

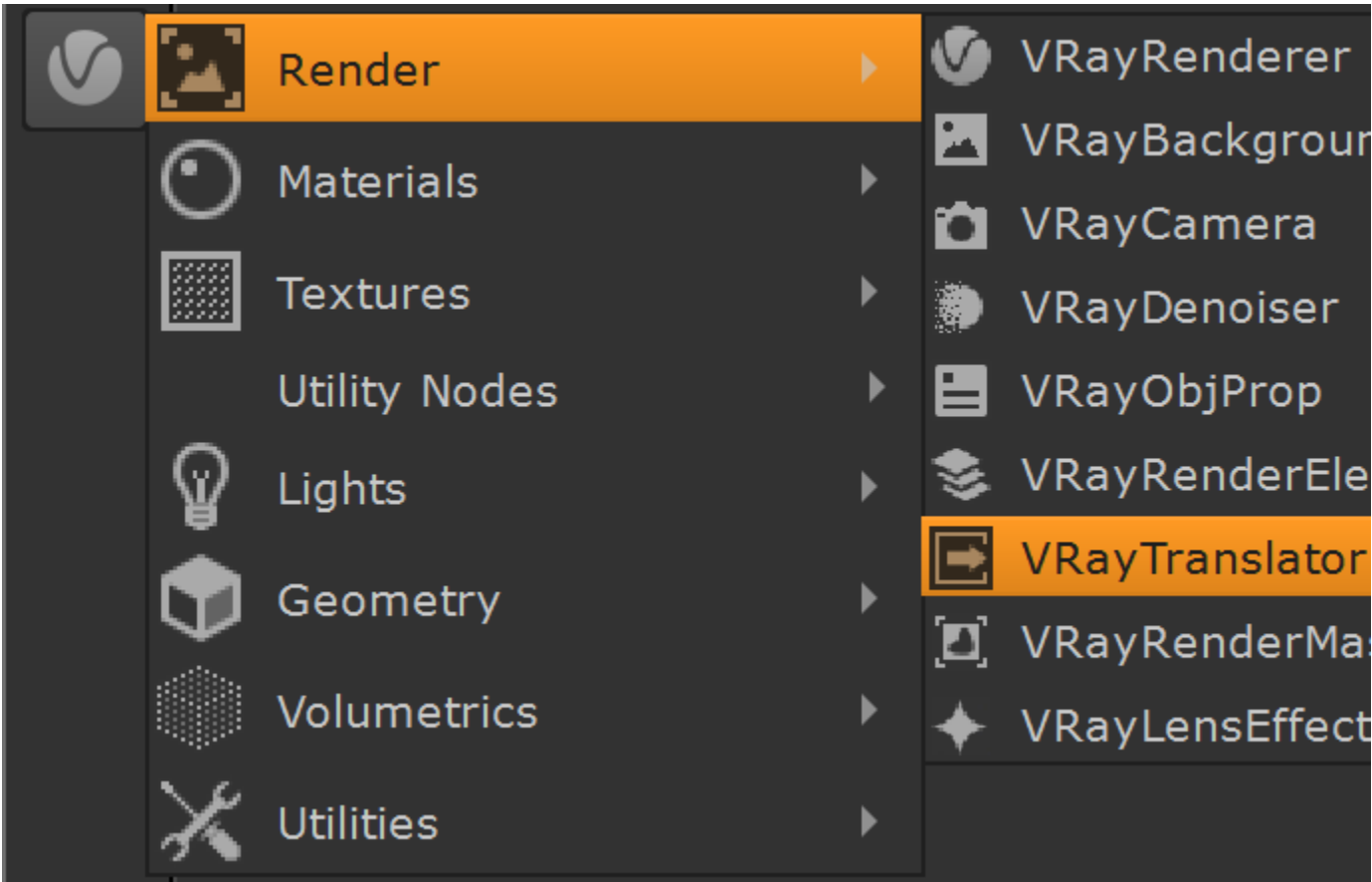
Scene Translator | V-RayTranslator

This page provides information on the V-RayTranslator node.

Overview

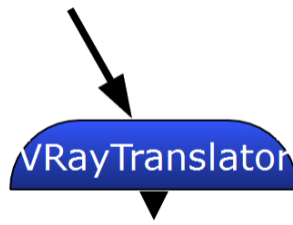
The V-RayTranslator node is an advanced Exporter for single frame or animated `.vrscene` files. V-Ray Standalone integration enables rendering the scene in V-Ray Standalone within Nuke using Progressive rendering, RT, and Distributed rendering.

UI Path: ||Toolbar|| > **V-Ray menu icon** > **Render** > **V-RayTranslator**

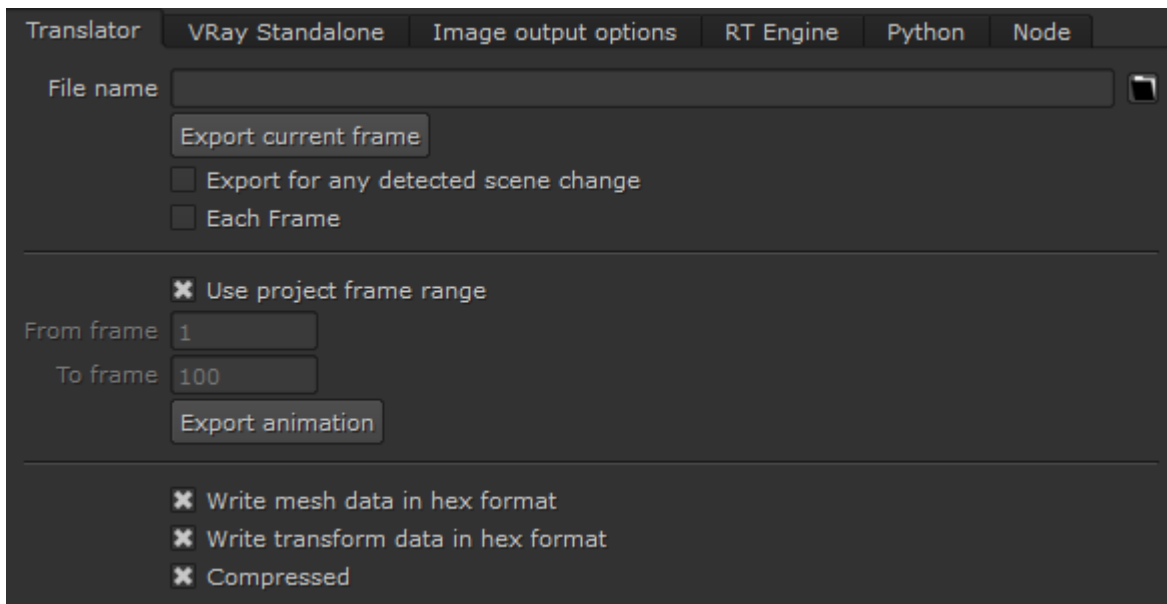


Inputs

V-RayTranslator requires the [V-RayRenderer](#) node as an input.



Translator Parameters



File name – Sets the name and location for the VRScene file export.

Export current frame – Exports the current frame to the specified VRScene file.

Export for any detected scene change – When enabled and the VRTranslator is connected on the path of the viewer or writer node, V-Ray will export the scene to a *.vrscene* file if any change in the Nuke scene is detected.

Each Frame – Forces export of every frame.

Use project frame range – When enabled, the project's frame range will be used as the default exported frame range. If disabled, the exported range can be manually defined using **From frame** and **To frame**.

From frame – Sets the first frame of the animation range to be exported.

To frame – Sets the final frame of the animation range to be exported.

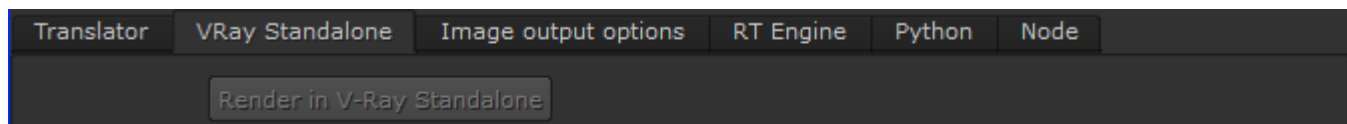
Export animation – Exports the given frame range into a single VRScene file.

Write mesh data in hex format – When enabled (by default), the mesh data will be written in hexadecimal format in the *.vrscene* file. This makes the files smaller and faster for processing, but harder for persons to read. When this option is disabled, the mesh data will be written in normal decimal format as text.

Write transform data in hex format – When enabled (by default), the transformation data will be written in hexadecimal format in the *.vrscene* file. This makes the files smaller and faster for processing, but harder for people to read them.

Compressed – When enabled, a basic compression will be applied to the *.vrscene* file.

VRay Standalone Parameters



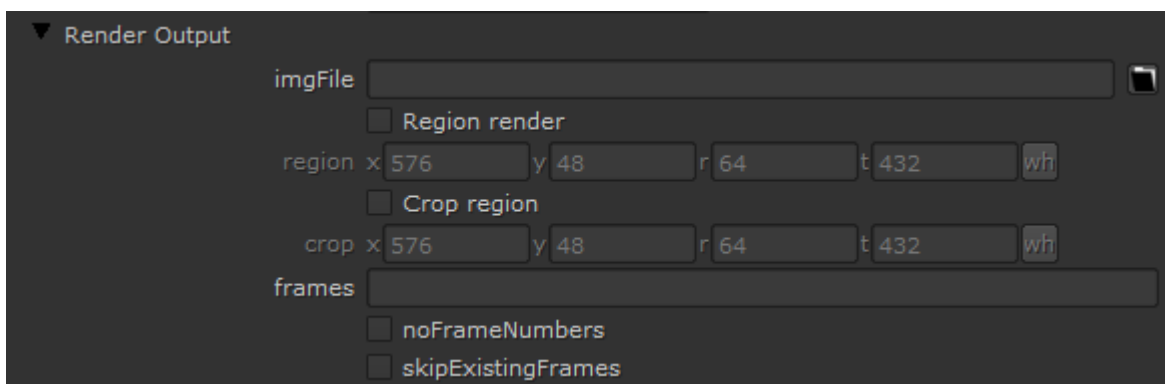
Render in V-Ray Standalone – Invokes the `VRayStandalone` to render the `VRScene` file from the `VRayTranslator` using all current settings.



This option is automatically disabled under two conditions:

1. The `VRayStandalone` executable was not found at the default location and/or the manually specified location under the others group is invalid.
2. The input `VRScene` file from the Translator tab is either invalid or does not exist.

Render Output



imgFile – Sets the base output image file name and location.

Region render – Only exports a region of the workspace. This region is defined either by the **region** parameter or by a locator in the viewport.

region – When **Region render** is enabled, defines the bounds of the region to render.

Crop region – Crops out a piece of the defined region.

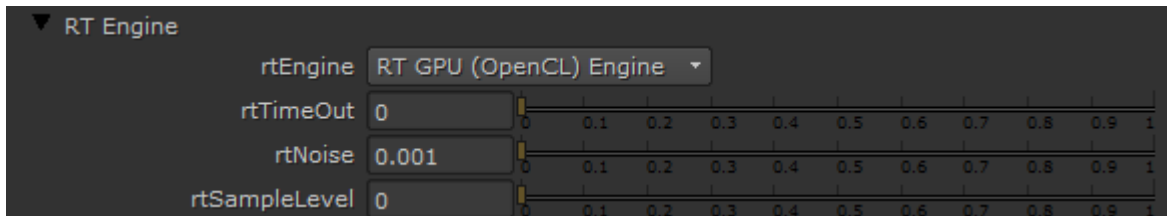
crop – When **Crop region** is enabled, defines the bounds of the region to render.

frames – Determines which frames to render.

noFrameNumbers – Disables the **automatic appending of** frame numbers to the names of the rendered image files.

skipExistingFrames – Enabling this option causes the renderer to skip frames currently in the output folder.

RT Engine



rtEngine – Specifies which rendering engine to use:

No RT Engine – Bucket render engine.

RT CPU Engine – CPU RT engine.

RT GPU (OpenCL) Engine – GPU RT engine running on OpenCL.

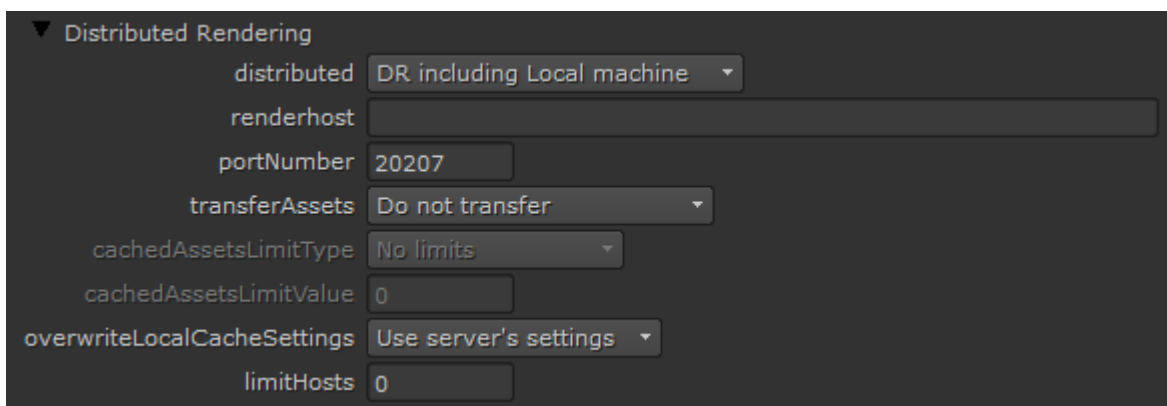
RT GPU (CUDA) Engine – GPU RT engine running on CUDA.

rtTimeout – Specifies a time limit when using the RT engine, expressed in minutes as a floating-point value. The default is 0.0, which indicates no time limit.

rtNoise – Specifies the noise threshold for a frame when using the RT engine, expressed as a floating-point value. The default is 0.001.

rtSampleLevel – Specifies the maximum number of paths per pixel for a frame when using the RT engine. The default is 0, which indicates no limit.

Distributed Rendering



distributed – Specifies whether and how to use distributed rendering:

No Distributed Rendering – Distributed rendering is disabled.

DR including local machine – Distributed rendering is enabled and the local machine is included in the rendering.

DR excluding Local machine – Distributed rendering is enabled and the local machine is excluded from the rendering.

renderhost – A list of hosts to use for distributed rendering. A host can be specified by either computer name or IP. Separate multiple hosts by semicolons. Example: 192.168.10.21; 192.168.10.22; 192.168.10.23

portNumber – Specifies the port number to use for distributed rendering. The port numbers of the render servers and the render client must match for DR to work.

transferAssets – Specifies whether missing assets should be transferred to the DR servers:

Do not transfer – Transfer disabled.

transfer missing assets – All missing assets will be transferred.

Check asset cache folder – Check for assets in the asset cache folder before transferring them.

Note: The default system temp folder, usually `C:\Users\<username>\AppData\Local\Temp` on Windows (%TEMP%) or `/tmp` on Linux (\$TMPDIR), is used for copying the assets if the `VRAY_ASSETS_CACHE_PATH` [environment variable](#) is not set.

cachedAssetsLimitType – Puts a limitation on the size and age of transferred assets.

No Limits – All assets will be transferred.

By age (in hours) – Only assets newer than the number of hours specified in the **cachedAssetsLimitValue** will be transferred.

By size (in GBs) – Only assets smaller than the number of GBs specified in the **cachedAssetsLimitValue** will be transferred.

cachedAssetsLimitValue – Sets the value for **cachedAssetsLimitType**.

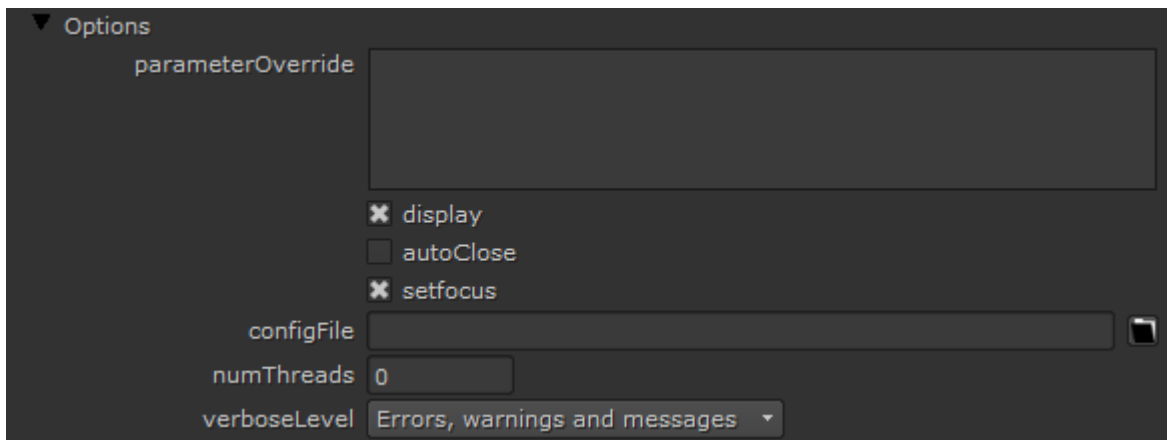
overwriteLocalCacheSettings – Selects whether to use Server or Client's settings for the Local Cache:

Use server's settings – Uses the server's settings for the Local Cache (default).

Use client's settings – Overrides the Local cache settings with Local client's settings.

limitHosts – The maximum number of hosts for distributed rendering. A value of 0 indicates no limit.

Options



parameterOverride – Enables changing values for plugin parameters without editing the VRscene file.

Parameters are controlled by "**<plugin-parameter-id>=<value>**"

<plugin-parameter-id> - Can be one of two formats:

PluginType::parameterName
pluginName.parameterName.

<value> - Format depends on the type of the parameter you want to override. Supported types are bool, int, float, color, acolor, texture and float texture.

For bool, int, and float, a simple number is expected.

For color - Color(1, 0.0, 1.0) is expected. For acolor - AColor(1.0, 0, 0, 0.3).

For texture, the format should be the same as for a color or the name of a plugin to connect to the parameter.

For float texture parameters either one float or a name of a plugin is expected.

More than one -parameterOverride options can be specified and they will be executed in the order they are specified on the command line.

Examples:

```
parameterOverride="SettingsImageSampler::type=3"  
parameterOverride="TexFresnel::fresnel_ior=1.3"  
parameterOverride="VRayMtl1@diffuse.color=Color(1.0,0.2,1.0)"  
parameterOverride="VRayMtl2@diffuse.color_tex=checker1"
```

display – Shows the rendered image in the V-Ray frame buffer.

autoClose – Automatically closes the displayed image when rendering is complete.

setfocus – Forces the onscreen focus to the VFB window if displayed. This works reliably only in Windows.

configFile – The path and file name for the V-Ray config file, which contains paths to the V-Ray plugins. Note that additional paths for V-Ray plugins can also be specified with the VRAY_PLUGINS_x86 and VRAY_PLUGINS_x64 [environment variables](#). The default is *vrayconfig.xml* in the same folder as *vray.exe*.

numThreads – Sets the number of rendering threads. The default is 0, which automatically calculates and chooses the most optimum number of threads.

verboseLevel – Specifies the verbose level of information printed to the standard output:

None – No information is printed.

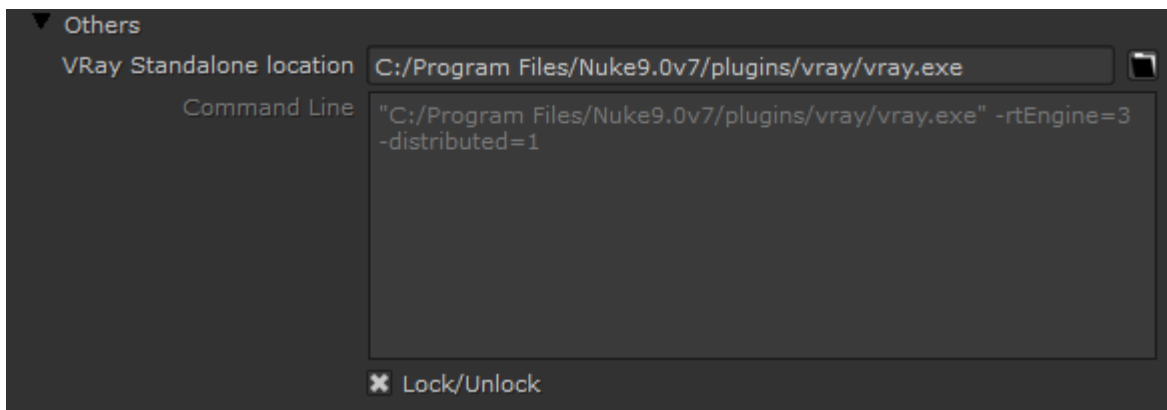
Only errors – Only errors are printed.

Errors and warnings – Errors and warnings are printed.

Errors, warnings and messages. – All errors, warnings, and messages are printed.

All – All output is printed.

Others



VRay Standalone location – Specifies the location of the standalone application *vray.exe*.

command Line – The parameters in this tab automatically create the options in the command line. These can be manually entered by unlocking the command line with the **Lock/Unlock** parameter. See the [V-Ray Standalone Command Line Options](#) page for variables.

Lock/Unlock – Enables locking and unlocking of the **Command Line** parameter.

Image Output Options Parameters

Image file output

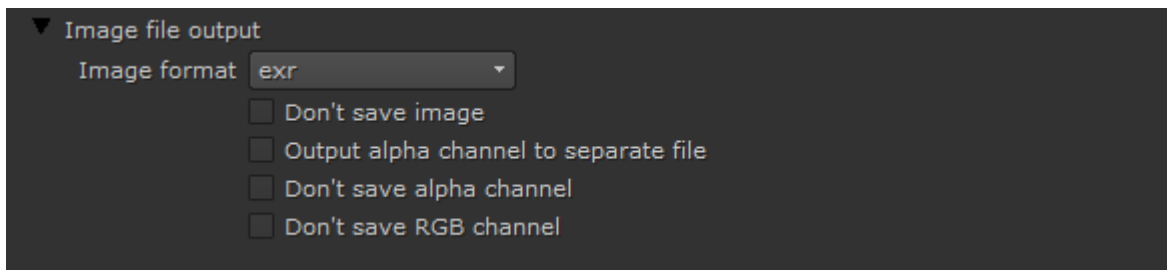


Image format – Specifies the file format in which the image(s) will be saved.

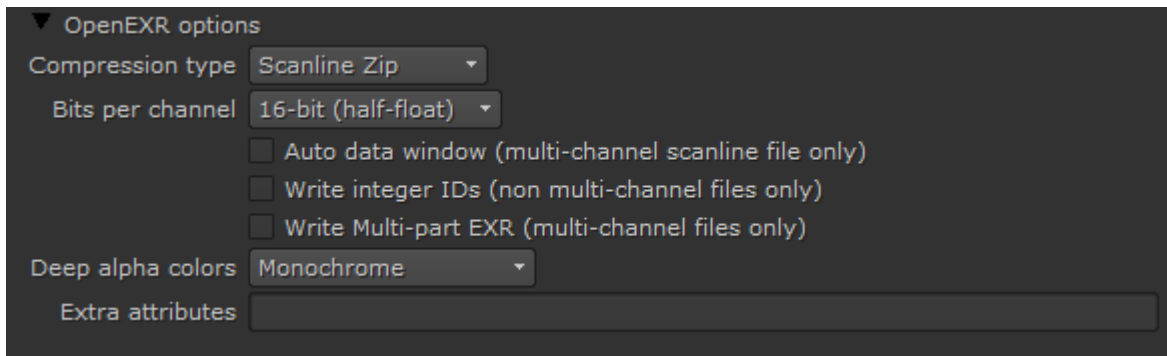
Don't save image – When enabled, the final image will not be saved automatically. This is ignored during batch rendering.

Output alpha channel to separate file – When enabled, the alpha channel of the rendered image will be saved to a separate file.

Don't save alpha channel – When enabled, the Alpha channels will not be saved (ignored with multichannel exr and vrimg).

Don't save RGB channel – When enabled, the RGB channels will not be saved (ignored with multichannel exr and vrimg).

OpenEXR options



Compression type – Select the compression type for the output file:

No compression – Disables all compression.

Run-length – Basic form of compression comparable to the one used by Targa/BMP files.

Scanline Zip – Zip style compression applied to individual scanlines.

Block Zip – Zip style compression applied to blocks of 16 scanlines at a time. More effective than **Scanline Zip** but can be slower to decompress.

PIZ – Uses a new combined wavelet / Huffman compression. This form of compression is quite effective when dealing with grainy images, and will often surpass any of the other options under grainy conditions.

PXR24 – Converts data to 24-bit then uses Zip compression. It is lossless for half and 32-bit integer data and slightly lossy for 32-bit float data.

B44 – This form of compression is lossy for half data, and stores 32-bit data uncompressed.

B44A – An extension to B44 where areas of flat color are further compressed. Regular B44 compresses uniformly regardless of image content.

DWAA – JPEG-like lossy compression format. Compresses 32 scan lines together.

DWAB – Same as DWAA, but compresses blocks of 256 scanlines to achieve a better compression ratio.

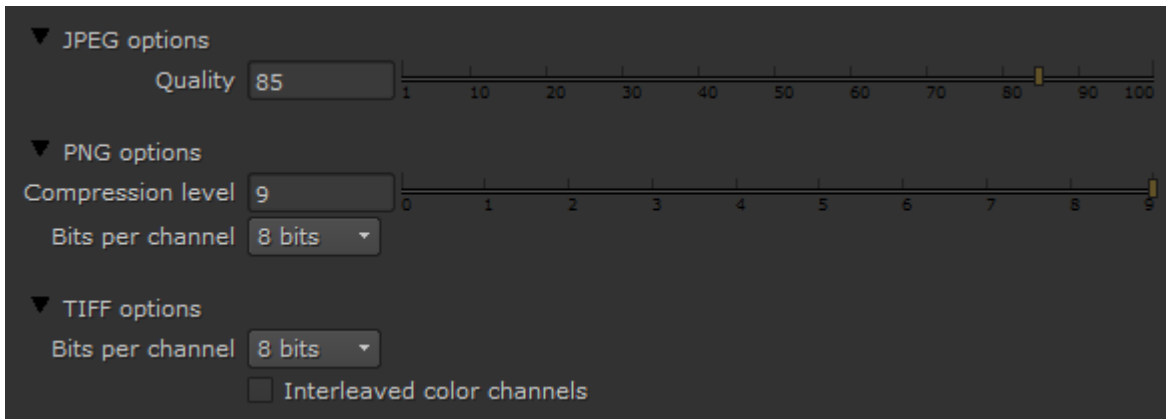
Bits per channel – Choose between 16-bit (half float) and 32-bit (full float) precision.

Auto data window (multi-channel scanline file only) – When enabled, V-Ray will automatically try to find the valid data window for the image based on the alpha channel. This option works only for multi-channel OpenEXR files when the Memory frame buffer option of V-Ray is set to Full.

Write integer IDs (non multi-channel files only) – Writes ID information in integer form rather than RGB colors.

Write Multi-part EXR (Multi-channel files only) – Multi-channel EXRs will be written to a single multi-part file. This feature requires OpenEXR 2 support.

File type options



JPEG options

Quality – Specifies the desired JPEG output quality as a percentage. Lower values represent smaller file size and worse quality, while higher values represent larger file size and better image quality.

PNG options

Compression level – Specifies the compression level for the resulting file. Lower values mean less compression and larger file size. Higher values mean better compression and smaller file sizes. Note that PNG files use a lossless compression, so the resulting image quality is not affected.

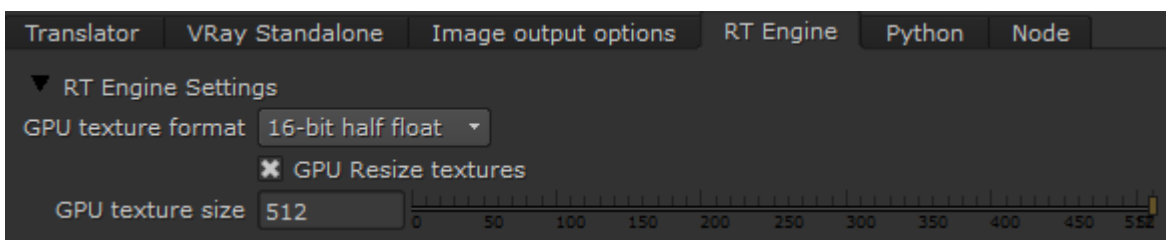
Bits per channel – The number of bits for each of the PNG channels.

TIFF options

Bits per channel – Specifies the desired number of bits per channel (8, 16 or 32) in the resulting image.

Interleaved color channels – When enabled, the color information for each pixel will be saved using the interleaved order, which means that the full color for each pixel will be saved before moving to the next one. Some image processing applications require that the image be saved in this mode.

RT Engine



GPU texture format – Controls the file format of the material textures used in GPU rendering. Choose between 8-bit, 16-bit and 32-bit. This does not affect textures used for lights and displacement.

GPU Resize textures – When enabled, resizes high-resolution textures to a smaller resolution to optimize memory usage. This parameter is only effective when the **rtEngine** option uses Open CL or CUDA.

GPU texture size – When **GPU Resize textures** is enabled, this value specifies the resolution to which the textures will be resized.