



# OpenCore

Reference Manual (0.8.~~8~~.9)

[2023.01.08]

The OC config file, as with any property list file, can be edited with any text editor, such as nano or vim. However, specialised software may provide a better experience. On macOS, the preferred GUI application is Xcode. The ProperTree editor is a lightweight, cross-platform and open-source alternative.

It is strongly recommended to avoid configuration creation tools that are aware of the internal configuration structure as this may result in invalid configurations (since the structure gets constantly updated). If such tools are to be used despite this warning, ensure that only stable versions of OpenCore explicitly supported by such tools are used. In such cases, the use of open-source implementations with transparent binary generation (such as OCAT) is encouraged, given that other tools may contain malware. In addition, configurations created for a specific hardware setup should never be used on different hardware setups.

For BIOS booting, a third-party UEFI environment provider is required and `OpenDuetPkg` is one such UEFI environment provider for legacy systems. To run OpenCore on such a legacy system, `OpenDuetPkg` can be installed with a dedicated tool — `BootInstall` (bundled with OpenCore). Third-party utilities can be used to perform this on systems other than macOS.

For upgrade purposes, refer to the `Differences.pdf` document which provides information about changes to the configuration (as compared to the previous release) as well as to the `Changelog.md` document (which contains a list of modifications across all published updates).

### 3.3 Contribution

OpenCore can be compiled as a standard EDK II package and requires the EDK II Stable package. The currently supported EDK II release is hosted in `acidanthera/audk`. Required patches for this package can be found in the `Patches` directory.

When updating the LaTeX documentation (e.g. `Configuration.tex`) please do *not* rebuild the PDF files till merging to master happens. This avoids unnecessary merge conflicts:

- External contributors using the pull-request approach should request the maintainers to handle the PDF rebuild in the pull-request message.
- Internal contributors should rebuild the documentation at merge time in the same or in a separate commit. One can ask another maintainer to rebuild the documentation when lacking the necessary tools in the pull-request message.

The only officially supported toolchain is `XCODE5`. Other toolchains might work but are neither supported nor recommended. Contributions of clean patches are welcome. Please do follow EDK II C Codestyle.

To compile with `XCODE5`, besides Xcode, users should also install NASM and MTOC. The latest Xcode version is recommended for use despite the toolchain name. An example command sequence is as follows:

---

```
git clone --depth=1 https://github.com/acidanthera/audk UDK
cd UDK
git submodule update --init --recommend-shallow
rm -rf OpenCorePkg
git clone --depth=1 https://github.com/acidanthera/OpenCorePkg
. ./edksetup.sh
make -C BaseTools
build -a X64 -b RELEASE -t XCODE5 -p OpenCorePkg/OpenCorePkg.dsc
```

---

Listing 1: Compilation Commands

For IDE usage Xcode projects are available in the root of the repositories. Another approach could be using Language Server Protocols. For example, Sublime Text with LSP for Sublime Text plugin. Add `compile_flags.txt` file with similar content to the UDK root:

---

```
-I/UefiPackages/MdePkg
-I/UefiPackages/MdePkg/Include
-I/UefiPackages/MdePkg/Include/X64
-I/UefiPackages/MdeModulePkg
-I/UefiPackages/MdeModulePkg/Include
-I/UefiPackages/MdeModulePkg/Include/X64
```

*Note 1:* This quirk shall not be used to workaround macOS limitation to address BARs over 1 GB. `ResizeAppleGpuBars` should be used instead.

*Note 2:* While this quirk can increase GPU PCI BAR sizes, this will not work on most firmware as is, because the quirk does not relocate BARs in memory, and they will likely overlap. ~~Contributions to improve this feature are welcome~~In most cases it is best to either update the firmware to the latest version or customise it with a specialised driver like ReBarUEFI.

### 13. `TscSyncTimeout`

**Type:** plist integer

**Failsafe:** 0

**Description:** Attempts to perform TSC synchronisation with a specified timeout.

The primary purpose of this quirk is to enable early bootstrap TSC synchronisation on some server and laptop models when running a debug XNU kernel. For the debug kernel the TSC needs to be kept in sync across the cores before any kext could kick in rendering all other solutions problematic. The timeout is specified in microseconds and depends on the amount of cores present on the platform, the recommended starting value is 500000.

This is an experimental quirk, which should only be used for the aforementioned problem. In all other cases, the quirk may render the operating system unstable and is not recommended. The recommended solution in the other cases is to install a kernel extension such as `VoodooTSCSync`, `TSCAdjustReset`, or `CpuTscSync` (a more specialised variant of `VoodooTSCSync` for newer laptops).

*Note:* This quirk cannot replace the kernel extension because it cannot operate in ACPI S3 (sleep wake) mode and because the UEFI firmware only provides very limited multicore support which prevents precise updates of the MSR registers.

### 14. `UnblockFsConnect`

**Type:** plist boolean

**Failsafe:** false

**Description:** Some types of firmware block partition handles by opening them in `By Driver` mode, resulting in an inability to install File System protocols.

*Note:* This quirk is useful in cases where unsuccessful drive detection results in an absence of boot entries.

## 11.19 ReservedMemory Properties

### 1. Address

**Type:** plist integer

**Failsafe:** 0

**Description:** Start address of the reserved memory region, which should be allocated as reserved effectively marking the memory of this type inaccessible to the operating system.

The addresses written here must be part of the memory map, have a `EfiConventionalMemory` type, and be page-aligned (4 KBs).

*Note:* Some types of firmware may not allocate memory areas used by S3 (sleep) and S4 (hibernation) code unless CSM is enabled causing wake failures. After comparing the memory maps with CSM disabled and enabled, these areas can be found in the lower memory and can be fixed up by doing the reservation. Refer to the `Sample.plist` file for details.

### 2. Comment

**Type:** plist string

**Failsafe:** Empty

**Description:** Arbitrary ASCII string used to provide human readable reference for the entry. Whether this value is used is implementation defined.

### 3. Size

**Type:** plist integer

**Failsafe:** 0

**Description:** Size of the reserved memory region, must be page-aligned (4 KBs).

### 4. Type

**Type:** plist string