OpenCore

Reference Manual (0.7.~~0~~.1)

[2021.06.19]

2. `NormalizeHeaders`
   Type: `plist boolean`
   Failsafe: `false`
   Description: Cleanup ACPI header fields to workaround macOS ACPI implementation flaws that result in boot crashes. Reference: Debugging AppleACPIPlatform on 10.13 by Alex James (also known as theracermaster). The issue was fixed in macOS Mojave (10.14).

3. `RebaseRegions`
   Type: `plist boolean`
   Failsafe: `false`
   Description: Attempt to heuristically relocate ACPI memory regions. Not recommended.

   ACPI tables are often generated dynamically by the underlying firmware implementation. Among the position-independent code, ACPI tables may contain the physical addresses of MMIO areas used for device configuration, typically grouped by region (e.g. `OperationRegion`). Changing firmware settings or hardware configuration, upgrading or patching the firmware inevitably leads to changes in dynamically generated ACPI code, which sometimes results in the shift of the addresses in the aforementioned `OperationRegion` constructions.

   For this reason, the application of modifications to ACPI tables is extremely risky. The best approach is to make as few changes as possible to ACPI tables and to avoid replacing any tables, particularly DSDT tables. When this cannot be avoided, ensure that any custom DSDT tables are based on the most recent DSDT tables or attempt to remove reads and writes for the affected areas.

   When nothing else helps, this option could be tried to avoid stalls at `PCI Configuration Begin` phase of macOS booting by attempting to fix the ACPI addresses. It is not a magic bullet however, and only works with the most typical cases. Do not use unless absolutely required as it can have the opposite effect on certain platforms and result in boot failures.

4. `ResetHwSig`
   Type: `plist boolean`
   Failsafe: `false`
   Description: Reset `FACS` table `HardwareSignature` value to `0`.

   This works around firmware that fail to maintain hardware signature across the reboots and cause issues with waking from hibernation.

5. `ResetLogoStatus`
   Type: `plist boolean`
   Failsafe: `false`
   Description: Reset `BGRT` table `Displayed` status field to `false`.

   This works around firmware that provide a `BGRT` table but fail to handle screen updates afterwards.

6. `SyncTableIds`
   Type: `plist boolean`
   Failsafe: `false`
   Description: Sync table identifiers with the `SLIC` table.

   This works around patched tables becoming incompatible with the `SLIC` table causing licensing issues in older Windows operating systems.

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. `CustomSMBIOSGuid`
Type: `plist boolean`
Failsafe: `false`
Requirement: 10.4
Description: Performs GUID patching for `UpdateSMBIOSMode Custom` mode. Usually relevant for Dell laptops.

6. `DisableIoMapper`
Type: `plist boolean`
Failsafe: `false`
Requirement: 10.8 (not required for older)
Description: Disables `IOMapper` support in XNU (VT-d), which may conflict with the firmware implementation.

Note 1: This option is a preferred alternative to deleting `DMAR` ACPI table and disabling VT-d in firmware preferences, which does not obstruct VT-d support in other systems in case they need this.

Note 2: Misconfigured IOMMU in the firmware may result in broken devices such as ethernet or Wi-Fi adapters. For instance, an ethernet adapter may cycle in link-up link-down state infinitely and a Wi-Fi adapter may fail to discover networks. Gigabyte is one of the most common OEMs with these issues.

7. `DisableLinkeditJettison`
Type: `plist boolean`
Failsafe: `false`
Requirement: 11
Description: Disables `__LINKEDIT` jettison code.

This option lets `Lilu.kext`, and possibly other kexts, function in macOS Big Sur at their best performance levels without requiring the `keepsyms=1` boot argument.

8. `DisableRtcChecksum`
Type: `plist boolean`
Failsafe: `false`
Requirement: 10.4
Description: Disables primary checksum (`0x58-0x59`) writing in AppleRTC.

Note 1: This option will not protect other areas from being overwritten, see RTCMemoryFixup kernel extension if this is desired.

Note 2: This option will not protect areas from being overwritten at firmware stage (e.g. macOS bootloader), see `AppleRtcRam` protocol description if this is desired.

9. `ExtendBTFeatureFlags`
Type: `plist boolean`
Failsafe: `false`
Requirement: 10.8-11
Description: Set `FeatureFlags` to `0x0F` for full functionality of Bluetooth, including Continuity.

Note: This option is a substitution for BT4LEContinuityFixup.kext, which does not function properly due to late patching progress.

10. `ExternalDiskIcons`
Type: `plist boolean`
Failsafe: `false`
Requirement: 10.4
Description: Apply icon type patches to AppleAHCIPort.kext to force internal disk icons for all AHCI disks.

11.3 Tools and Applications

Standalone tools may help to debug firmware and hardware. Some of the known tools are listed below. While some tools can be launched from within OpenCore (Refer to the Tools subsection for more details), most should be run separately either directly or from `Shell`.

To boot into OpenShell or any other tool directly save `OpenShell.efi` under the name of `EFI\BOOT\BOOTX64.EFI` on a FAT32 partition. It is typically unimportant whether the partition scheme is `GPT` or `MBR`.

While the previous approach works both on Macs and other computers, an alternative Mac-only approach to bless the tool on an HFS+ or APFS volume:

```
sudo bless --verbose --file /Volumes/VOLNAME/DIR/OpenShell.efi \
  --folder /Volumes/VOLNAME/DIR/ --setBoot
```

Listing 3: Blessing tool

Note 1: `/System/Library/CoreServices/BridgeVersion.bin` should be copied to `/Volumes/VOLNAME/DIR`.
Note 2: To be able to use the `bless` command, disabling System Integrity Protection is necessary.
Note 3: To be able to boot Secure Boot might be disabled if present.

Some of the known tools are listed below (builtin tools are marked with *):

`BootKicker*`	Enter Apple BootPicker menu (exclusive for Macs with compatible GPUs).
`ChipTune*`	Test BeepGen protocol and generate audio signals of different style and length.
`CleanNvram*`	Reset NVRAM alternative bundled as a standalone tool.
`CsrUtil*`	Simple implementation of SIP-related features of Apple `csrutil`.
`GopStop*`	Test GraphicsOutput protocol with a simple scenario.
`KeyTester*`	Test keyboard input in `SimpleText` mode.
`MemTest86`	Memory testing utility.
`OpenControl*`	Unlock and lock back NVRAM protection for other tools to be able to get full NVRAM access when launching from OpenCore.
`OpenShell*`	OpenCore-configured `UEFI Shell` for compatibility with a broad range of firmware.
`PavpProvision`	Perform EPID provisioning (requires certificate data configuration).
`ResetSystem*`	Utility to perform system reset. Takes reset type as an argument: `coldreset`, `firmware`, `shutdown`, `warmreset`. Defaults to `coldreset`.
`RtcRw*`	Utility to read and write RTC (CMOS) memory.
`ControlMsrE2*`	Check `CFG Lock` (MSR `0xE2` write protection) consistency across all cores and change such hidden options on selected platforms.

11.4 OpenCanopy

OpenCanopy is a graphical OpenCore user interface that runs in `External PickerMode` and relies on OpenCorePkg `OcBootManagementLib` similar to the builtin text interface.

OpenCanopy requires graphical resources located in `Resources` directory to run. Sample resources (fonts and images) can be found in OcBinaryData repository. Customised icons can be found over the internet (e.g. here or there).

OpenCanopy provides full support for `PickerAttributes` and offers a configurable builtin icon set. The chosen icon set may depend on the `DefaultBackgroundColor` variable value. Refer to `PickerVariant` for more details.

Predefined icons are saved in the `PickerVariant`-derived subdirectory of the `\EFI\OC\Resources\Image` directory. A full list of supported icons (in `.icns` format) is provided below. When optional icons are missing, the closest available icon will be used. External entries will use `Ext`-prefixed icon if available (e.g. `OldExtHardDrive.icns`).

Note: In the following all dimensions are normative for the 1x scaling level and shall be scaled accordingly for other levels.

- `Cursor` — Mouse cursor (mandatory, up to 144x144).
- `Selected` — Selected item (mandatory, 144x144).
- `Selector` — Selecting item (mandatory, up to 144x40).
- `SetDefault` — Selecting default (mandatory, up to 144x40; must be same width as `Selector`).
- `Left` — Scrolling left (mandatory, 40x40).