



# OpenCore

Reference Manual (0.7.~~7~~.8)

[2022.01.13]

**Failsafe:** false

**Description:** Set to `true` to hide auxiliary entries from the picker menu.

An entry is considered auxiliary when at least one of the following applies:

- Entry is macOS recovery.
- Entry is macOS Time Machine.
- Entry is explicitly marked as `Auxiliary`.
- Entry is system (e.g. `Reset NVRAM`).

To display all entries, the picker menu can be reloaded into “Extended Mode” by pressing the `Spacebar` key. Hiding auxiliary entries may increase boot performance on multi-disk systems.

#### 4. `LauncherOption`

**Type:** `plist string`

**Failsafe:** Disabled

**Description:** Register the launcher option in the firmware preferences for persistence.

Valid values:

- `Disabled` — do nothing.
- `Full` — create or update the top priority boot option in UEFI variable storage at bootloader startup.
  - For this option to work, `RequestBootVarRouting` is required to be enabled.
- `Short` — create a short boot option instead of a complete one.
  - This variant is useful for some older types of firmware, typically from Insyde, that are unable to manage full device paths.
- `System` — create no boot option but assume specified custom option is blessed.
  - This variant is useful when relying on `ForceBooterSignature` quirk and OpenCore launcher path management happens through `bless` utilities without involving OpenCore.

This option allows integration with third-party operating system installation and upgrades (which may overwrite the `\EFI\BOOT\BOOTx64.efi` file). The `BOOTx64.efi` file is no longer used for bootstrapping OpenCore if a custom option is created. The custom path used for bootstrapping can be specified by using the `LauncherPath` option.

*Note 1:* Some types of firmware may have NVRAM implementation flaws, no boot option support, or other incompatibilities. While unlikely, the use of this option may result in boot failures and should only be used exclusively on boards known to be compatible. Refer to [acidanthera/bugtracker#1222](#) for some known issues affecting Haswell and other boards.

*Note 2:* While NVRAM resets executed from OpenCore would not typically erase the boot option created in `Bootstrap`, executing NVRAM resets prior to loading OpenCore will erase the boot option. Therefore, for significant implementation updates, such as was the case with OpenCore 0.6.4, an NVRAM reset should be executed with `Bootstrap` disabled, after which it can be re-enabled.

*Note 3:* [Some versions of Intel Visual BIOS \(e.g. on Intel NUC\) have an unfortunate bug whereby if any boot option refers to a path on a USB drive, then that is the only boot option shown when any USB drive is inserted. In the case of OpenCore, this means that only the OpenCore boot entry will be shown when any USB is inserted \(but this highly non-standard BIOS behaviour affects other software as well\). The best way to avoid this is to leave `LauncherOption` at `Disabled` on any version of OpenCore which will be started from a USB drive on this BIOS. However, if the problem has already occurred the quickest reliable fix is:](#)

- [Enable the system UEFI Shell in Intel Visual BIOS](#)
- [With power off, insert an OpenCore USB](#)
- [Power up and select the system UEFI Shell](#)
- [Since the system shell does not include `bcfg`, use the system shell to start OpenCore's `OpenShell` \(e.g. by entering the command `FS2:\EFI\OC\Tools\OpenShell.efi`, but you will need to work out which drive is correct for OpenCore and modify the drive number `FS#`: accordingly\)](#)
- [Within `OpenShell`, use `bcfg boot dump` to display the NVRAM boot options and then use `bcfg boot rm #` \(where `#` is the number of the OpenCore boot entry\) to remove the OpenCore entry](#)

[It is alternatively possible to start `OpenShell` directly from the OpenCore boot menu, if you have a working configured OpenCore for the system. In that case, and if OpenCore has `RequestBootVarRouting` enabled, it](#)