



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

Rocky Enterprise Linux 9.2 Manual Pages on command 'bootctl.1'

\$ man bootctl.1

BOOTCTL(1) bootctl BOOTCTL(1)

NAME

bootctl - Control EFI firmware boot settings and manage boot loader

SYNOPSIS

bootctl [OPTIONS...] {COMMAND}

DESCRIPTION

bootctl can check the EFI firmware and boot loader status, list and manage available boot loaders and boot loader entries, and install, update, or remove the systemd-boot(7) boot loader on the current system.

GENERIC EFI FIRMWARE/BOOT LOADER COMMANDS

These commands are available on any EFI system, regardless of the boot loader used.

status

Shows brief information about the system firmware, the boot loader that was used to boot the system, the boot loaders currently available in the ESP, the boot loaders listed in the firmware's list of boot loaders and the current default boot loader entry. If

no command is specified, this is the implied default.

See the example below for details of the output.

reboot-to-firmware [BOOL]

Query or set the "Reboot-Into-Firmware-Setup" flag of the EFI firmware. Takes a boolean argument which controls whether to show the firmware setup on next system reboot. If the argument is omitted shows the current status of the flag, or whether the flag is supported. This controls the same flag as `systemctl reboot --firmware-setup`, but is more low-level and allows setting the flag independently from actually requesting a reboot.

Hint: use `systemctl reboot --firmware-setup` to reboot into firmware setup once. See `systemctl(1)` for details.

systemd-efi-options [STRING]

When called without the optional argument, prints the current value of the "SystemdOptions" EFI variable. When called with an argument, sets the variable to that value. See `systemd(1)` for the meaning of that variable.

BOOT LOADER SPECIFICATION COMMANDS

These commands are available for all boot loaders that implement the Boot Loader Specification[1] and/or the Boot Loader Interface[2], such as `systemd-boot`.

list

Shows all available boot loader entries implementing the Boot Loader Specification[1], as well as any other entries discovered or automatically generated by a boot loader implementing the Boot Loader Interface[2]. JSON output may be requested with `--json=`.

See the example below for details of the output.

set-default ID, set-oneshot ID

Sets the default boot loader entry. Takes a single boot loader entry ID string or a glob pattern as argument. The `set-oneshot` command will set the default entry only for the next boot, the `set-default` will set it persistently for all future boots.

`bootctl list` can be used to list available boot loader entries and

their IDs.

In addition, the boot loader entry ID may be specified as one of:

@default, @oneshot or @current, which correspond to the current default boot loader entry for all future boots, the current default boot loader entry for the next boot, and the currently booted boot loader entry. These special IDs are resolved to the current values of the EFI variables LoaderEntryDefault, LoaderEntryOneShot and LoaderEntrySelected, see Boot Loader Specification[1] for details.

These special IDs are primarily useful as a quick way to persistently make the currently booted boot loader entry the default choice, or to upgrade the default boot loader entry for the next boot to the default boot loader entry for all future boots, but may be used for other operations too.

If set to @saved the chosen entry will be saved as an EFI variable on every boot and automatically selected the next time the boot loader starts.

When an empty string ("") is specified as the ID, then the corresponding EFI variable will be unset.

Hint: use `systemctl reboot --boot-loader-entry=ID` to reboot into a specific boot entry and `systemctl reboot --boot-loader-menu=timeout` to reboot into the boot loader menu once. See `systemctl(1)` for details.

`set-timeout TIMEOUT`, `set-timeout-oneshot TIMEOUT`

Sets the boot loader menu timeout in seconds. The `set-timeout-oneshot` command will set the timeout only for the next boot. See `systemd.time(7)` for details about the syntax of time spans.

If this is set to `menu-hidden` or `0` no menu is shown and the default entry will be booted immediately, while setting this to `menu-force` disables the timeout while always showing the menu. When an empty string ("") is specified the bootloader will revert to its default menu timeout.

These commands manage the systemd-boot EFI boot loader, and do not work in conjunction with other boot loaders.

install

Installs systemd-boot into the EFI system partition. A copy of systemd-boot will be stored as the EFI default/fallback loader at ESP/EFI/BOOT/BOOT*.EFI. The boot loader is then added to the top of the firmware's boot loader list.

update

Updates all installed versions of systemd-boot(7), if the available version is newer than the version installed in the EFI system partition. This also includes the EFI default/fallback loader at ESP/EFI/BOOT/BOOT*.EFI. The boot loader is then added to end of the firmware's boot loader list if missing.

remove

Removes all installed versions of systemd-boot from the EFI system partition and the firmware's boot loader list.

is-installed

Checks whether systemd-boot is installed in the ESP. Note that a single ESP might host multiple boot loaders; this hence checks whether systemd-boot is one (of possibly many) installed boot loaders ? and neither whether it is the default nor whether it is registered in any EFI variables.

random-seed

Generates a random seed and stores it in the EFI System Partition, for use by the systemd-boot boot loader. Also, generates a random 'system token' and stores it persistently as an EFI variable, if one has not been set before. If the boot loader finds the random seed in the ESP and the system token in the EFI variable it will derive a random seed to pass to the OS and a new seed to store in the ESP from the combination of both. The random seed passed to the OS is credited to the kernel's entropy pool by the system manager during early boot, and permits userspace to boot up with an entropy pool fully initialized very early on. Also see systemd-boot-system-

token.service(8).

See Random Seeds[3] for further information.

OPTIONS

The following options are understood:

`--esp-path=`

Path to the EFI System Partition (ESP). If not specified, `/efi/`, `/boot/`, and `/boot/efi/` are checked in turn. It is recommended to mount the ESP to `/efi/`, if possible.

`--boot-path=`

Path to the Extended Boot Loader partition, as defined in the Boot Loader Specification[1]. If not specified, `/boot/` is checked. It is recommended to mount the Extended Boot Loader partition to `/boot/`, if possible.

`--root=root`

Takes a directory path as an argument. All paths will be prefixed with the given alternate root path, including config search paths.

`--image=image`

Takes a path to a disk image file or block device node. If specified, all operations are applied to file system in the indicated disk image. This option is similar to `--root=`, but operates on file systems stored in disk images or block devices. The disk image should either contain just a file system or a set of file systems within a GPT partition table, following the Discoverable Partitions Specification[4]. For further information on supported disk images, see `systemd-nspawn(1)`'s switch of the same name.

`--install-source=`

When installing binaries with `--root=` or `--image=`, selects where to source them from. Takes one of "auto" (the default), "image" or "host". With "auto" binaries will be picked from the specified directory or image, and if not found they will be picked from the host. With "image" or "host" no fallback search will be performed if the binaries are not found in the selected source.

-p, --print-esp-path

This option modifies the behaviour of status. Only prints the path to the EFI System Partition (ESP) to standard output and exits.

-x, --print-boot-path

This option modifies the behaviour of status. Only prints the path to the Extended Boot Loader partition if it exists, and the path to the ESP otherwise to standard output and exit. This command is useful to determine where to place boot loader entries, as they are preferably placed in the Extended Boot Loader partition if it exists and in the ESP otherwise.

Boot Loader Specification Type #1 entries should generally be placed in the directory "\$(`bootctl -x`)/loader/entries/". Existence of that directory may also be used as indication that boot loader entry support is available on the system. Similarly, Boot Loader Specification Type #2 entries should be placed in the directory "\$(`bootctl -x`)/EFI/Linux/".

Note that this option (similarly to the `--print-booht-path` option mentioned above), is available independently from the boot loader used, i.e. also without `systemd-boot` being installed.

--no-variables

Do not touch the firmware's boot loader list stored in EFI variables.

--graceful

Ignore failure when the EFI System Partition cannot be found, when EFI variables cannot be written, or a different or newer boot loader is already installed. Currently only applies to `is-installed`, `update`, and `random-seed` verbs.

-q, --quiet

Suppress printing of the results of various commands and also the hints about ESP being unavailable.

--make-entry-directory=yes|no

Controls creation and deletion of the Boot Loader Specification[1] Type #1 entry directory on the file system containing resources

such as kernel and initrd images during install and remove, respectively. The directory is named after the entry token, as specified with `--entry-token=` parameter described below, and is placed immediately below the `$BOOT` root directory (i.e. beneath the file system returned by the `--print-boot-path` option, see above).

Defaults to "no".

`--entry-token=`

Controls how to name and identify boot loader entries for this OS installation. Accepted during install, and takes one of "auto", "machine-id", "os-id", "os-image-id" or an arbitrary string prefixed by "literal:" as argument.

If set to machine-id the entries are named after the machine ID of the running system (e.g. "b0e793a9baf14b5fa13ecbe84ff637ac"). See [machine-id\(5\)](#) for details about the machine ID concept and file.

If set to os-id the entries are named after the OS ID of the running system, i.e. the `ID=` field of `os-release(5)` (e.g. "fedora"). Similarly, if set to os-image-id the entries are named after the OS image ID of the running system, i.e. the `IMAGE_ID=` field of `os-release` (e.g. "vendorx-cashier-system").

If set to auto (the default), the `/etc/kernel/entry-token` file will be read if it exists, and the stored value used. Otherwise if the local machine ID is initialized it is used. Otherwise `IMAGE_ID=` from `os-release` will be used, if set. Otherwise, `ID=` from `os-release` will be used, if set.

Unless set to "machine-id", or when `--make-entry-directory=yes` is used the selected token string is written to a file `/etc/kernel/entry-token`, to ensure it will be used for future entries. This file is also read by `kernel-install(8)`, in order to identify under which name to generate boot loader entries for newly installed kernels, or to determine the entry names for removing old ones.

Using the machine ID for naming the entries is generally preferable, however there are cases where using the other

identifiers is a good option. Specifically: if the identification data that the machine ID entails shall not be stored on the (unencrypted) \$BOOT partition, or if the ID shall be generated on first boot and is not known when the entries are prepared. Note that using the machine ID has the benefit that multiple parallel installations of the same OS can coexist on the same medium, and they can update their boot loader entries independently. When using another identifier (such as the OS ID or the OS image ID), parallel installations of the same OS would try to use the same entry name. To support parallel installations, the installer must use a different entry token when adding a second installation.

`--all-architectures`

Install binaries for all supported EFI architectures (this implies `--no-variables`).

`--efi-boot-option-description=`

Description of the entry added to the firmware's boot option list.

Defaults to "Linux Boot Manager".

Using the default entry name "Linux Boot Manager" is generally preferable as only one bootloader installed to a single ESP partition should be used to boot any number of OS installations found on the various disks installed in the system. Specifically distributions should not use this flag to install a branded entry in the boot option list. However in situations with multiple disks, each with their own ESP partition, it can be beneficial to make it easier to identify the bootloader being used in the firmware's boot option menu.

`--no-pager`

Do not pipe output into a pager.

`--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).

-h, --help

Print a short help text and exit.

--version

Print a short version string and exit.

SIGNED .EFI FILES

bootctl install and update will look for a systemd-boot file ending with the ".efi.signed" suffix first, and copy that instead of the normal ".efi" file. This allows distributions or end-users to provide signed images for UEFI SecureBoot.

EXIT STATUS

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

ENVIRONMENT

If \$SYSTEMD_RELAX_ESP_CHECKS=1 is set the validation checks for the ESP are relaxed, and the path specified with --esp-path= may refer to any kind of file system on any kind of partition.

Similarly, \$SYSTEMD_RELAX_XBOOTLDR_CHECKS=1 turns off some validation checks for the Extended Boot Loader partition.

EXAMPLES

Example 1. Output from status and list

```
$ bootctl status
```

System:

Firmware: UEFI 2.40 (firmware-version) ? firmware vendor and version

Secure Boot: disabled (setup) ? secure boot status

TPM2 Support: yes

Boot into FW: supported ? does the firmware support booting into itself

Current Boot Loader: ? details about sd-boot or another boot loader

Product: systemd-boot version implementing the Boot Loader Interface[2]

Features: ? Boot counting

? Menu timeout control

? One-shot menu timeout control

? Default entry control

? One-shot entry control

- ? Support for XBOOTLDR partition
- ? Support for passing random seed to OS
- ? Load drop-in drivers
- ? Boot loader sets ESP information

ESP: /dev/disk/by-partuuid/01234567-89ab-cdef-dead-beef00000000

File: ??/EFI/systemd/systemd-bootx64.efi

Random Seed: ? random seed used for entropy in early boot

Passed to OS: yes

System Token: set

Exists: yes

Available Boot Loaders on ESP:

ESP: /boot/efi (/dev/disk/by-partuuid/01234567-89ab-cdef-dead-beef00000000)

File: ??/EFI/systemd/systemd-bootx64.efi (systemd-boot 251)

File: ??/EFI/BOOT/BOOTX64.EFI (systemd-boot 251)

Boot Loaders Listed in EFI Variables:

Title: Linux Boot Manager

ID: 0x0001

Status: active, boot-order

Partition: /dev/disk/by-partuuid/...

File: ??/EFI/systemd/systemd-bootx64.efi

Title: Fedora

ID: 0x0000

Status: active, boot-order

Partition: /dev/disk/by-partuuid/...

File: ??/EFI/fedora/shimx64.efi

Title: Linux-Firmware-Updater

ID: 0x0002

Status: active, boot-order

Partition: /dev/disk/by-partuuid/...

File: ??/EFI/fedora/fwupdx64.efi

Boot Loader Entries:

\$BOOT: /boot/efi (/dev/disk/by-partuuid/01234567-89ab-cdef-dead-beef00000000)

Default Boot Loader Entry:

type: Boot Loader Specification Type #1 (.conf)

title: Fedora Linux 36 (Workstation Edition)

id: ...

source: /boot/efi/loader/entries/entry-token-kernel-version.conf

version: kernel-version

machine-id: ...

linux: /entry-token/kernel-version/linux

initrd: /entry-token/kernel-version/initrd

options: root=...

\$ bootctl list

Boot Loader Entries:

type: Boot Loader Specification Type #1 (.conf)

title: Fedora Linux 36 (Workstation Edition) (default) (selected)

id: ...

source: /boot/efi/loader/entries/entry-token-kernel-version.conf

version: kernel-version

machine-id: ...

linux: /entry-token/kernel-version/linux

initrd: /entry-token/kernel-version/initrd

options: root=...

type: Boot Loader Specification Type #2 (.efi)

title: Fedora Linux 35 (Workstation Edition)

id: ...

source: /boot/efi/EFI/Linux/fedora-kernel-version.efi

version: kernel-version

machine-id: ...

linux: /EFI/Linux/fedora-kernel-version.efi

options: root=...

type: Automatic

title: Reboot Into Firmware Interface

id: auto-reboot-to-firmware-setup

source: /sys/firmware/efi/efivars/LoaderEntries-4a67b082-0a4c-41cf-b6c7-440b29bb8c4f

default, and "(selected)" specifies the entry that was selected the last time (i.e. is currently running).

SEE ALSO

systemd-boot(7), Boot Loader Specification[1], Boot Loader Interface[2], systemd-boot-system-token.service(8)

NOTES

1. Boot Loader Specification

https://systemd.io/BOOT_LOADER_SPECIFICATION

2. Boot Loader Interface

https://systemd.io/BOOT_LOADER_INTERFACE

3. Random Seeds

https://systemd.io/RANDOM_SEEDS

4. Discoverable Partitions Specification

https://systemd.io/DISCOVERABLE_PARTITIONS

systemd 252

BOOTCTL(1)