



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 Pulse Deployment Guide

Cleanup Snapshot Data

12/18/2025

---

## Contents

- 1 Cleanup Snapshot Data
  - 1.1 Deployment
  - 1.2 How the cleanuptool works
  - 1.3 Command-line options
  - 1.4 Configuration File

# Cleanup Snapshot Data

Genesys Pulse Collector can generate large amounts of layout data snapshot files each day. These files are usually useful for only a short time period.

## Important

For the troubleshooting purposes Genesys strongly recommends to keep Genesys Pulse Collector outputs for at least recent 2-3 days.

Genesys provides a filesystem cleanup utility (cleantool), to simplify the removal of older files in data intensive environments. The cleantool is included within the Genesys Pulse Collector installation for all supported operating systems.

## Important

Beginning with Genesys Pulse Collector release 9.0.007.00, the installation package for Linux provides three executable variants, each of which uses a distinct memory allocator:

- collector.default - uses standard glibc memory allocator
- collector.jemalloc - uses jemalloc memory allocator
- collector.tcmalloc - uses tcmalloc memory allocator

The cleantool executable is built only for the standard glibc memory allocator.

## Deployment

After you install Genesys Pulse Collector, you can find the executable file — called **cleantool** in Linux or **cleantool.exe** in Windows — in the same folder that contains the Genesys Pulse Collector executable. You control this console application using a configuration file and command-line options.

Genesys recommends you run the cleantool regularly, using standard task scheduling software available within your operating system, such as Cron for Linux or Windows Task Scheduler for Windows. You must run the cleantool under a user account that has permissions to read, write, and remove files and directories on the filesystem where Genesys Pulse Collector writes the layout data snapshots.

Define your scheduling period based on data generation intensity, and your target file system capacity. Genesys recommends that you run the cleantool every hour.

## How the cleanuptool works

The cleanuptool scans for files in the specified folder path (and subfolders). It checks to see how much time has elapsed between the time when the cleanuptool started and the last time each file was modified, compares the result to values you specify in the configuration file, and preserves or removes files accordingly.

The cleanuptool also gives you the option to use *granulation points* to determine whether to preserve a file. If you do not define granulation, the tool preserves all matching files that fall into one of the active intervals. Use granulation points if you want to preserve files created at specific points within the interval. For example, you might preserve files that were generated every *N* minutes; this greatly reduces the number of files stored, while retaining a representative sample that can still be useful for troubleshooting. More information and examples are given below.

## Command-line options

Execute the cleanuptool as follows:

```
cleanuptool [options] <path> ...
```

Where **[options]** can be any of the values listed in the following table:

Command-line options		
Option	Extended usage	Definition
-c	--config-file <file-name>	Specify new configuration file to use (default is './cleanuptool.ini', if present)
-p	--preserve-last-file	Always preserve last file, even outdated
-l	--follow-symlinks	Follow symbolic links
-m	--cross-mountpoints	Cross filesystem mount point boundaries
-s	--stop-on-error	Stop processing if error occurs
-d	--dry-run	Do not perform actual file removals
-V	--verbose	Show verbose messages
-VV	--extra-verbose	Show extra verbose messages
-q	--quiet	Do not show verbose messages
-v	--version	Show version
-h	--help	Show help message

## Configuration File

The configuration file defines time intervals at which the `cleantool` preserves files. If the modification timestamp of a file matches one of the active time intervals, the file is preserved. Otherwise, it is deleted.

You can test the configuration file by running the `cleantool` in the **dry-run** mode with the **--extra-verbose** log option, which causes it to simulate a cleanup. In this mode, the tool collects and outputs information about the files that qualify to be deleted (but does not actually delete them) and displays output similar to the following:

```
Directory 'output/63/5a78424ef873-8ef1-11ec-c242-586644a1': 86 files to preserve, 6 files to remove.
```

The configuration file has the standard INI-file format: It is divided into sections, and allows comment lines that start with semicolon. A sample configuration file is provided in the Genesys Pulse Collector installation directory, and is called **cleantool.ini.sample**.

### General Section

The **general** section has two parameters:

- **active-intervals** — This required parameter lists the names of active intervals. There is no default value.
- **measure** — This optional parameter defines the time measurement units for interval points. Valid values: m, min, minutes—minutes, s, sec, seconds—seconds. The default value is minutes.

### Intervals Section

The **intervals** section defines a set of intervals to use when evaluating whether to preserve a file. You can define multiple intervals, and then specify which intervals are used:

- Populate the **active-intervals** option in the **general** section with the names of the intervals to use.
- Define each interval as a separate INI file parameter in the **Intervals** section, using the format **NAME=VALUE**, where **VALUE** is a parameter definition string with the following format:

```
[ALIGNMENT]<BEGIN>-<END>[:<GRANULARITY>]
```

Where:

- **[ALIGNMENT]**— This optional parameter defines the alignment of granulation points. By default, the intervals are aligned to granularity number. Possible values: A = aligned, or U = unaligned.
  - **<BEGIN>** — This required parameter defines the beginning of the interval, inclusive.
  - **<END>** — This required parameter defines the end of the interval, not inclusive.
- **[GRANULARITY]** — This optional parameter defines the granularity of the interval. A value of 0 (the default) means there are no granulation points. Specify a positive integer to set the spacing at which granulation points appear within the interval.

---

**Interval examples**

Interval Definition	Result
NAME=0-1	Preserve all files that were modified between 0 to 1 units before the current time.
NAME=10-20:1	From the interval between 10 and 20 units before the current time, preserve files near granulation points located at each 1 unit of time, aligned to granularity. (Alignment is not specified, so the default is used: the intervals are aligned to the granularity value). The resulting granulation points are: 10, 11, 12, 13, 14, 15, 16, 17, 18, and 19.
NAME=U10-20:3	From the interval between 10 and 20 units before the current time, preserve files near granulation points located at every 3rd unit of time. The first granulation point <i>is not</i> aligned to granularity value, so the first point is at the beginning of this interval. The resulting granulation points are: 10, 13, 16, and 19.
NAME=A20-60:7	From the interval between 10 and 60 units before the current time, preserve files near granulation points located at every 7th unit of time. The first granulation point <i>is</i> set to be aligned to the granularity value, so the first granulation point appears at the first time within the interval that is divisible by the granularity value. The resulting granulation points are: 21, 28, 35, 42, 49, and 56.