


# JSON Report

JSON file output, checkpoint/resume support

## Description

Metric data gathered is written to a hierarchical JSON file. A final report is written at render termination, with intermediary stats written on checkpoint-exit events. Optionally (default: True) the data from each checkpoint will also be written.

 The end-of-render report requires knowledge of a render's endpoint in order to write out the final report. With Solaris, there really is no "end" to a render so the report isn't written out until that stats session closes which doesn't happen until you close the scene.

## Configuration

Required type for configuring the JSON report listener from an `ini` file.

`type "jsonreport"`

## Options

Listener Options	Type	Description	Required?	Default
type	string	Must be "jsonreport"	y	
name	string	Unique name of the listener	y	""
outputFile name	string	File name of final JSON report.	y	""
keepAllCheck points	bool	Turn this off to save space by only writing out JSON stats immediately before exit, otherwise, a block of stats is written on every checkpoint during the course of the render.	n	1
Metric Rules				
regex	string	One or more metrics to be observed.	n	""
samplingIn terval	integer	Sampling interval in MS for the metrics requested in preceding <code>regex</code> param.	n	6000 (1 minute)

## Configuration Example

The following excerpt from a configuration file will enable the JSON report output, writing to: `checkpoint_stats.json`. The **MetricRules** section indicates that all metrics should be gathered, with data sampled once per second. See below for an example output for this configuration.

**checkpoint\_stats.ini**

```
[Listener]
  type "json"
  name "jsonListener"
  outputFilename "checkpoint_stats.json"
  keepAllCheckpoints 1

[MetricRules]
  # Save all metrics to the JSON file, sampled once per second
  [Rule]
    regex ".*"
    samplingInterval 1000
```

## Output Example

`prman -statsconfig checkpoint_stats.ini -recover 1 scene.rib`

Metric data block omitted for brevity. Actual iteration values will vary depending on your run.

# checkpoint\_stats.json

```
{
  "header": {
    "version": "24.0b1"
  },
  "frame": {
    "attempts": [
      {
        "iteration": 20,
        "metrics": {...},
        "reason": "checkpoint"
      },
      {
        "iteration": 62,
        "metrics": {...},
        "reason": "checkpoint"
      },
      {
        "iteration": 83,
        "metrics": {...},
        "reason": "exiting"
      },
      {
        "metrics": {...},
      }
    ]
  }
}
```