



# OpenCore

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

[2021.06.11]

## 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

**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.