



Rocky Enterprise Linux 9.2 Manual Pages on command 'kernel-command-line.7'

\$ man kernel-command-line.7

KERNEL-COMMAND-LINE(7) kernel-command-line KERNEL-COMMAND-LINE(7)

NAME

kernel-command-line - Kernel command line parameters

SYNOPSIS

/proc/cmdline

DESCRIPTION

The kernel, the programs running in the initrd and in the host system may be configured at boot via kernel command line arguments. In addition, various systemd tools look at the EFI variable "SystemdOptions" (if available). Both sources are combined, but the kernel command line has higher priority. Please note that the EFI variable is only used by systemd tools, and is ignored by the kernel and other user space tools, so it is not a replacement for the kernel command line.

For command line parameters understood by the kernel, please see [kernel-parameters.html\[1\]](#) and [bootparam\(7\)](#).

For command line parameters understood by the initrd, see [dracut.cmdline\(7\)](#), or the documentation of the specific initrd

implementation of your installation.

CORE OS COMMAND LINE ARGUMENTS

systemd.unit=, rd.systemd.unit=, systemd.dump_core,
systemd.early_core_pattern=, systemd.crash_chvt, systemd.crash_shell,
systemd.crash_reboot, systemd.confirm_spawn, systemd.service_watchdogs,
systemd.show_status, systemd.status_unit_format=, systemd.log_target=,
systemd.log_level=, systemd.log_location=, systemd.log_color,
systemd.default_standard_output=, systemd.default_standard_error=,
systemd.setenv=, systemd.machine_id=, systemd.set_credential=,
systemd.import_credentials=

Parameters understood by the system and service manager to control system behavior. For details, see `systemd(1)`.

systemd.mask=, systemd.wants=, systemd.debug_shell

Additional parameters understood by `systemd-debug-generator(8)`, to mask or start specific units at boot, or invoke a debug shell on `tty9`.

systemd.run=, systemd.run_success_action=, systemd.run_failure_action=

Additional parameters understood by `systemd-run-generator(8)`, to run a command line specified on the kernel command line as system service after booting up.

systemd.early_core_pattern=

During early boot, the generation of core dump files is disabled until a core dump handler (if any) takes over. This parameter allows specifying an absolute path where core dump files should be stored until a handler is installed. The path should be absolute and may contain specifiers, see `core(5)` for details.

systemd.restore_state=

This parameter is understood by several system tools to control whether or not they should restore system state from the previous boot. For details, see `systemd-backlight@.service(8)` and `systemd-rfkill.service(8)`.

systemd.volatile=

This parameter controls whether the system shall boot up in

volatile mode. Takes a boolean argument, or the special value "state". If false (the default), normal boot mode is selected, the root directory and /var/ are mounted as specified on the kernel command line or /etc/fstab, or otherwise configured. If true, full state-less boot mode is selected. In this case the root directory is mounted as volatile memory file system ("tmpfs"), and only /usr/ is mounted from the file system configured as root device, in read-only mode. This enables fully state-less boots were the vendor-supplied OS is used as shipped, with only default configuration and no stored state in effect, as /etc/ and /var/ (as well as all other resources shipped in the root file system) are reset at boot and lost on shutdown. If this setting is set to "state" the root file system is mounted read-only, however /var/ is mounted as a volatile memory file system ("tmpfs"), so that the system boots up with the normal configuration applied, but all state reset at boot and lost at shutdown. If this setting is set to "overlay" the root file system is set up as "overlayfs" mount combining the read-only root directory with a writable "tmpfs", so that no modifications are made to disk, but the file system may be modified nonetheless with all changes being lost at reboot. For details, see `systemd-volatile-root.service(8)` and `systemd-fstab-generator(8)`.

quiet

Parameter understood by both the kernel and the system and service manager to control console log verbosity. For details, see `systemd(1)`.

debug

Parameter understood by both the kernel and the system and service manager to control console log verbosity. For details, see `systemd(1)`.

-b, rd.emergency, emergency, rd.rescue, rescue, single, s, S, 1, 2, 3, 4, 5

Parameters understood by the system and service manager, as

compatibility and convenience options. For details, see `systemd(1)`.

`locale.LANG=`, `locale.LANGUAGE=`, `locale.LC_CTYPE=`, `locale.LC_NUMERIC=`,
`locale.LC_TIME=`, `locale.LC_COLLATE=`, `locale.LC_MONETARY=`,
`locale.LC_MESSAGES=`, `locale.LC_PAPER=`, `locale.LC_NAME=`,
`locale.LC_ADDRESS=`, `locale.LC_TELEPHONE=`, `locale.LC_MEASUREMENT=`,
`locale.LC_IDENTIFICATION=`

Parameters understood by the system and service manager to control
locale and language settings. For details, see `systemd(1)`.

`fsck.mode=`, `fsck.repair=`

Parameters understood by the file system checker services. For
details, see `systemd-fsck@.service(8)`.

`quotacheck.mode=`

Parameter understood by the file quota checker service. For
details, see `systemd-quotacheck.service(8)`.

`systemd.journald.forward_to_syslog=`, `systemd.journald.forward_to_kmsg=`,
`systemd.journald.forward_to_console=`, `systemd.journald.forward_to_wall=`

Parameters understood by the journal service. For details, see
`systemd-journald.service(8)`.

`vconsole.keymap=`, `vconsole.keymap_toggle=`, `vconsole.font=`,
`vconsole.font_map=`, `vconsole.font_unimap=`

Parameters understood by the virtual console setup logic. For
details, see `vconsole.conf(5)`.

`udev.log_level=`, `rd.udev.log_level=`, `udev.children_max=`,
`rd.udev.children_max=`, `udev.exec_delay=`, `rd.udev.exec_delay=`,
`udev.event_timeout=`, `rd.udev.event_timeout=`, `udev.timeout_signal=`,
`rd.udev.timeout_signal=`, `udev.blockdev_read_only=`,
`rd.udev.blockdev_read_only=`, `net.ifnames=`, `net.naming-scheme=`

Parameters understood by the device event managing daemon. For
details, see `systemd-udev.service(8)`.

`plymouth.enable=`

May be used to disable the Plymouth boot splash. For details, see
`plymouth(8)`.

`luks=`, `rd.luks=`, `luks.crypttab=`, `rd.luks.crypttab=`, `luks.name=`,

`rd.luks.name=`, `luks.uuid=`, `rd.luks.uuid=`, `luks.options=`,

`rd.luks.options=`, `luks.key=`, `rd.luks.key=`

Configures the LUKS full-disk encryption logic at boot. For details, see `systemd-cryptsetup-generator(8)`.

`fstab=`, `rd.fstab=`

Configures the `/etc/fstab` logic at boot. For details, see `systemd-fstab-generator(8)`.

`root=`, `rootfstype=`, `rootflags=`, `ro`, `rw`

Configures the root file system and its file system type and mount options, as well as whether it shall be mounted read-only or read-write initially. For details, see `systemd-fstab-generator(8)`.

`mount.usr=`, `mount.usrfstype=`, `mount.usrflags=`

Configures the `/usr` file system (if required) and its file system type and mount options. For details, see `systemd-fstab-generator(8)`.

`veritytab=`, `rd.veritytab=`, `roothash=`, `systemd.verity=`,

`rd.systemd.verity=`, `systemd.verity_root_data=`,

`systemd.verity_root_hash=`, `systemd.verity.root_options=`, `usrhash=`,

`systemd.verity_usr_data=`, `systemd.verity_usr_hash=`,

`systemd.verity_usr_options=`

Configures the integrity protection root hash for the root and `/usr` file systems, and other related parameters. For details, see `systemd-veritysetup-generator(8)`.

`systemd.getty_auto=`

Configures whether the `serial-getty@.service` will run. For details, see `systemd-getty-generator(8)`.

`systemd.gpt_auto=`, `rd.systemd.gpt_auto=`

Configures whether GPT based partition auto-discovery shall be attempted. For details, see `systemd-gpt-auto-generator(8)`.

`systemd.default_timeout_start_sec=`

Overrides the default start job timeout `DefaultTimeoutStartSec=` at boot. For details, see `systemd-system.conf(5)`.

`systemd.watchdog_device=`

Overrides the watchdog device path `WatchdogDevice=`. For details, see `systemd-system.conf(5)`.

`systemd.watchdog_sec=`

Overrides the watchdog timeout settings otherwise configured with `RuntimeWatchdog=`, `RebootWatchdog=` and `KExecWatchdogSec=`. Takes a time value (if no unit is specified, seconds is the implicitly assumed time unit) or the special strings "off" or "default". For details, see `systemd-system.conf(5)`.

`systemd.watchdog_pre_sec=`

Overrides the watchdog pre-timeout settings otherwise configured with `RuntimeWatchdogPreSec=`. Takes a time value (if no unit is specified, seconds is the implicitly assumed time unit) or the special strings "off" or "default". For details, see `systemd-system.conf(5)`.

`systemd.watchdog_pretimeout_governor=`

Overrides the watchdog pre-timeout settings otherwise configured with `RuntimeWatchdogPreGovernor=`. Takes a string value. For details, see `systemd-system.conf(5)`.

`systemd.cpu_affinity=`

Overrides the CPU affinity mask for the service manager and the default for all child processes it forks. This takes precedence over `CPUAffinity=`, see `systemd-system.conf(5)` for details.

`modules_load=`, `rd.modules_load=`

Load a specific kernel module early at boot. For details, see `systemd-modules-load.service(8)`.

`resume=`, `resumeeflags=`

Enables resume from hibernation using the specified device and mount options. All `fstab(5)`-like paths are supported. For details, see `systemd-hibernate-resume-generator(8)`.

`systemd.firstboot=`

Takes a boolean argument, defaults to on. If off, `systemd-firstboot.service(8)` will not query the user for basic system settings, even if the system boots up for the first time and the

relevant settings are not initialized yet. Not to be confused with `systemd.condition-first-boot=` (see below), which overrides the result of the `ConditionFirstBoot=` unit file condition, and thus controls more than just `systemd-firstboot.service` behaviour.

`systemd.condition-needs-update=`

Takes a boolean argument. If specified, overrides the result of `ConditionNeedsUpdate=` unit condition checks. See `systemd.unit(5)` for details.

`systemd.condition-first-boot=`

Takes a boolean argument. If specified, overrides the result of `ConditionFirstBoot=` unit condition checks. See `systemd.unit(5)` for details. Not to be confused with `systemd.firstboot=` which only controls behaviour of the `systemd-firstboot.service` system service but has no effect on the condition check (see above).

`systemd.clock-usec=`

Takes a decimal, numeric timestamp in ?s since January 1st 1970, 00:00am, to set the system clock to. The system time is set to the specified timestamp early during boot. It is not propagated to the hardware clock (RTC).

`systemd.random-seed=`

Takes a base64 encoded random seed value to credit with full entropy to the kernel's random pool during early service manager initialization. This option is useful in testing environments where delays due to random pool initialization in entropy starved virtual machines shall be avoided.

Note that if this option is used the seed is accessible to unprivileged programs from `/proc/cmdline`. This option is hence a security risk when used outside of test systems, since the (possibly) only seed used for initialization of the kernel's entropy pool might be easily acquired by unprivileged programs. It is recommended to pass 512 bytes of randomized data (as that matches the Linux kernel pool size), which may be generated with a command like the following:

```
dd if=/dev/urandom bs=512 count=1 status=none | base64 -w 0
```

Again: do not use this option outside of testing environments, it's a security risk elsewhere, as secret key material derived from the entropy pool can possibly be reconstructed by unprivileged programs.

systemd.hostname=

Accepts a hostname to set during early boot. If specified takes precedence over what is set in /etc/hostname. Note that this does not bar later runtime changes to the hostname, it simply controls the initial hostname set during early boot.

HISTORY

systemd 252

Kernel command-line arguments `systemd.unified_cgroup_hierarchy` and `systemd.legacy_systemd_cgroup_controller` were deprecated. Please switch to the unified cgroup hierarchy.

SEE ALSO

`systemd(1)`, `systemd-system.conf(5)`, `bootparam(7)`, `dracut.cmdline(7)`, `systemd-debug-generator(8)`, `systemd-fsck@.service(8)`, `systemd-quotacheck.service(8)`, `systemd-journald.service(8)`, `systemd-vconsole-setup.service(8)`, `systemd-udev.service(8)`, `plymouth(8)`, `systemd-cryptsetup-generator(8)`, `systemd-veritysetup-generator(8)`, `systemd-fstab-generator(8)`, `systemd-getty-generator(8)`, `systemd-gpt-auto-generator(8)`, `systemd-volatile-root.service(8)`, `systemd-modules-load.service(8)`, `systemd-backlight@.service(8)`, `systemd-rfkill.service(8)`, `systemd-hibernate-resume-generator(8)`, `systemd-firstboot.service(8)`, `bootctl(1)`

NOTES

1. `kernel-parameters.html`

<https://docs.kernel.org/admin-guide/kernel-parameters.html>

systemd 252

KERNEL-COMMAND-LINE(7)