NAME

systemd.journal-fields − Special journal fields

DESCRIPTION

Entries in the journal resemble an environment block in their syntax but with fields that can include binary data. Primarily, fields are formatted UTF−8 text strings, and binary formatting is used only where formatting as UTF−8 text strings makes little sense. New fields may freely be defined by applications, but a few fields have special meaning. All fields with special meanings are optional. In some cases, fields may appear more than once per entry.

USER JOURNAL FIELDS

User fields are fields that are directly passed from clients and stored in the journal.

MESSAGE=

The human−readable message string for this entry. This is supposed to be the primary text shown to the user. It is usually not translated (but might be in some cases), and is not supposed to be parsed for metadata.

MESSAGE_ID=

A 128−bit message identifier ID for recognizing certain message types, if this is desirable. This should contain a 128−bit ID formatted as a lower−case hexadecimal string, without any separating dashes or suchlike. This is recommended to be a UUID−compatible ID, but this is not enforced, and formatted differently. Developers can generate a new ID for this purpose with **systemd−id128 new**.

PRIORITY=

A priority value between 0 ("emerg") and 7 ("debug") formatted as a decimal string. This field is compatible with syslog's priority concept.

CODE_FILE=, *CODE_LINE=*, *CODE_FUNC=*

The code location generating this message, if known. Contains the source filename, the line number and the function name.

ERRNO=

The low−level Unix error number causing this entry, if any. Contains the numeric value of **errno**(3) formatted as a decimal string.

SYSLOG_FACILITY=, *SYSLOG_IDENTIFIER=*, *SYSLOG_PID=*, *SYSLOG_TIMESTAMP=*

Syslog compatibility fields containing the facility (formatted as decimal string), the identifier string (i.e. "tag"), the client PID, and the timestamp as specified in the original datagram. (Note that the tag is usually derived from glibc's *program_invocation_short_name* variable, see **program_invocation_short_name**(3).)

SYSLOG_RAW=

The original contents of the syslog line as received in the syslog datagram. This field is only included if the *MESSAGE=* field was modified compared to the original payload or the timestamp could not be located properly and is not included in *SYSLOG_TIMESTAMP=*. Message truncation occurs when when the message contains leading or trailing whitespace (trailing and leading whitespace is stripped), or it contains an embedded **NUL** byte (the **NUL** byte and anything after it is not included). Thus, the original syslog line is either stored as *SYSLOG_RAW=* or it can be recreated based on the stored priority and facility, timestamp, identifier, and the message payload in *MESSAGE=*.

TRUSTED JOURNAL FIELDS

Fields prefixed with an underscore are trusted fields, i.e. fields that are implicitly added by the journal and cannot be altered by client code.

_PID=, *_UID=*, *_GID=*

The process, user, and group ID of the process the journal entry originates from formatted as a decimal string. Note that entries obtained via "stdout" or "stderr" of forked processes will contain credentials valid for a parent process (that initiated the connection to **systemd−journald**).

_COMM=, *_EXE=*, *_CMDLINE=*

The name, the executable path, and the command line of the process the journal entry originates from.

_CAP_EFFECTIVE=
    The effective **capabilities**(7) of the process the journal entry originates from.

_AUDIT_SESSION=, *_AUDIT_LOGINUID=*
    The session and login UID of the process the journal entry originates from, as maintained by the
    kernel audit subsystem.

_SYSTEMD_CGROUP=, *_SYSTEMD_SLICE=*, *_SYSTEMD_UNIT=*, *_SYSTEMD_USER_UNIT=*,
_SYSTEMD_SESSION=, *_SYSTEMD_OWNER_UID=*
    The control group path in the systemd hierarchy, the the systemd slice unit name, the systemd unit
    name, the unit name in the systemd user manager (if any), the systemd session ID (if any), and the
    owner UID of the systemd user unit or systemd session (if any) of the process the journal entry
    originates from.

_SELINUX_CONTEXT=
    The SELinux security context (label) of the process the journal entry originates from.

_SOURCE_REALTIME_TIMESTAMP=
    The earliest trusted timestamp of the message, if any is known that is different from the reception time
    of the journal. This is the time in microseconds since the epoch UTC, formatted as a decimal string.

_BOOT_ID=
    The kernel boot ID for the boot the message was generated in, formatted as a 128−bit hexadecimal
    string.

_MACHINE_ID=
    The machine ID of the originating host, as available in **machine-id**(5).

_SYSTEMD_INVOCATION_ID=
    The invocation ID for the runtime cycle of the unit the message was generated in, as available to
    processes of the unit in *$INVOCATION_ID* (see **systemd.exec**(5)).

_HOSTNAME=
    The name of the originating host.

_TRANSPORT=
    How the entry was received by the journal service. Valid transports are:

    audit
        for those read from the kernel audit subsystem

    driver
        for internally generated messages

    syslog
        for those received via the local syslog socket with the syslog protocol

    journal
        for those received via the native journal protocol

    stdout
        for those read from a service's standard output or error output

    kernel
        for those read from the kernel

_STREAM_ID=
    Only applies to "_TRANSPORT=stdout" records: specifies a randomized 128bit ID assigned to the
    stream connection when it was first created. This ID is useful to reconstruct individual log streams
    from the log records: all log records carrying the same stream ID originate from the same stream.

_LINE_BREAK=
    Only applies to "_TRANSPORT=stdout" records: indicates that the log message in the standard
    output/error stream was not terminated with a normal newline character ("\n", i.e. ASCII 10).
    Specifically, when set this field is one of **nul** (in case the line was terminated by a NUL byte),

line−max (in case the maximum log line length was reached, as configured with *LineMax=* in **journald.conf**(5)) or **eof** (if this was the last log record of a stream and the stream ended without a final newline character). Note that this record is not generated when a normal newline character was used for marking the log line end.

KERNEL JOURNAL FIELDS

Kernel fields are fields that are used by messages originating in the kernel and stored in the journal.

_KERNEL_DEVICE=
The kernel device name. If the entry is associated to a block device, the major and minor of the device node, separated by ":" and prefixed by "b". Similar for character devices but prefixed by "c". For network devices, this is the interface index prefixed by "n". For all other devices, this is the subsystem name prefixed by "+", followed by ":", followed by the kernel device name.

_KERNEL_SUBSYSTEM=
The kernel subsystem name.

_UDEV_SYSNAME=
The kernel device name as it shows up in the device tree below /sys.

_UDEV_DEVNODE=
The device node path of this device in /dev.

_UDEV_DEVLINK=
Additional symlink names pointing to the device node in /dev. This field is frequently set more than once per entry.

FIELDS TO LOG ON BEHALF OF A DIFFERENT PROGRAM

Fields in this section are used by programs to specify that they are logging on behalf of another program or unit.

Fields used by the **systemd−coredump** coredump kernel helper:

COREDUMP_UNIT=, *COREDUMP_USER_UNIT=*
Used to annotate messages containing coredumps from system and session units. See **coredumpctl**(1).

Privileged programs (currently UID 0) may attach *OBJECT_PID=* to a message. This will instruct **systemd−journald** to attach additional fields on behalf of the caller:

OBJECT_PID=PID
PID of the program that this message pertains to.

OBJECT_UID=, *OBJECT_GID=*, *OBJECT_COMM=*, *OBJECT_EXE=*, *OBJECT_CMDLINE=*, *OBJECT_AUDIT_SESSION=*, *OBJECT_AUDIT_LOGINUID=*, *OBJECT_SYSTEMD_CGROUP=*, *OBJECT_SYSTEMD_SESSION=*, *OBJECT_SYSTEMD_OWNER_UID=*, *OBJECT_SYSTEMD_UNIT=*, *OBJECT_SYSTEMD_USER_UNIT=*
These are additional fields added automatically by **systemd−journald**. Their meaning is the same as *_UID=*, *_GID=*, *_COMM=*, *_EXE=*, *_CMDLINE=*, *_AUDIT_SESSION=*, *_AUDIT_LOGINUID=*, *_SYSTEMD_CGROUP=*, *_SYSTEMD_SESSION=*, *_SYSTEMD_UNIT=*, *_SYSTEMD_USER_UNIT=*, and *_SYSTEMD_OWNER_UID=* as described above, except that the process identified by *PID* is described, instead of the process which logged the message.

ADDRESS FIELDS

During serialization into external formats, such as the **Journal Export Format**[1] or the **Journal JSON Format**[2], the addresses of journal entries are serialized into fields prefixed with double underscores. Note that these are not proper fields when stored in the journal but for addressing metadata of entries. They cannot be written as part of structured log entries via calls such as **sd_journal_send**(3). They may also not be used as matches for **sd_journal_add_match**(3)

__CURSOR=
The cursor for the entry. A cursor is an opaque text string that uniquely describes the position of an entry in the journal and is portable across machines, platforms and journal files.

__REALTIME_TIMESTAMP=
>   The wallclock time (**CLOCK_REALTIME**) at the point in time the entry was received by the journal, in microseconds since the epoch UTC, formatted as a decimal string. This has different properties from "_SOURCE_REALTIME_TIMESTAMP=", as it is usually a bit later but more likely to be monotonic.

__MONOTONIC_TIMESTAMP=
>   The monotonic time (**CLOCK_MONOTONIC**) at the point in time the entry was received by the journal in microseconds, formatted as a decimal string. To be useful as an address for the entry, this should be combined with the boot ID in "_BOOT_ID=".

SEE ALSO
>   **systemd**(1), **journalctl**(1), **journald.conf**(5), **sd-journal**(3), **coredumpctl**(1), **systemd.directives**(7)

NOTES
>   1.   Journal Export Format
>   https://www.freedesktop.org/wiki/Software/systemd/export
>
>   2.   Journal JSON Format
>   https://www.freedesktop.org/wiki/Software/systemd/json