PCI-ROOT-PFADE UND UUIDS für Custom Entries und verstecken von Partitionen (Detailversion)

Inhaltsverzeichnis

- 1 Warum das Ganze?
- 2 Erster Schritt - die PCI-Root-Pfade
- 3 Zweiter Schritt - die UUIDs herausfinden
- 4 Dritter Schritt - Matchen der UUIDs mit den PCI-Root-Pfaden
- 5 Vierter Schritt - Anpassen der Custom Entries in OC (NDK)

Hi zusammen,

ich dachte mir, dass könnte dem einen oder anderen helfen - gerade wenn man CustomEntries haben möchte.

Zudem nutze ich es auch, um den Auto-Mount von z.B. Windows-Platten oder des CarbonCopy-Klons unter macOS zu unterbinden.

1 Warum das Ganze?

Hier gibt es aus meiner Sicht zwei häufige Wünsche. Zum einen möchte man seinen Bootpicker/Bootmenü einfach so haben, wie man es selber will.

Das betrifft nicht nur die Reihenfolge der Einträge sondern auch die Benennung.

Gerade wenn man z.B. CarbonCopy nutzt, hat man sonst in OC zwei mal den Eintrag "Recovery Partition" - nämlich einmal von eurer macOS Installation und natürlich auch vom CarbonCopy-Klon.

Zum anderen braucht man die UUIDs, damit man den Auto-Mount (also das automatische mounten von Partition) verhindern kann.

Praktische Beispiele sind hier Windows-Systempartition oder auch der eben erwähnte CarbonCopy-Klon, die ich z.B. nicht jedes mal auf meinem Mac sehen will.

Vermutlich gibt es noch weitere Themen, wo diese beiden Werte eine entscheidende Rolle spielen - sollte jmd. also noch was einfallen - feel free

Hinweis: Durch das Löschen/Formatieren einer Festplatte ändern sich die UUIDs!! Sollte ihr z.B. euren CC-Klon formatieren und neu erstellen, wird dieser eine andere UUID bekommen.

Das ist wichtig, wenn ihr via `fstab` eigentlich die Partition vom mounten abhalten wollt.

2 Erster Schritt - die PCI-Root-Pfade

Hier hat karacho netterweise schon eine Anleitung erstellt, die ich an dieser Stelle mal mit Prosa angereichert habe.

Wichtig: In der Basiskonfiguration von OC bzw dem Fork ist keine EFI-Shell vorhanden und in der config.plist zusätzlich deaktiviert.

In diesem Fall einfach die hier angehangene Shell unter EFI\OC\TOOLS ablegen und in der config.plist unter MISC/TOOLS/0 einfach den Eintrag aktivieren.

Optional: Wer, wie ich, viele Partitionen hat, sollte vor dem Start noch die UUID seiner EFI Partition ausfindig machen, damit man in der Shell nachher nicht erneut suchen muss.

1. Terminal öffnen
2. `Diskutil list`eingeben - das Ergebnis sieht dann ungefähr so aus
    1. Bildschirmfoto-2020-02-09-um-12-27-45.png

Image not found or type unknown

3. Eintrag der EFI suchen (normalerweise fast immer die erste oder zweite Partition - Typ = EFI, Name = EFI (außer ihr habt diesen geändert!) und sich den Identifier merken/notieren z.B. disk0s1
4. Danach im Terminal den Befehl `diskutil info /dev/disk0s1 | grep UUID` eingeben - wobei "disk0s1" natürlich durch euren Identifier ersetzt werden muss. Hinweis: Der Identifier ändert sich mit jedem Reboot!!
    1. Bildschirmfoto-2020-02-09-um-12-26-53.png

Image not found or type unknown

5. In der dann erscheinenden Ausgabe die Partition UUID notieren - hier reichen auch normalerweise die ersten 4-6 stellen, da es extrem unwahrscheinlich ist, dass ihr eine weitere Partition mit einer auf den ersten 4-6 Stellen identischen UUID habt.

Jetzt macht ihr einen Reboot und startet in die Shell.

Am Anfang läuft doch viel Informationen den Rhein runter ... sobald das fertig ist, einfach mit den "Bild hoch/runter" Tasten nach oben scrollen und nach den oben erwähnten 4-6 suchen.

Sobald ihr die gefunden habt, lasst ihr euren Blick nach links schleifen ... bis zu Beginn der Zeile wo in geld z.B. FS1 steht.

Als nächstes - und jetzt wird auch klar, warum es bei vielen Partitionen einfacher ist, die Information vorher zu erarbeiten - müsst ihr auf das entsprechende FS navigieren

- In unserem Beispiel also jetzt `fs1:` eingeben.
- Danach mit `ls EFI` prüfen, ob ihr auch wirklich auf der EFI Partition seid und sich der Ordner dort befindet.
- Wenn dem so ist, mit `cd EFI\OC` in das Verzeichnis gehen. (Das \ Zeichen bekommt ihr mit #)
- Und jetzt mit `map > out.txt` alle Mappings in eine Datei schreiben lassen. (Das > Zeichen bekommt ihr mit `SHIFT + .`)
- Mit dem Befehl `exit` verlasst ihr die Shell und bootet jetzt in eure macOS.

Zurück in eurem macOS mountet ihr jetzt die EFI-Partition und findet dort im Ordner EFI\OC die Datei `out.txt`

.

Wenn ihr diese öffnet, seht ihr alle PCI-Root-Pfade eures Systems.

3 Zweiter Schritt - die UUIDs herausfinden

Mit den Root-Pfaden alleine ist es (außer man hat nur eine Platte) nicht möglich, die korrekten Pfade für sein Multi-Boot-System heraus zu finden.

Also suchen wir uns jetzt die UUIDs der einzelnen Partitionen. Jetzt solltet ihr euch entscheiden, welche Informationen ihr braucht

1) Braucht ihr nur die PCI-Root-Pfade für den Bootpicker von OC?

2) Braucht ihr nur die UUIDs zur Verhinderung des Auto-Mounts?

3) Braucht ihr beides bzw. alles?

Für Punkt 1 macht ihr folgendes:

- Terminal öffnen und `diskutil list` eingeben
- Jetzt seht ihr all eure Partitionen. Notiert euch die jeweiligen Identifier zu den Partitionen, die ihr benötigt.
- Achtung: Für Windows benötigt ihr den Identifier der EFI. Für Mac geht das nicht. Hier benötigt ihr den Identifier der Preboot Partition.

Für Punkt 2 und 3 macht ihr eigentlich das Gleiche, nur dass ihr natürlich die UUIDs von den z.B. nicht automatisch zu mountenden Partitionen notiert.

Je nach dem wie viele Partitionen ihr habt, empfehle ich euch eine einfache Tabelle zu erstellen, wo zu jeder Partition die UUID gelistet wird. So verliert man auch nicht den Überblick.

Für eine macOS Festplatte auf der nur macOS läuft, sieht das dann z.B. so aus.

Nicht erschrecken, ich habe - weil es quasi ein Arbeitsgang ist - mir von allen Partitionen die Volume UUIDs, Partition UUIDs und PCI-Root-Pfade notiert. **Das muss man natürlich nicht machen!!**

Bildschirmfoto-2020-02-09-um-12-25-08.png

Image not found or type unknown

4 Dritter Schritt - Matchen der UUIDs mit den PCI-Root-Pfaden

Jetzt wird's interessant.

Ihr kopiert euch jetzt einfach die UUID zu der ihr den PCI-Root-Pfad haben wollt und sucht diesen in der

erstelltenout.txt.

Für Windows-Partitionen ist das auch schnell erledigt, da die UUID 1:1 in der gleichen Reihenfolge im PCI-ROOT-Pfad steht.

Für die Mac-Partitionen ist es mir erst nach einer Zeit aufgefallen.

Die Mac-Partitionen werden nämlich (warum auch immer) in den ersten vier Abschnitten (ich glaube es nennt sich) bit flipped.

Beispiel: Eure UUIDs der Preboot Partionen ist = 87F0A9AE-3006-4E5B-B5CC-BE8E082C5477

Der PCI-Root-Pfad dazu sieht aber so aus = PciRoot(0x0)/Pci(0x1D,0x0)/Pci(0x0,0x0)/NVMe(0x1,CA-2D-B4-81-58-38-25-00)/HD(2,GPT,E110BDAF-B514-4D6D-AF72-85604729F566,0x64028,0x3A321FE0)/VenMedia(BE74FCF7-0B7C-49F3-9147-01F4042E6842,AEA9F0870630**5B4EB5CCBE8E082C5477**)

Wie ihr seht, ist lediglich das letzte fünftel der UUID (fett markiert) in der richtigen Reihenfolge. Die anderen vier Teile (rosa, blau, grün und rot) sind bit flipped.

Am Ende habt ihr somit zu jeder Partition die korrekte UUID und den PCI-Root-Pfad.

5 Vierter Schritt - Anpassen der Custom Entries in OC (NDK)

Jetzt könnt ihr eure config.plist mit z.B. dem PlistEditorPro öffnen, navigiert zum Knoten MISC/ENTRIES und erstellt dort den entsprechenden Eintrag.

Für Windows ist es einfach: Man nehme den PCI-Root-Pfad und hängt am Ende ein `/\EFI\Microsoft\Boot\bootmgfw.efi` an. Sieht dann so aus.

Bildschirmfoto-2020-02-09-um-12-50-38.png

Image not found or type unknown

Für macOS müsst ihr über die Preboot Partition gehen. Sprich ihr nehmt den PCI-Root-Pfad der Preboot Partion und hängt danach `/\UUID_MACOS-PARTITION\System\Library\CoreServices\boot.efi`

Noch mal am Beispiel meiner Konfig - um es farblich zu kennzeichnen, habe ich hier auf den inline-code extra verzichtet.

Die Preboot Partition hat den Pfad:

PciRoot(0x0)/Pci(0x1D,0x0)/Pci(0x0,0x0)/NVMe(0x1,CA-2D-B4-81-58-38-25-00)/HD(2,GPT,E110BDAF-B514-4D6D-AF72-85604729F566,0x64028,0x3A321FE0)/VenMedia(BE74FCF7-0B7C-49F3-9147-01F4042E6842,AEA9F08706305B4EB5CCBE8E082C5477)

Die UUID der macOS Partition lautet:

0E9CC786-3E76-43E0-82C4-37EB4573CF30

Der feste Wert lautet:

\System\Library\CoreServices\boot.efi

Sodass daraus folgender kompletter Pfad für den CustomEntry wird

PciRoot(0x0)/Pci(0x1D,0x0)/Pci(0x0,0x0)/NVMe(0x1,CA-2D-B4-81-58-38-25-00)/HD(2,GPT,E110BDAF-B514-4D6D-AF72-85604729F566,0x64028,0x3A321FE0)/VenMedia(BE74FCF7-0B7C-49F3-9147-01F4042E6842,AEA9F08706305B4EB5CCBE8E082C5477)/\0E9CC786-3E76-43E0-82C4-37EB4573CF30\System\Library\CoreServices\boot.efi[/tt]

Fertig

Hoffe der Beitrag hilft dem ein oder anderen.

Vielen Dank noch mal an karacho und JimSalabim für die Infos!

Sollte jemand noch Fehler finden oder zusätzliche Hinweise geben können, bitte einfach in die Kommentare. Ich ergänze dann den Hauptbeitrag.

Schönen Sonntag