



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
- keep-startup-file
- messagefile
- segment
- standard
- trace
- verbose
- x-json-log
- x-json-log.archive
- x-json-log.compress
- x-json-log.flush
- x-json-log.MaxLogRecordSize
- x-json-log.purgeAge
- x-json-log.purgeCount
- x-json-log.rotateOnOpen
- x-json-log.rotation
- x-json-log.times
- x-ScheduleBuilderProgressTrace
- x-ScheduleBuilderTrace
- x-ScheduleLogPath
- x-ScheduleMaxLogs
- x-SwordTrace

all

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 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. Setting this option to `false` disables buffering.

When buffering is enabled, there might be a delay before log messages appear at the console.

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.

keep-startup-file

Default Value: false

Valid Values: true, false

Changes Take Effect: After restart

Dependencies: None

Introduced: 8.5.210

Specifies whether WFM Builder keeps or deletes a startup segment of the log containing the initial configuration options when it reaches the value in the **expire** option.

If this option value is set to `false`, WFM Builder does not keep the startup segment of the log.

If this option value is set to `true`, WFM Builder keeps a startup segment of the log. The size of the segment equals the value of the **segment** option. The value `<number> KB` (kilobytes) or `<number> MB` (megabytes) sets the maximum size for a startup segment of the log.

If this option value is set to `true`, the size of the initial segment is equal to the size of the regular log segment, as defined by the **segment** option. The value of this option is ignored if segmentation is turned off (that is, if the **segment** option value is set to `false`).

messagefile

Default Value: `wfmbuilder.lms`

Valid Values: `wfmbuilder.lms`

Changes Take Effect: Immediately

Dependencies: None

This option is mandatory.

Specifies the name of the input file that stores application-specific log or error messages that might be generated when WFM Builder communicates with other Genesys components.

The only valid value is `wfmbuilder.lms`.

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. If the log output is not configured to be sent to a log file, WFM Builder ignores this option.

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.

x-json-log

Default Value: "" (empty string)

Valid Values: stdout, stderr, [filename with or without path]

Changes Take Effect: After restart

Introduced: 8.5.220.02

Enables the JSON logging for WFM servers, if not empty.

WFM now supports structured logging to the file containing one-line log records formatted in JSON format. This facilitates easy integration with the centralized log aggregation and viewing systems, like Grafana/Loki.

Important

The log level for structured logging is the same as for logging to a file and is defined by a regular verbose option.

Important

All **x-json-log** configuration options are not included in the application template. You can configure them manually.

Sample Configuration

```
[Log]
x-json-log = c:\logs\structured-log.log
x-json-log.rotation = 10 M
x-json-log.archive = timestamp
x-json-log.purgeCount = 10
x-json-log.purgeAge = 30 days
x-json-log.rotateOnOpen = true
x-json-log.compress = true
x-json-log.times = local
x-json-log.flush = false
```

x-json-log.archive

Default Value: No default value

Valid Values: number, timestamp

Changes Take Effect: After restart

Dependencies: Used if **x-json-log** option is set

Introduced: 8.5.220.02

The log file's archive mode.

Using the **archive** property, it is possible to specify how archived log files are named. The following values for the **archive** property are supported:

- **number:** A number, starting with 0, is appended to the name of archived log files. The most recent archived log file always has the number 0. For example, if the log file is named *access.log*, and if it fulfils the rotation criteria, the file is renamed to *access.log.0*. If a file named *access.log.0* already

exists, it is renamed to *access.log.1*.

- **timestamp:** A timestamp is appended to the log file name. For example, if the log file is named *access.log*, and it fulfils the criteria for rotation, the file is renamed to **access.log.20050802110300**.

x-json-log.compress

Default Value: No default value

Valid Values: true, false

Changes Take Effect: After restart

Dependencies: Used if **x-json-log** option is set

Introduced: 8.5.220.02

Enable or disable compression of archived files. Archived log files can be compressed using the gzip compression method. Compressing can be controlled with the **compress** property.

The following values are supported:

- **true:** Compress archived log files.
- **false:** Do not compress archived log files.

x-json-log.flush

Default Value: true

Valid Values: true, false

Changes Take Effect: After restart

Dependencies: Used if **x-json-log** option is set

Introduced: 8.5.220.02

Specifies whether messages are immediately flushed to the log file. Valid values are:

- **true:** Every message is immediately flushed to the log file (which may hurt application performance but ensures that everything is in the log if there is a system crash).
- **false:** Messages are not immediately flushed to the log file and stay in the system file buffer for some time.

x-json-log.MaxLogRecordSize

Default Value: -1

Valid Values: any integer

Changes Take Effect: After restart

Dependencies: Used if **x-json-log** option is set

Introduced: 8.5.220.02

Contains the maximum size in bytes for logging records.

The default value is -1 which means unlimited, but that is only if the logging level is "all" or "debug". If the logging level is "trace" or "standard" then the default value is 10000 - that is because Genesys

log library does not accept messages larger than 10000 when the logging level is "trace" or "standard".

x-json-log.purgeAge

Default Value: "" (empty string)

Valid Values:

Changes Take Effect: After restart

Dependencies: Used if **x-json-log** option is set

Introduced: 8.5.220.02

Maximum age of an archived log file before it is purged. Archived log files can be automatically purged, either if they reach a certain age, or if the number of archived log files reaches a given maximum number. This is controlled by the **purgeAge** and `x-json-log.purgeCount` properties. The **purgeAge** property can have the following values:

- `<n> [seconds]`: The maximum age is `<n>` seconds.
- `<n> minutes`: the maximum age is `<n>` minutes.
- `<n> hours`: The maximum age is `<n>` hours.
- `<n> days`: The maximum age is `<n>` days.
- `<n> weeks`: The maximum age is `<n>` weeks.
- `<n> months`: The maximum age is `<n>` months, where a month has 30 days.

x-json-log.purgeCount

Default Value: "" (empty string)

Valid Values: integer, none

Changes Take Effect: After restart

Dependencies: Used if **x-json-log** option is set

Introduced: 8.5.220.02

Maximum number of archived log files before it is purged. The **purgeCount** property has an integer value that specifies the maximum number of archived log files. If the number is exceeded, archived log files are deleted, starting with the oldest. When *none* or empty string are supplied, they reset **purgeCount** to none (no purging).

x-json-log.rotateOnOpen

Default Value: false

Valid Values: true, false

Changes Take Effect: After restart

Dependencies: Used if **x-json-log** option is set

Introduced: 8.5.220.02

Specifies whether an existing log file should be rotated and archived when the file is opened. Valid values are:

- `true`: The log file is rotated (and archived) when the channel is opened.
- `false`: (Default value) Log messages will be appended to an existing log file if the file exists (unless other conditions for a rotation are met).

x-json-log.rotation

Default Value: no default value

Valid Values:

Changes Take Effect: After restart

Dependencies: Used if **x-json-log** option is set

Introduced: 8.5.220.02

The log file's rotation strategy. Possible values:

- `never`: No log rotation
- `[day, ][hh]:mm`: The file is rotated on specified day/time
 - `day` - Day is specified as a long day name (Monday, Tuesday,...) or short day name (Mon, Tue,...)
 - `day` can be omitted and when omitted, the log is rotated every day.
 - `hh` - Valid hour range is 00-23.
 - `hour` can be omitted, in which case log is rotated every hour.
 - `mm` - Valid minute range is 00-59.
 - `minute` must be specified.
- `daily`: The file is rotated daily.
- `weekly`: The file is rotated every seven days.
- `monthly`: The file is rotated every 30 days.
- `<n> minutes`: The file is rotated every `<n>` minutes, where `<n>` is an integer greater than zero.
- `<n> hours`: The file is rotated every `<n>` hours, where `<n>` is an integer greater than zero.
- `<n> days`: The file is rotated every `<n>` days, where `<n>` is an integer greater than zero.
- `<n> weeks`: The file is rotated every `<n>` weeks, where `<n>` is an integer greater than zero.
- `<n> months`: The file is rotated every `<n>` months, where `<n>` is an integer greater than zero and a month has 30 days.
- `<n>`: The file is rotated when its size exceeds `<n>` bytes.
- `<n> K`: The file is rotated when its size exceeds `<n>` Kilobytes.
- `<n> M`: The file is rotated when its size exceeds `<n>` Megabytes.

Important

For periodic log file rotations (daily, weekly, monthly, etc.), the date and time of log file creation or last rotation are written to the first line of the log file.

x-json-log.times

Default Value: No default value

Valid Values: true, false

Changes Take Effect: After restart

Dependencies: Used if **x-json-log** option is set

Introduced: 8.5.220.02

Enable or disable compression of archived files. Archived log files can be compressed using the gzip compression method. Compressing can be controlled with the **compress** property.

The following values are supported:

- true: Compress archived log files.
- false: Do not compress archived log files.

x-ScheduleBuilderProgressTrace

Default Value: false

Valid Values: true, false

Changes Take Effect: Immediately

Dependencies: None

Enables or disables the logging of schedule progress in the log file.

Used for troubleshooting

x-ScheduleBuilderTrace

Default Value: false

Valid Values: true, false

Changes Take Effect: Immediately

Dependencies: **x-ScheduleLogPath**

Specifies whether or not WFM Builder writes scheduling data, output data, and messages to the Schedule log file.

When this option is enabled (value is set to `true`), you must specify a Schedule log path. One log file is created for every schedule build.

Used for Schedule debugging only.

x-ScheduleLogPath

Default Value: " " (string)

Valid Values: Path to Schedule log directory

Changes Take Effect: Immediately

Dependencies: None

Specifies the output path or directory into which WFM Builder writes the file containing Schedule log data and messages, if schedule logging is enabled. If the path is not specified, WFM uses the existing executable folder.

Use this option for Schedule debugging only.

x-ScheduleMaxLogs

Default Value: 1000

Valid Values: 1 - 1000

Changes Take Effect: Immediately

Dependencies: x-ScheduleLogPath

Specifies the maximum number of Schedule log files (or pairs, if the x-SwordTrace option value is also set to yes) to retain the folder specified in the x-ScheduleLogPath option.

When the maximum number of files is reached, WFM Builder deletes files as necessary, to stay within the limit, starting with the oldest file.

x-SwordTrace

Default Value: false

Valid Values: true, false

Changes Take Effect: Immediately

Dependencies: x-ScheduleLogPath

Specifies whether or not the scheduling algorithm output messages are written to the Schedule log.

Use this option for Schedule debugging only.