



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.

Workforce Management

Log Section

12/19/2025

Log Section

- all
- buffering
- debug
- expire
- extended
- segment
- standard
- trace
- verbose

all

Default Value: " " (string)

Valid Values: stdout, stderr, network, memory, [filename]

Changes Take Effect: Immediately

Dependencies: None

This option is mandatory.

Specifies the log output type that applications send for log events of all levels (Standard, Trace, and Debug). The log output types must be separated by a comma when more than one output is configured. For example: `all = stdout, logfile`.

The output types are described as follows:

- `stdout`—Log events are sent to the Standard output.
- `stderr`—Log events are sent to the Standard error output.
- `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 applications to send log events of the Standard, 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.

Warning

Do not use the option value `network` unless requested to do so by Genesys Professional Services, because it generates extremely heavy message loads that can degrade system performance.

buffering

Default Value: false

Valid Values: true, false

Changes Take Effect: Immediately

Dependencies: None

This option is mandatory.

Specifies whether operating system file buffering is enabled or disabled. The option is applicable only to the `stderr` and `stdout` output types.

Setting this option to `true` enables buffering and increases the output performance. When buffering is enabled, there might be a delay before log messages appear at the console. Setting this option to `false` disables buffering.

debug

Default Value: " " (string)

Valid Values: `stdout`, `stderr`, `memory`, `[filename]`

Changes Take Effect: Immediately

Dependencies: None

This option is mandatory.

Specifies the log output type that applications send for log events of the Debug level and higher (that is, Standard, Interaction, Trace, and Debug levels). The log output types must be separated by a comma when more than one output is configured. For example: `debug = stderr, /usr/local/genesys/logfile`.

Debug-level log events are never sent to Message Server or stored in the Log database.

The output types are described as follows:

- `stdout`—Log events are sent to the Standard output.
- `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.

expire

Default Value: false

Valid Values: false, `[number]`, `[number] file`, `[number] day`

Changes Take Effect: Immediately

Dependencies: None

This option is mandatory.

Specifies whether or not log files expire and if they do, sets the expiration mode for old segments and the maximum number of files (segments) or days before the files are removed. The number stored cannot be:

- Less than 1 file or 1 day
- More than 100 files or 100 days.

Setting this option value to `false` indicates that files do not expire. This option is ignored if the log output is not configured to be sent to a log file.

Tip

If you set this option value incorrectly (out of the range of valid values) WFM automatically resets it to 10.

The valid values are described as follows:

- `false`—No expiration; all generated segments are stored.
- `[number]` or `[number] file`—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.

extended

Default Value: `false`

Valid Values: `true`, `false`

Changes Take Effect: Immediately

Dependencies: none

Introduced: 8.5.216.07

Specifies whether or not WFM enables additional log messages related to session creation, notifications, and reporting.

Set this option value to `true` to enable additional log messages.

segment

Default Value: `false`

Valid Values: `false`, `[number]`, `[number] KB`, `[number] MB`, `[number] hr`

Changes Take Effect: Immediately

Dependencies: None

This option is mandatory.

Specifies whether or not there is a segmentation limit for a log file and if there is, sets the increment (KB, MB, or hours) and maximum size for the log file segment. The number cannot be less than 100 KB or one hour.

If this option value is set to `false` the log file is not segmented. If the current log segment exceeds the size set by this option, WFM Builder closes the file and creates a new one. WFM Builder ignores this option if the log output is not configured to be sent to a log file.

The valid values are described, as follows:

- `false`—No segmentation is allowed.
- `[number]` or `[number] KB`—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.

standard

Default Value: `stdout`

Valid Values: `stdout`, `stderr`, `network`, `memory`, `[filename]`

Changes Take Effect: Immediately

Dependencies: None

This option is mandatory.

Specifies the log output type of which applications will send the log events of the Standard level. For centralized logging, set this option value to `network`. You can also use a local file name or `stdout`.

The log output types must be separated by a comma when more than one output is configured. For example: `standard = stderr, network`

The valid values are described, as follows:

- `stdout`—Log events are sent to the Standard output.
- `stderr`—Log events are sent to the Standard error output.
- `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, 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.

trace

Default Value: " " (string)

Valid Values: stdout, stderr, network, memory, [filename]

Changes Take Effect: Immediately

Dependencies: None

This option is mandatory.

Specifies the output type to which applications send Trace and Standard level log events. For centralized logging, set this option value to network. You can also use a local file name or stdout.

Log outputs must be separated by a comma when more than one output is configured. For example, trace = stderr, network.

The valid values are described, as follows:

- stdout—Log events are sent to the Standard output.
- stderr—Log events are sent to the Standard error output.
- 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 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 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.

Warning

Do not use the option value network unless requested to do so by Genesys Professional Services, because it generates extremely heavy message loads that can degrade system performance.

verbose

Default Value: standard

Valid Values: all, debug, trace, standard, none, yes (= all), no (= none)

Changes Take Effect: Immediately

Dependencies: None

This option is mandatory.

Specifies whether or not a log output is created and if it is, sets the minimum level of log events generated. The log events levels, starting with the highest priority level, are Standard, Trace, and Debug.

The valid values are described, as follows:

- `all`—All log events (that is, Standard, Trace, and Debug levels) are generated.
- `debug`—The same as `all`.
- `trace`—Log events of the Trace level and higher (that is, Standard, and Trace levels) are generated, but log events of the Debug level are not generated.
- `standard`—Log events of the Standard level are generated, but log events of the Trace, and Debug levels are not generated.
- `none`—No output is produced.