



OpenCore

Reference Manual (0.6.~~8~~.9)

[2021.04.06]

Failsafe: 0

Description: Minimal allowed APFS driver version.

The APFS driver version connects the APFS driver with the macOS release. Apple ultimately drops support for older macOS releases and APFS drivers from such releases may contain vulnerabilities that can be used to compromise a computer if such drivers are used after support ends. This option permits restricting APFS drivers to current macOS versions.

- 0 — require the default supported version of APFS in OpenCore. The default version will increase with time and thus this setting is recommended. Currently set to the latest point release from High Sierra from App Store (748077008000000).
- -1 — permit any version to load (strongly discouraged).
- Other — use custom minimal APFS version, e.g. 1412101001000000 from macOS Catalina 10.15.4. APFS versions can be found in OpenCore boot log and `0cApfsLib`.

11.8 AppleInput Properties

1. AppleEvent

Type: plist string

Failsafe: Auto

Description: Determine whether OC builtin or OEM Apple Event protocol is used.

This option determines whether Apple's OEM Apple Event protocol is used (where available), or whether OpenCore's reversed engineered and updated re-implementation is used. In general OpenCore's re-implementation should be preferred, since it contains updates such as noticeably improved fine mouse cursor movement and configurable key repeat delays.

- ~~Auto — Performs automatic choice of implementation. — Because of optimisations used to achieve fast boot times, this actually means that the OpenCore re-implementation will be found and used except in the case that OpenCore was been explicitly selected and started (not just auto-booted) from Apple's boot picker (where present). Use OEM Apple Event implementation if available, connected and recent enough to be used, otherwise use OC reimplementation. On some Macs (e.g. Mac Pro) this will find the Apple implementation. On both older and newer Macs than this, this option will always or often use the OC implementation. On older Macs this is because the implementation available is too old to be used, on newer Macs it is because of optimisations added by Apple which do not connect the Apple Event protocol except when needed — e.g. except when the Apple boot picker is explicitly started. Due to its somewhat unpredictable results, this option is not normally recommended.~~
- **Builtin** — Use OpenCore's updated re-implementation of the Apple Event protocol. ~~Recommended~~ Using this setting is recommended even on Apple hardware where the OEM version can be used, due to improvements (better fine mouse control, configurable key delays) made in the OC re-implementation of the protocol.
- **OEM** — Assume Apple's protocol will be available at driver connection. This results in Apple's implementation being ~~reliably used on~~ used on most (except the oldest) Apple systems. It results in no keyboard or mouse support otherwise. On all Apple hardware where a recent enough version of the OEM protocol is available — whether connected automatically by Apple's firmware or not — this option will work reliably; nevertheless for the reasons stated Builtin is recommended in preference to this option in most use cases.

2. CustomDelays

Type: plist string

Failsafe: Auto

Description: Enable custom key repeat delays ~~— when using the OpenCore implementation of the Apple Event protocol. Has no effect when using the OEM Apple implementation (see AppleEvent setting).~~

- **Auto** — Treated as **Enabled** when **KeySupport** is **true** and **Disabled** otherwise.
- **Enabled** — The values **KeyInitialDelay** and **KeySubsequentDelay** are used.
- **Disabled** — Apple default values of 500ms (50) and 50ms (5) are used.

3. KeyInitialDelay

Type: plist integer

Failsafe: 0 (no initial delay, immediate repeat)

Description: Configure initial keyboard repeat delay in OpenCore implementation of Apple Event protocol, in units of 10ms.

Configures the initial delay before key repeat. The Apple OEM default value is 50 (500ms).

Note 1: When using **KeySupport**, you may find that you get an additional slow key repeat before normal speed key repeat starts. If so, ~~your~~ the initial key repeat delay is being driven by your BIOS firmware and cannot be overridden by OC (due to technical limitations of how **KeySupport** works, to derive raw key data from the non-raw key data which is all that UEFI makes available). To avoid this minor but undesired effect of two long repeats, you can simply cancel the second, Apple Event, repeat by setting **KeyInitialDelay** to 0. When doing this you should also set **KeySubsequentDelay** to at least the value of your **KeyForgetThreshold** setting (see more information in the **KeySubsequentDelay** setting). ~~The instructions in this note only apply on~~

Note 2: On systems using **KeySupport**, the recommended value is 0. On other systems the recommended value is 50, which may optionally be tuned up or down according to user preference.

4. **KeySubsequentDelay**

Type: plist integer

Failsafe: 1

Description: Configure subsequent keyboard repeat delay in OpenCore implementation of Apple Event protocol, in units of 10ms.

Configures the gap between key repeats. The Apple OEM default value is 5 (50ms). 0 is an invalid value for this option.

Note 1: When using **KeySupport**, you may find that you get one additional slow key repeat before normal speed key repeat starts. If so, set **KeyInitialDelay** to 0 and set **KeySubsequentDelay** to at least the value of your **KeyForgetThreshold** setting. The reason for this is that the key-smoothing parameter **KeyForgetThreshold** effectively acts as the shortest time for which a key can appear to be held, therefore a key repeat delay less than this will guarantee at least one extra repeat for every key press, however quickly the key is physically tapped. (In the unlikely event that you still get frequent, or occasional, double key responses after setting **KeySubsequentDelay** equal to your system's value of **KeyForgetThreshold**, then increase **KeySubsequentDelay** by one or two more until this effect goes away ~~for greatest keyboard responsiveness, use the lowest value which avoids multiple keypresses.~~) ~~The instructions in this note only apply on systems.~~

Note 2: On systems using **KeySupport**, the recommended setting is to use the same value as **KeyForgetThreshold**. This value may then be tuned upwards for slower key repeat if desired, but not downwards. On systems not using **KeySupport** the recommended starting value is 5, and this may optionally be tuned up or down according to user preference.

5. **PointerSpeedDiv**

Type: plist integer

Failsafe: 1)

Description: Configure pointer speed divisor in OpenCore implementation of Apple Event protocol. Has no effect when using the OEM Apple implementation (see **AppleEvent** setting).

Configures the divisor for pointer movements. The Apple OEM default value is 1. 0 is an invalid value for this option.

Note: The recommended value for this option is 1. This value may optionally be modified in combination with **PointerSpeedMul**, according to user preference, to achieve customised mouse movement scaling.

6. **PointerSpeedMul**

Type: plist integer

Failsafe: 0)

Description: Configure pointer speed multiplier in OpenCore implementation of Apple Event protocol. Has no effect when using the OEM Apple implementation (see **AppleEvent** setting).

Configures the multiplier for pointer movements. The Apple OEM default value is 1.

Note: The recommended value for this option is 1. This value may optionally be modified in combination with **PointerSpeedDiv**, according to user preference, to achieve customised mouse movement scaling.

11.9 Audio Properties

1. **AudioCodec**

Type: plist integer

Description: Replaces the Apple Boot Policy protocol with a builtin version. This may be used to ensure APFS compatibility on VMs and legacy Macs.

Note: This option is advisable on certain Macs, such as the MacPro5,1, that are APFS compatible but on which the Apple Boot Policy protocol has recovery detection issues.

3. AppleDebugLog

Type: plist boolean

Failsafe: false

Description: Replaces the Apple Debug Log protocol with a builtin version.—

4. ~~AppleEvent~~~~**Type:** plist boolean~~~~**Failsafe:** false~~~~**Description:** Replaces the Apple Event protocol with a builtin version. This may be used to ensure FileVault 2 compatibility on VMs and legacy Macs.~~

5. AppleFramebufferInfo

Type: plist boolean

Failsafe: false

Description: Replaces the Apple Framebuffer Info protocol with a builtin version. This may be used to override framebuffer information on VMs and legacy Macs to improve compatibility with legacy EfiBoot such as the one in macOS 10.4.

Note: The current implementation of this property results in it only being active when GOP is available (it is always equivalent to **false** otherwise).

6. AppleImageConversion

Type: plist boolean

Failsafe: false

Description: Replaces the Apple Image Conversion protocol with a builtin version.

7. AppleImg4Verification

Type: plist boolean

Failsafe: false

Description: Replaces the Apple IMG4 Verification protocol with a builtin version. This protocol is used to verify im4m manifest files used by Apple Secure Boot.

8. AppleKeyMap

Type: plist boolean

Failsafe: false

Description: Replaces Apple Key Map protocols with builtin versions.

9. AppleRtcRam

Type: plist boolean

Failsafe: false

Description: Replaces the Apple RTC RAM protocol with a builtin version.

Note: Builtin version of Apple RTC RAM protocol may filter out I/O attempts to certain RTC memory addresses. The list of addresses can be specified in `4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:rtc-blacklist` variable as a data array.

10. AppleSecureBoot

Type: plist boolean

Failsafe: false

Description: Replaces the Apple Secure Boot protocol with a builtin version.

11. AppleSmcIo

Type: plist boolean

Failsafe: false

Description: Replaces the Apple SMC I/O protocol with a builtin version.

This protocol replaces the legacy `VirtualSmc` UEFI driver, and is compatible with any SMC kernel extension. However, in case the `FakeSMC` kernel extension is used, manual NVRAM key variable addition may be needed.

12. AppleUserInterfaceTheme

Type: plist boolean