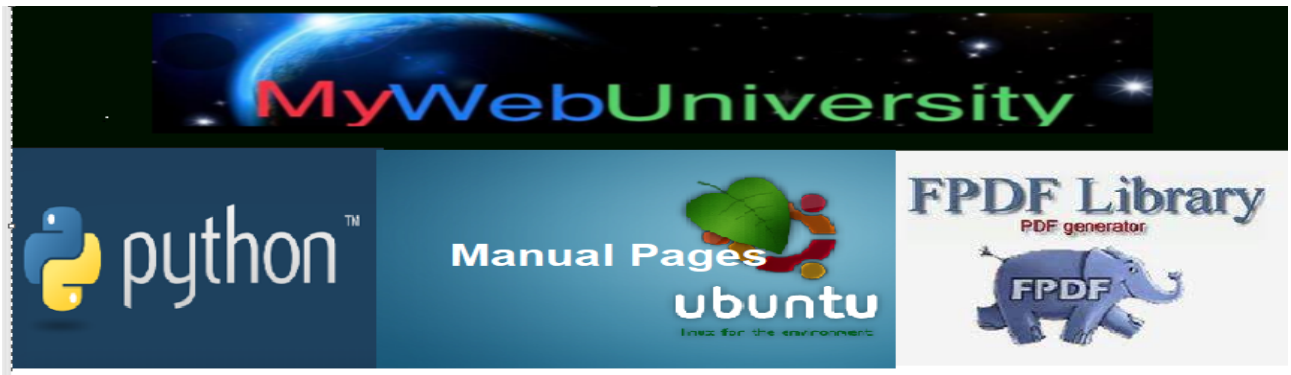

Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'loginctl.1'

$ man loginctl.1

LOGINCTL(1) loginctl LOGINCTL(1)

NAME

loginctl - Control the systemd login manager

SYNOPSIS

loginctl [OPTIONS...] {COMMAND} [NAME...]

DESCRIPTION

loginctl may be used to introspect and control the state of the systemd(1) login manager systemd-logind.service(8).

COMMANDS

The following commands are understood:

Session Commands

list-sessions

List current sessions. The JSON format output can be toggled using --json= or -j option.

session-status [ID...]

Show terse runtime status information about one or more sessions, followed by the most recent log data from the journal. Takes one or more session identifiers as parameters. If no session identifiers are passed, the status of the caller's session is shown. This function is intended to generate human-readable output. If you are looking for computer-parsable output, use show-session instead.

Added in version 233.

show-session [ID...]

Show properties of one or more sessions or the manager itself. If no argument is specified, properties of the manager will be shown. If a session ID is specified,

properties of the session are shown. Specially, if the given ID is "self", the session to which the loginctl process belongs is used. If "auto", the current session is used as with "self" if exists, and falls back to the current user's graphical session. By default, empty properties are suppressed. Use --all to show those too. To select specific properties to show, use --property=. This command is intended to be used whenever computer-parsable output is required. Use session-status if you are looking for formatted human-readable output.

Added in version 233.

activate [ID]

Activate a session. This brings a session into the foreground if another session is currently in the foreground on the respective seat. Takes a session identifier as argument. If no argument is specified, the session of the caller is put into foreground.

Added in version 219.

lock-session [ID...], unlock-session [ID...]

Activates/deactivates the screen lock on one or more sessions, if the session supports it. Takes one or more session identifiers as arguments. If no argument is specified, the session of the caller is locked/unlocked.

Added in version 233.

lock-sessions, unlock-sessions

Activates/deactivates the screen lock on all current sessions supporting it.

Added in version 188.

terminate-session ID...

Terminates a session. This kills all processes of the session and deallocates all resources attached to the session. If the argument is specified as empty string the session invoking the command is terminated.

Added in version 233.

kill-session ID...

Send a signal to one or more processes of the session. Use --kill-whom= to select which process to kill. Use --signal= to select the signal to send. If the argument is specified as empty string the signal is sent to the session invoking the command.

Added in version 233.

User Commands

list-users

List currently logged in users. The JSON format output can be toggled using --json= or -j option.

user-status [USER...]

Show terse runtime status information about one or more logged in users, followed by the most recent log data from the journal. Takes one or more user names or numeric user IDs as parameters. If no parameters are passed, the status is shown for the user of the session of the caller. This function is intended to generate human-readable output. If you are looking for computer-parsable output, use show-user instead.

Added in version 233.

show-user [USER...]

Show properties of one or more users or the manager itself. If no argument is specified, properties of the manager will be shown. If a user is specified, properties of the user are shown. By default, empty properties are suppressed. Use --all to show those too. To select specific properties to show, use --property=. This command is intended to be used whenever computer-parsable output is required. Use user-status if you are looking for formatted human-readable output.

Added in version 233.

enable-linger [USER...], disable-linger [USER...]

Enable/disable user lingering for one or more users. If enabled for a specific user, a user manager is spawned for the user at boot and kept around after logouts. This allows users who are not logged in to run long-running services. Takes one or more user names or numeric UIDs as argument. If no argument is specified, enables/disables lingering for the user of the session of the caller.

See also KillUserProcesses= setting in logind.conf(5).

Added in version 233.

terminate-user USER...

Terminates all sessions of a user. This kills all processes of all sessions of the user and deallocates all runtime resources attached to the user. If the argument is specified as empty string the sessions of the user invoking the command are terminated.

Added in version 233.

kill-user USER...

Send a signal to all processes of a user. Use --signal= to select the signal to send. If the argument is specified as empty string the signal is sent to the sessions of the user

invoking the command.

Added in version 233.

Seat Commands

list-seats

List currently available seats on the local system. The JSON format output can be toggled using --json= or -j option.

seat-status [NAME...]

Show terse runtime status information about one or more seats. Takes one or more seat names as parameters. If no seat names are passed the status of the caller's session's seat is shown. This function is intended to generate human-readable output. If you are looking for computer-parsable output, use show-seat instead.

Added in version 233.

show-seat [NAME...]

Show properties of one or more seats or the manager itself. If no argument is specified, properties of the manager will be shown. If a seat is specified, properties of the seat are shown. By default, empty properties are suppressed. Use --all to show those too. To select specific properties to show, use --property=. This command is intended to be used whenever computer-parsable output is required. Use seat-status if you are looking for formatted human-readable output.

Added in version 233.

attach NAME DEVICE...

Persistently attach one or more devices to a seat. The devices should be specified via device paths in the /sys/ file system. To create a new seat, attach at least one graphics card to a previously unused seat name. Seat names may consist only of a?z, A?Z, 0?9, "-" and "_" and must be prefixed with "seat". To drop assignment of a device to a specific seat, just reassign it to a different seat, or use flush-devices.

Added in version 233.

flush-devices

Removes all device assignments previously created with attach. After this call, only automatically generated seats will remain, and all seat hardware is assigned to them.

terminate-seat NAME...

Terminates all sessions on a seat. This kills all processes of all sessions on the seat and deallocates all runtime resources attached to them.

Added in version 233.

OPTIONS

The following options are understood:

-p, --property=

When showing session/user/seat properties, limit display to certain properties as specified as argument. If not specified, all set properties are shown. The argument should be a property name, such as "Sessions". If specified more than once, all properties with the specified names are shown.

--value

When showing session/user/seat properties, only print the value, and skip the property name and "=".

Added in version 230.

-a, --all

When showing session/user/seat properties, show all properties regardless of whether they are set or not.

-l, --full

Do not ellipsize process tree entries.

Added in version 198.

--kill-whom=

When used with kill-session, choose which processes to kill. Takes one of "leader" or "all", to select whether to kill only the leader process of the session or all processes of the session. If omitted, defaults to all.

Added in version 252.

-s, --signal=

When used with kill-session or kill-user, choose which signal to send to selected processes. Must be one of the well known signal specifiers, such as SIGTERM, SIGINT or SIGSTOP. If omitted, defaults to SIGTERM.

The special value "help" will list the known values and the program will exit immediately, and the special value "list" will list known values along with the numerical signal numbers and the program will exit immediately.

-n, --lines=

When used with user-status and session-status, controls the number of journal lines to show, counting from the most recent ones. Takes a positive integer argument. Defaults to

10.

Added in version 219.

-o, --output=

When used with user-status and session-status, controls the formatting of the journal entries that are shown. For the available choices, see journalctl(1). Defaults to "short".

Added in version 219.

-H, --host=

Execute the operation remotely. Specify a hostname, or a username and hostname separated by "@", to connect to. The hostname may optionally be suffixed by a port ssh is listening on, separated by ":", and then a container name, separated by "/", which connects directly to a specific container on the specified host. This will use SSH to talk to the remote machine manager instance. Container names may be enumerated with machinectl -H HOST. Put IPv6 addresses in brackets.

-M, --machine=

Execute operation on a local container. Specify a container name to connect to, optionally prefixed by a user name to connect as and a separating "@" character. If the special string ".host" is used in place of the container name, a connection to the local system is made (which is useful to connect to a specific user's user bus: "--user --machine=lennart@.host"). If the "@" syntax is not used, the connection is made as root user. If the "@" syntax is used either the left hand side or the right hand side may be omitted (but not both) in which case the local user name and ".host" are implied.

--no-ask-password

Do not query the user for authentication for privileged operations.

--no-pager

Do not pipe output into a pager.

--no-legend

Do not print the legend, i.e. column headers and the footer with hints.

--json=MODE

Shows output formatted as JSON. Expects one of "short" (for the shortest possible output without any redundant whitespace or line breaks), "pretty" (for a pretty version of the same, with indentation and line breaks) or "off" (to turn off JSON output, the default).

-j

Equivalent to --json=pretty if running on a terminal, and --json=short otherwise.

-h, --help

Print a short help text and exit.

--version

Print a short version string and exit.

EXIT STATUS

On success, 0 is returned, a non-zero failure code otherwise.

EXAMPLES

Example 1. Querying user status

```
$ loginctl user-status
fatima (1005)
           Since: Sat 2016-04-09 14:23:31 EDT; 54min ago
           State: active
        Sessions: 5 *3
          Unit: user-1005.slice
                ??user@1005.service
                  ...
                ??session-3.scope
                  ...
                ??session-5.scope
                  ??3473 login -- fatima
                  ??3515 -zsh
Apr 09 14:40:30 laptop login[2325]: pam_unix(login:session):
                    session opened for user fatima by LOGIN(uid=0)
Apr 09 14:40:30 laptop login[2325]: LOGIN ON tty3 BY fatima
```

There are two sessions, 3 and 5. Session 3 is a graphical session, marked with a star. The tree of processing including the two corresponding scope units and the user manager unit are shown.

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_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. When true, systemd and related utilities will use colors in their output, otherwise the output will be monochrome. Additionally, the variable can take one of the following special values: "16", "256" to restrict the use of colors to the base 16 or 256 ANSI colors, respectively. This can be specified to override the automatic decision based on $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.

SEE ALSO

systemd(1), systemctl(1), systemd-logind.service(8), logind.conf(5)

NOTES

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