

...USB Anschlüsse Patchen

Unter Verwendung von **Clover Configurator**, dem **Hackintool** und dem **USBInjectAll.kext...**

Vorweg: Ab macOS 10.14.1 funktionieren die 'USB port limit patches' nicht mehr, somit werden nicht alle USB Anschlüsse mit ihren technischen Möglichkeiten bereitgestellt.

Um deine USB-Ports korrekt eingelesen und deklariert zu bekommen benötigst Du die **USBInjectAll.kext**, das Programm **Hackintool** und temporär die zu Deiner Systemversion passenden '**Port Limet Patches**':

Die passenden '**Port Limet Patches**' findest Du unter:

<https://www.hackintosh-forum.de/forum/thread/45023-usb-port-limit-patches-zusammenfassung/>

Anmerkung: Ausser einem Keyboard und einer Maus sollten keine weiteren externen Geräte an vorhandenen USB-Ausgängen Angeschlossen sein. (Keyboard und Maus für gewöhnlich an USB2).

1. Vorbereitung) Mounte Deine EFI mit dem **Clover Configurator**, öffne den Ordner CLOVER/kexts/Other. Kopiere den **USBInjectAll.kext** dort hinein.

Achtung: 'Open Core' User benötigen die **USBInjectAll.kext** NICHT – einfach unter '**Kernel/Quirks**' - '**KhciPortLimit**' auf '**Enabled**' setzen. Auch die folgenden '**Port Limit Patches**' werden bei '**Open Core**' nicht benötigt (...nach Neustart weiter mit 2b).

1b. Port Limet Patches setzen) Öffne Deine 'config.plist' (.../EFI/Clover) mit **Clover Configurator**, gehe im Menue auf '**Kernel and Kext Patches**' und trage in der Tabelle '**KextToPatch**' die für Dein System passende **Patches** ein (aus der 'port-limit-patches-zusammenfassung' siehe oben).

-> **sichern** -> **ummount EFI** -> **Clover Configurator beenden** -> **Neustart**.

2. Hackintool) Öffne nach dem Neustart das **Hackintool** und schaue im Reiter '**Installed Kexte**', ob USBInjectAll erfolgreich installiert und geladen wurde Bsp. USBInjectAll: Yes (Release-0.7.1).

2b. Hackintool) Gehe zum Reiter -> '**USB**', markiere alle Elemente der USB Anschluss-Liste und drücke erst die '**entfernen**' Schaltfläche und danach die Schaltfläche '**Aktualisieren**'. Zwei Ports, für Maus und Keyboard, müssten **grün** sein.

2c. aktive USB 2; 3; TypeC Anschlüsse und Connector-Typ deklarieren)

• Schliesse ein **USB2.0** Gerät nacheinander an alle vorhandenen USB Anschlüsse deines Systems an (einstecken -> auswerfen usw...), dadurch werden die gefundenen aktiven Anschlüsse **grün** hervorgehoben. Setze bei '**Connector**' über die Drop-Down-Liste für jeden grünen Anschluss den passenden Connector-Typ. Reine USB2 Ports (Keyboard und Maus) auf USB2. USB2 (HSxx) Anschlüsse die über eine USB3-Buchse erkannt wurden, sollten als '**USB3**' deklariert werden (= der abwärts kopatibele Teil eines USB3 Ports).

• Schliesse ein **USB3.0** Gerät nacheinander an alle vorhandenen USB3 Anschlüsse deines Systems an, die gefundenen aktiven Anschlüsse werden **grün** hervorgehoben. Setze jeden grünen gewordenen Anschluss auf Connector-Typ USB3 (wenn er es nicht schon ist).

• Schliesse ein **USB TypeC** Gerät an alle vorhandenen TypeC Anschlüsse an (in beiden Ausrichtungen), dadurch werden die gefundenen aktiven Anschlüsse **grün** hervorgehoben.

TypeC: Wenn bei verschiedener Ausrichtung der gleiche HSxx/SSxx Anschluss benutzt wird, enthält der TypeC Anschluss einen internen Umschalter und sollte als '**TypeC+Sw**' deklariert werden. Wenn bei verschiedener Ausrichtung unterschiedliche HSxx/SSxx Anschlüsse benutzt werden, enthält der TypeC Anschluss keinen internen Umschalter und sollte als '**TypeC**' deklariert werden.

USB-Anschlüsse mit dauerhaft verbundenen Geräten (z.B. Bluetooth aus WLAN/BT-Kombi-Karte) sollten als '**Internal**' deklariert werden. Interne USB-HUBs, welche standardmäßig über die ACPI-Geräte PR11 und PR21 erreichbar sind, sollten als '**Internal**' deklariert werden.

-> Markiere alle **nicht grün** hervorgehobenen Anschlüsse, drücke die '**entfernen**' Schaltfläche!

...>

3. Kext und SSDT-EC.aml erstellen) Benutze die **'Exportieren'** Schaltfläche um das Ergebnis als Datei auf deinem Schreibtisch zu speichern. Wurde eine **SSDT-EC.aml**-Datei erstellt, öffne **Clover Configurator**, kopiere diese nach `.../EFI/CLOVER/ACPI/patched`, kopiere den **USBPorts.kext** nach `.../EFI/CLOVER/kexts/Other`. Eine **SSDT-UIAC.aml** wird nicht benötigt und kann gelöscht werden.

4. Nicht mehr benötigten Kext'e und Patches bereinigen) Entferne den **USBInjectAll.kext** aus `EFI/CLOVER/kexts/Other`. Öffne Deine 'config.plist', gehe im Menue auf **'Kernel and Kext Patches'** und entferne die unter Punkt **1b.** gesetzten Einträge/Patches.

→ **sichern** → **unmount EFI** → **beenden** → **Neustart.**

5. Ergebnis prüfen) Öffne **Hackintool** und gehe zum Reiter → **'USB'**, markiere alle Elemente der USB Anschluss-Liste und wähle nacheinander die **'Entfernen'** und **'Aktualisieren'** Schaltfläche. Damit kannst du prüfen, ob alle vorher deklarierten Anschlüsse korrekt eingelesen wurden. Wenn du einen Connector-Typ mit dem **Hackingtool** ändern möchtest, musst du im Nachgang den **USBPorts.kext** wieder exportieren und den aktuell eingesetzten Kext ersetzen.

→ Sollte dir ein Fehler unterlaufen sein, lösche alle Ergebnisse und beginne von vorne!

Q. Was ist der **USBPorts.kext**?

A. Das ist eine Codeless Kernel Extension zur Deklaration der verschiedenen USB Anschlüsse

Q. Benötige ich dazu eine **SSDT-UIAC.aml**?

A. Nein, wird hier nicht benötigt, da die "codeless" Kext ausreicht

