



# OpenCore

Reference Manual (0.9.~~7~~.8)

[2024.01.17]

disklabel utility or the `bleed --folder {FOLDER_PATH} --label {LABEL_TEXT}` command. When pre-rendered labels are disabled or missing, use label text in `.contentDetails` (or `.disk_label.contentDetails`) file next to bootloader if present instead, otherwise the entry name itself will be rendered.

- `0x0004` — `OC_ATTR_USE_GENERIC_LABEL_IMAGE`, provides predefined label images for boot entries without custom entries. This may however give less detail for the actual boot entry.
- `0x0008` — `OC_ATTR_HIDE_THEMED_ICONS`, prefers builtin icons for certain icon categories to match the theme style. For example, this could force displaying the builtin Time Machine icon. Requires `OC_ATTR_USE_VOLUME_ICON`.
- `0x0010` — `OC_ATTR_USE_POINTER_CONTROL`, enables pointer control in the OpenCore picker when available. For example, this could make use of mouse or trackpad to control UI elements.
- `0x0020` — `OC_ATTR_SHOW_DEBUG_DISPLAY`, enable display of additional timing and debug information, in Builtin picker in `DEBUG` and `NOOPT` builds only.
- `0x0040` — `OC_ATTR_USE_MINIMAL_UI`, use minimal UI display, no Shutdown or Restart buttons, affects OpenCanopy and builtin picker.
- `0x0080` — `OC_ATTR_USE_FLAVOUR_ICON`, provides flexible boot entry content description, suitable for picking the best media across different content sets:

When enabled, the entry icon in OpenCanopy and the audio assist entry sound in OpenCanopy and builtin boot picker are chosen by something called content flavour. To determine content flavour the following algorithm is used:

- For a Tool the value is read from `Flavour` field.
- For an automatically discovered entry, including for boot entry protocol entries such as those generated by the OpenLinuxBoot driver, it is read from the `.contentFlavour` file next to the bootloader, if present.
- For a custom entry specified in the `Entries` section it is read from the `.contentFlavour` file next to the bootloader if `Flavour` is `Auto`, otherwise it is specified via the `Flavour` value itself.
- If read flavour is `Auto` or there is no `.contentFlavour`, entry flavour is chosen based on the entry type (e.g. Windows automatically gets Windows flavour).

The Flavour value is a sequence of : separated names limited to 64 characters of printable 7-bit ASCII. This is designed to support up to approximately five names. Each name refers to a flavour, with the first name having the highest priority and the last name having the lowest priority. Such a structure allows describing an entry in a more specific way, with icons selected flexibly depending on support by the audio-visual pack. A missing audio or icon file means the next flavour should be tried, and if all are missing the choice happens based on the type of the entry. Example flavour values: `BigSur:Apple`, `Windows10:Windows`, `OpenShell:UEFIShell:Shell`.

Using flavours means that you can switch between icon sets easily, with the flavour selecting the best available icons from each set. E.g. specifying icon flavour `Debian:Linux` will use the icon `Debian.icns` if provided, then will try `Linux.icns`, then will fall back to the default for an OS, which is `HardDrive.icns`.

Things to keep in mind:

- For security reasons `Ext<Flavour>.icns` and `<Flavour>.icns` are both supported, and only `Ext<Flavour>.icns` will be used if the entry is on an external drive (followed by default fallback `ExtHardDrive.icns`).
- Where both apply `.VolumeIcon.icns` takes precedence over `.contentFlavour`.
- In order to allow icons and audio assist to work correctly for tools (e.g. for UEFI Shell), system default boot entry icons (see `Docs/Flavours.md`) specified in the `Flavour` setting for `Tools` or `Entries` will continue to apply even when flavour is disabled. Non-system icons will be ignored in this case. In addition, the flavours `UEFIShell` and `NVRAMReset` are given special processing, identifying their respective tools to apply correct audio-assist, default builtin labels, etc.
- A list of recommended flavours is provided in `Docs/Flavours.md`.
- `0x0100` — `OC_ATTR_USE_REVERSED_UI`, reverse position of Shutdown and Restart buttons, affects OpenCanopy and builtin picker. The reversed setting matches older macOS, and since it was the previous default in OpenCore it may better match some custom backgrounds. Only applicable when `OC_ATTR_USE_MINIMAL_UI` is not set.
- `0x0200` — `OC_ATTR_REDUCE_MOTION`, reduce password and menu animation in OpenCanopy, leaving only animations which communicate information not otherwise provided.  
*Note:* These same animations, plus additional animations whose information is provided by voice-over, are automatically disabled when `PickerAudioAssist` is enabled.

## 9. PickerAudioAssist

Type: plist boolean

# 11 UEFI

## 11.1 Introduction

UEFI (Unified Extensible Firmware Interface) is a specification that defines a software interface between an operating system and platform firmware. This section allows loading additional UEFI modules as well as applying tweaks to the onboard firmware. To inspect firmware contents, apply modifications and perform upgrades UEFITool and supplementary utilities can be used.

## 11.2 Drivers

Depending on the firmware, a different set of drivers may be required. Loading an incompatible driver may lead the system to unbootable state or even cause permanent firmware damage. Some of the known drivers are listed below:

AudioDxe*	HDA audio support driver in UEFI firmware for most Intel and some other analog audio controllers. Staging driver, refer to <a href="#">acidanthera/bugtracker#740</a> for known issues in AudioDxe.
btrfs_x64	Open source BTRFS file system driver, required for booting with OpenLinuxBoot from a file system which is now quite commonly used with Linux.
BiosVideo*	CSM video driver implementing graphics output protocol based on VESA and legacy BIOS interfaces. Used for UEFI firmware with fragile GOP support (e.g. low resolution). Requires <b>ReconnectGraphicsOnConnect</b> . Included in OpenDuet out of the box.
CrScreenshotDxe*	Screenshot making driver saving images to the root of OpenCore partition (ESP) or any available writeable filesystem upon pressing F10. Accepts optional driver argument <b>--enable-mouse-click</b> to additionally take screenshot on mouse click. (It is recommended to enable this option only if a keypress would prevent a specific screenshot, and disable it again after use.) This is a modified version of <b>CrScreenshotDxe</b> driver by Nikolaj Schlej.
EnableGop{Direct}*	Early beta release firmware-embeddable driver providing pre-OpenCore non-native GPU support on MacPro5,1. Installation instructions can be found in the <b>Utilities/EnableGop</b> directory of the OpenCore release zip file - proceed with caution.
ExFatDxe	Proprietary ExFAT file system driver for Bootcamp support commonly found in Apple firmware. For Sandy Bridge and earlier CPUs, the <b>ExFatDxeLegacy</b> driver should be used due to the lack of RDRAND instruction support.
ext4_x64	Open source EXT4 file system driver, required for booting with OpenLinuxBoot from the file system most commonly used with Linux.
<u>FirmwareSettings*</u>	OpenCore plugin <a href="#">implementing OC_BOOT_ENTRY_PROTOCOL to add an entry to the boot picker menu which reboots into UEFI firmware settings, when this is supported by the firmware.</a>
HfsPlus	Recommended. Proprietary HFS file system driver with bless support commonly found in Apple firmware. For Sandy Bridge and earlier CPUs, the <b>HfsPlusLegacy</b> driver should be used due to the lack of RDRAND instruction support.
HiiDatabase*	HII services support driver from <b>MdeModulePkg</b> . This driver is included in most types of firmware starting with the Ivy Bridge generation. Some applications with GUI, such as UEFI Shell, may need this driver to work properly.
EnhancedFatDxe	FAT filesystem driver from <b>FatPkg</b> . This driver is embedded in all UEFI firmware and cannot be used from OpenCore. Several types of firmware have defective FAT support implementation that may lead to corrupted filesystems on write attempts. Embedding this driver within the firmware may be required in case writing to the EFI partition is needed during the boot process.
NvmExpressDxe*	NVMe support driver from <b>MdeModulePkg</b> . This driver is included in most firmware starting with the Broadwell generation. For Haswell and earlier, embedding it within the firmware may be more favourable in case a NVMe SSD drive is installed.
OpenCanopy*	OpenCore plugin implementing graphical interface.
OpenRuntime*	OpenCore plugin implementing <b>OC_FIRMWARE_RUNTIME</b> protocol.
OpenLegacyBoot*	OpenCore plugin implementing <b>OC_BOOT_ENTRY_PROTOCOL</b> to allow detection and booting of legacy operating systems from OpenCore on Macs, OpenDuet and systems with a CSM.

- `ResetNVRAM` — Reset NVRAM system action or tool (128x128).
- `Shell` — Entry with UEFI Shell name for e.g. `OpenShell` (128x128).
- `Tool` — Any other tool (128x128).

Predefined labels are saved in the `\EFI\OC\Resources\Label` directory. Each label has `.1b1` or `.12x` suffix to represent the scaling level. Full list of labels is provided below. All labels are mandatory.

- `EFIBoot` — Generic OS.
- `Apple` — Apple OS.
- `AppleRecovery` — Apple Recovery OS.
- `AppleTM` — Apple Time Machine.
- `Windows` — Windows.
- `Other` — Custom entry (see `Entries`).
- `ResetNVRAM` — Reset NVRAM system action or tool.
- `SIPDisabled` — Toggle SIP tool with SIP disabled.
- `SIPEnabled` — Toggle SIP tool with SIP enabled.
- `Shell` — Entry with UEFI Shell name (e.g. `OpenShell`).
- [`FirmwareSettings` — Firmware settings menu entry.](#)
- `Tool` — Any other tool.

*Note:* All labels must have a height of exactly 12 px. There is no limit for their width.

Label and icon generation can be performed with bundled utilities: `disklabel` and `icnspack`. Font is Helvetica 12 pt times scale factor.

Font format corresponds to AngelCode binary BMF. While there are many utilities to generate font files, currently it is recommended to use `dpFontBaker` to generate bitmap font (using `CoreText` produces best results) and `fonverter` to export it to binary format.

## 11.5 OpenRuntime

`OpenRuntime` is an OpenCore plugin implementing `OC_FIRMWARE_RUNTIME` protocol. This protocol implements multiple features required for OpenCore that are otherwise not possible to implement in OpenCore itself as they are needed to work in runtime, i.e. during operating system functioning. Feature highlights:

- NVRAM namespaces, allowing to isolate operating systems from accessing select variables (e.g. `RequestBootVarRouting` or `ProtectSecureBoot`).
- Read-only and write-only NVRAM variables, enhancing the security of OpenCore, Lilu, and Lilu plugins, such as `VirtualSMC`, which implements `AuthRestart` support.
- NVRAM isolation, allowing to protect all variables from being written from an untrusted operating system (e.g. `DisableVariableWrite`).
- UEFI Runtime Services memory protection management to workaround read-only mapping (e.g. `EnableWriteUnprotector`).

## 11.6 OpenLegacyBoot

`OpenLegacyBoot` is an OpenCore plugin implementing `OC_BOOT_ENTRY_PROTOCOL`. It aims to detect and boot legacy installed operating systems on supported systems, such as OpenDuet and Mac models capable of legacy booting.

Usage:

- Add `OpenLegacyBoot.efi` and also optionally (see below) `OpenNtfsDxe.efi` to the `config.plist Drivers` section.
- Install Windows or another legacy operating system as normal if this has not been done earlier – `OpenLegacyBoot` is not involved in this stage and may be unable to boot from installation media such as a USB device.
- Reboot into OpenCore: the installed legacy operating system should appear and boot directly from OpenCore when selected.

`OpenLegacyBoot` does not require any additional filesystem drivers such as `OpenNtfsDxe.efi` to be loaded for base functionality, but loading them will enable the use of `.contentDetails` and `.VolumeIcon.icns` files for boot entry customisation.

- `--show-csr` - Boolean flag, enabled if present.

If enabled, show the current hexadecimal value of `csr-active-config` in the boot entry name. This option will not work in OpenCanopy when used in combination with `OC_ATTR_USE_GENERIC_LABEL_IMAGE` in `PickerAttributes`.

- Numerical value - Default value `0x27F`.

Specify the `csr-active-config` value to use to disabled SIP. This can be specified as hexadecimal, beginning with `0x`, or as decimal. For more info see Note 2 below.

*Note 1:* It is recommended not to run macOS with SIP disabled. Use of this boot option may make it easier to quickly disable SIP protection when genuinely needed - it should be re-enabled again afterwards.

*Note 2:* The default value for disabling SIP with this boot entry is `0x27F`. For comparison, `csrutil disable` with no other arguments on macOS Big Sur and Monterey sets `0x7F`, and on Catalina it sets `0x77`. The OpenCore default value of `0x27F` is a variant of the Big Sur and Monterey value, chosen as follows:

- `CSR_ALLOW_UNAPPROVED_KEXTS` (`0x200`) is included in the default value, since it is generally useful, in the case where you need to have SIP disabled anyway, to be able to install unsigned kexts without manual approval in System Preferences.
- `CSR_ALLOW_UNAUTHENTICATED_ROOT` (`0x800`) is not included in the default value, as it is very easy when using it to inadvertently break OS seal and prevent incremental OTA updates.
- If unsupported bits from a later OS are specified in `csr-active-config` (e.g. specifying `0x7F` on Catalina) then `csrutil status` will report that SIP has a non-standard value, however protection will be functionally the same.

### 11.8.3 [FirmwareSettings](#)

[Adds a menu entry which will reboot into UEFI firmware settings, when supported. No menu entry added and logs a warning when not supported.](#)

## 11.9 AudioDxe

High Definition Audio (HDA) support driver in UEFI firmware for most Intel and some other analog audio controllers.

*Note:* AudioDxe is a staging driver, refer to [acidanthera/bugtracker#740](#) for known issues.

### 11.9.1 Configuration

Most UEFI audio configuration is handled via the `UEFI Audio Properties` section, but in addition some of the following configuration options may be required in order to allow AudioDxe to correctly drive certain devices. All options are specified as text strings, separated by space if more than one option is required, in the `Arguments` property for the driver within the `UEFI/Drivers` section:

- `--codec-setup-delay` - Integer value, default 0.

Amount of time in milliseconds to wait for all widgets to come fully on, applied per codec during driver connection phase. In most systems this should not be needed and a faster boot will be achieved by using `Audio` section `SetupDelay` if any audio setup delay is required. Where required, values of up to one second may be needed.

- `--force-codec` - Integer value, no default.

Force use of an audio codec, this value should be equal to `Audio` section `AudioCodec`. Can result in faster boot especially when used in conjunction with `--force-device`.

- `--force-device` - String value, no default.

When this option is present and has a value (e.g. `--force-device=PciRoot(0x0)/Pci(0x1f,0x3)`), it forces AudioDxe to connect to the specified PCI device, even if the device does not report itself as an HDA audio controller.

During driver connection, AudioDxe automatically provides audio services on all supported codecs of all available HDA controllers. However, if the relevant controller is misreporting its identity (typically, it will be reporting itself as a legacy audio device instead of an HDA controller) then this argument may be required.