

Configuration

Overview

RenderMan diagnostics have a built-in configuration for live data that is always presented in the `it` tool, toggled on with the "V" (capital "vee") keyboard shortcut. For DCC plugins the live stats are also automatically available in a separate Live Stats panel.

Advanced users may also configure other presentation [Listeners](#) through the use of an INI [configuration file](#).


Live Statistics Configuration

Each DCC has a window dedicated to the presentation of live statistics. In RfM the live stats will automatically connect when a render starts. In other DCCs the user will need to explicitly connect to the live stats using the "Attach to Render" button in the live stats panel.

Status

The line below the Attach to Render button is the current state of the connection of the internal Live Stats Server. The states are as follows:

Server State	Meaning
Not Connected	The live stats panel is not currently connected to a render.
Connected	The live stats panel is connected to the render's server and is receiving data.
Server Not Running	The live stats are not enabled. Make sure "liveStatsEnabled" is set to "1" in the configuration file.
Error: <i><error string></i>	There was an error when trying to connect to the render's stats server.

 This UI is under active development and as such, configuration options are limited for live stats. As the system evolves, so too will the ability to interactively create custom configurations of live stats.

Advanced Configuration: INI File

The stats configuration file (default name: "stats.ini") holds the basic default settings for the stats session. It can also be utilized for advanced configuration to build a list of listeners to attach, and per-listener rules for metric data to be observed by each listener.

Below is an example of a configuration file that enables an end-of-render JSON report, including checkpoints:

checkpoint_stats.ini

```
# checkpoint_stats.ini
# Stats default configuration file
#
# Copy this file to: /a/path/of/your/choosing
# then run:
# prman -statsconfig /a/path/of/your/choosing/checkpoint_stats.ini

version 0.1

# Set this to 1 for extra information when parsing this file
verbose 1

# Stats processing log level.
# Range is 0 (none) to 5 (debug output). Default is 3 (warnings)
logLevel 5

# Session configuration
[Session]
name "Checkpoint Session"

# Enable the internal live stats server
liveStatsEnabled 1

# List of listeners which the session should create and manage.
[ManagedListeners]

# JSON output report listener
[Listener]
    type "jsonreport"
    name "jsonListener"
    outputFilename "checkpoint_stats.json"
    keepAllCheckpoints 1

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



This example configuration file has both the `verbose` and `logLevel` settings dialed up for maximum verbosity when parsing the configuration file and starting the stats system, respectively. Once it is fully configured and working as expected these can be set back to their defaults of `verbose 0` and `logLevel 3`.

Listener settings that specify a filename can include environment variables which will be expanded when processed by the Listener. Standard environment variable formats are supported, as well as Linux shell variable expansion syntax:

```
$VAR
${VAR}
${VAR:-fallback}
${VAR:=fallback}
```

For example:

```
outputFilename "${HOME}/checkpoint_stats.json"
```

See [JSON Report Listener](#) reference page for JSON output configuration options.

Config File Search Order

The stats configuration file is provided to the system with the use of an environment variable or via the `prman` command line `"-statsconfig"` option. The environment variable `RMAN_STATS_CONFIG_PATH` sets a path to find a "stats.ini" configuration file. The `statsconfig` command-line option can specify a relative path to a uniquely-named configuration file or can override the path entirely if given an absolute path.

Using the example above the command line might include:

```
prman -statsconfig /path/to/config/file/checkpoint_stats.ini /path/to/rib/scene.rib
```

If prman is not provided a config file via the -statsconfig command-line option then the default configuration filename "stats.ini" will be used.

The stats system will search for the configuration file in the following order:

Order	Location	Default
1	/stats/configpath setting in rendermn.ini	.\${RMANTREE}/etc
2	RMAN_STATS_CONFIG_PATH environment variable override	none
3	prman -statsconfig </path/to/filename.ini>	stats.ini

If you specify an absolute path on the command line it will override any requested search paths. This is a convenient way to do quick testing without having to modify an existing config file. For example, suppose you normally run with a certain configuration of listeners, but then want to do a render with details printed to the console about a specific metric or group of metrics. You could do a debugging run that temporarily overrides the default configuration in one of two ways - either by setting/pre-pending the environment variable override:

```
setenv RMAN_STATS_CONFIG_PATH /my/test/directory
prman -statsconfig debug_stats.ini complicatedScene.rib
```

Or by providing a full-path, absolute file name:

```
prman -statsconfig /my/test/directory/debug_stats.ini complicatedScene.rib
```

Both of these methods will load the stats configuration from "/my/test/directory/debug_stats.ini"



Currently, configuration files are not merged if more than one is found. The last one found wins. Future releases will include advanced configuration mechanisms, including the merging of configuration files.



Listener control and configuration are not yet dynamic. In most cases, a render must be restarted in order to see the configuration change. In the case of RfM and RfB, the application will need to be restarted.

DCC Configuration

A live stats configuration UI pane is available in all RenderMan bridge products, with the exception of Solaris which does not yet have support for the new stats system. In addition, advanced configuration with an INI file is also available through the use of the config environment variable. See below for DCC-specific details.

Blender

RfB uses the prman command-line mechanism as described above, including the use of the RMAN_STATS_CONFIG_PATH override environment variable. Interactive configuration of live stats is available in the Blender preferences.

Katana

If the RMAN_STATS_CONFIG_PATH environment variable is set RfK will use that search path to look for a file named stats.ini.

If no file is found, or if that environment variable is not set then the default configuration will be used.

Additional configuration is also available in RfK through the following attributes:

```
prmanGlobalStatements.stats.configPath (default: ".")
prmanGlobalStatements.stats.configFile (default: "stats.ini")
```

These attributes are currently not exposed in PrmanGlobalStatements, they must be set via AttributeSet or OpScript at the moment. Below is an OpScript which exposes these two attributes as user args with defaults as listed above. Copy and paste into your Katana scene and modify the path and config file name as needed.

Configuration OpScript

```
<katana release="6.0v1" version="6.0.1.000002">
  <node name="__SAVE_exportedNodes" type="Group">
    <node baseType="OpScript" edited="true" name="AdvancedStatsConfiguration" ns_colorb="0.05" ns_colorg="0.26"
ns_colorr="0.09" ns_errorGlow="0.0" ns_fromContext="legacy" selected="true" type="OpScript" viewed="true">
      <port name="i0" type="in"/>
      <port name="out" type="out"/>
      <group_parameter name="AdvancedStatsConfiguration">
        <string_parameter name="CEL" value="(/root)"/>
        <string_parameter name="location" value="/root/world/location"/>
        <group_parameter name="script">
          <string_parameter name="lua" value="
-- Default: 'stats.ini'
local configFile = Interface.GetOpArg('user.configFile'):getValue()

-- Default: '.:${RMANTREE}/etc'
-- Can be overridden with RMAN_STATS_CONFIG_PATH
local configPath = Interface.GetOpArg('user.configPath'):getValue()

Interface.SetAttr(
  'prmanGlobalStatements.stats.configFile',
  StringAttribute(configFile))

Interface.SetAttr(
  'prmanGlobalStatements.stats.configPath',
  StringAttribute(configPath))
"/>
        </group_parameter>
        <string_parameter name="executionMode" value="immediate"/>
        <string_parameter name="applyWhere" value="at locations matching CEL"/>
        <string_parameter name="applyWhen" value="during op resolve"/>
        <string_parameter name="modifierNameMode" value="node name"/>
        <string_parameter name="modifierName" value="modifier"/>
        <string_parameter name="resolveIds" value=""/>
        <number_parameter name="recursiveEnable" value="0"/>
        <string_parameter name="disableAt" value=""/>
        <string_parameter name="inputBehavior" value="by index"/>
        <number_parameter name="multisampleUserOpArgs" value="0"/>
        <group_parameter hints="{}" name="user">
          <string_parameter hints="{ 'widget': 'fileInput' }" name="configFile" value="stats.ini"/>
          <string_parameter expression="'.:'+getenv('RMANTREE', '.')+'/etc'" hints="{}" name="configPath"/>
        </group_parameter>
      </group_parameter>
    </node>
  </node>
</katana>
```

Maya

If the RMAN_STATS_CONFIG_PATH environment variable is set RfM will use that search path to look for a file named stats.ini.

If no file is found, or if that environment variable is not set then the default configuration will be used.

Houdini/Solaris

If the RMAN_STATS_CONFIG_PATH environment variable is set RfH will use that search path to look for a file named stats.ini.

If no file is found, or if that environment variable is not set then the default configuration will be used.



Houdini/Solaris Known Issue: Live stats will only work properly with either Solaris or Classic Houdini. If a scene is being rendered in one and then switched to the other the live stats will no longer work reliably. If this happens the scene must be closed and reopened, or the application must be restarted.