OpenCore

Reference Manual (0.8.7.8)

[2022.12.25]

8 Misc

8.1 Introduction

This section contains miscellaneous configuration options affecting OpenCore operating system loading behaviour in addition to other options that do not readily fit into other sections.

OpenCore broadly follows the "bless" model, also known as the "Apple Boot Policy". The primary purpose of the "bless" model is to allow embedding boot options within the file system (and be accessible through a specialised driver) as well as supporting a broader range of predefined boot paths as compared to the removable media list set out in the UEFI specification.

Partitions can only booted by OpenCore when they meet the requirements of a predefined `Scan policy`. This policy sets out which specific file systems a partition must have, and which specific device types a partition must be located on, to be made available by OpenCore as a boot option. Refer to the `ScanPolicy` property for details.

The scan process starts with enumerating all available partitions, filtered based on the `Scan policy`. Each partition may generate multiple primary and alternate options. Primary options represent operating systems installed on the media, while alternate options represent recovery options for the operating systems on the media.

- Alternate options may exist without primary options and vice versa.
- Options may not necessarily represent operating systems on the same partition.
- Each primary and alternate option can be an auxiliary option or not.
  - Refer to the `HideAuxiliary` section for details.

The algorithm to determine boot options behaves as follows:

1. Obtain all available partition handles filtered based on the `Scan policy` (and driver availability).
2. Obtain all available boot options from the `BootOrder` UEFI variable.
3. For each boot option found:
   - Retrieve the device path of the boot option.
   - Perform fixups (e.g. NVMe subtype correction) and expansion (e.g. for Boot Camp) of the device path.
   - On failure, if it is an OpenCore custom entry device path, pre-construct the corresponding custom entry and succeed.
   - Obtain the device handle by locating the device path of the resulting device path (ignore it on failure).
   - Locate the device handle in the list of partition handles (ignore it if missing).
   - For disk device paths (not specifying a bootloader), execute "bless" (may return > 1 entry).
   - For file device paths, check for presence on the file system directly.
   - ~~On the OpenCore boot partition, exclude all OpenCore bootstrap files by file header checks~~Exclude entries if there is a `.contentVisibility` file near the bootloader or inside the boot directory with `Disabled` contents (ASCII).
   - Mark device handle as *used* in the list of partition handles if any.
   - Register the resulting entries as primary options and determine their types.
     The option will become auxiliary for some types (e.g. Apple HFS recovery) or if its `.contentVisibility` file contains `Auxiliary`.
4. For each partition handle:
   - If the partition handle is marked as *unused*, execute "bless" primary option list retrieval.
     In case a `BlessOverride` list is set, both standard and custom "bless" paths will be found.
   - On the OpenCore boot partition, exclude OpenCore bootstrap files using header checks.
   - Register the resulting entries as primary options and determine their types if found.
     The option will become auxiliary for some types (e.g. Apple HFS recovery).
   - If a partition already has any primary options of the "Apple Recovery" type, proceed to the next handle.
   - Lookup alternate entries by "bless" recovery option list retrieval and predefined paths.
   - Register the resulting entries as alternate auxiliary options and determine their types if found.
5. Custom entries and tools, except such pre-constructed previously, are added as primary options without any checks with respect to `Auxiliary`.
6. System entries, such as `Reset NVRAM`, are added as primary auxiliary options.

The display order of the boot options in the OpenCore picker and the boot process are determined separately from the scanning algorithm.