OpenCore

Reference Manual (1.0.2.3)

[2024.11.18]

If using OpenLinuxBoot with Secure Boot, users may wish to install a user built, user signed Shim bootloader giving SBAT and MOK integration, as explained in OpenCore ShimUtils.

11.7.1 Configuration

The default parameter values should work well with no changes under most circumstances, but if required the following options for the driver may be specified in `UEFI/Drivers/Arguments`:

- `flags` - Default: all flags are set except the following:
  - `LINUX_BOOT_ADD_RW`,
  - `LINUX_BOOT_LOG_VERBOSE`,
  - `LINUX_BOOT_LOG_GRUB_VARS` and
  - `LINUX_BOOT_ADD_DEBUG_INFO`.

  Available flags are:

  - `0x00000001` (bit `0`) — `LINUX_BOOT_SCAN_ESP`, Allows scanning for entries on EFI System Partition.
  - `0x00000002` (bit `1`) — `LINUX_BOOT_SCAN_XBOOTLDR`, Allows scanning for entries on Extended Boot Loader Partition.
  - `0x00000004` (bit `2`) — `LINUX_BOOT_SCAN_LINUX_ROOT`, Allows scanning for entries on Linux Root filesystems.
  - `0x00000008` (bit `3`) — `LINUX_BOOT_SCAN_LINUX_DATA`, Allows scanning for entries on Linux Data filesystems.
  - `0x00000080` (bit `7`) — `LINUX_BOOT_SCAN_OTHER`, Allows scanning for entries on file systems not matched by any of the above.

    The following notes apply to all of the above options:

    Note 1: Apple filesystems APFS and HFS are never scanned.

    Note 2: Regardless of the above flags, a file system must first be allowed by `Misc/Security/ScanPolicy` before it can be seen by OpenLinuxBoot or any other `OC_BOOT_ENTRY_PROTOCOL` driver.

    Note 3: It is recommended to enable scanning `LINUX_ROOT` and `LINUX_DATA` in both OpenLinuxBoot flags and `Misc/Security/ScanPolicy` in order to be sure to detect all valid Linux installs, since Linux boot filesystems are very often marked as `LINUX_DATA`.

  - `0x00000100` (bit `8`) — `LINUX_BOOT_ALLOW_AUTODETECT`, If set allows autodetecting and linking `vmlinuz*` and `init*` ramdisk files when `loader/entries` files are not found.
  - `0x00000200` (bit `9`) — `LINUX_BOOT_USE_LATEST`, When a Linux entry generated by OpenLinuxBoot is selected as the default boot entry in OpenCore, automatically switch to the latest kernel when a new version is installed.

    When this option is set, an internal menu entry id is shared between kernel versions from the same install of Linux. Linux boot options are always sorted highest kernel version first, so this means that the latest kernel version of the same install always shows as the default, with this option set.

    Note: This option is recommended on all systems.

  - `0x00000400` (bit `10`) — `LINUX_BOOT_ADD_RO`, This option applies to autodetected Linux only (i.e. not to BLSpec or Fedora-style distributions which have `/loader/entries/*.conf` files). Some distributions run a filesystem check on loading which requires the root filesystem to initially be mounted read-only via the `ro` kernel option, which requires this option to be added to the autodetected options. Set this bit to add this option on autodetected distros; should be harmless but very slightly slow down boot time (due to requried remount as read-write) on distros which do not require it. When there are multiple distros and it is required to specify this option for specific distros only, use `autoopts:{PARTUUID}+=ro` to manually add the option where required, instead of using this flag.
  - `0x00000800` (bit `11`) — `LINUX_BOOT_ADD_RW`, Like `LINUX_BOOT_ADD_RO`, this option applies to autodetected Linux only. It is not required for most distros (which usually require either `ro` or nothing to be added to detected boot options), but is required on some Arch-derived distros, e.g. EndeavourOS. When there are multiple distros and it is required to specify this option for specific distros only, use `autoopts:{PARTUUID}+=rw` to manually add the option where required, instead of using this flag. If this option and `LINUX_BOOT_ADD_RO` are both specified, only this option is applied and `LINUX_BOOT_ADD_RO` is ignored.
  - `0x00002000` (bit `13`) — `LINUX_BOOT_ALLOW_CONF_AUTO_ROOT`, In some instances of `BootLoaderSpecByDefault` in combination with `ostree`, the `/loader/entries/*.conf` files do not specify a required `root=...` kernel option – it is added by GRUB. If this bit is set and this situation is detected, then automatically add this option. (Required for example by Endless OS.)

- `0x00004000` (bit `14`) — `LINUX_BOOT_LOG_VERBOSE`, Add additional debug log info about files encountered and autodetect options added while scanning for Linux boot entries.
- `0x00008000` (bit `15`) — `LINUX_BOOT_ADD_DEBUG_INFO`, Adds a human readable file system type, followed by the first eight characters of the partition's unique partition uuid, to each generated entry name. Can help with debugging the origin of entries generated by the driver when there are multiple Linux installs on one system.
- `0x00010000` (bit `16`) — `LINUX_BOOT_LOG_GRUB_VARS`, When a `BootLoaderSpecByDefault` setup is detected, log available GRUB variables found in `grub2/grubenv` and `grub2/grub.cfg`.
- `0x00020000` (bit `17`) — `LINUX_BOOT_FIX_TUNED`, In some circumstances, such as after upgrades which add TuneD to existing systems, the TuneD system tuning plugin may add its GRUB variables to `loader/entries/*.conf` files but not initialise them in `grub2/grub.cfg`. In order to avoid incorrect boots, OpenLinuxBoot treats used, non-initialised GRUB variables as an error. When this flag is set, empty values are added for the TuneD variables `tuned_params` and `tuned_initrd` if they are not present. This is required for OpenLinuxBoot on TuneD systems with this problem, and harmless otherwise.

Flag values can be specified in hexadecimal beginning with `0x` or in decimal, e.g. `flags=0x80` or `flags=128`. It is also possible to specify flags to add or remove, using syntax such as `flags+=0xC000` to add all debugging options or `flags-=0x400` to remove the `LINUX_BOOT_ADD_RO` option.

- `autoopts:{PARTUUID}[+]="{options}"` - Default: not set.

  Allows manually specifying kernel options to use in autodetect mode for a given partition only. Replace the text `{PARTUUID}` with the specific partition UUID on which the kernels are stored (in normal layout, the partition which contains `/boot`), e.g. `autoopts:11223344-5566-7788-99aa-bbccddeeff00+="vt.handoff=7"`. If specified with `+=` then these options are used in addition to any autodetected options, if specified with `=` they are used instead. Used for autodetected Linux only – values specified here are never used for entries created from `/loader/entries/*.conf` files.

  Note: The `PARTUUID` value to be specified here is typically the same as the `PARTUUID` seen in `root=PARTUUID=...` in the Linux kernel boot options (view using `cat /proc/cmdline`). Alternatively, and for more advanced scenarios, it is possible to examine how the distro's partitions are mounted using the Linux `mount` command, and then find out the partuuid of relevant mounted partitions by examining the output of `ls -l /dev/disk/by-partuuid`.

- `autoopts[+]="{options}"` - Default: None specified.

  Allows manually specifying kernel options to use in autodetect mode. The alternative format `autoopts:{PARTUUID}` is more suitable where there are multiple distros, but `autoopts` with no PARTUUID required may be more convenient for just one distro. If specified with `+=` then these are used in addition to autodetected options, if specified with `=` they are used instead. Used for autodetected Linux only – values specified here are never used for entries created from `/loader/entries/*.conf` files.

  As example usage, it is possible to use `+=` format to add a `vt.handoff` options, such as `autoopts+="vt.handoff=7"` or `autoopts+="vt.handoff=3"` (check `cat /proc/cmdline` when booted via the distro's default bootloader) on Ubuntu and related distros, in order to add the `vt.handoff` option to the auto-detected GRUB defaults, and avoid a flash of text showing before the distro splash screen.

11.7.2 Additional information

OpenLinuxBoot can detect the `loader/entries/*.conf` files created according to the Boot Loader Specification or the closely related ~~systemd~~ Fedora BootLoaderSpecByDefault. The former is specific to systemd-boot and is used by Arch Linux, the latter applies to most Fedora-related distros including Fedora itself, RHEL and variants.

Where the above files are not present, OpenLinuxBoot can autodetect and boot `{boot}/vmlinuz*` kernel files directly. It links these automatically – based on the kernel version in the filename – to their associated `{boot}/init*` ramdisk files. This applies to most Debian-related distros, including Debian itself, Ubuntu and variants.

When autodetecting in `/boot` as part of the root filesystem, OpenLinuxBoot looks in `/etc/default/grub` for kernel boot options and `/etc/os-release` for the distro name. When autodetecting in a standalone boot partition (i.e. when `/boot` has its own mount point), OpenLinuxBoot cannot autodetect kernel arguments and all kernel arguments except `initrd=...` must be fully specified by hand using `autoopts=...` or `autoopts:{partuuid}=...` (`+=` variants of these options will not work, as these only add additional arguments).

~~BootLoaderSpecByDefault~~ Fedora `BootLoaderSpecByDefault` (but not pure Boot Loader Specification) can expand GRUB variables in the `*.conf` files – and this is used in practice in certain distros such as CentOS. In order to handle this correctly, when this situation is detected OpenLinuxBoot extracts all variables from `{boot}/grub2/grubenv` and also any unconditionally set variables from `{boot}/grub2/grub.cfg`, and then expands these where required in `*.conf` file entries.

The only currently supported method of starting Linux kernels from OpenLinuxBoot relies on their being compiled with EFISTUB. This applies to almost all modern distros, particularly those which use systemd. Note that most modern distros use systemd as their system manager, even though most do not use systemd-boot as their bootloader.

systemd-boot users (probably almost exclusively Arch Linux users) should be aware that OpenLinuxBoot does not support the systemd-boot–specific Boot Loader Interface; therefore `efibootmgr` rather than `bootctl` must be used for any low-level Linux command line interaction with the boot menu.

11.8 Other Boot Entry Protocol drivers

In addition to the OpenLinuxBoot plugin, the following `OC_BOOT_ENTRY_PROTOCOL` plugins are made available to add optional, configurable boot entries to the OpenCore boot picker.

11.8.1 ResetNvramEntry

Adds a menu entry which resets NVRAM and immediately restarts. Additionally adds support for hotkey `CMD+OPT+P+R` to perform the same action. Note that on some combinations of firmware and drivers, the `TakeoffDelay` option must be configured in order for this and other builtin hotkeys to be reliably detected.

Note 1: It is known that some Lenovo laptops have a firmware bug, which makes them unbootable after performing NVRAM reset. Refer to acidanthera/bugtracker#995 for details.

Note 2: If `LauncherOption` is set to `Full` or `Short` then the OpenCore boot entry is protected. Resetting NVRAM will normally erase any other boot options not specified via `BlessOverride`, for example Linux installations to custom locations and not using the `OpenLinuxBoot` driver, or user-specified UEFI boot menu entries. To obtain reset NVRAM functionality which does not remove other boot options, it is possible to use the `--preserve-boot` option (though see the warning specified).

The following configuration options may be specified in the `Arguments` section for this driver:

- `--preserve-boot` - Boolean flag, enabled if present.

  If enabled, BIOS boot entries are not cleared during NVRAM reset. This option should be used with caution, as some boot problems can be fixed by clearing these entries.

- `--apple` - Boolean flag, enabled if present.

  On Apple firmware only, this performs a system NVRAM reset. This can result in additional, desirable operations such as NVRAM garbage collection. This is achieved by setting the `ResetNVRam` NVRAM variable. Where available, this has the same effect as pressing `CMD+OPT+P+R` during native boot, although note that if accessed from the menu entry only one boot chime will be heard.

  Note 1: Due to using system NVRAM reset, this option is not compatible with the `--preserve-boot` option and will override it, therefore all BIOS boot entries will be removed.

  Note 2: Due to using system NVRAM reset, the OpenCore boot option cannot be preserved and OpenCore will have to either be reselected in the native boot picker or re-blessed.

  Note 3: On non-Apple hardware, this option will still set this variable but the variable will not be recognised by the firmware and no NVRAM reset will happen.