OpenCore

Reference Manual (0.8.~~0~~.1)

[2022.05.15]

2. `AppleXcpmCfgLock`
   Type: `plist boolean`
   Failsafe: `false`
   Requirement: 10.8 (not required for older)
   Description: Disables `PKG_CST_CONFIG_CONTROL` (`0xE2`) MSR modification in XNU kernel, commonly causing early kernel panic, when it is locked from writing (XCPM power management).

   Note: This option should be avoided whenever possible. Refer to the `AppleCpuPmCfgLock` description for details.

3. `AppleXcpmExtraMsrs`
   Type: `plist boolean`
   Failsafe: `false`
   Requirement: 10.8 (not required for older)
   Description: Disables multiple MSR access critical for certain CPUs, which have no native XCPM support.

   This is typically used in conjunction with the `Emulate` section on Haswell-E, Broadwell-E, Skylake-SP, and similar CPUs. More details on the XCPM patches are outlined in acidanthera/bugtracker#365.

   Note: Additional not provided patches will be required for Ivy Bridge or Pentium CPUs. It is recommended to use `AppleIntelCpuPowerManagement.kext` for the former.

4. `AppleXcpmForceBoost`
   Type: `plist boolean`
   Failsafe: `false`
   Requirement: 10.8 (not required for older)
   Description: Forces maximum performance in XCPM mode.

   This patch writes `0xFF00` to `MSR_IA32_PERF_CONTROL` (`0x199`), effectively setting maximum multiplier for all the time.

   Note: While this may increase the performance, this patch is strongly discouraged on all systems but those explicitly dedicated to scientific or media calculations. Only certain Xeon models typically benefit from the patch.

5. `CustomPciSerialDevice`
   Type: `plist boolean`
   Failsafe: `false`
   Requirement: 10.7
   Description: Performs change of PMIO register base address on a customised PCI serial device.

   The patch changes the PMIO register base address that the XNU kernel uses for serial input and output, from that of the default built-in COM1 serial port `0x3F8`, to the base address stored in the first IO BAR of a specified PCI device or to a specific base address (e.g. `0x2F8` for COM2).

   Note: By default, serial logging is disabled. `serial=3` boot argument, which enables serial input and output, should be used for XNU to print logs to the serial port.

   Note 2: In addition to this patch, kext `Apple16X50PCI0` should be prevented from attaching to have `kprintf` method working properly. This can be achieved by setting (i.e. `Delete`, then `Add`) the `class-code` property of the PCI serial port device to `FFFFFFFF` in `DeviceProperties` section. As an alternative solution, a code-less kext `PCIeSerialDisable.kext` shown in the spoiler `PCIeSerialDisable.kext/Contents/Info.plist` at acidanthera/bugtracker#1954, may also be used. In addition, for certain Thunderbolt cards the IOKit personality `IOPCITunnelCompatible` also needs to be set to `true`, which can be done by the `PCIeSerialThunderboltEnable.kext` attached at acidanthera/bugtracker#2003.

   Note 3: For this patch to be correctly applied, `Override` must be enabled with all keys properly set in `Custom`, under section `Misc->Serial`.

   Note 4: This patch is for PMIO support and is therefore not applied if `UseMmio` under section `Misc->Serial->Custom` is false. For MMIO, there are boot arguments `pcie_mmio_uart=ADDRESS` and `mmio_uart=ADDRESS` that allow the kernel to use MMIO for serial port access.

   Note 5: The serial baud rate must be correctly set in both `BaudRate` under section `Misc->Serial->Custom` and via `serialbaud=VALUE` boot argument, both of which should match against each other. The default baud rate is `115200`.

This option enables Aquantia AQtion based 10GbE network cards support, which used to work natively before macOS 10.15.4.

Note: In order for Aquantia cards to properly function, `DisableIoMapper` must be disabled, `DMAR` ACPI table must not be dropped, and `VT-d` must be enabled in BIOS.

Note 2: While this patch should enable ethernet support for all Aquantia AQtion series, it has only been tested on AQC-107s based 10GbE network cards.

Note 3: To address `AppleVTD` incompatibilities after applying this quirk, the `Reserved Memory Region` section of the corresponding device in the `DMAR` ACPI table might be removed. This table should be disassembled and edited, then recompiled to `AML` with tool `iASL`. For the patched `DMAR` table to be added, the original one should be deleted. More details can be found at comment on commit 2441455.

13. `ForceSecureBootScheme`
Type: `plist boolean`
Failsafe: `false`
Requirement: 11
Description: Force `x86` scheme for IMG4 verification.

Note: This option is required on virtual machines when using `SecureBootModel` different from `x86legacy`.

14. `IncreasePciBarSize`
Type: `plist boolean`
Failsafe: `false`
Requirement: 10.10
Description: Allows IOPCIFamily to boot with 2 GB PCI BARs.

Normally macOS restricts PCI BARs to 1 GB. Enabling this option (still) does not let macOS actually use PCI devices with larger BARs.

Note: This option should be avoided whenever possible. A need for this option indicates misconfigured or defective firmware.

15. `LapicKernelPanic`
Type: `plist boolean`
Failsafe: `false`
Requirement: 10.6 (64-bit)
Description: Disables kernel panic on LAPIC interrupts.

16. `LegacyCommpage`
Type: `plist boolean`
Failsafe: `false`
Requirement: 10.4 - 10.6
Description: Replaces the default 64-bit commpage bcopy implementation with one that does not require SSSE3, useful for legacy platforms. This prevents a `commpage no match for last` panic due to no available 64-bit bcopy functions that do not require SSSE3.

17. `PanicNoKextDump`
Type: `plist boolean`
Failsafe: `false`
Requirement: 10.13 (not required for older)
Description: Prevent kernel from printing kext dump in the panic log preventing from observing panic details. Affects 10.13 and above.

18. `PowerTimeoutKernelPanic`
Type: `plist boolean`
Failsafe: `false`
Requirement: 10.15 (not required for older)
Description: Disables kernel panic on setPowerState timeout.

An additional security measure was added to macOS Catalina (10.15) causing kernel panic on power change timeout for Apple drivers. Sometimes it may cause issues on misconfigured hardware, notably digital audio, which

(g) As the `Default` value will increase with time to support the latest major released operating system, it is not recommended to use the `ApECID` and the `Default` settings together.
(h) Installing macOS with Apple Secure Boot enabled is not possible while using HFS+ target volumes. This may include HFS+ formatted drives when no spare APFS drive is available.

The installed operating system may have sometimes outdated Apple Secure Boot manifests on the `Preboot` partition, resulting in boot failures. This is likely to be the case when an "OCB: Apple Secure Boot prohibits this boot entry, enforcing!" message is logged.

When this happens, either reinstall the operating system or copy the manifests (files with `.im4m` extension, such as `boot.efi.j137.im4m`) from `/usr/standalone/i386` to `/Volumes/Preboot/<UUID>/System/Library/CoreServices`. Here, `<UUID>` is the system volume identifier. On HFS+ installations, the manifests should be copied to `/System/Library/CoreServices` on the system volume.

For more details on how to configure Apple Secure Boot with UEFI Secure Boot, refer to the UEFI Secure Boot section.

8.6 Serial Properties

1. `Custom`
   Type: `plist dict`
   Description: Update serial port properties in `BaseSerialPortLib16550`.

   This section lists the PCD values that are used by the `BaseSerialPortLib16550`. When option `Override` is set to `false`, this dictionary is optional.

2. `Init`
   Type: `plist boolean`
   Failsafe: `false`
   Description: Perform serial port initialisation.

   This option will perform serial port initialisation within OpenCore prior to enabling (any) debug logging.

   Refer to the `Debugging` section for details.

3. `Override`
   Type: `plist boolean`
   Failsafe: `false`
   Description: Override serial port properties. When this option is set to `false`, no keys from `Custom` will be overridden.

   This option will override serial port properties listed in the `Serial Custom Properties` section below.

8.6.1 Serial Custom Properties

1. `BaudRate`
   Type: `plist integer`
   Failsafe: `115200`
   Description: Set the baud rate for serial port.

   This option will override the value of `gEfiMdeModulePkgTokenSpaceGuid.PcdSerialBaudRate` defined in MdeModulePkg.dec.

2. `ClockRate`
   Type: `plist integer`
   Failsafe: `1843200`
   Description: Set the clock rate for serial port.

   This option will override the value of `gEfiMdeModulePkgTokenSpaceGuid.PcdSerialClockRate` defined in MdeModulePkg.dec.

3. `DetectCable`
   Type: `plist boolean`

Failsafe: `false`
Description: Enable serial port cable detection.

This option will override the value of `gEfiMdeModulePkgTokenSpaceGuid.PcdSerialDetectCable` defined in MdeModulePkg.dec.

4. `ExtendedTxFifoSize`
   Type: `plist integer`
   Failsafe: `64`
   Description: Set the extended transmit FIFO size for serial port.

   This option will override the value of `gEfiMdeModulePkgTokenSpaceGuid.PcdSerialExtendedTxFifoSize` defined in MdeModulePkg.dec.

5. `FifoControl`
   Type: `plist integer`
   Failsafe: `0x07`
   Description: Configure serial port FIFO Control settings.

   This option will override the value of `gEfiMdeModulePkgTokenSpaceGuid.PcdSerialFifoControl` defined in MdeModulePkg.dec.

6. `LineControl`
   Type: `plist integer`
   Failsafe: `0x07`
   Description: Configure serial port Line Control settings.

   This option will override the value of `gEfiMdeModulePkgTokenSpaceGuid.PcdSerialLineControl` defined in MdeModulePkg.dec.

7. `PciDeviceInfo`
   Type: `plist data`
   Failsafe: `0xFF`
   Description: Set PCI serial device information.

   This option will override the value of `gEfiMdeModulePkgTokenSpaceGuid.PcdSerialPciDeviceInfo` defined in MdeModulePkg.dec.

   Note: The maximum allowed size of this option is 41 bytes. Refer to acidanthera/bugtracker#1954 for more details.

   Note 2: This option can be set by running the `FindSerialPort` tool.

8. `RegisterAccessWidth`
   Type: `plist integer`
   Failsafe: `8`
   Description: Set serial port register access width.

   This option will override the value of `gEfiMdeModulePkgTokenSpaceGuid.PcdSerialRegisterAccessWidth` defined in MdeModulePkg.dec.

9. `RegisterBase`
   Type: `plist integer`
   Failsafe: `0x03F8`
   Description: Set the base address of serial port registers.

   This option will override the value of `gEfiMdeModulePkgTokenSpaceGuid.PcdSerialRegisterBase` defined in MdeModulePkg.dec.

10. `RegisterStride`
    Type: `plist integer`
    Failsafe: `1`
    Description: Set the serial port register stride in bytes.

    This option will override the value of `gEfiMdeModulePkgTokenSpaceGuid.PcdSerialRegisterStride` defined in MdeModulePkg.dec.

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 acidanthera/bugtracker#740 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 `ReconnectGraphicsOnConnect`. 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`. This is a modified version of `CrScreenshotDxe` driver by Nikolaj Schlej.
`ExFatDxe`	Proprietary ExFAT file system driver for Bootcamp support commonly found in Apple firmware. For Sandy Bridge and earlier CPUs, the `ExFatDxeLegacy` 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.
`HfsPlus`	Recommended. Proprietary HFS file system driver with bless support commonly found in Apple firmware. For Sandy Bridge and earlier CPUs, the `HfsPlusLegacy` driver should be used due to the lack of `RDRAND` instruction support.
`HiiDatabase*`	HII services support driver from `MdeModulePkg`. 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 `FatPkg`. 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 `MdeModulePkg`. 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 `OC_FIRMWARE_RUNTIME` protocol.
`OpenLinuxBoot*`	OpenCore plugin implementing `OC_BOOT_ENTRY_PROTOCOL` to allow direct detection and booting of Linux distributiuons from OpenCore, without chainloading via GRUB.
`OpenNtfsDxe*`	New Technologies File System (NTFS) read-only driver. NTFS is the primary file system for Microsoft Windows versions that are based on Windows NT.
`OpenUsbKbDxe*`	USB keyboard driver adding support for `AppleKeyMapAggregator` protocols on top of a custom USB keyboard driver implementation. This is an alternative to builtin `KeySupport`, which may work better or worse depending on the firmware.
`OpenPartitionDxe*`	Partition management driver with Apple Partitioning Scheme support. This driver can be used to support loading older DMG recoveries such as macOS 10.9 using Apple Partitioning Scheme. OpenDuet already includes this driver.

On certain firmware, the controllers that produce the console protocols (simple text out) must be reconnected when the screen resolution is changed via GOP. Otherwise, they will not produce text based on the new resolution.

Note: On several boards this logic may result in black screen when launching OpenCore from Shell and thus it is optional. In versions prior to 0.5.2 this option was mandatory and not configurable. Please do not use this unless required.

13. `SanitiseClearScreen`
Type: `plist boolean`
Failsafe: `false`
Description: Some types of firmware reset screen resolutions to a failsafe value (such as `1024x768`) on the attempts to clear screen contents when large display (e.g. 2K or 4K) is used. This option attempts to apply a workaround.

Note: This option only applies to the `System` renderer. On all known affected systems, `ConsoleMode` must be set to an empty string for this option to work.

14. `UIScale`
Type: `plist integer`, 8 bit
Failsafe: `-1`
Description: User interface scaling factor.

Corresponds to `4D1EDE05-38C7-4A6A-9CC6-4BCCA8B38C14:UIScale` variable.

- `1` — 1x scaling, corresponds to normal displays.
- `2` — 2x scaling, corresponds to HiDPI displays.
- `-1` — leaves the current variable unchanged.
- `0` — automatically chooses scaling based on the current resolution.

Note 1: Automatic scale factor detection works on the basis of total pixel area and may fail on small HiDPI displays, in which case the value may be manually managed using the NVRAM section.

Note 2: When switching from manually specified NVRAM variable to this preference an NVRAM reset may be needed.

15. `UgaPassThrough`
Type: `plist boolean`
Failsafe: `false`
Description: Provide UGA protocol instances on top of GOP protocol instances.

Some types of firmware do not implement the legacy UGA protocol but this may be required for screen output by older EFI applications such as EfiBoot from 10.4.

11.15 ProtocolOverrides Properties

1. `AppleAudio`
Type: `plist boolean`
Failsafe: `false`
Description: Replaces Apple audio protocols with builtin versions.

Apple audio protocols allow OpenCore and the macOS bootloader to play sounds and signals for screen reading or audible error reporting. Supported protocols are beep generation and VoiceOver. The VoiceOver protocol is ~~specific to~~ only provided natively by Gibraltar machines (T2)~~and~~, however versions of macOS which support VoiceOver will see and use the implementation provided by OpenCore, on screens such as FileVault 2 unlock. VoiceOver is not supported before macOS High Sierra (10.13). Older macOS versions use the AppleHDA protocol (which is not currently implemented) instead.

Only one set of audio protocols can be available at a time, so this setting should be enabled in order to enable audio playback in the OpenCore user interface on Mac systems implementing some of these protocols.

Note: The backend audio driver needs to be configured in `UEFI Audio` section for these protocols to be able to stream audio.

2. `AppleBootPolicy`
Type: `plist boolean`