



OpenCore

Reference Manual (0.5.~~6~~.7)

[2020.03.06]

6 DeviceProperties

6.1 Introduction

Device configuration is provided to macOS with a dedicated buffer, called [EfiDevicePropertyDatabase](#)[EfiDevicePathPropertyData](#). This buffer is a serialised map of DevicePaths to a map of property names and their values.

Property data can be debugged with gxfutil. To obtain current property data use the following command in macOS:

```
ioreg -lw0 -p IODeviceTree -n efi -r -x | grep device-properties |  
sed 's/.*<///;s/>.*//' > /tmp/device-properties.hex &&  
gxfutil /tmp/device-properties.hex /tmp/device-properties.plist &&  
cat /tmp/device-properties.plist
```

6.2 Properties

1. Add

Type: `plist dict`

Description: Sets device properties from a map (`plist dict`) of device paths to a map (`plist dict`) of variable names and their values in `plist metadata` format. Device paths must be provided in canonic string format (e.g. `PciRoot(0x0)/Pci(0x1,0x0)/Pci(0x0,0x0)`). Properties will only be set if not present and not blocked.

Note: Currently properties may only be (formerly) added by the original driver, so unless a separate driver was installed, there is no reason to block the variables.

2. Block

Type: `plist dict`

Description: Removes device properties from a map (`plist dict`) of device paths to an array (`plist array`) of variable names in `plist string` format.

6.3 Common Properties

Some known properties include:

- `device-id`
User-specified device identifier used for I/O Kit matching. Has 4 byte data type.
- `vendor-id`
User-specified vendor identifier used for I/O Kit matching. Has 4 byte data type.
- `AAPL,ig-platform-id`
Intel GPU framebuffer identifier used for framebuffer selection on Ivy Bridge and newer. Has 4 byte data type.
- `AAPL,snb-platform-id`
Intel GPU framebuffer identifier used for framebuffer selection on Sandy Bridge. Has 4 byte data type.
- `layout-id`
Audio layout used for AppleHDA layout selection. Has 4 byte data type.

- Entry is macOS recovery.
- Entry is [macOS Time Machine](#).
- [Entry is](#) explicitly marked as Auxiliary.
- Entry is system (e.g. Clean NVRAM).

To see all entries picker menu needs to be reloaded in extended mode by pressing `Spacebar` key. Hiding auxiliary entries may increase boot performance for multidisk systems.

3. HideSelf

Type: plist boolean

Failsafe: false

Description: Hides own boot entry from boot picker. This may potentially hide other entries, for instance, when another UEFI OS is installed on the same volume and driver boot is used.

4. PickerAttributes

Type: plist integer

Failsafe: 0

Description: Sets specific attributes for picker.

Builtin picker supports colour arguments as a sum of foreground and background colors according to UEFI specification. The value of black background and black foreground (0) is reserved. List of colour names:

- 0x00 — EFI_BLACK
- 0x01 — EFI_BLUE
- 0x02 — EFI_GREEN
- 0x03 — EFI_CYAN
- 0x04 — EFI_RED
- 0x05 — EFI_MAGENTA
- 0x06 — EFI_BROWN
- 0x07 — EFI_LIGHTGRAY
- 0x08 — EFI_DARKGRAY
- 0x09 — EFI_LIGHTBLUE
- 0x0A — EFI_LIGHTGREEN
- 0x0B — EFI_LIGHTCYAN
- 0x0C — EFI_LIGHTRED
- 0x0D — EFI_LIGHTMAGENTA
- 0x0E — EFI_YELLOW
- 0x0F — EFI_WHITE
- 0x10 — EFI_BACKGROUND_BLACK
- 0x11 — EFI_BACKGROUND_BLUE
- 0x12 — EFI_BACKGROUND_GREEN
- 0x13 — EFI_BACKGROUND_CYAN
- 0x14 — EFI_BACKGROUND_RED
- 0x15 — EFI_BACKGROUND_MAGENTA
- 0x16 — EFI_BACKGROUND_BROWN
- 0x17 — EFI_BACKGROUND_LIGHTGRAY

Note: This option may not work well with `System` text renderer. Setting a background different from black could help testing proper GOP functioning.

5. PickerAudioAssist

Type: plist boolean

Failsafe: false

Description: Enable screen reader by default in boot picker.

For macOS bootloader screen reader preference is set in `preferences.efires` archive in `isV0Enabled.int32` file and is controlled by the operating system. For OpenCore screen reader support this option is an independent equivalent. Toggling screen reader support in both OpenCore boot picker and macOS bootloader FileVault 2 login window can also be done with `Command + F5` key combination.

Note: screen reader requires working audio support, see `UEFI Audio Properties` section for more details.

7. Quirks

Type: plist dict

Failsafe: None

Description: Apply individual firmware quirks described in Quirks Properties section below.

11.3 Audio Properties

1. AudioCodec

Type: plist integer

Failsafe: ~~empty string~~0

Description: Codec address on the specified audio controller for audio support.

Normally this contains first audio codec address on the builtin analog audio controller (HDEF). Audio codec addresses, e.g. 2, can be found in the debug log (marked in bold):

```
OCAU: 1/3 PciRoot(0x0)/Pci(0x1,0x0)/Pci(0x0,0x1)/VenMsg(<redacted>,00000000) (4 outputs)
```

```
OCAU: 2/3 PciRoot(0x0)/Pci(0x3,0x0)/VenMsg(<redacted>,00000000) (1 outputs)
```

```
OCAU: 3/3 PciRoot(0x0)/Pci(0x1B,0x0)/VenMsg(<redacted>,02000000) (7 outputs)
```

As an alternative this value can be obtained from IOHDACodecDevice class in I/O Registry containing it in IOHDACodecAddress field.

2. AudioDevice

Type: plist string

Failsafe: ~~empty string~~

Description: Device path of the specified audio controller for audio support.

Normally this contains builtin analog audio controller (HDEF) device path, e.g. PciRoot(0x0)/Pci(0x1b,0x0). The list of recognised audio controllers can be found in the debug log (marked in bold):

```
OCAU: 1/3 PciRoot(0x0)/Pci(0x1,0x0)/Pci(0x0,0x1)/VenMsg(<redacted>,00000000) (4 outputs)
```

```
OCAU: 2/3 PciRoot(0x0)/Pci(0x3,0x0)/VenMsg(<redacted>,00000000) (1 outputs)
```

```
OCAU: 3/3 PciRoot(0x0)/Pci(0x1B,0x0)/VenMsg(<redacted>,02000000) (7 outputs)
```

As an alternative `gfxutil -f HDEF` command can be used in macOS. Specifying empty device path will result in the first available audio controller to be used.

3. AudioOut

Type: plist integer

Failsafe: 0

Description: Index of the output port of the specified codec starting from 0.

Normally this contains the index of the green out of the builtin analog audio controller (HDEF). The number of output nodes (N) in the debug log (marked in bold):

```
OCAU: 1/3 PciRoot(0x0)/Pci(0x1,0x0)/Pci(0x0,0x1)/VenMsg(<redacted>,00000000) (4 outputs)
```

```
OCAU: 2/3 PciRoot(0x0)/Pci(0x3,0x0)/VenMsg(<redacted>,00000000) (1 outputs)
```

```
OCAU: 3/3 PciRoot(0x0)/Pci(0x1B,0x0)/VenMsg(<redacted>,02000000) (7 outputs)
```

The quickest way to find the right port is to bruteforce the values from 0 to N - 1.

4. AudioSupport

Type: plist boolean

Failsafe: false

Description: Activate audio support by connecting to a backend driver.

Enabling this setting routes audio playback from builtin protocols to a dedicated audio port (AudioOut) of the specified codec (AudioCodec) located on the audio controller (AudioDevice).

5. MinimumVolume

Type: plist integer

Failsafe: 0

Description: Minimal heard volume level from 0 to 100.