OpenCore
Referenzhandbuch (0.8.0)

[2022.04.16]

1 Einleitung

Dieses Dokument enthält Informationen über das Format der OpenCore-Benutzerkonfigurationsdatei, die verwendet wird, um das korrekte Funktionieren des macOS-Betriebssystems einzurichten. Es ist als offizielle Klarstellung des erwarteten OpenCore-Verhaltens zu verstehen. Alle Abweichungen, die in veröffentlichten OpenCore-Versionen gefunden werden, sind als Dokumentations- oder Implementierungsprobleme zu betrachten, die über den Acidanthera Bugtracker gemeldet werden sollten. Ein Errata Sheet ist im OpenCorePkg Repository verfügbar.

Dieses Dokument ist als Spezifikation strukturiert und soll keine Schritt-für-Schritt-Anleitung für die Konfiguration eines Board Support Packages (BSP) für den Endbenutzer darstellen. Die Zielgruppe des Dokuments sind Programmierer und Ingenieure mit einem grundlegenden Verständnis von macOS-Interna und UEFI-Funktionalität. Aus diesen Gründen ist dieses Dokument ausschließlich in englischer Sprache verfügbar, und alle anderen Quellen oder Übersetzungen dieses Dokuments sind inoffiziell und können Fehler enthalten.

Artikel, Hilfsprogramme, Bücher und Ähnliches von Drittanbietern können für ein breiteres Publikum nützlicher sein, da sie leitfadenähnliches Material bieten können. Sie unterliegen jedoch den Vorlieben ihrer Autoren, Fehlinterpretationen dieses Dokuments und unvermeidlicher Veralterung. Wenn Sie solche Quellen verwenden, wie z.B. den OpenCore Install Guide von Dortania und verwandtes Material, beziehen Sie sich bitte bei jeder Entscheidung auf dieses Dokument und bewerten Sie die möglichen Auswirkungen neu.

Bitte beachten Sie, dass unabhängig von den verwendeten Quellen die Benutzer verpflichtet sind, jede OpenCore-Konfigurationsoption und die dahinter stehenden Prinzipien vollständig zu verstehen, bevor sie Probleme in den Acidanthera Bugtracker eintragen.

Anmerkung: Die Erstellung dieses Dokuments wäre ohne die unschätzbaren Beiträge anderer Personen nicht möglich gewesen: Andrey1970, Goldfish64, dakanji, PMheart, und einige andere. Die vollständige Liste ist in der OpenCorePkg-Historie zu finden.

1.1 Allgemeine Begriffe

- plist - Untermenge des ASCII-Eigenschaftslistenformats, geschrieben in XML, auch bekannt als XML plist Format Version 1. Uniform Type Identifier (UTI): com.apple.property-list. Plists bestehen aus Plist-Objekten, die zu einer hierarchischen Struktur zusammengefasst sind. Da das plist-Format nicht genau definiert ist, können alle Definitionen dieses Dokuments erst angewendet werden, nachdem plist durch Ausführen von plutil -lint als gültig eingestuft wurde. Externe Referenzen: https://www.apple.com/DTDs/PropertyList-1.0.dtd, man plutil.
- plist type - plist Sammlungen (plist array, plist dictionary, plist key) und Primitive (plist string, plist data, plist date, plist boolean, plist integer, plist real).
- plist object - endgültige Realisierung des plist-Typs, der als Wert interpretiert werden kann.
- plist array - Array-ähnliche Sammlung, entspricht einem Array. Besteht aus null oder mehr plist-Objekten.
- plist dictionary - map-ähnliche (assoziatives Array) Sammlung, entspricht dict. Besteht aus null oder mehr plist Schlüsseln.
- plist key - enthält ein plist-Objekt mit dem Namen plist key, konform zu key. Besteht aus druckbaren 7-Bit-ASCII-Zeichen.
- plist string - druckbare 7-Bit-ASCII-Zeichenfolge, konform zu string.
- plist data - base64-kodierter Blob, entspricht data.
- plist date - ISO-8601-Datum, entspricht date, wird nicht unterstützt.
- plist boolean - logisches Zustandsobjekt, das entweder wahr (1) oder falsch (0) ist, konform zu true und false.
- plist integer - möglicherweise vorzeichenbehaftete Ganzzahl zur Basis 10, konform mit integer. Passt in eine 64-Bit-Ganzzahl ohne Vorzeichen in Zweierkomplement-Darstellung, es sei denn, ein kleinerer vorzeichenbehafteter oder vorzeichenloser ganzzahliger Typ wird in der spezifischen plist-Objektbeschreibung ausdrücklich erwähnt.
- plist real - Fließkommazahl, konform zu real, nicht unterstützt.

- plist multidata - Wert, der von der Implementierung in Daten umgewandelt wird. Erlaubt die Übergabe von plist string, in diesem Fall wird das Ergebnis durch eine null-terminierte Folge von Bytes (C-String) dargestellt, plist integer, in diesem Fall wird das Ergebnis durch eine 32-Bit Little-Endian-Folge von Bytes in Zweierkomplement-Darstellung dargestellt, plist boolean, in diesem Fall ist der Wert ein Byte: 01 für true und 00 für false, und plist data selbst. Alle anderen Typen oder größere Ganzzahlen führen zu undefiniertem Verhalten.

2 Konfiguration
2.1 Begriffe der Konfiguration
- OC config - OpenCore Konfigurationsdatei im plist-Format mit dem Namen config.plist. Sie bietet eine erweiterbare Möglichkeit, OpenCore zu konfigurieren und ist so strukturiert, dass sie in mehrere benannte Abschnitte aufgeteilt ist, die sich unter dem Root-Plist-Dictionary befinden. Diese Abschnitte können plist array oder plist dictionary Typen haben und werden in den entsprechenden Abschnitten dieses Dokuments beschrieben.
- gültiger Schlüssel - plist-Schlüsselobjekt der OC-Konfiguration, das in diesem Dokument oder seinen zukünftigen Überarbeitungen beschrieben wird. Neben den explizit beschriebenen gültigen Schlüsseln werden auch Schlüssel, die mit dem #-Symbol beginnen (z.B. #Hello), als gültige Schlüssel betrachtet, und obwohl sie sich wie Kommentare verhalten, d.h. ihre Werte effektiv verwerfen, müssen sie dennoch gültige Plist-Objekte sein. Alle anderen Plist-Schlüssel sind nicht gültig, und ihr Vorhandensein führt zu undefiniertem Verhalten.
- valid value - gültiges Plist-Objekt der in diesem Dokument beschriebenen OC-Config, das alle zusätzlichen Anforderungen in den spezifischen Plist-Objekt-Beschreibungen erfüllt, falls vorhanden.
- ungültiger Wert - gültiges Plist-Objekt der in diesem Dokument beschriebenen OC-Config, das von einem anderen Plist-Typ ist, nicht mit den zusätzlichen Anforderungen in den spezifischen Plist-Objekt-Beschreibungen übereinstimmt (z.B. Wertebereich) oder in der entsprechenden Collection fehlt. Ungültige Werte werden mit oder ohne Fehlermeldung als jeder mögliche Wert dieses Plist-Objekts auf unbestimmte Weise gelesen (d.h. die Werte sind möglicherweise nicht bei allen Neustarts gleich). Während das Lesen eines ungültigen Wertes dem Lesen bestimmter definierter gültiger Werte entspricht, kann die Anwendung inkompatibler Werte auf das Host-System zu undefiniertem Verhalten führen.
- optionaler Wert - gültiger Wert der in diesem Dokument beschriebenen OC-Konfiguration, der auf eine bestimmte, in der Beschreibung des spezifischen Plist-Objekts angegebene Weise gelesen wird (anstelle eines ungültigen Werts), wenn er in der OC-Konfiguration nicht vorhanden ist. Alle anderen Fälle von ungültigem Wert gelten weiterhin. Sofern nicht ausdrücklich als optionaler Wert gekennzeichnet, muss jeder andere Wert vorhanden sein und wird bei Fehlen als ungültiger Wert gelesen.
- fatal behaviour - Verhalten, das zum Abbruch des Bootvorgangs führt. Implementierungen müssen verhindern, dass der Boot-Prozess fortgesetzt wird, bis das Host-System neu gestartet wird. Es ist zulässig, aber nicht erforderlich, in solchen Fällen kalte Neustarts durchzuführen oder Warnmeldungen anzuzeigen.
- undefiniertes Verhalten - Verhalten, das in diesem Dokument nicht vorgeschrieben ist. Implementierungen können beliebige Maßnahmen ergreifen, einschließlich, aber nicht beschränkt auf Maßnahmen, die mit fatalem Verhalten, der Annahme eines Zustands oder Werts oder der Nichtberücksichtigung zugehöriger Zustände oder Werte verbunden sind. Dies gilt jedoch nur, wenn diese Maßnahmen die Systemintegrität nicht beeinträchtigen.
2.2 Verarbeitung der Konfiguration

Es wird garantiert, dass die OC-Konfigurationsdatei mindestens einmal verarbeitet wird, wenn sie gefunden wird. Vorbehaltlich des OpenCore-Bootstrapping-Mechanismus kann das Vorhandensein mehrerer OC-Config-Dateien zum Lesen einer beliebigen dieser Dateien führen. Es ist zulässig, dass keine OC-Config-Datei auf der Platte vorhanden ist. In solchen Fällen müssen, wenn die Implementierung den Bootvorgang nicht abbricht, alle Werte den Regeln für ungültige Werte und optionale Werte folgen.
Für die OC-Config-Datei gelten Beschränkungen hinsichtlich der Größe, der Verschachtelungsebenen und der Anzahl der Schlüssel:
- Die Größe der OC-Config-Datei darf 32 MB nicht überschreiten.
- Die OC-Config-Datei darf nicht mehr als 32 Verschachtelungsebenen haben.
- Die OC-Config-Datei darf bis zu 32.768 XML-Knoten in jedem plist-Objekt enthalten.
- Ein Plist-Dictionary-Element wird als ein Knotenpaar gezählt.
Das Lesen missgestalteter OC-Config-Dateien führt zu undefiniertem Verhalten. Beispiele für fehlerhafte OC-Konfigurationsdateien sind
die folgenden:
- OC-Konfigurationsdateien, die nicht der DTD PLIST 1.0 entsprechen.
- OC-Konfigurationsdateien mit nicht unterstützten oder nicht konformen Plist-Objekten, die in diesem Dokument gefunden wurden. - OC-Konfigurationsdateien, die gegen Beschränkungen hinsichtlich Größe, Verschachtelungsebenen und Anzahl der Schlüssel verstoßen.
Es wird empfohlen, aber nicht erforderlich, das Laden von fehlerhaften OC-Config-Dateien abzubrechen und so fortzufahren, als ob keine OC-Config-Datei vorhanden wäre. Aus Gründen der Vorwärtskompatibilität ist es empfehlenswert, aber nicht erforderlich, dass die Implementierung vor der Verwendung ungültiger Werte warnt. Übersetzt mit www.DeepL.com/Translator (kostenlose Version)

S. 4

Es wird empfohlen, bei der Interpretation ungültiger Werte die folgende Konvention einzuhalten, sofern sie anwendbar ist:

Type	**Value**
`Plist string`	Empty string (`<string></string>`)
`Plist data`	Empty data (`<data></data>`)
`Plist integer`	**`0 (<integer>0</integer>)`**
`Plist boolean`	False (`<false/>`)
`Plist tristate`	False (`<false/>`)

2.3 Aufbau der Konfiguration
Die OC-Config-Datei ist in Unterabschnitte unterteilt, die in separaten Abschnitten dieses Dokuments beschrieben werden, und ist so konzipiert, dass sie versucht, nichts standardmäßig zu

aktivieren und Kill-Switches über eine Enable-Eigenschaft für plist-dict-Einträge bereitzustellen, die optionale Plugins und Ähnliches darstellen.
Die Datei ist so strukturiert, dass verwandte Elemente in Unterabschnitten wie folgt gruppiert sind:
- Add bietet Unterstützung für das Hinzufügen von Daten. Vorhandene Daten werden nicht überschrieben, sondern müssen bei Bedarf separat mit Delete behandelt werden.
- Delete bietet Unterstützung für das Entfernen von Daten.
- Patch bietet Unterstützung für die Änderung von Daten.
- Quirks bietet Unterstützung für spezielle Workarounds.
Die Root-Konfigurationseinträge bestehen aus den folgenden Elementen:

- `ACPI`
- `Booter`
- `DeviceProperties` • `Kernel`
- `Misc`
- `NVRAM`
- `PlatformInfo`
- `UEFI`

Eine grundlegende Validierung einer OC-Konfigurationsdatei ist mit dem Dienstprogramm ocvalidate möglich. Bitte beachten Sie, dass die verwendete Version von ocvalidate mit der OpenCore-Version übereinstimmen muss und dass ungeachtet dessen möglicherweise nicht alle in einer OC-Config-Datei vorhandenen Konfigurationsprobleme erkannt werden.
Hinweis: Um die Systemintegrität zu wahren, haben Eigenschaften typischerweise vordefinierte Werte, auch wenn solche vordefinierten Werte nicht in der OC-Konfigurationsdatei angegeben sind. Allerdings müssen alle Eigenschaften explizit in der OC-Config-Datei angegeben werden, und auf dieses Verhalten sollte man sich nicht verlassen.

3 Setup

3.1 Directory Structure

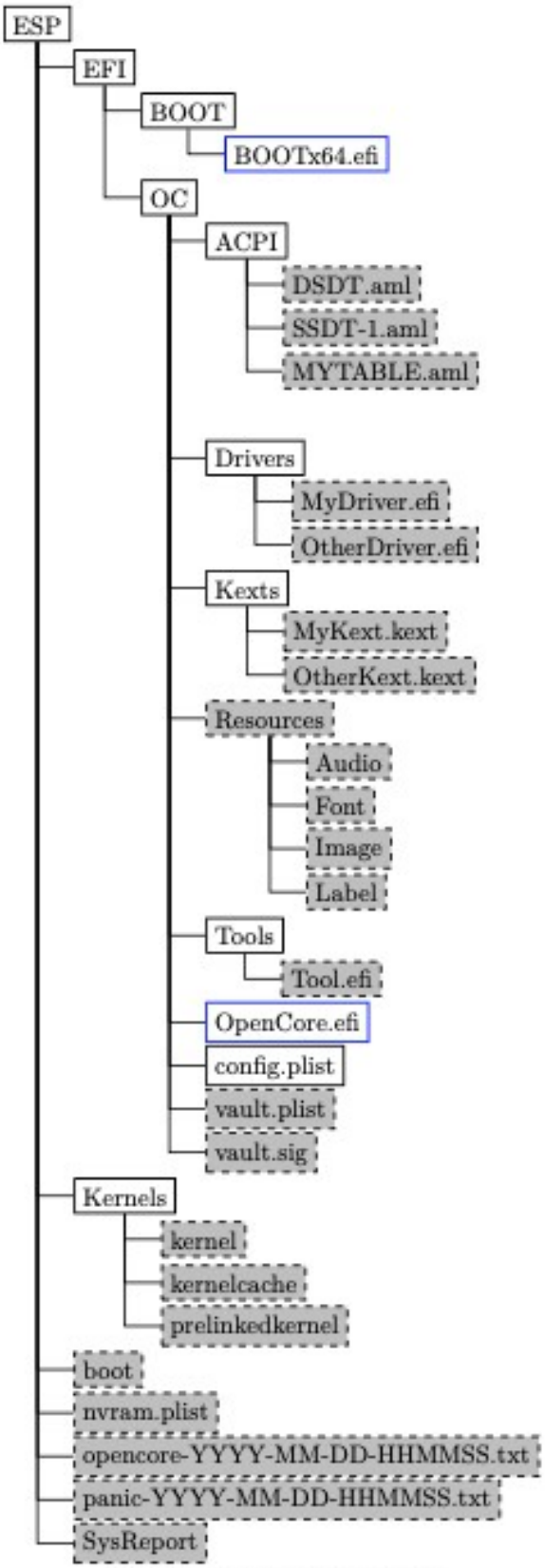

Figure 1. Directory Structure

Wenn das Verzeichnis-Boot verwendet wird, sollte die verwendete Verzeichnisstruktur den Beschreibungen im Abschnitt Verzeichnisstruktur entsprechen. S. 6