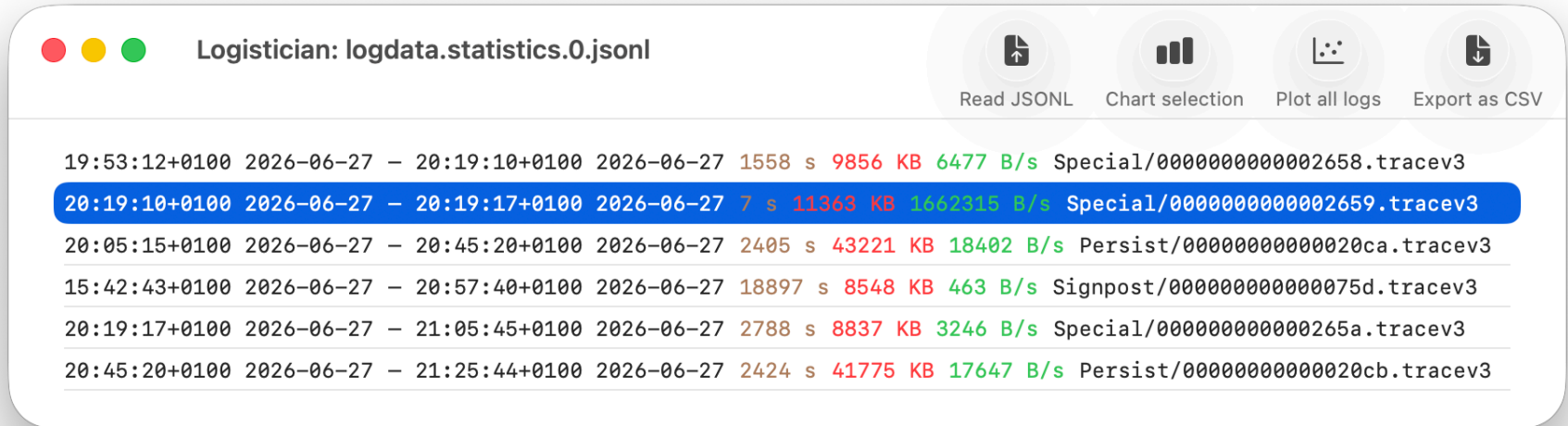


Start

→ [Start](#)



Logistician carries out basic statistical analysis on your Mac's log files to help identify periods of unusual activity that may be the result of problems. This can also help you discover why your Mac is writing excessively to its log, and how you might be able to extend the period covered by full log records.

Get started by copying those files named `logdata.statistics.n.jsonl`, where `n` is a digit, normally 0 or 1, from `/var/db/diagnostics` to a convenient folder in a location like `~/Documents` where Logistician can analyse them in safety.

→ [Log list](#)
→ [Browse details](#)

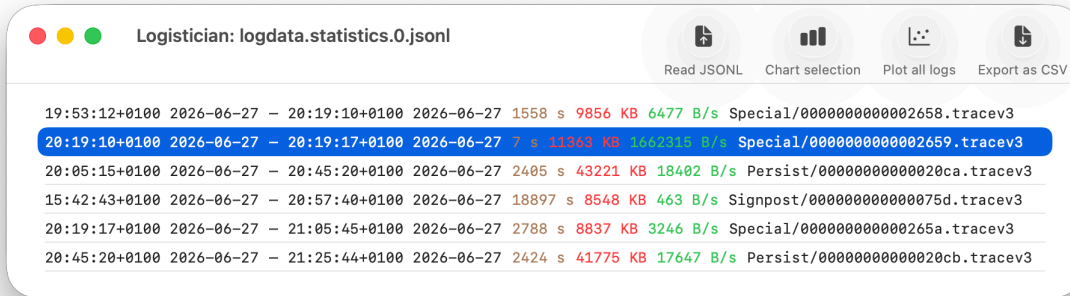
→ [Point plot](#)
→ [Export as CSV](#)

→ [Chart](#)
→ [Technical information](#)

→ [Navigation](#)

→ [Start](#)

Log list



To list the log summaries available in one of those logdata.statistics files, click the **Read JSONL** tool on the left of the three at the top right of the window. Locate, select and open that file in the dialog, and the log files analysed in that statistics file will be listed with the oldest at the top. The window title bar shows the name of the current statistics file.

For each log file listed, this shows:

- the **time and date** in UTC/GMT that log file was opened, and when it was closed;
- the **period** in seconds that file was storing log entries, in *brown*;
- the **size** of the data in that log file in KiB (when expanded), in *red*;
- the average **rate** at which log data was written to that file, in B/s (size/period), in *green*;
- the **type**, Persist, Special or Signpost, and **name** of that log file.

To analyse one file in detail, select it and click the **Chart selection** tool at the top. To see a point plot of data for all log files, click the **Plot all logs** tool at the top. To export this list, click the **Export as CSV** tool at the top right.

→ [Point plot](#) → [Chart](#) → [Navigation](#) → [Browse details](#) → [Export as CSV](#) → [Technical information](#)

→ [Start](#)

Point plot

This shows salient data for all log files analysed, in three point plots:



- **Rate** shows the average rate for Persist files only, in KiB/s.

- **Persist** shows the size of the data written by the three processes that wrote the most in that file, in KiB, for Persist files only;

- **Special** shows the size of the data written by the three processes that wrote the most in that file, in KiB, for Special files only.

Show or hide each using the checkboxes at the top.

Each plot shows data for a period of days, and you can scroll through the whole period, although the three scrollers work independently and the plots don't (yet) scroll together. Set the period shown using the slider at the foot of the plots, from a minimum of 1 day to a maximum of 14. Using a shorter period helps you identify points of interest, and by giving a more precise time for them, helps you identify which log file to chart.

→ [Log list](#)

→ [Chart](#)

→ [Navigation](#)

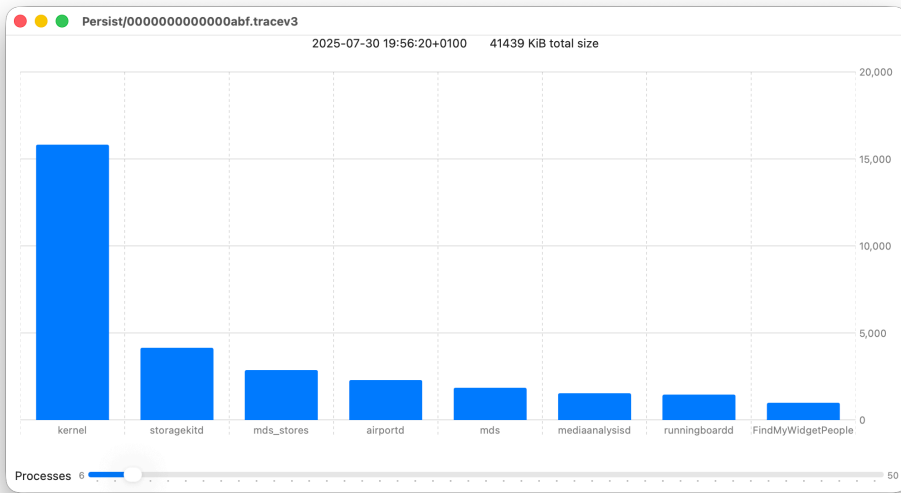
→ [Browse details](#)

→ [Export as CSV](#)

→ [Technical information](#)

→ [Start](#)

Chart



Each bar in the Chart view shows the size of log entries written by major processes and subsystems during that period. The window title bar gives the name of the log file, and below that are the time and date that log file was closed and saved, and the total size in KiB of log data written to it.

Processes and subsystems responsible for those log entries are listed along the X axis, and the height of each bar gives the size of their log data on the Y axis, again given in KiB.

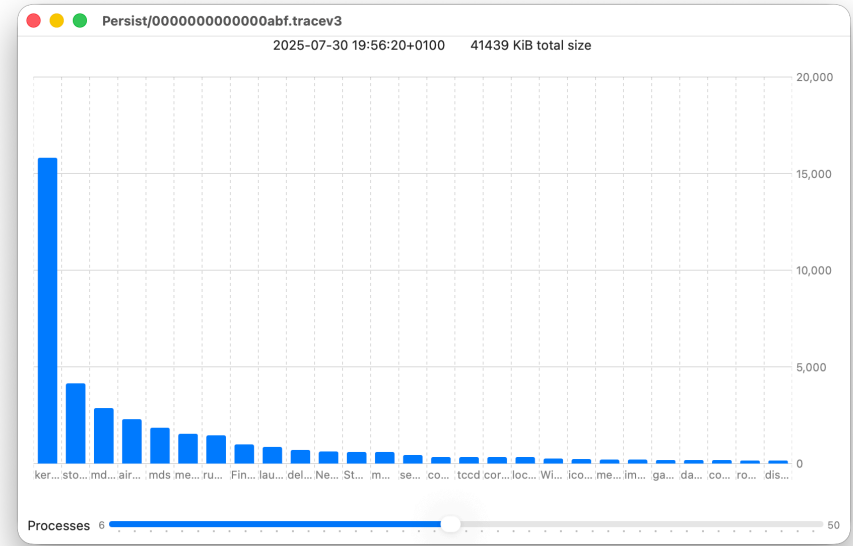
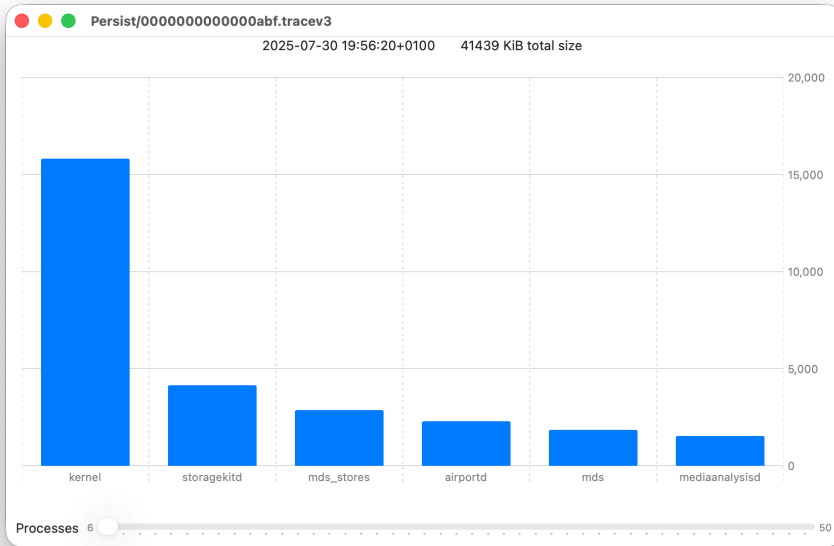
At the foot of the chart is a slider to control the number of bars to show, from 6 to 50. You can also scroll along the bars.

To view data for a different log file, simply select that file in the list and click on the **Chart selection** tool again.

→ [Log list](#) → [Point plot](#) → [Navigation](#) → [Browse details](#) → [Export as CSV](#) → [Technical information](#)

→ [Start](#)

Navigation

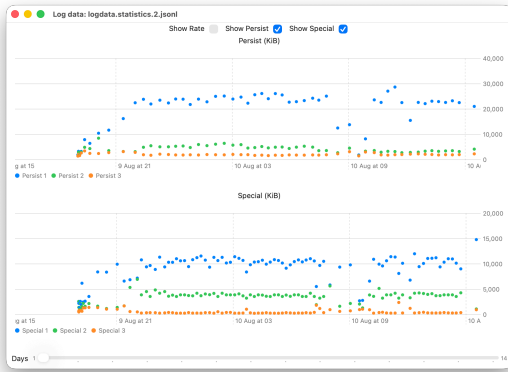


The slider below the bar chart sets how many bars to show, from 6 (above left) to 50, and you can scroll through all the bars for that log file. The fewer the bars, the easier it is to read their process names at the foot.

→ [Log list](#) → [Point plot](#) → [Chart](#) → [Browse details](#) → [Export as CSV](#) → [Technical information](#)

→ [Start](#)

Browse details



Logistician doesn't currently provide a direct way to browse log entries for periods of interest, but it gives you the information needed to do so.

For example, say you notice a high point of nearly 30,000 KiB in the Point Plot for Persist files. Setting the period to 1 day suggests it has a timestamp of about 11:00:00 on 10 Aug. Looking through the log list, that corresponds to a Persist file that was only open for 35 seconds, from 11:04:43 on that date, and closed at 11:05:18. Selecting that file and charting it confirms it has a lot of data written by the `cloudd` process that you want to investigate further. Use that time as your starting point for browsing the log using [LogUI](#).

Open LogUI and first check whether its log records cover that time by opening its **Diagnostics Tool** from the **Window** menu. Click the **Get Info** tool in that window and select the `/var/db/diagnostics` folder. That will report the oldest Persist log entry available, and provided that's older than the Persist file you want to view, its log entries should still be accessible.

In LogUI's main browser window, set the **Start** date and time to the timestamp shown for the start of that Persist file, and a **Period** in seconds that lies within the time between then and 11:05:18, the period of interest. Click on the **Get Log** tool there to obtain log entries for that set period.

One caution to be aware of: starting timestamps are taken from the closing timestamp of the previous log file. If that older file was closed because the Mac was shut down, then there will be no log entries until the Mac starts up again, which may be long after the starting timestamp given.

→ [Log list](#) → [Point plot](#) → [Chart](#) → [Navigation](#) → [Export as CSV](#) → [Technical information](#)

→ [Start](#)

Export as CSV

To export all the data from the main window's listing of log files, click on the Export as CSV tool at the top right. This exports the following fields:

- **enddate** – the date and time when that log file was closed, in `dd/mm/yyyy hh:mm:ss` format
- **middate** – the date and time of the middle of the period that log file was open, in the same format
- **duration** – the time in seconds that the log file was open for writing new entries
- **size** – the size in KiB of the uncompressed data in that log file
- **rate** – the average rate in bytes/second of uncompressed log data written to that log file
- **file** – the type and name of that log file.

These are all identical to those shown in the window, with the exception of **middate**. This is to enable you to plot data against the middle of the period in which that log file was open, without having to perform date and time calculations in a spreadsheet or other app. It's simply the **enddate** less half the duration in seconds.

→ [Log list](#) → [Point plot](#) → [Chart](#) → [Navigation](#) → [Browse details](#) → [Technical information](#)

→ [Start](#)

Technical Information

Each time `logd` closes a `tracev3` log file and opens a new one, it writes a summary of the data in the file it's closing to `logdata.statistics` files alongside the folders containing log files. There are two sets, one in plain text for access in a text editor, the other in JSON Lines format and going back further in time. Although full log files may be aged out after just a few days, summary data in the JSON files persists for several weeks or more.

Persist and **Special** log files contain different types of log entry. Persist files include 'core' entries from major processes, and are removed completely when `logd` needs to recover the space. Special files include larger and more unusual entries, and their contents are weeded progressively by `logd` as they age. **Signpost** logs contain different entries again, primarily intended for assessing performance.

Logistician opens those JSON records and displays their contents using SwiftUI and Swift Charts. Currently it only supports one Log list window, and single Chart and Plot windows, and has basic features for data display.

All data shown are obtained from those `logdata.statistics` files, and no use is made of `tracev3` log files. Timestamps are currently set in UTC/GMT for consistency rather than convenience.

Change list

- 1.4 (23):
 - enhanced chart and plot views with window slider and scrolling.
- 1.3 (19):
 - added Export as CSV.
- 1.2 (16):
 - fixed two more divide by zero crashes.
- 1.1 (14):
 - fixed a divide by zero crash (thank you, Jake).
- 1.1 (12):
 - added Plot view
 - additional data in log list.
- 1.0 (7):
 - first release.

3 September 2026.