

# 1-Click-Dualboot Lösung für Mac und Win unter Opencore

Beitrag von „Lokus“ vom 11. Januar 2021, 21:53

Alles klar. Dann hier mal die ausführliche Anleitung. Kopiere manches aus meiner [alten Clover-Anleitung](#) und passe es an Opencore an.

## 1. Schritt rEFInd installieren

Auf der furchtbar unübersichtlichen Seite von refind hier die aktuellste Version laden: <https://www.rodsbooks.com/refind/getting.html>

(Ich habe binary zip file heruntergeladen). Dann der Installationsanleitung hier folgen: <https://www.rodsbooks.com/refind/sip.html>

In kurz: Zip-File herunterladen, Pfad merken, in die Recovery-Partition rebooten, [SIP](#) deaktivieren, neustarten und dann per Terminal refind installieren. Ist aber alles auf der Seite beschrieben. Nicht vergessen [SIP](#) wieder zu aktivieren.

## 2. Booten von MacOS in Windows per AppleSkript:

Erstellt nun ein Skript mit Appleskript das folgenden Inhalt hat (muss ans System angepasst werden!):

Code

1. do shell script "diskutil mount 63C9CEF7-CA88-4CED-B372-91DBEF873452" user name "Benutzernamen einfügen" password "passwort einfügen" with administrator privileges
- 2.
3. do shell script "cp Volumes/EFI/EFI/refind/refind\_bootwin.conf Volumes/EFI/EFI/refind/refind.conf"
- 4.
5. do shell script "diskutil umount 63C9CEF7-CA88-4CED-B372-91DBEF873452" user name "Benutzernamen einfügen" password "Passwort einfügen" with administrator privileges
- 6.
7. do shell script "/Users/hendrik/reboot.sh"

Damit das mounten der EFI-Partition klappt, muss nach mount und umount die Disk UUID angegeben werden. Die findet man im Clover Configurator unter Mount EFI -> Partition Scheme.

Benutzer und Passwort müssen natürlich auch noch angepasst werden.

Mit dem nächsten Befehl wird die ursprüngliche `refind.conf` mit `refind_bootwin.conf` überschrieben. Dafür muss man im `refind`-Ordner der EFI die `refind.conf` zweimal kopieren und umbenennen, sodass man drei verschiedene Dateien hat:

`refind.conf`

`refind_bootwin.conf`

`refind_bootmac.conf`

In der `refind_bootwin.conf` habe ich die folgenden Zeilen geändert (Achtung, die Werte entsprechen meinem System. Bei euch kanns anders sein!)

Code

1. `timeout -1`
2. `default_selection 1`

Anmerkung: `timeout -1` führt dazu, dass der Bootpicker von `rEFInd` übersprungen wird und die `default_selection` gebootet wird.

`default_selection 1` ist bei mir die Windows-Partition.

Wenn man, so wie ich das Skript nun als App exportieren würde und dann ins Dock zieht, würde es beim nächsten Neustart wieder geöffnet werden und einen direkten Reboot verursachen, wenn man den Reboot-Befehl im Skript hat und auch MacOS so konfiguriert hat, dass Programme beim Neustart wieder geöffnet werden.

Lösen kann man das Problem, indem der Restart-Befehl per Shell-Skript ausgelagert wird. Das macht die Zeile mit dem `reboot.sh`-Skript.

Legt ein neues Textdokument an und fügt den folgenden Code ein:

Shell-Script

1. `#!/bin/bash`

2. sleep 5
3. launchctl remove application.com.apple.ScriptEditor.id.In-Windows-booten2.5509087.5509090
4. sleep 5
5. osascript -e 'tell application "System Events" to restart

Abspeichern und die Dateiendung in .sh umbenennen.

Wichtig: Die Pfade zum Skript und die dritte Zeile müssen an euer System angepasst werden! Die dritte Zeile mit "launchctl remove" schließt das Appleskript, das ja sonst nach einem Neustart geöffnet wird und wieder einen Reboot initiiert. Um herauszufinden, wie bei euch das Skript heißt, kommentiert die Zeile mit dem restart (osascript usw.) aus indem ein # an den Anfang der Zeile gesetzt wird. Damit deaktiviert ihr den Befehl. Ändert die sleep-Zeit auf 50. Dadurch habt ihr genug Zeit um nach dem starten des Appleskriptes folgendes zu machen: Öffnet Terminal und gebt folgendes ein:

Code

1. launchctl list

In der nun erscheinenden Liste such ihr mit Cmd+F nach dem Namen der App des Appleskripts (ich habe bei mir einfach Windows eingegeben).

Dann kopiert ihr die Zeile (siehe mein Shell-Skript) in euer Skript.

### 3. Booten von Windows in Mac OS

Das war etwas komplizierter und es gab weitere Problemchen, die aber gelöst werden konnten. Es funktioniert nun wie folgt:

1. Zunächst muss man die [EFI-Partition mounten](#) und ihr einen Laufwerksbuchstaben zuweisen. Ich habe es mit einem Programm gemacht, wie hier beschrieben: [klick](#)
2. Folgendes Powershell Skript muss erstellt werden:

Code

1. copy Volumes/EFI/EFI/refind/refind\_bootmac.conf Volumes/EFI/EFI/refind/refind.conf
2. Restart-Computer

Ähnlich wie beim oberen Skript, wird die refind.conf mit einer refind\_bootmac.conf überschrieben, die nun wieder so aufgebaut ist, dass automatisch bei jedem Neustart MacOS gebootet wird. Die refind\_bootmac.conf hat folgende geänderte Einträge:

Code

1. timeout -1
2. default\_selection 2

Anmerkung: "2" ist bei mir die Preboot-Partition, die MacOS startet (die "normale" Mac-Partition klappt nicht!)

Das Powershell-Skript muss zwingend mit Administrator-Rechten ausgeführt werden. Damit das geht, muss eine Verknüpfung für das Skript in Windows erstellt werden, wo man mit einem Rechts-Klick auf die Verknüpfung und bei -> Eigenschaften -> Erweitert ... ein Häkchen bei dem entsprechenden Feld setzen kann, sodass die Verknüpfung bzw. dann das Skript immer mit Admin-Rechten ausgeführt wird.

So sollte nun alles reibungslos funktionieren! Falls doch nicht, bitte melden! Ich hatte nämlich noch einen Zwischenschritt (Rechte ändern unter Windows für das Ausführen von Powershell-Skripten), ich glaube, dass der aber nicht notwendig war. Leider habe ich auch vergessen, was genau ich gemacht habe.

#### **4. Die Icons aufhübschen:**

Da ich mir in MacOS das App des Appleskriptes ins Dock gezogen habe, wollte ich auch ein schönes Icon haben. Meins habe ich mal hier angehängt.

Wer ein eigenes erstellen will, wird mit der kostenlosen App image2icon aus dem AppStore glücklich.

Das Logo der Windows-Verknüpfung habe ich auch geändert, habe ich auch angehängt.

## 5. Extra: EFI-Mount Skript

Da man ja als passionierter Hackintosher ab und zu seine [EFI mounten](#) muss, kann man das Appleskript, das wir erstellt haben, nutzen. Einfach nur die erste Zeile stehen lassen, den Rest löschen. Als App exportieren, ins Dock ziehen und: Tadaaa! Man hat eine 1-Klick-EFI-mount-Lösung 😊 Das Leben kann so einfach sein!