

# Custom Entries in OpenCore

Um in OpenCore die Windows-Partition auch mit anderer ScanPolicy als „0“ anzeigen zu lassen und um andere Volumes anzeigen zu lassen, die sonst nicht erscheinen, können wir Custom-Entries erstellen. Hierfür benötigen wir die **PciRoot**-Pfade der Volumes.

## Beispiel 1: Windows

Im OpenCore-Menü drücken wir die Leertaste, navigieren zum Eintrag für die **Shell** und starten diese durch Betätigung der Return- oder Enter-Taste.

Wir drücken kurz eine beliebige Taste oder warten ein paar Sekunden, bis wir zur Eingabe gelangen.

Wir müssen jetzt eine andere Tastaturbelegung beachten:

**Doppelpunkt:** **Umschalt** (also Shift) + **Ö**

**Größer-Zeichen (>):** **Umschalt** + **Punkt**

**Backslash (\):** **#** (Raute-Zeichen)

**y:** **z**

Wir suchen zunächst unsere OpenCore-EFI-Partition.

Wir geben ein:

```
ls fs1:EFI
```

und lassen uns so den Inhalt des EFI-Ordners auf „fs1“ anzeigen (sofern dort einer vorhanden ist).

Wir machen weiter mit

```
ls fs2:EFI
```

```
ls fs3:EFI
```

usw., bis uns der Inhalt unseres OpenCore-EFI-Ordners angezeigt wird, also die Ordner „BOOT“ und „OC“.

Fehlermeldungen bei den anderen Volumes, dass der Pfad oder Ordner nicht existiert oder ähnliches, ignorieren wir einfach. Auf diese Weise identifizieren wir die richtige Partition aber am schnellsten.

Nun wechseln wir auf genau dieses Volume (in meinem Fall ist es „fs8“), indem wir eingeben:

```
fs8:
```

(Ihr gebt natürlich stattdessen die Nummer ein, bei der bei euch der Inhalt der OpenCore-EFI-Partition angezeigt wurde).

Jetzt lassen wir uns zunächst die Map mit allen Pfaden als Textdatei auf unsere OpenCore-EFI-Partition schreiben:

```
map > map.txt
```

Anschließend suchen wir unsere **Windows-EFI-Partition**:

```
ls fs1:EFI
```

```
ls fs2:EFI
```

usw., so lange, bis uns die Ordner „Microsoft“ und „Boot“ angezeigt werden. Auch hier gilt: Fehlermeldungen bei einigen anderen Volumes, dass der Pfad oder Ordner nicht existiert oder ähnliches, ignorieren wir einfach.

Wir notieren uns das Volume, bei der die genannten beiden Ordner gelistet werden. Bei mir ist das zum Beispiel fs5. Wir schreiben uns auf, dass es sich dabei um die Windows-EFI-Partition handelt.

Dann geben wir ein:

```
exit
```

und verlassen damit die Shell.

Dann starten wir macOS und mounten die EFI-Partition. Dort befindet sich nun unsere Datei „**map.txt**“, die wir vorhin erzeugt haben.

Wir öffnen sie mit TextEdit und öffnen daneben auch die **config.plist** mit dem Plist Editor unserer Wahl.

Unter „Misc“ --> „Entries“ machen wir nun folgenden Eintrag:

▼ Misc	Dictionary	⌵ 6 items
▶ BlessOverride	Array	⌵ 0 items
▶ Boot	Dictionary	⌵ 10 items
▶ Debug	Dictionary	⌵ 4 items
▼ Entries	Array	⌵ 1 item
▼ Item 0	Dictionary	⌵ 6 items
Arguments	String	⌵
Auxiliary	Boolean	⌵ <input type="checkbox"/>
Comment	String	⌵ Not signed for security reasons
Enabled	Boolean	⌵ <input checked="" type="checkbox"/>
Name	String	⌵ Windows
Path	String	⌵ PciRoot(0x0)/Pci(0x17,0x0)/Sata(0x2,0xFFFF,0x0)/HD(2,GPT,300DD52-5D1F-45B7-8CF4-80D6557243B2,0x109000,0x32000)\EFI\Microsoft\Boot\bootmgfw.efi

Wir kopieren aus der map.txt den PciRoot-Pfad des File Systems, das wir als Windows-EFI identifiziert haben. Und fügen ihn bei „Path“ ein. Dahinter kommt direkt:

```
\/EFI\Microsoft\Boot\bootmgfw.efi
```

Der vollständige Windows-Pfad könnte also zum Beispiel lauten:

```
PciRoot(0x0)/Pci(0x17,0x0)/Sata(0x2,0xFFFF,0x0)/HD(2,GPT,9A5BA866-EBC5-47F5-9BDD-9A75E30139F2,0xFA000,0x32000)\EFI\Microsoft\Boot\bootmgfw.efi
```

So sähe der vollständige Eintrag als Text aus:

```
<key>Entries</key>
<array>
  <dict>
    <key>Arguments</key>
    <string></string>
    <key>Auxiliary</key>
    <false/>
    <key>Comment</key>
    <string>Not signed for security reasons</string>
    <key>Enabled</key>
    <true/>
    <key>Name</key>
    <string>Windows</string>
    <key>Path</key>
    <string>DEIN-PciROOT-PFAD\/EFI\Microsoft\Boot\bootmgfw.efi</string>
  </dict>
</array>
```

## Beispiel 2: macOS-Preboot-Volume

Zunächst brauchen wir die ID des **Preboot**-Volumes. Wir starten **macOS** und geben dort ins Terminal ein:

```
diskutil mount Preboot
```

Falls wir mehrere macOS-Systeme parallel betreiben (beispielsweise ein Volume für

Mojave und eines für Catalina haben etc.), machen wir das gewünschte Preboot-Volume vorher mit `diskutil list` ausfindig und mounten es dann stattdessen über den Identifier (beispielsweise `diskutil mount disk2s3`).

Anschließend geben wir ein:

```
open /Volumes/Preboot
```

Dadurch öffnet sich ein Finder-Fenster mit dem Preboot-Volume. Wir klicken zweimal auf den Ordner mit der langen Nummer, um diese kopieren zu können und speichern sie anschließend beispielsweise in den Notizen.

Dann machen wir einen Neustart, um wieder ins **OpenCore-Menü** zu gelangen und starten wieder die **Shell**.

Jetzt suchen wir das macOS-**Preboot**-Volume:

```
ls fs1:
```

```
ls fs2:
```

usw., bis uns die lange Nummer des Ordners auf dem Preboot-Volume angezeigt wird, die wir uns notiert haben.

Jetzt stellen wir sicher, dass es sich auch wirklich um die richtige Partition handelt. Wir geben `ls` gefolgt vom entsprechenden Volume (in meinem Fall fs10) ein, dahinter wieder einen Doppelpunkt und drücken anschließend die Tab-Taste und erst danach Return oder Enter. Dadurch wird der Pfad vervollständigt.

```
ls fs10: und Tab-Taste
```

wird so beispielsweise zu

```
ls fs10:AE846573-4AE4-4768-A9AB-B1203F075BC5\
```

So wird uns nun der Inhalt dieses Ordners angezeigt. Wenn er die Unterordner „usr“, „var“, „Library“ und „System“ enthält, haben wir das richtige Volume identifiziert. Enthält er dagegen etwas anderes, ist es nicht das richtige Volume und wir suchen mit `ls fs1:` etc. weiter, bis wir ein weiteres Volume gefunden haben, das ebenfalls einen Ordner mit dieser langen Nummer enthält, der dann aber wiederum die genannten Unterordner enthalten muss.

Wir notieren uns kurz das Volume (in meine Fall wie gesagt fs10) und schreiben uns dazu auf, dass es das Preboot-Volume ist.

Dann verlassen wir die Shell wieder und machen unseren Custom Entry für das Preboot-Volume genauso wie oben für Windows beschrieben.

Für das macOS-Preboot-Volume setzt sich die Pfadangabe beispielsweise aus dem **PciRoot-Pfad des Volumes** gefolgt von `/\` und der langen **Preboot-Ordner-ID**, die wir uns vorhin notiert haben, sowie der **weiteren Pfadangabe für die Datei „boot.efi“** zusammen:

```
PciRoot(0x0)/Pci(0x1B,0x0)/Pci(0x0,0x0)/NVMe(0x1,32-42-04-14-28-
A7-79-64)/HD(2,GPT,15E09D68-
D0A7-4A56-8B29-3CF5A315B194,0x64028,0x6FC1DA60)/
VenMedia(BE74FCF7-0B7C-49F3-9147-01F4042E6842,924A7EAE550FF840A2A2
B30EE22241EF)/\AE846573-4AE4-4768- A9AB-
B1203F075BC5\System\Library\CoreServices\boot.efi
```

Fertig!