



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.

# Orchestration Server Deployment Guide

Common Log Options

5/5/2025

---

## Contents

- 1 Common Log Options
  - 1.1 log Section
  - 1.2 Log Output Options
  - 1.3 Log File Extensions

# Common Log Options

Configure the common log options in the same section as the [ORS-specific log options](#).

For information on the log options listed below, see the [Framework 8.1. Configuration Options Reference Manual](#).

## log Section

This section must be called `log`.

For applications configured via a configuration file, changes to log options take effect after the application is restarted.

- buffering
- check-point
- compatible-output-priority
- expire
- keep-startup-file
- memory
- memory-storage-size
- message\_format
- messagefile
- print-attributes
- segment
- spool
- time\_convert
- time\_format
- verbose

## Log Output Options

To configure log outputs, set log level options (`all`, `alarm`, `standard`, `interaction`, `trace`, 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, to log 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.

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.

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

- `all`
- `alarm`
- `standard`
- `interaction`
- `trace`
- `debug`

## Log File Extensions

You can use the following file extensions to identify log files that an application 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 = confservlog` for Configuration Server, it prints log messages into a text file called `confservlog.<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` for Configuration Server, it prints log messages into a file called `confserv.<time_stamp>.qsp` during the time the network is not available.
- `.snapshot.log`—Assigned to files that contain the output snapshot when you configure output to a log file. The file contains the last log messages that an application generates before it terminates abnormally. For example, if you set `standard = confservlog` for Configuration Server, it prints the last log message into a file called `confserv.<time_stamp>.snapshot.log` in case of failure.

**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 = confserv` for Configuration Server, it prints the latest memory output to a file called `confserv.<time_stamp>.memory.log`.