-blur using two different strategies:

- **Linear motion only:** for objects which will only move in a straight line, use this mode to save memory
- **Full-range motion:** for objects which will rotate or move in a non-linear way, this mode stores info for each mesh point

When these are enabled for an object, the plugin stores the object's current position for use during export. At that time, the object will be written into the MXS file using both the stored position, and its position at export-time. Maxwell will figure motion-blur based on the difference between these two positions.

At any time, an object's 'initial' position may be manually reset by selecting the object and clicking the 'Reset Initial Positions' button. During an animation, both the 'initial' and 'current' object-positions will be figured automatically, frame-to-frame, by the plugin.

Maxwell for Rhino :: Maxwell Fire



The new Maxwell Fire interactive rendering feature is accessed using this window. Using Maxwell Fire, it is possible to view changes to materials, environment, etc., interactively as you work with the model. This greatly improves the general Maxwell workflow, as it removes the need to do many incremental exports and test renders.

The plugin's Maxwell Fire window has been designed to provide this functionality in as simple a manner as possible. At the top of the window you will find a toolbar which contains the three main commands used to control Maxwell Fire:

- **Enable Preview:** enables and disables the engine; if Cache Meshes is disabled, disabling invalidates the scene
- **Refresh Scene:** re-exports the scene, including changes to geometry, then re-starts the engine
- **Save Image to Disk:** save the Fire-rendered image to disk (saves at the actual rendered resolution)
- **Enable in-viewport display:** displays the rendered image directly in the appropriate application viewport
- **Lock to Current Camera:** prevents the plugin from dynamically changing the camera to that of the current viewport

Additionally, the Fire window contains a context menu, which is available via right-click when Maxwell Fire is enabled. The options here are fairly self-explanatory, as to their respective purposes:

- **SL:** the sampling level at which the engine will stop rendering
- **Quality:** the actual resolution at which to render; the image is interpolated to the Fire display size
- **Threads:** the number of threads which will be used by the engine (zero means auto-detect)

General Notes on Maxwell Fire Usage

Maxwell Fire functionality is tightly integrated with the plugin Export option *Cache Meshes*. When Fire is active, the plugin's behavior is identical to what it would be when Cache Meshes is activated; the reason for this is that it is neither necessary nor desirable that the scene's geometry be re-exported, unless it has actually been altered. As such exporting an MXS, or rendering in Maxwell Render, when Fire is active requires no unnecessary geometry-export.

As it is the case that geometry changes (note that this includes UV coordinates, motion-blur positions, etc.) are not currently able to be handled interactively, the question arises as to what is; the answer is: everything else. Namely, changes to materials, material assignments (except for block instances), changes to environment, render output settings, etc. – these are all fully interactive.

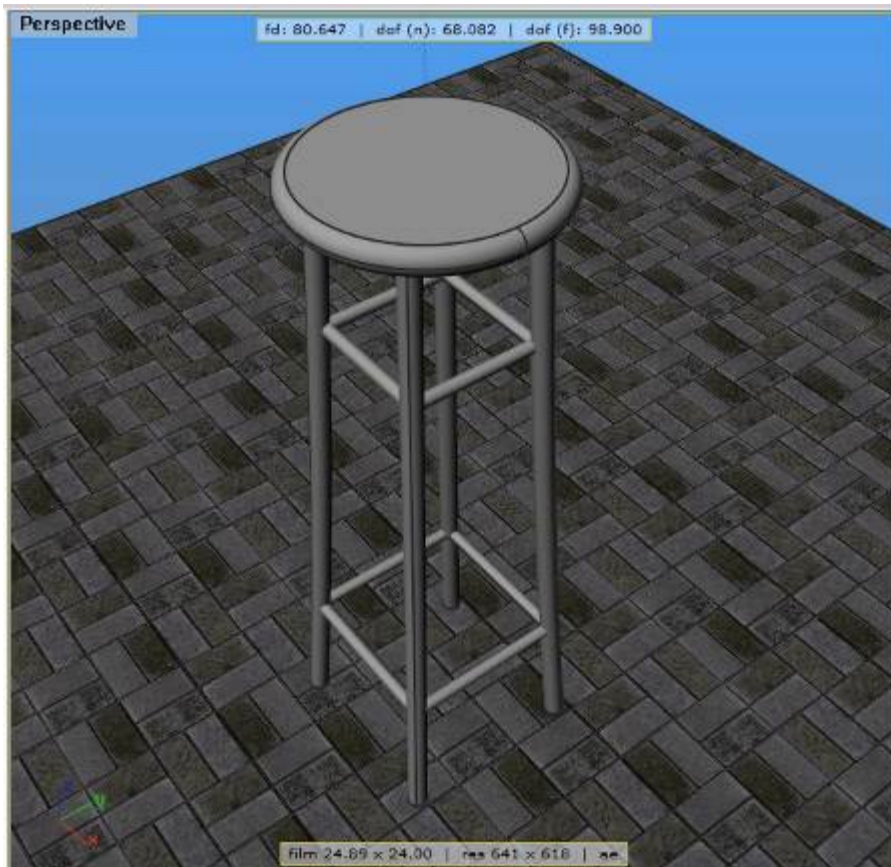
Note that the *Save Image to Disk* button is only enabled when an image exists to save. Additionally, the plugin provides a Rhino command, *Maxwell_SaveFireImage*, which allows for outputting simple camera animations using Maxwell Fire. It provides two parameters, *Path* and *Sleep*. While *Path* is used to set the location where the image will be saved, *Sleep* directs the plugin to wait a given number of seconds (in order to allow Maxwell Fire to render the image) before writing the image.

Maxwell for Rhino :: Real Scale Tutorial

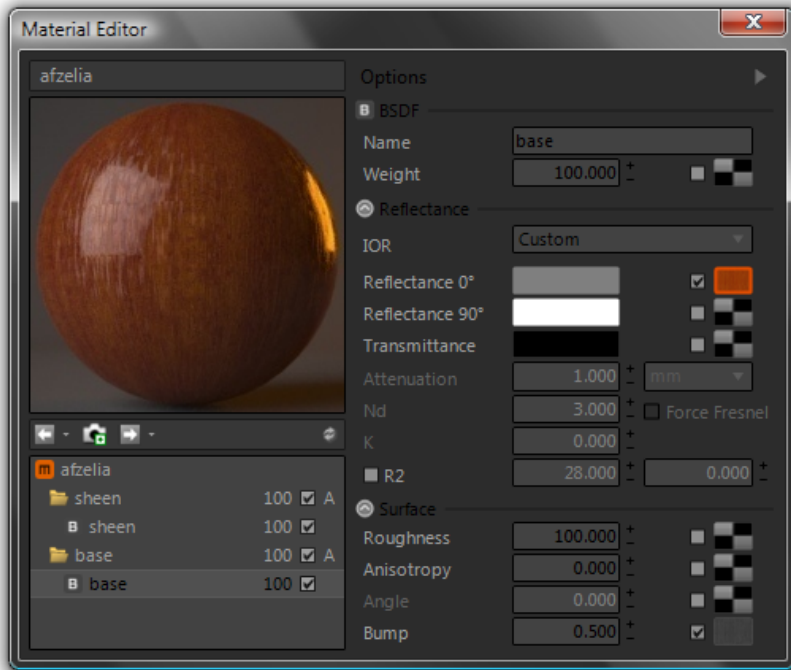
In many texturing scenarios, it is necessary to ensure that differently-sized objects receive commonly-sized texture coordinates.

Consider, for example, assigning a wood grain texture to a piece of furniture; although the various objects which make up the piece have very different sizes, the size of a given wood grain does not. Therefore, it is necessary that the size of any texture maps assigned to these objects may be adjusted on a per-object basis in such a way that the apparent size of the wood grain is the same, as projected on each object. Rhinoceros does not offer any native texture-mapping mode which would accomplish this, other than selecting the entire piece and assigning a single texture mapping. This does not help us in this scenario because we also need to be able to control the orientation of the wood grain on a per-object basis. So we are left with only one real option: assign separate mappings to each object, then go through and re-size each one so that they all have the same size in world coordinates. This is an extremely tedious process.

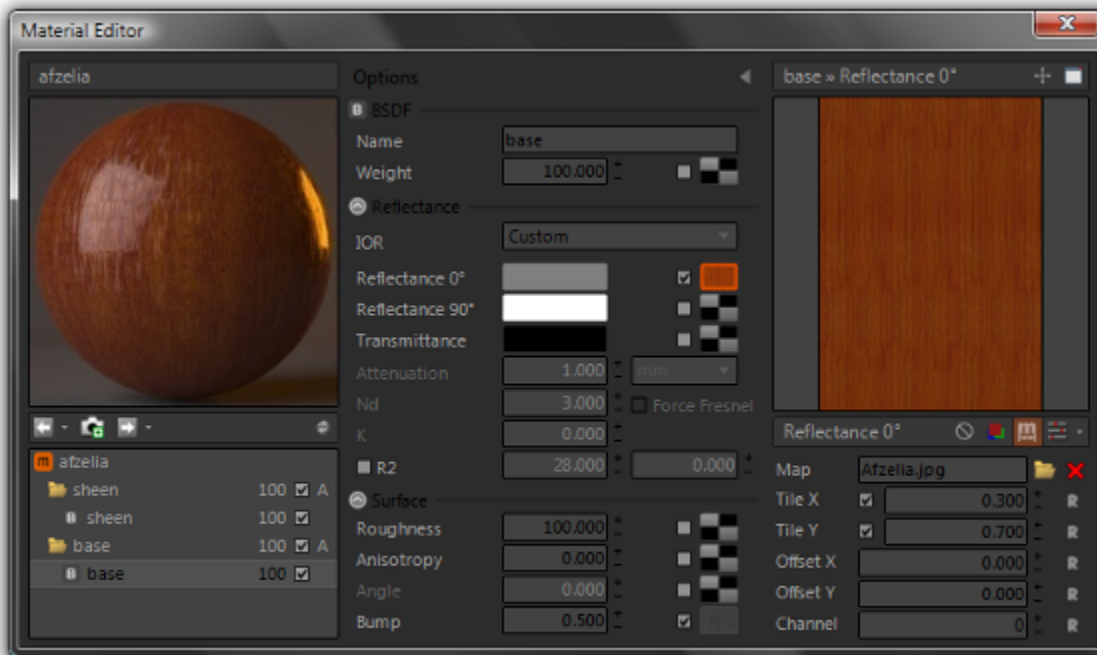
To this end, the Maxwell for Rhino plugin provides a tool which allows for addressing these issues in a way which, while being nearly automatic, still provides the necessary control to orient textures on individual objects. This feature is called Real Scale. To demonstrate how this works in practice, let's begin with the tutorial file 'Real Scale.3dm':



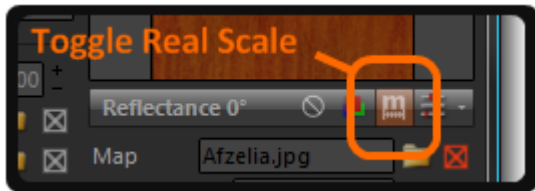
The objective will be to map this stool with a wood grain texture which is sized and oriented correctly using the Maxwell for Rhino plugin's Real Scale functionality. For this tutorial, I began by downloading an MXM file from the Maxwell MXM Gallery (<http://mxmgallery.maxwellrender.com>). Once the MXM file has been unzipped, it can be imported into the plugin's database by right-clicking in the Scene Manager's materials window and choosing 'Import MXM'. To begin with, the material looks like this in the plugin's Material Editor:



To begin editing the material's textures, select the 'base' BSDF and right-click the texture-button for its Reflectance 0° texture. This will open the texture editor and set the Reflectance 0° as the selected texture. The texture editor may be closed again, also by right-clicking on any texture button.



Real Scale is enabled by toggling-on the 'Enable Real Scale' button in the texture editor's toolbar.

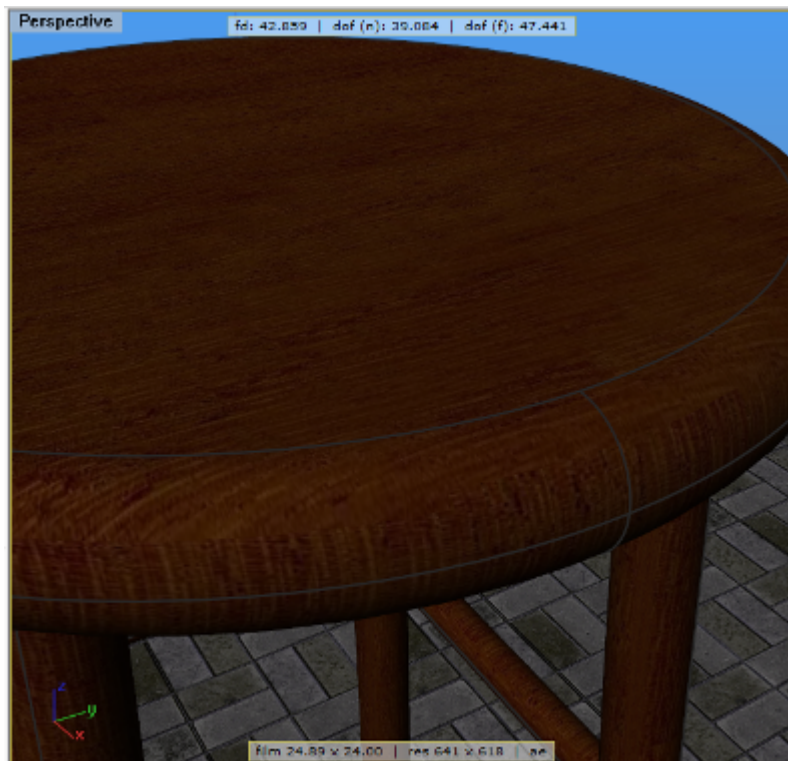


Before doing so, however, hold down the CTRL and SHIFT buttons simultaneously; when this key-combination is used for parameters in the texture-editor, it will cause the change being made to the texture currently being edited to be applied to all textures in the entire material. If only CTRL is pressed, the change will be applied to all textures in the current layer (BSDF, Coating, etc.).

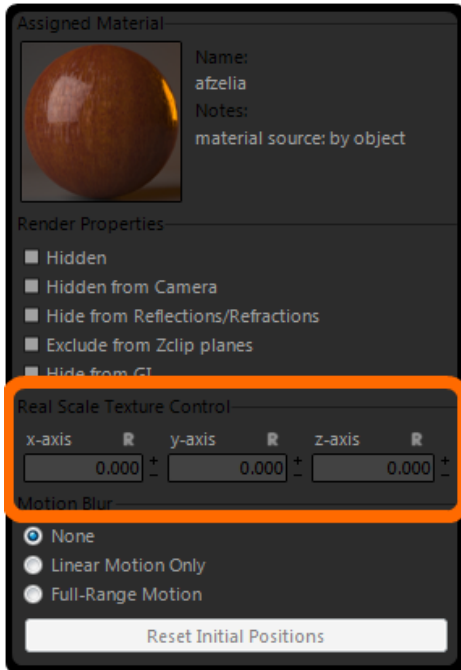
In addition to enabling the Real Scale feature, it is good now to also (while holding down CTRL+SHIFT) adjust the texture's Channel to 0. If it already is 0, use the mouse wheel to adjust it to 1, then back again to 0. This is done to quickly make sure that all textures in this material are using Channel 0 - this is important because the plugin will only enforce Real Scale texture-sizing for textures which (A) have Real Scale enabled, and (B) use Channel 0. For textures which do not use Channel 0, enabling Real Scale will only have the effect of reversing the meaning of the texture's Tile X and Tile Y parameters.

Speaking of Tile X and Y, when Real Scale is not enabled, the basic function of these parameters could be described as meaning *'squeeze X-number of copies of the texture into the size of the UV mapping size'*. Conversely, when Real Scale is enabled (and the Channel is 0), the meaning becomes *'make each tile X-meters in width'*, where the Tile X and Y values represent the actual size, i.e. a Tile X of 0.25 means that each tile should be one-quarter of a meter in size along the texture's X-axis.

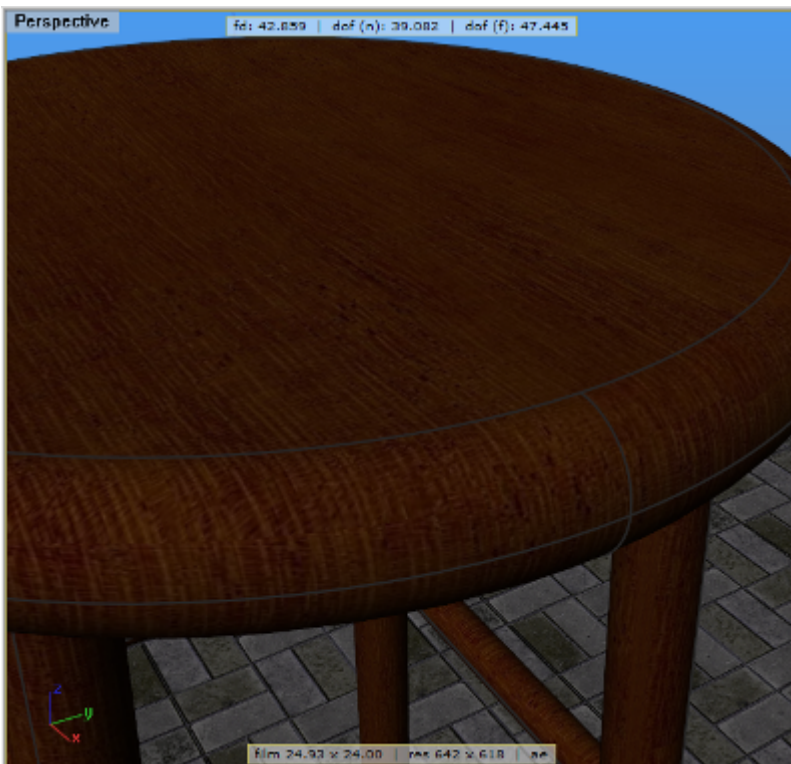
Moving along, select all of the pieces of the stool, right-click on the material's thumbnail (in the scene Manger's materials window), and choose 'Assign to Selected'. As applied, the top of the stool looks like this:



While the plugin has set the size of the textures to be consistent, their orientation leaves something to be desired. To remedy this, select the top of the stool, open the Rhino Object Properties window, then from the pull-down, select 'Maxwell Object Properties'. In the middle of the window, we find a section named 'Real Scale Texture Control', which has axis-rotation controls for X, Y, and Z axes:



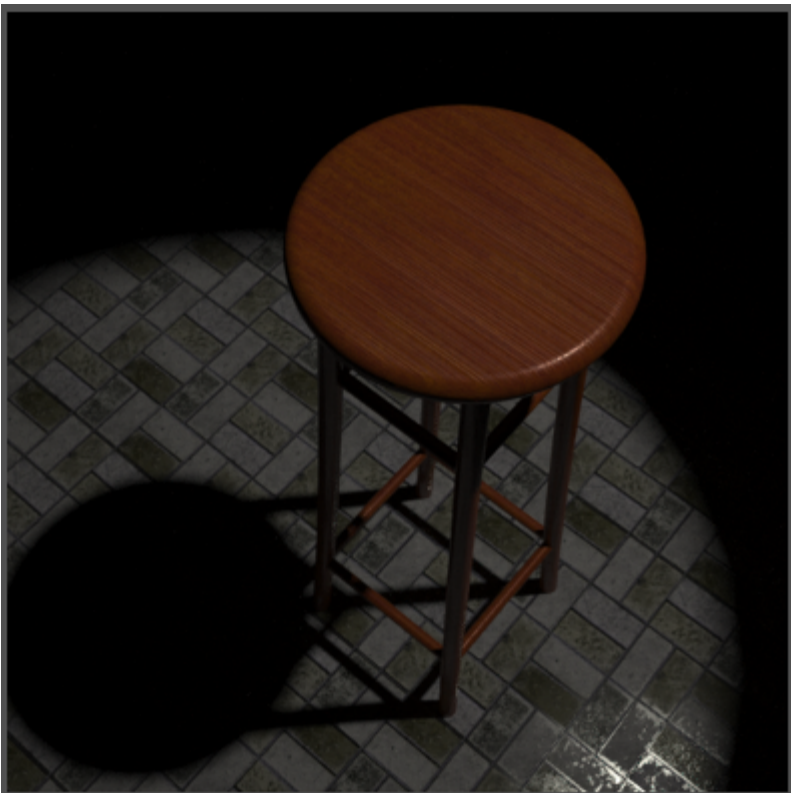
In the case of this stool, adjusting the X and Z axes to 90° achieves a better texture alignment:



It is worth mentioning here, Real Scale always uses a cubic texture-mapping. The reason for this is that only cubic and planar mapping types have a logical ability to map objects at a globally-constant size. This means that there are shapes which do not lend themselves to this type of usage, and in these cases it is still necessary to map the objects manually. After adjusting the texture-orientations for the other pieces, we have a finished product:



Which we can now render:

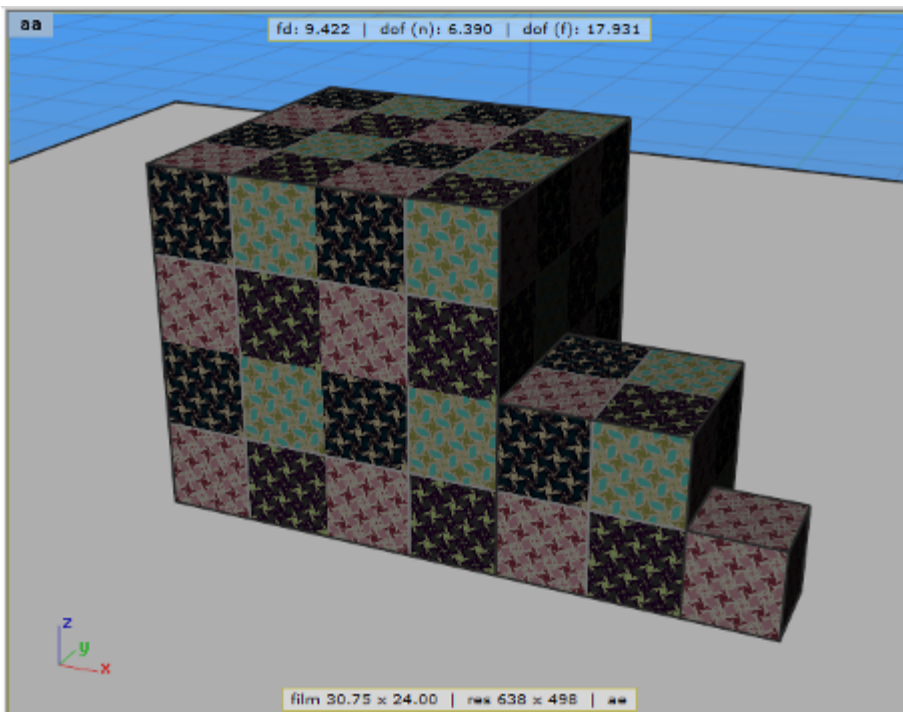


The floor in this scene was also mapped in the same way; after downloading an MXM from the MXM Gallery, it was adjusted to use Real Scale and Channel 0, assigned to the floor surface, and then the texture size was adjusted by changing the Tile X and Y parameters while holding down CTRL+SHIFT.

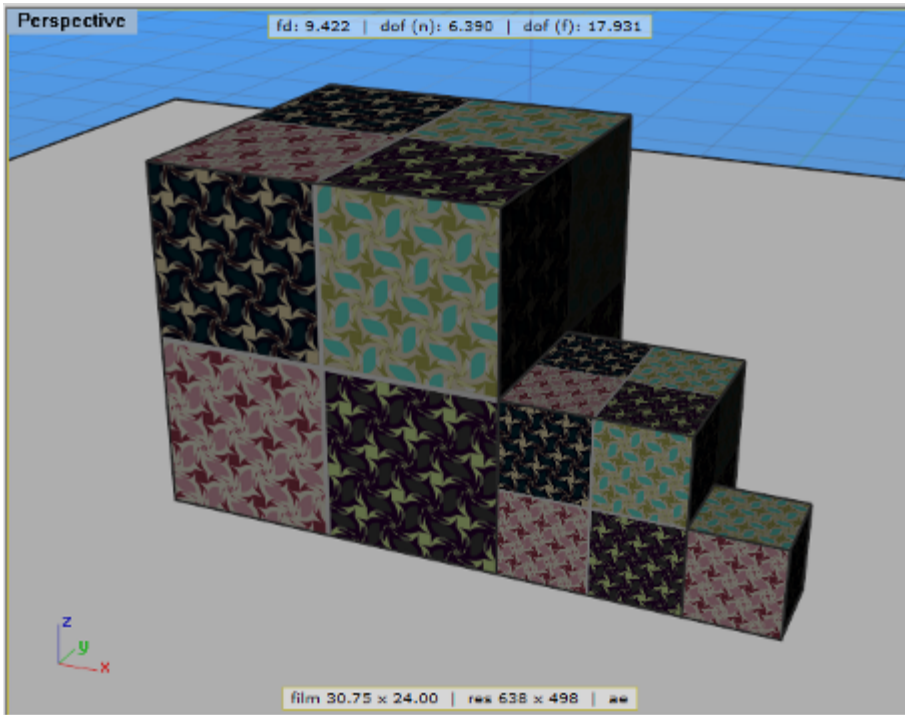
The main difference between using the explicit texture-mapping tools provided by Rhino and using the plugin's Real Scale feature is how the mapping is affected when the object it is working on is changed, i.e. moved, rotated, or scaled. When you use a Rhino mapping, the size and orientation of which you have carefully adjusted, you stand to lose your sizing work when you modify the geometry it is mapping. Generally speaking, if you scale the geometry, the mappings will also be scaled, and any real-world correspondence to the texture-sizes will be lost. If, after scaling the mapping, you move or rotate the object, the mapping may move away from the object in unpredictable ways.

This is not the case with the plugin's Real Scale feature. You may move, rotate, and scale objects however you wish; the Real Scale mapping will be maintained, and the orientation set using the Real Scale Texture Control will not change. It is still possible to make changes to geometry which will require the use of the plugin's 'Refresh Viewport materials' command (in the Functions toolbar); this is due to the fact that the plugin listens for changes to geometry, but there are classes of change which do not cause a notification. Even so, though the viewport-appearance may temporarily become unsynchronized, the underlying data is tied permanently to the object it is working on.

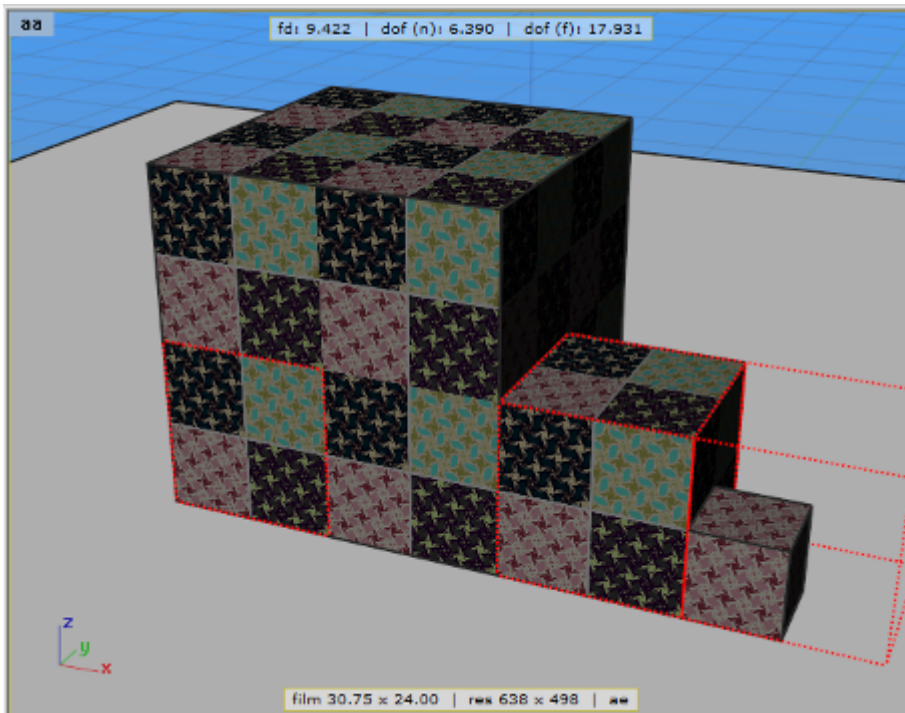
To demonstrate, here are three cubes, 2.0m, 1.0m, and 0.5m, with a Real Scale texture assigned and no explicit Rhino texture-mappings. As you can see, the map has four quadrants, and has Tile X/Y of 1.0:



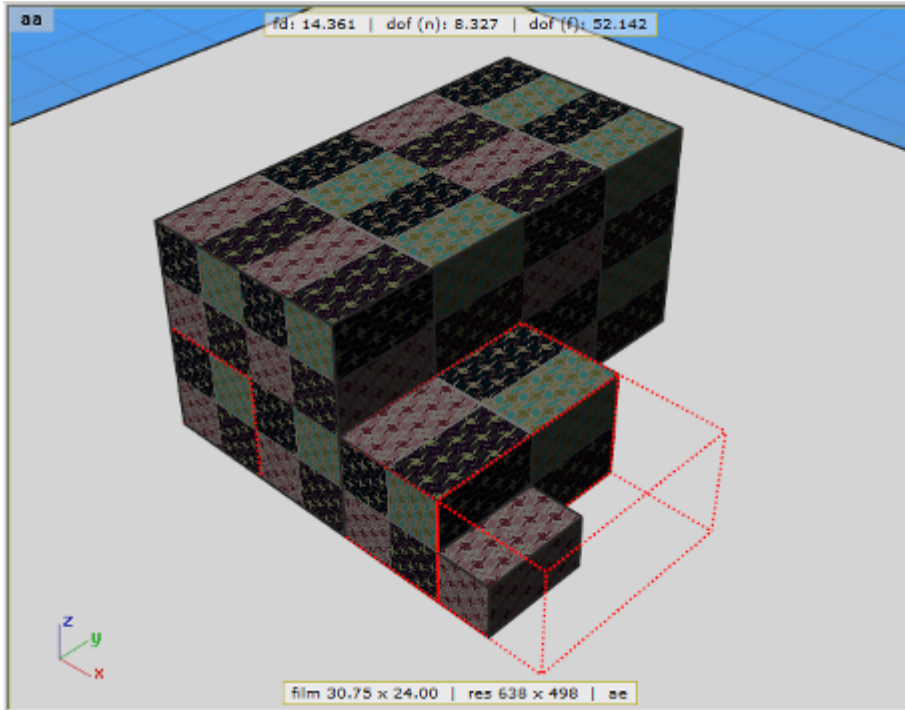
Now here are three identical cubes, using a material which is identical to the first, but with Real Scale disabled. Each cube has had a box projector assigned to it by way of Rhino's texture mapping window:



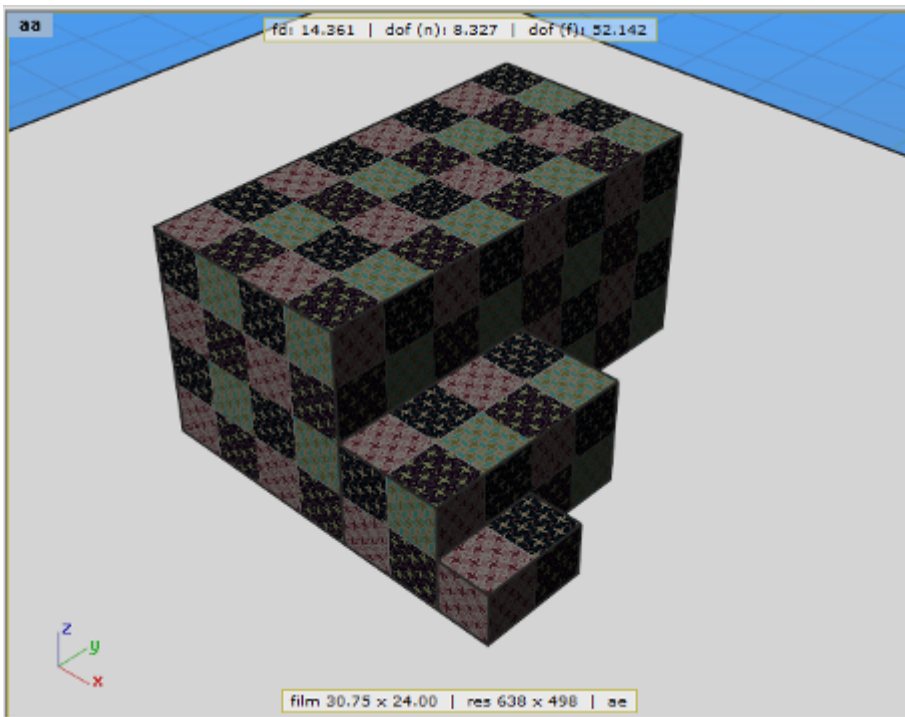
As you can see, the default behavior of a Rhino box projector is to size itself according to the bounding box of the object to which it is applied. This is not our desired result, so it is necessary to return to the texture-mapping window, set the Size parameter to 1.0 x/y/z for each of the cubes, and re-position the origin for the mapping boxes so they line up with the cube-corners:



So we have now achieved a 1-tile-per-meter mapping using both the Real Scale feature and the built-in box mapping provided by Rhino. What will happen if we now scale these cubes in one dimension? A problem shows up right away...



The mappings have also been scaled. This is not the case with the cubes being mapped by the plugin:



The plugin mapping is based on a fixed point on each box, rather than the arbitrary location used by the Rhino mapping. This is exactly why the plugin must provide you with the Real Scale Texture Control x/y/z axis controls in the Maxwell Object Properties page – without them you would have no control over the texture orientation. The mapping scale is not (as with the Rhino mapping) modified when the boxes are scaled; instead, the plugin detects the change and re-figures the correct mapping.

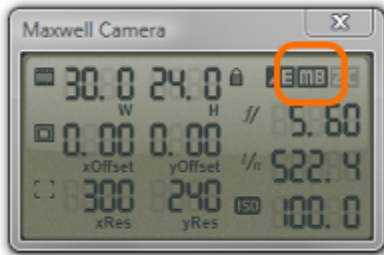
Maxwell for Rhino :: Motion Blur Tutorial

The Maxwell for Rhino plugin provides tools for accessing Maxwell Render's motion blur capabilities. There are two types of motion blur available in Maxwell:

1. **camera:** blur based on the difference in position of the camera from frame to frame
2. **object:** blur based on the difference in position of objects from frame to frame

Camera Motion Blur

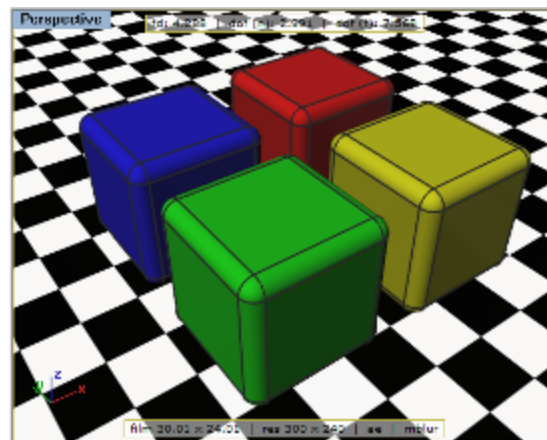
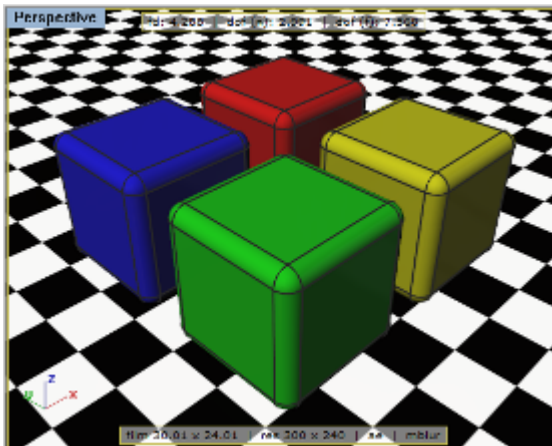
As such, each camera created in the plugin has a switch for enabling blur:



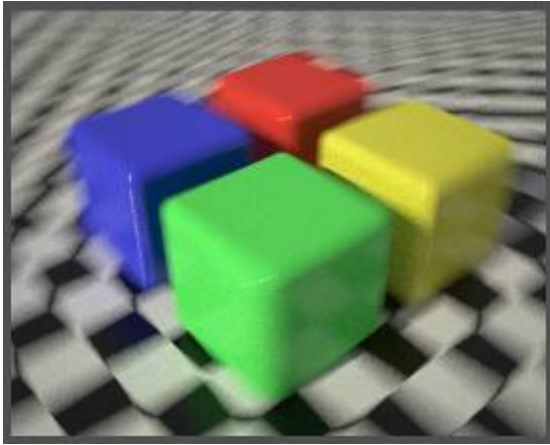
When this is enabled, the plugin will write information for two camera positions into the .mxs file at export time. This implies that there *are* two positions. However, Rhino has no concept of this, so the plugin must provide some way for you to communicate your intention. It does so using the command `Maxwell_SetCameraInitialPositionForMotionBlur`. This command may be run by right-clicking the last button in the plugin's Animation Control toolbar:



As the name of the command indicates, it records the current camera position for use later – when the scene is rendered, it is the difference between this recorded position and the position of the camera at export-time which will be used by Maxwell to figure camera blur. For example, setting the camera's initial position with the left view, then rendering with the camera moved to that show on the right...



Results in this image:



As the name of the command indicates, it records the current camera position for use later – when the scene is rendered, it is the difference between this recorded position and the position of the camera at export-time which will be used by Maxwell to figure camera blur.

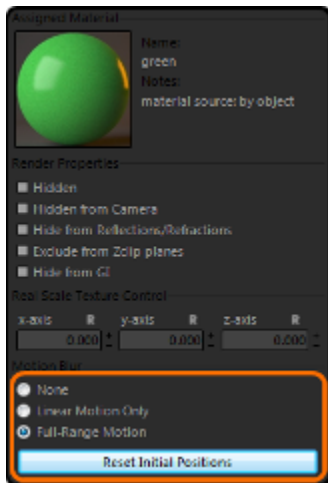
When the plugin is in an animation loop (see the Animation tutorial), and it has motion blur enabled, this frame-to-frame position recording is done automatically.

Object Motion Blur

Similar to cameras, objects also need to have a 'previous' position recorded so that Maxwell can calculate blur based on the difference between that position and the current position at export-time. This may be done by using the *Maxwell_SetObjectsInitialPositionForMotionBlur* command, which may be run by left-clicking the last button in the plugin's Animation Control toolbar:



Additionally, the motion blur state of objects may be managed using the Maxwell Object Properties page, in the Rhino Object Properties window:



As shown in the window, there are two types of blur supported by the plugin:

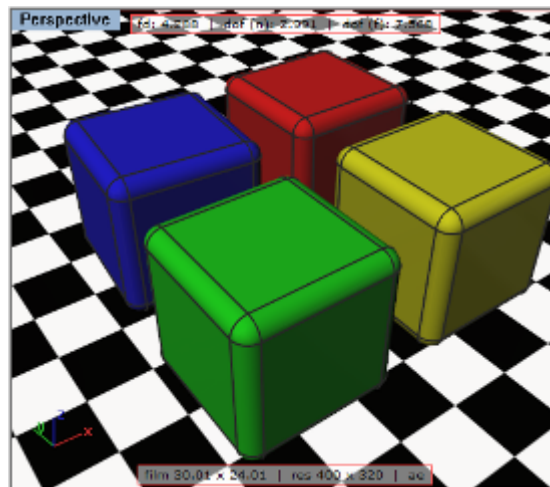
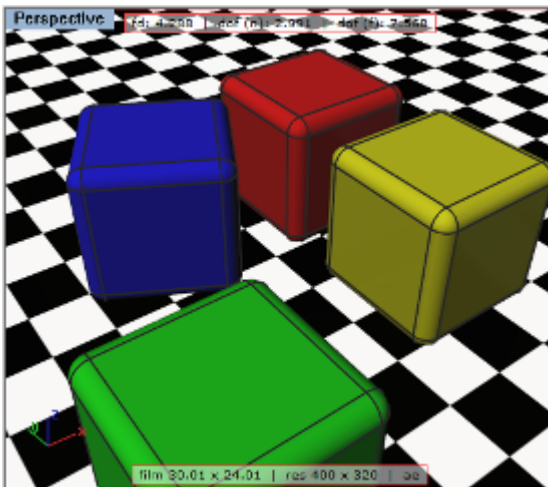
1. Linear Motion Only: objects which use this must only be moved in a linear fashion
2. Full-Range Motion: objects which use this may be rotated or moved in arbitrary ways

The reason for having two types is based in how Maxwell needs to get mesh-data for motion blur-enabled objects. Technically, a full copy of each mesh must be contained in the .mxs file, in addition to the regular mesh – blur is figured by the difference between all points in the two meshes. As you may figure, this also means that there must be double the memory available, since there is essentially a copy of each mesh. When you use Linear Motion Only, the plugin does not store an entire extra mesh; instead it just stores one point and one vector. These are used at export time to write the two meshes into the .mxs, therefore reducing the amount of memory required by the plugin. If you cannot restrict an object to moving in a linear path, then you will need to use Full-Range Motion blur, and the plugin will indeed store an entire copy of the mesh for motion blur purposes.

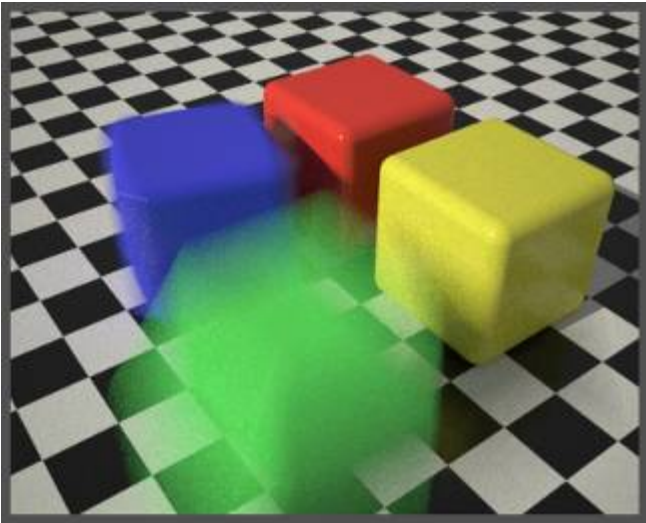
In addition to the toolbar and Object Properties methods of managing object motion blur, the plugin offers several other commands for working with objects and motion blur (Full-Range is referred to as 'Rotational' here):

- **Maxwell_DisableLinearBlurForAllObjects**
- **Maxwell_DisableRotationalBlurForAllObjects**
- **Maxwell_DisableSelectedObjectsLinearBlur**
- **Maxwell_DisableSelectedObjectsRotationalBlur**
- **Maxwell_EnableSelectedObjectsLinearBlur**
- **Maxwell_EnableSelectedObjectsRotationalBlur**
- **Maxwell_SelectRotationalBlurEnabledObjects**
- **Maxwell_SelectLinearBlurEnabledObjects**

For an example, we will try using both types of object motion blur. The green box will be set to use linear blur, and the blue one full-range. Here are the two positions which will be rendered; the initial positions, and the positions at export-time:



This results in the following image:



As you can see, motion blur is not only for use in animation – it is quite easy to use it manually for still shots as well. Even so, as with camera motion blur, initial position will be automatically inferred from the previous frame when rendering animations.

Note that motion blur will be calculated from frame to frame automatically when using Rhino's AnimationTools or Bongo animation plugins; all that is necessary is that the desired objects (and camera, if desired) have the proper motion blur enabled.

Maxwell for Rhino :: HDRI Tutorial

The Maxwell for Rhino plugin offers two ways to work with HDRIs (**H**igh **D**ynamic **R**ange **I**mages):

1. as environment maps used to light the scene
2. as Maxwell emitter materials, to be applied to geometry

HDRIs as scene Lighting

Maxwell is able to use HDRIs (.mxi, .hdr, and .exr) as the source of global illumination; this allows for very realistic environmental backgrounds and optionally, reflection, refraction, and illumination. In fact, you can see in the Scene Manager's Environment Settings page, all of the settings which allow you to specify 'Image Based' lighting for separate Background, Illumination, Reflection, and Refraction channels.

The HDRIs placed in these channels will be spherically mapped in such a way as to completely enclose the scene, and the plugin is able to show you the effects of this in a Rhino (perspective projection) viewport. Changes to the channel's Scale and Offset parameters are also shown in the viewport, which makes it easy to set up your scene.

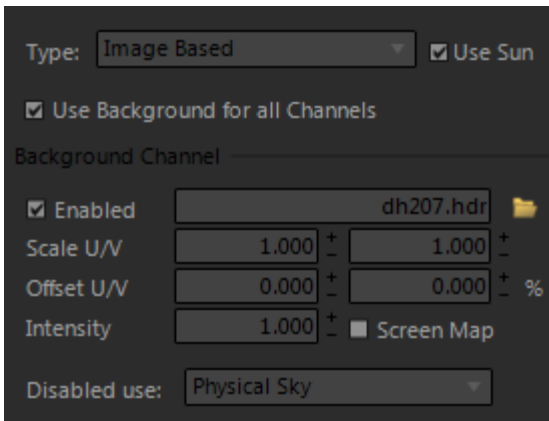
The fastest way to set up Image Based lighting in your scene is by right-clicking on one of the buttons in the plugin's Environments toolbar (if you don't see this, go to Main Menu > Maxwell > Plugin Windows > Show All Toolbars):



When you right-click, for example, the Background channel's button, you will be presented with a file dialog which has a preview window for the various types of HDRI. Choose a file from your file system and click OK – now this image will be shown in the Rhino viewport.

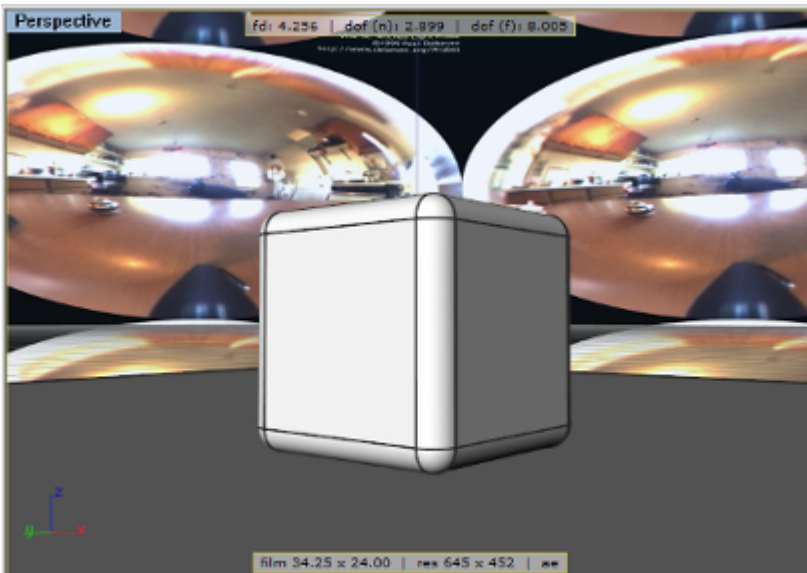


To adjust how this image is mapped, open the Scene Manager and go to Environment Settings. The plugin will have already switched the scene's environment Type to 'Image Based' as soon as you chose a valid map, so you can scroll to the Image Based section of the window and expand the Background section:

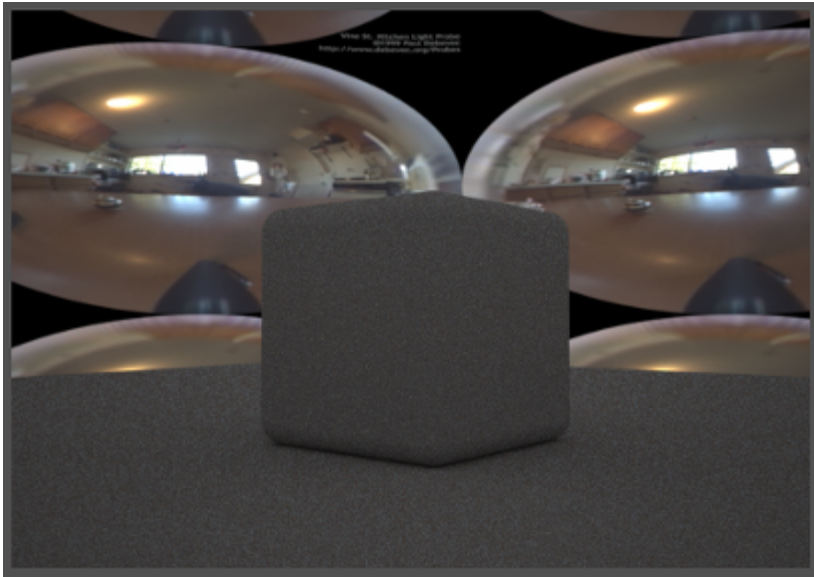


As you can see, the map chosen for the Background channel has also been assigned to the other channels – this is the case when the Link Channels option is enabled. This is just a workflow shortcut; though not always, it is common to use the same map in all channels, and using Link Channels allows you to assign and adjust the parameters for all channels at once.

If you now adjust the Scale U/V or Offset U/V parameters, the changes you make will be shown interactively in the viewport. As a convenience (also the case with all Scene Manager parameters), rather than typing numbers, adjustment of numerical parameters may also be accomplished by pressing down the left mouse button over the desired parameter value and rolling the mouse up and down. Adjusting the Scale U and V parameters to 10.0/10.0 yields this result in the viewport:



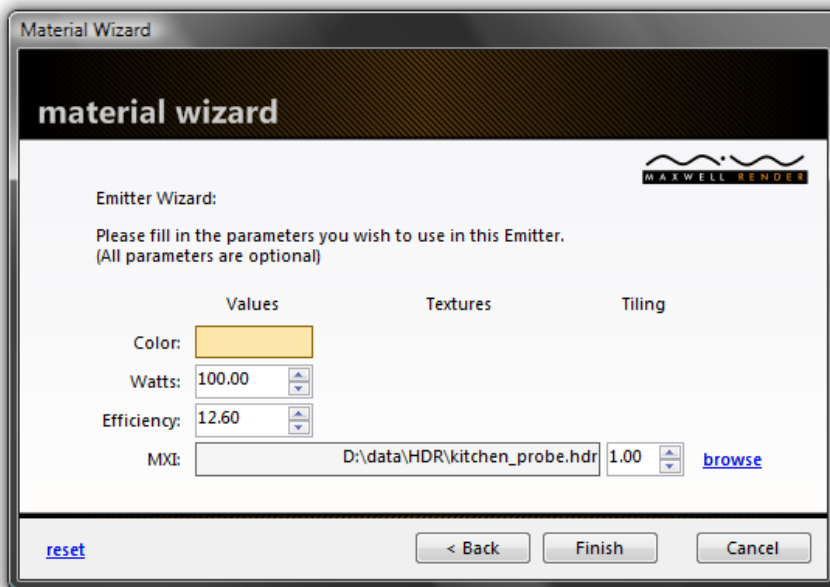
As you can see, the viewport accurately depicts what will appear in the actual render:



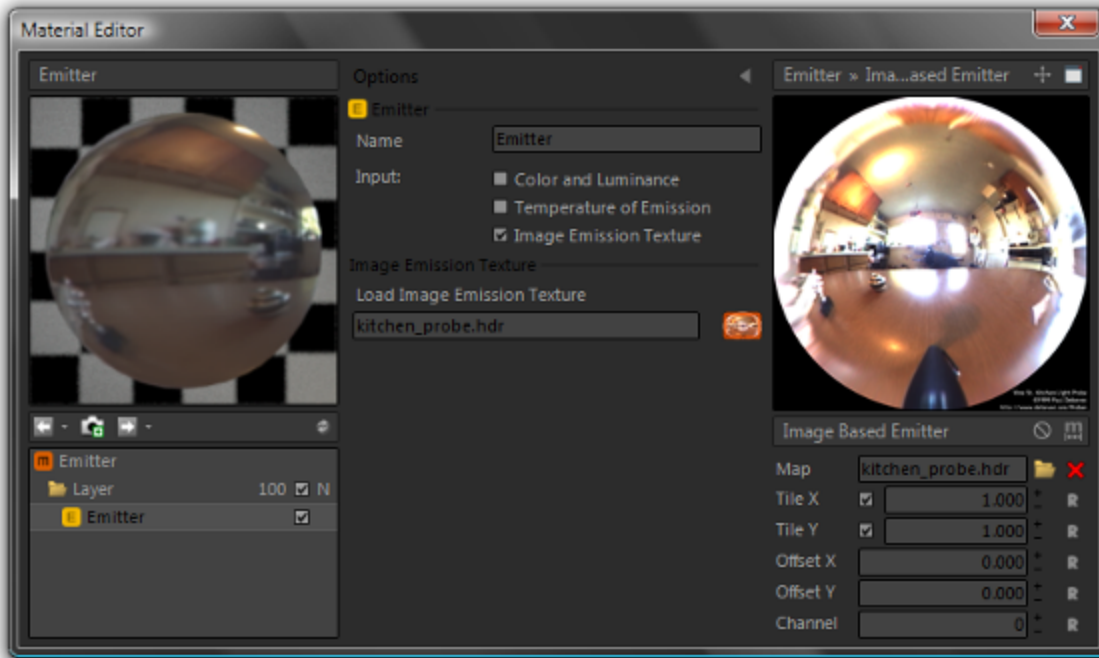
To remove the HDRI preview from the viewport, click the 'none' button in the Environments toolbar.

HDRIs as Maxwell materials

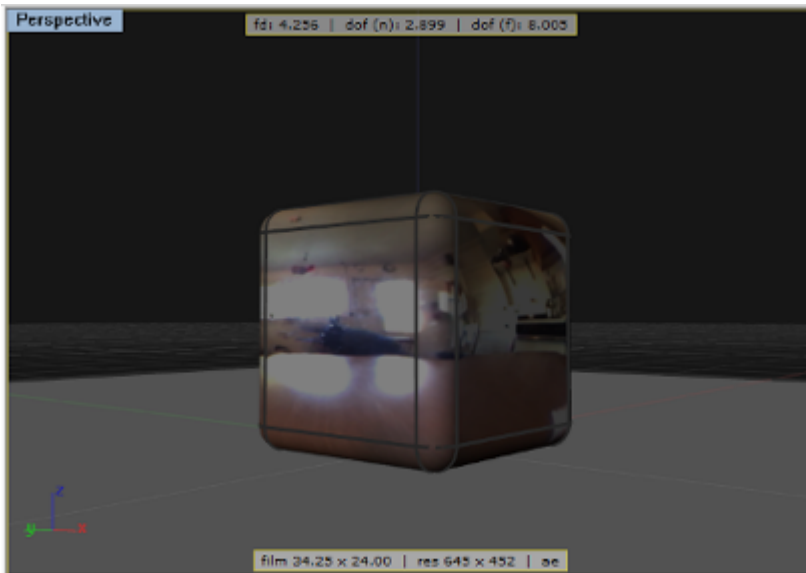
In addition to using HDRIs as scene lighting, they may also be used in Maxwell materials. This may be useful for placing an exterior image into a window, or rendering an emissive computer screen. To quickly create an Image Based material, start the plugin's Material Wizard and choose the *Emitter* option. In the next wizard page, browse to an .mxi, .hdr, or .exr file:



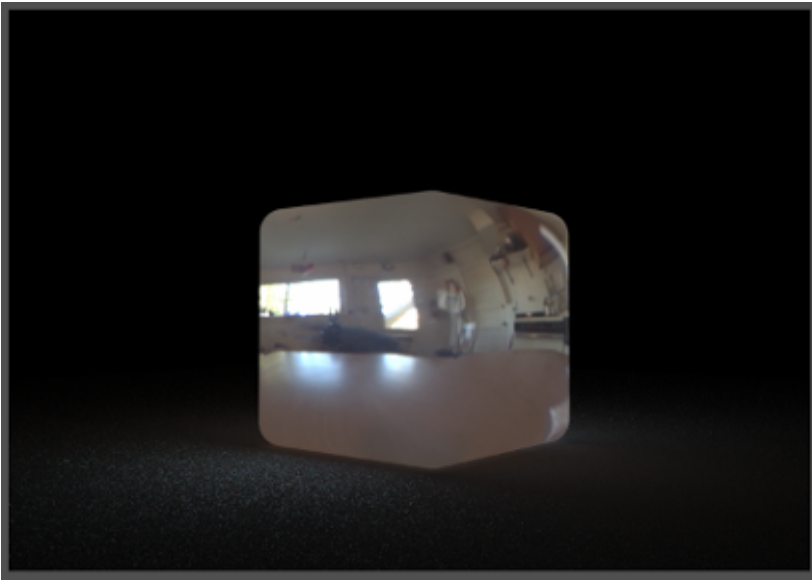
When the wizard is completed, there will be a new emitter material shown in the plugin's Material Editor:



Assigning this material to an object in Rhino will result in the HDRI being shown mapped on the object:



Rendering with no environment shows how this material looks:



materials created this way may be mapped using all of the same methods as any other textures, i.e. by using the plugin's Real Scale function (see the Real Scale tutorial), or by using Rhino's texture mapping tools (see *Rhino Help > Object Properties > Texture Mapping*).

Maxwell Script for .NET Plugins

Scripting the Maxwell Plugin

The plugin includes a library, *Maxwell.Script.dll*, which allows access to the current scene from VBA, VBScript, .Net, and any other environment which is able to create and query .Net or COM objects.

The library is installed to the .Net Framework Global Assembly Cache when the plugin is installed. A separate copy is also installed in the plugin's directory located @ *[ProgramFiles]\MaxwellDotNET\Maxwell.Script*. The library installer installs five files at this location:

- **Maxwell.Script.dll:** this is the .Net class-library which contains the *IScriptObject* .Net Interface, which may be Linked directly in .Net code
- **Maxwell.Script.tlb:** this type library is registered by the installer, to enable the library to be shown in the VBA 'Links' and Visual Studio 'Links/COM' dialogs. It may be used for COM interop, and provides a definition for the *IScriptObject* COM interface
- **Maxwell.Script.xml:** provides IntelliSense in the Visual Studio IDE when *Maxwell.Script.dll* is referenced in a .Net project
- **register.bat:** calls *regasm.exe* to register the *Maxwell.Script.dll* library for use in VBA, etc.
- **unregister.bat:** calls *regasm.exe* to unregister the *Maxwell.Script.dll* library

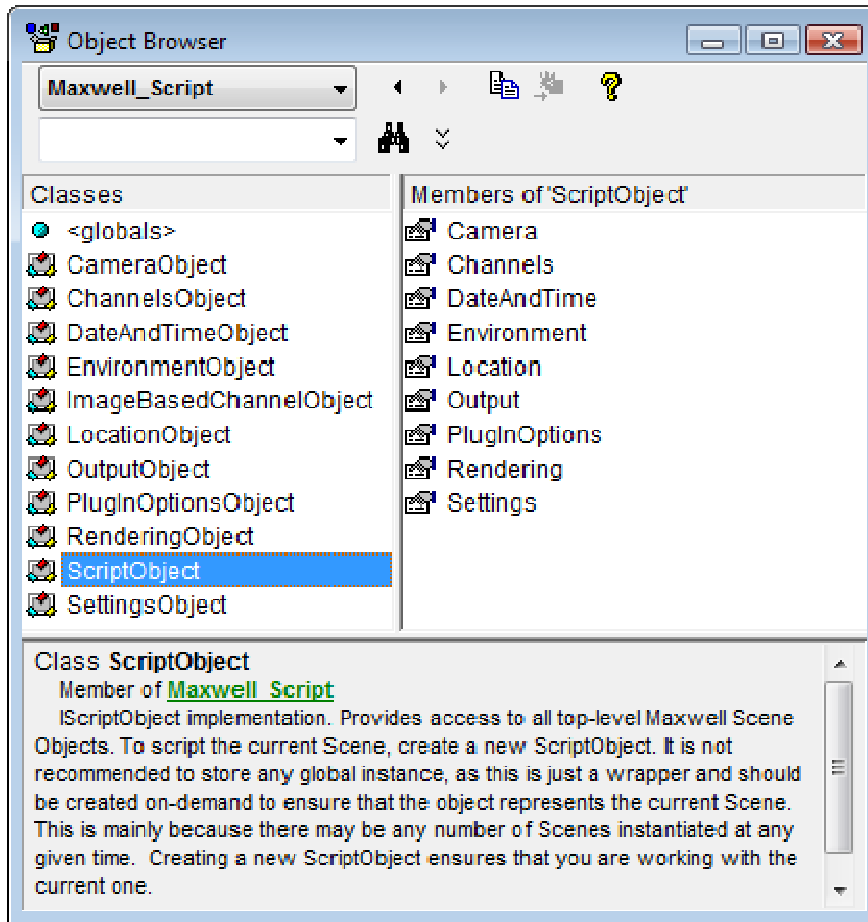
Methodology

The main scripting interface, *IScriptObject*, is implemented internally by *Maxwell.Script.dll* in the object *ScriptObject*. *ScriptObject*'s constructor is the only public entry-point to the library, as all other objects have no public constructor. *ScriptObject*, is a very light-weight object - in fact, it has absolutely no data members, and exists purely as a static event broadcaster. For this reason, as well as the fact that there may be any number of scenes in existence, it is not appropriate to keep any global instance - clients should always create a new *ScriptObject* when one is needed to ensure they are working with the current scene.

If *ScriptObject* is just a broadcaster, then there must also be a listener to do the actual work - this is implemented inside the plugin. If no listener currently exists, then events will not be handled, and values will simply be set to default for their parameter type. If however there is a listener, it will handle the raised events, getting or setting requested values as necessary to establish a communication between clients and the current Maxwell scene - which is always hosted by the event-listener (i.e. the plugin).

Object Model

In addition to `IScriptObject`, `Maxwell.Script.dll` defines several other interfaces. `IScriptObject` has as its properties one of each of these child objects:



The supplementary interfaces are implemented as child-properties of `IScriptObject`. Scene parameters are always accessed through `IScriptObject`'s child sub-objects:

- **ICameraObject** : properties corresponding to those in Manager > Camera
- **IChannelsObject** : properties corresponding to those in Scene Manager > Channels
- **IDateAndTimeObject** : properties corresponding to those in Scene Manager > Date & Time
- **IEnvironmentObject** : properties corresponding to those in Scene Manager > Environment
- **IImageBasedChannelObject** : properties corresponding to those in Scene Manager > Environment > Image Based Channels
- **ILocationObject** : properties corresponding to those in Scene Manager > Location
- **IOutputObject** : properties corresponding to those in Scene Manager > Output
- **IPlugInOptionsObject** : properties corresponding to those in Scene Manager > Options
- **IRenderingObject** : methods to invoke the plugin's Rendering functions
- **ISettingsObject** : properties corresponding to those in Scene Manager > Settings

All parameters are represented as simple get/set properties, with no parameterized methods. The only data-types required are string, integer, double, and boolean, with no need to pass arrays. Parameters which are represented as enums in the underlying type accept string input, e.g. setting the scene's Environment type to Physical Sky takes the string: "PhysicalSky". In general, property names follow their corresponding names as shown in the Scene Manager, with some small differences from time to time. On the overall though, if one is able to use the Scene Manager user-interface, it should be quite easy to script the plugin.

Sample Code

Here is a simple example showing how to render an adjustable Sun-cycle study using the plugin's animation capability through RhinoScript:

Option Explicit

```
Sub Maxwell_SunStudy()

    Dim Maxwell
    ' create the ScriptObject

    Set Maxwell = CreateObject("Maxwell.Script.ScriptObject")

    ' successfully created?
    If Maxwell Is Nothing Then
        Rhino.Print("Unable to create Maxwell.Script.ScriptObject.")
        Exit Sub
    End If

    ' successfully connected?
    If Not Maxwell.IsConnected Then
        Rhino.Print("Unable to connect to the current Maxwell scene.")
        Exit Sub
    End If

    ' enable sun and set physical sky active
    Maxwell.Environment.SunEnabled = True
    Maxwell.Environment.EnvironmentType = "PhysicalSky"

    ' no objects will move, so cache meshes
    Maxwell.PluginOptions.CacheMXSMeshes = True

    ' loop vars
    Dim endTime, frameLength

    ' set start time
    Maxwell.DateAndTime.TimeOfDay = 7

    ' set end time
    endTime = 19.5

    ' set frame interval
    frameLength = 0.1

    ' activate the plugin's animation loop
    Maxwell.Rendering.BeginAnimation()

    ' loop
    While Maxwell.DateAndTime.TimeOfDay < endTime

        ' increment
        Maxwell.DateAndTime.TimeOfDay = Maxwell.DateAndTime.TimeOfDay + frameLength

        ' write frame to disk
        Maxwell.Rendering.RenderToMxs()

    Wend

    ' done
    Maxwell.Rendering.EndAnimation()

End Sub

Maxwell_SunStudy
```

Interface Definitions:

The following is a complete listing of the actual interface definitions (in VB.Net) for the library.

Interfaces:

```
Public Interface IScriptObject
Public Interface IRenderingObject
Public Interface ICameraObject
Public Interface IDateAndTimeObject
Public Interface ILocationObject
Public Interface IEnvironmentObject
Public Interface IImageBasedChannelObject
Public Interface IOutputObject
Public Interface ISettingsObject
Public Interface IChannelsObject
Public Interface IPlugInOptionsObject
```

Object Model:

```
Public Interface IScriptObject

    ReadOnly Property IsConnected() As Boolean
    ReadOnly Property Rendering() As IRenderingObject
    ReadOnly Property Camera() As ICameraObject
    ReadOnly Property DateAndTime() As IDateAndTimeObject
    ReadOnly Property Location() As ILocationObject
    ReadOnly Property Environment() As IEnvironmentObject
    ReadOnly Property Output() As IOutputObject
    ReadOnly Property Settings() As ISettingsObject
    ReadOnly Property Channels() As IChannelsObject
    ReadOnly Property PlugInOptions() As IPlugInOptionsObject

Public Interface IRenderingObject

    Sub RenderToMxs ()
    Sub RenderToMxcl ()
    Sub RenderToMxst ()
    Sub RenderRegion ()
    Sub RenderBlowup ()
    Sub BeginAnimation ()
    Sub EndAnimation ()
    Sub CancelAnimation ()

End Interface
```

Public Interface ICameraObject

```
Property Name() As String
Property Notes() As String
Property ShutterSpeed() As Double
Property fStop() As Double
Property ResolutionMode() As String
Property LockRatio() As Boolean
Property XRes() As Integer
Property YRes() As Integer
Property PixelAspect() As Double
Property FilmWidth() As Double
Property FilmHeight() As Double
Property ISO() As Double
Property DiaphragmType() As String
Property Blades() As Integer
Property Angle() As Integer
Property ZClipEnabled() As Boolean
Property ZClipFar() As Double
Property ZClipNear() As Double
Property ShiftLensX() As Double
Property ShiftLensY() As Double
```

End Interface

Public Interface IDateAndTimeObject

```
Property Name() As String
Property Notes() As String
Property SimpleTime() As String
Property SimpleDate() As String
Property Hour() As Integer
Property Minute() As Integer
Property AmOrPm() As String
Property TimeOfDay() As Double
Property DST() As Boolean
Property DayOfMonth() As Integer
Property Month() As String
Property Year() As Integer
Property DayOfYear() As Integer
Property JulianDay() As Double
```

End Interface

Public Interface ILocationObject

```
Property Name() As String
Property Notes() As String
Property Longitude() As Double
Property Latitude() As Double
ReadOnly Property GMTOffset() As Double
```

End Interface

Public Interface IEnvironmentObject

```
Property Name() As String
Property Notes() As String
Property EnvironmentType() As String
Property SunEnabled() As Boolean
Property sceneRotation() As Double
Property PhysicalSky_Turbidity() As Double
Property PhysicalSky_Water() As Double
Property PhysicalSky_Ozone() As Double
Property SkyDome_Color_R() As Integer
Property SkyDome_Color_G() As Integer
Property SkyDome_Color_B() As Integer
Property SkyDome_Intensity() As Double
Property ImageBased_LinkChannels() As Boolean
Property ImageBased_DisabledChannelEnvironmentType() As String
ReadOnly Property BackgroundChannel() As IImageBasedChannelObject
ReadOnly Property IlluminationChannel() As IImageBasedChannelObject
ReadOnly Property ReflectionChannel() As IImageBasedChannelObject
ReadOnly Property RefractionChannel() As IImageBasedChannelObject
```

Public Interface IImageBasedChannelObject

```
ReadOnly Property ChannelType() As String
Property Path() As String
Property Enabled() As Boolean
Property Intensity() As Double
Property ScaleU() As Double
Property ScaleV() As Double
Property OffsetU() As Double
Property OffsetV() As Double
Property ScreenMapped() As Boolean
```

End Interface

End Interface

Public Interface IOutputObject

```
Property Name() As String
Property Notes() As String
Property scenePath() As String
Property ImagePath() As String
Property ImageType() As String
Property CommandLine() As String
Property RenderEngine() As String
Property RenderSilent() As Boolean
Property LowPriority() As String
Property DisableBitmaps() As Boolean
Property EnableProtection() As Boolean
```

End Interface

Public Interface ISettingsObject

```
Property Name() As String
Property Notes() As String
Property SamplingLevel() As Integer
Property Time() As Integer
Property CPUThreads() As Integer
Property MultiLightEnabled() As Boolean
Property Burn() As Double
Property Gamma() As Double
Property VignettingEnabled() As Boolean
Property Vignetting() As Double
Property WriteMXI() As Boolean
Property MXIFileName() As String
Property DoDirectLayer() As Boolean
Property DoDirectReflectionLayer() As Boolean
Property DoDirectRefractionLayer() As Boolean
Property DoIndirectLayer() As Boolean
Property DoIndirectReflectionLayer() As Boolean
Property DoIndirectRefractionLayer() As Boolean
```

End Interface

Public Interface IChannelsObject

```
Property Name() As String
Property Notes() As String
Property DoAlpha() As Boolean
Property DomaterialID() As Boolean
Property DoObjectID() As Boolean
Property DoOpaque() As Boolean
Property DoRender() As Boolean
Property DoShadow() As Boolean
Property DoZBuffer() As Boolean
Property ZBufferNear() As Double
Property ZBufferFar() As Double
```

End Interface

Public Interface IPlugInOptionsObject

```
Property ShowInfoTips() As Boolean
Property UseCustomCursors() As Boolean
Property DecimalPrecision() As Integer
Property NoPreviewImagePath() As String
Property VerboseMessages() As Boolean
Property SuppressWarnings() As Boolean
Property AlwaysWriteMXI() As Boolean
Property DisablePathCheck() As Boolean
Property CacheMXSMeshes() As Boolean
Property ExportSpotLights() As Boolean
Property ExportPointLights() As Boolean
Property IgnoreLinkedMXM() As Boolean
Property materialDefaults_Previewscene() As String
Property materialDefaults_PreviewEngine() As String
Property materialDefaults_PreviewSL() As Integer
Property materialDefaults_PreviewSaveQuality() As Integer
Property DefaultPaths_scenesPath() As String
Property DefaultPaths_RendersPath() As String
Property DefaultPaths_MXIPath() As String
Property SearchPaths_CheckPaths() As Boolean
Property SearchPaths_SearchDepth() As Integer
Property SearchPaths_CheckFileVersion() As Boolean
Property SearchPaths_IORFiles() As String
Property SearchPaths_R2Files() As String
Property SearchPaths_Textures1() As String
Property SearchPaths_Textures2() As String
Property SearchPaths_Textures3() As String
Property SearchPaths_Textures4() As String
Property SearchPaths_Textures5() As String
```

End Interface

End Interface

Maxwell for Rhino :: RhinoScript

The Maxwell for Rhino 4 plugin offers many commands, available for use in the Rhino command-line, in toolbar-button macros, or via RhinoScript.

RhinoScript Access:

Maxwell_CameraSettings : Provides RhinoScript access to the parameters found in Scene Manager > Camera Settings. Changes will be made to the currently-selected Camera. To select a Camera programmatically, you should use RhinoScript to select the associated Rhino view.

Maxwell_DateAndTimeSettings : Provides RhinoScript access to the parameters found in Scene Manager > Date and Time Settings.

Maxwell_LocationSettings : Provides RhinoScript access to the parameters found in Scene Manager > Location Settings.

Maxwell_EnvironmentSettings : Provides RhinoScript access to the parameters found in Scene Manager > Environment Settings.

Maxwell_OutputSettings : Provides RhinoScript access to the parameters found in Scene Manager > Output Settings.

Maxwell_RenderSettings : Provides RhinoScript access to the parameters found in Scene Manager > Render Settings.

Maxwell_RenderChannels : Provides RhinoScript access to the parameters found in Scene Manager > Channels Settings.

Maxwell_ImageBasedEnvironments : Enables the chosen Image Based Environment channel, if it has a valid file assigned.

Maxwell_PluginOptions : Provides RhinoScript access to many of the parameters found in Scene Manager > Plugin Options.

Plugin Windows and Toolbars:

Maxwell_Show : Shows all plugin windows.

Maxwell_ShowToolbar : Shows all of the Maxwell toolbars.

Maxwell_ToolbarReset : Deletes the plugin's .tb files, re-creates them, and re-loads the fresh .tb into Rhino. This is useful if your Maxwell toolbars become corrupted, or you wish to return them to the default state.

Maxwell_SceneManager : Prompts for a page-name, and opens the Scene Manager to the specified page.

Maxwell_ToggleCameraBackPanel : Toggles the visibility of the Camera LCD panel.

Maxwell_ToggleCameraHeadsUpDisplay : Toggles the plugin's heads-up-display feature. The heads-up-display is used to visualize the Maxwell film, in regards to the Rhino viewport.

Maxwell_ShowSceneManager : Shows the Scene Manager, whether it is set to be used as a standalone window, or as a Rhino dock bar.

Maxwell_HidesceneManager : Hides the Scene Manager window.

Maxwell_ToggleSceneManager : Toggles the visibility of the Scene Manager.

Maxwell_ShowDBManager : Shows the Database Manager window, whether it is set to be used as a standalone window, or as a Rhino dock bar.

Maxwell_HideDBManager : Hides the Database Manager window.

Maxwell_ToggleDBManager : Toggles the visibility of the Database Manager.

Maxwell_ShowLogViewer : Shows the Log Viewer window.

Maxwell_HideLogViewer : Hides the Log Viewer window.

Maxwell_ToggleLogViewer : Toggles the visibility of the Log Viewer.

Maxwell_ShowMaterialEditor : Shows the Material Editor window.

Maxwell_HideMaterialEditor : Hides the Material Editor window.

Maxwell_ToggleMaterialEditor : Toggles the visibility of the Material Editor.

Maxwell_ShowMxmBrowser : Shows the MXM Browser, whether it is set to be used as a standalone window, or as a Rhino dock bar.

Maxwell_HideMxmBrowser : Hides the MXM Browser window.

Maxwell_ToggleMxmBrowser : Toggles the visibility of the MXM Browser.

Plugin Help:

Maxwell_ShowHelp : Shows the plugin's local help files if they are available, otherwise this shows the online version.

Maxwell_ShowLocalHelp : Shows the local version of the plugin's help files, if you have downloaded them. This command uses the file specified in Scene Manager > Plugin Options > Plugin Help URL, and it should point to the index.htm file in the unzipped help download. The help files .zip is available from the same location as the plugin's installer.

Maxwell_ShowRemoteHelp : Shows the online version of the plugin's help, hosted on NL servers. This help may be updated mid-version, and so may be more up-to-date than the help files you have on your machine, if you installed them.

Maxwell_ShowHelpPage : Shows the specified help page.

Rendering:

Maxwell_Render : Exports the scene to an MXS file located at the path specified in Scene Manager > Output Settings > scene Path, and starts rendering it in maxwell.exe. Optionally, you may choose whether to export all, or just a selected set of objects. Additionally, an option, 'UseViewNameAsOutputName', is offered, the function of which is to use the current view's name as the scene output name, rather than whatever is specified in Scene Manager > Output. This can be useful for rendering a series of views without having to manually set the output name to match each one.

Maxwell_RenderBlowUp : Prompts for a rectangular selection. Once selected, exports the scene to an MXS file located at the path specified in Scene Manager > Output Settings > scene Path, and starts rendering it in maxwell.exe using the 'Blow-up' rendering mode. Optionally, you may choose whether to export all, or just a selected set of objects.

Maxwell_RenderRegion : Prompts for a rectangular selection. Once selected, exports the scene to an MXS file located at the path specified in Scene Manager > Output Settings > scene Path, and starts rendering it in maxwell.exe using the 'Region' rendering mode. Optionally, you may choose whether to export all, or just a selected set of objects.

Maxwell_RenderToMxs : Exports the scene to an MXS file located at the path specified in Scene Manager > Output Settings > scene Path. Optionally, you may choose whether to export all, or just a selected set of objects.

Maxwell_RenderToMxst : Exports the scene to an MXS file located at the path specified in Scene Manager > Output Settings > scene Path, and opens it in Maxwell Studio (MXST). Optionally, you may choose whether to export all, or just a selected set of objects.

Objects and materials:

Maxwell_SetCurrentmaterial : Prompts for a material name, and sets the specified material as current, if it exists in the scene. Use double-quotes to specify any name which has spaces.

Maxwell_ApplyCurrentmaterialToObjectsByColor : Applies the currently-selected material to objects which use the specified RGB color.

Maxwell_ApplyCurrentmaterialToSelectedObjects : Applies the currently-selected material to all selected objects.

Maxwell_SelectObjectsAndApplyCurrentmaterial : Prompts for an object selection, and applies the current material to the selected objects.

Maxwell_SelectObjectsWithCurrentmaterial : Selects any objects which use the current material.

Maxwell_SelectNextmaterial : Selects the 'Next' material. This will only have an effect if there is a 'Next' material (i.e. the 'Back' button in the Database Manager has been clicked, or the Maxwell_SelectPreviousmaterial command has been executed).

Maxwell_SelectPreviousmaterial : Selects the material which was current previously to the current one.

Maxwell_ImportClipboardMxm : Opens the Windows clipboard to see if there is the path to an .mxm file contained there. If so, the .mxm is imported.

Maxwell_CheckFixAndReportBadGeometry : Traverses the model checking render-meshes for validity. A report will be printed indicating any deficient meshes found. Meshes with missing texture-coordinates or face normals will be fixed during the check, and this will also be included in the report. Individual objects reported in the Log Viewer may be selected in the viewport by selecting their ID in the Log Viewer, right-clicking, and choosing 'Select Object'

Maxwell_RestoreViewport : Traverses the scene, refreshing the viewport appearance of objects, based on the current state of the Maxwell material which is assigned to them.

Maxwell_RunmaterialWizard : Starts the material Wizard.

Maxwell_PathStatusReport : Checks all materials used in the scene for invalid paths. Prints a report of the results.

Maxwell_GatherFilesAndMakePathsRelative : Copies all Linked files to a 'textures' folder next to the current .3dm location, Makes copies of all materials used in the scene, and re-assigns objects to use the newly-copied materials. The copies are made because the paths are made relative to the new 'textures' folder during this process, and this would break other .3dms which also Link the original materials. After this has been done, it is possible to move the .3dm, along with the 'textures' folder, to another machine without needing to manually change any paths.

Maxwell_Extractmaterials : Traverses the model and generates Maxwell materials for all native (i.e. 'Basic')

Plugin Data:

Maxwell_ClearCachedData : Clears all data which has been cached by the plugin. This includes texture-previews and all currently-loaded materials. This command will erase the history for any materials which have stored previous versions.

Maxwell_GetCommandList : This command shows a window which lists all of the commands which the plugin provides.

Maxwell_GetVersionInfo : Gets the version information of the currently-running plugin.

Maxwell_RemoveAllMaxwellData : Removes all Maxwell data from the scene, including materials, Hide-to-X flags, etc.

Maxwell_SaveAsWithoutMaxwellData : Saves the document to a new filename, without including any Maxwell data.

Maxwell_SaveWithoutMaxwellData : Saves the current document, without including any Maxwell data.

Animation:

Maxwell_BeginAnimation : Starts the plugin's animation loop. While the loop is active, MXS output will auto-named using incremental numbers, and maxwell.exe will not be started. Any frames rendered between the execution of this command and the Maxwell_EndAnimation command will be included in the animation, which will be started rendering in maxwell.exe upon the execution of the Maxwell_EndAnimation command. Maxwell_CancelAnimation may be used at any time to quit the loop without rendering the animation.

Maxwell_EndAnimation : Ends the plugin's current animation loop, if one is active. Any MXS files which have been written since execution of the Maxwell_BeginAnimation command will then be rendered in maxwell.exe.

Maxwell_CancelAnimation : Stops the plugin's animation loop, if it is active.

Motion Blur:

Maxwell_DisableLinearBlurForAllObjects : Disables Motion Blur for any objects which are currently set to use Linear Motion Blur.

Maxwell_DisableRotationalBlurForAllObjects : Disables Motion Blur for all objects which are currently set to use rotational (i.e. 'Full-Range') Motion Blur.

Maxwell_DisableSelectedObjectsLinearBlur : Disables Motion Blur for all selected objects which are currently set to use Linear Motion Blur.

Maxwell_DisableSelectedObjectsRotationalBlur : Disables Motion Blur for all selected objects which are currently set to use rotational (i.e. 'Full-Range') Motion Blur.

Maxwell_EnableSelectedObjectsLinearBlur : Enables Motion Blur for all selected objects which are currently set to use Linear Motion Blur.

Maxwell_EnableSelectedObjectsRotationalBlur : Enables Motion Blur for all selected objects which are currently set to use rotational (i.e. 'Full-Range') Motion Blur.

Maxwell_SelectRotationalBlurEnabledObjects : Selects all objects which have rotational (i.e. Full-range) Motion Blur enabled.

Maxwell_SetCameraInitialPositionForMotionBlur : Stores the current Camera's current position for use in Camera Motion Blur. When this Camera is subsequently moved, then rendered, Maxwell will figure blur based on the difference between this position, and the position the Camera is in when it is rendered.

Maxwell_SetObjectsInitialPositionForMotionBlur : Sets the 'initial' position for any objects which have either Linear or rotational (i.e. Full-range) Motion Blur enabled. If these objects are then moved, and the scene rendered, Maxwell will calculate Motion Blur based on the difference between their locations when initial position was set, and when the scene was rendered.

Maxwell_SelectLinearBlurEnabledObjects : Selects any objects which have Linear Motion Blur enabled.

Interactive Camera Tools:

Maxwell_AutoFocusObjects : Prompts for an object-selection. The selection is then used to set the selected Camera's near and far focus limits by adjusting its' fStop. Auto-exposure should be enabled before using this command to avoid a radically-altered exposure.

Maxwell_PickCameraFocalDistance : Prompts you to select a point in the viewport. The chosen point is used to set the current Camera's focal distance.

Maxwell_PickFarZClipPlane : Prompts you to select a point in the viewport. The chosen point is used to set the current Camera's far Z-clip plane distance.

Maxwell_PickFilmSizeRectangle : Sets the size of the film for the Maxwell Camera using a rectangle drawn on the associated viewport. Note that this function uses the Camera's Shift Lens feature - as a result, the chosen rectangle cannot be more than 100% of its' width or height off of the viewport's center.

Maxwell_PickNearZClipPlane : Prompts you to select a point in the viewport. The chosen point is used to set the current Camera's near Z-clip plane distance.

Hide Objects to scene flag:

Maxwell_HideSelectedObjectsToscene : Sets the Maxwell 'Hide' flag for any currently-selected objects.

Maxwell_SelectObjectsHiddenToscene : Selects all objects which have the Maxwell 'Hide' flag enabled.

Maxwell_UnhideAllObjectsToscene : Disables the Maxwell 'Hide' flag for all objects.

Maxwell_UnHideSelectedObjectsToscene : Disables the Maxwell 'Hide' flag for all selected objects.

Hide Objects to Camera flag:

Maxwell_HideSelectedObjectsToCamera : Sets the Maxwell 'Hide to Camera' flag for any currently-selected objects.

Maxwell_SelectObjectsHiddenToCamera : Selects all objects which have the Maxwell 'Hide to Camera' flag enabled.

Maxwell_UnhideAllObjectsToCamera : Disables the Maxwell 'Hide to Camera' flag for all objects.

Maxwell_UnHideSelectedObjectsToCamera : Disables the Maxwell 'Hide to Camera' flag for all selected objects.

Hide Objects to GI flag:

Maxwell_HideSelectedObjectsToGI : Sets the Maxwell 'Hide to GI' flag for any currently-selected objects.

Maxwell_SelectObjectsHiddenToGI : Selects all objects which have the Maxwell 'Hide to GI' flag enabled.

Maxwell_UnhideAllObjectsToGI : Disables the Maxwell 'Hide to GI' flag for all objects.

Maxwell_UnHideSelectedObjectsToGI : Disables the Maxwell 'Hide to GI' flag for all selected objects.

Hide Objects to ZClip flag:

Maxwell_HideSelectedObjectsToZClip : Sets the Maxwell 'Hide to ZClip' flag for any currently-selected objects.

Maxwell_SelectObjectsHiddenToZClip : Selects all objects which have the Maxwell 'Hide to ZClip' flag enabled.

Maxwell_UnhideAllObjectsToZClip : Disables the Maxwell 'Hide to ZClip' flag for all objects.

Maxwell_UnHideSelectedObjectsToZClip : Disables the Maxwell 'Hide to ZClip' flag for all selected objects.

Hide Objects to Reflections and Refractions flag:

Maxwell_HideSelectedObjectsToReflectionsAndRefractions : Sets the Maxwell 'Hide to Reflections and Refractions' flag for any currently-selected objects.

Maxwell_SelectObjectsHiddenToReflectionsAndRefractions : Selects all objects which have the Maxwell 'Hide to Reflections and Refractions' flag enabled.

Maxwell_UnhideAllObjectsToReflectionsAndRefractions : Disables the Maxwell 'Hide to Reflections and Refractions' flag for all objects.

Maxwell_UnHideSelectedObjectsToReflectionsAndRefractions : Disables the Maxwell 'Hide to Reflections and Refractions' flag for all selected objects.

Miscellaneous:

Maxwell_SaveFireImage : Saves the current Maxwell Fire image (if there is one) to disk using the specified Path. Optionally, accepts a Sleep parameter which can be used to allow Maxwell Fire to render for a given number of seconds before the image is saved.

Exposure values (ISO 100 speed) for various lighting conditions

Lighting Condition	EV ₁₀₀
Daylight	
Light sand or snow in full or slightly hazy sunlight (distinct shadows)	16
Typical scene in full or slightly hazy sunlight (distinct shadows)	15
Typical scene in hazy sunlight (soft shadows)	14
Typical scene, cloudy bright (no shadows)	13
Typical scene, heavy overcast	12
Areas in open shade, clear sunlight	12
Outdoor, Natural light	
Rainbows	
Clear sky background	15
Cloudy sky background	14
Sunsets and skylines	
Just before sunset	12–14
At sunset	12
Just after sunset	9–11
e Moon, altitude > 40°	
Full	15
Gibbous	14
Quarter	13
Crescent	12
Moonlight, Moon altitude > 40°	
Full	–3 to –2
Gibbous	–4
Quarter	–6
Aurora borealis and australis	
Bright	–4 to –3
Medium	–6 to –5
Outdoor, Artificial Light	
Neon and other bright signs	9–10
Night sports	9
Fires and burning buildings	9
Bright street scenes	8
Night street scenes and window displays	7–8
Night vehicle traffic	5
Fairs and amusement parks	7
Christmas tree lights	4–5
Floodlit buildings, monuments, and fountains	3–5
Distant views of lighted buildings	2
Indoor, Artificial Light	
Galleries	8–11
Sports events, stage shows, and the like	8–9
Circuses, floodlit	8
Ice shows, floodlit	9
Offices and work areas	7–8
Home interiors	5–7
Christmas tree lights	4–5

(table from wikipedia.org @ http://en.wikipedia.org/wiki/Exposure_value)