

# Diagnostic Report

The end-of-render report is a hierarchical JSON document containing some or all of the diagnostic data from a render, including a full set of data at each checkpoint. The report(s) can be customized in various ways, including restricting the output to a subset of the available metric data (e.g. only the diagnostics for the lighting subsystem). It is also possible to set up multiple reports for tracking different areas of the render.



The JSON file is **not** a direct replacement for the legacy stats XML file. It is an entirely new way of representing the data which is still in the early stages of its evolution.

There are three ways to configure the stats to produce a JSON report

1. The RenderMan option: `jsonFilename`
2. Configuring a JSON Report listener in the `stats.ini` configuration file
3. Both an option and a configuration file, which will provide either two reports or override a configured JSON Report output file name

If no filename is provided, and no JSON Report listeners are found in the configuration file, the final report is disabled.

The viewing of the diagnostic report is best done through the new RenderMan Stats Portal application, [stportal](#) which can be used for both live data presentation and offline report introspection. Its content is currently limited but useful for viewing the JSON metric hierarchy and comparing two or more reports.



Unknown Attachment

## Configuration

### Easy Setup: Render option

The simplest way to generate a report is to provide an output filename via the RenderMan option: `jsonFilename`. This will produce a default report with all available data at all checkpoints. This is essentially a shortcut for configuring a [JSON Report](#) listener with the specified `outputFilename` and an otherwise default configuration. There is currently no other control over this default report.

In the DCCs this option can be found under the "Statistics" heading where other global render settings are found. For example, in RfH:



Unknown Attachment

In RIB, the output filename can be specified via an Option:

```
Option "statistics" "string jsonFilename" ["/path/to/report.json"]
```

Similar to the configuration file's output filenames, this option string supports the use of environment variables. Standard environment variable formats are supported, as well as Linux shell variable expansion syntax. For example:

```
Option "statistics" "string jsonFilename" ["${PROD}/${SHOT}/stats${F4}.json"]
```

### Flexible Setup: Configuration file

For more control over the resulting output, a report can be enabled by specifying a [JSON Report](#) listener(s) in the `stats.ini` configuration file. For example, the following excerpt from a configuration file will enable the JSON report output, writing to: `raytrace.json`. The checkpoint gathering is disabled, and the "MetricRules" section indicates that only metrics from the `/rman/raytracing` subsystem should be included in the report.

```
[Listener]
  type "json"
  name "raytracingReport"
  outputFilename "${PROD}/${SHOT}/stats/raytracing.json"
  keepAllCheckpoints 0

[MetricRules]
  # Save only metrics from the raytracing subsystem
  [Rule]
  regexp "/rman/raytracing.*"
  samplingInterval 1000
```

Please see the [JSON Report](#) documentation for further details on options and configuration.

### Combined Setup: Output file name override

When setting up a report using the configuration file it is possible to override the `outputFilename` parameter for the configured report by specifying a qualified file name for the `RenderMan jsonFilename` option. It is not sufficient to simply specify a new JSON filename since it is technically possible to configure multiple JSON reports in a single configuration file. For the option override to take effect the option value must contain the override file name and the name of the listener to which the override should be applied.

A user can qualify the intended listener for the `jsonFilename` option override by prefixing the option value string with the listener name followed by a plus-sign ('+'), e.g.

```
Option "statistics" "string jsonfilename" ["listener_name+/new/path/to/report.json"]
```

Without this qualification, a separate report will be generated for the file name specified by the option.