

V-Ray Profiler

This page provides information about the V-Ray Profiler tool.

Overview

V-Ray Profiler helps determine what parts of the scene are taking the longest time to render, compare the performance of different material settings and figure out the best settings for your scene. It does that by tracking the time spent in sampling shaders (materials and textures) and volumes during rendering and breaking down the different types of calculations done for each shader, such as GI, Reflections, Refractions, etc. It writes a JSON formatted report and an HTML report. Both can be read by a browser tracking software, the .json file can also be opened in a text editor offline.

The V-Ray Profiler report breaks down the tracked time into a few categories: Initialization, GI, Reflection, Refraction, and Others. These categories are only tracked if they were used by the shader. For example, a V-Ray Material with no refraction enabled does not report anything for Refraction. The same happens if a shader with no Reflection layer is used.

For volumes, the Profiler reports the time spent calculating each volume in your scene.

V-Ray Profiler is partially supported on V-Ray GPU. The **Full** Profiler mode is not supported on V-Ray GPU.

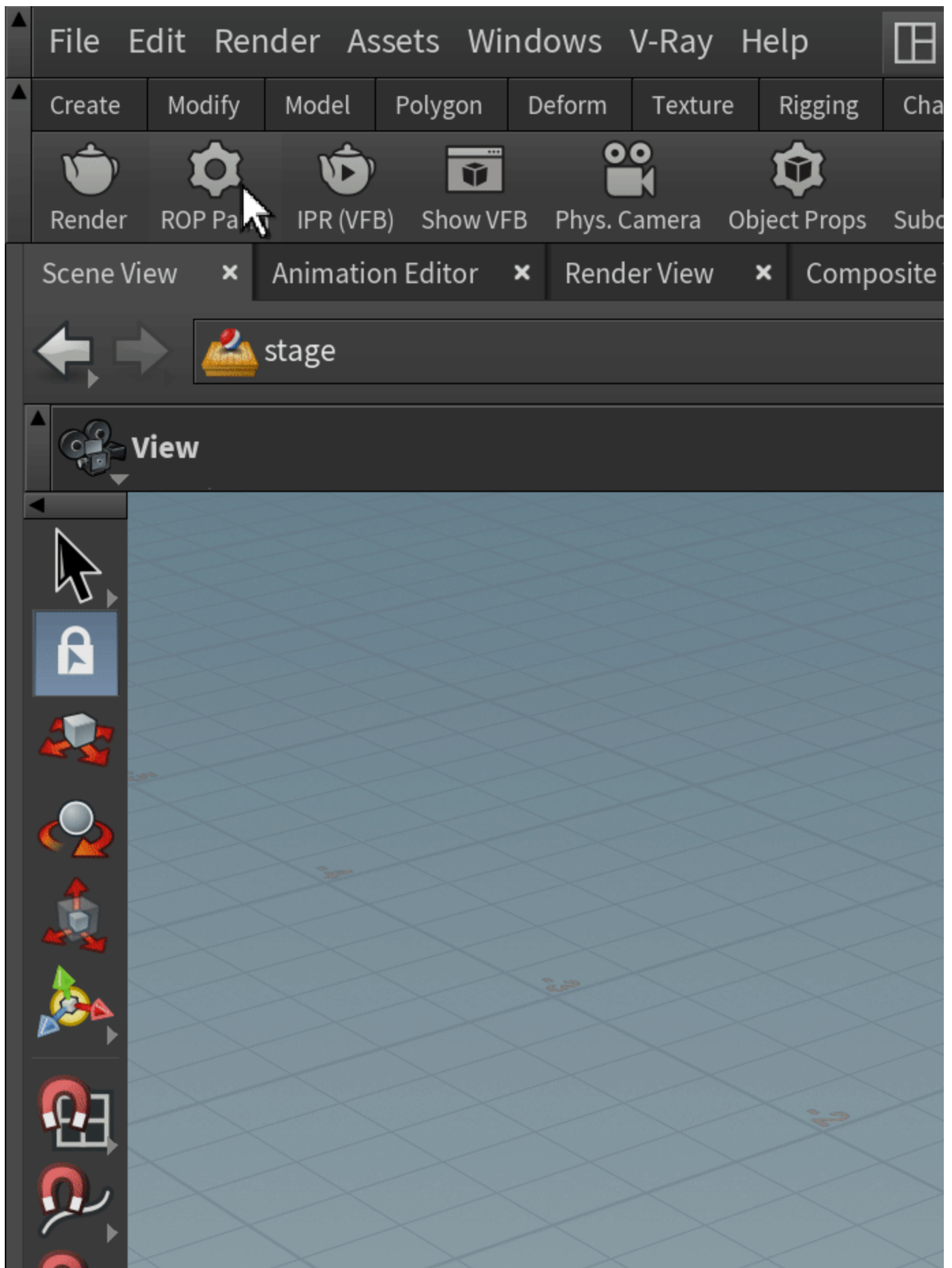
UI Paths:

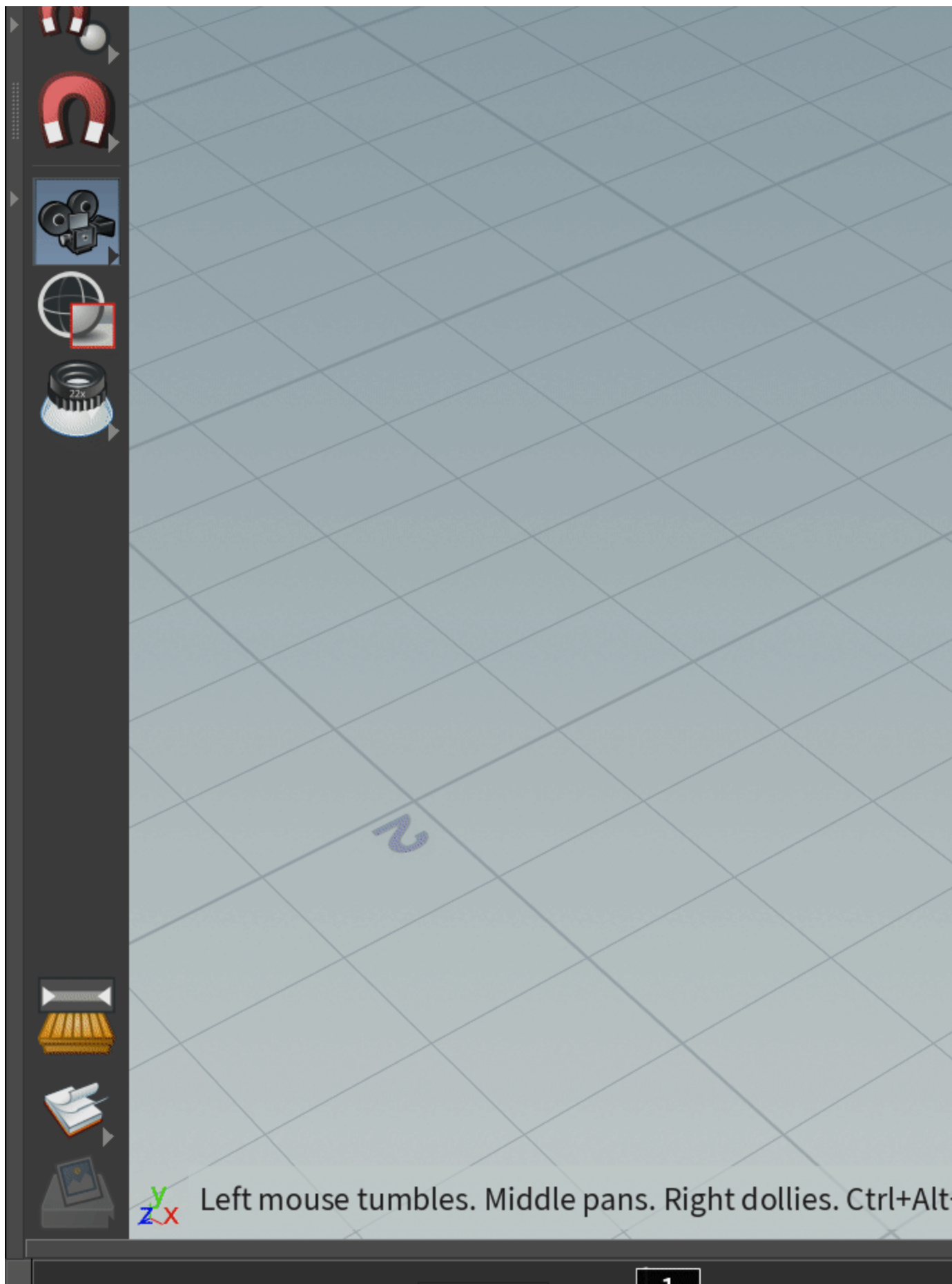
||V-Ray Shelf|| > **ROP Parm** > **Renderer** tab > **Options** tab > **Rendering** tab > **V-Ray Profiler**

||out Network|| > **V-Ray** > **V-Ray Renderer** > **Renderer** tab > **Options** tab > **Rendering** tab > **V-Ray Profiler**

V-Ray menu > **Render Settings** > **Renderer** tab > **Options** tab > **Rendering** tab > **V-Ray Profiler**

||stage Network|| > **Rendering** > **Render Settings** > **V-Ray** tab > **Options** tab > **Rendering** tab > **V-Ray Profiler**







Settings

Mode – Determines the mode in which V-Ray Profiler works.

Off – V-Ray Profiler is disabled.

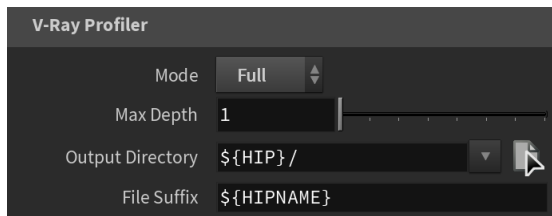
Simple – Reports the time spent on system processes such as compiling geometries, building the Light Cache, exporting the scene from Houdini to V-Ray etc. Suitable for most scenes.

Full – Provides an in-depth report. It includes detailed information such as time of calculation per shader e.g. how much time is spent calculating the reflection of a shader, as well as what shaders are reflected in it and how long it took sampling them. This mode is suitable for scenes that include views through glass window or in a mirror reflection. This mode can slow down the render by up to 20%.

Max Depth¹ – Determines the amount of bounces the rays make after coming into contact with an object. Keep in mind that this option depends on the materials' Reflection and Refraction Max depth parameters. If the GI is set to Brute Force, its Depth parameter influences the value of the V-Ray Profiler's Max depth. This is reflected in the resulting profile.

Output Directory – Specifies the location and name of the output files.

File Suffix – Specifies a suffix that is appended at the end of the profiler report files.



Workflow

- Turn on the V-Ray Profiler by changing the Profiler **Mode** to *Simple* or *Full*.
- Set the output location of the *.json* and *.html* files. If this field is left empty, the generated files are saved in the default Temp folder in your system.
- Render your scene². Once you finish the render, you are able to find the *.html* and *.json* files in your designated directory. Rendering a second or third time generates more reports. Up to 3 reports can be generated in the same folder. Generating more, without changing the **Output directory**, causes the oldest report to be replaced by the new one.

You can check the output file location in the log file.


- Open the generated report³.

To open the .json file:

Go to a third-party tracking software. We recommend using Google Chrome's tracer. Open Google Chrome and enter **chrome://tracing/** in the URL field. Drag and drop your *.json* report onto Google Chrome and explore. The report shows the duration it took to render each element in nanoseconds. Alternatively, you can open the *.json* file in any text editing software on your machine.

If you want to open the .html file:

Double-click on the file and it opens the report in the default browser. Most browsers are suitable for this. Explore the report. The report shows the duration it took to render each element in nanoseconds.

- You can view the report's metadata either through the  button in the browser report, or at the top of the .json file, when read through a text editor, under "product".

If, at any point during the render, the CPU performance drops, a **High bundle size wait** section appears in the profiler. You can explore the stats associated with it.

▼ Process 1	
▼ render	

▼ Process 0	
▼ process	
▼ Process 1	
▼ render	

Example of a result in All mode opened in a browser

▼ Process 0					
▼ process	<table><tr><td>sequence render</td><td>t...</td></tr><tr><td>frame render</td><td>t...</td></tr></table>	sequence render	t...	frame render	t...
sequence render	t...				
frame render	t...				

Example of a result in System mode opened in a browser

```

1  {
2    "product": {
3      "name": "Houdini",
4      "version": "19.5.403",
5      "V-Ray core version": "6.10.07",
6      "hash": "3bffa0",
7      "build date": "May 2 2023"
8    },
9    "traceEvents": [
10     { "args": { "name": "process", "cat": "__metadata", "name": "thread_name", "ph": "M", "pid": 0, "tid": 0, "ts": 0 },
11     { "args": { "name": "render", "cat": "__metadata", "name": "thread_name", "ph": "M", "pid": 1, "tid": 0, "ts": 0 },
12     { "cat": "vray", "name": "frame begin", "ph": "B", "pid": 0, "tid": 0, "ts": 330661 },
13     { "cat": "vray", "name": "plugin initialization", "ph": "B", "pid": 0, "tid": 0, "ts": 331019 },
14     { "cat": "vray", "name": "plugin initialization", "ph": "E", "pid": 0, "tid": 0, "ts": 331226 },
15     { "cat": "vray", "name": "geometry compilation", "ph": "B", "pid": 0, "tid": 0, "ts": 331243 },
16     { "cat": "vray", "name": "|obj|box_object1|node", "ph": "B", "pid": 0, "tid": 0, "ts": 331267 },
17     { "cat": "vray", "name": "|obj|box_object1|node", "ph": "E", "pid": 0, "tid": 0, "ts": 331519 },
18     { "cat": "vray", "name": "|obj|sphere_object1|node", "ph": "B", "pid": 0, "tid": 0, "ts": 331523 },

```

Excerpt from a .json report opened through a text editor

Notes

1. If there is only layered materials in the scene (e.g. V-RayBlendMtl) and the Max depth parameter of the Profiler is set to 8, no more than 5 ray bounces are displayed in the profile. Make sure to lower the Max depth for an accurate reading of such a scene.
2. If you are rendering a scene with Bitmaps multiple times, the Bitmaps will not be read by the Profiler after the first render. To profile the Bitmaps after the initial render, you need to go to Render Settings > Overrides > Rendering and click on Clear bitmap cache. Then you can proceed with a second render.
3. When reading a Profiler report in a browser, keep in mind that only up to 16 simultaneous events can be displayed. Simultaneous events are stacked down on each other when displayed this way and only up to 16 can fit at the same time, even though V-Ray can render more.