



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.

Genesys Engage Digital (eServices)

[log](#)

log

- all
- buffering
- expire
- messagefile
- segment
- standard
- time-format
- trace
- verbose

all

Default Value: stdout

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 an application sends all log events. The log output types must be separated by a comma when more than one output is configured. For example: all = stdout, logfile

buffering

Default Value: false

Valid Values:

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

Changes Take Effect: Immediately

Turns on/off operating system file buffering. The option is applicable only to the stderr and stdout output. Setting this option to true increases the output performance.

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 1-100.
- *<number>* day - Sets the maximum number of days before log files are deleted. Specify a number from 1-100.

If an option's value is set incorrectly-out of the range of valid values - it is automatically reset to 10.

Changes Take Effect: Immediately

Determines whether log files expire. If they do, sets the measurement for determining when they expire, along with the maximum number of files (segments) or days before the files are removed. This option is ignored if log output is not configured to be sent to a log file.

messagefile

Default Value: ChatServer.lms

Valid Values: ChatServer.lms

Changes Take Effect: Immediately

Logging message file used by Chat Server.

segment

Default Value: 10MB

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.
- *<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. This option is ignored if log output is not configured to be sent to a log file.

standard

Default Value: stdout

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 an application sends the log events of the Standard level. The log output types must be separated by a comma when more than one output is configured. For example: standard = stderr, network

time-format

Default Value: ISO8601

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. Fractions of seconds are given in milliseconds.

Changes Take Effect: Immediately

Specifies how to represent, in a log file, the time when an application generates log records.

trace

Default Value: stdout

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 an application sends the log events of the Trace level and higher (that is, log events of the Standard, Interaction, and Trace levels). The log outputs must be separated by a comma when more than one output is configured. 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.
- **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, but 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, but log events of the Trace and Debug levels are not generated.
- **standard:** Log events of the Standard level are generated, but log events of the Interaction, Trace, and Debug levels are not generated.
- **none:** No output is produced.

Changes Take Effect: Immediately

Determines whether a log output is created. If it is, specifies the minimum level of log events generated. The log events levels, starting with the highest priority level, are Standard, Interaction, Trace, and Debug.