



OpenCore

Reference Manual (0.8.~~2~~.3)

[2022.07.08]

If audio appears to be ‘playing’ on the correct codec, e.g. based on the debug log, but no sound is heard on any channel, it is suggested to use `--gpio-setup` (with no value) in the AudioDxe driver arguments. If specified with no value, all stages will be enabled (equivalent of specifying 7). If this produces sound, it is then possible to try fewer bits, e.g. `--gpio-setup=1`, `--gpio-setup=3`, to find out which stages are actually required.

Note: Value 7 (all flags enabled) of this option – as required for the **MacPro5,1** – is compatible with most systems, but is known to cause problems with sound (previous sounds are not allowed to finish before new sounds start) on a small number of other systems, hence this option is not enabled by default.

- `--gpio-pins` - Default: 0, auto-detect.

Specifies which GPIO pins should be operated on by `--gpio-setup`. This is a bit mask, with possible values from 0x0 to 0xFF. The usable maximum depends on the number of available pins on the audio out function group of the codec in use, e.g. it is 0x3 (lowest two bits) if two GPIO pins are present, 0x7 if three pins are present, etc.

When `--gpio-setup` is enabled (i.e. non-zero), then 0 is a special value for `--gpio-pins`, meaning that the pin mask will be auto-generated based on the reported number of GPIO pins on the specified codec (see **AudioCodec**), e.g. if the codec’s audio out function group reports 4 GPIO pins, a mask of 0xF will be used. The value in use can be seen in the debug log in a line such as:

```
HDA: GPIO setup on pins 0x0F - Success
```

Values for driver parameters can be specified in hexadecimal beginning with 0x or in decimal, e.g. `--gpio-pins=0x12` or `--gpio-pins=18`.

- `--restore-nosnoop` - Boolean flag, enabled if present.

AudioDxe clears the Intel HDA No Snoop Enable (NSNPEN) bit. On some systems, this change must be reversed on exit in order to avoid breaking sound in Windows or Linux. If so, this flag should be added to AudioDxe driver arguments. Not enabled by default, since restoring the flag can prevent sound from working in macOS on some other systems.

11.9 Properties

1. APFS

Type: plist dict

Failsafe: None

Description: Provide APFS support as configured in the APFS Properties [section below](#).

2. [AppleInput](#)

Type: [plist dict](#)

Failsafe: [None](#)

Description: [Configure the re-implementation of the Apple Event protocol described in the AppleInput Properties section below](#).

3. Audio

Type: plist dict

Failsafe: None

Description: Configure audio backend support described in the **Audio Properties** section below.

Unless documented otherwise (e.g. **ResetTrafficClass**) settings in this section are for UEFI audio support only (e.g. OpenCore generated boot chime and audio assist) and are unrelated to any configuration needed for OS audio support (e.g. **AppleALC**).

UEFI audio support provides a way for upstream protocols to interact with the selected audio hardware and resources. All audio resources should reside in `\EFI\OC\Resources\Audio` directory. Currently the supported audio file formats are MP3 and WAVE PCM. While it is driver-dependent which audio stream format is supported, most common audio cards support 16-bit signed stereo audio at 44100 or 48000 Hz.

Audio file path is determined by audio type, audio localisation, and audio path. Each filename looks as follows: `[audio type]_[audio localisation]_[audio path].[audio ext]`. For unlocalised files filename does not include the language code and looks as follows: `[audio type]_[audio path].[audio ext]`. Audio extension can either be `mp3` or `wav`.

- Audio type can be `OCEFIAudio` for OpenCore audio files or `AXEFIAudio` for macOS bootloader audio files.