Environment

`$SYSTEMD_LOG_LEVEL`
The maximum log level of emitted messages (messages with a higher log level, i.e. less important ones, will be suppressed). Takes a comma-separated list of values. A value may be either one of (in order of decreasing importance) `emerg`, `alert`, `crit`, `err`, `warning`, `notice`, `info`, `debug`, or an integer in the range 0...7. See `syslog(3)` for more information. Each value may optionally be prefixed with one of `console`, `syslog`, `kmsg` or `journal` followed by a colon to set the maximum log level for that specific log target (e.g. `SYSTEMD_LOG_LEVEL=debug,console:info` specifies to log at debug level except when logging to the console which should be at info level). Note that the global maximum log level takes priority over any per target maximum log levels.

`$SYSTEMD_LOG_COLOR`
A boolean. If true, messages written to the tty will be colored according to priority.

This setting is only useful when messages are written directly to the terminal, because `journalctl(1)` and other tools that display logs will color messages based on the log level on their own.

`$SYSTEMD_LOG_TIME`
A boolean. If true, console log messages will be prefixed with a timestamp.

This setting is only useful when messages are written directly to the terminal or a file, because `journalctl(1)` and other tools that display logs will attach timestamps based on the entry metadata on their own.

`$SYSTEMD_LOG_LOCATION`
A boolean. If true, messages will be prefixed with a filename and line number in the source code where the message originates.

Note that the log location is often attached as metadata to journal entries anyway. Including it directly in the message text can nevertheless be convenient when debugging programs.

`$SYSTEMD_LOG_TID`
A boolean. If true, messages will be prefixed with the current numerical thread ID (TID).

Note that the this information is attached as metadata to journal entries anyway. Including it directly in the message text can nevertheless be convenient when debugging programs.

`$SYSTEMD_LOG_TARGET`
The destination for log messages. One of `console` (log to the attached tty), `console-prefixed` (log to the attached tty but with prefixes encoding the log level and "facility", see `syslog(3)`, `kmsg` (log to the kernel circular log buffer), `journal` (log to the journal), `journal-or-kmsg` (log to the journal if

available, and to kmsg otherwise), `auto` (determine the appropriate log target automatically, the default), `null` (disable log output).

`$SYSTEMD_LOG_RATELIMIT_KMSG`
Whether to ratelimit kmsg or not. Takes a boolean. Defaults to `true`. If disabled, systemd will not ratelimit messages written to kmsg.

`$SYSTEMD_LOG_EXECUTABLE`
A boolean. If true, messages will be prefixed with the current executable name.

Note that this information is attached as metadata to journal entries anyway. Including it directly in the message text can nevertheless be convenient when debugging programs.

`$SYSTEMD_PAGER`; `$PAGER`
Pager to use when `--no-pager` is not given. `$SYSTEMD_PAGER` is used if set; otherwise `$PAGER` is used. If neither `$SYSTEMD_PAGER` nor `$PAGER` are set, a set of well-known pager implementations is tried in turn, including `less(1)` and `more(1)`, until one is found. If no pager implementation is discovered, no pager is invoked. Setting those environment variables to an empty string or the value `cat` is equivalent to passing `--no-pager`.

Note: if `$SYSTEMD_PAGERSECURE` is not set, `$SYSTEMD_PAGER` and `$PAGER` can only be used to disable the pager (with `cat` or ``), and are otherwise ignored.

`$SYSTEMD_LESS`
Override the options passed to `less` (by default `FRSXMK`).

Users might want to change two options in particular:

`K`
This option instructs the pager to exit immediately when Ctrl+C is pressed. To allow `less` to handle Ctrl+C itself to switch back to the pager command prompt, unset this option.

If the value of `$SYSTEMD_LESS` does not include `K`, and the pager that is invoked is `less`, Ctrl+C will be ignored by the executable, and needs to be handled by the pager.

`X`
This option instructs the pager to not send termcap initialization and deinitialization strings to the terminal. It is set by default to allow command output to remain visible in the terminal even after the pager exits. Nevertheless, this prevents some pager functionality from working, in particular paged output cannot be scrolled with the mouse.

Note that setting the regular `$LESS` environment variable has no effect for `less` invocations by systemd tools.

See `less(1)` for more discussion.

`$SYSTEMD_LESSCHARSET`
Override the charset passed to `less` (by default `utf-8`, if the invoking terminal is determined to be UTF-8 compatible).

Note that setting the regular `$LESSCHARSET` environment variable has no effect for `less` invocations by systemd tools.

`$SYSTEMD_PAGERSECURE`
Common pager commands like `less(1)`, in addition to "paging", i.e. scrolling through the output, support opening of or writing to other files and running arbitrary shell commands. When commands are invoked with elevated privileges, for example under `sudo(8)` or `pkexec(1)`, the pager becomes a security boundary. Care must be taken that only programs with strictly limited functionality are used as pagers, and unintended interactive features like opening or creation of new files or starting of subprocesses are not allowed. "Secure mode" for the pager may be enabled as described below, *if the pager supports that* (most pagers are not written in a way that takes this into consideration). It is recommended to either explicitly enable "secure mode" or to completely disable the pager using `--no-pager` or `PAGER=cat` when allowing untrusted users to execute commands with elevated privileges.

This option takes a boolean argument. When set to true, the "secure mode" of the pager is enabled. In "secure mode", `LESSSECURE=1` will be set when invoking the pager, which instructs the pager to disable commands that open or create new files or start new subprocesses. Currently only `less(1)` is known to understand this variable and implement "secure mode".

When set to false, no limitation is placed on the pager. Setting `SYSTEMD_PAGERSECURE=0` or not removing it from the inherited environment may allow the user to invoke arbitrary commands.

When `$SYSTEMD_PAGERSECURE` is not set, systemd tools attempt to automatically figure out if "secure mode" should be enabled and whether the pager supports it. "Secure mode" is enabled if the effective UID is not the same as the owner of the login session, see `geteuid(2)` and `sd_pid_get_owner_uid(3)`, or when running under `sudo(8)` or similar tools (`$SUDO_UID` is set [^1]). In those cases, `SYSTEMD_PAGERSECURE=1` will be set and pagers which are not known to implement "secure mode" will not be used at all. Note that this autodetection only covers the most common mechanisms to elevate privileges and is intended as convenience. It is recommended to explicitly set `$SYSTEMD_PAGERSECURE` or disable the pager.

Note that if the `$SYSTEMD_PAGER` or `$PAGER` variables are to be honoured, other than to disable the pager, `$SYSTEMD_PAGERSECURE` must be set too.

`$SYSTEMD_COLORS`
Takes a boolean argument, or a special value. By default (unset), `systemd` and related utilities will use colors in their output if possible. If `$COLORTERM` is set to `truecolor` or `24bit`, 24-bit colors will be enabled, 256 colors otherwise, unless `$NO_COLOR` or `$TERM` indicates colors are disabled.

`true`
Same as unset, except that `$NO_COLOR` is ignored.

`false`
The output will be monochrome.

`16`; `256`; `24bit`
Always use the base 16 ANSI colors, 256 colors, or 24 bit color, respectively.

`auto-16`; `auto-256`; `auto-24bit`
Use the given quantity of colours, subject to `$TERM`, and what the console is connected to.

`$SYSTEMD_URLIFY`
The value must be a boolean. Controls whether clickable links should be generated in the output for terminal emulators supporting this. This can be specified to override the decision that `systemd` makes based on `$TERM` and other conditions.

[^1]: It is recommended for other tools to set and check `$SUDO_UID` as appropriate, treating it is a common interface.

□□□ #2
□ □□□ □□□ 2026-10-31 14:46:19 UTC
□ □□□ □□□ 2026-10-31 14:46:20 UTC