This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Stat Server

log

4/22/2025

log

- alarm
- all
- buffering
- check-point
- compatible-output-priority
- debug
- enable-thread
- expire
- interaction
- keep-startup-file
- memory
- memory-storage-size
- message_format
- messagefile
- print-attributes
- segment
- spool
- standard
- throttle-period
- throttle-threshold
- time_convert
- time_format
- trace
- verbose
- x-conn-debug-all
- x-conn-debug-api
- x-conn-debug-dns
- x-conn-debug-open
- x-conn-debug-security
- x-conn-debug-select
- x-conn-debug-timers
- x-conn-debug-write

The `[log]` section is defined on the `Options` tab of the Stat Server Application object.

This section describes log configuration options that are common to all Genesys server applications and applicable to any Framework server component. Note that to use these options, you must actively set them, manually on the `Options` tab of the Stat Server `Application` object within Genesys Administrator. These options are generic options that apply to all Genesys server applications. Refer to the *Configuration Options Reference Manual*, available on the Management Framework page, and to the Genesys Security Deployment Guide for additional information.

For your convenience, the SIP Server product provides a troubleshooting tool that parses the log output of several Genesys servers including Stat Server. Refer to the *SipSpan2 User's Guide*, available on the SIP Server CD, for information on how to use this tool.

See also Log Output Options.

alarm

Default Value: No default value

Valid Values:

- **stdout** Log events are sent to the Standard output (stdout).

- **stderr** Log events are sent to the Standard error output (stderr).
- **network** Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
- **memory** Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
- **<filename>** Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends log events of Alarm level. You must separate log output types with commas when you configure more than one output type.

For example, `alarm = stdout, logfile`

all

Default Value: No default value

Valid Values:

- **stdout** Log events are sent to the Standard output (stdout).
- **stderr** Log events are sent to the Standard error output (stderr).
- **network** Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
  Setting the all log level option to the network output enables an application to send log events of the Standard, Interaction, and Trace levels to Message Server. Debug-level log events are neither sent to Message Server nor stored in the Log Database.
- **memory** Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
- **<filename>** Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which Stat Server sends all log events. You must separate log output types with commas when you configure more than one output type.

For example, `all = stdout, logfile`

Note: To ease the troubleshooting process, consider using unique prefixes for log files that different Stat Server applications generate.

buffering

Default Value: true

Valid Values:

- **true** Enables buffering.
- **false** Disables buffering.

Changes Take Effect: Immediately

Turns operating system file buffering on or off. This option applies only to `stderr` and `stdout` output. Setting this option to `true` increases output performance.

Note: When you enable buffering, messages may appear at the console with delay.

check-point

Default Value: 1
Valid Values: 0 - 24
Changes Take Effect: Immediately

Specifies, in hours, how often Stat Server generates a check point log event, to divide the log into sections of equal time. By default, Stat Server generates this log event every hour. Setting the option to 0 prevents the generation of check-point events.

compatible-output-priority

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately

Specifies whether the application uses 6.x output logic.

Valid values:

- **true**: The log of the level specified by Log Output Options is sent to the specified output.

- **false**: The log of the level specified by Log Output Options and higher levels is sent to the specified output.

For example, you configure the following options in the log section for a 6.x application and for a 8.x application:

```
[log]
verbose=all
debug=file1
standard=file2
```

Stat Server 6.x log file content is as follows:

- `file1` contains `debug` messages only.
- `file2` contains `standard` messages only.

Stat Server 8.x log file content is as follows:

- `file1` contains `debug`, `trace`, `interaction`, and `standard` messages.
- `file2` contains `standard` messages only.

Warning! Genesys does not recommend changing the default value of the `compatible-output-priority` option unless you have specific reasons to use the 6.x log output logic—that is, to mimic the output priority as implemented in releases 6.x. Setting this option to true affects log consistency.

debug

Default Value: No default value

Valid Values:

- **stdout** Log events are sent to the Standard output (`stdout`).
- **stderr** Log events are sent to the Standard error output (`stderr`).
- **memory** Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
- **<filename>** Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the Debug level and higher (that is, log events of the `Standard`, `Interaction`, `Trace`, and `Debug` levels). You must separate log output types with commas when you configure more than one output type.

For example, debug = `stderr, /usr/local/genesys/logfile`

Note: Log events of debug level are never sent to Message Server nor are they stored in the Log Database.

enable-thread

Default Value: false
Valid Values: true, false
Changes Take Effect: Immediately
Introduced: 8.5.106
Related Options: throttle-period, throttle-threshold
Related Links: log throttling

Specifies whether to enable or disable the logging thread. If set to `true` (the logging thread is enabled), the logs are stored in an internal queue to be written to the specified output by a dedicated logging thread. This setting also enables the log throttling feature, which allows the `verbose` level to be dynamically reduced when a logging performance issue is detected. Refer to the Framework Management Layer User’s Guide for more information about the log throttling feature.

If this option is set to `false` (the logging thread is disabled), each log is written directly to the outputs by the thread that initiated the log request. This setting also disables the log throttling feature.

expire

Default Value: 10

Valid Values:

- **false** No expiration; all generated segments are stored.
- **<number> file or <number>** Sets the maximum number of log files to store. Specify a number from 1-1000.
- **<number> day** Sets the maximum number of days before log files are deleted. Specify a number from 1-100.

Changes Take Effect: Immediately

Determines whether log files expire. If they do, this option sets the measurement for determining when they expire, along with the maximum number of files (segments) or days before the files are removed. Stat Server ignores this option if you configure log output to be sent to other than a log file.

interaction

Default Value: No default value

Valid Values:

- **stdout** Log events are sent to the Standard output (`stdout`).
- **stderr** Log events are sent to the Standard error output (`stderr`).
- **network** Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
- **memory** Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
- **<filename>** Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Interaction` level and higher (that is, log events of the `Standard` and `Interaction` levels). You must separate log outputs with commas when you configure more than one output type.

For example, `interaction = stderr, network`

keep-startup-file

Default Value: false

Valid Values:

- **false** No startup segment of the log is kept.
- **true** A startup segment of the log is kept. The size of the segment equals the value of the `segment` option.
- **<number> KB** Sets the maximum size, in kilobytes, for a startup segment of the log.
- **<number> MB** Sets the maximum size, in megabytes, for a startup segment of the log.

Changes Take Effect: After restart
Dependencies: segment is not false

Specifies whether a startup segment of the log, containing the initial configuration, is to be kept. If it is, this option can be set to true or to a specific size. If set to `true`, the size of the initial segment will be equal to the size of the regular log segment defined by the `segment` option. The value of this option will be ignored if segmentation is turned off (that is, if the `segment` option set to `false`).

memory

Default Value: No default value
Valid Values: <string> (memory file name)
Changes Take Effect: Immediately

Specifies the name of the file to which Stat Server regularly prints a snapshot of the memory output, if it is configured to do this. The new snapshot overwrites the previously written data. If Stat Server terminates abnormally, this file will contain the latest log messages. Memory output is not recommended for processors with a CPU frequency lower than 600 MHz.

Note: If the file specified is located on a network drive, Stat Server does not create a snapshot file (with the extension `*.memory.log`).

Refer to Log Output Options for more information.

memory-storage-size

Default Value: 2 MB

Valid Values:

- **<number> KB or <number>** The size of the memory output, in kilobytes. The minimum value is 128 KB.
- **<number> MB** The size of the memory output, in megabytes. The maximum value is 64 MB

Changes Take Effect: When memory output is created

Specifies the buffer size for log output to the memory. Refer to Log Output Options for more information.

message_format

Default Value: short

Valid Values:

- **short** An application uses compressed headers when writing log records in its log file.
- **full** An application uses complete headers when writing log records in its log file.

Changes Take Effect: Immediately

Specifies the format of log record headers that Stat Server uses when it writes to its log file. Using compressed log record headers improves Stat Server performance and reduces the log's file size. With the value set to short:

- A header of the log file or the log file segment contains information about the application (such as the application name, application type, host type, and time zone), whereas single log records within the file or segment omit this information.
- A log message priority is abbreviated to `Std`, `Int`, `Trc`, or `Dbg`, for `Standard`, `Interaction`, `Trace`, or `Debug` messages, respectively.
- The `message ID` does not contain the prefix GCTI or the application type ID.

A log record in the `full` format looks like this:

```
2015-05-07T18:11:38.196 Standard localhost StatServer GCTI-00-05060 Application
started
```

A log record in the `short` format looks like this:

```
2015-05-07T18:15:33.952 Std 05060 Application started
```

Note: Whether the full or short format is used, time is printed as specified by the time_format option.

messagefile

Default Value: StatServer.lms
Valid Values: <string>.lms
Changes Take Effect: After restart or Immediately, if Stat Server cannot locate the statserver.lms file at startup

Specifies the file name for Stat Server log events. The name must be valid for the operating system on which Stat Server is running. The option value can also contain the absolute path to the `StatServer.lms` file. Otherwise, Stat Server looks for the file in its working directory.

Warning! An application that does not find its `*.lms` file at startup cannot generate application-specific log events and send them to Message Server.

print-attributes

Default Value: false

Valid Values:

- **true** Attaches extended attributes, if any exist, to a log event sent to log output.
- **false** Does not attach extended attributes to a log event sent to log output.

Changes Take Effect: Immediately

Specifies whether the application will attach extended attributes, if any exist, to a log event that the application sends to log output. Typically, log events of the `Interaction` log level and Audit-related log events contain extended attributes. Setting this option to `true` enables audit capabilities, but negatively affects performance. Genesys recommends enabling this option only when testing new interaction scenarios.

Refer to the *Genesys Combined Log Events Help*, available on the Management Framework page, for information about extended attributes.

segment

Default Value: 100 MB

Valid Values:

- **false** No segmentation is allowed.
- **<number> KB or <number>** Sets the maximum segment size in kilobytes. The minimum segment size is 100 KB.
- **<number> MB** Sets the maximum segment size in megabytes. The maximum value is 2047 MB.
- **<number> hr** Sets the number of hours for the segment to stay open. The minimum number is 1 hour.

Changes Take Effect: Immediately

Specifies whether there is a segmentation limit for a log file. If there is, sets the mode of measurement, along with the maximum size. If the current log segment exceeds the size set by this option, the file is closed and a new one is created. Stat Server ignores this option if log output is not configured to be sent to a log file.

spool

Default Value: The Stat Server working directory
Valid Values: <path> The full path of the folder
Changes Take Effect: Immediately

Specifies the folder, including full path to it, in which Stat Server creates temporary files related to network log output. If you change the option value while Stat Server is running, the change does not affect the currently open network output.

standard

Default Value: No default value

Valid Values:

- **stdout** Log events are sent to the Standard output (stdout).
- **stderr** Log events are sent to the Standard error output (stderr).
- **network** Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
- **memory** Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
- **<filename>** Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Standard` level. You must separate log output types with commas when you configure more than one output type.

For example, `standard = stderr, network`

throttle-period

Default Value: 30
Valid Values: 0-3600
Changes Take Effect: Immediately
Dependencies: enable-thread = true
Introduced: 8.5.106

Specifies, in seconds, how long to keep the throttled verbose level. When this period of time has expired, the original log verbose level will be restored when the log queue size has decreased to less than 50% of the threshold.

This option applies only if enable-thread is set to `true`.

throttle-threshold

Default Value: 5000
Valid Values: 0-10000
Changes Take Effect: Immediately
Dependencies: enable-thread = true
Introduced: 8.5.106
Related Links: log throttling

Specifies the size of the internal log queue at which the verbose level is to be reduced so as to lessen the load generated by logging. If this option is set to 0 (zero), throttling does not occur. For more information about log throttling, refer to the *Framework Management Layer User's Guide*.

This option applies only if enable-thread is set to `true`.

time_convert

Default Value: local

Valid Values:

- **local** The time of log-record generation is expressed as a local time, based on the time zone and any seasonal adjustments. Time zone information of Stat Server's host computer is used.
- **utc** The time of log-record generation is expressed as Coordinated Universal Time (UTC).

Changes Take Effect: Immediately

Specifies the system in which Stat Server calculates the log record time when generating a log file. The time is converted from the time in seconds since the Epoch (00:00:00 UTC, January 1, 1970).

time_format

Default Value: time

Valid Values:

- **time** The time string is formatted according to the HH:MM:SS.sss (hours, minutes, seconds, and milliseconds) format.
- **locale** The time string is formatted according to the system's locale.
- **ISO8601** The date in the time string is formatted according to the ISO 8601 format. Fractional seconds are given in milliseconds.

Changes Take Effect: Immediately

Specifies how to represent, in a log file, the time when Stat Server generates log records.
A log record's time field in the ISO 8601 format looks like this:

```
2001-07-24T04:58:10.123
```

trace

Default Value: No default value

Valid Values:

- **stdout** Log events are sent to the Standard output (`stdout`).
- **stderr** Log events are sent to the Standard error output (`stderr`).
- **network** Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database.
- **memory** Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance.
- **<filename>** Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory.

Changes Take Effect: Immediately

Specifies the outputs to which an application sends the log events of the `Trace` level and higher (that is, log events of the `Standard`, `Interaction`, and `Trace` levels). You must separate log outputs with commas when you configure more than one output type.

For example, `trace = stderr, network`

verbose

Default Value: all

Valid Values:

- **all** All log events (that is, log events of the `Standard`, `Trace`, `Interaction`, and `Debug` levels) are generated if you set the debug-level option in the statserver section to `all`.
- **debug** The same as `all`.
- **trace** Log events of the `Trace` level and higher (that is, log events of the `Standard`, `Interaction`, and `Trace` levels) are generated, while log events of the `Debug` level are not generated.
- **interaction** Log events of the `Interaction` level and higher (that is, log events of the `Standard` and `Interaction` levels) are generated, while log events of the `Trace` and `Debug` levels are not generated.
- **standard** Log events of the `Standard` level are generated, while log events of the `Interaction`, `Trace`, and `Debug` levels are not generated.
- **none** Produces no output.

Changes Take Effect: Immediately

Determines whether a log output is created. If it is, this option specifies the minimum level of log events generated. The log events levels, starting with the highest-priority level, are `standard`, `interaction`, `trace`, and debug. Refer to Log Output Options for more information.

Refer to the *Framework Management Layer User's Guide*, available on the Management Framework page, for more information on the `standard`, `trace`, `interaction`, and debug log levels.

x-conn-debug-all

Default Value: 0

Valid Values:

- **0** Log records are not generated.
- **1** Log records are generated.

Changes Take Effect: After restart

Generates debug log records about open connection, socket select, timer creation and deletion, write, security-related, DNS operation, and connection library function calls. This option is the same as enabling or disabling all of the previous `x-conn-debug-<optype>` options.

x-conn-debug-api

Default Value: 0

Valid Values:

- **0** Log records are not generated.
- **1** Log records are generated.

Changes Take Effect: After restart

Generates debug log records about connection library function calls.

x-conn-debug-dns

Default Value: 0

Valid Values:

- **0** Log records are not generated.
- **1** Log records are generated.

Changes Take Effect: After restart

Generates debug log records about DNS operations.

x-conn-debug-open

Default Value: 0

Valid Values:

- **0** Log records are not generated.
- **1** Log records are generated.

Changes Take Effect: After restart

Generates debug log records about "open connection" operations.

x-conn-debug-security

Default Value: 0

Valid Values:

- **0** Log records are not generated.
- **1** Log records are generated.

Changes Take Effect: After restart

Generates debug log records about security-related operations, such as Transport Layer Security (TLS) and security certificates. This option has no effect on Stat Server 7.6 and earlier releases, which do not support TLS operations.

x-conn-debug-select

Default Value: 0

Valid Values:

- **0** Log records are not generated.
- **1** Log records are generated.

Changes Take Effect: After restart

Generates debug log records about "socket select" operations.

x-conn-debug-timers

Default Value: 0

Valid Values:

- **0** Log records are not generated.
- **1** Log records are generated.

Changes Take Effect: After restart

Generates debug log records about the timer creation and deletion operations.

x-conn-debug-write

Default Value: 0

Valid Values:

- **0** Log records are not generated.

- **1** Log records are generated.

Changes Take Effect: After restart

Generates Debug log records about “write” operations of the application.

Log Output Options

To configure log outputs, set log level options (all, alarm, standard, interaction, trace, memory, and/or debug) to the desired types of log output (`stdout`, `stderr`, `network`, `memory`, and/or `[filename]` for log file output).

You can use:

- One log level option to specify different log outputs.
- One log output type for different log levels.
- Several log output types simultaneously for logging the events of the same or different log levels.

You must separate the log output types by a comma when you are configuring more than one output for the same log level. See examples below.

Note: The log output options are activated according to the setting of the verbose configuration option.

Warnings:

- If you direct log output to a file on the network drive, an application does not create a snapshot log file (with the extension `*.snapshot.log`) in case it terminates abnormally.
- Directing log output to the console (by using the `stdout` or `stderr` settings) can affect application performance. Avoid using these log output settings in a production environment.

Example 1: Production Mode [log] Section

```
[log]
verbose=standard
standard=network,statservlogfile
```

With this configuration, Stat Server generates only log events of the `Standard` level and sends them to the standard output, to Message Server, and to a file named `statservlogfile`, which Stat Server creates in its working directory. Genesys recommends that you use this or a similar configuration in a production environment.

Warning! Directing log output to the console (by using the `stdout` or `stderr` settings) can affect application performance. Avoid using these log output settings in a production environment.

Example 2: Lab Mode [log] Section

```
[log]
verbose=all
all=stdout,/usr/local/genesys/statservlogfile
trace=network
```

With this configuration, Stat Server generates log events of the `standard`, `interaction`, `trace`, and debug levels and sends them to the standard output and to a file named `statservlogfile`, which Stat Server creates in the `/usr/local/genesys/` directory. In addition, Stat Server sends log events of the `standard`, `interaction`, and `trace` levels to Message Server. Use this configuration to test new interaction scenarios in a laboratory environment. Be sure to appropriately set the debug-level option in the `statserver` section.

Example 3: Failure-Troubleshooting [log] Section

```
[log]
verbose=all
standard=network
all=memory
memory=statservlogfile
memory-storage-size=32 MB
```

With this configuration, Stat Server generates log events of the `standard` level and sends them to Message Server. It also generates log events of all levels and sends them to the memory output. The most current log is stored to a file named `statservlogfile`, which the application creates in its working directory. An increased memory storage enables Stat Server to save more log information generated before a failure. Use this configuration when trying to reproduce an application failure. The memory log file would contain the snapshot of Stat Server’s log at the moment of failure. This should help you and Genesys Customer Care identify the reason for the failure. Be sure to appropriately set the debug-level option in the `statserver` section.

Note: If you are operating Stat Server on Unix and do not specify any files in which to store the memory output snapshot, the core file that Stat Server produces before terminating contains the most current Stat Server log. Provide the Stat Server’s core file to Genesys Customer Care when reporting problems.

Log File Extensions

You can use the following file extensions to identify log files that Stat Server creates for various types of output:

- `*.log`—Assigned to log files when you configure output to a log file. For example, if you set `standard = statservlog`, Stat Server prints log messages into a text file called `statservlog.<time_stamp>.log`.
- `*.qsp`—Assigned to temporary (spool) files when you configure output to the network, but the network is temporarily unavailable. For example, if you set standard = network, Stat Server prints log messages into a file called statserv.<time_stamp>.qsp during the time the network is unavailable.
- `*.snapshot.log`—Assigned to files containing the output snapshot when you configure output to a log

file. The file contains the last log messages that Stat Server generates before abnormal termination. For example, if you set `standard = statservlog`, Stat Server prints the last log message into a file called `statserv.<time_stamp>.snapshot.log` in case of failure. If Stat Server terminates normally, the snapshots logs are deleted.
Note: Provide *.snapshot.log files to Genesys Customer Care when reporting a problem.

- `*.memory.log`—Assigned to log files that contain the memory output snapshot when you configure output to memory and redirect the most recent memory output to a file. For example, if you set `standard = memory` and `memory = statserv`, Stat Server prints the latest memory output to a file called `statserv.<time_stamp>.memory.log`.

Debug Logs

The `x-conn-debug-<optype>` configuration options:

- x-conn-debug-all
- x-conn-debug-api
- x-conn-debug-dns
- x-conn-debug-open
- x-conn-debug-security
- x-conn-debug-select
- x-conn-debug-timers
- x-conn-debug-write

enable you to generate debug logs containing information about specific Stat Server operations. You designate these options in the log section of the Stat Server application.

Warning! Genesys advises you to use these options only when requested by Genesys Customer Care.