Getting Started

Rendering with RenderMan for Maya

Render Settings

Attributes

Maya Materials and Custom Shaders

The Tools

Navigate using the menus on the left.

Rendering with RenderMan for Maya

Pixar Animation Studios

Setting up RenderMan for Maya

Load RenderMan for Maya

RenderMan for Maya must be successfully loaded into Maya before it can be used. Follow the steps below to load RenderMan for Maya.

To load RenderMan for Maya:

1) First, start Maya

2) Open the Plug-in Manager:

```
Window> Settings/Preferences> Plug-in Manager
```

3) Find the RenderMan for Maya plug-in. Check "loaded." This will load RenderMan for Maya.

4) Close the Plug-in Manager

5) You can now switch to the RenderMan renderer:

```
Render> Render Using> RenderMan
```

Now you're ready to begin rendering with RenderMan

Set the Default Renderer

Pixar's RenderMan can be set as Maya's default renderer so that it is the active renderer when Maya is started. This makes using RenderMan for Maya a little more straightforward. In any case, you can always switch renderers during a Maya session.

To set RenderMan as the default renderer:

1) Open the Preferences Window:

```
Window> Settings/Preferences> Preferences
```

2) In the Rendering category, select "RenderMan" as the default renderer.

3) Save the new setting and close the window.

To switch between renderers:

1) Go to the Render menu and select:

```
Render> Rendering Using
```

2) Pick a renderer: RenderMan, Maya Software, etc.

How to Render

Rendering Methods

RenderMan supports several ways of rendering scene files, both interactive and batch. The best method will often depend upon on whether the rendered image is an iterative preview or frames for a final animation. RenderMan for Maya supports these methods (which are quite similar to Maya's own):

1. Interactive renders from Maya, using the render menu.

   Best for preview rendering
2. Batch rendering via the Maya UI.

   Best for rendering final images, especially animation, of an open Maya file.
3. Batch rendering from a shell or command line.

   Best for launching jobs on any number of Maya files.
4. Spooling renders via Alfred

   This is another method of distributing Maya batch renders, but it also offers additional RIB-spooling features for RenderMan for Maya Pro users.

How to Render

The following instructions outline the various workflows available for rendering with RenderMan for Maya.

Interactive Preview Rendering:

1. Pick an active window as your camera in your current Maya scene.
2. From Maya's Render menu select:

```
Render> Render Current Frame
```

3. The image renders into the Render View.

Batch Rendering from Maya

1. Ensure that the camera (and other settings) are set correctly in the Render Globals.
2. From Maya's Render menu select:

```
Render> Batch Render
```

3. The batch job will begin as a background process.

Batch Rendering from the Command Line

From a command line use the following format:

```
Render -r rman sceneFile
```

● **Note**

If no project is specified, the current project is used.

Spooling Jobs via Alfred

Spooling renders via Alfred require Alfred and the Alfred maitre d' to be installed. Additionally, Alfserver and RfM (or RfM Pro) need to be installed on the remote render nodes. Spooled RIB jobs require RenderMan Pro Server on the remote render nodes.

1. Open the Batch Render options window.
2. Check the *Alfred Spool* option box.
3. RenderMan for Maya users have two spooling options:
    - `mayabatch local` executes a batch render on the local client.
    - `mayabatch remote` distributes your batch renders to remote render clients.
4. RenderMan for Maya Pro users have three additional options:
    - `immediate rib, local render` generates RIB files and renders them on the local client.
    - `deferred rib, remote render` generates RIB locally (via a new Maya process) and distributes the RIB files to remote render clients for rendering.
    - `remote rib, remote render` executes the ribgen and the render on remote render clients.

Rendering Options

Multithreading

You can specify the number of threads to use via the *Render Current Frame* and *Batch Render* option boxes. The default setting, `0`, means use *all* available processors. Note that the licensing policy permits four threads per license.

Batch Render Flags

The following flags are supported for Batch Renders via the command line (i.e. `Render -r rman foo.ma`):

Common options:

-help

Print help

-test

Print Mel commands but do not execute them

-verb

Print Mel commands before they are executed

-keepMel

Keep the temporary Mel file

-listRenderers

List all available renderers

-renderer string

Use this specific renderer

-r string

Same as -renderer

-proj string

Use this Maya project to load the file

-log string

Save output into the given file

Specific options for -r rman (the RenderMan renderer):

All purpose flags:

-setAttr string string

This flag can be used to set any of the global attributes listed in RenderMan_for_Maya.ini. It takes a name value pair. Attribute values which have multiple data elements should be surrounded by quotes. The flag can be used multiple times. Example:

```
Render -r rman -setAttr ShadingRate 5 -setAttr PixelSamples "3 3" -setAttr
motionBlur 1 -setAttr Format:resolution "320 240" filename
```

-setPref string string

This flag can be used to set any of the preferences listed in RenderMan_for_Maya.ini. It takes a name value pair. Attribute values that have multiple data elements should be surrounded by quotes. The flag can be used multiple times. For example:

```
Render -r rman -setPref BatchCompileMode zealous filename
```

General purpose flags:

-rd path

Directory in which to store image files

-fnc string

File Name Convention:

```
name, name.ext, name.#.ext, name.ext.# name.#, name#.ext, name_#.ext
```

As a shortcut, numbers 1, 2, ⟩ can be used.

-im filename

Image file output name

-of string

File format of output images: Alias, Cineon, MayaIFF, OpenEXR, SGI8, SGI16, SoftImage, Targa, Tiff8, Tiff16, Tiff32

Frame numbering options:

`-s float`

Starting frame for a sequence

`-e float`

End frame for a sequence

`-b float`

By frame/step for a sequence

`-pad int`

Number of digits in the frame number included in the output image file name

`-rfs int`

The initial (renumbered) frame number for the first frame when rendering

`-rfb int`

The step by which frames are renumbered (used in conjunction with `-rfs`).

Camera options:

`-cam name`

The name of the camera from which you are rendering

`-rgb boolean`

Enable/disable RGB output

`-alpha boolean`

Enable/disable Alpha output

`-depth boolean`

Enable/disable Depth output

`-iip`

Disable all image planes before rendering

`-res int int`

Specify the resolution (X Y) of the rendered image

`-crop float float float float`

Specify a crop window for the rendered image

Render Layers:

`-rl boolean|name(s)`

Render each listed layer separately

MEL callbacks:

`-pre string`

MEL code executed before each frame

`-post string`

MEL code executed after each frame

- **MEL callbacks for Maya 7.0**

  `-preRender string`

  MEL code executed before rendering

-postRender string

MEL code executed after rendering

-preLayer string

MEL code executed before each render layer

-postLayer string

MEL code executed after each render layer

-preFrame string

MEL code executed before each frame

-postFrame string

MEL code executed after each frame

Bake Options:

-bake int

- **0**: Don't bake, but do regular rendering
- **1**: Bake texture maps
- **2**: Bake texture maps and do regular rendering

-bakeChannels string

Comma delimited list of one or more channels: _ambient,_diffuse,_diffuse_noshadow, _incandescence,_indirect,_indirectdiffuse, _irradiance,_occlusion,_reflection,_refraction, _shadow,_specular,_subsurface,_surfacecolor, _translucence

-bakeResolution int int

Set X Y resolution of baked maps

-bakeCamera string

Camera to use while baking

-bakeFileFormat string

File format of output images: Alias, Cineon, It, MayaIFF, OpenEXR, SGI8, SGI16, SoftImage, Targa, Tiff8, Tiff16, Tiff32

-bakeFileDepth string

Depth of output images: byte, short, float

Other:

-rep boolean

Do not replace the rendered image if it already exists

-n int

Number of processors to use. `0` indicates use all available.

-compile boolean

Forces compilation of all shaders, even if they already exist.

● **Important**

- Remember to place a space between option flags and their arguments.
- Any boolean flag will take the following values as TRUE: on, yes, true, or 1.
- Any boolean flag will take the following values as FALSE: off, no, false, or 0.

Additional options available in RenderMan for Maya Pro can be found in the RfM and RIB section of the RenderMan Studio documentation. Additionally, a complete list of the options can also be seen by running the `Render -r rman -h` command.

Prev | Next

The Render Menu and Render View

The Render Menu

To access Maya's Render menu you need switch to Maya's Rendering Menu set (shortcut is F5). RenderMan for Maya (when set as the active renderer) will display the following menu items in the Render menu:

Render Current Frame

Renders an image using Maya's currently active view as the camera (this behavior can be overridden). While this may appear to be a single process, two things are going on: first RenderMan for Maya translates the scene, then Pixar's RenderMan, the core rendering engine, renders the images. As the image renders, it will appear in the Render View.

Redo Previous Render

Renders from the camera used during the last render, regardless of the active window.

Test Resolution>

A selection of resolution presets for preview rendering.

Camera Panel

Uses the resolution of the active Maya window.

Render Globals

Uses the full resolution as defined in the Maya Render Globals window; the default setting is 320x240.

50% Globals

50% of the resolution as defined in the Render Globals window.

25% Globals

25% of the resolution as defined in the Render Globals window.

10% Globals

10% of the resolution as defined in the Render Globals window.

Batch Render

Starts a batch render of the Maya file. The images are rendered to a directory, defined by the user. Batch render jobs run in the background, as a separate process, which has the benefit allowing the user to continue to work in the current Maya session (as long as the computer system has sufficient resources).

Render using>

Specify a renderer.

RenderMan

Sets RenderMan as the active renderer.

Other Renderers

Select from a variety of other supported renderers for Maya.

Note

The menu items shown in Maya's Render menu will change depending on which renderer is active.

The Render View

RenderMan for Maya displays rendering images in Maya's Render View. The Render View has its own selection of menus and buttons. All of these features are functional (except for IPR) and can be used seamlessly with RenderMan for Maya. See the Maya documentation for more information.

To Open the Render View:

1) Select the following:

```
Window > Rendering Editors > Render View
```

To switch Renderers:

The active renderer can be set in the following ways:

1) Choose a renderer from drop-down menu.

2) Select a renderer from the Render View sub-menu:

```
Options > Render> [pick a renderer]
```

Prev | Next

Transitioning from RAT to RMS

If you're familiar with using the RenderMan Artist Tools and MTOR, RenderMan Studio (RMS) is going to present a slightly different workflow. The RMS way of doing things is multifaceted :

1. Wherever possible, use the natural Maya-centric method for achieving/controlling various affects.
2. Never sacrifice the RenderMan-centric method of doing things.

We achieve #1 by carefully translating all the effects and knobs that we are aware of) As Maya is a moving target, we're always on the lookout for holes in our translation and new or changed Maya features and workflows.

We achieve #2 by providing you direct access to the Ri stream (Ri Injectors) and dynamic attributes that can be assigned to shaders, lights, shapes, and the camera, or wired-in for sharing across all these. The collection of attributes and options available through these channels is completely open-ended. The files `$RMSTREE/etc/rfm_param.rman`, `rfm_nodetemplate.rman`, etc. embody the policies presented in the GUI and, should you need something we haven't provided, itis possible to extend the system using our standard extension machinery.

For example, you can add a custom Matte attribute to primitive shape nodes using the *RenderMan-> Manage Attributes* menu item in the Maya's Attribute Editor. There, you'll find the entire list of powerful and obscure RenderMan options for assignment to your objects. You can also create a node representing the sharing of a collection of RenderMan attributes and wire these into a bunch of shape nodes via *Window-> Rendering Editors-> RenderMan-> Shared Geometric Attributes*.

RenderMan attributes are divided into a few categories:

- `riopt` and `riattr` represent standard Ri Attributes and Options that can vary per-shape (RiAttributes) or per-frame (RiOptions).
- `toropt` and `torattr` represent translator configuration knobs. `toropts` are intended to be job and/or frame-wide; `torattrs` are intra-frame. If, for example, you don't want the translator to output surface shaders during a particular pass, there's a `torattr` that controls this.

For additional practical information on making the transition from RAT to RMS, please consult the RAT to RMS FAQ in the *Recipes and Tutorials* section of the documentation.

Render Settings

Selecting *RenderMan* from the *Rendering Using* menu causes several tabs to appear which provide control over global RenderMan settings. These tabs are: *Common*, *Quality*, *Features*, *Passes*, *Advanced*.

> The RenderMan Settings are accessible via Maya's *Render Settings*, which can be opened via clicking on an icon (e.g. in the RenderMan Shelf) or via the *Window* menu:
>
> - `Window> Rendering Editors> Render Settings`
>
> The RenderMan Settings can also be controlled directly through MEL scripts. For more information about the Render Settings attributes names refer to: MEL Scripting> How to set Render Global Values

Prev | Next

Common

The Common Tab controls setting shared between all Maya renderers and includes some special features specific to RenderMan for Maya.

(Not Pictured)
Path: Displays the current project setting.
File Name: Displays the output name of the final image.
Image Sise: Displays the resolution (generally in pixels and inches).

Image File Output
The name of rendered image files can be composed of up to three elements: file name, frame number, and file format.

File Name Prefix: Determines the base name for rendered images.

Frame/Animation Ext: The format used to name output images.

Image Format: Specify which format to save output image files.

Start Frame, End Frame: Specifies the first and last frames to render. Frame/Animation Ext must be set to an option containing # to enable these parameters, otherwise a single frame will be used.

By Frame: The increment between the output names of rendered frames. By Frame is only available if Frame/Animation Ext is set to an option containing #.

Frame Padding: The minimum number of digits in frame number extensions. For example, if Frame/Animation Ext is set to renderman.ext, and Frame Padding is 2, Maya names rendered image files renderman.01, renderman.02, and so on.

Renderable Camera: Select your render camera from this list.

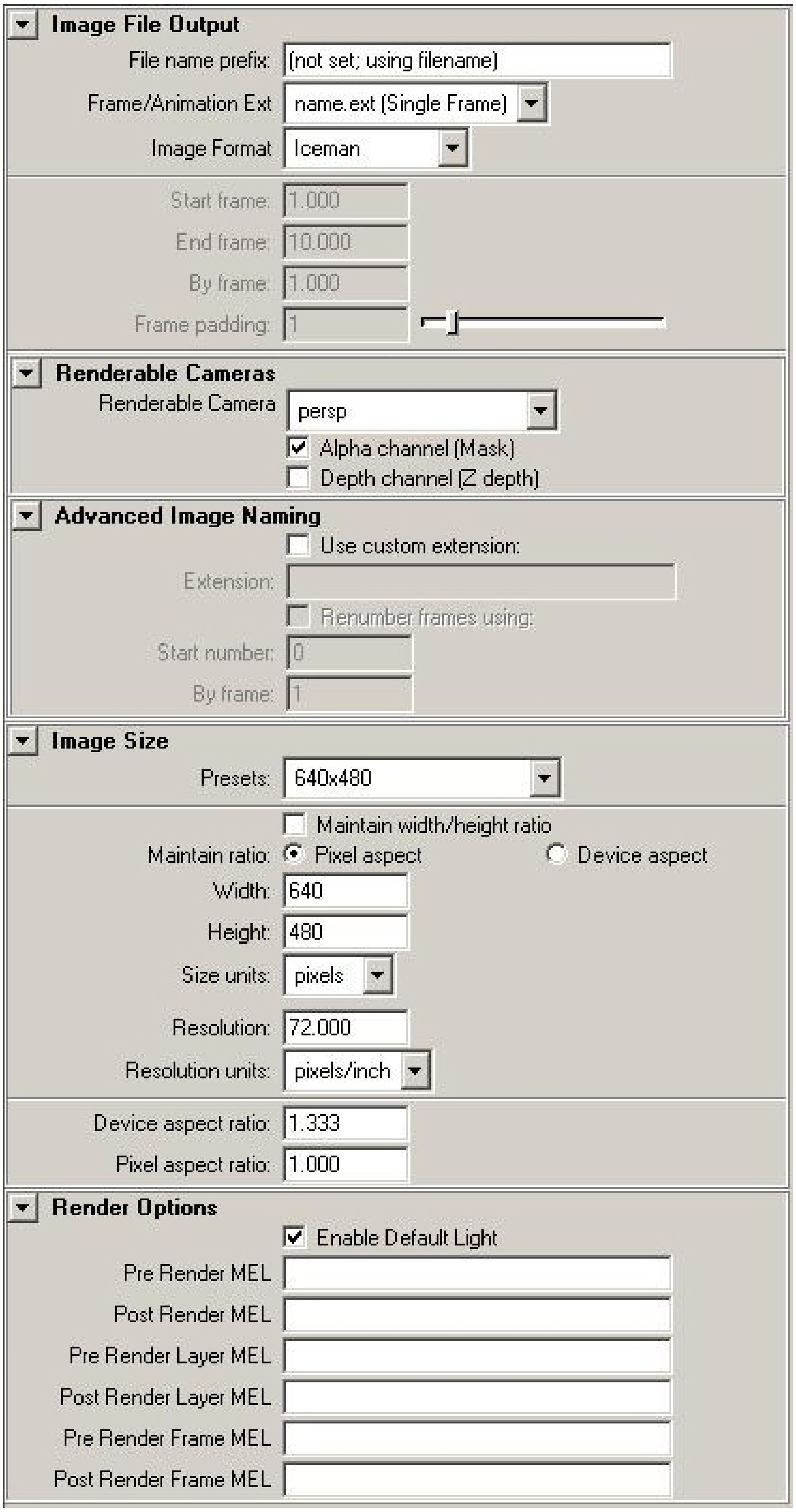


Image File Output
File name prefix: (not set; using filename)
Frame/Animation Ext: name.ext (Single Frame)
Image Format: Iceman
Start frame: 1.000
End frame: 10.000
By frame: 1.000
Frame padding: 1

Renderable Cameras
Renderable Camera: persp
☑ Alpha channel (Mask)
□ Depth channel (Z depth)

Advanced Image Naming
□ Use custom extension:
Extension:
□ Renumber frames using:
Start number: 0
By frame: 1

Image Size
Presets: 640x480
□ Maintain width/height ratio
Maintain ratio: ● Pixel aspect ○ Device aspect
Width: 640
Height: 480
Size units: pixels
Resolution: 72.000
Resolution units: pixels/inch
Device aspect ratio: 1.333
Pixel aspect ratio: 1.000

Render Options
☑ Enable Default Light
Pre Render MEL:
Post Render MEL:
Pre Render Layer MEL:
Post Render Layer MEL:
Pre Render Frame MEL:
Post Render Frame MEL:

- If your scene has only one renderable camera (for example, the perspective camera), whatever camera is selected in this drop-down list becomes the renderable camera. In other words, the newly selected camera becomes the renderable one in your scene (the default perspective camera becomes unrenderable).

Alpha Channel (Mask): Determines whether rendered images contain a mask channel.

Depth Channel (Z Depth): Determines whether rendered images contain a depth channel.

Advanced Image Naming

Use Custom Extension: Create a custom extension by enabling Use Custom Extension, and entering your custom extension in the field. This extension replaces the standard extension based on file format, i.e. .tif, .tex, etc.

Renumber Frames: Adjust the numbering of rendered image files for an animated sequence. The Renumber Frames attributes are only available if Frame/Animation Ext is set to an option with # (such as name.#.ext).

Start Number: The frame number extension you want the first rendered image file name to have.

By Frame: The increment between frame number extensions you want rendered image file names to have.

Image Size

Presets: Choose a resolution for your output images based on a selection of industry standards.

Maintain Width/Height Ratio: If this box is checked, custom resolutions will maintain a locked ratio between heigth and widht. By disabling this parameter, any numbers may be enter for image resolution.

Width/Height: For custom resolutions, enter the desired resolutions here.

Size Units: Select an appropriate unit of measurement.

Device Aspect Ratio: The aspect ratio of your target display (e.g. it or Maya's Render View).

Pixel Aspect Ratio: The aspect ratio of the individual pixels of the display device on which you are viewing the rendered image.

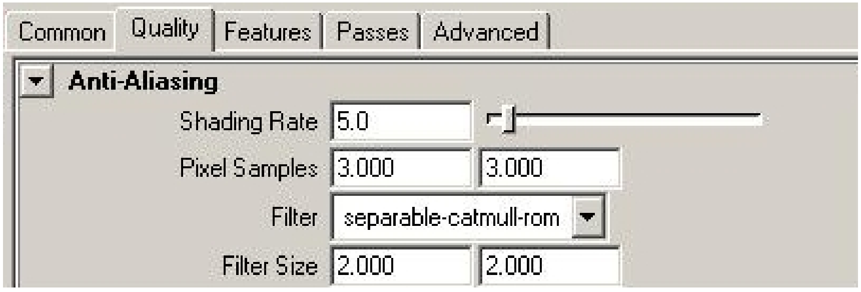
Render Options

Enable Default Light: Turn the default lighting on or off during rendering.

Pre/Post MEL scripts: There are fields for entering MEL scripts/commands to be run at the specified times.

Quality

Anti-Aliasing



Shading Rate 5.0
Pixel Samples 3.000 3.000
Filter separable-catmull-rom
Filter Size 2.000 2.000

Shading Rate: Shading Rate is the main global quality control for Pixar's RenderMan. It's also probably the second most critical factor in speed performance (next to the resolution). Shading Rate controls two important factors. First, Shading Rate governs the quality of shaders. Smaller numbers mean the shaders must be evaluated at more places on the surface of the primitives. Second, Shading Rate governs how much detail is created on a per-pixel basis, and more detail means more work, requiring more hidden-surface evaluation and more memory. The end result of all this is that doubling the Shading Rate (from 1 to 2) usually gets you nearly twice the rendering speed.

A typical Shading Rate for final rendering is 1.0, or less. Test renderings can usually be done at 3.0, 5.0, or even 20.0. What is the disadvantage? A shading rate that is too large tends to give blocky looking colors and excessive blur on textures. As long as the color of an object changes slowly and smoothly across its surface, this will look fine.

One of the most important things to remember about the Shading Rate is that it is an Attribute. That is, it can be set globally or it can be changed from one primitive to the next. So if you have a finely patterned vase sitting in a room with flat white walls, the vase can have a small shading rate (like 1.0) to capture its detail while the walls can have a very large shading rate (like 64.0) to save time (with no undesired artifacts). Per-object Shading Rates are a very powerful technique that can be used to tell the renderer how much time to spend on individual objects.

Pixel Samples: Pixar's RenderMan provides advanced control over antialiasing, and "Pixel Samples" is used to control the amount of super-sampling around a pixel. For speed, this can be set to zero, with the expected loss of image quality. "Pixel Samples" controls hidden-surface antialiasing (which eliminates "jaggies") and low values (e.g. 1x1) can benefit both speed and memory use. Values of 4x4 and up are considered high quality values.

Filter: "Pixel Filter" controls filtering to smooth out other hard color transitions, but changing it has only a small effect on overall speed. However, the interaction of the selection of filters can have a profound effect on the "look" of the resulting image.

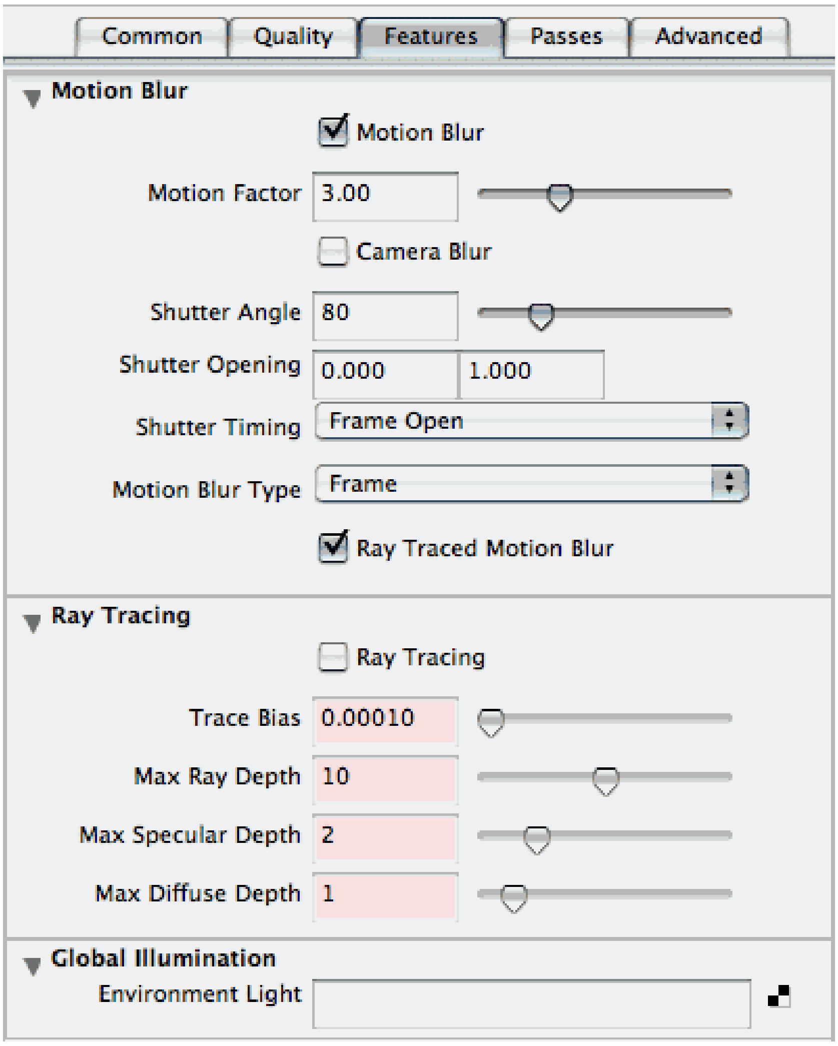
In general, a soft, photographic look can be obtained by using jitter and a wide gaussian filter. This will produce good-looking antialiased results at a relatively low supersampling rate. The Catmull-Rom filter, with jitter, results in an image that has sharper details, at a slightly increased cost. A harder-edged look can be obtained by not using jitter, increasing the amount of supersampling, and using a narrow box filter. Experimentation with these controls can allow

Filter Size: The filter width (measured in pixels, in each dimension) is used by the current filter and controls scope of the effect. The appropriate setting for Filter Size depends on which filter is chosen, but typical values range between 2 and 5. Setting this value to 1 eliminates filtering.

Features

Motion Blur

RenderMan has a number of controls for creating fast and efficient motion blur.



Motion Blur

☑ Motion Blur

Motion Factor 3.00

□ Camera Blur

Shutter Angle 80

Shutter Opening 0.000 1.000

Shutter Timing Frame Open

Motion Blur Type Frame

☑ Ray Traced Motion Blur

Ray Tracing

□ Ray Tracing

Trace Bias 0.00010

Max Ray Depth 10

Max Specular Depth 2

Max Diffuse Depth 1

Global Illumination

Environment Light

Motion Blur: Enabling motion blur causes moving objects to be blurred along their path of movement in order to prevent temporal aliasing and strobing. When using motion blur, you should increase the number of pixel samples (in the Quality subpanel), and expect longer rendering times.

Motion Factor: A special geometric approximation type, Motion Factor provides a processing/quality tradeoff for motion blurred objects because the objects are blurred, less detail is required and a higher shading rate will be sufficient. A Motion Factor value of 0.0 will turn this feature off. Values greater than 1.0 will cause motion blurred objects to have their effective shading rate raised (the default value is 3).

Camera Blur: The renderer will ignore the movement of the camera when computing motion blur when this feature is disabled. If you do move the camera and want everything in the scene to blur accordingly, you will need to turn this feature on; however, you should expect this to be a potentially expensive operation since everything in the scene will now undergo motion blurring calculations.

Shutter Angle: This setting is only meaningful when Motion Blur is enabled. It controls how long the shutter is open during a frame. Measured in degrees a setting of 360 opens the shutter for the entire frame. The larger the angle, the greater the blur, which increases render times.

Shutter Timing: This field allows you to control when the virtual camera shutter opens, and is only meaningful when Motion Blur is enabled. The default value Open on Frame causes the shutter to be open from frame to frame + shutter angle. If you choose Center on Frame, the shutter will be open from frame - shutter angle/2 to frame + shutter angle/2. Note that, in RenderMan, shading calculations occur only at frame open time, and thus shadows and reflection maps will be computed only at shutter open time.

Motion Blur Type: The Frame option causes the positions of all objects to be calculated only at frame boundary times, and will rely on the renderer to perform linear interpolation of the object positions in order to determine where they are at shutter open and close times. The Frame option results in faster rendering, at the expense of less accurate motion blur. Enabling subframe motion will cause the position of objects to be calculated exactly at shutter open and close time, creating blur with higher fidelity, at a slightly increased cost.

Ray-Traced Motion Blur: Controls whether motion blurred objects appear in ray traced results. If your reflective object is moving or deforming you should disable this parameter to avoid self-intersection artifacts. If your mirror isn't moving, set this to 1 to see the reflected objects properly blurred.

Ray Tracing

Ray tracing provides a system for creating a variety of effects, from reflections and refractions to fancy global illumination effects like occlusion and color bleeding. Occlusion determines the amount a point is obscured by other surfaces, creating subtle shadows. Color bleeding is similar to occlusion but bleeds color from other materials. Occlusion and color bleeding data are calculated by casting many hemispherical samples from a given surface point.

Ray Tracing: Control for enabling/disabling ray tracing. Ray tracing effects can be expensive, so ray tracing can be turned off to enable faster scanline rendering.

Trace Bias: This value is added to the ray origin when tracing rays to overcome numerical precision issues resulting in false ray-primitive intersections. This value acts as the scene default and you can override this value on a per-primitive basis via RenderMan Attributes.

Max Ray Depth: This sets a limit to the number of bounces any ray can travel regardless of its type.

Max Specular Depth: Limits the number of specular bounces (reflections and refractions) for rays traced from the associated primitive. A value of 1 or 2 is a reasonable default unless you need multi-bounce effects. This value acts as the scene default and you can override this value on a per-primitive basis.

Max Diffuse Depth: Limits the number of bounces (diffuse or specular) for indirect illuminance relative to the associated primitive. You can use this attribute to limit the number of diffuse and specular bounces of photons in the photon tracing pass as well as to limit the depth of diffuse illuminance gathering shaders. A value of 1 is suggested. This value acts as the scene default and you can override this value on a per-primitive basis via RenderMan Attributes.

Global Illumination

Global illumination effects, such as occlusion, color bleeding, and Image-Based Illumination (IBI) can be created using the RenderMan Environment Light. Clicking on the checkered box creates a new node and displays the controls for the `RenderManEnvLight` in the Attribute Editor. Ray tracing must be enabled to create an Environment Light. For more information on global illumination, see the Global Illumination Tutorial.

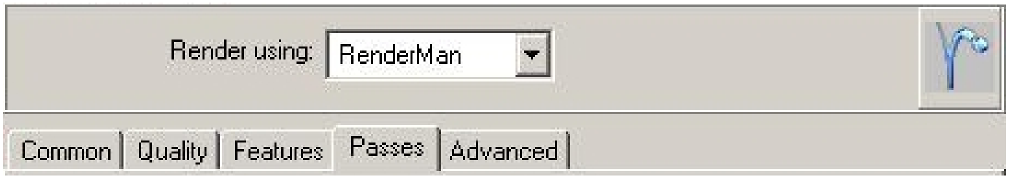
Prev | Next

Passes

RenderMan for Maya has a flexible system for the creation of custom passes. Passes can be created and deleted. When a pass is selected the settings for that pass will appear. Passes can be used to generate shadow maps, environment maps, reference images, occlusion effects, etc. In fact, RenderMan for Maya allows the creation of custom passes. Additionally, RenderMan for Maya observes Maya's render layers, when enabled (via the check box). See the Maya documentation for more information.

The Passes Tab

When you first open the Passes tab you are presented, by default, with the outputs for your Final pass. You can switch to existing passes, create new ones, or apply filters by clicking on the arrow next to the Pass window to access a pop-up window. Each pass has a separate tab for Pass Settings and Outputs.

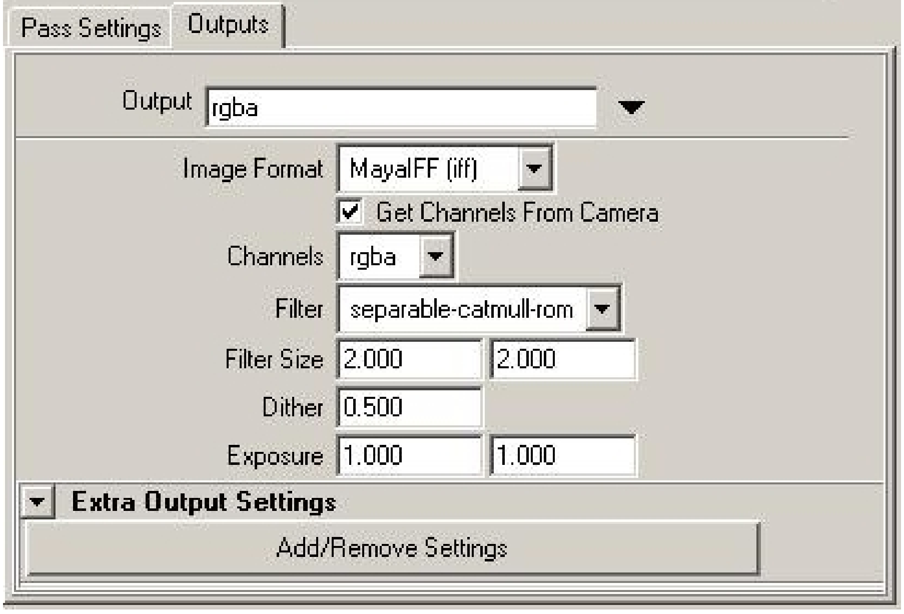


Render using: RenderMan

The Outputs Sub-Tab

The default output of the Final Pass is, as you might expect, rgba, the color and alpha channels of a typical image. You can select other outputs, including Custom outputs by clicking on the arrow to the right of the output window.

The Outputs Sub-Tab also includes settings for the output *Image Format*, *Channels*, *Filter*, *Filter Size*, *Dither*, and *Exposure*. The defaults and options for these settings will change, depending on the output you have selected (e.g. the default image format for rgba is *mayaiff*, but for _shadow it is *tiff*). For more information on outputs, please see the Outputs section.



Output: rgba

Image Format: MayaIFF (iff)

☑ Get Channels From Camera

Channels: rgba

Filter: separable-catmull-rom

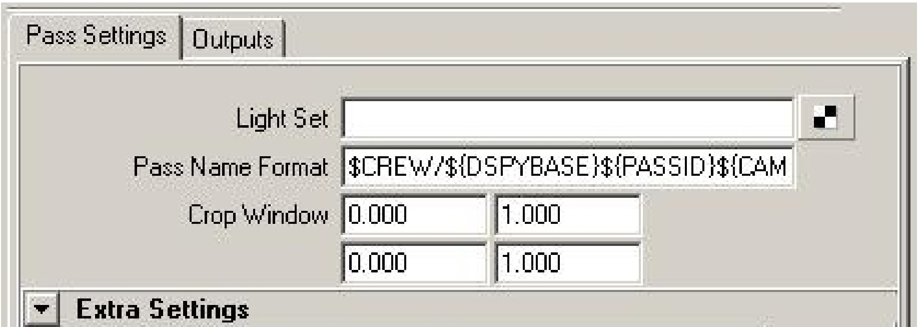
Filter Size: 2.000 2.000

Dither: 0.500

Exposure: 1.000 1.000

The Settings Sub-Tab

The Pass Settings for the Final pass are, essentially, pre-defined, but you can add extra settings, such as Bucket Size , for additional control over the



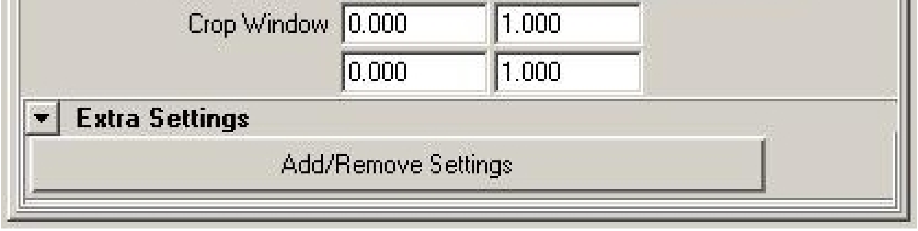
Light Set:

Pass Name Format: $CREW/${DSPYBASE}${PASSID}${CAM

Crop Window: 0.000 1.000

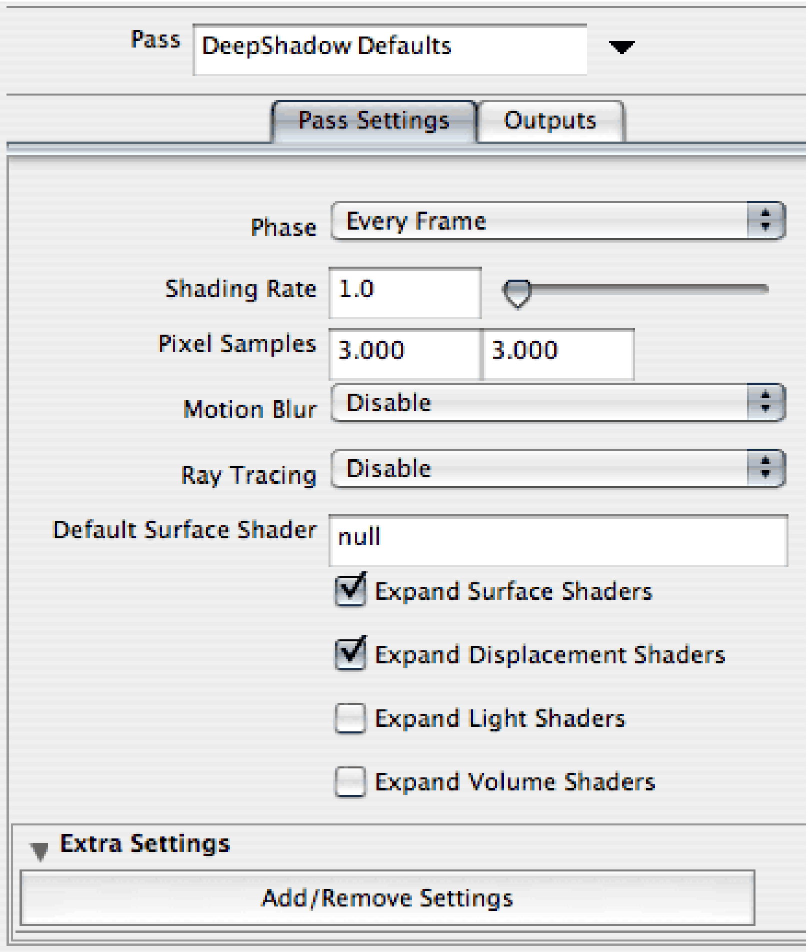
0.000 1.000

rendering, via this sub-tab.



Crop Window 0.000 1.000
0.000 1.000

Auto-Updating Sub-Menus for Individual Passes
When a pass is selected in the Pass Window, the settings for that pass will appear in the *Settings* tab. In this example, the *DeepShadow Defaults* have been selected. The settings used for different passes are similar, so let's take a quick look at the DeepShadow settings.



Pass: DeepShadow Defaults

Phase: Every Frame
Shading Rate: 1.0
Pixel Samples: 3.000 3.000
Motion Blur: Disable
Ray Tracing: Disable
Default Surface Shader: null
☑ Expand Surface Shaders
☑ Expand Displacement Shaders
☐ Expand Light Shaders
☐ Expand Volume Shaders

Phase: [Every Frame, Once Per Job] Determines how often passes are computed. Every Frame causes passes to be computed every frame, which is preferable if the objects in a pass are changing from frame to frame. Otherwise, if objects in a map do not change through an animated sequence, "Once per Job" can be used which can save valuable rendering cycles.

Shading Rate: See, Shading Rate

Pixel Samples: See, Pixel Samples

Motion Blur: See, Motion Blur. Note that this setting does not appear in the regular *Shadow Defaults*.

Ray Tracing: See, Ray Tracing

Default Surface Shader: Useful in special cases, like occlusion. Allows a single shader to be attached to all objects in a pass. By default this is null .

Expand Surface Shaders: If surface shaders are needed in a pass, this setting can be enabled so that surface shaderswill be calculated. Turning this off causes surface shaders to be ignored, speeding up rendering calculations.

Expand Displacement Shaders: If displacements are needed in a pass, this setting can be enabled so that displacements will be calculated. Turning this off causes displacements to be ignored, speeding up rendering calculations.

Expand Light Shaders: If light shaders are needed in a pass, this setting can be enabled so that lights will be calculated. Turning this off causes lights to be ignored, speeding up rendering calculations. Lights aren't important in shadows, but are needed for reference images.

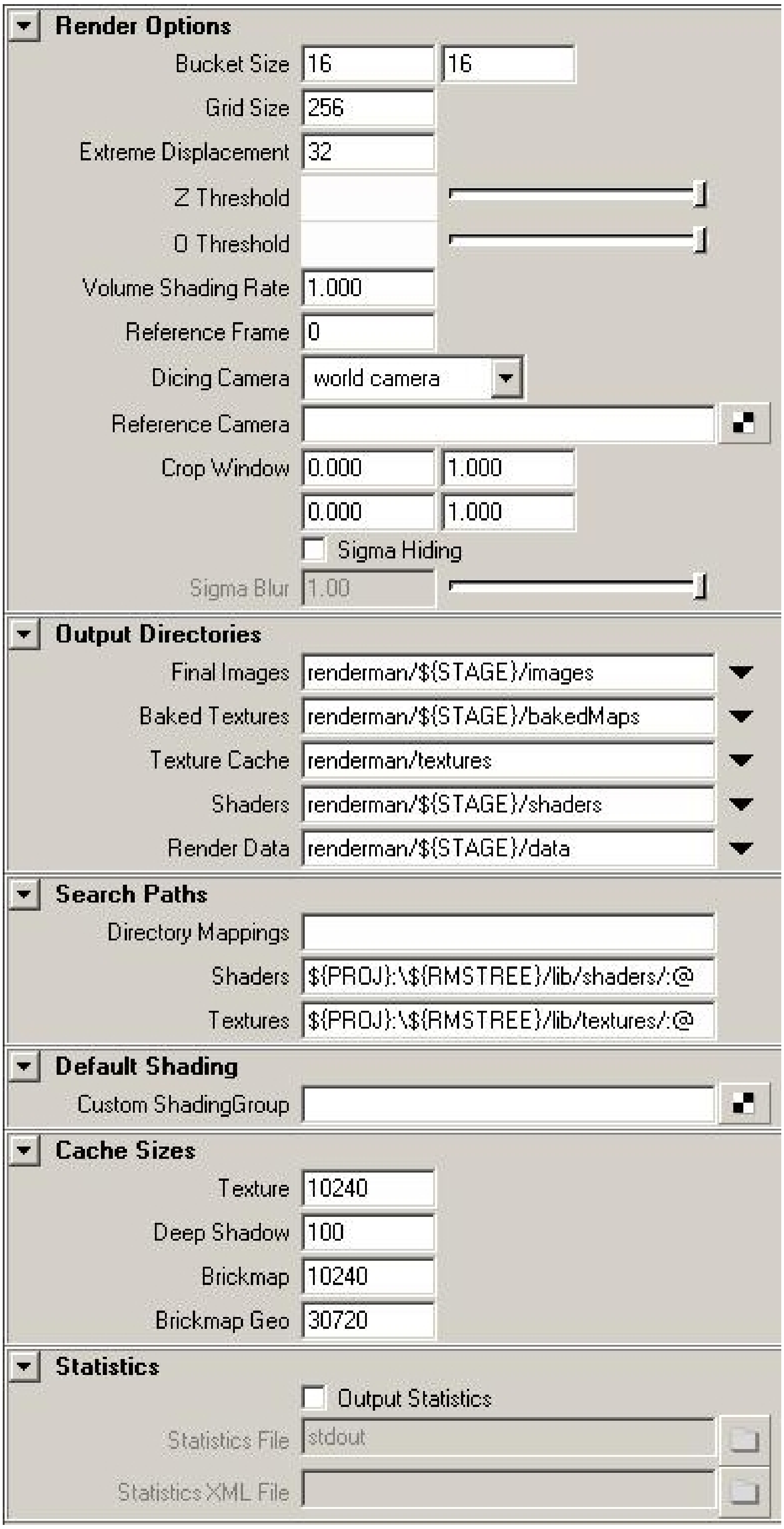
Expand Volume Shaders: If volume shaders are needed in a pass, this setting can be enabled so that volume shaders will be calculated. Turning this off causes volume shaders to be ignored, speeding up rendering calculations.

Prev | Next

Advanced

Bucket Size: RenderMan subdivides the output image into small rectangular regions called *buckets*, and renders a single bucket at a time. This allows a small portion of the image to be in memory at any one time. Larger buckets, in conjunction with a larger *grid size*, allow shading of more micropolygons in parallel and is thus more efficient, however this will require more memory to render. Conversely, smaller buckets are less efficient for shading, but will use less memory during rendering. If your scene is using a lot of memory, you may want to try setting this field down to 8 by 8 or even 4 by 4 buckets.

Grid Size: In RenderMan, geometry is split into *grids*, which are then shaded at the same time. Larger grids (which will require a larger *bucket size* to produce) allow more geometry to be shaded at once, which is more efficient at the cost of memory. Note that the grid size is also affected by the *shading rate*, because a larger shading rate requires a bigger grid, which may not fit in the current bucket.



Render Options

Bucket Size 16 16

Grid Size 256

Extreme Displacement 32

Z Threshold

O Threshold

Volume Shading Rate 1.000

Reference Frame 0

Dicing Camera world camera

Reference Camera

Crop Window 0.000 1.000

0.000 1.000

□ Sigma Hiding

Sigma Blur 1.00

Output Directories

Final Images renderman/${STAGE}/images

Baked Textures renderman/${STAGE}/bakedMaps

Texture Cache renderman/textures

Shaders renderman/${STAGE}/shaders

Render Data renderman/${STAGE}/data

Search Paths

Directory Mappings

Shaders ${PROJ}:\${RMSTREE}/lib/shaders/:@

Textures ${PROJ}:\${RMSTREE}/lib/textures/:@

Default Shading

Custom ShadingGroup

Cache Sizes

Texture 10240

Deep Shadow 100

Brickmap 10240

Brickmap Geo 30720

Statistics

□ Output Statistics

Statistics File stdout

Statistics XML File

Extreme Displacement: When the displacement of an object on the screen is very large, that is, the displaced point is far from the original point, RenderMan invokes a special displacement procedure, *Extreme Displacement* in order to save large amounts of memory, though at some additional computational cost. The maximum permissible displacement before the special procedure is invoked is measured in vertical scanlines. If this value is increased, larger displacements are permitted to use memory rather than incur the additional computation. If this value is decreased, memory usage is minimized even for less severe displacements. The default value is 32 scanlines.

Z Threshold: Control for adjusting the threshold of Z.

O Threshold: Controls the O Threshold.

Volume Shading Rate: Contols the volume frequency relative to the global Shading Rate. Set this value to 0 to shade volumes at all subsamples.

Reference Frame: Once per job passes and static objects are evaluated at the frame designated here.

Dicing Camera: Enables the use of a reference camera for dicing. This can be done globally, or restricted to specific objects (e.g. objects with displacements that sparkle as the camera moves) by also attaching this attribute per-object.

Reference Camera: Specify the camera to be used as your dicing camera here. Click on the widget to create a new camera, or right click in the field to select an existing camera. Note that the Reference Camera needs to be different from render camera.

Output Directories
The Output Directories can be used to specify where RenderMan for Maya puts rendering output. By default assets are placed in the *renderman* directory of your current Maya project.

- **Final Images:** Where final renders are placed.
- **Baked Textures:** Specifies a directory for pre-baked texture maps.
- **Texture Cache:** Textures that are converted to Pixar's file format on the fly are placed here.
- **Shaders:** Sets the location of shader assets.
- **Render Data:** Other misc data is placed here.
- **Rib:** RIB files are stowed here (RfM Pro only).

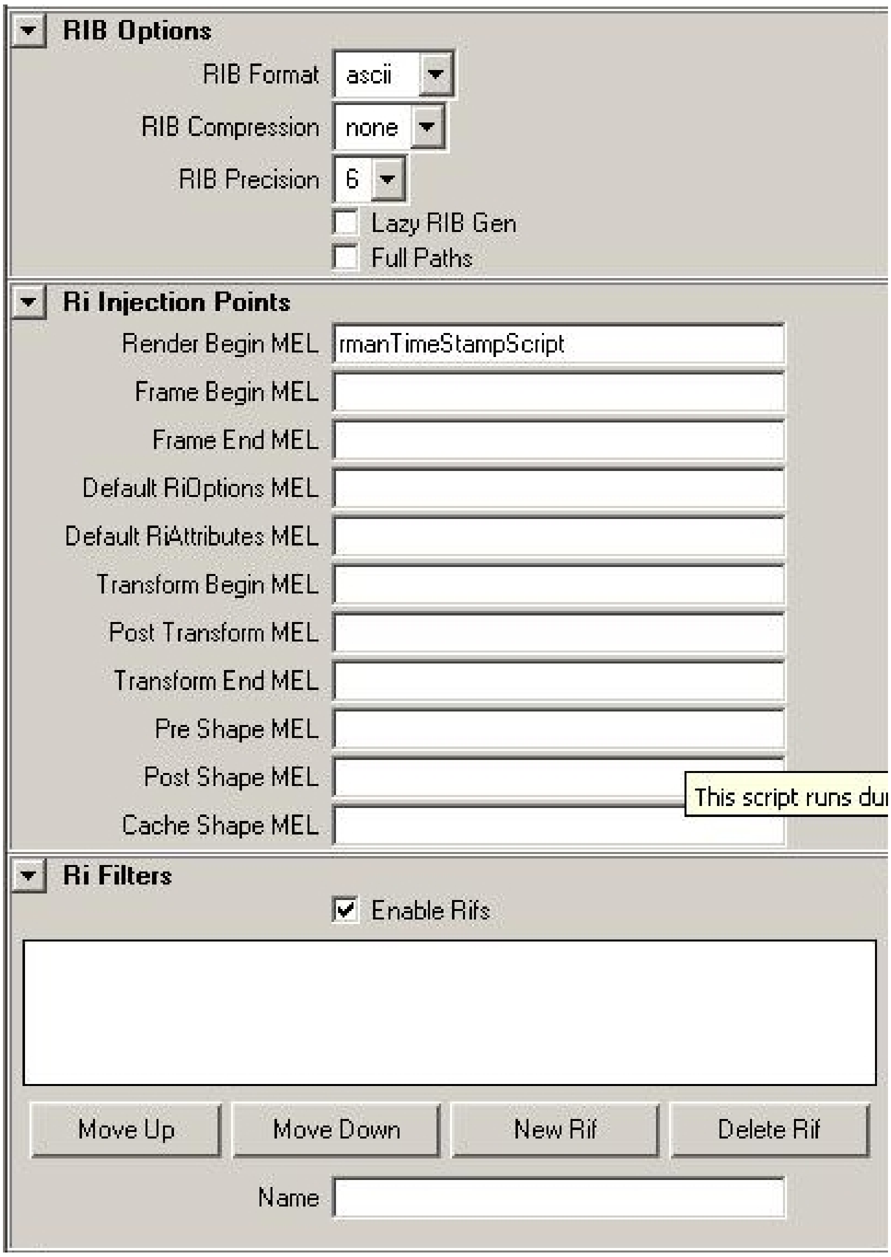
Directory Options: Click on the triangular widget to the right of each Output Directory specification to access additional options, including **Job/Frame Cleanup** (where applicable).

Search Paths: These allow you to map directories and specify a series of paths for your resources. RfM Pro includes settings for RIB Archives and Procedurals.

Statistics: Enable this to output information from render jobs. Useful for timing, debugging, and seeing what RenderMan is doing. You can specify a target for both Old School and XML statistics.

RenderMan for Maya Pro Settings

RenderMan for Maya Pro includes additional controls under the Advanced tab, including a dropdown menu to choose between ASCII and binary RIB formats, compression, and precision, a checkbox to enable lazy RIB generation (unchanged textures, shaders, and/or archives are not regenerated), use of fully-qualified paths in RIB files, entry point specifications for Ri for MEL scripts, and an interface for creating and managing Ri Filters.



RIB Options

RIB Format: ascii
RIB Compression: none
RIB Precision: 6
□ Lazy RIB Gen
□ Full Paths

Ri Injection Points

Render Begin MEL: rmanTimeStampScript
Frame Begin MEL:
Frame End MEL:
Default RiOptions MEL:
Default RiAttributes MEL:
Transform Begin MEL:
Post Transform MEL:
Transform End MEL:
Pre Shape MEL:
Post Shape MEL:
Cache Shape MEL:

This script runs dur

Ri Filters

☑ Enable Rifs

Name:

Prev | Next

Pixar Animation Studios

Attributes

Attribute Overview

Attributes

RenderMan has many unique qualities, from advanced features to rendering knobs and buttons. These unique qualities can be controlled using RenderMan specific attributes, and RenderMan for Maya provides numerous strategies for controlling these attributes. There are several levels of granularity that attributes may be controlled: *job*, *display*, and *object*.

Job Attributes - Define global settings for an entire scene.

Display Attributes - Define settings for individual render passes.

Object-Level Attributes - Allow attributes to be set on a per-object basis.

Priorities

Job attributes affect the scene by default, *display attributes* override job attributes, and *object-level attributes* override display attributes. In is important to note that before object-level attributes can be effective, they must first be set explicitly. RenderMan for Maya does not populate the Maya scene with unneeded RenderMan attributes.

For example, the quality control attribute, *Shading Rate*, is defined in the Render Globals. That job attribute setting, however, could be overridden during the execution of a shadow pass by adjusting the Shading Rate for that shadow pass, using the RenderMan Passes tab. Taking it one step further, Shading Rate attribute could be set on a particular object, by adding an object-level attribute to the object, which would override both the Render Globals and Passes settings. Here is a list of the priorities (object wins over display, display wins over job):

1) Object Attributes:

Object-level Attributes, which are attached directly to specific Maya surfaces, lights, and shaders

Shared Attribute Nodes, which are attached to specialty RenderMan specific nodes

2) Display Attributes:

Render Pass

3) Job Attributes:

Render Globals

RenderManforMaya.ini

Object Attributes

Object Attributes

The render settings for an object, like a 3D model, are generally inherited from the Render Globals. These settings, however, can be adjusting on a per-object basis by explicitly adding RenderMan attributes. An attribute that is explicitly set for an object will override the Render Globals setting. Object attributes may also be attached to special RenderMan for Maya nodes, which can be shared among these objects.

Adding Object Attributes

RenderMan for Maya provides two methods of attaching attributes to objects, either by using Maya's main menus or the Attribute Editor:

To create attributes directly on surfaces, lights, or shaders:

1) Select the *shape* node you'd like to attach attributes to

2) Open the attribute editor:

```
Select-> Attribute Editor: Attributes> RenderMan> Manage Attributes
```

3) Use the UI to add new attributes to an object.

4) These attributes can be adjusted via the shape node's respective *Extra RenderMan Attributes*

Creating shared attribute nodes:

1) Open the RenderMan Attributes Nodes UI

```
Maya: Window> Rendering Editors> RenderMan>
```

2) Select *create* from the menu

3) Next select *edit*, the shadred attribute node appears in the attribute editor.
4) Add attributes directly to the node as outlined about in method #1.

5) Use the menus in the UI to attach, detach, or delete these shared nodes.

● **Important**

Shared attribute nodes allow one attribute setting to affect many different objects and are a powerful tool that can be used to streamline workflow.

Context Specific Object Attributes

When attaching attributes using the Attribute editor, RenderMan for Maya will present a list of context specific attributes for the type of object that is selected (in the Attribute Editor RenderMan menu). Certain attributes are relevant only when attached to certain types of objects. Some attributes are meant to control lights, shaders, or surfaces. You are free, however, to still attach any attribute you see fit.

Prev | Next

Prev | Next

Display Attributes

Display Attributes

The settings for a render pass, like a shadow map, are controlled by the display attributes for that pass. These settings can be adjusting using the Passes tab of the Render Globals. The attributes not explicitly set for a pass will be inherited from the Render Globals.

Adding Display Attributes

For any type of pass additional attributes may be added. Additional attributes can be helpful to add in situations where a specific pass requires controls that the pass does not have by default. For instance, a reference pass may require additional attributes to control effects specific to that pass, like shading rate. To add additional attributes to a pass:

1) Open the Passes tab of the Render Globals.
2) Switch to the pass requiring additional attributes (using the pull-down menu next to the *Passes* field).
3) With the pass active, press the "Add/Remove Settings" button.
4) Add the desired attribute(s) from the pop-up menu.

Prev | Next

Job Attributes

The Anatomy of a Job

The job is the collection of tasks which comprise a single rendered frame, which includes sub-passes and other outputs. Settings are defined globally by the job attributes, but can be overriden by other attributes when finer level of control is needed.

In RenderMan For Maya, a job is defined as the series of Maya dag nodes that must be executed and evaluated in order to render a single frame. There are three levels of granularity: scene, pass, and object. RenderMan-specific *scene* settings are controlled by *job attributes*, settings that affect the scene globally. RenderMan-specific *pass* settings are controlled by *display attributes*. RenderMan-specific *object* settings are controlled via *object level attributes*.

Setting Job Attributes

Most of the time, job attributes will be set in the **Render Globals**. This is where all of the knobs and buttons exist for globally configuring settings for an entire Maya scene.

For advanced control, however, the RenderMan for Maya initilization file, *redermanformaya.ini,* defines all of the default settings and behaviors for RenderMan for Maya. Users and studios who wish to create their own custom defaults, including custom menus, can do so.

Maya Materials and Custom Shaders

Using Maya Materials

Material Support

RenderMan for Maya translates the majority of Maya Materials, including complex material hierarchies, bumps, and displacements. Using Maya Materials with RenderMan For Maya is as simple as creating Maya Materials for use with the Maya renderer. RenderMan for Maya supports the majority of Maya Materials, and a full list of supported Materials can be found **here**.

● **Note**

> RenderMan converts Maya Materials into high fidelity RenderMan shaders, which take advantage of RenderMan's high quality shading algorithms with can help prevent artifacts such as temporal aliasing and strobing, crucial when rendering sequences of images for animation.

Overriding Global Settings

Materials & Object Attributes

In addition to supporting the majority of Maya Materials, RenderMan for Maya allows special RenderMan attributes to be attached to Maya Materials. These attributes can be used to "unlock" many features uniques to RenderMan. For instance, attributes can be used to create special effects like Deep Shadows in lights and subsurface scattering in Maya Materials. Attributes can also be added to simply provide a few extra controls, like adding Displacement Bounds for displaced materials and other knobs and buttons useful for fine-tuning scenes.

These attributes that can be attached to geometry, materials, and lights are known as *Object Attributes*, and are an essential ingredient for taking full advantage of RenderMan for Maya. More information about using attributes can be found in the Object Attributes section.

Importing Custom RenderMan Shaders

Custom RenderMan Shaders

RenderMan for Maya allows users to import custom RenderMan shaders. RenderMan shaders are handled like regular Maya shading nodes, except they cannot accept connections from other Maya Material nodes. Imported RenderMan shaders must have been created previously, either written in the RenderMan Shading language or created using a shading tool like Pixar's Slim.

● Note

Once imported into Maya the parameters of these custom RenderMan shaders can be animated, but other Maya Materials cannot be wired into them. These RenderMan shaders can be connected to a top-level shading group, but these connections are constrained by the three available slots: surface, displacement, and volume.

Importing a Custom RenderMan Shader

Any RenderMan shader can be imported into RenderMan for Maya. The representation of the shader in Maya will have the correct UI, with all of the shader parameters. To import a custom shader follow these steps:

1) Open the Hypershade.
2) Create a *RenderMan Shader* node.

3) Open the *RenderMan Shader* node in the Attribute Editor.

4) For the *Shader* parameter, browse for your custom RenderMan shader.

5) Attach to objects in your scene.

● Note

Custom RenderMan shaders will render correctly only when use with the RenderMan renderer. They will not display correctly in other renderers, including Maya's interactive UI.

For more information about using RenderMan shaders with RenderMan for Maya, check out the RenderMan Shaders tutorial. RenderMan Studio users should also consult the Slim documentation.

Custom Shading Groups

In addition to individual RenderMan Shaders, you can create a *Custom Shading Group*, by attaching a `Custom ShadingGroup` attribute to individual nodes (including specific passes). This is particularly useful for doing things like creating global atmospheres.

You can add atmosphere shaders to your scene on a Global, Per-Pass, or Per-Shading Group basis. You can *Add Atmosphere Shader* via the Attributes menu of any Shading Group.

To add a Global atmosphere, create a new Custom Shading Group in the Advanced tab of the Render Settings, under **Default Shading**, then select *Add Atmosphere Shader* from the **Attributes> RenderMan>** menu, and, finally, add an atmospher shader (a compiled .slo file) via the browse button for the new setting.

To do it Per-Pass, select the Pass Settings subtab in the Passes tab of the Render Settings, click on the *Add/Remove Settings* button and add the `Custom ShadingGroup` setting, then proceed as above. Alternatively, you can add a *Default Atmosphere Shader* via *Add/Remove Settings*. You will need to provide an explicit path to the shader using the latter method.

Prev | Next

Prev | Next

RenderMan Environment Light

Introduction

Image based lighting and global illumation effects can be created with RenderMan for Maya by using the RenderMan Environment Light. You can create an Environment Light any of a number of ways: via Maya's *Hypershade*, via the Env Light icon in the RenderMan Shelf, or via the *Features* tab of the *Render Settings*. Below is a rundown of the User Interface for the RenderMan Environment Light, which is accessed via the Attribute Editor.

The User Interface

Environment Image An environment map may be added here. If there is no environment map, the Environment Color will be used instead.

Environment Color The environment color is set here. If there is an environment map, that will override Environment Color.

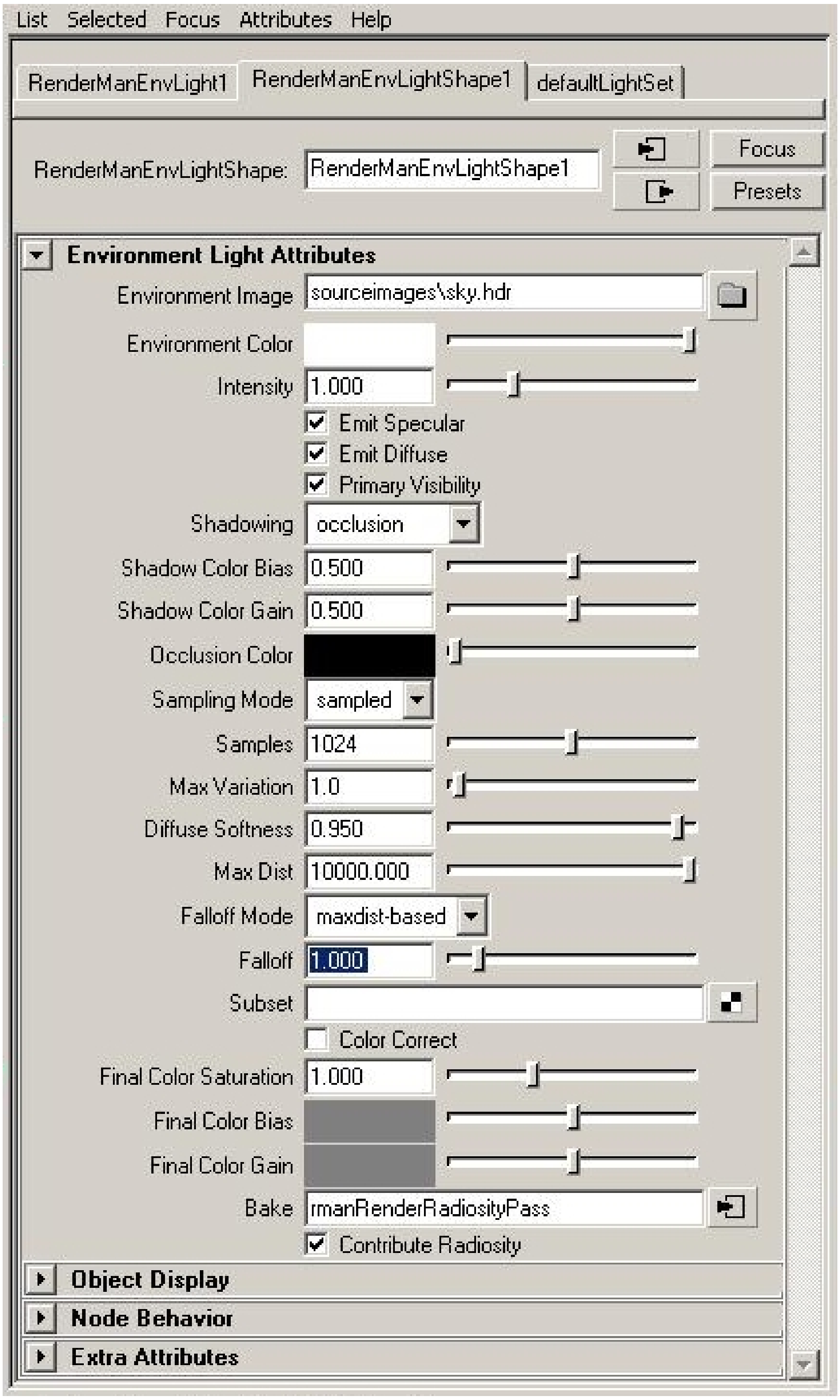
Intensity The brightness of the environment color or image.

Emit Specular Controls whether specular (reflective) light is projected from the light.

Emit Diffuse Controls whether diffuse light is projected from the light.

Primary Visibility Toggles whether the light itself appears in renderings.

Shadowing There are three shadowing modes: *none*, *occlusion*, and *colorbleeding*. When set to *none*, no hemispherical sampling is calculated. When set to *occlusion*, the amount that points are covered by geometry is calculated with hemispherical sampling. When set to *colorbleeding*, the color from nearby objects will bounce onto other objects, the amount of



RenderManEnvLightShape: RenderManEnvLightShape1

Environment Image: sourceimages\sky.hdr

Environment Color

Intensity: 1.000

☑ Emit Specular

☑ Emit Diffuse

☑ Primary Visibility

Shadowing: occlusion

Shadow Color Bias: 0.500

Shadow Color Gain: 0.500

Occlusion Color

Sampling Mode: sampled

Samples: 1024

Max Variation: 1.0

Diffuse Softness: 0.950

Max Dist: 10000.000

Falloff Mode: maxdist-based

Falloff: 1.000

Subset:

□ Color Correct

Final Color Saturation: 1.000

Final Color Bias

Final Color Gain

Bake: rmanRenderRadiosityPass

☑ Contribute Radiosity

which is determined by hemispherical sampling. Colorbleeding is more accurate, but also much more expensive, than occlusion.

Shadow Color Bias Darkens or lightens the shadow color when shadowing is enabled above.

Shadow Color Gain When shadowing is enabled, this parameter adjusts the contrast of the shadow.

Sampling Mode Choose the method to sample the environment map: *filtered*, *sampled*, or *baked*. *Filtered* is cheapest, but the least accurate. *Sampled* is the most accurate, but also the most expensive. It is required for image base illumination (IBI). Select *Baked* to reuse previously baked data; this works in conjunction with the Bake parameter.

Samples Specify the number of rays sent into the hemisphere above each point; more samples increases the quality of colorbleeding or occlusion, at the cost of increased render times.

Max Variation This is the quality/speed knob. Set this to "10" for quick/low quality renders. Set this to "1" or lower for high quality final frames.

Diffuse Softness fraction of the hemisphere above a point that is taken into account for ambient occlusion.

Max Dist The maximum distance, in world units, that a ray is shot. This value should be lowered for global illumination calculations for interior scenes.

Falloff Mode When falloff is enabled, you can specify the mode by which it is calculated. The *maxdist-based* mode is recommended for interior scenes. This parameter is disabled when the *Falloff* parameter (below) is set to 0 (zero).

Falloff The value of this parameter serves as power by which falloff is calculated. Falloff is disabled when this parameter is set to 0 (zero).

Subset A set of objects for inclusion in global illumination calculations can be chosen here (right-click for options).

Color Correct Enables color correction.

Final Color Saturation Adjust the intensity of the color.

Final Color Bias When Color Correct is enabled, this parameter lightens or darkens the color.

Final Color Gain This parameter adjusts the contrast when Color Correct is enabled.

Bake Connect a bake node here to compute the effect in a pre-pass for reuse later, or to calculate point-based global illumination. There are three baking methods to choose from: *RenderRadiosity* to render point-based global illumination, *MakeApproxGlobalDiffuse* to bake point-based GI data for reuse, and *MakeGlobalDiffuse3d* to bake ray-traced global illumination data for reuse. Additional configuration for the bake pass is done via the Passes Tab of the Render Settings.

Contribute Radiosity Enable or disable the radiosity contribution of the Environment Light.

Prev | Next

Prev | Next

Getting Started with The Tools

Prev | Next

Getting Started with Alfred

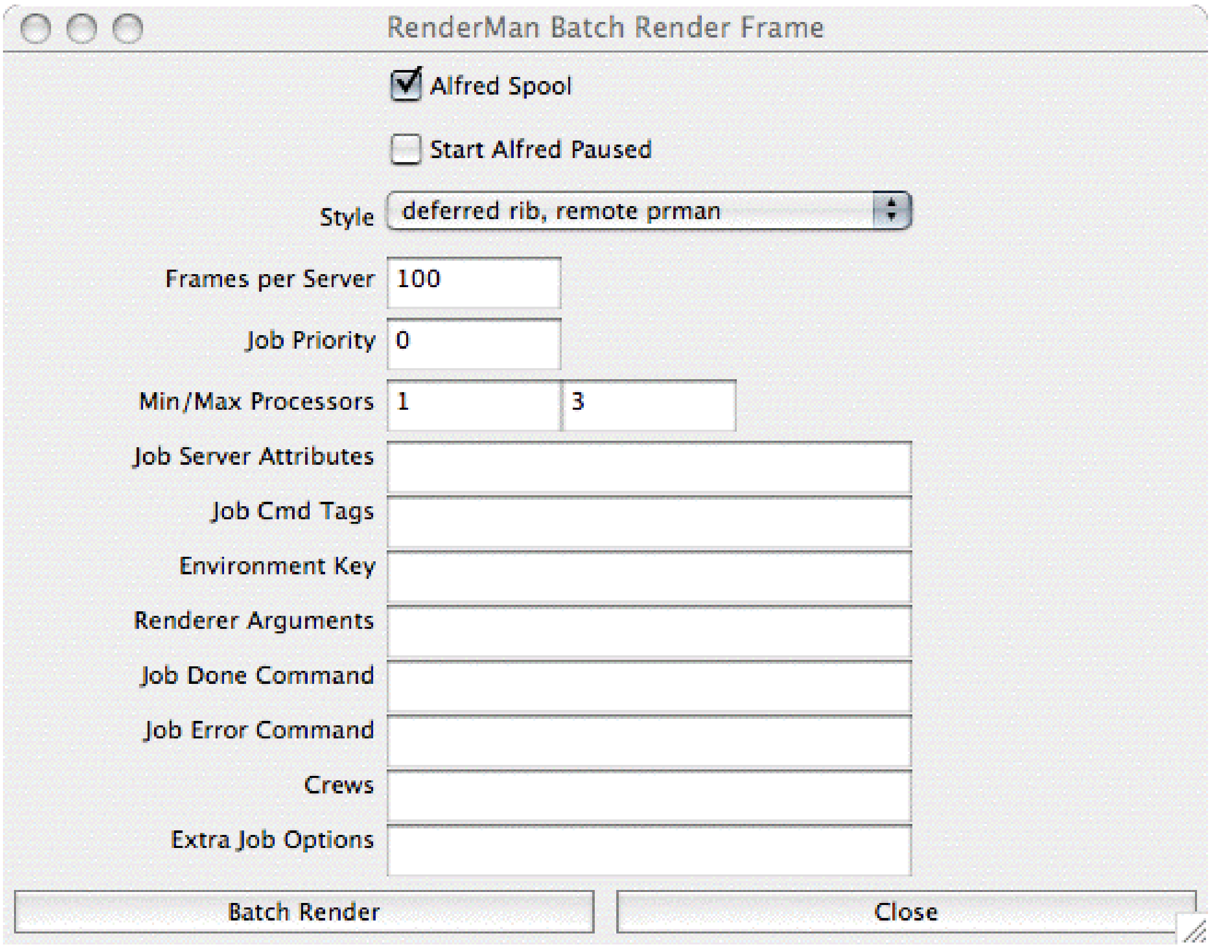
About Alfred

Alfred is a work distribution system that manages a hierarchy of parallel client applications connected to remote servers. The Alfred components are general purpose but they are especially well-suited for managing network-distributed rendering in the context of the RenderMan Studio.

In operation, Alfred traverses a **job** script called a *worklist*, which defines a tree-structured hierarchy of dependent **tasks.** Each task node in the tree contains **commands** that are executed on the local **client** host, possibly using associated remote **servers**. Multiple jobs may be queued up and performed in succession or in parallel. These terms are defined in more detail elsewhere.

Using Alfred with RenderMan Studio

Alfred is fundamentally unchanged in RenderMan Studio. It is, however, more tightly integrated with Maya. You can still run Alfred in a standalone mode, through the User Interface or via the command line, but when you select RenderMan as your renderer, you can also access a new Alfred window via Maya's *Batch Render* options. Note that this is a different way to access the Alfred options, if you've used the RenderMan Artist Tools; the options are no longer part of the Render Globals.


RenderMan Batch Render Frame

☑ Alfred Spool

☐ Start Alfred Paused

Style: deferred rib, remote prman

Frames per Server: 100

Job Priority: 0

Min/Max Processors: 1 3

Job Server Attributes:

Job Cmd Tags:

Environment Key:

Renderer Arguments:

Job Done Command:

Job Error Command:

Crews:

Extra Job Options:

The *RenderMan Batch Render Frame* window offers you with a simple interface for controlling Alfred's distribution of your renders. To enable Alfred to distribute your batch renders, click the *Alfred Spool* checkbox (if this is unchecked, the Maya batch mechanism is used, even if you are using RenderMan as your renderer).

You can choose the style of distributed render that you wish to use from the dropdown list. The options are *mayabatch local*, *mayabatch remote*, *immediate rib, netrender*, *deferred rib, remote prman*, and *remote rib, remote prman*. Please note that the *netrender* and *remote prman* options require RenderMan Pro Server installed on your target machines.

For more detailed information on the ins and out of Alfred, please see the Alfred documentation. For more information on RIB and the different methods of ribgen, please see the RfM Pro RIB overview.

Prev | Next

Prev | Next

Getting Started with it

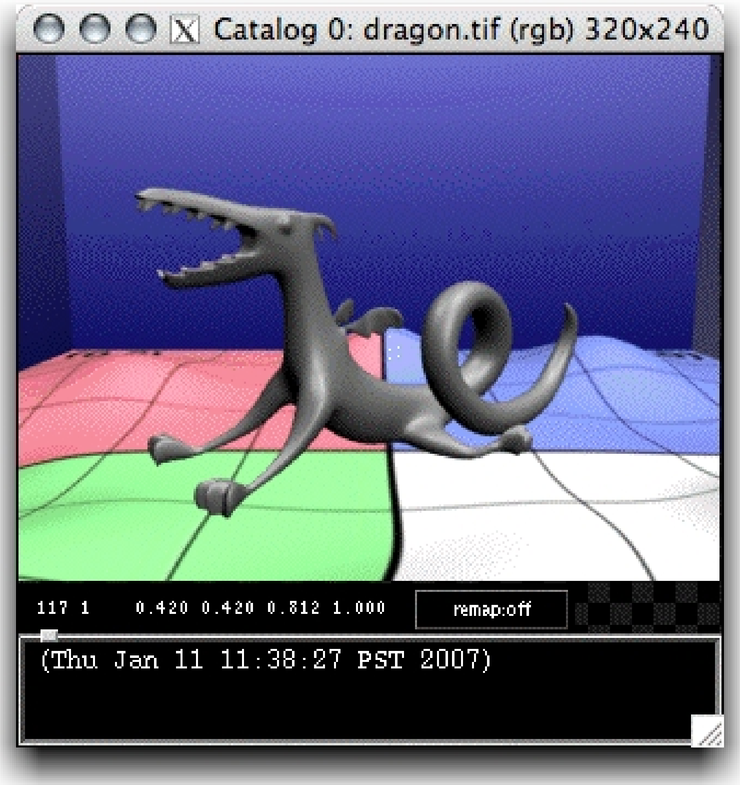
About it

it is a robust framebuffer/render view window, offering complete floating point support and a powerful and flexible catalog, as well as a fast and powerful imaging tool that is capable of production-qualilty image manipulation and compositing usually found only in high-end standalone products.

Using it with RenderMan Studio

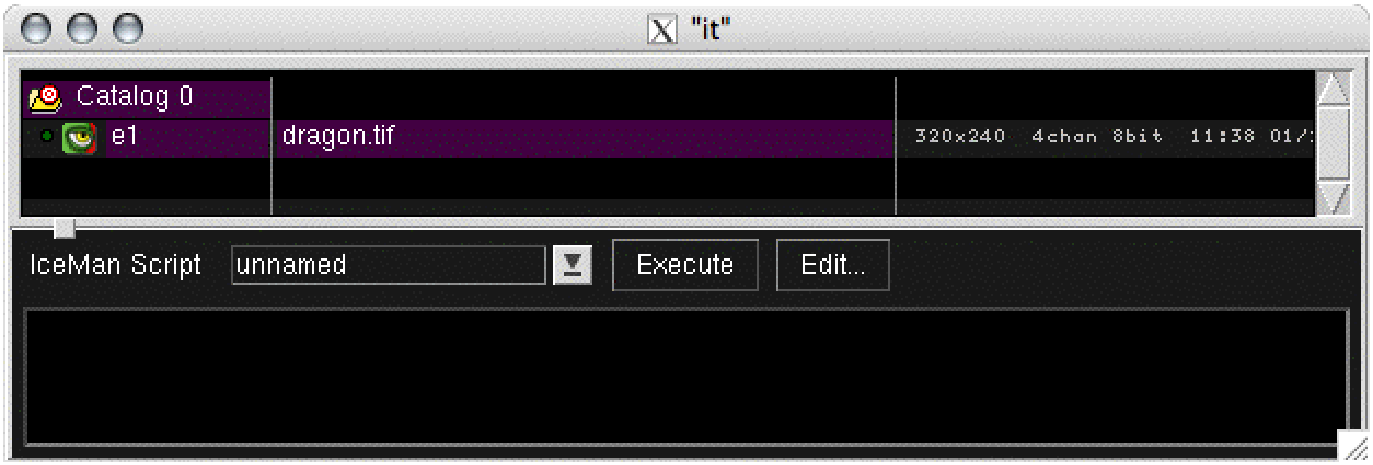
Unlike Alfred, it has been completely overhauled, from the ground up, for RenderMan Studio. it is more powerful, more stable, more useful, and more tightly integrated with Maya. You can still run it in a standalone mode, through the User Interface or via the command line, or you can select it as your preview render display from the *Render* menu. When RenderMan is selected as your renderer you can select the options menu for *Render Current Frame* and choose between internal renders to Maya's Render View or it , as well as external renders (via netrender) to it .

When rendering to it through Maya, your images are no longer viewed through Maya's Render View window; a new it window is opened. The it window can be customized by selecting view options via the it menu (right-click in the it window to access the menu). The status bar gives you progress information during renders and reports the position and rgba values relative to your pointer's placement upon your render's completion. In addition, there is a separate field for notes, which can be entered via the Common tab of the Render Settings.



```
Catalog 0: dragon.tif (rgb) 320x240
117 1     0.420 0.420 0.812 1.000    remap:off
(Thu Jan 11 11:38:27 PST 2007)
```

When using it via the Maya UI, right-clicking in the frame window provides the user with a basic menu that includes access to the it *hub*. The hub integrates the basic Catalog view with a script editor window for loading and executing IceMan scripts. Additionally, starting it in standalone mode opens the it hub.



For more detailed information on the long and short of it , please see the it documentation.

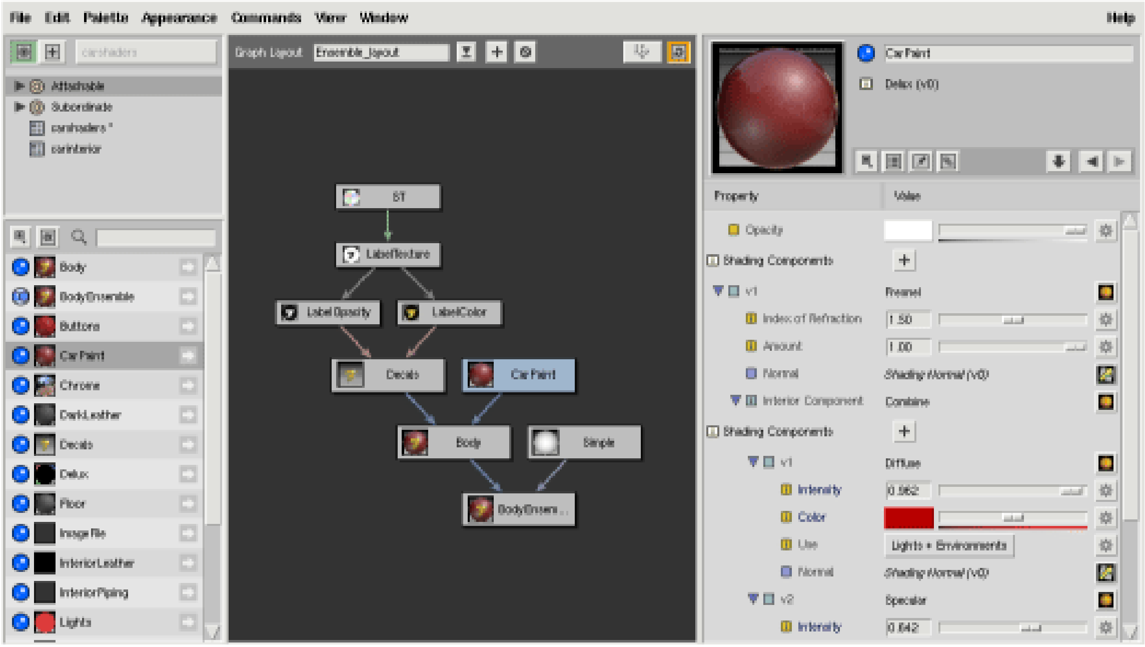
Prev | Next

Getting Started with Slim

About Slim

Slim is a powerful, extensible tool for creating, manipulating, using, and reusing RenderMan shaders, bringing the power of RenderMan shaders into the hands of artists. Users can construct RenderMan shaders by interactively combining modules into networks, editing their parameters, and previewing the results, all without writing any code.

Slim can be invoked in two different modes: *client* (via the Maya UI) or *standalone* (via the command line our your programs/applications menu). Whereas Slim is found on the RenderMan menu in the RenderMan Artist Tools, in RenderMan Studio you access Slim via Maya's *Window* menu; go to **Window** -> **Rendering Editors** -> **RenderMan** -> **Slim...** to open the completely redesigned Slim UI.



Slim's basic workflow is quite simple:

1. From the **File** menu, choose *New Palette* to start from scratch, or *Open Palette* to select an existing Slim palette.
2. Click on the + icon to switch from the *Browse* to the *Create* mode.
3. Choose an appearance from the Palette View, an icon representing this primary node for your Slim network will be displayed in the Network View, and the appearance and its parameters will show up in the Appearance View.
4. Tweak your appearance as you see fit. **Note that by setting parameters to *External* you will be able to continue tweaking after adding the appearance to your Maya scene.** Click on the swatch to compile the shader.
5. To add your Slim appearance to your Maya scene, go to the **Appearance** menu and select *Add To Scene*. You can then attach the appearance to an object in Maya from Maya's **Lighting/Shading** menu; it has been added to your existing materials , so attaching it is a matter of selecting your object and choosing if from the *Assign Existing Material* list.

For a complete rundown of Slim, please consult the Slim documentation.

Prev | Next

Pixar Animation Studios