



Home / Browse / Clover EFI bootloader / Wiki



Clover EFI bootloader

Boot macOS, Windows, and Linux on Mac or PC with UEFI or BIOS firmware
Brought to you by: apianti, slice2009

This project can now be found
here.

Boot



[Home](#) > [Configuration](#) > [Boot](#)

Table of Contents

- [Arguments](#)
- [Legacy](#)
- [Timeout](#)
- [DefaultVolume](#)
- [DefaultLoader](#)
- [IgnoreNVRAMBoot](#)
- [Debug](#)
- [Fast](#)
- [XMPDetection](#)
- [Secure](#)
- [NeverHibernate](#)
- [CustomLogo](#)
- [HibernationFixup](#)
- [RtcHibernateAware](#)
- [SignatureFixup](#)

Arguments

```
<key>Arguments</key>
<string>ARGUMENTS</string>
```

At this section you can add "Boot Flags" and "Kernel Flags" to be used by the system... Here we will list only Clover's proprietary "Boot Flags", different flags, like for example, npci=0x2000, npci=0x3000, darkwake=0, etc..., should work as expected...

- `-v` - Verbose Boot.
- `-s` - Boot OS X into Single User Mode.
- `-x` - Boot OS X into SafeBoot (Safe Mode).
- `MountEFI=yes/diskX` - Mount EFI partition at every boot or from disk X (X = disk number).
- `LogLineCount=0` - Set maximum number of lines for log file, default 0 (no Limit).
- `LogLocation=PATH` - Set the path for the log files to be saved.
- `LogEveryBoot=Yes/No` - Save log files at every boot.

Legacy

```
<key>Legacy</key>
<string>PBR</string>
```

Necessary for loading old versions of Windows and Linux. Greatly depends on hardware and BIOS. Several algorithms were developed to allow customisation:

- `LegacyBiosDefault` - for UEFI BIOS variants containing protocol *LegacyBios*
- `PBRtest` - PBR boot algorithm
- `PBR` - Another PBR boot algorithm

Get latest updates about
Open Source Projects,
Conferences and News.

[Sign Up](#)

No, Thank you

Additionally for UEFI boot you may specify which HDD to boot (not only the first one).

```
<key>Legacy</key>
<string>LegacyBiosDefault</string>
<key>LegacyBiosDefaultEntry</key>
<integer>2</integer>
```

Timeout

```
<key>Timeout</key>
<integer>5</integer>
```

The boot loader will pause for 5 seconds after starting before automatically loading an operating system. If a key is pressed during this period of time, the countdown is stopped. Options:

- `0` - GUI is not invoked, the OS is loaded instantly
- `-1` - GUI is invoked, automatic OS loading is turned off

Clover must have be able to find a default entry for it to automatically boot into an OS see [DefaultVolume](#) below.

NOTE: If the Timeout integer is set to `0`, hold any key when powering on to invoke the bootloader GUI.

DefaultVolume

```
<key>DefaultVolume</key>
<string>MacHDD</string>
```

DefaultVolume is used to specify which entry is the default boot entry in Clover GUI. See also [DefaultLoader](#). It can be set to:

- Volume Name - The name of the volume. E.g. `Macintosh`.
- GUID - Globally Unique ID of the volume shown in Clover's boot, preboot or debug log. E.g. `57272A5A-7EFE-4404-9CDA-C33761D0DB3C`.
- Part of Device Path - Also shown in Clover's logs. E.g. `HD(1,GPT,57272A5A-7EFE-4404-9CDA-C33761D0DB3C,0x800,0xFF000)`.
- `LastBootedVolume` - The last booted volume will be set as default one in Clover GUI.

OS X Startup Disk can be used to reboot into another volume, but for the following reboot `DefaultVolume` will be used again.

DefaultLoader

```
<key>DefaultLoader</key>
<string>BOOTX64.efi</string>
```

In addition to [DefaultVolume](#) above, the path of the loader can be specified as DefaultLoader. This provides more precise default entry selection for Volumes that have multiple Loaders. The value can be the complete path or a unique portion like like file name.

IgnoreNVRAMBoot

```
<key>IgnoreNVRAMBoot</key>
<true/>
```

Ignores the setting of Startup Disk panel and allows booting into the same operating system at all times.

Debug

Formerly known as "Log" before revision 3064

```
<key>Debug</key>
<false/>
```

Get latest updates about
Open Source Projects,
Conferences and News.

If you are unable to boot into the Clover you can use this setting to produces a debug output to `/EFI/CLOVER/misc/debug.log`. This log then can be assessed to find out what the problem is. **!WARNING** turning on this log will dramatically increase loading time as it has to write the log to the disk as

No, Thank you

10.01.21, 11:35

it goes. So please be patient and give plenty of time before resetting your computer.

Fast

```
<key>Fast</key>
<true/>
```

Similar to setting `Timeout` to `0` but: * nvram.plist will be searched only until first occurrence * does not search for the best video mode * does not load themes and graphics * no chance to enter the GUI

Having this parameter enabled might not really have any big effect.

Instead, you can try a custom compilation: * disable GUI in CloverEFI (3 seconds difference) * only search for Sata0 drivers (9 seconds difference)

XMPDetection

```
<key>XMPDetection</key>
<true/> OR <false/> OR <string>Yes/No</string>
```

Detect best eXtreme Memory Profile when detecting memory or disable XMP detection.

```
<key>XMPDetection</key>
<integer>1/2</integer>
```

Secure

```
<key>Secure</key>
<true/>
```

Secure Boot protocol, a controversial subject, will restrict the booting process to signed binaries only. By enabling this option in your BIOS, booting of many operating systems won't be possible anymore, except Windows 8, 2013 and some Linux distributions.

The aim is to use Clover while Secure Boot is enabled. Clover's binary will need to be signed and a certificate to be loaded.

NeverHibernate

```
<key>NeverHibernate</key>
<true/>
```

Disables the hibernation state detection.

CustomLogo

```
<key>CustomLogo</key>
<true/> OR <false/> OR <string>Apple/Alternate/Theme/None/Path</string> OR <data>PNG/BMP/ICNS base64 data</data>
```

Enable the drawing of the custom boot logo.

- `true` - Uses the default boot style, Apple.
- `false` - Disables custom boot logo.
- `Apple` - Use the default gray on gray apple logo.
- `Alternate` - Use the alternate white on black apple logo.
- `Theme` - Use the theme boot screen for entry type - NOT IMPLEMENTED.
- `None` - Use no logo only background color, gray if not specified by custom entry.
- `Path` - A file path to load a custom image from
- `<data>` - A base64 encoded PNG, BMP, or ICNS data.

Get latest updates about
Open Source Projects,
Conferences and News.

If no option is specified then the boot screen will be drawn only for >= 10.10 Yosemite, so it remains compatible with previous behavior.

The `CustomLogo` key can also be used under GUI/Custom/Entries in conjunction with `BootBgColor` for a different screen for every OS. However, the graphics output protocol is not in anyway modified so if the OS draws after it is started then it is after the boot screen is drawn and will overwrite the

Sign Up
No, Thank you

custom logo, at least for now.

HibernationFixup

Rev 4223

```
<key>HibernationFixup</key>
<true/>
```

To be used for Hibernation modes 25 & 3 with Lilu.kext and HibernationFixup.kext.

RtcHibernateAware

Rev 4450

A data leak issue was identified in the hibernation code, allowing hibernation encryption key to be passed to the system through RTC and preserved till the next hibernation without a subsequent erase. More details could be found in [this message](#). To workaround this issue a new option enabling RTC memory erase upon waking from hibernation was added:

```
Boot > RtcHibernateAware = YES (BOOLEAN, off by default)
```

This option relies on a poorly documented (or rather undocumented) RTC memory access, and unspecified RTC memory layout, which is implementation-specific. While it is extremely recommended to be turned on if you rely on hibernation, it may not work on your hardware (should be fine on Ivy Bridge and newer at least), and is thus optional and disabled by default.

Note, that AppleRTC or FixRTC patches effectively break hibernation by reducing the available RTC memory and avoiding encryption key preservation. You should DISABLE them if you have no issues with BIOS preferences afterwards or use HibernationFixup. However, if RtcHibernateAware does not work for you, enabling AppleRTC patch and using HibernationFixup may be a safer workaround.

Revision 4515

Revision 4450 introduced a new key (Boot → RtcHibernateAware) which improved the situation with hibernation compatibility and reduced the impact of some security issues in this process. Starting with 10.13.6 a lot of legacy code got (finally) ditched on Apple side, and some changes are necessary to get hibernation to work on hacks. Revision 4515 incorporates them in Clover.

Mandatory stuff:

- 1) Clover must be r4450 and newer
- 2) AptioMemoryFix must be R20 (b83c025) or newer
- 3) Boot → RtcHibernateAware must be set to YES

Some additional notes:

- Config changes (and driver updates) will benefit all the systems (starting from Yosemite if I remember correctly), not just 10.13.6
- OsxAptioFix1/2/3 will NOT work with hibernation on 10.13.6 and newer, and they may be removed from the default Clover installer in the future (time to upgrade).
- If you can boot with AppleRTC/FixRTC patches OFF without BIOS settings reset, disable them, they break stuff in macOS
- If you have to use [HibernationFixup](#), you may want to update it to 1.2.1, currently only available from source.

Last, but not least:

If Hibernation does not work for you... well, RIP. Believe me or not, but it is not something you should rely on. Especially if you care about security and privacy of your data.

SignatureFixup

Rev 4270

```
<key>SignatureFixup</key>
<false/>
```

Rev 4291

If not SignatureFixup, it will be zeroed.

Anyway, the Signature must be fixed.

If True then the value is from hibernate image

If False then the value is zero and it will be copied by kernel into the image.

**Get latest updates about
Open Source Projects,
Conferences and News.**

[Sign Up](#)

No, Thank you

10.01.21, 11:35



© 2021 Slashdot Media. All Rights Reserved.

[Terms](#) [Privacy](#) [Privacy Choices](#) [Advertise](#)

**Get latest updates about
Open Source Projects,
Conferences and News.**

[Sign Up](#)

[No, Thank you](#)