sd_journal_get_seqnum

- Description
- Return Value
- Notes
- History
- See Also

systemd

sd_journal_get_seqnum

3

sd_journal_get_seqnum

Read sequence number from the current journal entry

#include <systemd/sd-journal.h>

int

sd_journal_get_seqnum

sd_journal *

j

uint64_t *

ret_seqnum

sd_id128_t *

ret_seqnum_id

Description

`sd_journal_get_seqnum()` returns the sequence number of the current journal entry. It takes three arguments: the journal context object, a pointer to a 64-bit unsigned integer to store the sequence number in, and a buffer to return the 128-bit sequence number ID in.

When writing journal entries to disk each `systemd-journald` instance will number them sequentially, starting from 1 for the first entry written after subsystem initialization. Each such series of sequence numbers is associated with a 128-bit sequence number ID which is initialized randomly, once at `systemd-journal` initialization. Thus, while multiple instances of `systemd-journald` will assign the same sequence numbers to their written journal entries, they will have a distinct sequence number IDs. The sequence number is assigned at the moment of writing the entry to disk. If log entries are rewritten (for example because the volatile logs from `/run/log/` are flushed to `/var/log/` via `systemd-journald-flush.service`) they will get new sequence numbers assigned.

Sequence numbers may be used to order entries (entries associated with the same sequence number ID and lower sequence numbers should be ordered chronologically before those with higher sequence numbers), and to detect lost entries. Note that journal service instances typically write to multiple journal files in parallel (for example because `SplitMode=` is used), in which case each journal file will only contain a subset of the sequence numbers. To recover the full stream of journal entries the files must be combined ("interleaved"), a process that primarily relies on the sequence numbers. When journal files are rotated (due to size or time limits), the series of sequence numbers is continued in the replacement files. All journal files generated from the same journal instance will carry the same sequence number ID.

As the sequence numbers are assigned at the moment of writing the journal entries to disk they do not exist if storage is disabled via `SplitMode=`.

The `ret_seqnum` and `ret_seqnum_id` parameters may be specified as `NULL` in which case the relevant data is not returned (but the call will otherwise succeed).

Note that these functions will not work before `sd_journal_next(3)` (or related call) has been called at least once, in order to position the read pointer at a valid entry.

Return Value

`sd_journal_get_seqnum()` returns 0 on success or a negative errno-style error code..

Notes

History

`sd_journal_get_seqnum()` was added in version 254.

See Also

`systemd(1)`, `sd-journal(3)`, `sd_journal_open(3)`, `sd_journal_next(3)`, `sd_journal_get_data(3)`, `sd_journal_get_monotonic_usec(3)`

□□□ #2
□ □□□ □□□ 2026-10-31 14:57:53 UTC
□ □□□ □□□ 2026-10-31 14:57:54 UTC