



OpenCore

Reference Manual (0.6.~~3~~.4)

[2020.11.04]

Note 1: It is known that some Lenovo laptops have a firmware bug, which makes them unbootable after performing NVRAM reset. See [acidanthera/bugtracker#995](#) for more details.

Note 2: Resetting NVRAM will also erase all the boot options otherwise not backed up with bless (e.g. Linux).

2. AllowSetDefault

Type: plist boolean

Failsafe: false

Description: Allow CTRL+Enter and CTRL+Index handling to set the default boot option in boot picker.

3. ApECID

Type: plist integer, 64 bit

Failsafe: 0

Description: Apple Enclave Identifier.

Setting this value to any non-zero 64-bit integer will allow using personalised Apple Secure Boot identifiers. To use this setting, make sure to generate a random 64-bit number with a cryptographically secure random number generator. As an alternative, first 8 bytes of `SystemUUID` can be used for `ApECID`, this is found in macOS 11.0 for Macs without the T2 chip.

With this value set and `SecureBootModel` valid and not `Disabled` it is possible to achieve `Full Security` of Apple Secure Boot.

To start using personalised Apple Secure Boot, the operating system will have to be reinstalled or personalised. Unless the operating system is personalised, macOS DMG recovery cannot be loaded. If DMG recovery is missing, it can be downloaded with `macrecovery` utility and put to `com.apple.recovery.boot` as explained in Tips and Tricks section. Note that DMG loading needs to be set to `Signed` to use any DMG with Apple Secure Boot.

To personalise an existing operating system use `bless` command after loading to macOS DMG recovery. Mount the system volume partition, unless it has already been mounted, and execute the following command:

```
bless bless --folder "/Volumes/Macintosh HD/System/Library/CoreServices" \  
--bootefi --personalize
```

Before macOS 11.0, which introduced a dedicated `x86legacy` model for models without the T2 chip, personalised Apple Secure Boot may not work as expected. When reinstalling the operating system, macOS Installer from macOS 10.15 and older, will usually run out of free memory on the `/var/tmp` partition when trying to install macOS with the personalised Apple Secure Boot. Soon after downloading the macOS installer image an `Unable to verify macOS` error message will appear. To workaround this issue allocate a dedicated RAM disk of 2 MBs for macOS personalisation by entering the following commands in macOS recovery terminal before starting the installation:

```
disk=$(hdiutil attach -nomount ram://4096)  
diskutil erasevolume HFS+ SecureBoot $disk  
diskutil unmount $disk  
mkdir /var/tmp/OSPersonalizationTemp  
diskutil mount -mountpoint /var/tmp/OSPersonalizationTemp $disk
```

4. AuthRestart

Type: plist boolean

Failsafe: false

Description: Enable `VirtualSMC`-compatible authenticated restart.

Authenticated restart is a way to reboot FileVault 2 enabled macOS without entering the password. A dedicated terminal command can be used to perform authenticated restarts: `sudo fdesetup authrestart`. It is also used when installing operating system updates.

`VirtualSMC` performs authenticated restart by saving disk encryption key split in NVRAM and RTC, which despite being removed as soon as OpenCore starts, may be considered a security risk and thus is optional.

5. [BlacklistAppleUpdate](#)

Type: [plist boolean](#)

Failsafe: `false`

Description: Ignore boot options trying to update Apple peripheral firmware (e.g. `MultiUpdater.efi`).

Note: This option exists due to some operating systems, namely macOS Big Sur, being incapable of disabling firmware updates with the NVRAM variable (`run-efi-updater`).

6. `BootProtect`

Type: `plist string`

Failsafe: `None`

Description: Attempt to provide bootloader persistence.

Valid values:

- `None` — do nothing.
- `Bootstrap` — create or update top-priority `\EFI\OC\Bootstrap\Bootstrap.efi` boot option (Boot9696) in UEFI variable storage at bootloader startup. For this option to work `RequestBootVarRouting` is required to be enabled.

This option provides integration with third-party operating system installation and upgrade at the times they overwrite `\EFI\BOOT\BOOTx64.efi` file. By creating a custom option in `Bootstrap` mode this file path becomes no longer used for bootstrapping `OpenCore`.

Note 1: Some types of firmware may have faulty NVRAM, no boot option support, or other incompatibilities. While unlikely, the use of this option may even cause boot failures. This option should be used without any warranty exclusively on the boards known to be compatible. Check [acidanthera/bugtracker#1222](#) for some known issues with Haswell and other boards.

Note 2: Be aware that while NVRAM reset executed from `OpenCore` should not erase the boot option created in `Bootstrap`, executing NVRAM reset prior to loading `OpenCore` will remove it.

7. `DmgLoading`

Type: `plist string`

Failsafe: `Signed`

Description: Define Disk Image (DMG) loading policy used for macOS Recovery.

Valid values:

- `Disabled` — loading DMG images will fail. `Disabled` policy will still let macOS Recovery to load in most cases as there usually are `boot.efi` files compatible with Apple Secure Boot. Manually downloaded DMG images stored in `com.apple.recovery.boot` directories will not load, however.
- `Signed` — only Apple-signed DMG images will load. Due to Apple Secure Boot design `Signed` policy will let any Apple-signed macOS Recovery to load regardless of Apple Secure Boot state, which may not always be desired.
- `Any` — any DMG images will mount as normal filesystems. `Any` policy is strongly not recommended and will cause a boot failure when Apple Secure Boot is activated.

8. `EnablePassword`

Type: `plist boolean`

Failsafe: `false`

Description: Enable password protection to allow sensitive operations.

Password protection ensures that sensitive operations such as booting a non-default operating system (e.g. macOS recovery or a tool), resetting NVRAM storage, trying to boot into a non-default mode (e.g. verbose mode or safe mode) are not allowed without explicit user authentication by a custom password. Currently password and salt are hashed with 5000000 iterations of SHA-512.

Note: This functionality is currently in development and is not ready for daily usage.

9. `ExposeSensitiveData`

Type: `plist integer`

Failsafe: `0x6`

Description: Sensitive data exposure bitmask (sum) to operating system.

- `0x01` — Expose printable booter path as an UEFI variable.
- `0x02` — Expose `OpenCore` version as an UEFI variable.

- `4D1EDE05-38C7-4A6A-9CC6-4BCCA8B38C14:HW_BID`
Hardware BoardProduct (e.g. `Mac-35C1E88140C3E6CF`). Not present on real Macs, but used to avoid extra parsing of SMBIOS tables, especially in `boot.efi`.
- `4D1EDE05-38C7-4A6A-9CC6-4BCCA8B38C14:HW_MLB`
Hardware BoardSerialNumber. Override for MLB. Present on newer Macs (2013+ at least).
- `4D1EDE05-38C7-4A6A-9CC6-4BCCA8B38C14:HW_ROM`
Hardware ROM. Override for ROM. Present on newer Macs (2013+ at least).
- `7C436110-AB2A-4BBB-A880-FE41995C9F82:prev-lang:kbd`
ASCII string defining default keyboard layout. Format is `lang-COUNTRY:keyboard`, e.g. `ru-RU:252` for Russian locale and ABC keyboard. Also accepts short forms: `ru:252` or `ru:0` (U.S. keyboard, compatible with 10.9). Full decoded keyboard list from `AppleKeyboardLayouts-L.dat` can be found [here](#). Using non-latin keyboard on 10.14 will not enable ABC keyboard, unlike previous and subsequent macOS versions, and is thus not recommended in case 10.14 is needed.
- `7C436110-AB2A-4BBB-A880-FE41995C9F82:security-mode`
ASCII string defining FireWire security mode. Legacy, can be found in `IOFireWireFamily` source code in `IOFireWireController.cpp`. It is recommended not to set this variable, which may speedup system startup. Setting to `full` is equivalent to not setting the variable and `none` disables FireWire security.
- `4D1EDE05-38C7-4A6A-9CC6-4BCCA8B38C14:UIScale`
One-byte data defining `boot.efi` user interface scaling. Should be `01` for normal screens and `02` for HiDPI screens.
- `4D1EDE05-38C7-4A6A-9CC6-4BCCA8B38C14:DefaultBackgroundColor`
Four-byte BGRA data defining `boot.efi` user interface background colour. Standard colours include `BF BF BF 00` (Light Gray) and `00 00 00 00` (Syrah Black). Other colours may be set at user's preference.

9.5 Other Variables

The following variables may be useful for certain configurations or troubleshooting:

- `7C436110-AB2A-4BBB-A880-FE41995C9F82:boot-args`
Kernel arguments, used to pass configuration to Apple kernel and drivers. There are many arguments, which may be found by looking for the use of `PE_parse_boot_argn` function in the kernel or driver code. Some of the known boot arguments include:
 - `acpi_layer=0xFFFFFFFF`
 - `acpi_level=0xFFFFF5F` (implies `ACPI_ALL_COMPONENTS`)
 - `arch=i386` (force kernel architecture to `i386`, see `KernelArch`)
 - `batman=VALUE` (`AppleSmartBatteryManager` debug mask)
 - `batman-nosmc=1` (disable `AppleSmartBatteryManager` SMC interface)
 - `cpus=VALUE` (maximum number of CPUs used)
 - `debug=VALUE` (debug mask)
 - `io=VALUE` (`IOKit` debug mask)
 - `keepsyms=1` (show panic log debug symbols)
 - `kextlog=VALUE` (kernel extension loading debug mask)
 - `nvram-log=1` ([enables AppleEFINVRAM logs](#))
 - `nv_disable=1` (disables NVIDIA GPU acceleration)
 - `nvda_drv=1` (legacy way to enable NVIDIA web driver, removed in 10.12)
 - `npci=0x2000` (legacy, disables `kIOPCIConfiguratorPFM64`)
 - `lapic_dont_panic=1`
 - `slide=VALUE` (manually set KASLR slide)
 - `smcdebug=VALUE` (`AppleSMC` debug mask)
 - `-amd_no_dgpu_accel` (alternative to WhateverGreen's `-radvesa` for new GPUs)
 - `-nehalem_error_disable`
 - `-no_compat_check` (disable model checking on 10.7+)
 - `-s` (single mode)
 - `-v` (verbose mode)
 - `-x` (safe mode)

There are multiple external places summarising macOS argument lists: [example 1](#), [example 2](#).

- `7C436110-AB2A-4BBB-A880-FE41995C9F82:bootercfg`
Booter arguments, similar to `boot-args` but for `boot.efi`. Accepts a set of arguments, which are hexadecimal