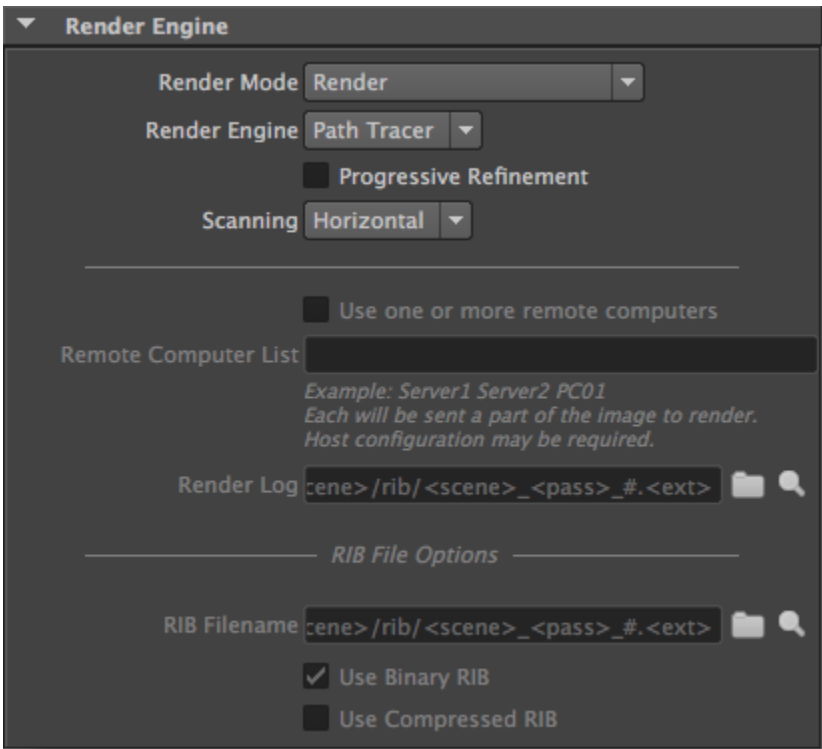


# Render Engine

## Overview

The *Render Engine* group of attributes contains controls for the core aspects of the rendering process, such as which render engine to use and the type of output produced in the process (images or RIB files).



The Render Engine group

### Render Mode

Controls the type of output to be produced when rendering. The available options are:

Option	Description
Render	Renders the layers of images as specified by the <a href="#">Output: Image Layers (AOVs)</a> group of rendering attributes.
Export RIB File Only	Instead of rendering the images, the renderer exports everything to one or more RIB files suitable for rendering the images at a later time.
Export RIB Archive Only	The scene will be exported to one or more RIB archives that can be read later by a <i>RIB Archive Node</i> in this scene or another one. Advanced options can be specified in the <a href="#">RIB Archive</a> group of rendering attributes. (Refer to <a href="#">RIB Archive Guidelines</a> for examples on how to use this option).

Export RIB File and Render*	This mode exports everything to a RIB file and then invoke the standalone <code>'renderdl'</code> program to render it in a background process. This means that once the RIB file is saved, the rendering is done using a completely separate and independent process from <i>Maya</i> . The log output of the renderer is saved into a log file named after the RIB file with a <code>'.log'</code> extension. The images outputted by <code>'renderdl'</code> are the same as if selecting <i>Render</i> for the <i>Render Mode</i> (i.e. the images are outputted as specified by the <i>Displays</i> group of attributes). Note that in this mode, those images can not be outputted to the <i>Maya Render View</i> .
*Option offered as part of the <i>3Delight Studio Pro</i> package.	

### Render Engine

Specifies the rendering engine to use for rendering. The available options are the 'Path Tracer' (default) for the ray tracing based algorithm and the 'REYES' based algorithm and. A discussion of the Pros and Cons of each is presented in the last section of this page.

### Progressive Refinement

This option is only available when using the 'Path Tracer' *Render Engine*. When this toggle is turned on, the rendering rapidly shows a coarse image and then refines it progressively until completed. This option is effective only for images outputted directly to a window on screen (i.e. to the *Maya Render View* or to *3Delight i-Display*). During batch rendering (i.e. when using *Maya Batch*) this option is ignored. Note that this is not required when rendering in *Maya IPR*, which is progressive by nature and support dynamic edits of materials, lights and cameras.



When using *Progressive Refinement*, the *Filter Type* and *Filter Width* attributes (specified in the *Quality* group of rendering attributes) are ignored. A Box filter with a width of 1 is used instead. Because of this, your image rendered with this option may differ slightly from your final render using *Maya Batch*.

### Scanning

An image is rendered progressively in small groups of pixels called buckets. This option allows you to specify in what order the buckets are rendered. Rendering will be more memory-efficient if a bucket order that minimizes the number of objects being rendered simultaneously is chosen. The available options are:

Option	Description
Horizontal	The buckets are rendered row by row, from left to right and from top to bottom. This is the default.
Vertical	The buckets are rendered column by column, from top to bottom, and from left to right.
Zigzag	The buckets are rendered row by row, from left to right on even rows, and from right to left on odd rows.
Spiral	The buckets are rendered in a clockwise spiral starting at the centre of the image.
Circle	The buckets are rendered in concentric circles starting at the centre of the image.

The following attributes are only available when *Render Mode* is set to *Export RIB file and Render* (as they can work only when the rendering is done through the standalone rendering program `'renderdl'`).

#### Use one or more remote computers\*

Turning this toggle on enables distributed rendering (i.e. to use one or more remote computers to render a single image). Each computer listed in *Remote Computer List* will be rendering a part of the image. This toggle is off by default.

\*Option offered as part of the *3Delight Studio Pro* package.

#### Remote Computer List

Specifies the list of computers to use for distributed rendering (list of their names separated by a space). Each computer will need to be configured to allow remote execution of a program using SSH or RSH (i.e. allowing password-less SSH or RSH). This control only applies when *Use one or more remote computers* is enabled.

#### Render Log

Specifies the file to receive the various messages outputted during the execution of the standalone rendering program `'renderdl'`. Clicking on the view button will open the log file using the application designated in the *3Delight Preferences* dialog.

## RIB File Options

The following attributes are only available when *Render Mode* is set to *Export RIB File Only*, *Export RIB Archive Only* or *Export RIB File and Render*.

#### *RIB Filename*

Specifies the name of the RIB file to save. If this attribute is empty when the *Render Mode* is set to export to a RIB file, a decent default name is automatically generated. If the *Animation* attribute is set to 'on' (see the [Output: Frame Range](#) group of attributes) and the *RIB Filename* remains the same for a series of frames, the RIB file will contain the data for every frame in the frame range. To save into separate files for each frame, the '#' frame number token must be inserted in the *RIB Filename*. See [File Path Expressions](#) for details on how it is possible to construct relative paths and / or paths containing dynamically expanded tokens.

#### *Use Binary RIB*

When this options is on, the RIB file will be saved in binary format, producing a smaller file. Turn this option off to produce a text RIB file, which is bigger but easier to read and edit. By default, this option is on.



Binary RIB files are faster to output, faster to render and use less disk space.

#### *Use Compressed RIB*

This option controls if the RIB file will be compressed, producing a smaller file. By default this option is off.

## Pros and Cons of vs Path Tracing vs REYES

### Path Tracing

Pros:

- Efficient at rendering densely tessellated geometry and dense scenes (forests, crowds, etc).
- Massive instancing allows for memory efficient rendering of trees/crowds and other redundant scene elements.
- Shading is usually sharper because shading is performed at each sub-sample (and not per pixel as in REYES). Although it is possible to make REYES shaper by increasing the *Shading Rate*, it is usually avoided (as it affects performance). Whereas the Path Tracer do not have options to reduce the shading rate.
- Scales better than REYES with increased number of cores.

Cons:

- Slower with displacements.
- Need more samples to render smooth (noise free) motion blur and depth of field. This happens because it is actually more precise than REYES and there is more detail in the motion blurred effect, but this also induces more noise.
- Increasing pixel samples (for aliasing and to reduce noise) has a direct impact on performance (though does not affect sampling of indirect illumination).

### REYES

Pros:

- Extremely efficient rendition curved surfaces of average to large size; that is surfaces covering more than a few pixels on the image.
- High quality motion blur and depth of field are extremely fast. This is because the shading calculation is decoupled from hiding calculation.
- Displacements are rendered at a lesser cost than in path tracing (micro-polygons).
- Efficient at rendering fluids because of screen space under-sampling.
- Efficient rendering of millions of particles.
- Performance almost independent of oversampling (pixel samples). This makes it easy (and fast) to render images without noise and without aliasing.

Cons:

- Looses performance when rendering densely tessellated geometry (i.e. objects with a density of hundreds or thousands of surfaces covering just a few pixels).
- Not suited for rendering scenes with high "pixel complexity" (e.g. a crowd seen from afar).
- Motion blur shading is an approximation. For example, a spinning wheel will have its specular highlight blurred along with other details on the wheel, whereas the highlight should remain sharp.
- Takes more memory when used alongside ray tracing (for GI for example). This happens because we need ray tracing structures alongside REYES structures.