

Vanilla Desktop-Guides

Dies ist eine Übersetzung des Desktop-Guides (erstellt unter Verwendung von <https://www.deepl.com/translatorDeepL>)

(Diese Anleitung ist nicht mehr auf dem neusten Stand, daher nur zur Orientierung zu verwenden.

Sie ähnelt im Aufbau der Anleitung für den Bootloader OpenCore)

Die neuere Entwicklung wird u. a. hier beschrieben:

<https://www.hackintosh-forum.de/forum/thread/50695-guide-clover-updaten-ab-v5123-inkl-openruntime-und-quirks-uefi-only/>

Das Original findet man hier:

<https://hackintosh.gitbook.io/-r-hackintosh-vanilla-desktop-guide/>

Inhaltsverzeichnis

Erste Schritte	3
Sammeln von Kexts	4
Erstellen des USB-Installers	5
Clover-Einrichtung	6
Ivy Bridge	14
Haswell	31
Skylake	50
Kaby Lake	67
Coffee Lake	85

Erste Schritte

Sie wollen also eine Vanilla-Installation?

Was bedeutet das überhaupt?

Eine Vanilla-Installation bedeutet, dass das Betriebssystem selbst relativ unangetastet bleibt - und dass der Großteil der Hackintosh-bezogenen Kexts, Patches usw. auf der EFI-Partition enthalten ist. In jeder Hinsicht ist die Hauptpartition einer Vanilla-Installation identisch mit der eines offiziellen Apple-Computers.

Kurzes Glossar der Begriffe

Es gibt eine Reihe von Begriffen, die Ihnen in diesem Handbuch begegnen werden - ich werde hier einige von ihnen und ihre Definitionen kurz erläutern:

- **Clover** - dies ist der Bootloader, den wir verwenden werden. Echte Macs haben eine eigene Firmware, die es ihnen erlaubt, macOS zu booten. PC-Hardware braucht ein wenig Hilfe, um das zum Laufen zu bringen; Clover hilft uns, das zu erreichen. Es kümmert sich auch um Kext-Injektion, ACPI-Umbenennungen, Kext-Patches und eine ganze Menge anderer Funktionen.

- **Kexts** - das Wort "kext" ist eigentlich die Kombination aus Kernel Extension; und Sie können sich kexts einfach als Treiber für macOS vorstellen.

- **Config.plist** - dies ist die Datei, die Clover sagt, was zu tun ist. Es ist eine XML-formatierte Eigenschaftsliste (sieht HTML sehr ähnlich) und ist einer der wichtigsten Teile beim Einrichten Ihres Hackintosh.

- **OOB** - ein Akronym für "Out of the Box", was soviel bedeutet wie "funktionierende Unterstützung ohne Tweaking".

- Mehr wird hinzugefügt, während ich an dieser Anleitung arbeite (wahrscheinlich)

Vorraussetzungen

Diese Anleitung konzentriert sich NUR auf Desktops. Es gibt auch andere Anleitungen für Laptops (siehe RehabMan's Anleitung bei TMac) - aber die sind oft viel spezifischer als diese Anleitung sein wird.

Für den Anfang brauchen wir ein paar Dinge:

1. Ein 8+GB USB-Flash-Laufwerk
2. Die Install OS X/macOS.app (vorzugsweise direkt aus dem App Store heruntergeladen)
3. Das Install-Paket von Clover (mit freundlicher Genehmigung von Dids)
4. Clover Configurator (der Mutige kann mit jedem Texteditor bearbeiten - aber CC ist typischerweise schneller)
 - Stellen Sie sicher, dass Sie die Global Edition erhalten
5. VirtualSMC.kext - dies ersetzt FakeSMC.kext als unseren SMC-Emulator und entweder VirtualSMC oder FakeSMC sind für das Booten unseres Hackintosh unerlässlich. Ohne einen von beiden würden wir nie booten können.
6. Alle anderen Kexts für unser Mobo/etc
 - Wir werden dies im nächsten Abschnitt durchgehen!
7. Etwas Geduld, Ausdauer und Google-Fu

Sammeln von Kexts

Welche Kexts benötige ich?

[VirtualSMC.kext](#) ist eine Voraussetzung - es emuliert den SMC-Chip, der in echten Macs zu finden ist, und überzeugt das Betriebssystem davon, dass dies ein echter Mac ist.

Ohne ihn kein Hackintosh :(

Alle folgenden Kexts sind in diesem [this repo](#) mit freundlicher Genehmigung von Goldfish64 verfügbar. Jedes Kext wird automatisch gebaut, sobald ein neuer Commit gemacht wird. Wenn Sie es vorziehen, sie selbst zu erstellen, können Sie mein [Lilu And Friends](#)

Ethernet

- [IntelMausiEthernet.kext](#) - diese Datei funktioniert mit den meisten neueren Intel-LAN-Chipsätzen
- [AppleIntelE1000e.kext](#) - dies funktioniert mit älteren Intel-LAN-Chipsätzen - kann aber bei neueren Chipsätzen KPs verursachen
- [AtherosE2200Ethernet.kext](#) - dies funktioniert bei den meisten Atheros- oder Killer-Netzwerk-Chipsätzen
- [RealtekRTL8111.kext](#) - dies funktioniert mit den meisten Gigabit Realtek LAN-Chipsätzen
- [RealtekRTL8100.kext](#) - für 10/100 Realtek LAN-Chipsätze

USB

Sie sollten sich [USBInjectAll.kext](#) besorgen. Wenn Sie ein H370-, B360- und H310-Coffee-Lake-System oder ein X79/X99/X299-System verwenden, sollten Sie sicherstellen, dass Sie auch die Datei [XHCI-unsupported.kext](#) hinzufügen. Ab 10.11 hat Apple ein Limit von 15 Ports pro USB-Controller eingeführt. Das hört sich nicht nach einem sehr großen Problem an, bis Sie erkennen, dass jeder USB-3-Anschluss als 2 zählt - einer für USB 2, einer für USB 3. Auf Skylake und neueren Builds, wo USB 2 und 3 nur über XHCI gehandhabt werden und jeder USB-3-Anschluss als 2 zählt, kann dieses Limit schnell erreicht werden. Es gibt jedoch eine Möglichkeit, alle USB-2-Anschlüsse über EHCI zu routen - unter Verwendung von RehabMan's [FakePCIID.kext](#) + [FakePCIID_XHCIMux.kext](#) (dies funktioniert allerdings nur bei einigen Chipsätzen), was den XHCI-Controller etwas entlasten kann.

Audio

Für Audio sollten Sie das [AppleALC.kext](#) von /u/vit9696 und das dazugehörige [Lilu.kext](#) nehmen - vorausgesetzt, Sie haben einen unterstützten Codec. AppleALC ist in der Lage, AppleHDA.kext im laufenden Betrieb zu damit patchen, für natives Audio mit nicht unterstützten Codecs. Es hat auch eine Reihe von Codec-Verben eingebaut, die bei Audio-after-sleep helfen.

Grafiken

Für GPUs - Sie sollten sich [WhateverGreen.kext](#) und das dazugehörige [Lilu.kext](#) schnappen - dies hat die Funktionalität von IntelGraphicsFixup, NvidiaGraphicsFixup, CoreDisplayFixup und Shiki in sich vereint. Früher waren alle diese Kexts separat - aber da viele von ihnen Ressourcen gemeinsam nutzen, wurden sie zusammengefasst.

WiFi und Bluetooth

Apple ist ziemlich minimal mit seiner WiFi-Unterstützung, also werde ich nur die beiden Hauptchipsätze behandeln, mit denen ich vertraut bin. Ich habe einen BCM94360CD + PCIe-Adapter und einen BCM94352HMB/BCM94352Z in meinen Hackintoshes verwendet. Die BCM94360CD funktionierte OOB ohne Extras, da sie eine native Karte ist. Für die BCM94352-Varianten habe ich [AirportBrcmFixup.kext](#) und das dazugehörige [Lilu.kext](#) für die WiFi-Einrichtung und [BrcmBluetoothInjector.kext](#) (auf 10.13.6+) oder [BrcmPatchRAM2.kext](#) zusammen mit [BrcmFirmwareData.kext](#) verwendet - alle Brcm*-Kexts sind von Rehab
Man's OS-X-BrcmPatchRAM Repo.

Extras

Abhängig von der restlichen Hardware benötigen Sie möglicherweise noch weitere Kexts, aber diese Anleitung ist als allgemeine Grundlage gedacht, so dass Sie sich auf Ihre Google-Fähigkeiten verlassen müssen.

Erstellen des USB-Installers

Sobald Sie Ihren 8+GB USB-Installer haben, müssen wir sicherstellen, dass er richtig eingerichtet ist. Wenn Sie nicht vorhaben, den Installer zu patchen (was ich nicht tue), sollten Sie den USB-Installer wie folgt einrichten (Für neuere MacOS-Versionen wird mindestens 16 GB für den USB-Stick benötigt.):

- GUID Partition Map
- 1 Partition
- OS X Erweitert (Journaled)

Um dies zu tun, starten Sie das Terminal (unter /Programme/Dienstprogramme) und geben Sie `diskutil list` ein.

Dadurch erhalten Sie eine Liste aller angeschlossenen Festplatten und deren Partitionen. Notieren Sie sich die Laufwerkskennung für Ihr USB-Laufwerk. **RATEN SIE DIESE NICHT, DA WIR ES GLEICH LÖSCHEN WERDEN!** Führen Sie dann den folgenden Befehl aus und ersetzen Sie `disk#` durch Ihre tatsächliche Kennung:

```
diskutil partitionDisk /dev/disk# GPT JHFS+ "USB" 100%
```

Dies partitioniert den USB-Stick wie oben aufgeführt und benennt sie in "USB" um.

(Das Ganze lässt sich auch mit dem Festplattendienst-Programm bewerkstelligen.)

Sie können nun den entsprechenden Befehl aus [Apples eigener Anleitung](#) ausführen - für dieses Beispiel verwenden wir den Mojave-Befehl:

```
sudo "/Programme/Install macOS Mojave.app/Contents/Resources/createinstallmedia" --volume /Volumes/USB
```

Dies wird einige Zeit in Anspruch nehmen, und es wird nicht viel für den Status-Updates angezeigt. Es kann bis zu 30-40 Minuten dauern, haben Sie also etwas Geduld. Nehmen Sie sich eine Tasse Kaffee, lesen Sie die Nachrichten, treffen Sie sich mit Freunden und Familie - Sie werden eine Weile hier sein.

Wenn dies abgeschlossen ist, haben Sie einen USB-Installer, der auf einem echten Mac booten kann. Wir müssen nur noch die Hackintosh-bezogenen Dinge einrichten, und schon sind wir im Geschäft!

(Für diejenigen, die Angst vor der Kommandozeile haben - ich habe vor einiger Zeit ein Skript erstellt, das diese Aktionen für Sie durchführen kann.)

Clover-Einrichtung

Installieren von Clover

Starten Sie Ihr Clover-Installationspaket. Auf der 3. Seite des Installationsprogramms stellen Sie sicher, dass Sie Ihren USB als Ziel auswählen. Außerdem wollen wir die Installation anpassen - denn die Standardeinstellungen sind ziemlich dürftig.



3. Seite des Clover-Installationsprogramms - Beachten Sie die Schaltfläche "Anpassen" unten links

Die üblichen Optionen, die Sie im Menü "Customize" (Anpassen) überprüfen möchten, sind in den folgenden Screenshots mit anschließender Erläuterung dargestellt (Hinweis: Jedes Board, das nicht der Z370 300-Serie angehört, benötigt auch EmuVariableUefi-64 - es befindet sich in aktuellen Clover-Paketen unter UEFI-Treiber -> Zusätzliche Treiber):



Empfohlene Anpassungseinstellungen:

- Installieren Sie Clover nur für UEFI-Booting
- Installieren Sie Clover auf dem ESP
- Unter UEFI-Treibern:
 - AptioMemoryFix (der neue Hotness, der NVRAM-Fixes sowie eine bessere Speicherverwaltung enthält).
 - Hinweis: Sie können OsxAptioFix3Drv verwenden, das sich unter UEFI-Treiber -> Speicherfix-Treiber befindet, wenn Sie den offiziellen Clover-Installer von SourceForge verwenden. Optional können Sie die finale Version von AptioMemoryFix von [here](#) herunterladen
 - HFSPlus (oder VBoxHfs) - eines davon wird benötigt, damit Clover HFS+-Volumes erkennen und booten kann. Wenn Sie die Option sehen, sie im Installationsprogramm zu aktivieren, stellen Sie sicher, dass sie ausgewählt ist - wenn Sie sie nicht im Installationsprogramm sehen, überprüfen Sie, ob eine von ihnen in Ihrem Clover installierten EFI-Treiber-Ordner existiert
 - ApfsDriverLoader - (Erhältlich in Dids' Clover-Builds - oder [here](#)) dies ermöglicht es Clover, APFS-Volumes zu sehen und von ihnen zu booten, indem apfs.efi aus dem ApfsContainer geladen wird, der sich auf dem Blockgerät befindet (wenn Sie auch AptioMemoryFix verwenden, stellen Sie sicher, dass Sie Version R21 oder neuer haben)

Das war's schon.

Wenn Sie FileVault nicht benötigen und eine Standard-UEFI-Installation einrichten, sind dies die einzigen Einträge, die Sie benötigen sollten. Es kann sein, dass Sie im Abschnitt UEFI-Treiber und den Unterabschnitten mehr als die oben genannten ausgewählt haben - Sie können alle nicht aufgeführten Einträge abwählen. Es gibt einige, die sogar Konflikte

mit anderen Einstellungen/Kexten verursachen können (z.B. [SMCHelper-64.efi](#)), daher ist es eine gute Idee, hier so schlank wie möglich zu sein.

Wo werden Uefi-Treiber installiert?

Wenn Sie manuell Änderungen an den EFI-Treibern Ihrer Clover-Installation vornehmen müssen (oder einfach nur überprüfen, ob Sie das haben, was Sie brauchen), kann sich deren Speicherort je nach Ihrer Clover-Version unterscheiden.

- Clover Version bis zu r4982: EFI -> CLOVER -> drivers64UEFI
- Clover Versionen r4983 bis r4985: EFI -> CLOVER -> UEFIDrivers
 - Verwendet weiterhin EFI -> CLOVER -> drivers64UEFI, wenn das oben genannte nicht gefunden wird
- Clover-Versionen r4986 bis heute: EFI -> CLOVER -> Treiber -> EFI
 - Wird weiterhin EFI -> CLOVER -> drivers64UEFI verwenden, wenn das oben genannte nicht gefunden wird.

Kopieren von Kexts

Sobald Clover installiert ist, sehen Sie die EFI-Partition Ihres USB auf dem Desktop - wir wollen zu EFI -> CLOVER -> kexts -> Other navigieren und die kexts, die wir zuvor heruntergeladen haben, dorthin kopieren. Stellen Sie sicher, dass Sie die kexts entpacken, bevor Sie sie rüberkopieren. Alle dSYM-Dateien können ignoriert werden, wir brauchen nur die .kext-Dateien.

Aber was ist mit den 10.xx-Ordnern?

Wenn Clover nach Kexts sucht, werden die Kexts in den 10.xx-Ordnern nur injiziert, wenn Clover feststellt, dass der Ordnername mit der gebooteten OS-Version übereinstimmt. Es gibt jedoch nur sehr wenige Kexts, die von der Betriebssystemversion abhängig sind, und wenn Sie das Betriebssystem aktualisieren und dabei vergessen, die Kexts zu migrieren, können Sie in einen nicht bootfähigen Zustand geraten. Die Kexts, die sich im Ordner "Other" befinden, werden unabhängig von der erkannten Betriebssystemversion injiziert.

Config.plist-Grundlagen

Die config.plist befindet sich unter /Volumes/EFI/EFI/CLOVER/config.plist und ist eine der schwierigeren Dateien für diejenigen, die neu in der Hackintosh-Welt sind. Wir werden hier einige der Grundlagen der Struktur erläutern und dann in verschiedene Abschnitte für unterschiedliche Hardwarekonfigurationen unterteilen.

Was ist das?

Die config.plist ist eine XML-Eigenschaftsliste. XML ist eine Auszeichnungssprache, die viele Ähnlichkeiten mit HTML aufweist. Das bedeutet, dass Ihnen ein paar verschiedene Datentypen zur Verfügung stehen, und der größte Teil der Struktur dreht sich darum, den Überblick über die öffnenden und schließenden Tags zu behalten.

Die Struktur

Wenn Clover eine config.plist durchsucht, erwartet es, dass bestimmte Teile an bestimmten Stellen zu finden sind. Die Reihenfolge und der Umfang Ihrer config.plist sind sehr wichtig, da das Ablegen von Informationen an der falschen Stelle diese effektiv vor

Clover verstecken kann. Sie können das allgemeine Layout, das Clover erwartet, im Clover Wiki einsehen.

Datentypen

Es gibt ein paar wichtige Datentypen, auf die wir bei der Arbeit mit der Konfiguration stoßen werden. Ich werde hier die gängigsten skizzieren. Hinweis - Ich verwende hier nur die öffnenden Tags, wenn wir tatsächlich mit den folgenden Typen arbeiten, müssen wir sicherstellen, dass wir hinter uns aufräumen und unsere Tags schließen.

Zeichenketten

```
<string>Dies ist ein String</string>
```

Strings sind einfach nur Text. Nicht besonders verrückt - Sie werden sie oft für Kommentare und andere solche Dinge verwenden.

Ganzzahlen

```
<ganzzahl>1</ganzzahl>
```

Dies sind einfach ganze Zahlen. Auch hier nichts allzu Wildes.

Daten

```
<data>RXh0ZXJuYWw=  
</data>
```

Dies sieht zwar ähnlich aus wie die obigen Zeichenketten, ist aber tatsächlich die base64-Darstellung einiger Daten. Was bedeutet das? Sie können es hier nachlesen, aber zusammengefasst ist es eine clevere Möglichkeit, binäre Daten in einem Textformat zu speichern, ohne dass sie beim Kopieren, Verschieben usw. verloren gehen. Noch verrückter ist, dass Sie die obigen base64-Daten mit folgendem Befehl in Terminal.app in ASCII konvertieren können:

```
echo RXh0ZXJuYWw= | python -m base64 -d && echo
```

Dies gibt Extern in der nächsten Zeile aus. Wir verwenden das && echo, um einen Zeilenumbruch auszugeben, nachdem unser Text ausgespuckt wurde - das macht ihn leichter lesbar.

Sie können auch von ASCII nach base64 konvertieren (praktisch für die Arbeit mit ACPI-Umbenennungen - mehr dazu später) mit dem folgenden Befehl in Terminal.app:

```
echo -n Extern | base64
```

Dies wird RXh0ZXJuYWw= ausspucken, was genau das ist, was wir erwarten würden.

Hinweis - Viele Plist-Editoren (Clover Configurator, Xcode, etc.) zeigen Daten als Hexadezimal statt als Base64 an, achten Sie also darauf, welchen Sie verwenden.

Boolesche Werte

```
<true/> oder <false/>
```

Dies sind boolesche Werte. Sie können sich diese als Ein/Aus-Werte vorstellen. Im Gegensatz zu den anderen hier aufgeführten Typen sind diese gleichzeitig ein öffnendes und ein schließendes Tag, sodass sie kein passendes Tag benötigen.

Arrays

```
<array>  
  <string>Bob</string>  
  <string>Jim</string>
```

```
<string>Chris</string>
</array>
```

Dies ist eine unsortierte Liste von Elementen. Wenn wir eine Sammlung von Namen zusammenstellen wollten, könnten wir sie als `<string>`-Werte in unserem `<array>` wie im obigen Beispiel speichern. Der Zugriff auf sie erfolgt über den Index (das ist einfach die Nummer, an der sie in der Liste stehen).

Dictionaries (Wörterbücher)

```
<dict>
  <key>Name</key>
  <string>Bob</string>
  <Schlüssel>Alter</Schlüssel>
  <ganzzahl>20</ganzzahl>
  <key>Kennt XML</key>
  <true>
</dict>
```

Dies bezeichnet ein Wörterbuch. Diese eignen sich wie Arrays zum Speichern zusätzlicher Datensammlungen, aber anstatt indexbasiert zu sein, verwenden sie eine Schlüssel/Wert-Organisation. Wie Sie aus dem obigen Beispiel ersehen können, sind wir in der Lage, spezifische Daten über Bob durch die Verwendung dieser Schlüssel/Wertpaare zu speichern. Alle Schlüssel sind nur Text (wie unsere Zeichenketten).

Beispiele

Lassen Sie uns einige Vorher/Nachher-Beispiele mit einigen vorgetäuschten `config.plist`-Daten durchgehen, um hoffentlich etwas von dem Mysterium zu entfernen, das sich in diesem Assistenten-Schrank abspielt.

Change True/False (Wahr/Falsch ändern)

In diesem ersten Beispiel werden wir einfach einen booleschen Wert von `true` auf `false` ändern, oder umgekehrt. Ich werde den `Disabled`-Wert innerhalb eines `KextsToPatch`-Eintrags verwenden, um dies zu einem realen Beispiel zu machen. Zuerst gebe ich uns den `KextsToPatch`-Eintrag, mit dem wir arbeiten werden:

```
<dict>
  <key>Kommentar</key>
  <string>Externer Icon-Patch</string>
  <key>Deaktiviert</key>
  <false/>
  <key>Finden</key>
  <data>
  RXh0ZXJuYWw=
  </data>
  <key>InfoPlistPatch</key>
  <false/>
  <key>Name</key>
  <string>AppleAHCIPort</string>
  <key>Ersetzen</key>
  <data>
```

```
SW50ZXJuYWw=
```

```
</data>
```

```
</dict>
```

Puh, das mag auf den ersten Blick nach viel aussehen, aber wir werden die Dinge aufschlüsseln. Als erstes gehe ich darauf ein, was die verschiedenen Schlüssel bedeuten:

- Comment - dies ist nur ein Kommentar, der beschreibt, was der Patch macht.

- Disabled - dies ist ein wenig kontraintuitiv, aber es ist ein boolescher Wert, der bestimmt, ob dieser Patch deaktiviert ist oder nicht. Wenn er auf `<true/>` gesetzt ist, ist der Patch deaktiviert und Clover ignoriert ihn. Wenn er auf `<false/>` gesetzt ist, ist der Patch nicht deaktiviert, und er wird angewendet.

- InfoPlistPatch - dies ist ein boolescher Wert, der Clover mitteilt, ob wir die Info.plist des Kextes anstelle der Binärdatei patchen.

- Name - das ist der eigentliche Kext, den wir patchen wollen.

- Find - das sind die base64-Daten, nach denen wir in der zu patchenden Binärdatei suchen wollen.

- Replace - das sind die Daten, mit denen wir die gefundenen Daten ersetzen wollen (wenn wir sie finden).

In Ordnung, jetzt erkläre ich, wofür dieser Patch eigentlich ist. Anhand der Informationen in diesem Patch sucht Clover nach dem AppleAHCIPort-Kext und sucht nach `RXh0ZXJuYWw=` (was beim Dekodieren der Daten zu External wird) und ersetzt es durch `SW50ZXJuYWw=` (was beim Dekodieren zu Internal wird). Das Endergebnis ist, dass Laufwerke, die Hot-Plug-fähig sind (und normalerweise als externe Laufwerke betrachtet werden), als interne Laufwerke angezeigt werden und nicht das orangefarbene Symbol auf dem Desktop haben. Dieses Patchen geschieht "on the fly" und ist nicht-destruktiv - das bedeutet, dass das AppleAHCIPort-Kext auf dem System unberührt bleibt.

So - ich denke, an dieser Stelle sollte ich erklären, wie wir einen booleschen Wert ändern würden, um diesen Patch zu deaktivieren. Ich habe bereits erwähnt, wie der Disabled-Schlüssel funktioniert - wo wir das `<false/>` in der nächsten Zeile in `<true/>` ändern, was diesen Patch wie folgt auf disabled setzt:

```
<dict>
```

```
  <key>Comment</key>
```

```
  <string>External icons patch</string>
```

```
  <key>Disabled</key>
```

```
  <true/>
```

```
  <key>Find</key>
```

```
  <data>
```

```
  RXh0ZXJuYWw=
```

```
  </data>
```

```
  <key>InfoPlistPatch</key>
```

```
  <false/>
```

```
  <key>Name</key>
```

```
  <string>AppleAHCIPort</string>
```

```
  <key>Replace</key>
```

```
  <data>
```

```
  SW50ZXJuYWw=
```

```
  </data>
```

```
</dict>
```

Nicht zu beängstigend, oder?

Hinzufügen eines neuen Dikt zu einem Array

Dies ist ein Fall, den ich ziemlich oft sehe und der für neue Leute ein wenig überwältigend sein kann. Wenn Ihnen gesagt wird, dass Sie einen neuen Patch unter config.plist -> ACPI -> DSDT -> Patches hinzufügen sollen, öffnen wir zuerst unsere config.plist und sehen, womit wir arbeiten.

Wir nehmen für dieses Beispiel an, dass die Konfiguration so aussieht:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>ACPI</key>
  <dict>
    <key>DSDT</key>
    <dict>
      <key>Fixes</key>
      <dict>
        <key>FixHPET</key>
        <true/>
        <key>FixIPIC</key>
        <true/>
        <key>FixRTC</key>
        <true/>
        <key>FixTMR</key>
        <true/>
      </dict>
      <key>Patches</key>
      <array>
        <dict>
          <key>Comment</key>
          <string>change OSID to XSID (to avoid match against _OSI
XOSI patch)</string>
          <key>Disabled</key>
          <false/>
          <key>Find</key>
          <data>
            T1NJRA==
          </data>
          <key>Replace</key>
          <data>
            WFNJRA==
          </data>
        </dict>
        <dict>
          <key>Comment</key>
          <string>change _OSI to XOSI</string>
          <key>Disabled</key>
          <false/>
          <key>Find</key>
          <data>
            X09TSQ==
          </data>
          <key>Replace</key>
          <data>
            WE9TSQ==
          </data>
        </dict>
      </array>
    </dict>
  </dict>
</plist>
```

Wenn wir in der Konfiguration von oben nach unten schauen, können wir dem Pfad folgen, den ich zuvor skizziert habe. Wir sehen ACPI, und darunter DSDT. Unterhalb von DSDT befindet sich Fixes und in der Zeile darunter Patches. Wir sind im Moment nicht mit dem Abschnitt Fixes beschäftigt, also ignorieren wir ihn und konzentrieren uns auf die Patches.

Zunächst möchte ich darauf hinweisen, dass sich unter dem <Schlüssel>Patches</key> ein öffnendes Array-Tag (<array>) befindet - und dann haben wir 2 Wörterbücher - jedes mit ähnlichen Schlüsseln, mit denen wir im vorherigen Beispiel gearbeitet haben (Comment, Disabled, Find, Replace). Nach den Dictionaries sehen wir das schließende Array-Tag (</array>). Unser Ziel ist es, ein neues Wörterbuch zwischen den Tags <array> und </array> hinzuzufügen und gleichzeitig zu vermeiden, dass die anderen vorhandenen Wörterbücher zerschnitten werden. Die Daten, die wir hinzufügen werden, sehen wie folgt aus:

```
<dict>
  <key>Comment</key>
  <string>change SAT0 to SATA</string>
  <key>Disabled</key>
  <false/>
  <key>Find</key>
  <data>
    U0FUMA==
  </data>
  <key>Replace</key>
  <data>
    U0FUQQ==
  </data>
</dict>
```

Wie ich bereits erwähnt habe, sind Arrays ungeordnet - das bedeutet, dass es keine Rolle spielt, ob wir unser neues Wörterbuch vor den bestehenden 2, nach ihnen oder dazwischen einfügen. Ich werde es jedoch am Ende hinzufügen - direkt über dem letzten </array>-Tag, etwa so:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>ACPI</key>
  <dict>
    <key>DSDT</key>
    <dict>
      <key>Fixes</key>
      <dict>
        <key>FixHPET</key>
        <true/>
        <key>FixIPIC</key>
        <true/>
        <key>FixRTC</key>
        <true/>
        <key>FixTMR</key>
        <true/>
      </dict>
    <key>Patches</key>
    <array>
      <dict>
        <key>Comment</key>
```

```

    <string>change OSID to XSID (to avoid match against _OSI
XOSI patch)</string>
    <key>Disabled</key>
    <false/>
    <key>Find</key>
    <data>
    T1NJRA==
    </data>
    <key>Replace</key>
    <data>
    WFNJRA==
    </data>
  </dict>
  <dict>
    <key>Comment</key>
    <string>change _OSI to XOSI</string>
    <key>Disabled</key>
    <false/>
    <key>Find</key>
    <data>
    X09TSQ==
    </data>
    <key>Replace</key>
    <data>
    WE9TSQ==
    </data>
  </dict>
  <dict>
    <key>Comment</key>
    <string>change SAT0 to SATA</string>
    <key>Disabled</key>
    <false/>
    <key>Find</key>
    <data>
    U0FUMA==
    </data>
    <key>Replace</key>
    <data>
    U0FUQQ==
    </data>
  </dict>
</array>

```

Mehr Beispiele

Ich werde versuchen, meine Ohren offen zu halten für weitere Beispiele zur Plist-Bearbeitung, mit denen Leute Probleme haben, und sie bei Bedarf hinzufügen.

Ivy Bridge

Wir gehen die Abschnitte der config.plist durch, einen nach dem anderen für ein Ivy Bridge-Desktop-Setup.

Ausgangspunkte

Ich beginne gerne entweder mit der config.plist, die Clover Ihnen zur Verfügung stellt, oder mit einer leeren Leinwand. In den nächsten Beispielen zeige ich Ihnen, wie ich die Dinge von Grund auf neu einrichte; wenn Sie von woanders anfangen, haben Sie vielleicht mehr Dinge überprüft/eingestellt als ich - aber Sie werden dem folgen wollen, was ich mache.

Ich werde auch die rohen xml-Beispiele mit einbeziehen, um denjenigen das Vorgehen zu zeigen, die mit einem Texteditor arbeiten (wie ich es bevorzuge).

ACPI

Die Standardeinstellungen von Clover sind ziemlich übertrieben und können einige Probleme verursachen. Wir werden diesen Abschnitt ziemlich minimal halten, und ich werde auch ein bisschen darauf eingehen, warum wir das für jeden Teil tun.

Raw-XML

```
<key>ACPI</key>
<dict>
  <key>DSDT</key>
  <dict>
    <key>Fixes</key>
    <dict>
      <key>AddMCHC</key>
      <true/>
      <key>FixHPET</key>
      <true/>
      <key>FixIPIC</key>
      <true/>
      <key>FixRTC</key>
      <true/>
      <key>FixShutdown</key>
      <true/>
      <key>FixTMR</key>
      <true/>
    </dict>
  <key>Patches</key>
  <array>
    <dict>
      <key>Comment</key>
      <string>change EHC1 to EH01</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
        RUhDMQ==
      </data>
      <key>Replace</key>
      <data>
        RUGwMQ==
      </data>
    </dict>
    <dict>
      <key>Comment</key>
      <string>change EHC2 to EH02</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
```

```

        RUhDMg==
      </data>
      <key>Replace</key>
      <data>
        RUGwMg==
      </data>
    </dict>
    <dict>
      <key>Comment</key>
      <string>change XHCI to XHC</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
        WEhDSQ==
      </data>
      <key>Replace</key>
      <data>
        WEhDXw==
      </data>
    </dict>
    <dict>
      <key>Comment</key>
      <string>change XHC1 to XHC</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
        WEhDMQ==
      </data>
      <key>Replace</key>
      <data>
        WEhDXw==
      </data>
    </dict>
    <dict>
      <key>Comment</key>
      <string>change SAT0 to SATA</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
        U0FUMA==
      </data>
      <key>Replace</key>
      <data>
        U0FUQQ==
      </data>
    </dict>
  </array>
</dict>
<key>DropTables</key>
<array>
  <dict>
    <key>Signature</key>
    <string>SSDT</string>
    <key>TableId</key>
    <string>CpuPm</string>
  </dict>
  <dict>
    <key>Signature</key>
    <string>SSDT</string>
    <key>TableId</key>
    <string>Cpu0Ist</string>
  </dict>
</array>

```

```

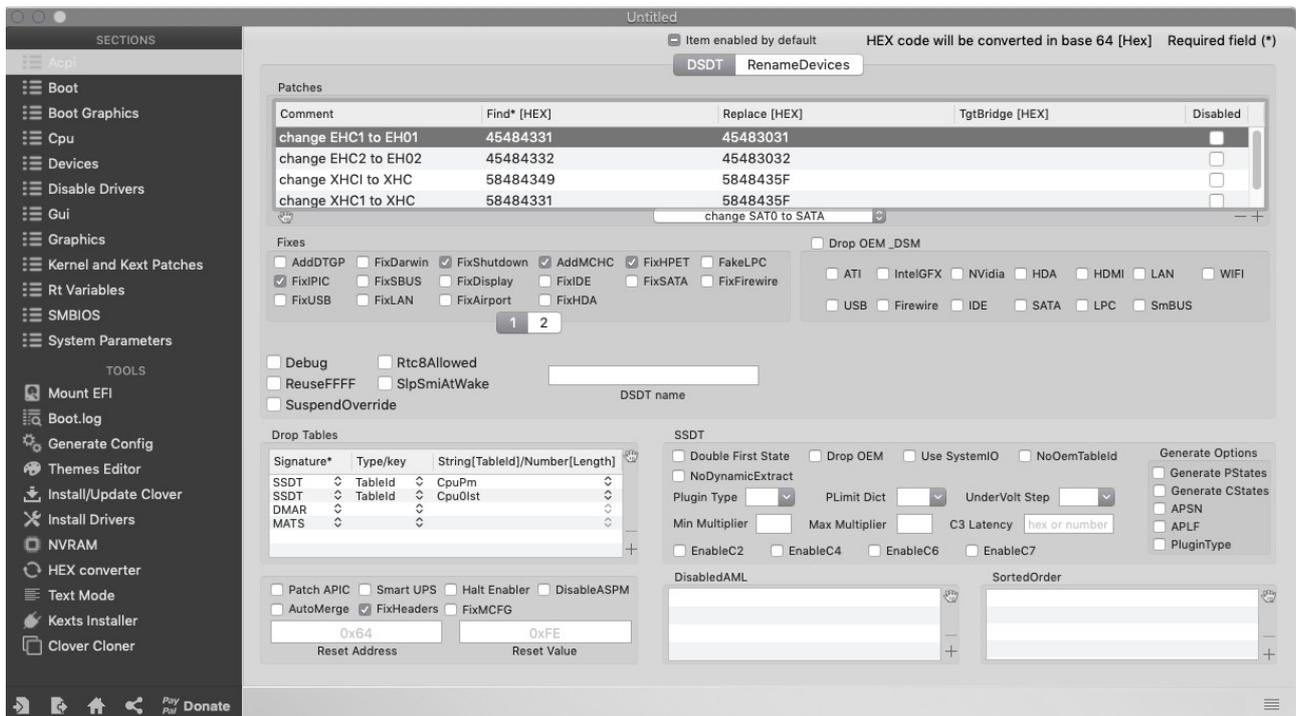
</dict>
<dict>
  <key>Signature</key>
  <string>DMAR</string>
</dict>
<dict>
  <key>Signature</key>
  <string>MATS</string>
</dict>
</array>
<key>FixHeaders</key>
<true/>
<key>SSDT</key>
<dict>
  <key>Generate</key>
  <dict/>
</dict>
</dict>

```

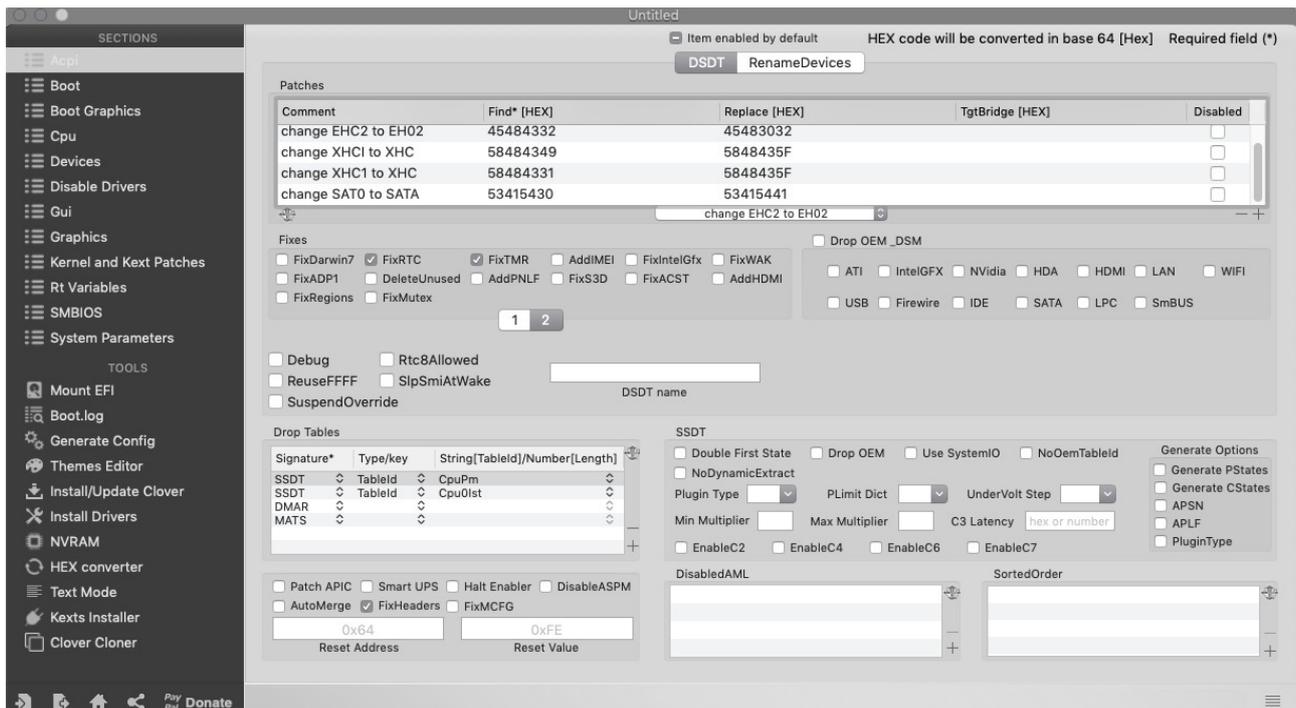
Clover-Konfigurator Screenshots

Erläuterung

Patches:



Ivy Acpi CC Section 1



Ivy Acpi CC Section 2

Als erstes werden wir den Bereich Patches durchgehen. Dieser Abschnitt ermöglicht es uns, Teile des DSDT über Clover dynamisch umzubenennen. Da wir keinen echten Mac betreiben und macOS ziemlich wählerisch ist, wie Dinge benannt werden, können wir nicht-destruktive Änderungen vornehmen, um die Dinge Mac-freundlich zu halten. Wir haben hier drei(?) Einträge:

- *change EHC1 to EH01* - helps avoid a conflict with built-in USB injectors
- *change EHC2 to EH02* - helps avoid a conflict with built-in USB injectors
- *change XHC1 to XHC* - helps avoid a conflict with built-in USB injectors
- *change XHCI to XHC* - helps avoid a conflict with built-in USB injectors
- *change SAT0 to SATA* - for potential SATA compatibility

Fixes:

Wenn wir uns dann den Abschnitt Fixes ansehen, werden wir sehen, dass wir ein paar Dinge angekreuzt haben (es gibt 2 Seiten, also habe ich 2 Screenshots eingefügt):

- **FixShutdown** - dies kann bei einigen Boards helfen, die es vorziehen, neu zu starten, anstatt herunterzufahren. Manchmal kann es bei anderen Boards zu Problemen beim Herunterfahren führen (ironisch, nicht wahr?), wenn Sie also Probleme beim Herunterfahren haben, wenn dies aktiviert ist, sollten Sie es deaktivieren.

- Die übrigen Korrekturen helfen, IRQ-Konflikte usw. zu vermeiden, und sind nicht dafür bekannt, dass sie Probleme verursachen. Sie sind vielleicht nicht für jede Hardware notwendig, haben aber keine negativen Auswirkungen, wenn sie angewendet werden. Hinweis: Wenn Sie eine Ivy Bridge-CPU mit einem Motherboard der 6er-Serie verwenden, müssen Sie auch AddDTGP und AddIMEI aktivieren und die IMEI auf 0x1e3a8086 fälschen (ich werde dies im Abschnitt "Geräte" erläutern).

Drop Tables:

Wir haben DSDT in unserem Abschnitt über Patches sanft gestreift - und dies ist eine kleine Erweiterung davon. SSDT ist wie ein Unterabschnitt von DSDT. Der Abschnitt "Drop Tables" erlaubt uns, bestimmte SSDT-Tabellen beim Laden auszulassen (wie ich bereits erwähnt habe, ist DSDT für Mac und PC unterschiedlich, und macOS kann ziemlich pingelig sein). Die beiden, die ich hinzugefügt habe, sind wie folgt:

- CpuPm und Cpu0Ist - diese lassen Tabellen fallen, die sich auf die CPU-Leistungsverwaltung beziehen, was uns erlaubt, die Daten zu ergänzen, indem wir eine SSDT.aml über das Skript ssdtPRGen.sh von Pike erzeugen.
- DMAR - dies verhindert einige Probleme mit Vt-d; das ist PCI-Passthrough für VMs, und nicht sehr funktional (wenn überhaupt?) auf Hackintoshes.
- MATS - ab High Sierra wird diese Tabelle geparkt, und kann manchmal nicht druckbare Zeichen enthalten, die zu einer Kernel-Panic führen können.

FixHeaders und Generate:

Die einzigen anderen Dinge, die wir auf dieser Seite getan haben, sind diese beiden Kontrollkästchen zu aktivieren.

- FixHeaders - dies ist nur eine Verdoppelung unseres MATS-Tabellenabwurfs. Dieses Kontrollkästchen weist Clover an, Header zu bereinigen, um Kernel-Panics im Zusammenhang mit nicht druckbaren Zeichen zu vermeiden.

- Generate - dies wird auf ein leeres Wörterbuch gesetzt (Sie können dasselbe erreichen, indem Sie eine der Generate-Optionen ankreuzen und sie dann in CC deaktivieren), was verhindert, dass Clover irgendwelche C- oder P-Zustände generiert, und auch verhindert, dass es PluginType hinzufügt (da dies für Haswell und neuere CPUs ist).

Boot

Wir brauchen hier nicht viel zu tun, aber wir werden ein paar Dinge ändern.

Raw XML

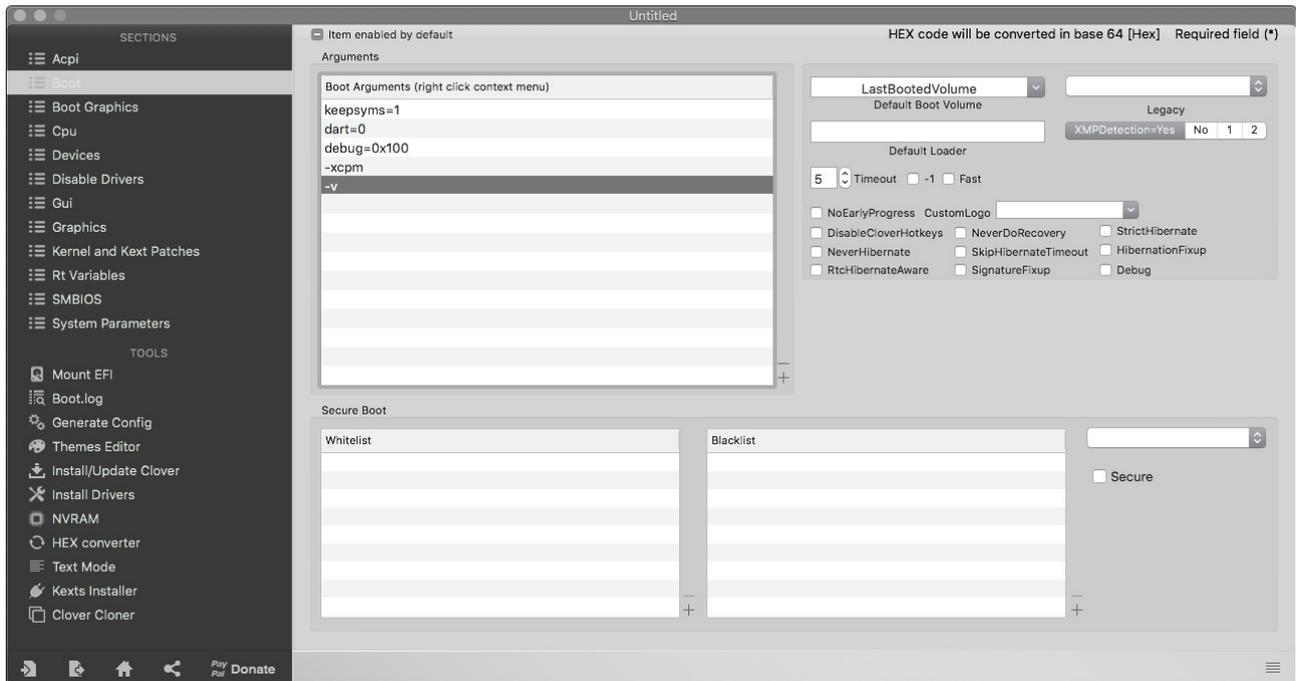
```
<key>Boot</key>
<dict>
  <key>Arguments</key>
  <string>keepsyms=1 dart=0 debug=0x100 -xcpm -v</string>
  <key>DefaultVolume</key>
  <string>LastBootedVolume</string>
  <key>Timeout</key>
  <integer>5</integer>
</dict>
```

Clover Configurator Screenshots

Erläuterungen

Argumente

Wir haben hier einige Boot-Args eingestellt:



-v - dies aktiviert den ausführlichen Modus, der den gesamten Text hinter den Kulissen anzeigt, der beim Booten anstelle des Apple-Logos und des Fortschrittsbalkens vorbeiläuft. Dies ist für jeden Hackintosh von unschätzbarem Wert, da es Ihnen einen Einblick in den Boot-Prozess gibt und Ihnen helfen kann, Probleme, Problem-Kexts usw. zu identifizieren.

- dart=0 - dies ist nur ein zusätzlicher Schutz gegen Vt-d-Probleme.

- debug=0x100 - dies verhindert einen Neustart bei einer Kernel-Panik. Auf diese Weise können Sie (hoffentlich) einige nützliche Informationen sammeln und den Brotkrumen folgen, um die Probleme zu umgehen.

- keepsyms=1 - dies ist eine begleitende Einstellung zu debug=0x100, die dem Betriebssystem sagt, dass es bei einer Kernel-Panik auch die Symbole ausgeben soll. Das kann einige hilfreichere Einblicke geben, was die Panik selbst verursacht.

-xcpm - versucht, Ivy-CPUs zur Verwendung von XnuCPUPowerManagement zu zwingen.

DefaultBootVolume und Timeout:

Dies sind die einzigen anderen Einstellungen, die ich in diesem Abschnitt aktualisiert habe.

- DefaultBootVolume - dies verwendet NVRAM, um sich zu merken, welches Volume zuletzt von Clover gebootet wurde, und wählt dieses beim nächsten Booten automatisch aus.

Timeout - dies ist die Anzahl der Sekunden, bevor das DefaultBootVolume automatisch bootet. Sie können diesen Wert auf -1 setzen, um alle Timeouts zu vermeiden, oder auf 0, um die GUI komplett zu überspringen. Wenn Sie diesen Wert auf 0 setzen, können Sie beim Booten beliebige Tasten drücken, damit die GUI bei Problemen wieder angezeigt wird.

Boot-Grafik

Hier ist nichts - nur die Standardeinstellungen. Sie könnten dies anpassen, wenn die Skalierung von Clover geändert werden muss, aber ich pfusche nicht daran herum.

Cpu

Auch hier wird in den meisten Setups, mit denen ich gearbeitet habe, nichts verändert.

Devices

Hier werden wir einige Eigenschaften für WhateverGreen einfügen und einige grundlegende Audioeinstellungen vornehmen.

Raw XML

```
<key>Devices</key>
<dict>
  <key>Audio</key>
  <dict>
    <key>Inject</key>
    <integer>1</integer>
    <key>ResetHDA</key>
    <true/>
  </dict>
  <key>Properties</key>
  <dict>
    <key>PciRoot(0x0)/Pci(0x2,0x0)</key>
    <dict>
      <key>AAPL,ig-platform-id</key>
      <data>
        CgBMAQ==
      </data>
    </dict>
  </dict>
  <key>USB</key>
  <dict>
    <key>FixOwnership</key>
    <true/>
  </dict>
</dict>
```

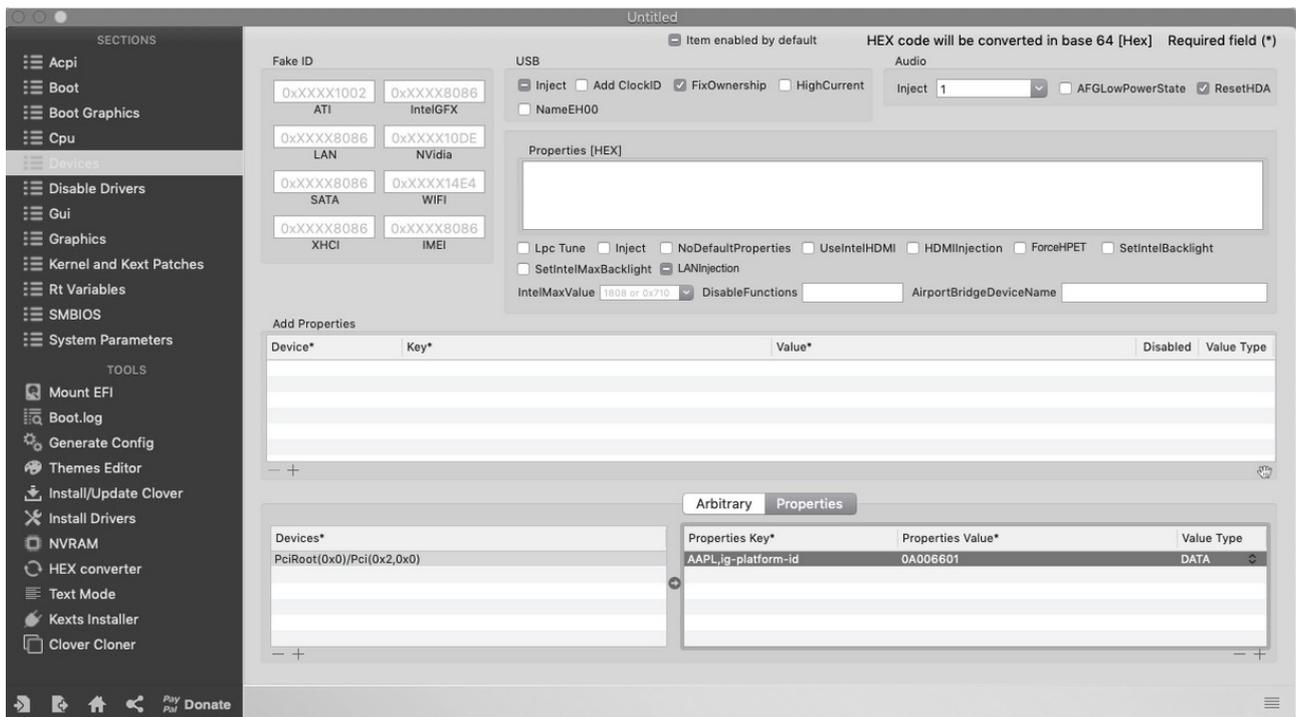
Clover Configurator Screenshots

Explanation

Fake ID:

Dieser Abschnitt bleibt für unser Beispiel-Setup leer. Wenn Sie jedoch eine Ivy-CPU mit einem 6er-Board betreiben, müssen Sie Ihre IMEI wie folgt fälschen lassen:

```
<key>FakeID</key>
<dict>
  <key>IMEI</key>
  <string>0x1e3a8086</string>
</dict>
```



Ivy Devices CC Section

USB:

In diesem Abschnitt stellen wir sicher, dass **Inject** und **FixOwnership** ausgewählt sind, um Probleme mit dem Hängenbleiben an einer halb ausgedruckten Zeile irgendwo um die **Verbose-Zeile Enabling Legacy Matching** zu vermeiden. Sie können das auch umgehen, indem Sie **XHCI Hand Off** im BIOS aktivieren.

Audio:

Hier haben wir unser **Audio** auf **inject Layout 1** gesetzt - dies kann mit Ihrem Codec kompatibel sein oder auch nicht, aber Sie können auf der **AppleALC**-Seite für unterstützte Codecs nachsehen.

Wir haben auch **ResetHDA** aktiviert, was den Codec zwischen den Neustarts des Betriebssystems in einen neutralen Zustand versetzt. Dies verhindert einige Probleme mit fehlendem Audio nach dem Booten zu einem anderen Betriebssystem und dann zurück.

Properties:

Dieser Abschnitt wird über [Intel Framebuffer Patching Guide](#) eingerichtet und wendet nur eine tatsächliche Eigenschaft an, nämlich die **ig-platform-id**. Um den richtigen Wert zu erhalten, schauen wir uns die **ig-platform-id** an, die wir verwenden wollen, und tauschen dann die Hex-Byte-Paare aus. Wir gehen hier davon aus, dass der Benutzer eine **HD 4000** hat - **HD 2000** und **2500** werden von macOS nicht unterstützt, und wenn Sie diese haben, können Sie die folgenden iGPU-bezogenen Informationen ignorieren.

Wenn wir uns unsere **ig-plat** als **0xAABBCCDD** vorstellen, würde unsere getauschte Version wie **0xDDCCBBAA** aussehen.

Die **ig-platform-id**, die wir verwenden, lautet wie folgt:

- 0x0166000A - this is the standard hex for the ig-plat
 - 0A006601 when hex-swapped
 - CgBmAQ== when the hex-swapped version is converted to base64

Disable Drivers

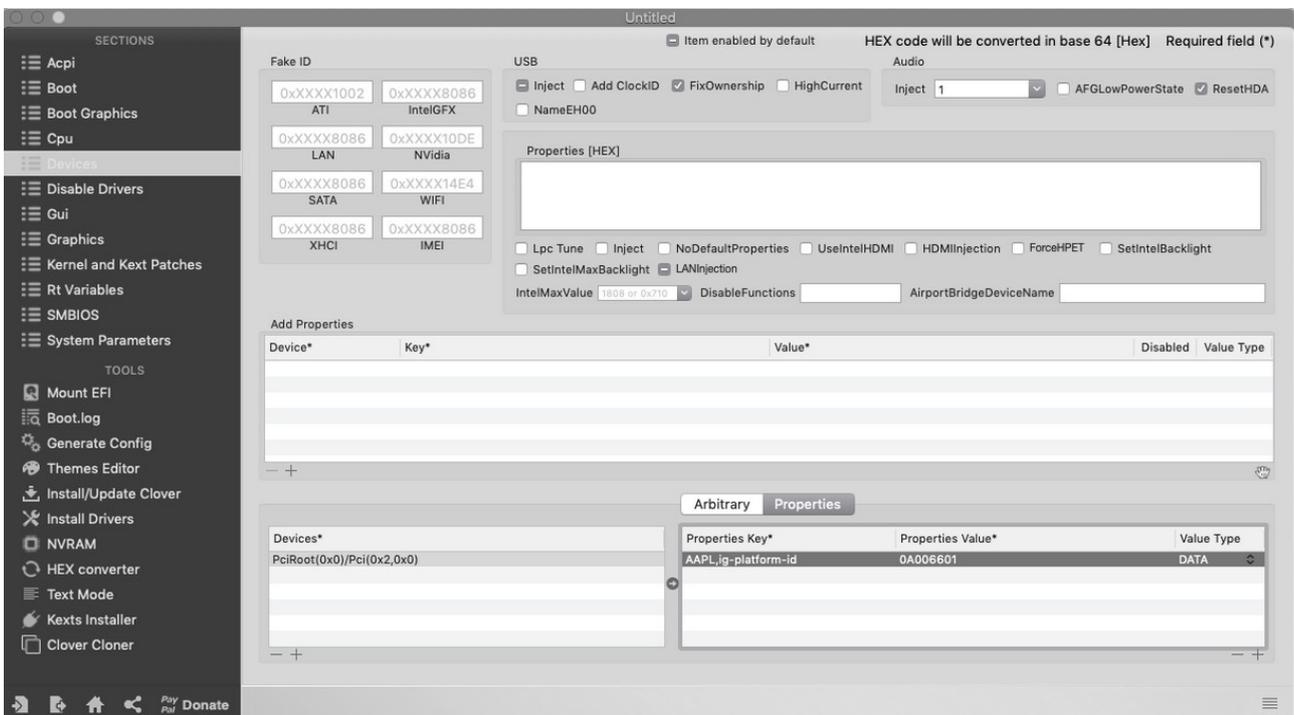
Hier gibt es nichts zu tun.

Gui

Raw XML

```
<key>GUI</key>
<dict>
  <key>Scan</key>
  <dict>
    <key>Entries</key>
    <true/>
    <key>Tool</key>
    <true/>
  </dict>
</dict>
```

Clover Configurator Screenshots



Ivy Gui CC Section

Erläuterung

Scan:

Die einzigen Einstellungen, die ich auf dieser Seite geändert habe, sind die Scan-Einstellungen. Ich habe "Benutzerdefiniert" ausgewählt und dann alles außer "Legacy" und "Kernel" markiert. Dies lässt nur einige der nicht bootfähigen Einträge in Clover weg, um das Menü aufzuräumen.

Volumes ausblenden:

Ich habe hier nichts hinzugefügt, aber Sie können unerwünschte Volumes hier ausblenden. Sie können dies tun, indem Sie entweder den Namen des Volumes oder die UUID hinzufügen.

Um zusätzliche APFS-Einträge auszublenden, fügen Sie Folgendes zu dieser Liste hinzu:

- Preboot
- VM

Um alle Recovery-Partitionen auszublenden, fügen Sie Recovery zur Liste hinzu.

Um die UUID eines auszublendenden Laufwerks zu erhalten, können Sie den folgenden Terminalbefehl verwenden:

```
diskutil info diskXsY | grep -i "Partition UUID" | rev | cut -d ' ' -f 1 | rev
```

Stellen Sie sicher, dass Sie diskXsY durch die tatsächliche Festplattennummer des Volumes ersetzen, das Sie ausblenden möchten

Theme:

Wenn Sie ein neues Thema ausprobieren möchten (und ich empfehle Ihnen, sich [clover-next-black](#) anzusehen), können Sie den entpackten Themenordner zum Verzeichnis /Volumes/EFI/EFI/CLOVER/themes hinzufügen und dann den Namen des Ordners in das Textfeld Thema eingeben, um ihn anzuwenden.

Graphics

Raw XML

```
<key>Graphics</key>
<dict>
  <key>Inject</key>
  <false/>
</dict>
```

Kernel And Kext Patches

Raw XML

Raw XML

```
<key>KernelAndKextPatches</key>
<dict>
  <key>AppleIntelCPUPM</key>
  <true/>
  <key>KernelPm</key>
  <true/>
  <key>KextsToPatch</key>
  <array>
    <dict>
      <key>Comment</key>
      <string>Port limit increase</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
        g710////EA==
      </data>
      <key>InfoPlistPatch</key>
      <false/>
      <key>MatchOS</key>
      <string>10.12.x</string>
      <key>Name</key>
      <string>com.apple.driver.usb.AppleUSBXHCI</string>
      <key>Replace</key>
      <data>
        g710////Gw==
      </data>
    </dict>
  <dict>
    <key>Comment</key>
    <string>Port limit increase (RehabMan)</string>
    <key>Disabled</key>
    <false/>
    <key>Find</key>
    <data>
      g32IDw+DpwQAAA==
    </data>
    <key>InfoPlistPatch</key>
    <false/>
    <key>MatchOS</key>
    <string>10.13.x</string>
    <key>Name</key>
    <string>com.apple.driver.usb.AppleUSBXHCI</string>
    <key>Replace</key>
    <data>
      g32ID5CQkJCQkA==
    </data>
  </dict>
  <dict>
    <key>Comment</key>
    <string>Port limit increase (Ricky)</string>
    <key>Disabled</key>
    <false/>
    <key>Find</key>
    <data>
      g/SPD40PBAAA
    </data>
    <key>InfoPlistPatch</key>
    <false/>
    <key>MatchOS</key>
    <string>10.14.x</string>
  </dict>
</dict>
```

```

    <key>Name</key>
    <string>com.apple.driver.usb.AppleUSBXHCI</string>
    <key>Replace</key>
    <data>
    g/sPkJCQkJCQ
    </data>
  </dict>
  <dict>
    <key>Comment</key>
    <string>External Icons Patch</string>
    <key>Disabled</key>
    <false/>
    <key>Find</key>
    <data>
    RXh0ZXJuYWw=
    </data>
    <key>InfoPlistPatch</key>
    <false/>
    <key>Name</key>
    <string>AppleAHCIPort</string>
    <key>Replace</key>
    <data>
    SW50ZXJuYWw=
    </data>
  </dict>
</array>
</dict>

```

Clover Configurator Screenshots

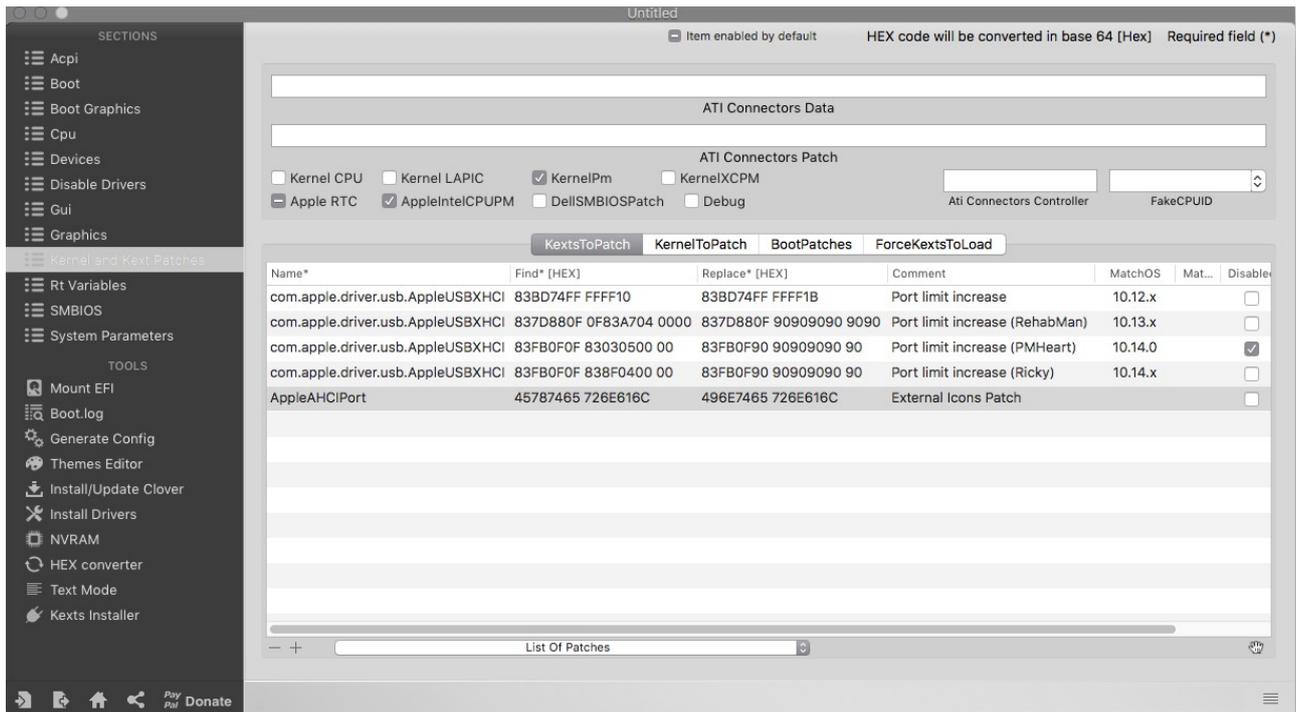
Explanation

In diesem Abschnitt haben wir ein paar Einstellungen aktiviert und einige Kext-Patches hinzugefügt.

Checkboxes:

In diesem Abschnitt haben wir ein paar Einstellungen aktiviert und einige Kext-Patches hinzugefügt:

- Apple RTC - dies stellt sicher, dass wir keinen BIOS-Reset beim Neustart haben.
- KernelPM - diese Einstellung verhindert das Schreiben auf MSR 0xe2, was eine Kernel-Panik beim Booten verhindern kann, wenn XCPM verwendet wird.
- AppleIntelCPUPM - dies bewirkt das Gleiche wie KernelPM, aber bei Verwendung von AppleIntelCPUPowerManagement stattdessen.



Ivy KernelAndKextPatches CC Section

KextsToPatch:

Wir haben hier 5 verschiedene Kexts zum Patchen hinzugefügt. Vier davon sind für die Erhöhung des USB-Port-Limits und der letzte dient als Fix für orangefarbene Icons - wenn interne Laufwerke hotplug-fähig sind und als externe Laufwerke behandelt werden. Sie werden feststellen, dass für jeden der USB-Port-Limit-Patches MatchOS-Werte gesetzt sind. Sie können alle Einträge für Betriebssystemversionen, die Sie nicht verwenden wollen, entfernen. Sie schaden nicht, wenn sie da sind, aber wenn Sie eine saubere, minimale Liste wollen, ist es nicht sinnvoll, sie zu haben.

RtVariables And SMBIOS

Raw XML

```
<key>RtVariables</key>
<dict>
  <key>BooterConfig</key>
  <string>0x28</string>
  <key>CsrActiveConfig</key>
  <string>0x3E7</string>
  <key>MLB</key>
  <string>C02253902J9F2FRCB</string>
  <key>ROM</key>
  <string>UseMacAddr0</string>
</dict>
<key>SMBIOS</key>
<dict>
  <key>BoardSerialNumber</key>
  <string>C02253902J9F2FRCB</string>
```

```
<key>ProductName</key>
<string>iMac13,2</string>
<key>SerialNumber</key>
<string>C02JX0KSDNCW</string>
<key>SmUUID</key>
<string>FDDCE665-D5C0-4738-8F80-77380686E42B</string>
</dict>
```

Clover Configurator Screenshots

Erläuterung

Zum Einrichten der SMBIOS-Informationen verwende ich die [macserial](#) von acidanthera. Ich habe ein [python script](#) geschrieben, das es ebenfalls nutzen kann (und automatisch in der config.plist speichert, wenn es ausgewählt wird). Es gibt viele Informationen, die leer gelassen werden, damit Clover die Leerstellen ausfüllen kann; das bedeutet, dass eine Aktualisierung von Clover die übergebenen Informationen aktualisiert und Sie nicht auch Ihre config.plist aktualisieren müssen.

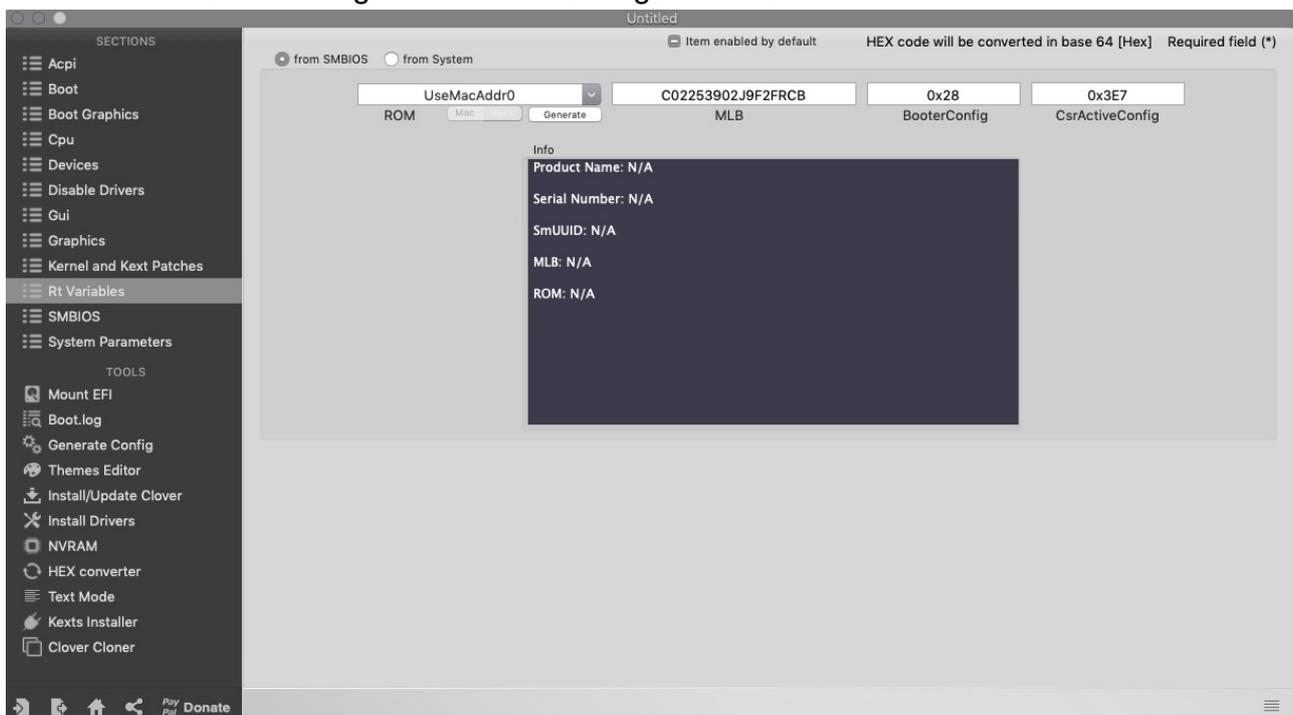
Für dieses Ivy Bridge Beispiel habe ich das iMac13,2 SMBIOS gewählt.

Um die SMBIOS-Informationen zu erhalten, die mit macserial generiert wurden, können Sie es mit dem Argument -a ausführen (das Seriennummern und Board-Seriennummern für alle unterstützten Plattformen generiert). Sie können es auch mit grep parsen, um Ihre Suche auf einen SMBIOS-Typ zu beschränken.

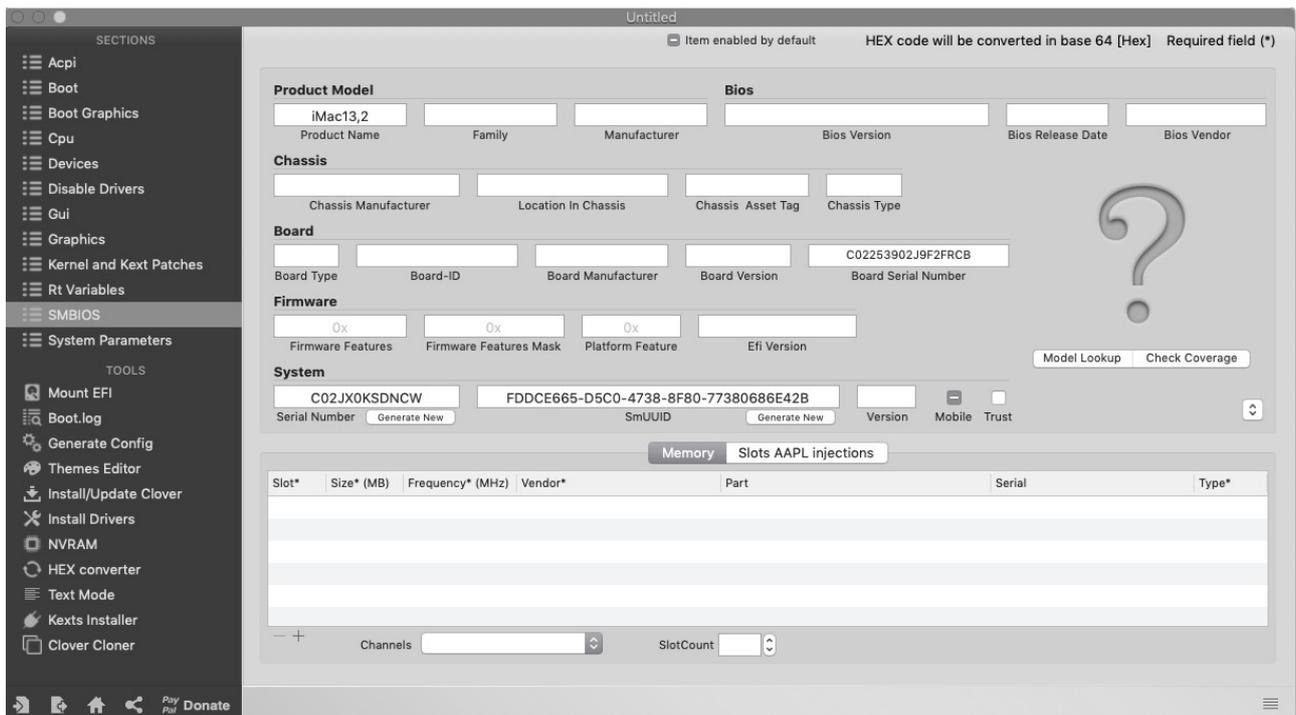
Bei unserem iMac13,2-Beispiel würden wir macserial wie folgt über das Terminal ausführen:

```
macserial -a | grep -i iMac13,2
```

Das würde uns eine Ausgabe ähnlich der folgenden liefern:



Ivy Rt Variables CC Section



Ivy SMBIOS CC Section

Product	Serial	Board Serial (MLB)
iMac13,2	C02JX0KSDNCW	C02253902J9F2FRCB
iMac13,2	C02KC3Y9DNCW	C02309401GUF2FR1M
iMac13,2	C02KPYDDNCW	C02319902J9F2FRJA
iMac13,2	C02LLSYJDNCW	C02343301J9F2FR1F
iMac13,2	C02JF0N3DNCW	C02238404GUF2FR1M
iMac13,2	C02JT6ZFDNCW	C02250104QXF2FRFB
iMac13,2	C02LT0GDDNCW	C02350130J9F2FR1H
iMac13,2	C02J1069DNCW	C02227310J9F2FRAD
iMac13,2	C02JN0Y4DNCW	C02245902CDF2FRFB
iMac13,2	C02JWJZSDNCW	C02252100GUF2FRCB

Der Teil iMac13,2 wird nach SMBIOS -> Produktname kopiert.

Der Teil "Serial" wird nach SMBIOS -> Serial Number kopiert.

Der Teil "Board Serial" wird zu SMBIOS -> Board Serial Number sowie zu Rt Variables -> MLB kopiert.

Wir können eine SmUUID erstellen, indem wir uuidgen im Terminal ausführen (oder sie wird automatisch über mein GenSMBIOS-Skript generiert) - und diese wird nach SMBIOS -> SmUUID kopiert.

Wir setzen Rt Variables -> ROM auf UseMacAddr0, was einfach unsere Onboard-Mac-Adresse verwendet - diese sollte eindeutig genug sein, um nicht mit anderen in Konflikt zu geraten.

BooterConfig wird auf 0x28 gesetzt, und CsrActiveConfig wird auf 0x3e7 gesetzt, was SIP effektiv deaktiviert. Sie können eine Reihe von anderen Optionen wählen, um Abschnitte von SIP zu aktivieren/deaktivieren. Einige gängige sind wie folgt:

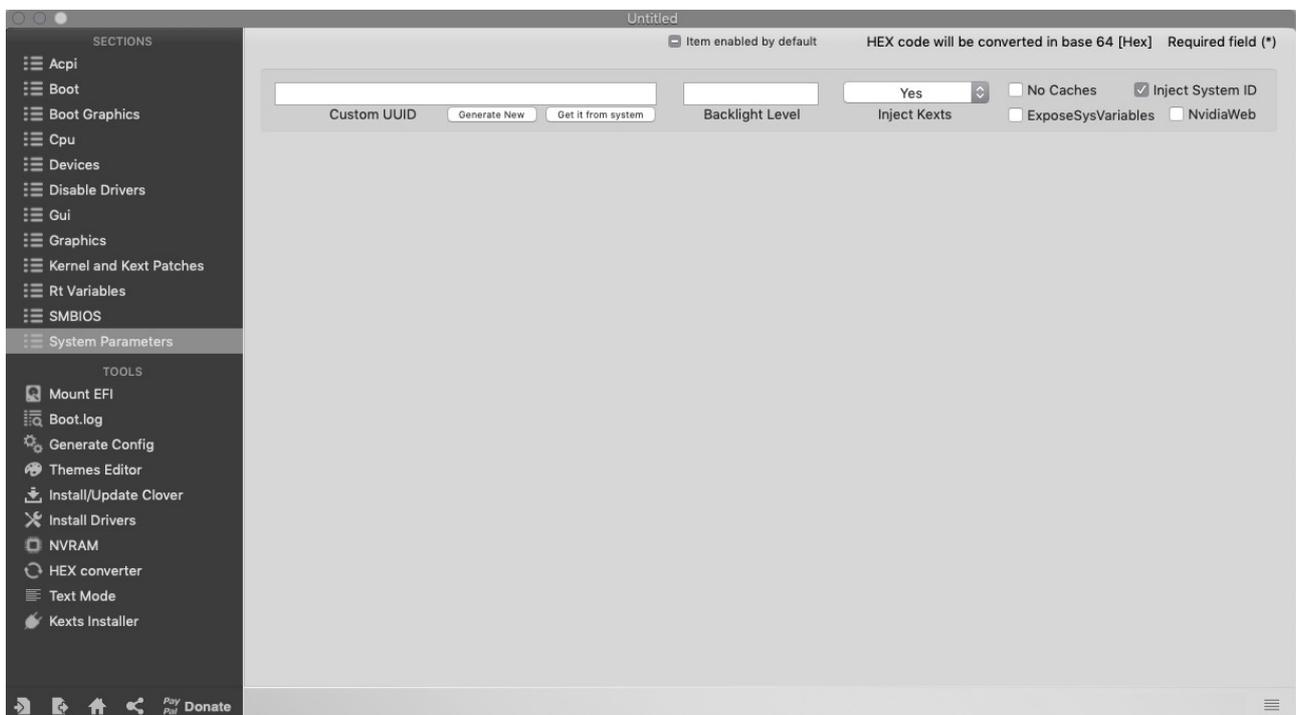
- 0x0 - SIP vollständig aktiviert
- 0x3 - Erlaubt unsignierte Kexts und das Schreiben auf geschützte fs-Speicherplätze
- 0x3e7 - SIP komplett deaktiviert

System Parameters

Raw XML

```
<key>SystemParameters</key>
<dict>
  <key>InjectKexts</key>
  <string>Yes</string>
  <key>InjectSystemID</key>
  <true/>
</dict>
```

Clover Configurator Screenshots



System Parameters CC Section

Explanation

Inject Kexts:

Diese Einstellung hat 3 Modi:

- Ja - damit wird Clover angewiesen, Kexts aus der EFI unabhängig davon zu injizieren.
- Nein - damit wird Clover angewiesen, keine Kexts von der EFI zu injizieren.
- Detect - Clover injiziert Kexts nur, wenn FakeSMC.kext oder VirtualSMC.kext nicht im Kext-Cache sind.

Wir setzen es auf Yes, um sicherzustellen, dass alle Kexts, die wir zuvor hinzugefügt haben, korrekt injiziert werden.

InjectSystemID

Diese Einstellung weist Clover an, die SmUUID als System-ID beim Booten zu setzen - was wichtig für iMessage und so ist.

Speichern

An dieser Stelle können Sie File -> Save wählen, um die config.plist zu speichern. Wenn Sie Probleme haben, direkt auf der EFI zu speichern, können Sie sie auf dem Desktop speichern und dann einfach rüberkopieren. Ich lasse die [sample config.plist here](#)

Haswell

Wir gehen die Abschnitte der config.plist durch, einen nach dem anderen für ein Haswell-Desktop-Setup.

Ausgangspunkte

Ich beginne gerne entweder mit der config.plist, die Clover Ihnen zur Verfügung stellt, oder mit einer leeren Leinwand. In den nächsten Beispielen zeige ich Ihnen, wie ich die Dinge von Grund auf neu einrichte; wenn Sie von woanders anfangen, haben Sie vielleicht mehr Dinge überprüft/eingestellt als ich - aber Sie werden dem folgen wollen, was ich mache. Ich werde auch die rohen xml-Beispiele mit einbeziehen, um denjenigen zu zeigen, die mit einem Texteditor arbeiten (wie ich es bevorzuge).

ACPI

Die Standardeinstellungen von Clover sind ziemlich übertrieben und können einige Probleme verursachen. Wir werden diesen Abschnitt ziemlich minimal halten, und ich werde auch ein bisschen darauf eingehen, warum wir das für jeden Teil tun.

Raw XML

```
<key>ACPI</key>
<dict>
  <key>DSDT</key>
  <dict>
    <key>Fixes</key>
    <dict>
      <key>AddMCHC</key>
      <true/>
      <key>FixHPET</key>
      <true/>
      <key>FixIPIC</key>
      <true/>
    
```

```
<key>FixRTC</key>
<true/>
<key>FixShutdown</key>
<true/>
<key>FixTMR</key>
<true/>
</dict>
<key>Patches</key>
<array>
  <dict>
    <key>Comment</key>
    <string>change EHC1 to EH01</string>
    <key>Disabled</key>
    <false/>
    <key>Find</key>
    <data>
      RUhDMQ==
    </data>
    <key>Replace</key>
    <data>
      RUGwMQ==
    </data>
  </dict>
  <dict>
    <key>Comment</key>
    <string>change EHC2 to EH02</string>
    <key>Disabled</key>
    <false/>
    <key>Find</key>
    <data>
      RUhDMg==
    </data>
    <key>Replace</key>
    <data>
      RUGwMg==
    </data>
  </dict>
  <dict>
    <key>Comment</key>
    <string>change XHCI to XHC</string>
    <key>Disabled</key>
    <false/>
    <key>Find</key>
    <data>
      WEhDSQ==
    </data>
    <key>Replace</key>
    <data>
      WEhDXw==
    </data>
  </dict>
  <dict>
    <key>Comment</key>
    <string>change XHC1 to XHC</string>
    <key>Disabled</key>
    <false/>
    <key>Find</key>
    <data>
      WEhDMQ==
    </data>
    <key>Replace</key>
    <data>
      WEhDXw==
    </data>
  </dict>

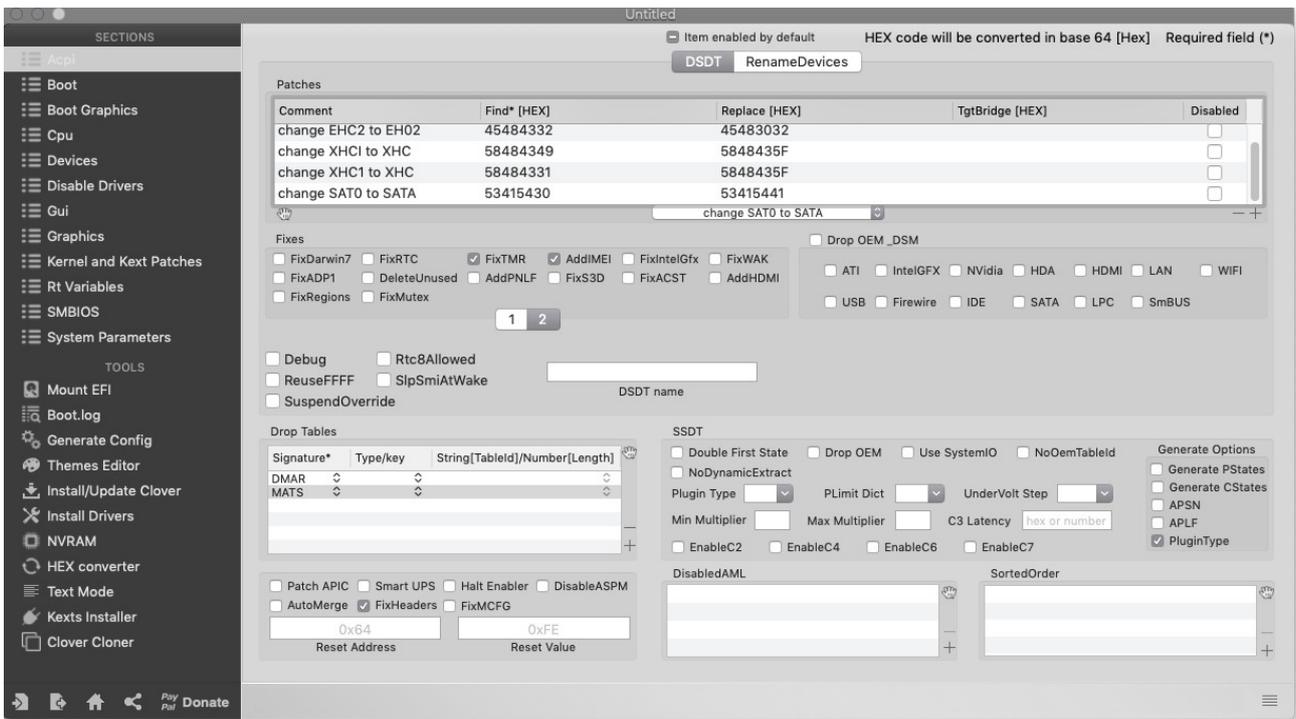
```

```

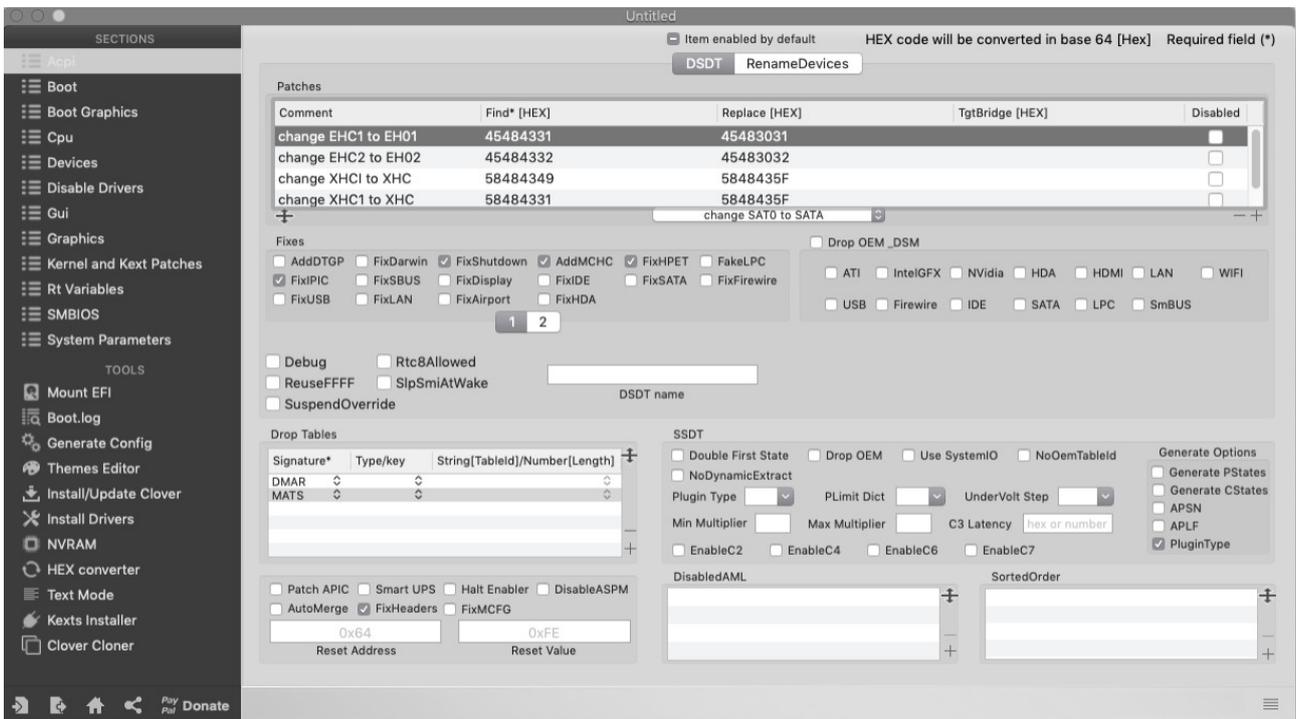
    </dict>
    <dict>
      <key>Comment</key>
      <string>change SAT0 to SATA</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
        U0FUUMA==
      </data>
      <key>Replace</key>
      <data>
        U0FUUQQ==
      </data>
    </dict>
  </array>
</dict>
<key>DropTables</key>
<array>
  <dict>
    <key>Signature</key>
    <string>DMAR</string>
  </dict>
  <dict>
    <key>Signature</key>
    <string>MATS</string>
  </dict>
</array>
<key>FixHeaders</key>
<true/>
<key>SSDT</key>
<dict>
  <key>Generate</key>
  <dict>
    <key>PluginType</key>
    <true/>
  </dict>
</dict>
</dict>
</dict>

```

Clover Configurator Screenshots



Haswell Acpi CC Section 1



Haswell Acpi CC Section 2

Erläuterung

Patches:

Das erste, was wir durchgehen, ist der Abschnitt Patches. Dieser Abschnitt ermöglicht es uns, Teile des DSDT über Clover dynamisch umzubenennen. Da wir keinen echten Mac verwenden und macOS ziemlich genau darauf achtet, wie Dinge benannt werden, können wir nicht-destruktive Änderungen vornehmen, um die Dinge Mac-freundlich zu halten. Wir haben hier drei Einträge:

- EHC1 in EH01 ändern - hilft, einen Konflikt mit eingebauten USB-Injektoren zu vermeiden
- EHC2 in EH02 ändern - hilft, einen Konflikt mit eingebauten USB-Injektoren zu vermeiden
- XHC1 nach XHC ändern - hilft, einen Konflikt mit eingebauten USB-Injektoren zu vermeiden
- XHCI zu XHC ändern - hilft, einen Konflikt mit eingebauten USB-Injektoren zu vermeiden
- SAT0 auf SATA ändern - für mögliche SATA-Kompatibilität

Fixes:

Wenn wir uns dann den Abschnitt Fixes ansehen, werden wir sehen, dass wir ein paar Dinge angekreuzt haben (es gibt 2 Seiten, daher habe ich 2 Screenshots eingefügt):

- FixShutdown - dies kann bei einigen Boards helfen, die es vorziehen, neu zu starten, anstatt herunterzufahren. Manchmal kann es bei anderen Boards zu Problemen beim Herunterfahren führen (ironisch, nicht wahr?), wenn Sie also Probleme beim Herunterfahren haben, wenn dies aktiviert ist, sollten Sie es deaktivieren.
- Die übrigen Korrekturen helfen, IRQ-Konflikte usw. zu vermeiden, und sind nicht dafür bekannt, dass sie Probleme verursachen. Sie sind vielleicht nicht für jede Hardware notwendig, haben aber keine negativen Auswirkungen, wenn sie angewendet werden.

Drop Tables:

Wir haben DSDT mit unserem Abschnitt "Patches" sanft gestreift - und dies ist eine kleine Erweiterung davon. SSDT ist wie ein Unterabschnitt von DSDT. Der Abschnitt Drop Tables erlaubt es uns, bestimmte SSDT-Tabellen beim Laden auszulassen (wie ich bereits erwähnt habe, ist Mac- und PC-DSDT unterschiedlich, und macOS kann ziemlich pingelig sein). Die beiden, die ich hinzugefügt habe, sind wie folgt:

- DMAR - dies verhindert einige Probleme mit Vt-d; das ist PCI-Passthrough für VMs, und nicht sehr funktionell (wenn überhaupt?) auf Hackintoshes.
- MATS - ab High Sierra wird diese Tabelle geparkt und kann manchmal nicht druckbare Zeichen enthalten, die zu einer Kernel-Panik führen können.

FixHeaders and PluginType:

Die einzigen anderen Dinge, die wir auf dieser Seite getan haben, sind diese beiden Kontrollkästchen zu aktivieren.

- FixHeaders - dies ist nur eine Verdoppelung unseres MATS-Tabellenabwurfs. Dieses Kontrollkästchen weist Clover an, Header zu bereinigen, um Kernel-Paniks im Zusammenhang mit nicht druckbaren Zeichen zu vermeiden.
- PluginType - dies injiziert einige DSDT-Daten, um X86PlatformPlugin zu laden - was uns einen Vorsprung bei der nativen CPU-Leistungsverwaltung verschafft. Diese Einstellung funktioniert allerdings nur auf Haswell und neueren CPUs.

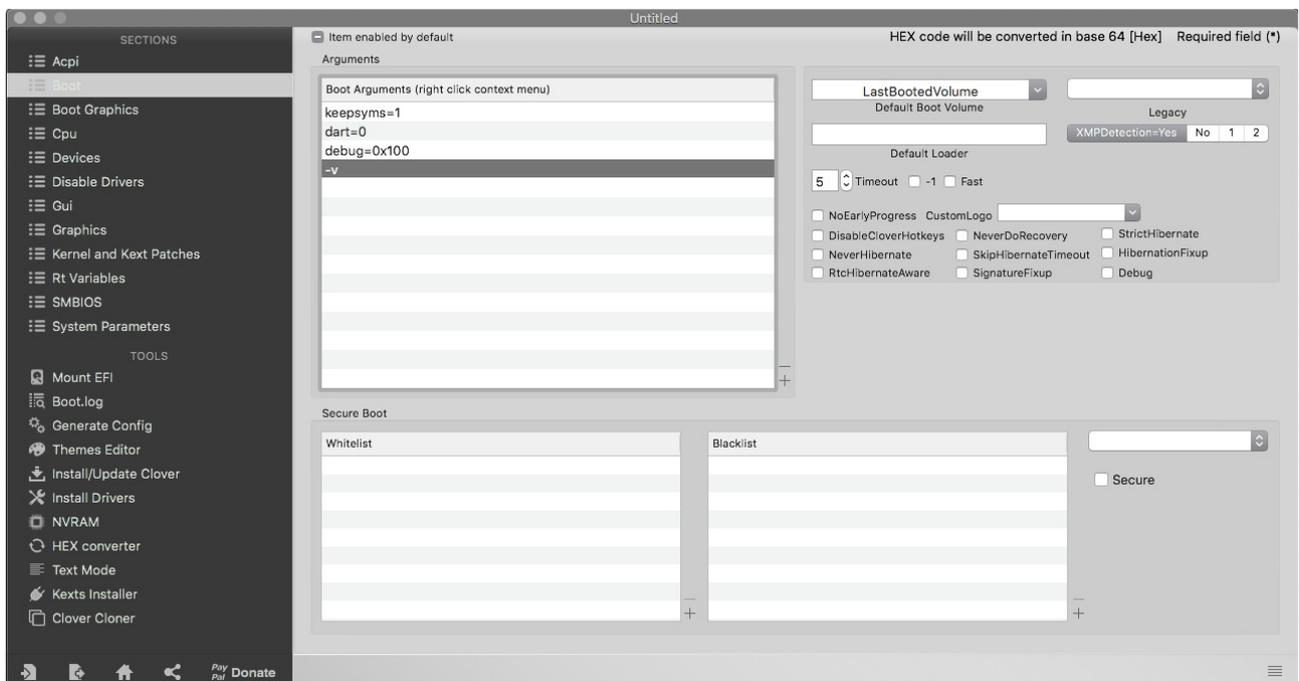
Boot

Wir brauchen hier nicht viel zu tun, aber wir werden ein paar Dinge ändern.

Raw XML

```
<key>Boot</key>
<dict>
  <key>Arguments</key>
  <string>keepsyms=1 dart=0 debug=0x100 -v</string>
  <key>DefaultVolume</key>
  <string>LastBootedVolume</string>
  <key>Timeout</key>
  <integer>5</integer>
</dict>
```

Clover Configurator Screenshots



Haswell Boot CC Section

Erläuterung

Argumente:

Hier sind einige Boot-Argumente gesetzt:

-v - dies aktiviert den ausführlichen Modus, der den gesamten Text hinter den Kulissen anzeigt, der während des Bootvorgangs anstelle des Apple-Logos und des Fortschrittsbalkens vorbeiläuft. Dies ist für jeden Hackintosher von unschätzbarem Wert, da es Ihnen einen Einblick in den Boot-Prozess gibt und Ihnen helfen kann, Probleme, Problem-Kexts usw. zu identifizieren.

- dart=0 - dies ist nur ein zusätzlicher Schutz gegen Vt-d-Probleme.

- debug=0x100 - dies verhindert einen Neustart bei einer Kernel-Panik. Auf diese Weise können Sie (hoffentlich) einige nützliche Informationen sammeln und den Brotkrumen folgen, um die Probleme zu umgehen.

- keepsyms=1 - dies ist eine begleitende Einstellung zu debug=0x100, die dem Betriebssystem sagt, dass es bei einer Kernel-Panik auch die Symbole ausgeben soll. Das kann weitere hilfreiche Einblicke geben, was die Panik selbst verursacht.

DefaultBootVolume und Timeout:

Dies sind die einzigen anderen Einstellungen, die ich in diesem Abschnitt aktualisiert habe.

- DefaultBootVolume - dies verwendet NVRAM, um sich zu merken, welches Volume zuletzt von Clover gebootet wurde, und wählt dieses beim nächsten Booten automatisch aus.

- Timeout - dies ist die Anzahl der Sekunden, bevor das DefaultBootVolume automatisch gebootet wird. Sie können diesen Wert auf -1 setzen, um alle Timeouts zu vermeiden, oder auf 0, um die GUI komplett zu überspringen. Wenn dieser Wert auf 0 gesetzt ist, können Sie beim Booten beliebige Tasten drücken, um die grafische Benutzeroberfläche wieder anzuzeigen, falls Probleme auftreten.

Boot Graphics

Hier ist nichts - nur die Standardeinstellungen. Sie könnten dies anpassen, wenn die Skalierung von Clover geändert werden muss, aber ich pfusche nicht daran herum.

Cpu

Auch hier wird in den meisten Setups, mit denen ich gearbeitet habe, nichts geändert.

Devices

Wir werden hier einige elegante Eigenschaftsinjektionen für WhateverGreen vornehmen und einige grundlegende Audioeinstellungen vornehmen.

Raw XML

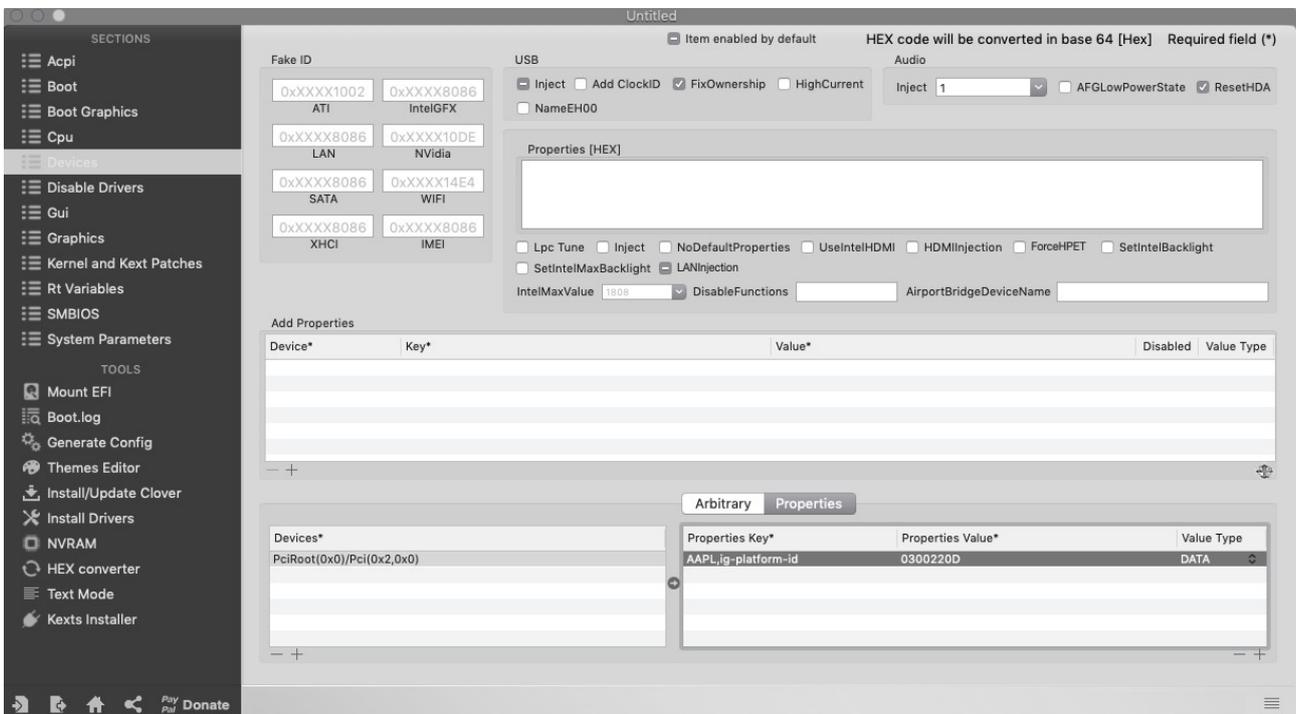
```
<key>Devices</key>
<dict>
  <key>Audio</key>
  <dict>
    <key>Inject</key>
    <integer>1</integer>
    <key>ResetHDA</key>
    <true/>
  </dict>
</dict>
```

```

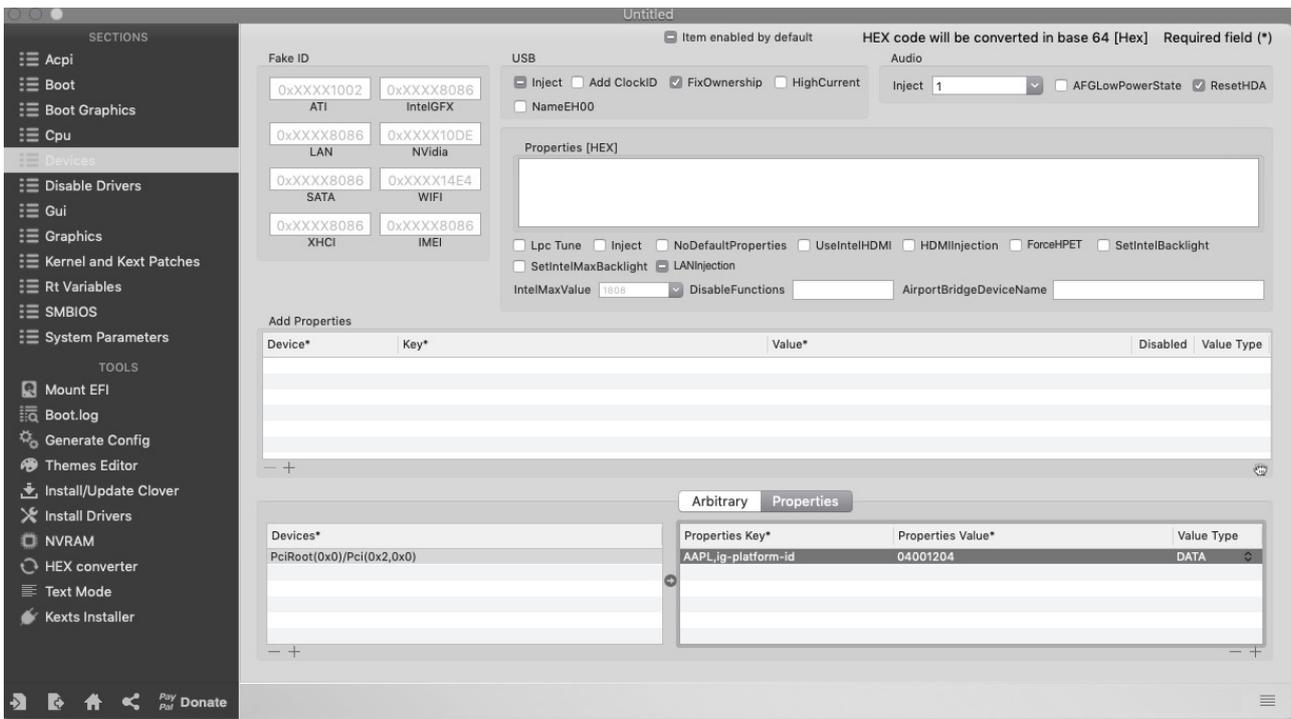
<key>Properties</key>
<dict>
  <key>PciRoot(0x0)/Pci(0x2,0x0)</key>
  <dict>
    <key>AAPL,ig-platform-id</key>
    <data>
      AwAiDQ==
    </data>
  </dict>
</dict>
<key>USB</key>
<dict>
  <key>FixOwnership</key>
  <true/>
</dict>
</dict>

```

Clover Configurator Screenshots



Haswell Devices CC Section - iGPU



Haswell Devices CC Section - iGPU (Connectorless) + dGPU

Properties Key*	Properties Value*	Value Type
device-id	12040000	DATA

Device-Id fake for HD 4400 --> HD 4600

Erläuterung

Fake ID:

Dieser Abschnitt bleibt für unser Beispiel-Setup leer. In der Vergangenheit wurden fast unterstützte iGPUs (wie die HD 4400) hier zu einer unterstützten iGPU gefälscht, aber wir werden den saubereren Eigenschaften-Abschnitt dafür verwenden.

USB:

In diesem Abschnitt stellen wir sicher, dass Inject und FixOwnership ausgewählt sind, um Probleme mit dem Hängenbleiben an einer halb ausgedruckten Zeile irgendwo um die Verbose-Zeile Enabling Legacy Matching zu vermeiden. Sie können das auch umgehen, indem Sie XHCI Hand Off im BIOS aktivieren.

Audio:

Hier haben wir unser Audio auf inject Layout 1 gesetzt - dies kann mit Ihrem Codec kompatibel sein oder auch nicht, aber Sie können auf der [AppleALC's Supported Codec Page](#) nachsehen.

Wir haben auch ResetHDA aktiviert, was den Codec zwischen den Neustarts des Betriebssystems in einen neutralen Zustand versetzt. Dies verhindert einige Probleme mit fehlendem Audio nach dem Booten zu einem anderen Betriebssystem und dann zurück.

Properties:

Dieser Abschnitt wird über [Intel Framebuffer Patching Guide](#) eingerichtet und wendet nur eine tatsächliche Eigenschaft an, nämlich die ig-platform-id. Um den richtigen Wert dafür zu erhalten, müssen wir uns die ig-platform-id ansehen, die wir verwenden wollen, und dann die Paare von Hex-Bytes vertauschen.

Wenn wir uns unsere ig-plat als 0xAABBCCDD vorstellen, würde unsere vertauschte Version wie 0xDDCCBBAA aussehen.

Die beiden ig-platform-id's, die wir verwenden, sind wie folgt:

- 0x0D220003 - dies wird verwendet, wenn die iGPU zur Ansteuerung eines Displays verwendet wird
 - 0300220D, wenn hex-swapped
 - AwAiDQ== wenn die hex-swapped Version in base64 konvertiert wird
- 0x04120004 - dies wird verwendet, wenn die iGPU nur für Rechenaufgaben verwendet wird und kein Display ansteuert
 - 04001204 wenn hex-swapped
 - BAASBA== wenn die hex-swapped Version nach base64 konvertiert wird

Ich habe auch einen weiteren Screenshot hinzugefügt, der eine Geräte-ID-Fälschung für den Fall zeigt, dass Sie eine HD 4400 haben, die in macOS nicht unterstützt wird. Hierfür folgen wir einer ähnlichen Prozedur wie bei unserem obigen ig-platform-id Hex-Swapping - aber dieses Mal arbeiten wir nur mit den ersten beiden Paaren von Hex-Bytes. Wenn wir uns unsere Geräte-ID als 0xAABB0000 vorstellen, würde unsere vertauschte Version wie 0xBBAA0000 aussehen. Wir machen nichts mit den letzten 2 Paaren von Hex-Bytes.

Die gefälschte Geräte-ID ist wie folgt aufgebaut:

- 0x04120000 - das ist die Geräte-ID für die HD 4600, die in macOS unterstützt wird
 - 12040000 bei Hex-Swap
 - EgQAAA== wenn die hex-swapped Version in base64 konvertiert wird

Wenn Sie die rohe xml-Datei verwenden, würden Ihre Eigenschaften wie folgt aussehen (stellen Sie sicher, dass Sie immer noch die passende ig-platform-id für Ihre Einrichtung verwenden):

```
<key>Properties</key>
<dict>
  <key>PciRoot(0x0)/Pci(0x2,0x0)</key>
  <dict>
    <key>device-id</key>
    <data>
      EgQAAA==
    </data>
    <key>AAPL,ig-platform-id</key>
    <data>
      AwAiDQ==
    </data>
  </dict>
</dict>
```

</dict>

Disable Drivers

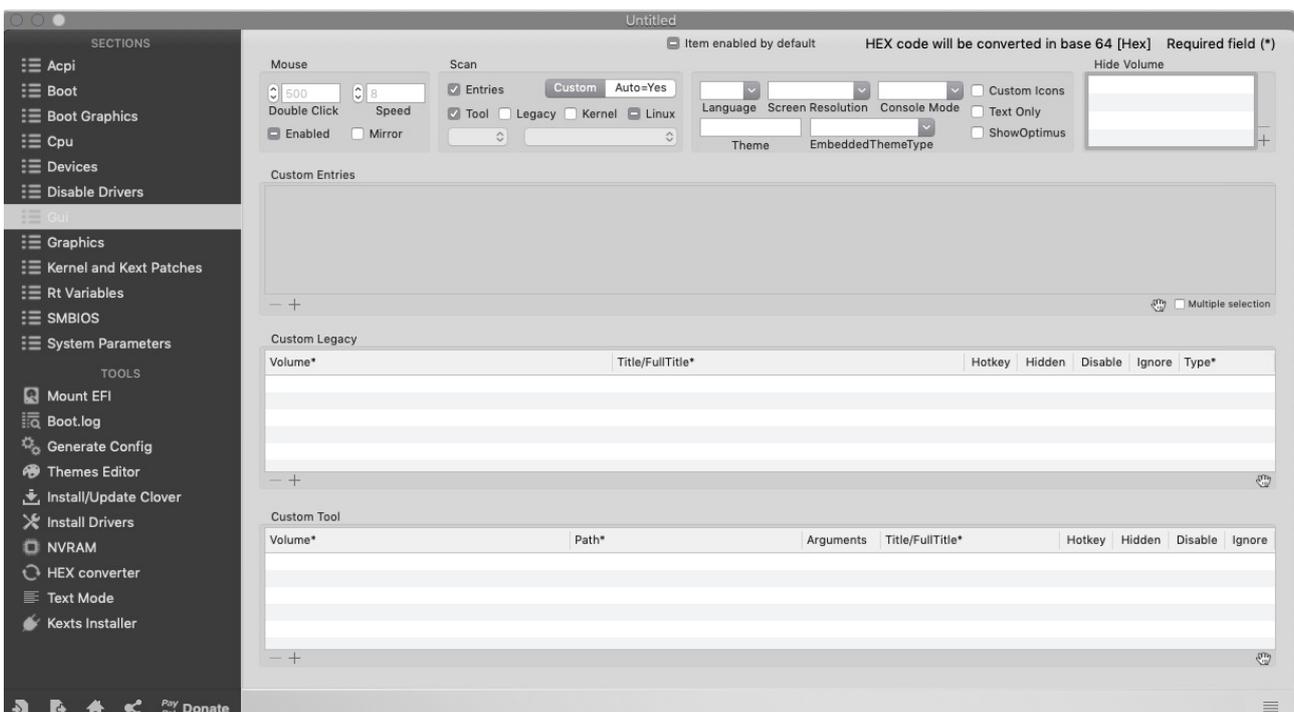
Hier gibt es nichts zu tun.

Gui

Raw XML

```
<key>GUI</key>
<dict>
  <key>Scan</key>
  <dict>
    <key>Entries</key>
    <true/>
    <key>Tool</key>
    <true/>
  </dict>
</dict>
```

Clover Configurator Screenshots



Haswell Gui CC Section

Scan:

Die einzigen Einstellungen, die ich auf dieser Seite geändert habe, sind die Scan-Einstellungen. Ich habe "Benutzerdefiniert" ausgewählt und dann alles außer "Legacy" und "Kernel" markiert. Dies lässt nur einige der nicht bootfähigen Einträge in Clover weg, um das Menü aufzuräumen.

Hide Volumes:

Ich habe hier nichts hinzugefügt, aber Sie können unerwünschte Volumes hier ausblenden. Sie können dies tun, indem Sie entweder den Namen des Volumes oder die UUID hinzufügen.

Um zusätzliche APFS-Einträge auszublenden, fügen Sie Folgendes zu dieser Liste hinzu:

- Preboot
- VM

Um alle Recovery-Partitionen auszublenden, fügen Sie Recovery zu dieser Liste hinzu.

Um die UUID eines zu versteckenden Laufwerks zu erhalten, können Sie den folgenden Terminalbefehl verwenden:

```
diskutil info diskXsY | grep -i "Partition UUID" | rev | cut -d' ' -f 1 | rev
```

Theme:

Wenn Sie ein neues Theme ausprobieren möchten (und ich empfehle Ihnen [clover-next-black](#)), können Sie den entpackten Theme-Ordner zum Verzeichnis /Volumes/EFI/EFI/CLOVER/themes hinzufügen und dann den Namen des Ordners in das Textfeld Theme eingeben, um ihn anzuwenden.

Graphics

In der Vergangenheit haben wir hier die iGPU eingerichtet, aber da wir dies bereits über *Properties* im Abschnitt *Devices* getan haben, müssen wir hier nichts mehr konfigurieren. HINWEIS: Wenn Clover eine Intel iGPU erkennt, aktiviert es automatisch die Intel Injection, wenn der Abschnitt Graphics in der config.plist nicht vorhanden ist. Um dies zu umgehen, können Sie die Injektion explizit deaktivieren, indem Sie das unten stehende Roh-XML verwenden, oder indem Sie in CC einmal auf die Schaltfläche "Inject Intel" klicken, um sie zu aktivieren, und einmal, um sie zu deaktivieren.

Raw XML

```
<key>Graphics</key>
<dict>
  <key>Inject</key>
  <false/>
</dict>
```

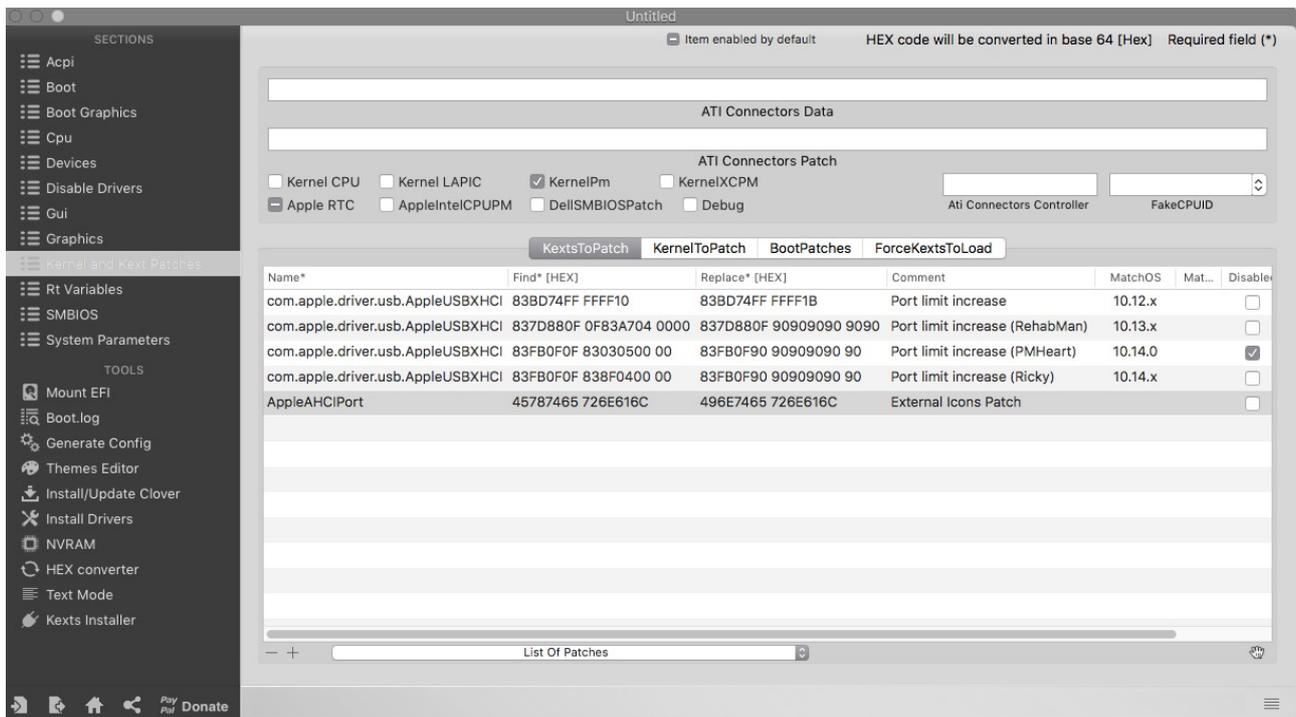
Kernel And Kext Patches

Raw XML

```
<key>KernelAndKextPatches</key>
<dict>
  <key>KernelPm</key>
  <true/>
  <key>KextsToPatch</key>
  <array>
    <dict>
      <key>Comment</key>
      <string>Port limit increase</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
        g710////EA==
      </data>
      <key>InfoPlistPatch</key>
      <false/>
      <key>MatchOS</key>
      <string>10.12.x</string>
      <key>Name</key>
      <string>com.apple.driver.usb.AppleUSBXHCI</string>
      <key>Replace</key>
      <data>
        g710////Gw==
      </data>
    </dict>
    <dict>
      <key>Comment</key>
      <string>Port limit increase (RehabMan)</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
        g32IDw+DpwQAAA==
      </data>
      <key>InfoPlistPatch</key>
      <false/>
      <key>MatchOS</key>
      <string>10.13.x</string>
      <key>Name</key>
      <string>com.apple.driver.usb.AppleUSBXHCI</string>
      <key>Replace</key>
      <data>
        g32ID5CQkJCQkA==
      </data>
    </dict>
    <dict>
      <key>Comment</key>
      <string>Port limit increase (PMHeart)</string>
      <key>Disabled</key>
      <true/>
      <key>Find</key>
      <data>
        g/SPD4MDBQAA
      </data>
      <key>InfoPlistPatch</key>
      <false/>
      <key>MatchOS</key>
      <string>10.14.0</string>
      <key>Name</key>
      <string>com.apple.driver.usb.AppleUSBXHCI</string>
    </dict>
  </array>
</dict>
```

```
    <key>Replace</key>
    <data>
    g/sPkJCQkJCQ
    </data>
  </dict>
  <dict>
    <key>Comment</key>
    <string>Port limit increase (Ricky)</string>
    <key>Disabled</key>
    <false/>
    <key>Find</key>
    <data>
    g/sPD40PBAAA
    </data>
    <key>InfoPlistPatch</key>
    <false/>
    <key>MatchOS</key>
    <string>10.14.x</string>
    <key>Name</key>
    <string>com.apple.driver.usb.AppleUSBXHCI</string>
    <key>Replace</key>
    <data>
    g/sPkJCQkJCQ
    </data>
  </dict>
  <dict>
    <key>Comment</key>
    <string>External Icons Patch</string>
    <key>Disabled</key>
    <false/>
    <key>Find</key>
    <data>
    RXh0ZXJuYWw=
    </data>
    <key>InfoPlistPatch</key>
    <false/>
    <key>Name</key>
    <string>AppleAHCIPort</string>
    <key>Replace</key>
    <data>
    SW50ZXJuYWw=
    </data>
  </dict>
</array>
</dict>
```

Clover Configurator Screenshots



Haswell KernelAndKextPatches CC Section

Explanation

In diesem Abschnitt haben wir ein paar Einstellungen aktiviert und einige Kext-Patches hinzugefügt

Checkboxes:

Wir haben hier ein paar Kontrollkästchen aktiviert:

- Apple RTC - dies stellt sicher, dass wir keinen BIOS-Reset beim Neustart haben.
- KernelPM - diese Einstellung verhindert das Schreiben auf MSR 0xe2, was eine Kernel-Panik beim Booten verhindern kann.

KextsToPatch:

Wir haben hier 4 verschiedene Kexts zum Patchen hinzugefügt. Drei davon sind für USB-Port-Limit-Erhöhungen und der letzte dient als Fix für orangefarbene Icons - wenn interne Laufwerke hotplug-fähig sind und als externe Laufwerke behandelt werden.

Sie werden feststellen, dass für jedes der USB-Port-Limit-Patches MatchOS-Werte eingestellt sind. Sie können alle Einträge für Betriebssystemversionen, die Sie nicht ausführen möchten, entfernen. Sie schaden nicht, wenn sie da sind, aber wenn Sie eine saubere, minimale Liste wollen, ist es nicht sinnvoll, sie zu haben.

RtVariables And SMBIOS

Raw XML

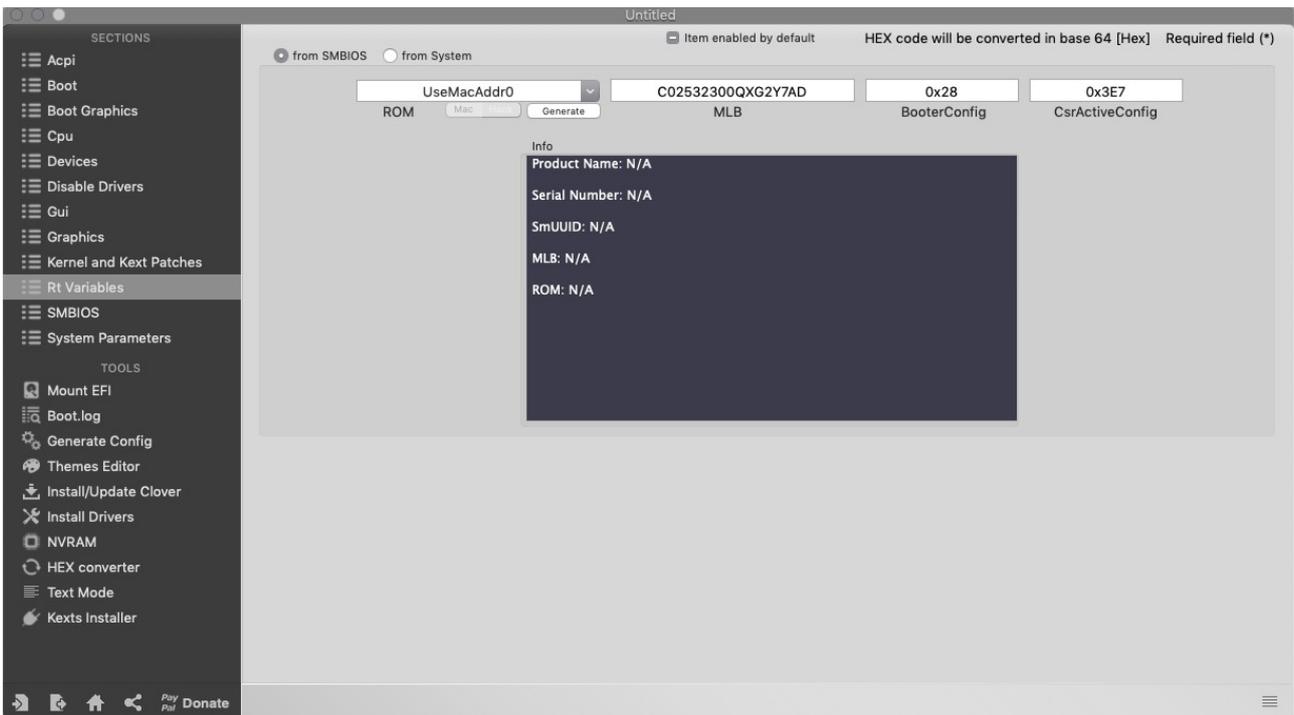
```
<key>RtVariables</key>
<dict>
  <key>BooterConfig</key>
  <string>0x28</string>
  <key>CsrActiveConfig</key>
  <string>0x3E7</string>
  <key>MLB</key>
  <string>C02532300QXG2Y7AD</string>
  <key>ROM</key>
```

```

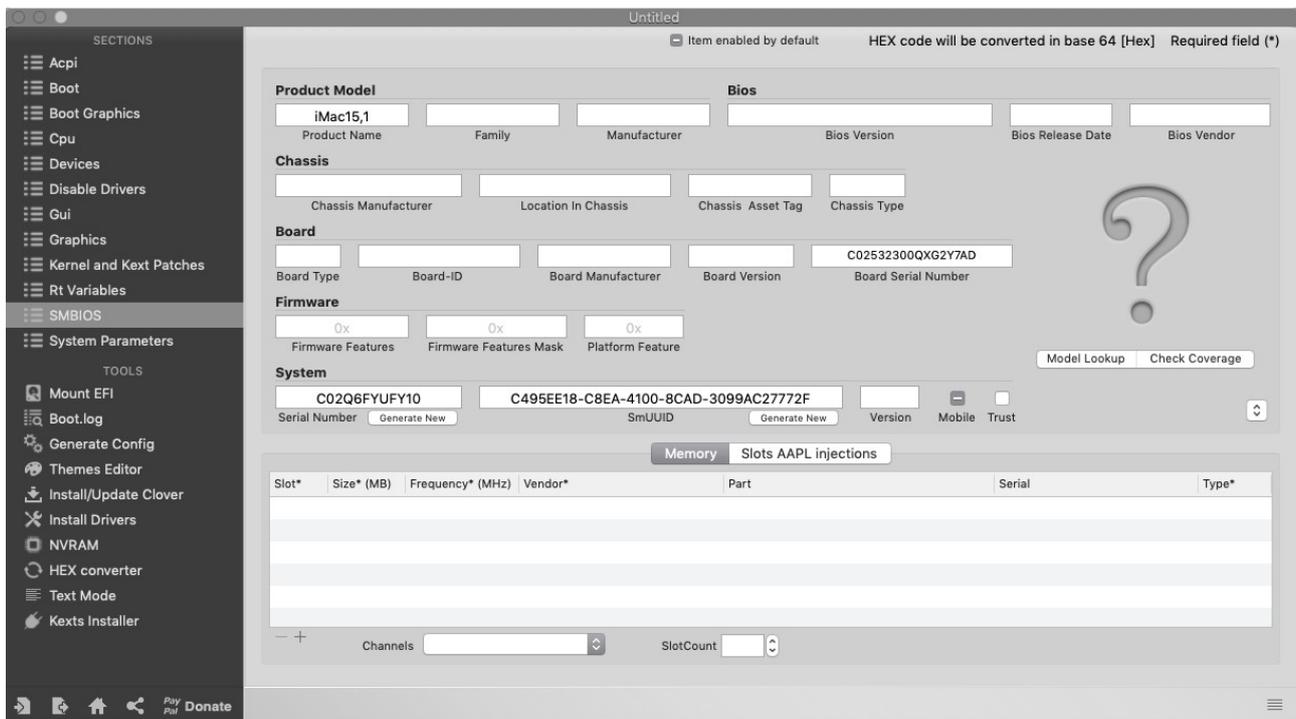
    <string>UseMacAddr0</string>
  </dict>
  <key>SMBIOS</key>
  <dict>
    <key>BoardSerialNumber</key>
    <string>C02532300QXG2Y7AD</string>
    <key>ProductName</key>
    <string>iMac15,1</string>
    <key>SerialNumber</key>
    <string>C02Q6FYUFY10</string>
    <key>SmUUID</key>
    <string>C495EE18-C8EA-4100-8CAD-3099AC27772F</string>
  </dict>

```

Clover Configurator Screenshots



Haswell Rt Variables CC Section



Haswell SMBIOS CC Section

Erläuterung

Zum Einrichten der SMBIOS-Informationen verwende ich die [macserial](#) von acidanthera. Ich habe ein [python script](#) geschrieben, das es ebenfalls nutzen kann (und automatisch in der config.plist speichert, wenn es ausgewählt wird). Es gibt viele Informationen, die leer gelassen werden, damit Clover die Leerstellen ausfüllen kann; das bedeutet, dass eine Aktualisierung von Clover die übergebenen Informationen aktualisiert und Sie nicht auch Ihre config.plist aktualisieren müssen.

Für dieses Haswell-Beispiel habe ich das iMac15,1 SMBIOS gewählt. Die typische Aufteilung ist wie folgt:

- Haswell mit nur iGPU - iMac14,1
- Haswell mit dGPU - iMac14,2
- Haswell Auffrischung - iMac15,1

Um die SMBIOS-Informationen zu erhalten, die mit macserial generiert wurden, können Sie es mit dem Argument -a ausführen (das Seriennummern und Board-Seriennummern für alle unterstützten Plattformen generiert). Sie können es auch mit grep parsen, um Ihre Suche auf einen SMBIOS-Typ zu beschränken.

Bei unserem iMac15,1-Beispiel würden wir macserial wie folgt über das Terminal ausführen:

```
macserial -a | grep -i iMac15,1
```

Dies würde eine Ausgabe ähnlich der folgenden ergeben:

Product	Serial	Board Serial (MLB)
iMac15,1	C02NFZZYFY10	C02438207QXG2Y7FB
iMac15,1	C02P32YJFY10	C02502303GUG2Y78C
iMac15,1	C02P2VZ7FY10	C02501306QXG2Y7AD
iMac15,1	C02NM0EDFY10	C02444701CDG2Y71H
iMac15,1	C02NVHZCFY10	C02451303CDG2Y7JA
iMac15,1	C02QLRZ4FY10	C02543300GUG2Y7JC
iMac15,1	C02QJ0UPFY10	C02541902GUG2Y7JA
iMac15,1	C02QG0NGFY10	C02539700J9G2Y71M
iMac15,1	C02N3XYEFY10	C02429104J9G2Y7UE
iMac15,1	C02QW0M3FY10	C02552700GUG2Y7JA

Die Reihenfolge ist Produkt | Serial | Board Serial (MLB)

Der iMac15,1-Teil wird nach SMBIOS -> Produktname kopiert.

Der serielle Teil wird nach SMBIOS -> Seriennummer kopiert.

Der Teil "Board Serial" wird in SMBIOS -> Board Serial Number sowie in Rt Variables -> MLB kopiert.

Wir können eine SmUUID erstellen, indem wir uuidgen im Terminal ausführen (oder sie wird automatisch über mein GenSMBIOS-Skript generiert) - und diese wird nach SMBIOS -> SmUUID kopiert.

Wir setzen Rt Variables -> ROM auf UseMacAddr0, was einfach unsere Onboard-Mac-Adresse verwendet - diese sollte eindeutig genug sein, um nicht mit anderen in Konflikt zu geraten.

BooterConfig wird auf 0x28 gesetzt, und CsrActiveConfig wird auf 0x3e7 gesetzt, was SIP effektiv deaktiviert. Sie können eine Reihe von anderen Optionen wählen, um Abschnitte von SIP zu aktivieren/deaktivieren. Einige gängige sind wie folgt:

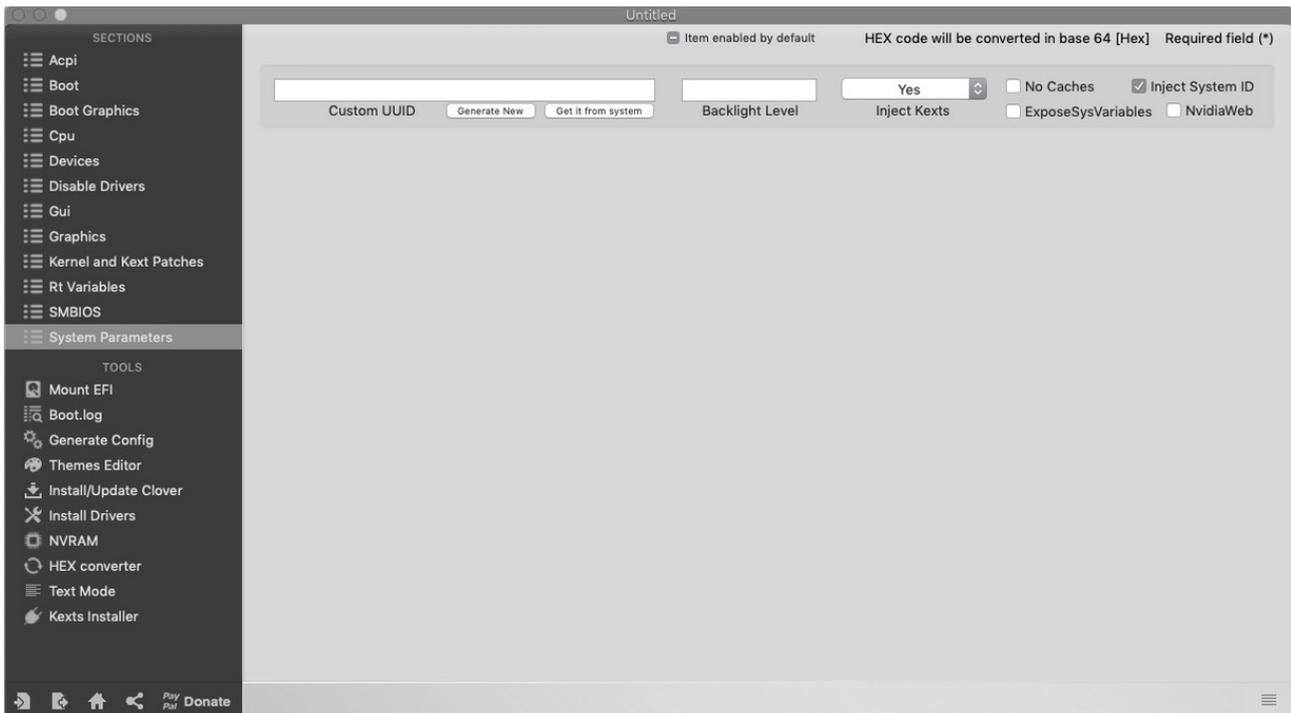
- 0x0 - SIP vollständig aktiviert
- 0x3 - Erlaubt unsignierte Kexts und das Schreiben auf geschützte fs-Speicherplätze
- 0x3e7 - SIP komplett deaktiviert

System Parameters

Raw XML

```
<key>SystemParameters</key>
<dict>
  <key>InjectKexts</key>
  <string>Yes</string>
  <key>InjectSystemID</key>
  <true/>
</dict>
```

Clover Configurator Screenshots



Haswell System Parameters CC Section

Explanation

Inject Kexts:

Diese Einstellung hat 3 Modi:

- Ja - damit wird Clover angewiesen, Kexts aus der EFI unabhängig davon zu injizieren.
- Nein - damit wird Clover angewiesen, keine Kexts von der EFI zu injizieren.
- Detect - Clover injiziert Kexts nur, wenn FakeSMC.kext nicht im Kext-Cache ist.

Wir setzen es auf Yes, um sicherzustellen, dass alle Kexts, die wir zuvor hinzugefügt haben, richtig injiziert werden.

InjectSystemID

Diese Einstellung weist Clover an, die SmUUID als System-ID beim Booten zu setzen - was für iMessage und so wichtig ist.

Saving

Diese Einstellung weist Clover an, die SmUUID als System-ID beim Booten zu setzen - was wichtig für iMessage und dergleichen ist. An diesem Punkt können Sie File -> Save machen, um die config.plist zu speichern. Wenn Sie Probleme haben, die Datei direkt im

EFI zu speichern, können Sie sie auch auf dem Desktop speichern und dann einfach rüberkopieren. Ich lasse die [sample config.plist here](#)

Skylake

Wir gehen die Abschnitte der config.plist durch, einen nach dem anderen für ein Skylake-Desktop-Setup.

Ausgangspunkte

Ich beginne gerne entweder mit der config.plist, die Clover Ihnen zur Verfügung stellt, oder mit einer leeren Leinwand. In den nächsten Beispielen zeige ich Ihnen, wie ich die Dinge von Grund auf neu einrichte; wenn Sie von woanders anfangen, haben Sie vielleicht mehr Dinge überprüft/eingestellt als ich - aber Sie werden dem folgen wollen, was ich mache. Ich werde auch die rohen xml-Beispiele mit einbeziehen, um denjenigen zu zeigen, die mit einem Texteditor arbeiten (wie ich es bevorzuge).

ACPI

Die Standardeinstellungen von Clover sind ziemlich übertrieben und können einige Probleme verursachen. Wir werden diesen Abschnitt ziemlich minimal halten, und ich werde auch ein wenig darauf eingehen, warum wir das für jeden Teil tun.

Raw XML

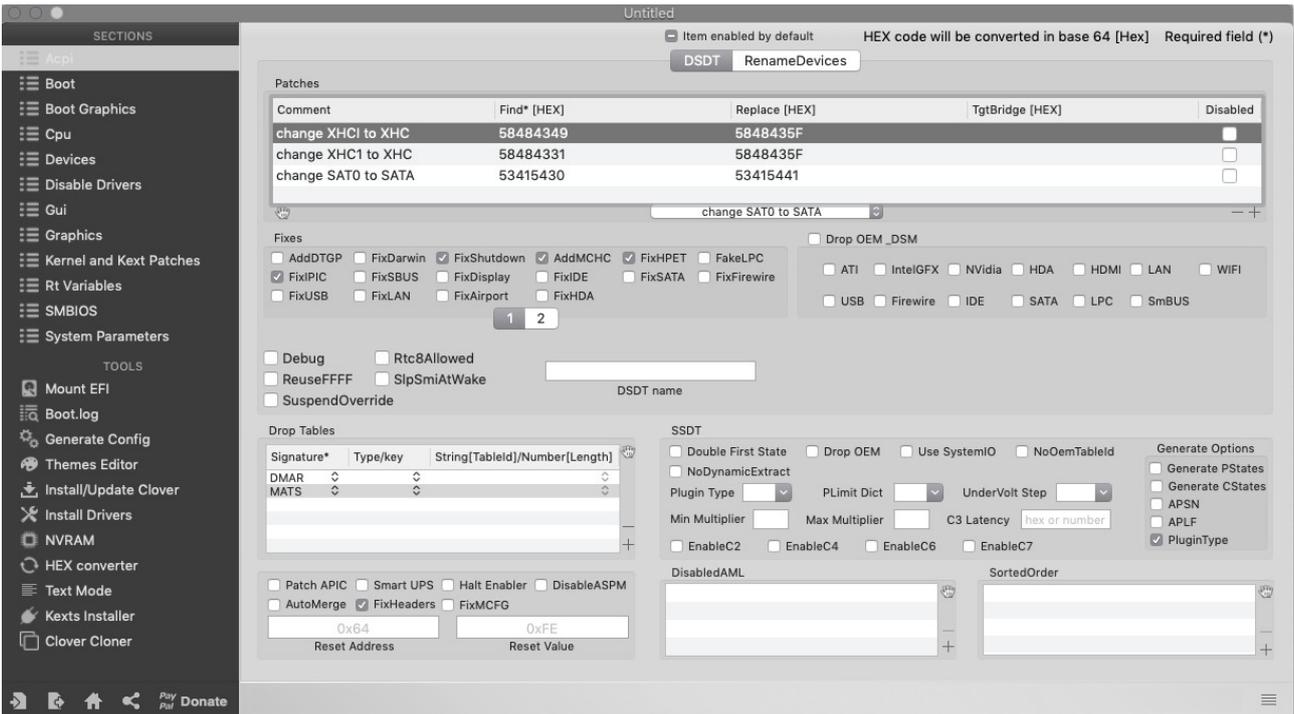
```
<key>ACPI</key>
<dict>
  <key>DSDT</key>
  <dict>
    <key>Fixes</key>
    <dict>
      <key>AddMCHC</key>
      <true/>
      <key>FixHPET</key>
      <true/>
      <key>FixIPIC</key>
      <true/>
      <key>FixRTC</key>
      <true/>
      <key>FixShutdown</key>
      <true/>
      <key>FixTMR</key>
      <true/>
    </dict>
  <key>Patches</key>
  <array>
    <dict>
      <key>Comment</key>
      <string>change XHCI to XHC</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
        WEhDSQ==
      </data>
    </dict>
  </array>
</dict>
```

```

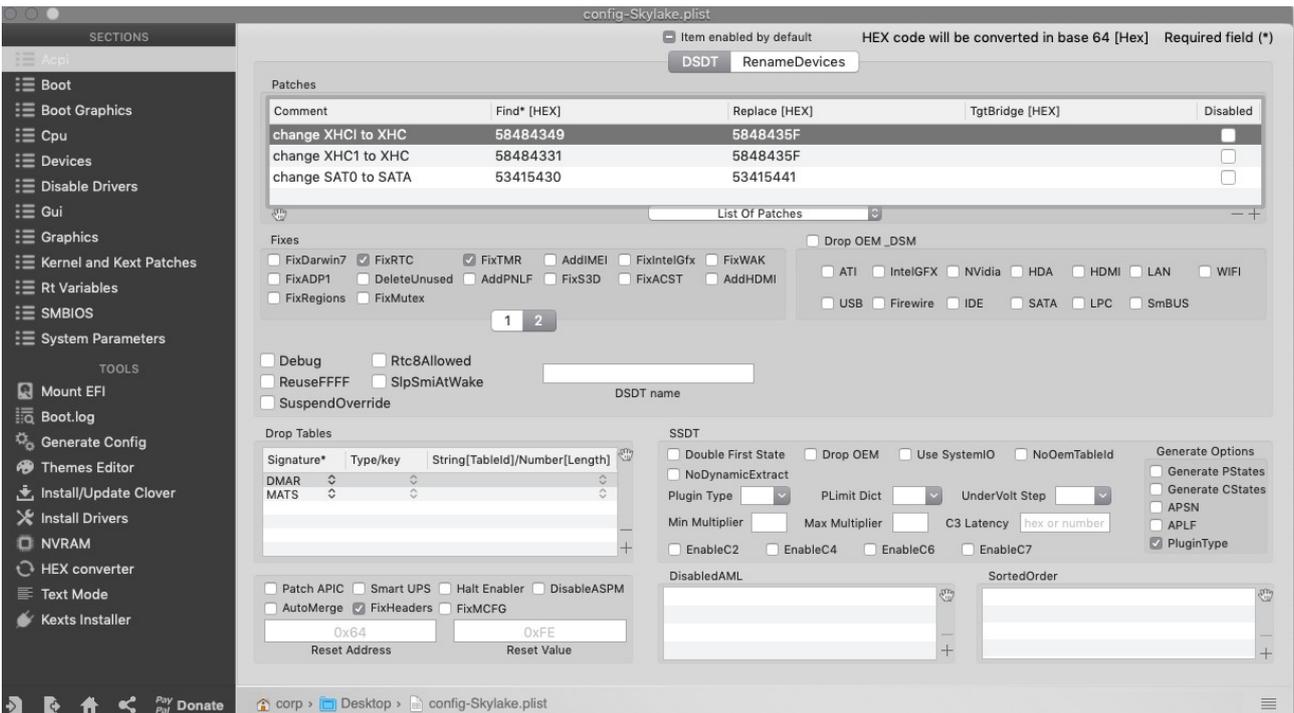
        </data>
        <key>Replace</key>
        <data>
        WEhDXw==
        </data>
    </dict>
    <dict>
        <key>Comment</key>
        <string>change XHC1 to XHC</string>
        <key>Disabled</key>
        <false/>
        <key>Find</key>
        <data>
        WEhDMQ==
        </data>
        <key>Replace</key>
        <data>
        WEhDXw==
        </data>
    </dict>
    <dict>
        <key>Comment</key>
        <string>change SAT0 to SATA</string>
        <key>Disabled</key>
        <false/>
        <key>Find</key>
        <data>
        U0FUQA==
        </data>
        <key>Replace</key>
        <data>
        U0FUQQ==
        </data>
    </dict>
</array>
</dict>
<key>DropTables</key>
<array>
    <dict>
        <key>Signature</key>
        <string>DMAR</string>
    </dict>
    <dict>
        <key>Signature</key>
        <string>MATS</string>
    </dict>
</array>
<key>FixHeaders</key>
<true/>
<key>SSDT</key>
<dict>
    <key>Generate</key>
    <dict>
        <key>PluginType</key>
        <true/>
    </dict>
</dict>
</dict>

```

Clover Configurator Screenshots



Skylake Acpi CC Section 1



Skylake Acpi CC Section 2

Erläuterung

Patches:

Das erste, was wir durchgehen, ist der Abschnitt Patches. Dieser Abschnitt ermöglicht es uns, Teile des DSDT über Clover dynamisch umzubenennen. Da wir keinen echten Mac verwenden und macOS ziemlich genau darauf achtet, wie Dinge benannt werden, können wir nicht-destruktive Änderungen vornehmen, um die Dinge Mac-freundlich zu halten. Wir haben hier drei Einträge:

- XHCI in XHC ändern - hilft, einen Konflikt mit eingebauten USB-Injektoren zu vermeiden
- XHC1 in XHC ändern - hilft, einen Konflikt mit den eingebauten USB-Injektoren zu vermeiden
- SAT0 zu SATA ändern - für mögliche SATA-Kompatibilität

Fixes:

Wenn wir uns dann den Abschnitt Fixes ansehen, werden wir sehen, dass wir ein paar Dinge überprüft haben (es gibt 2 Seiten, daher habe ich 2 Screenshots eingefügt):

- FixShutdown - dies kann bei einigen Boards helfen, die es vorziehen, neu zu starten, anstatt herunterzufahren. Manchmal kann es bei anderen Boards zu Problemen beim Herunterfahren führen (ironisch, nicht wahr?), wenn Sie also Probleme beim Herunterfahren haben, wenn dies aktiviert ist, sollten Sie es deaktivieren.
- Die übrigen Korrekturen helfen, IRQ-Konflikte usw. zu vermeiden, und sind nicht dafür bekannt, dass sie Probleme verursachen. Sie sind vielleicht nicht für jede Hardware notwendig, haben aber keine negativen Auswirkungen, wenn sie angewendet werden.

Drop Tables:

Wir haben DSDT in unserem Abschnitt über Patches schon leicht angerissen - und dies ist eine kleine Erweiterung davon. SSDT ist wie ein Unterabschnitt von DSDT. Der Abschnitt "Drop Tables" erlaubt es uns, bestimmte SSDT-Tabellen beim Laden auszulassen (wie ich bereits erwähnt habe, ist DSDT für Mac und PC unterschiedlich, und macOS kann ziemlich pingelig sein). Die beiden, die ich hinzugefügt habe, sind wie folgt:

- DMAR - dies verhindert einige Probleme mit Vt-d; das ist PCI-Passthrough für VMs, und nicht sehr funktionell (wenn überhaupt?) auf Hackintoshes.
- MATS - ab High Sierra wird diese Tabelle geparkt und kann manchmal nicht druckbare Zeichen enthalten, die zu einer Kernel-Panik führen können.

FixHeaders und PluginType:

Die einzigen anderen Dinge, die wir auf dieser Seite getan haben, sind diese beiden Kontrollkästchen zu aktivieren.

- FixHeaders - dies ist nur eine Verdoppelung unseres MATS-Tabellenabwurfs. Dieses Kontrollkästchen weist Clover an, Header zu bereinigen, um Kernel-Paniks im Zusammenhang mit nicht druckbaren Zeichen zu vermeiden.
- PluginType - dies injiziert einige DSDT-Daten, um X86PlatformPlugin zu laden - was uns einen Vorsprung bei der nativen CPU-Leistungsverwaltung verschafft. Diese Einstellung funktioniert allerdings nur auf Haswell und neueren CPUs.

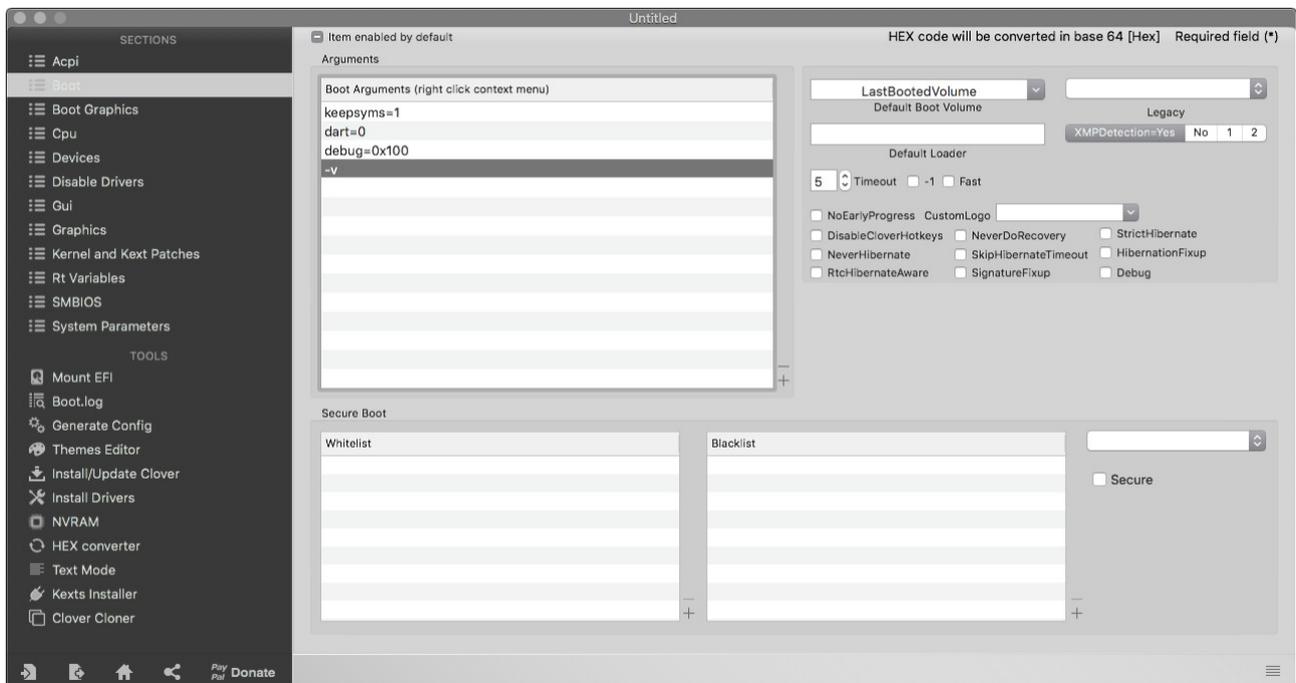
Boot

Wir müssen hier nicht viel tun, aber wir werden ein paar Dinge optimieren.

Raw XML

```
<key>Boot</key>
<dict>
  <key>Arguments</key>
  <string>keepsyms=1 dart=0 debug=0x100 -v</string>
  <key>DefaultVolume</key>
  <string>LastBootedVolume</string>
  <key>Timeout</key>
  <integer>5</integer>
</dict>
```

Clover Configurator Screenshots



Skylake Boot CC Section

Erläuterung

Argumente:

Hier sind einige Boot-Argumente gesetzt:

-v - dies aktiviert den ausführlichen Modus, der den gesamten Text hinter den Kulissen anzeigt, der während des Bootvorgangs anstelle des Apple-Logos und des Fortschrittsbalkens vorbeiläuft. Dies ist für jeden Hackintosher von unschätzbarem Wert, da es Ihnen einen Einblick in den Boot-Prozess gibt und Ihnen helfen kann, Probleme, Problem-Kexts usw. zu identifizieren.

- dart=0 - dies ist nur ein zusätzlicher Schutz gegen Vt-d-Probleme.

- debug=0x100 - dies verhindert einen Neustart bei einer Kernel-Panik. Auf diese Weise können Sie (hoffentlich) einige nützliche Informationen sammeln und den Brotkrumen folgen, um die Probleme zu umgehen.
- keepsyms=1 - dies ist eine begleitende Einstellung zu debug=0x100, die dem Betriebssystem sagt, dass es bei einer Kernel-Panik auch die Symbole ausgeben soll. Das kann einige hilfreichere Einblicke geben, was die Panik selbst verursacht.

DefaultBootVolume und Timeout:

Dies sind die einzigen anderen Einstellungen, die ich in diesem Abschnitt aktualisiert habe.

- DefaultBootVolume - dies verwendet NVRAM, um sich zu merken, welches Volume zuletzt von Clover gebootet wurde, und wählt dieses beim nächsten Booten automatisch aus.
- Timeout - dies ist die Anzahl der Sekunden, bevor das DefaultBootVolume automatisch gebootet wird. Sie können diesen Wert auf -1 setzen, um alle Timeouts zu vermeiden, oder auf 0, um die GUI komplett zu überspringen. Wenn Sie diesen Wert auf 0 setzen, können Sie beim Booten beliebige Tasten drücken, um die grafische Benutzeroberfläche wieder anzuzeigen, falls Probleme auftreten.

Boot-Graphics

Hier ist nichts zu tun - nur die Standardeinstellungen. Sie könnten dies anpassen, wenn die Skalierung von Clover geändert werden muss, aber ich pfusche nicht daran herum.

Cpu

Auch hier wird in den meisten Setups, mit denen ich gearbeitet habe, nichts geändert.

Devices

Hier werden wir einige Eigenschaften für WhateverGreen einfügen und einige grundlegende Audioeinstellungen vornehmen.

Raw XML

```

<key>Devices</key>
<dict>
  <key>Audio</key>
  <dict>
    <key>Inject</key>
    <integer>1</integer>
    <key>ResetHDA</key>
    <true/>
  </dict>
  <key>Properties</key>
  <dict>
    <key>PciRoot(0x0)/Pci(0x2,0x0)</key>
    <dict>
      <key>AAPL,ig-platform-id</key>
      <data>
        AAASGQ==
      </data>
    </dict>
  </dict>
</dict>

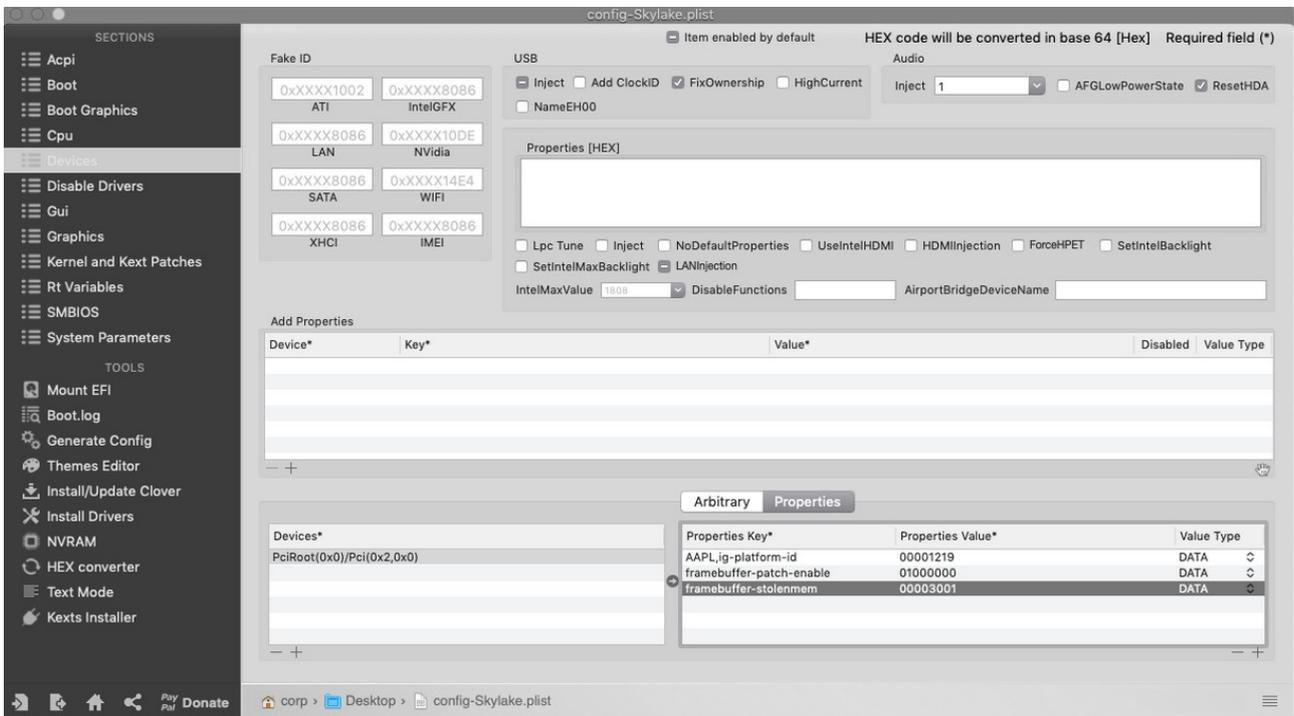
```

```

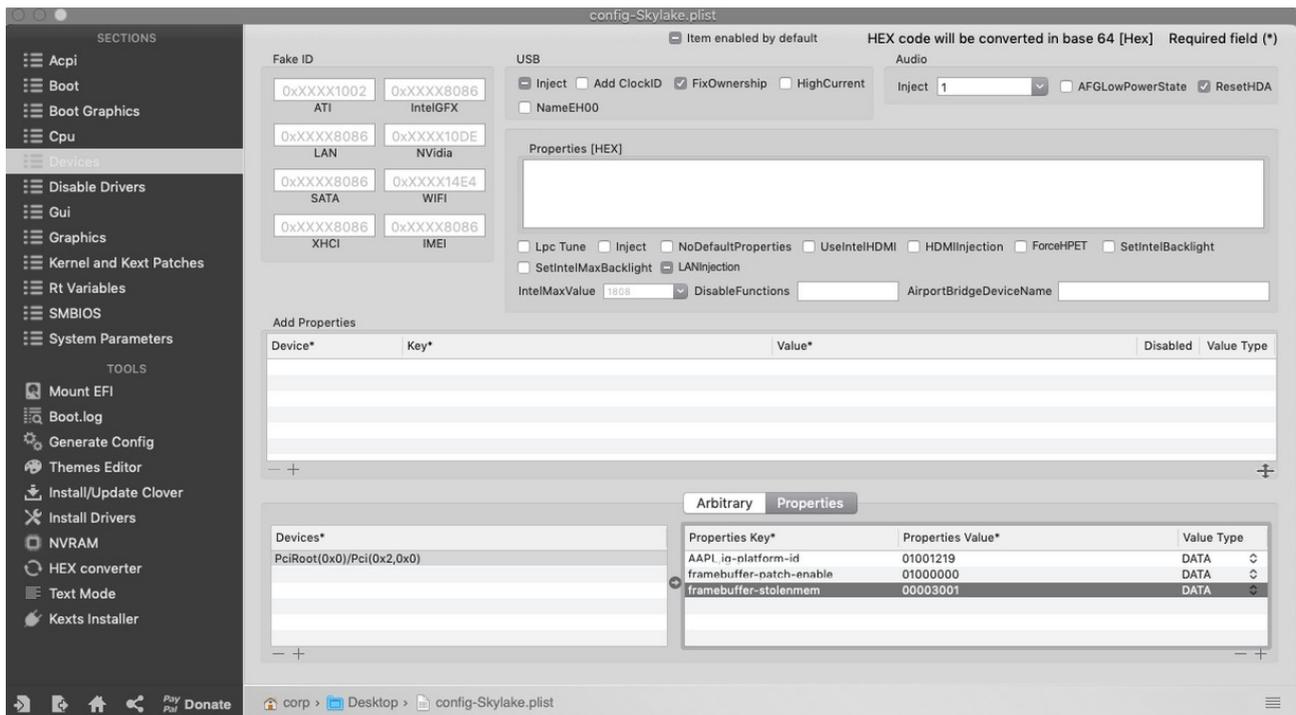
<key>framebuffer-patch-enable</key>
<data>
AQAAAA==
</data>
<key>framebuffer-stolenmem</key>
<data>
AAAwAQ==
</data>
</dict>
</dict>
<key>USB</key>
<dict>
<key>FixOwnership</key>
<true/>
</dict>
</dict>

```

Clover Configurator Screenshots



Skylake Devices CC Section - iGPU



Skylake Devices CC Section - iGPU Connectorless

Erläuterung

Gefälschte ID:

Dieser Abschnitt bleibt für unser Beispiel-Setup leer. In der Vergangenheit wurden fast unterstützte iGPUs (wie die HD 4400) hier zu einer unterstützten iGPU gefälscht, aber wir werden den Abschnitt "Clean Properties" dafür verwenden.

USB:

In diesem Abschnitt stellen wir sicher, dass Inject und FixOwnership ausgewählt sind, um Probleme mit dem Hängenbleiben an einer halb ausgedruckten Zeile irgendwo in der Nähe der Zeile Enabling Legacy Matching verbose zu vermeiden. Sie können das auch umgehen, indem Sie XHCI Hand Off im BIOS aktivieren.

Audio:

Hier haben wir unser Audio auf inject Layout 1 gesetzt - dies kann mit Ihrem Codec kompatibel sein oder auch nicht, aber Sie können auf der [AppleALC's Supported Codec Page](#) für unterstützte Codecs nachsehen.

Wir haben auch ResetHDA aktiviert, was den Codec zwischen den Neustarts des Betriebssystems in einen neutralen Zustand versetzt. Dies verhindert einige Probleme mit fehlendem Audio nach dem Booten zu einem anderen Betriebssystem und dann zurück.

Properties:

Dieser Abschnitt wird über Headkaze's [Intel Framebuffer Patching Guide](#) eingerichtet und wendet zu Beginn nur eine tatsächliche Eigenschaft an, nämlich die ig-platform-id. Um den richtigen Wert dafür zu erhalten, müssen wir uns die ig-platform-id ansehen, die wir verwenden wollen, und dann die Paare von Hex-Bytes vertauschen.

Wenn wir uns unsere ig-platform-id als 0xAABBCCDD vorstellen, würde unsere vertauschte Version wie 0xDDCCBBAA aussehen.

Die beiden ig-platform-id's, die wir verwenden, sind wie folgt:

- 0x19120000 - dies wird verwendet, wenn die iGPU zur Ansteuerung eines Displays verwendet wird

- 00001219 wenn hex-swapped
- AAASGQ== wenn die hex-swapped Version in base64 konvertiert wird

- 0x19120001 - dies wird verwendet, wenn die iGPU nur für Rechenaufgaben verwendet wird und kein Display ansteuert

- 01001219 wenn hex-swapped
- AQASGQ== wenn die hex-swapped Version in base64 konvertiert wird

Wir fügen auch 2 weitere Eigenschaften hinzu, framebuffer-patch-enable und framebuffer-stolenmem. Die erste aktiviert das Patchen über WhateverGreen.kext, und die zweite setzt den minimalen gestohlenen Speicher auf 19 MB.

Disable Drivers

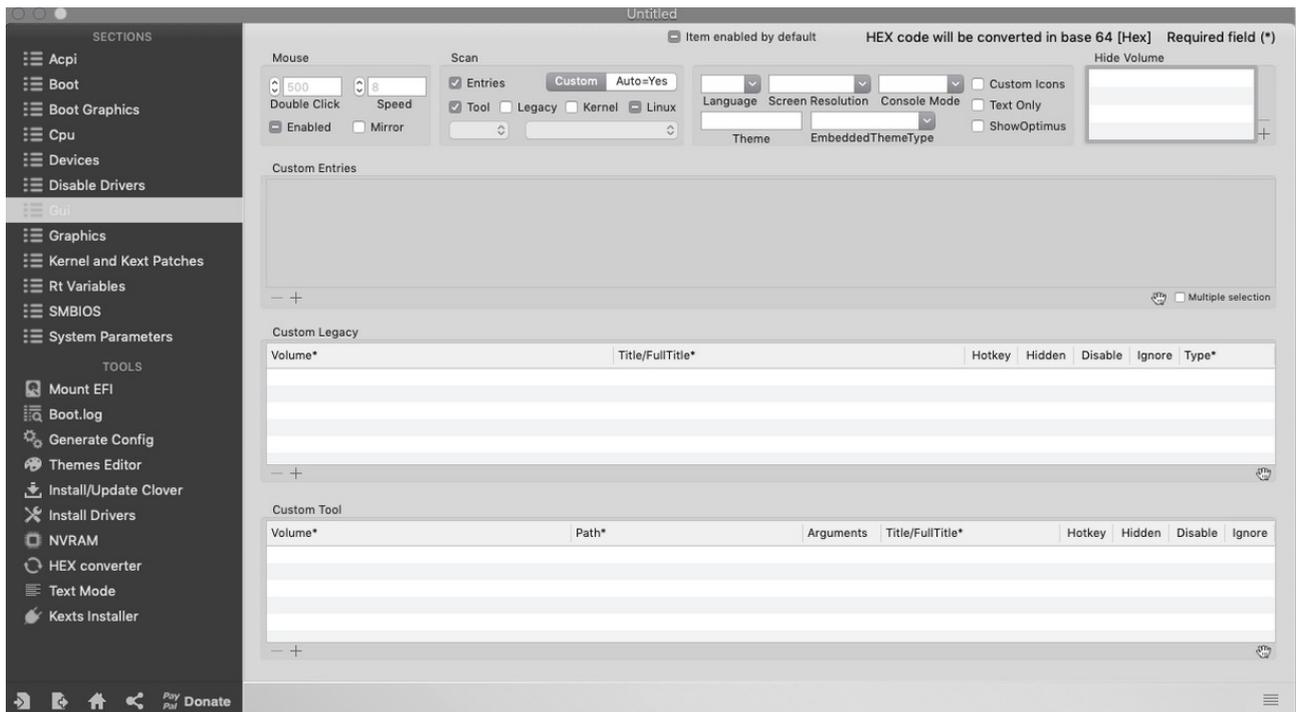
Hier gibt es nichts zu tun.

Gui

Raw XML

```
<key>GUI</key>
<dict>
  <key>Scan</key>
  <dict>
    <key>Entries</key>
    <true/>
    <key>Tool</key>
    <true/>
  </dict>
</dict>
```

Clover Configurator Screenshots



Skylake Gui CC Section

Erläuterung

Scannen:

Die einzigen Einstellungen, die ich auf dieser Seite geändert habe, sind die Scan-Einstellungen. Ich habe "Benutzerdefiniert" ausgewählt und dann alles außer "Legacy" und "Kernel" angekreuzt. Damit werden nur einige der nicht bootfähigen Einträge in Clover weggelassen, um das Menü aufzuräumen.

Volumes ausblenden:

Ich habe hier nichts hinzugefügt, aber Sie können unerwünschte Volumes hier ausblenden. Sie können dies tun, indem Sie entweder den Namen des Volumes oder die UUID hinzufügen.

Um zusätzliche APFS-Einträge auszublenden, fügen Sie Folgendes zu dieser Liste hinzu:

- Preboot
- VM

Um alle Recovery-Partitionen auszublenden, fügen Sie Recovery zu dieser Liste hinzu.

Um die UUID eines zu versteckenden Laufwerks zu erhalten, können Sie den folgenden Terminalbefehl verwenden:

```
diskutil info diskXsY | grep -i "Partition UUID" | rev | cut -d' ' -f 1 | rev
```

Theme:

Wenn Sie ein neues Thema ausprobieren möchten (und ich empfehle Ihnen, sich clover-next-black anzusehen), können Sie den entpackten Themenordner zum Verzeichnis /Volumes/EFI/EFI/CLOVER/themes hinzufügen und dann den Namen des Ordners in das Textfeld Thema eingeben, um ihn anzuwenden.

Graphics

In der Vergangenheit haben wir hier die iGPU eingerichtet, aber da wir dies bereits über Eigenschaften im Abschnitt Geräte getan haben, müssen wir hier nichts mehr konfigurieren. HINWEIS: Wenn Clover eine Intel iGPU erkennt, aktiviert es automatisch Intel Injection, wenn der Abschnitt Graphics in der config.plist nicht vorhanden ist. Um dies zu umgehen, können Sie die Injektion explizit deaktivieren, indem Sie das unten stehende Roh-XML verwenden, oder indem Sie die Schaltfläche "Inject Intel" einmal anklicken, um sie zu aktivieren, und einmal, um sie in CC zu deaktivieren.

Raw XML

```
<key>Graphics</key>
<dict>
  <key>Inject</key>
  <false/>
</dict>
```

Kernel And Kext Patches

Raw XML

```
<key>KernelAndKextPatches</key>
<dict>
  <key>KernelPm</key>
  <true/>
  <key>KextsToPatch</key>
  <array>
    <dict>
      <key>Comment</key>
      <string>Port limit increase</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
        g710///EA==
      </data>
      <key>InfoPlistPatch</key>
      <false/>
      <key>MatchOS</key>
      <string>10.12.x</string>
      <key>Name</key>
      <string>com.apple.driver.usb.AppleUSBXHCI</string>
      <key>Replace</key>
      <data>
        g710///Gw==
      </data>
    </dict>
  </array>
</dict>
```

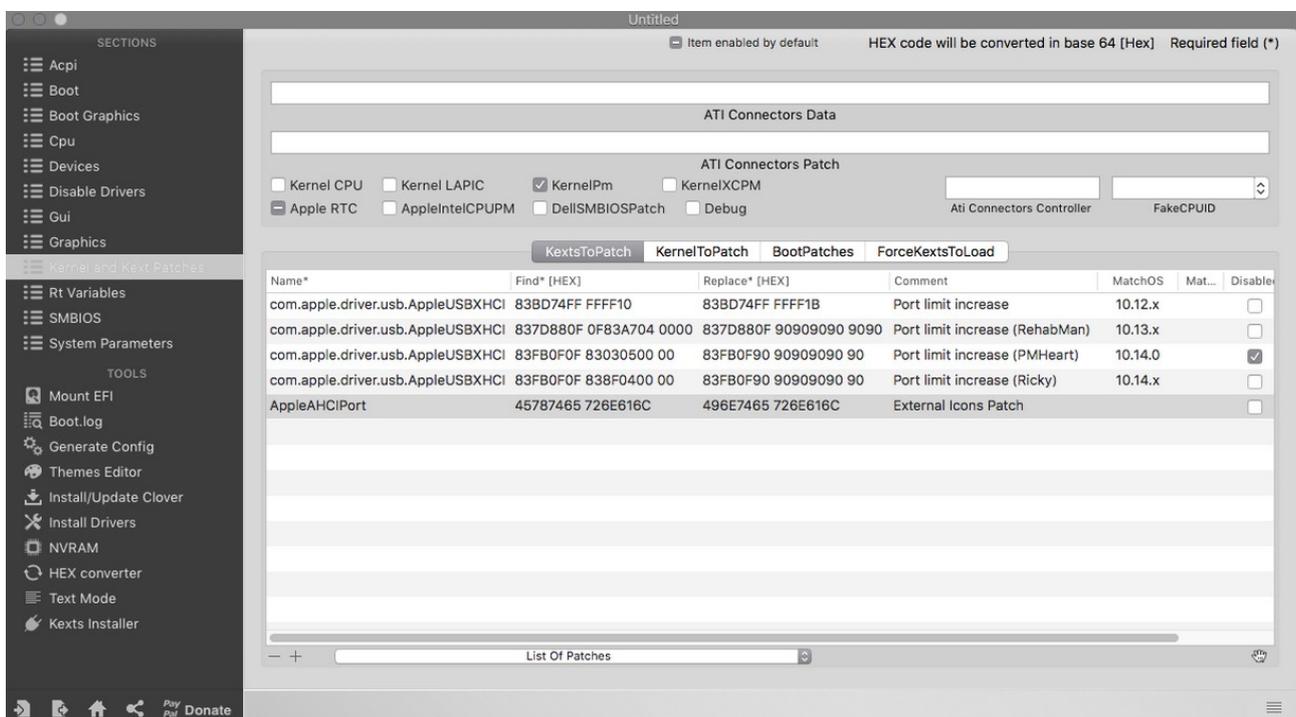
```
<key>Comment</key>
<string>Port limit increase (RehabMan)</string>
<key>Disabled</key>
<false/>
<key>Find</key>
<data>
g32IDw+DpwQAAA==
</data>
<key>InfoPlistPatch</key>
<false/>
<key>MatchOS</key>
<string>10.13.x</string>
<key>Name</key>
<string>com.apple.driver.usb.AppleUSBXHCI</string>
<key>Replace</key>
<data>
g32ID5CQkJCQkA==
</data>
</dict>
<dict>
<key>Comment</key>
<string>Port limit increase (PMHeart)</string>
<key>Disabled</key>
<true/>
<key>Find</key>
<data>
g/SPD4MDBQAA
</data>
<key>InfoPlistPatch</key>
<false/>
<key>MatchOS</key>
<string>10.14.0</string>
<key>Name</key>
<string>com.apple.driver.usb.AppleUSBXHCI</string>
<key>Replace</key>
<data>
g/sPkJCQkJCQ
</data>
</dict>
<dict>
<key>Comment</key>
<string>Port limit increase (Ricky)</string>
<key>Disabled</key>
<false/>
<key>Find</key>
<data>
g/SPD40PBAAA
</data>
<key>InfoPlistPatch</key>
<false/>
<key>MatchOS</key>
<string>10.14.x</string>
<key>Name</key>
<string>com.apple.driver.usb.AppleUSBXHCI</string>
<key>Replace</key>
<data>
g/sPkJCQkJCQ
</data>
</dict>
<dict>
<key>Comment</key>
<string>External Icons Patch</string>
<key>Disabled</key>
<false/>
```

```

</key>Find</key>
<data>
RXh0ZXJuYWw=
</data>
<key>InfoPlistPatch</key>
<false/>
<key>Name</key>
<string>AppleAHCIPort</string>
<key>Replace</key>
<data>
SW50ZXJuYWw=
</data>
</dict>
</array>
</dict>

```

Clover Configurator Screenshots



Skylake KernelAndKextPatches CC Section

Erläuterung

In diesem Abschnitt haben wir ein paar Einstellungen aktiviert und einige Kext-Patches hinzugefügt.

Checkboxes:

Wir haben hier ein paar Checkboxes: aktiviert:

- Apple RTC - dies stellt sicher, dass wir keinen BIOS-Reset beim Neustart haben.
- KernelPM - diese Einstellung verhindert das Schreiben auf MSR 0xe2, was eine Kernel-Panik beim Booten verhindern kann.

KextsToPatch:

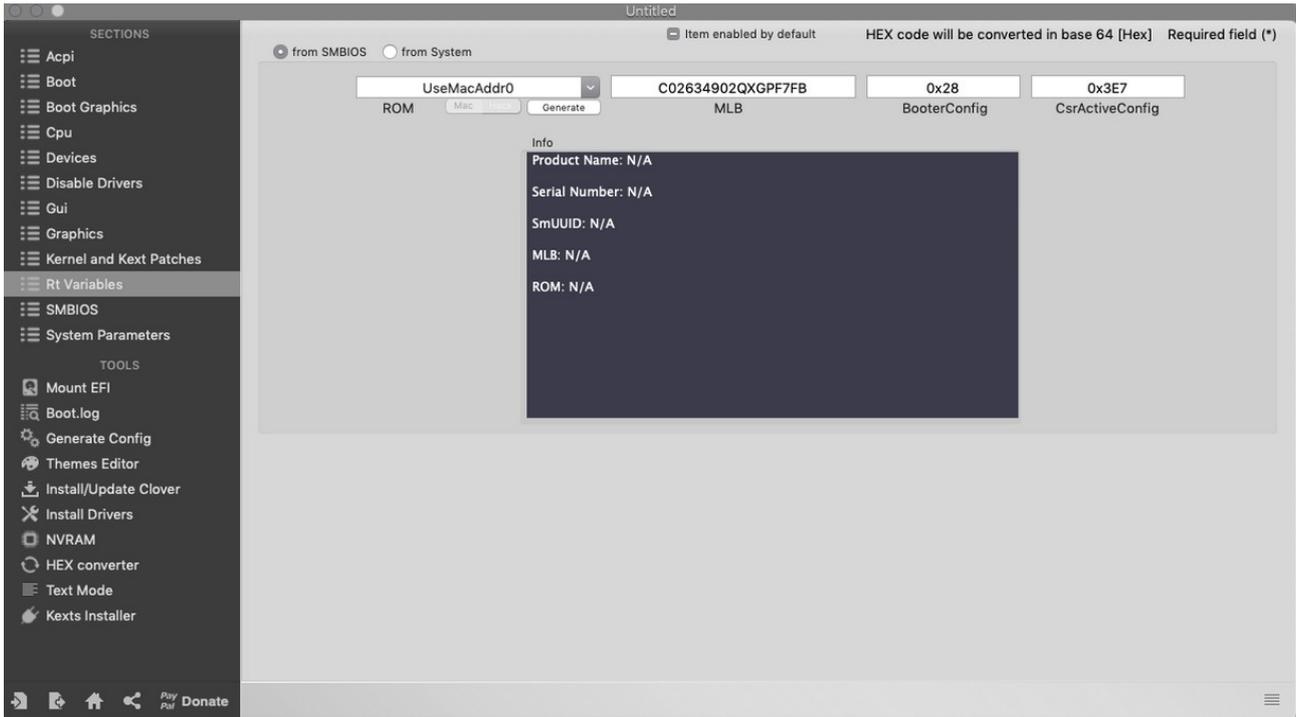
Wir haben hier 4 verschiedene Kexts zum Patchen hinzugefügt. Drei davon sind für die Erhöhung des USB-Port-Limits und der letzte dient als Fix für orangefarbene Icons - wenn interne Laufwerke hotplug-fähig sind und als externe Laufwerke behandelt werden. Sie werden feststellen, dass für jeden der USB-Port-Limit-Patches MatchOS-Werte gesetzt sind. Sie können alle Einträge für Betriebssystemversionen, die Sie nicht verwenden wollen, entfernen. Sie schaden nicht, wenn sie da sind, aber wenn Sie eine saubere, minimale Liste wollen, ist es nicht sinnvoll, sie zu haben.

RtVariables And SMBIOS

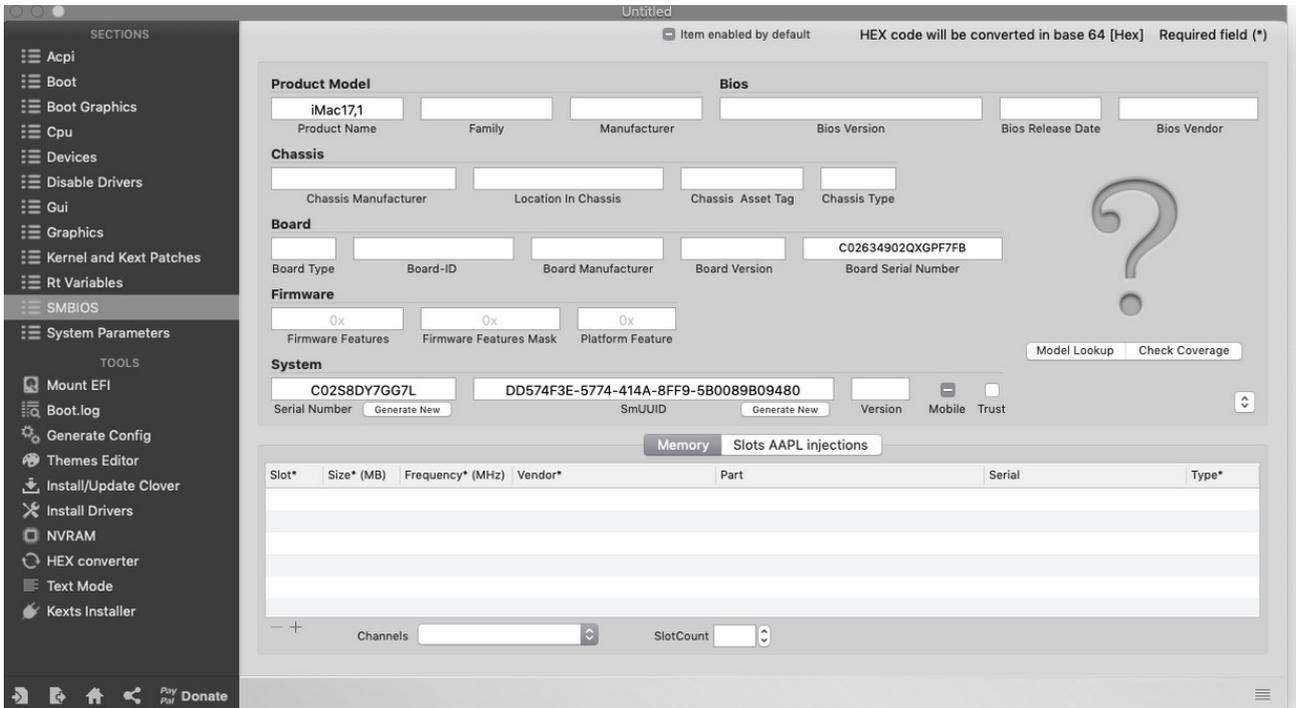
Raw XML

```
<key>RtVariables</key>
<dict>
  <key>BooterConfig</key>
  <string>0x28</string>
  <key>CsrActiveConfig</key>
  <string>0x3E7</string>
  <key>MLB</key>
  <string>C02634902QXGPF7FB</string>
  <key>ROM</key>
  <string>UseMacAddr0</string>
</dict>
<key>SMBIOS</key>
<dict>
  <key>BoardSerialNumber</key>
  <string>C02634902QXGPF7FB</string>
  <key>ProductName</key>
  <string>iMac17,1</string>
  <key>SerialNumber</key>
  <string>C02S8DY7GG7L</string>
  <key>SmUUID</key>
  <string>DD574F3E-5774-414A-8FF9-5B0089B09480</string>
</dict>
```

Clover Configurator Screenshots



Skylake RtVariables CC Section



Skylake SMBIOS CC Section

Erläuterung

Zum Einrichten der SMBIOS-Informationen verwende ich die [macserial](#) von acidanthera. Ich habe ein [python script](#) geschrieben, das es ebenfalls nutzen kann (und automatisch in der config.plist speichert, wenn es ausgewählt wird). Es gibt viele Informationen, die leer gelassen werden, damit Clover die Leerstellen ausfüllen kann; das bedeutet, dass eine Aktualisierung von Clover die übergebenen Informationen aktualisiert und Sie nicht auch Ihre config.plist aktualisieren müssen.

Für dieses Skylake-Beispiel habe ich das iMac17,1 SMBIOS gewählt.

Um die SMBIOS-Informationen zu erhalten, die mit macserial generiert wurden, können Sie es mit dem Argument -a ausführen (das Seriennummern und Board-Seriennummern für alle unterstützten Plattformen generiert). Sie können es auch mit grep parsen, um Ihre Suche auf einen SMBIOS-Typ zu beschränken.

Bei unserem iMac17,1-Beispiel würden wir macserial wie folgt über das Terminal ausführen:

```
macserial -a | grep -i iMac17,1
```

Dies würde eine Ausgabe ähnlich der folgenden ergeben:

Product	Serial	Board Serial (MLB)
iMac17,1	C02S8DY7GG7L	C02634902QXGPF7FB
iMac17,1	C02T4WZSGG7L	C02703104GUGPF71M
iMac17,1	C02QQAYPGG7L	C025474014NGPF7FB
iMac17,1	C02SNLZ3GG7L	C02645501CDGPF7AD
iMac17,1	C02QQRY8GG7L	C025474054NGPF71F
iMac17,1	C02QK1ZXGG7L	C02542200GUGPF7JC
iMac17,1	C02SL0YXGG7L	C026436004NGPF7JA
iMac17,1	C02QW0J5GG7L	C02552130QXGPF7JA
iMac17,1	C02RXDZYGG7L	C02626100GUGPF71H
iMac17,1	C02R4MYRGG7L	C02603200GUGPF7JA

Der Teil iMac17,1 wird nach SMBIOS -> Produktname kopiert.

Der Teil "Serial" wird nach SMBIOS -> Serial Number kopiert.

Der Teil "Board Serial" wird zu SMBIOS -> Board Serial Number sowie zu Rt Variables -> MLB kopiert.

Wir können eine SmUUID erstellen, indem wir uuidgen im Terminal ausführen (oder sie wird automatisch über mein GenSMBIOS-Skript generiert) - und diese wird nach SMBIOS -> SmUUID kopiert.

Wir setzen Rt Variables -> ROM auf UseMacAddr0, was einfach unsere Onboard-Mac-Adresse verwendet - diese sollte eindeutig genug sein, um nicht mit anderen in Konflikt zu geraten.

BooterConfig wird auf 0x28 gesetzt, und CsrActiveConfig wird auf 0x3e7 gesetzt, was SIP effektiv deaktiviert. Sie können eine Reihe von anderen Optionen wählen, um Abschnitte von SIP zu aktivieren/deaktivieren. Einige gängige sind wie folgt:

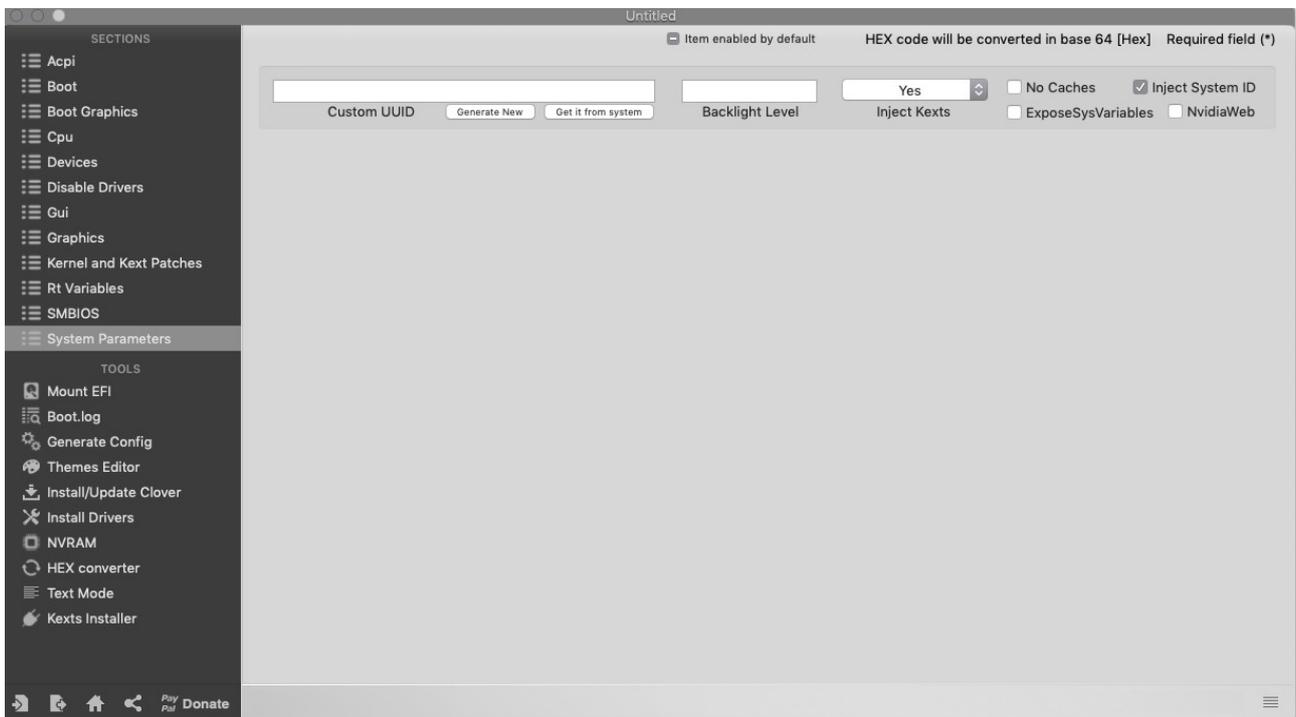
- 0x0 - SIP vollständig aktiviert
- 0x3 - Erlaubt unsignierte Kexts und das Schreiben auf geschützte fs-Speicherplätze
- 0x3e7 - SIP komplett deaktiviert

System Parameters

Raw XML

```
<key>SystemParameters</key>
<dict>
  <key>InjectKexts</key>
  <string>Yes</string>
  <key>InjectSystemID</key>
  <true/>
</dict>
```

Clover Configurator Screenshots



System Parameters CC Section

Erläuterung

Inject Kexts:

Diese Einstellung hat 3 Modi:

- Ja - damit wird Clover angewiesen, Kexts aus der EFI unabhängig davon zu injizieren.
- Nein - damit wird Clover angewiesen, keine Kexts von der EFI zu injizieren.
- Detect - Clover injiziert Kexts nur, wenn FakeSMC.kext nicht im Kext-Cache ist.

Wir setzen es auf Yes, um sicherzustellen, dass alle Kexts, die wir zuvor hinzugefügt haben, richtig injiziert werden.

InjectSystemID

Diese Einstellung weist Clover an, die SmUUID als System-ID beim Booten zu setzen - was wichtig für iMessage und so ist.

Saving

An dieser Stelle können Sie mit File -> Save die config.plist speichern. Wenn Sie Probleme haben, die Datei direkt im EFI zu speichern, können Sie sie auch auf dem Desktop speichern und dann einfach rüberkopieren. Ich lasse die [sample config.plist here](#) auch hier.

Kaby Lake

Wir gehen die Abschnitte der config.plist durch, einen nach dem anderen für ein Skylake-Desktop-Setup.

Ausgangspunkte

Ich beginne gerne entweder mit der config.plist, die Clover Ihnen zur Verfügung stellt, oder mit einer leeren Leinwand. In den nächsten Beispielen zeige ich Ihnen, wie ich die Dinge von Grund auf neu einrichte; wenn Sie von woanders anfangen, haben Sie vielleicht mehr Dinge überprüft/eingestellt als ich - aber Sie werden dem folgen wollen, was ich mache. Ich werde auch die rohen xml-Beispiele mit einbeziehen, um denjenigen zu zeigen, die mit einem Texteditor arbeiten (wie ich es bevorzuge).

ACPI

Die Standardeinstellungen von Clover sind ziemlich übertrieben und können einige Probleme verursachen. Wir werden diesen Abschnitt ziemlich minimal halten, und ich werde auch ein wenig darauf eingehen, warum wir das für jeden Teil tun.

Raw XML

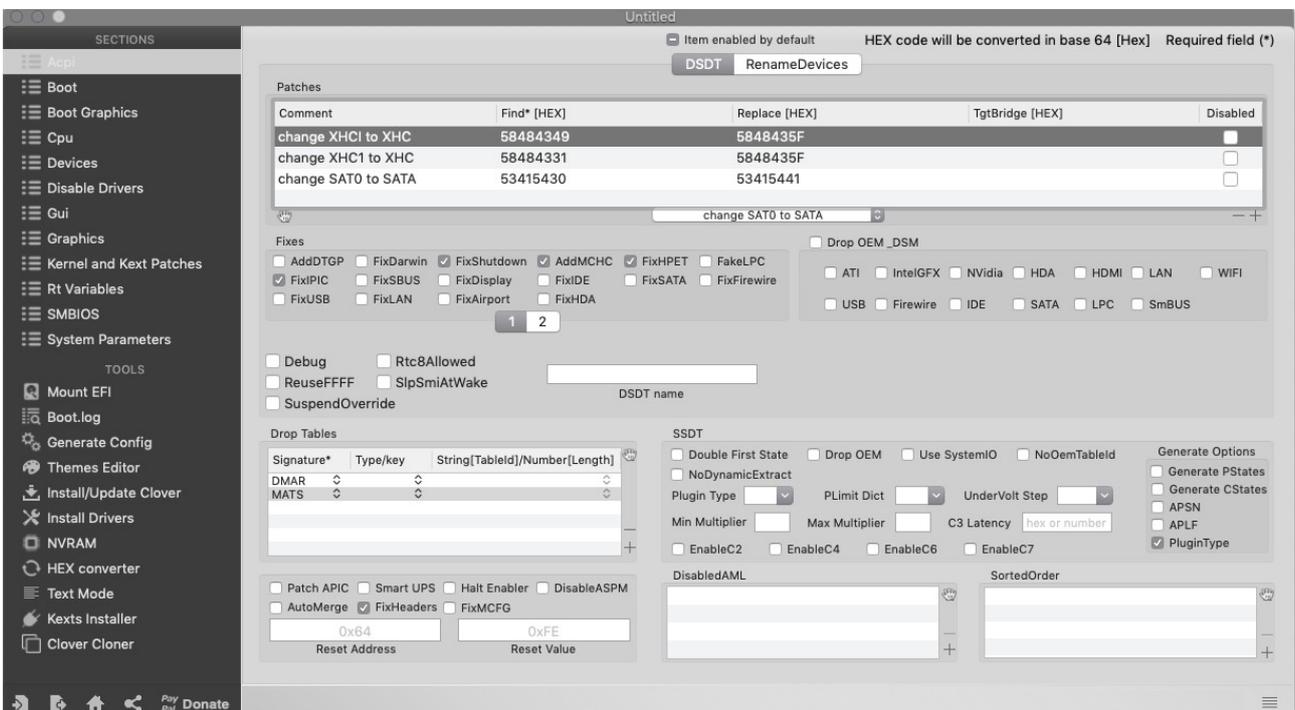
```
<key>ACPI</key>
<dict>
  <key>DSDT</key>
  <dict>
    <key>Fixes</key>
    <dict>
      <key>AddMCHC</key>
      <true/>
      <key>FixHPET</key>
      <true/>
      <key>FixIPIC</key>
      <true/>
      <key>FixRTC</key>
      <true/>
      <key>FixShutdown</key>
      <true/>
      <key>FixTMR</key>
      <true/>
    </dict>
    <key>Patches</key>
    <array>
      <dict>
        <key>Comment</key>
        <string>change XHCI to XHC</string>
        <key>Disabled</key>
        <false/>
        <key>Find</key>
        <data>
          WEhDSQ==
        </data>
        <key>Replace</key>
        <data>
          WEhDXw==
        </data>
      </dict>
      <dict>
        <key>Comment</key>
        <string>change XHC1 to XHC</string>
        <key>Disabled</key>
        <false/>
        <key>Find</key>
        <data>
          WEhDMQ==
        </data>
        <key>Replace</key>
        <data>
          WEhDXw==
        </data>
      </dict>
      <dict>
        <key>Comment</key>
        <string>change SAT0 to SATA</string>
        <key>Disabled</key>
        <false/>
      </dict>
    </array>
  </dict>
</dict>
```

```

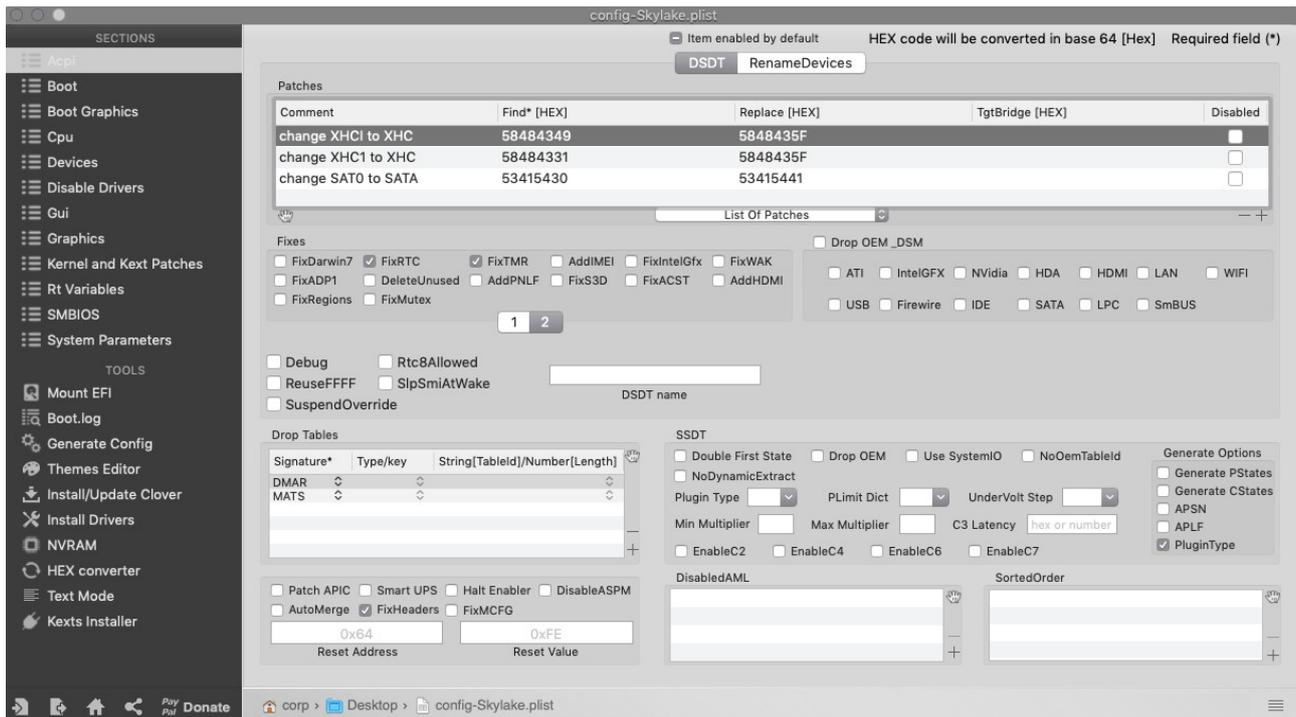
        <key>Find</key>
        <data>
        U0FUMA==
        </data>
        <key>Replace</key>
        <data>
        U0FUQQ==
        </data>
    </dict>
</array>
</dict>
<key>DropTables</key>
<array>
    <dict>
        <key>Signature</key>
        <string>DMAR</string>
    </dict>
    <dict>
        <key>Signature</key>
        <string>MATS</string>
    </dict>
</array>
<key>FixHeaders</key>
<true/>
<key>SSDT</key>
<dict>
    <key>Generate</key>
    <dict>
        <key>PluginType</key>
        <true/>
    </dict>
</dict>
</dict>

```

Clover Configurator Screenshots



Kaby Lake Acpi CC Section 1



Kaby Lake Acpi CC Section 2

Erläuterung

Patches:

Das erste, was wir durchgehen, ist der Abschnitt Patches. Dieser Abschnitt ermöglicht es uns, Teile des DSDT über Clover dynamisch umzubenennen. Da wir keinen echten Mac verwenden und macOS ziemlich genau darauf achtet, wie Dinge benannt werden, können wir nicht-destruktive Änderungen vornehmen, um die Dinge Mac-freundlich zu halten. Wir haben hier drei Einträge:

XHCI in XHC ändern - hilft, einen Konflikt mit eingebauten USB-Injektoren zu vermeiden

XHC1 in XHC ändern - hilft, einen Konflikt mit den eingebauten USB-Injektoren zu vermeiden

SATO zu SATA ändern - für mögliche SATA-Kompatibilität

Fixes:

Wenn wir uns dann den Abschnitt Fixes ansehen, werden wir sehen, dass wir ein paar Dinge überprüft haben (es gibt 2 Seiten, daher habe ich 2 Screenshots eingefügt):

FixShutdown - dies kann bei einigen Boards helfen, die es vorziehen, neu zu starten, anstatt herunterzufahren. Manchmal kann es bei anderen Boards zu Problemen beim Herunterfahren führen (ironisch, nicht wahr?), wenn Sie also Probleme beim Herunterfahren haben, wenn dies aktiviert ist, sollten Sie es deaktivieren.

Die übrigen Korrekturen helfen, IRQ-Konflikte usw. zu vermeiden, und sind nicht dafür bekannt, dass sie Probleme verursachen. Sie sind vielleicht nicht für jede Hardware notwendig, haben aber keine negativen Auswirkungen, wenn sie angewendet werden.

Drop Tables:

Wir haben DSDT in unserem Abschnitt über Patches schon leicht angerissen - und dies ist eine kleine Erweiterung davon. SSDT ist wie ein Unterabschnitt von DSDT. Der Abschnitt "Drop Tables" erlaubt es uns, bestimmte SSDT-Tabellen beim Laden auszulassen (wie ich bereits erwähnt habe, ist DSDT für Mac und PC unterschiedlich, und macOS kann ziemlich pingelig sein). Die beiden, die ich hinzugefügt habe, sind wie folgt:

DMAR - dies verhindert einige Probleme mit Vt-d; das ist PCI-Passthrough für VMs, und nicht sehr funktionell (wenn überhaupt?) auf Hackintoshes.

MATS - ab High Sierra wird diese Tabelle geparkt, und kann manchmal nicht druckbare Zeichen enthalten, die zu einer Kernel-Panic führen können.

FixHeaders und PluginType:

Die einzigen anderen Dinge, die wir auf dieser Seite getan haben, sind diese beiden Kontrollkästchen zu aktivieren.

FixHeaders - dies ist nur eine Verdoppelung unseres MATS-Tabellenabwurfs. Dieses Kontrollkästchen weist Clover an, Header zu bereinigen, um Kernel-Paniks im Zusammenhang mit nicht druckbaren Zeichen zu vermeiden.

PluginType - dies injiziert einige DSDT-Daten, um X86PlatformPlugin zu laden - was uns einen Vorsprung bei der nativen CPU-Leistungsverwaltung verschafft. Diese Einstellung funktioniert allerdings nur auf Haswell und neueren CPUs.

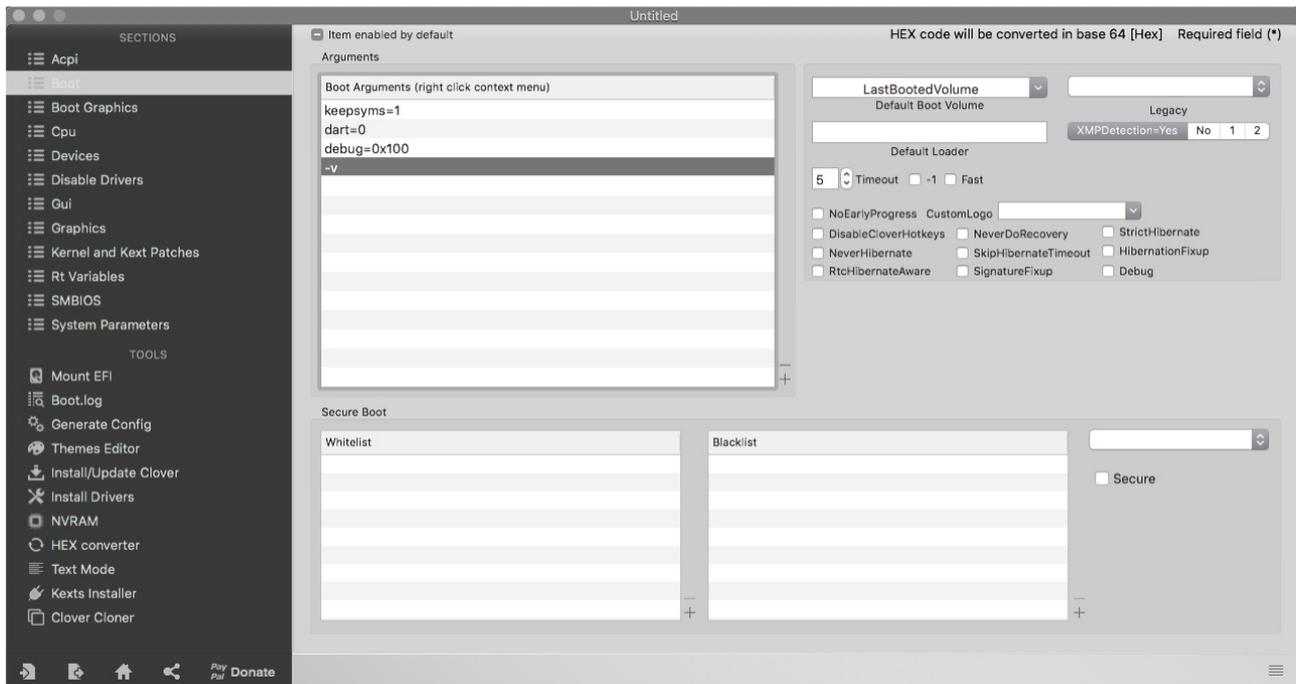
Boot

Wir brauchen hier nicht viel zu tun, aber wir werden ein paar Dinge ändern.

Raw XML

```
<key>Boot</key>
<dict>
  <key>Arguments</key>
  <string>keepsyms=1 dart=0 debug=0x100 -v</string>
  <key>DefaultVolume</key>
  <string>LastBootedVolume</string>
  <key>Timeout</key>
  <integer>5</integer>
</dict>
```

Clover Configurator Screenshots



Kaby Lake Boot CC Section

Erläuterung

Argumente:

Hier sind einige Boot-Argumente gesetzt:

-v - dies aktiviert den ausführlichen Modus, der den gesamten Text hinter den Kulissen anzeigt, der während des Bootvorgangs anstelle des Apple-Logos und des Fortschrittsbalkens vorbeiläuft. Dies ist für jeden Hackintosher von unschätzbarem Wert, da es Ihnen einen Einblick in den Boot-Prozess gibt und Ihnen helfen kann, Probleme, Problem-Kexts usw. zu identifizieren.

dart=0 - dies ist nur ein zusätzlicher Schutz gegen Vt-d-Probleme.

debug=0x100 - dies verhindert einen Neustart bei einer Kernel-Panik. Auf diese Weise können Sie (hoffentlich) einige nützliche Informationen sammeln und den Brotkrumen folgen, um die Probleme zu umgehen.

keepsyms=1 - dies ist eine begleitende Einstellung zu debug=0x100, die dem Betriebssystem sagt, dass es bei einer Kernel-Panik auch die Symbole ausgeben soll. Das kann einige hilfreichere Einblicke geben, was die Panik selbst verursacht.

DefaultBootVolume und Timeout:

Dies sind die einzigen anderen Einstellungen, die ich in diesem Abschnitt aktualisiert habe.

DefaultBootVolume - dies verwendet NVRAM, um sich zu merken, welches Volume zuletzt von Clover gebootet wurde, und wählt dieses beim nächsten Booten automatisch aus.

Timeout - dies ist die Anzahl der Sekunden, bevor das DefaultBootVolume automatisch gebootet wird. Sie können diesen Wert auf -1 setzen, um alle Timeouts zu vermeiden, oder auf 0, um die GUI komplett zu überspringen. Wenn Sie diesen Wert auf 0 setzen, können Sie beim Booten beliebige Tasten drücken, um die grafische Benutzeroberfläche wieder anzuzeigen, falls Probleme auftreten.

Boot Graphics

Hier steht nichts - nur die Standardeinstellungen. Sie könnten dies anpassen, wenn die Skalierung von Clover geändert werden muss, aber ich spiele damit nicht herum.

Cpu

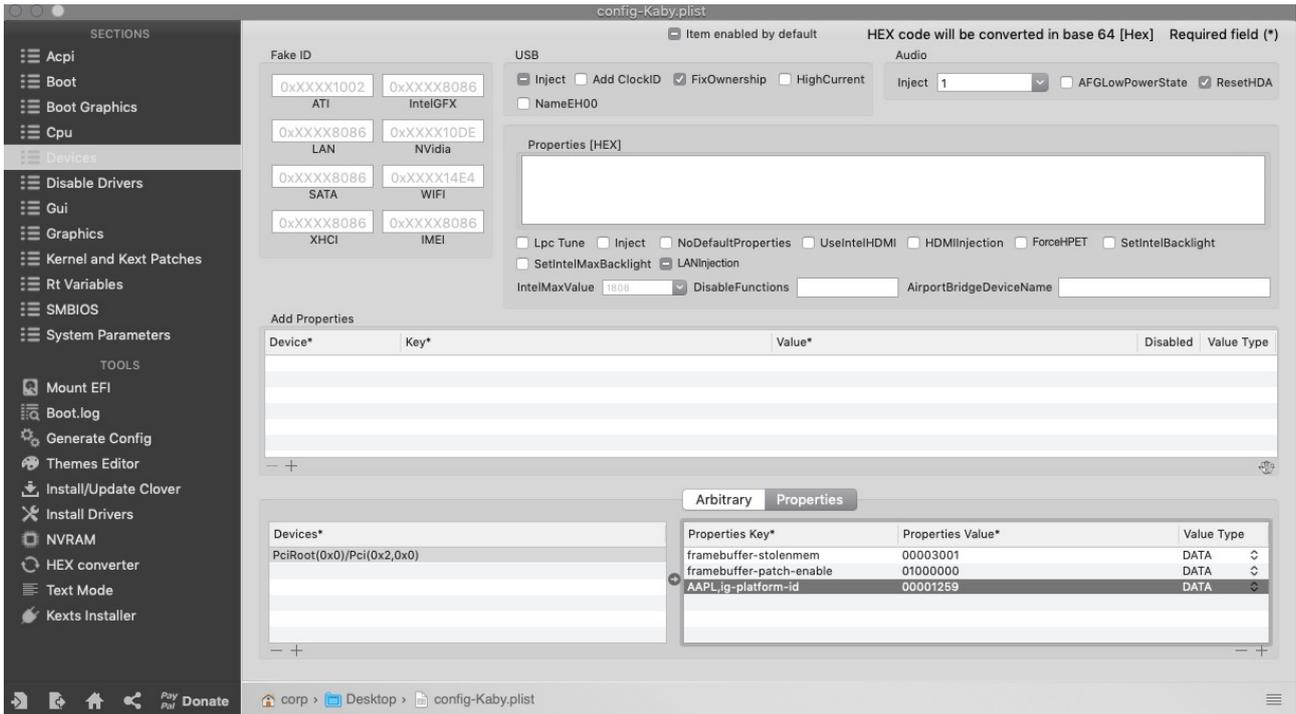
Auch hier wird in den meisten Setups, mit denen ich gearbeitet habe, nichts geändert.

Devices

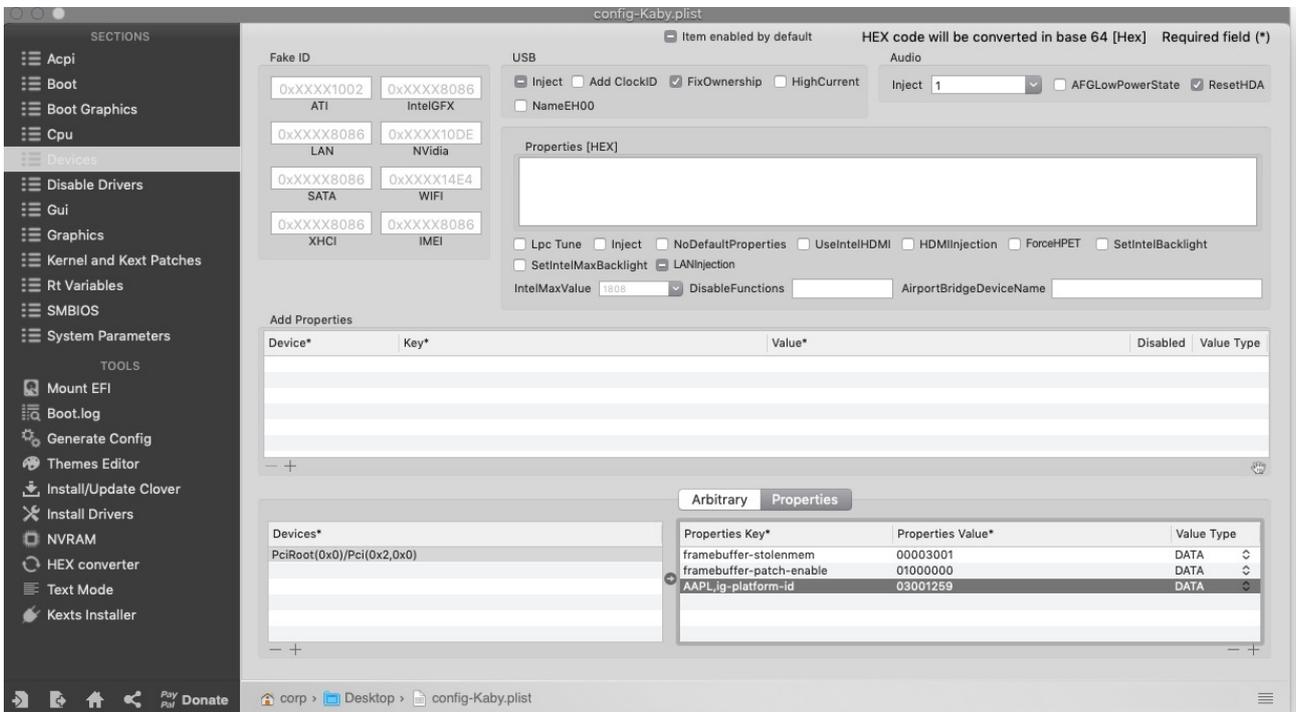
Raw XML

```
<key>Devices</key>
<dict>
  <key>Audio</key>
  <dict>
    <key>Inject</key>
    <integer>1</integer>
    <key>ResetHDA</key>
    <true/>
  </dict>
  <key>Properties</key>
  <dict>
    <key>PciRoot(0x0)/Pci(0x2,0x0)</key>
    <dict>
      <key>AAPL,ig-platform-id</key>
      <data>
        AAASWQ==
      </data>
      <key>framebuffer-patch-enable</key>
      <data>
        AQAAAA==
      </data>
      <key>framebuffer-stolenmem</key>
      <data>
        AAAwAQ==
      </data>
    </dict>
  </dict>
  <key>USB</key>
  <dict>
    <key>FixOwnership</key>
    <true/>
  </dict>
</dict>
```

Clover Configurator Screenshots



Kaby Lake Devices CC Section - iGPU



Kaby Lake Devices CC Section - iGPU Connectorless

Erläuterung

Fake ID:

Dieser Abschnitt bleibt für unser Beispiel-Setup leer. In der Vergangenheit wurden fast unterstützte iGPUs (wie die HD 4400) hier zu einer unterstützten iGPU gefälscht, aber wir werden den sauberen Properties Abschnitt dafür verwenden.

USB:

In diesem Abschnitt stellen wir sicher, dass Inject und FixOwnership ausgewählt sind, um Probleme mit dem Hängenbleiben an einer halb ausgedruckten Zeile irgendwo in der Nähe der Zeile Enabling Legacy Matching verbose zu vermeiden. Sie können das auch umgehen, indem Sie XHCI Hand Off im BIOS aktivieren.

Audio:

Hier haben wir unser Audio auf inject Layout 1 gesetzt - dies kann mit Ihrem Codec kompatibel sein oder auch nicht, aber Sie können auf der [AppleALC's Supported Codec Page](#) für unterstützte Codecs nachsehen.

Wir haben auch ResetHDA aktiviert, was den Codec zwischen den Neustarts des Betriebssystems in einen neutralen Zustand versetzt. Dies verhindert einige Probleme mit fehlendem Audio nach dem Booten zu einem anderen Betriebssystem und dann zurück.

Properties:

Dieser Abschnitt wird über Headkaze's [Intel Framebuffer Patching Guide](#) eingerichtet und wendet nur eine tatsächliche Eigenschaft an, nämlich die ig-platform-id. Um den richtigen Wert dafür zu erhalten, müssen wir uns die ig-platform-id ansehen, die wir verwenden wollen, und dann die Hex-Byte-Paare vertauschen.

Wenn wir uns unsere ig-platform-id als 0xAABBCCDD vorstellen, würde unsere vertauschte Version wie 0xDDCCBBAA aussehen.

Die beiden ig-platform-id's, die wir verwenden, sind wie folgt:

0x59120000 - dies wird verwendet, wenn die iGPU zur Ansteuerung eines Displays verwendet wird

00001259, wenn die hex-swapped

AAASWQ== wenn die hex-swapped Version in base64 konvertiert wird

0x59120003 - dies wird verwendet, wenn die iGPU nur für Rechenaufgaben verwendet wird und kein Display ansteuert

03001259 wenn hex-swapped

AwASWQ== wenn die hex-swapped Version nach base64 konvertiert wird

Wir fügen auch 2 weitere Eigenschaften hinzu, framebuffer-patch-enable und framebuffer-stolenmem. Die erste aktiviert das Patchen über WhateverGreen.kext, und die zweite setzt den minimalen gestohlenen Speicher auf 19 MB.

Disable Drivers

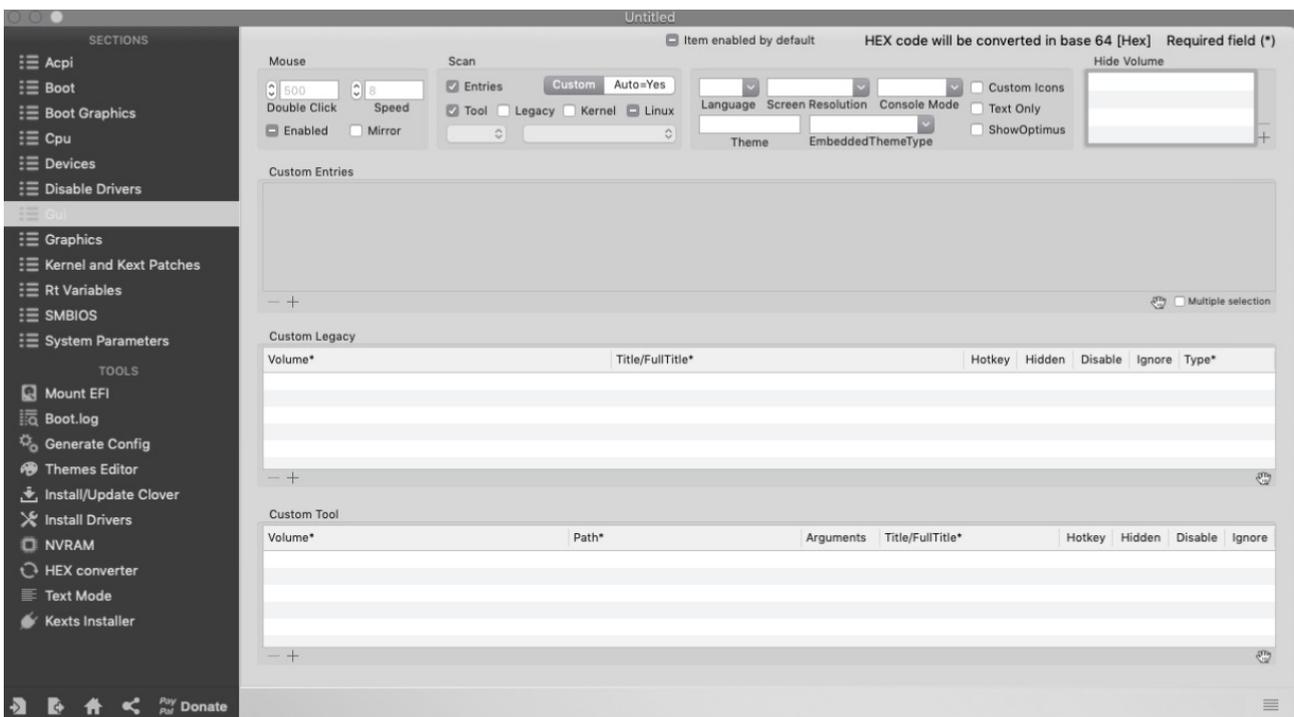
Hier gibt es nichts zu tun.

Gui

Raw XML

```
<key>GUI</key>
<dict>
  <key>Scan</key>
  <dict>
    <key>Entries</key>
    <true/>
    <key>Tool</key>
    <true/>
  </dict>
</dict>
```

Clover Configurator Screenshots



Kaby Lake Gui CC Section

Erläuterung

Scan:

Die einzigen Einstellungen, die ich auf dieser Seite geändert habe, sind die Scan-Einstellungen. Ich habe "Benutzerdefiniert" ausgewählt und dann alles außer "Legacy" und "Kernel" markiert. Dies lässt nur einige der nicht bootfähigen Einträge in Clover weg, um das Menü aufzuräumen.

Volumes ausblenden

Ich habe hier nichts hinzugefügt, aber Sie können unerwünschte Volumes hier ausblenden. Sie können dies tun, indem Sie entweder den Namen des Volumes oder die UUID hinzufügen.

Um zusätzliche APFS-Einträge auszublenden, fügen Sie Folgendes zu dieser Liste hinzu:

Preboot

VM

Um alle Recovery-Partitionen auszublenden, fügen Sie Recovery zu dieser Liste hinzu. Um die UUID eines zu versteckenden Laufwerks zu erhalten, können Sie den folgenden Terminalbefehl verwenden:

```
diskutil info diskXsY | grep -i "Partition UUID" | rev | cut -d ' ' -f 1 | rev
```

Stellen Sie sicher, dass Sie diskXsY durch die tatsächliche Festplattennummer des Volumes ersetzen, das Sie ausblenden möchten.

Theme:

Wenn Sie ein neues Thema ausprobieren möchten (ich empfehle Ihnen clover-next-black), können Sie den entpackten Themenordner zum Verzeichnis `/Volumes/EFI/EFI/CLOVER/themes` hinzufügen und dann den Namen des Ordners in das Textfeld Thema eingeben, um ihn anzuwenden.

Graphics

In der Vergangenheit haben wir hier die iGPU eingerichtet, aber da wir dies bereits über Eigenschaften im Abschnitt Geräte getan haben, müssen wir hier nichts mehr konfigurieren. **HINWEIS:** Wenn Clover eine Intel iGPU erkennt, aktiviert es automatisch Intel Injection, wenn der Abschnitt Graphics in der config.plist nicht vorhanden ist. Um dies zu umgehen, können Sie die Injektion explizit deaktivieren, indem Sie das unten stehende Roh-XML verwenden, oder indem Sie die Schaltfläche "Inject Intel" einmal anklicken, um sie zu aktivieren, und einmal, um sie in CC zu deaktivieren.

Raw XML

```
<key>Graphics</key>
<dict>
  <key>Inject</key>
  <false/>
</dict>
```

Kernel And Kext Patches

Raw XML

```
<key>KernelAndKextPatches</key>
<dict>
  <key>KernelPm</key>
  <true/>
  <key>KextsToPatch</key>
  <array>
    <dict>
      <key>Comment</key>
      <string>Port limit increase</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
        g710////EA==
      </data>
      <key>InfoPlistPatch</key>
      <false/>
      <key>MatchOS</key>
      <string>10.12.x</string>
      <key>Name</key>
      <string>com.apple.driver.usb.AppleUSBXHCI</string>
      <key>Replace</key>
      <data>
        g710////Gw==
      </data>
    </dict>
    <dict>
      <key>Comment</key>
      <string>Port limit increase (RehabMan)</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
        g32IDw+DpwQAAA==
      </data>
    </dict>
  </array>
</dict>
```

```
</data>
<key>InfoPlistPatch</key>
<false/>
<key>MatchOS</key>
<string>10.13.x</string>
<key>Name</key>
<string>com.apple.driver.usb.AppleUSBXHCI</string>
<key>Replace</key>
<data>
g32ID5CQkJCQkA==
</data>
</dict>
<dict>
<key>Comment</key>
<string>Port limit increase (PMHeart)</string>
<key>Disabled</key>
<true/>
<key>Find</key>
<data>
g/sPD4MDBQAA
</data>
<key>InfoPlistPatch</key>
<false/>
<key>MatchOS</key>
<string>10.14.0</string>
<key>Name</key>
<string>com.apple.driver.usb.AppleUSBXHCI</string>
<key>Replace</key>
<data>
g/sPkJCQkJCQ
</data>
</dict>
<dict>
<key>Comment</key>
<string>Port limit increase (Ricky)</string>
<key>Disabled</key>
<false/>
<key>Find</key>
<data>
g/sPD4OPBAAA
</data>
<key>InfoPlistPatch</key>
<false/>
<key>MatchOS</key>
<string>10.14.x</string>
<key>Name</key>
```

```

<string>com.apple.driver.usb.AppleUSBXHCI</string>
<key>Replace</key>
<data>
g/sPkJCQkJCQ
</data>
</dict>
<dict>
<key>Comment</key>
<string>External Icons Patch</string>
<key>Disabled</key>
<false/>
<key>Find</key>
<data>
RXh0ZXJuYWw=
</data>
<key>InfoPlistPatch</key>
<false/>
<key>Name</key>
<string>AppleAHCIPort</string>
<key>Replace</key>
<data>
SW50ZXJuYWw=
</data>
</dict>
</array>
</dict>

```

Clover Configurator Screenshots

The screenshot shows the Clover Configurator interface. The left sidebar lists various sections and tools. The main window displays the 'KernelAndKextPatches' section, which includes a table of patches. The table has the following columns: Name*, Find* [HEX], Replace* [HEX], Comment, MatchOS, Mat..., and Disable. The 'AppleAHCIPort' patch is highlighted in the table.

Name*	Find* [HEX]	Replace* [HEX]	Comment	MatchOS	Mat...	Disable
com.apple.driver.usb.AppleUSBXHCI	83BD74FF FFFF10	83BD74FF FFFF1B	Port limit increase	10.12.x		<input type="checkbox"/>
com.apple.driver.usb.AppleUSBXHCI	837D880F 0F83A704 0000	837D880F 90909090 9090	Port limit increase (RehabMan)	10.13.x		<input type="checkbox"/>
com.apple.driver.usb.AppleUSBXHCI	83FB0F0F 83030500 00	83FB0F90 90909090 90	Port limit increase (PMHeart)	10.14.0		<input checked="" type="checkbox"/>
com.apple.driver.usb.AppleUSBXHCI	83FB0F0F 838F0400 00	83FB0F90 90909090 90	Port limit increase (Ricky)	10.14.x		<input type="checkbox"/>
AppleAHCIPort	45787465 726E616C	496E7465 726E616C	External Icons Patch			<input type="checkbox"/>

Kaby Lake KernelAndKextPatches CC Section

Erläuterung

In diesem Abschnitt haben wir ein paar Einstellungen aktiviert und einige Kext-Patches hinzugefügt.

Checkboxes:

Wir haben hier ein paar Checkboxes aktiviert:

Apple RTC - dies stellt sicher, dass wir keinen BIOS-Reset beim Neustart haben.

KernelPM - diese Einstellung verhindert das Schreiben auf MSR 0xe2, was eine Kernel-Panik beim Booten verhindern kann.

KextsToPatch:

Wir haben hier 4 verschiedene Kexts zum Patchen hinzugefügt. Drei davon sind für die Erhöhung des USB-Port-Limits und der letzte dient als Fix für orangefarbene Icons - wenn interne Laufwerke hotplug-fähig sind und als externe Laufwerke behandelt werden.

Sie werden feststellen, dass für jeden der USB-Port-Limit-Patches MatchOS-Werte gesetzt sind. Sie können alle Einträge für Betriebssystemversionen, die Sie nicht verwenden wollen, entfernen. Sie schaden nicht, wenn sie da sind, aber wenn Sie eine saubere, minimale Liste wollen, ist es nicht sinnvoll, sie zu haben.

RtVariablen und SMBIOS

Raw XML

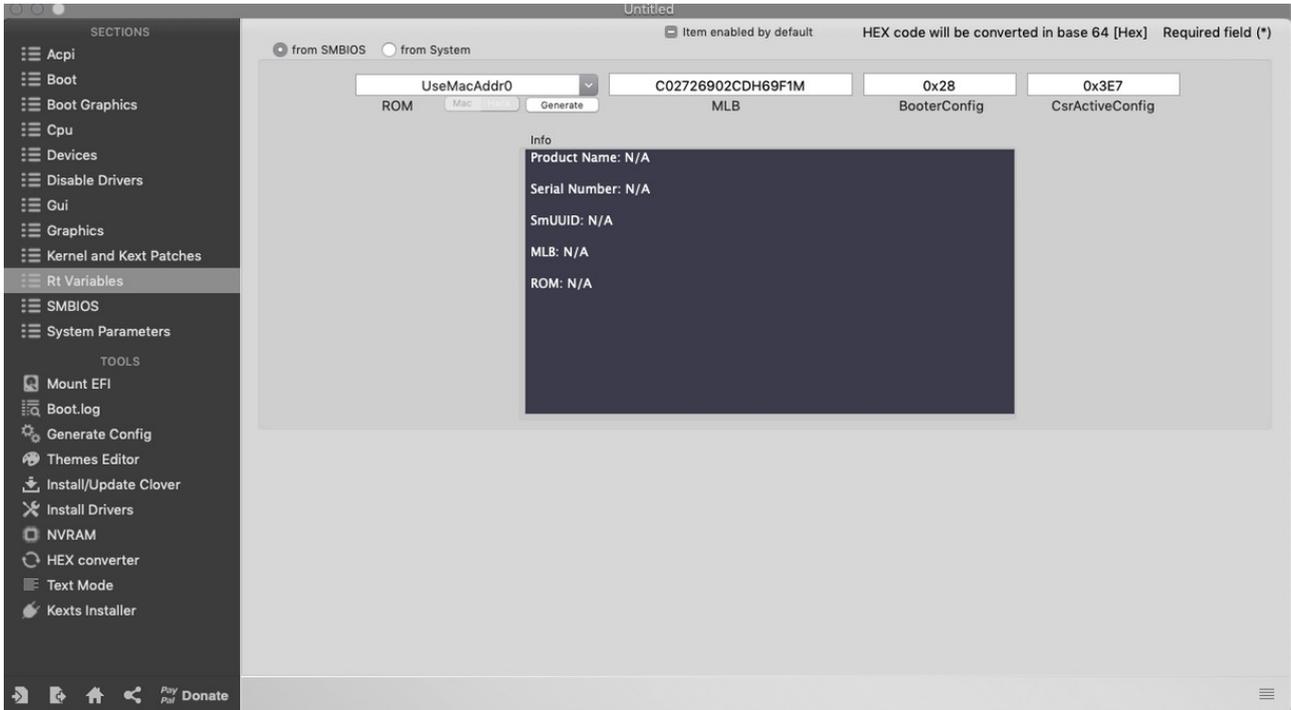
```
<key>RtVariables</key>
<dict>
  <key>BooterConfig</key>
  <string>0x28</string>
  <key>CsrActiveConfig</key>
  <string>0x3E7</string>
  <key>MLB</key>
  <string>C02726902CDH69F1M</string>
  <key>ROM</key>
  <string>UseMacAddr0</string>
</dict>
<key>SMBIOS</key>
<dict>
  <key>BoardSerialNumber</key>
  <string>C02726902CDH69F1M</string>
  <key>ProductName</key>
  <string>iMac18,1</string>
  <key>SerialNumber</key>
```

```

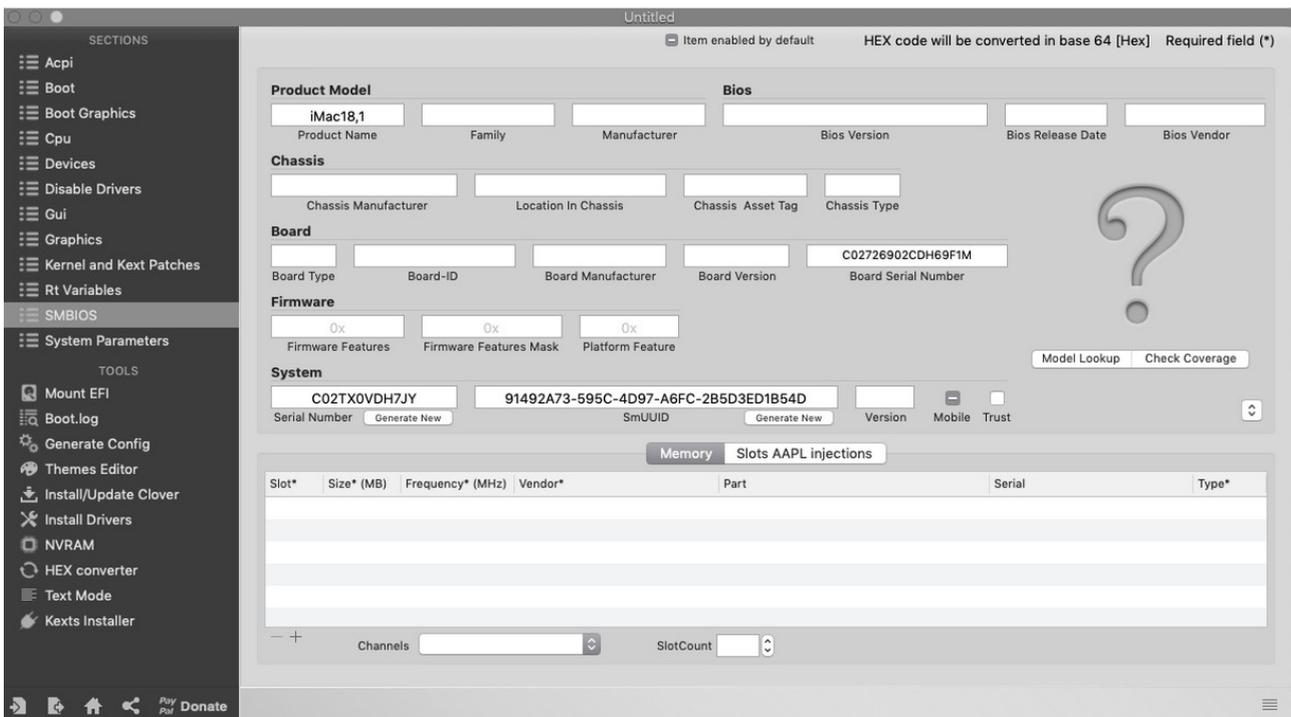
<string>C02TX0VDH7JY</string>
<key>SmUUID</key>
<string>91492A73-595C-4D97-A6FC-2B5D3ED1B54D</string>
</dict>

```

Clover Configurator Screenshots



Kaby Lake RtVariables CC Section



Kaby Lake SMBIOS CC Section

Erläuterung

Zum Einrichten der SMBIOS-Informationen verwende ich die [macserial](#) Anwendung von acidanthera. Ich habe ein [python script](#) geschrieben, das es ebenfalls nutzen kann (und automatisch in der config.plist speichert, wenn es ausgewählt wird). Es gibt viele Informationen, die leer gelassen werden, damit Clover die Leerstellen ausfüllen kann; das bedeutet, dass eine Aktualisierung von Clover die übergebenen Informationen aktualisiert, ohne dass Sie auch Ihre config.plist aktualisieren müssen.

Für dieses Kaby Lake Beispiel habe ich das iMac18,1 SMBIOS gewählt - dies geschieht absichtlich aus Kompatibilitätsgründen. Es gibt zwei Haupt-SMBIOS, die für Kaby Lake verwendet werden:

iMac18,1 - dies wird für Computer verwendet, die die iGPU für die Anzeige nutzen.

iMac18,3 - dies wird für Computer verwendet, die eine dGPU für die Anzeige und eine iGPU nur für Rechenaufgaben verwenden.

Um die SMBIOS-Informationen zu erhalten, die mit macserial generiert wurden, können Sie es mit dem Argument -a ausführen (das Seriennummern und Board-Seriennummern für alle unterstützten Plattformen generiert). Sie können es auch mit grep parsen, um Ihre Suche auf einen SMBIOS-Typ zu beschränken.

Bei unserem iMac18,1-Beispiel würden wir macserial wie folgt über das Terminal ausführen:

```
macserial -a | grep -i iMac18,1
```

Dies würde eine Ausgabe ähnlich der folgenden ergeben:

Product	Serial	Board Serial (MLB)
iMac18,1	C02T8SZNH7JY	C02707101J9H69F1F
iMac18,1	C02VXBYDH7JY	C02753100GUH69FCB
iMac18,1	C02T7RY6H7JY	C02706310GUH69FA8
iMac18,1	C02VD07ZH7JY	C02737301J9H69FCB
iMac18,1	C02TQPYPH7JY	C02720802CDH69FAD
iMac18,1	C02VXYVH7JY	C02753207CDH69FJC
iMac18,1	C02VDBZ0H7JY	C02737700QXH69FA8
iMac18,1	C02VP0H6H7JY	C02746300CDH69FJA
iMac18,1	C02VL0W9H7JY	C02743303CDH69F8C
iMac18,1	C02V2NYMH7JY	C02728600J9H69FAD

Der Teil iMac18,1 wird nach SMBIOS -> Produktname kopiert.

Der Teil "Serial" wird nach SMBIOS -> Serial Number kopiert.

Der Teil "Board Serial" wird zu SMBIOS -> Board Serial Number sowie zu Rt Variables -> MLB kopiert.

Wir können eine SmUUID erstellen, indem wir uuidgen im Terminal ausführen (oder sie wird automatisch über mein GenSMBIOS-Skript generiert) - und diese wird nach SMBIOS -> SmUUID kopiert.

Wir setzen Rt Variables -> ROM auf UseMacAddr0, was einfach unsere Onboard-Mac-Adresse verwendet - diese sollte eindeutig genug sein, um nicht mit anderen in Konflikt zu geraten.

BooterConfig wird auf 0x28 gesetzt, und CsrActiveConfig wird auf 0x3e7 gesetzt, was SIP effektiv deaktiviert. Sie können eine Reihe von anderen Optionen wählen, um Abschnitte von SIP zu aktivieren/deaktivieren. Einige gängige sind wie folgt:

0x0 - SIP vollständig aktiviert

0x3 - Erlaubt unsignierte Kexts und das Schreiben auf geschützte fs-Speicherplätze

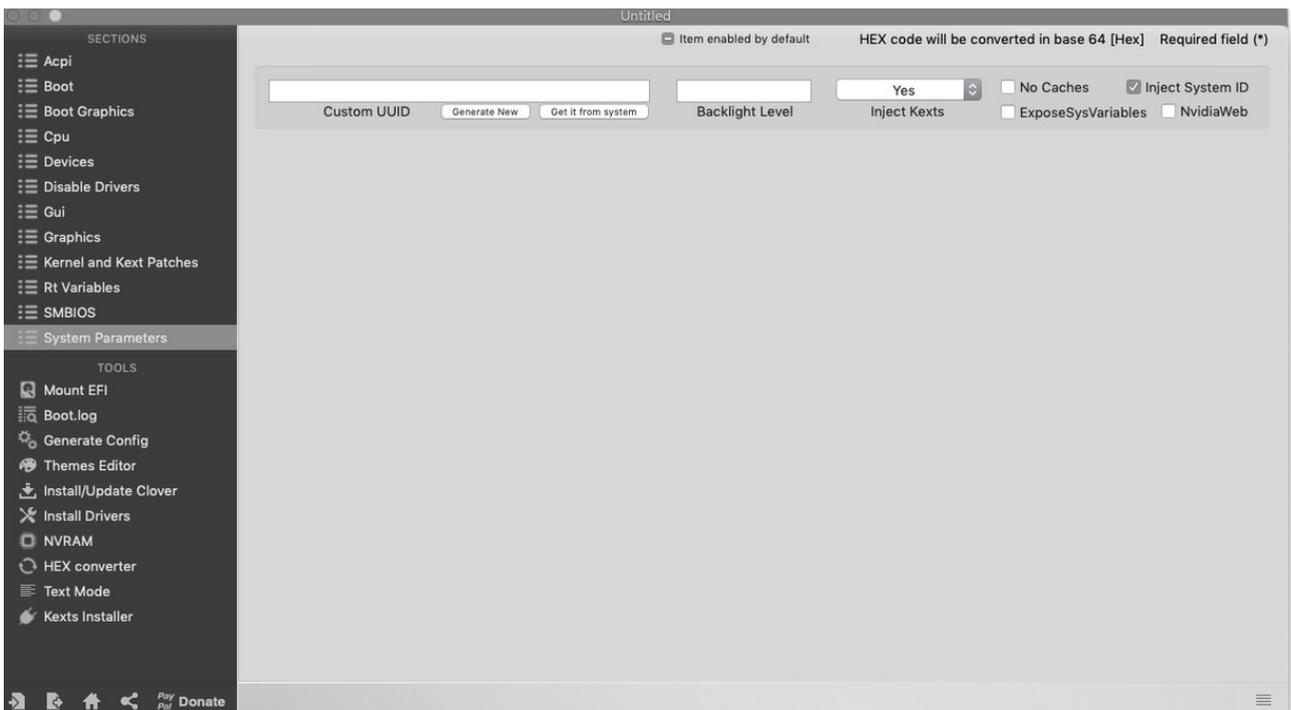
0x3e7 - SIP komplett deaktiviert

System-Parameter

Raw XML

```
<key>SystemParameters</key>
<dict>
  <key>InjectKexts</key>
  <string>Yes</string>
  <key>InjectSystemID</key>
  <true/>
</dict>
```

Clover Configurator Screenshots



System Parameters CC Section

Erläuterung
Kexts einspeisen:

Diese Einstellung hat 3 Modi:

Ja - damit wird Clover angewiesen, Kexts aus der EFI unabhängig davon zu injizieren.

Nein - damit wird Clover angewiesen, keine Kexts von der EFI zu injizieren.

Detect - Clover injiziert Kexts nur, wenn FakeSMC.kext nicht im Kext-Cache ist.

Wir setzen es auf Ja, um sicherzustellen, dass alle Kexts, die wir zuvor hinzugefügt haben, richtig injiziert werden.

InjectSystemID

Diese Einstellung weist Clover an, die SmUUID als System-ID beim Booten zu setzen - was wichtig ist für

Saving

An diesem Punkt können Sie mit File -> Save die config.plist speichern. Wenn Sie Probleme haben, die Datei direkt im EFI zu speichern, können Sie sie auch auf dem Desktop speichern und dann einfach rüberkopieren. Ich werde die Beispiel-Config.plist auch [sample config.plist here](#) hier lassen.

Coffee Lake

Wir gehen die Abschnitte der config.plist durch, einen nach dem anderen für eine Coffee Lake Desktop-Einrichtung.

Ausgangspunkte

Ich beginne gerne entweder mit der config.plist, die Clover Ihnen zur Verfügung stellt, oder mit einer leeren Leinwand. In den nächsten Beispielen zeige ich Ihnen, wie ich die Dinge

von Grund auf neu einrichte; wenn Sie von woanders anfangen, haben Sie vielleicht mehr Dinge überprüft/eingestellt als ich - aber Sie werden dem folgen wollen, was ich mache. Ich werde auch die rohen xml-Beispiele mit einbeziehen, um denjenigen zu zeigen, die mit einem Texteditor arbeiten (wie ich es bevorzuge).

ACPI

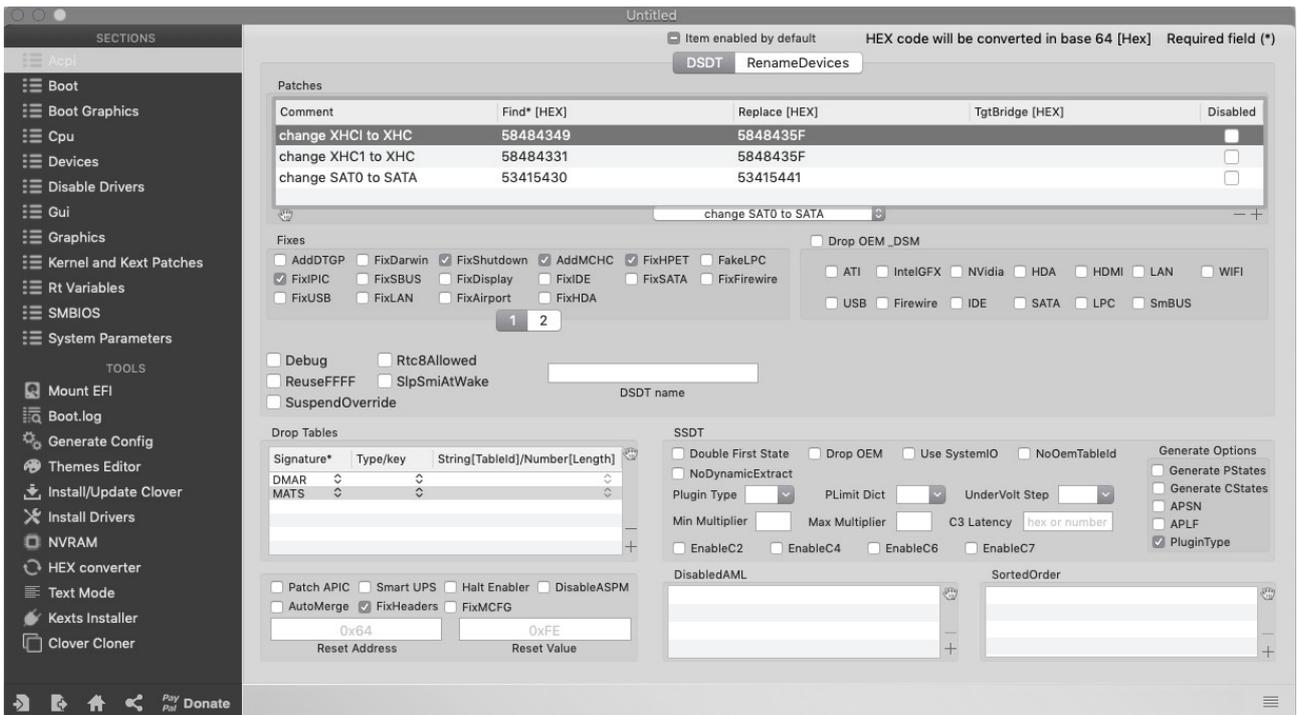
Die Standardeinstellungen von Clover sind ziemlich übertrieben und können einige Probleme verursachen. Wir werden diesen Abschnitt ziemlich minimal halten, und ich werde auch ein bisschen darauf eingehen, warum wir das für jeden Teil tun.

Raw XML

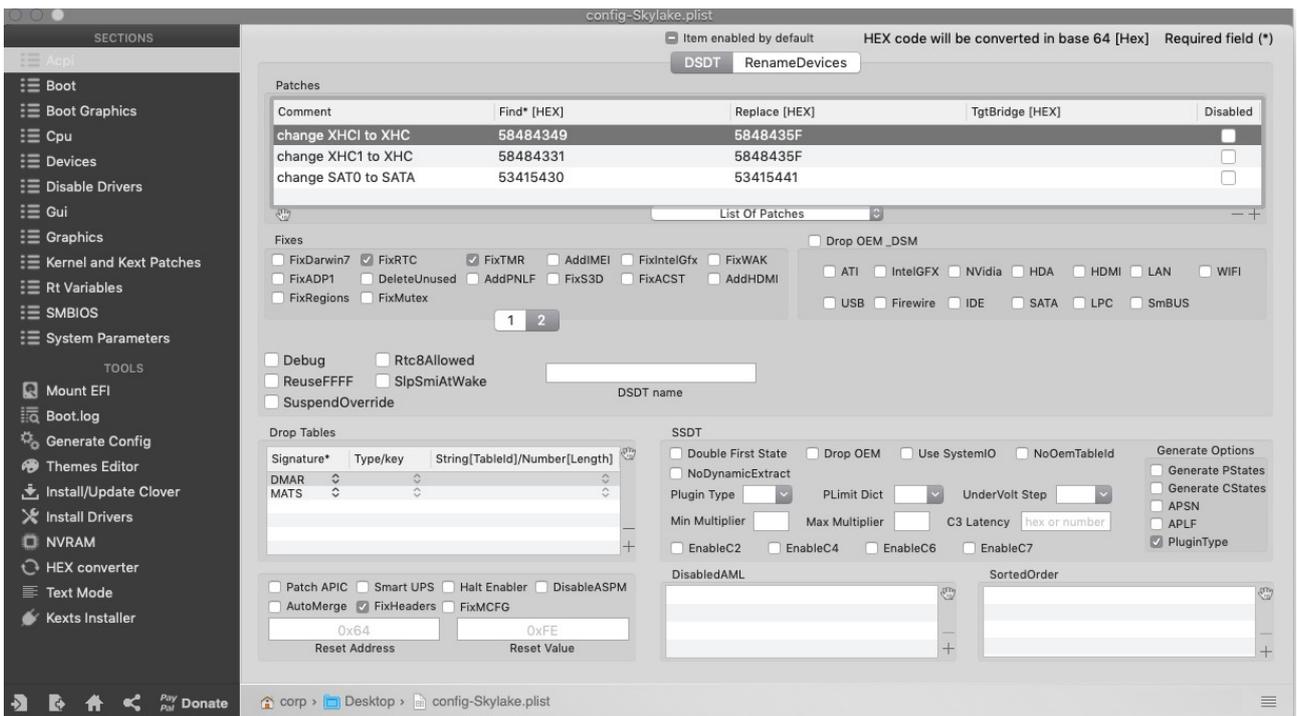
```
<key>ACPI</key>
<dict>
  <key>DSDT</key>
  <dict>
    <key>Fixes</key>
    <dict>
      <key>AddMCHC</key>
      <true/>
      <key>FixHPET</key>
      <true/>
      <key>FixIPIC</key>
      <true/>
      <key>FixRTC</key>
      <true/>
      <key>FixShutdown</key>
      <true/>
      <key>FixTMR</key>
      <true/>
    </dict>
  <key>Patches</key>
  <array>
    <dict>
      <key>Comment</key>
      <string>change XHCI to XHC</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
        WEhDSQ==
      </data>
      <key>Replace</key>
      <data>
        WEhDXw==
      </data>
    </dict>
    <dict>
      <key>Comment</key>
      <string>change XHC1 to XHC</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
        WEhDMQ==
      </data>
      <key>Replace</key>
      <data>
        WEhDXw==
      </data>
    </dict>
  </array>
</dict>
```

```
<dict>
  <key>Comment</key>
  <string>change SAT0 to SATA</string>
  <key>Disabled</key>
  <false/>
  <key>Find</key>
  <data>
    U0FUMA==
  </data>
  <key>Replace</key>
  <data>
    U0FUQQ==
  </data>
</dict>
</array>
</dict>
<key>DropTables</key>
<array>
  <dict>
    <key>Signature</key>
    <string>DMAR</string>
  </dict>
  <dict>
    <key>Signature</key>
    <string>MATS</string>
  </dict>
</array>
<key>FixHeaders</key>
<true/>
<key>SSDT</key>
<dict>
  <key>Generate</key>
  <dict>
    <key>PluginType</key>
    <true/>
  </dict>
</dict>
</dict>
```

Clover Configurator Screenshots



Coffee Lake Acpi CC Section 1



Coffee Lake Acpi CC Section 2

Comment	Find* [HEX]	Replace [HEX]	TgtBridge [HEX]	Disabled
Fix 300-series RTC Bug	A00A9353 54415301	A00A910A FF0BFFFF		<input type="checkbox"/>

Coffee Lake Read-Only RTC Fix

Erläuterung

Patches:

Das erste, was wir durchgehen, ist der Abschnitt Patches. Dieser Abschnitt ermöglicht es uns, Teile des DSDT über Clover dynamisch umzubenennen. Da wir keinen echten Mac verwenden und macOS ziemlich genau darauf achtet, wie Dinge benannt werden, können wir nicht-destruktive Änderungen vornehmen, um die Dinge Mac-freundlich zu halten. Wir haben hier drei Einträge:

- XHCI in XHC ändern - hilft, einen Konflikt mit eingebauten USB-Injektoren zu vermeiden
- XHC1 in XHC ändern - hilft, einen Konflikt mit den eingebauten USB-Injektoren zu vermeiden
- SAT0 auf SATA ändern - für mögliche SATA-Kompatibilität

Möglicherweise benötigen Sie auch den folgenden Read-Only-RTC-Fix (lesen Sie [here](#) darüber), wenn Sie ein echtes 300er-Board (nicht Z370) haben und an diesem Punkt ein Boot-Hänger auftritt:

```
ACPI: WSMT 0x000000007E3680A8 000020 (v01 ALASKA A M I 81072009 AMI 00010013)
ACPI: SSDT 0x000000007DEA2000 000053 (v01 PalRef CpuPa 00003000 INTL 20120320)
AppleCredentialManager: Init: called, instance = <ptr>.
AssertMacros: value (value: 0x0), file: /BuildRoot/Library/Caches/com.apple.xbs/Sources/AppleCredentialManager/AppleCredentialManager/AppleCredentialManager.m
ACM: InitCredentialEngine: Global credential set created, CSI101.
AppleCredentialManager: Init: returning, result = true, instance = <ptr>.
ACPI: 9 ACPI AML tables successfully acquired and loaded
AppleACPICPU: ProcessorId=1 LocalApicId=0 Enabled
AppleACPICPU: ProcessorId=2 LocalApicId=2 Enabled
AppleACPICPU: ProcessorId=3 LocalApicId=4 Enabled
AppleACPICPU: ProcessorId=4 LocalApicId=6 Enabled
AppleACPICPU: ProcessorId=5 LocalApicId=8 Enabled
AppleACPICPU: ProcessorId=6 LocalApicId=10 Enabled
AppleACPICPU: ProcessorId=7 LocalApicId=12 Enabled
AppleACPICPU: ProcessorId=8 LocalApicId=14 Enabled
AppleACPICPU: ProcessorId=9 LocalApicId=1 Enabled
AppleACPICPU: ProcessorId=10 LocalApicId=3 Enabled
AppleACPICPU: ProcessorId=11 LocalApicId=5 Enabled
AppleACPICPU: ProcessorId=12 LocalApicId=7 Enabled
AppleACPICPU: ProcessorId=13 LocalApicId=9 Enabled
AppleACPICPU: ProcessorId=14 LocalApicId=11 Enabled
AppleACPICPU: ProcessorId=15 LocalApicId=13 Enabled
AppleACPICPU: ProcessorId=16 LocalApicId=15 Enabled
calling mpo_policy_init for AMFI
Security policy loaded: Apple Mobile File Integrity (AMFI)
calling mpo_policy_init for Sandbox
Security policy loaded: Seatbelt sandbox policy (Sandbox)
calling mpo_policy_init for Quarantine
Security policy loaded: Quarantine policy (Quarantine)
calling mpo_policy_init for TMSafetyNet
Security policy loaded: Safety net for Time Machine (TMSafetyNet)
IOAPIC: Version 0x20 Vectors 64:111
ACPI: Executed 44 blocks of module-level executable AML code
ACPI: sleep states S3 S4 S5
pci (build 20:32:48 Nov 12 2018), flags 0x20e3000
HID: Legacyconsole relocated to 0x400000000
[ PCI configuration end, bridges 9, devices 17 ]
AppleNVMe Assert failed: ( 0 != data ) ReleaseIDNode file: /BuildRoot/Library/Caches/com.apple.xbs/Sources/IONVMeFamily/IONVMeFamily/IOVMeFamily.cpp
virtual IOReturn IONVMeController::CreateSubmissionQueue(uint16_t, uint8_t)::2777:SQ Index=0 entrysize=64
virtual IOReturn IONVMeController::CreateSubmissionQueue(uint16_t, uint8_t)::2777:SQ Index=1 entrysize=64
apfs_module_start:1334: load: com.apple.filesystems.apfs, v945.230.6, 945.230.6, 2018/11/12
SMCSuperIO: ssio @ detected device Nuvoton NCT6791D
```

Raw XML für diesen speziellen Patch sieht so aus:

```
<dict>
  <key>Comment</key>
  <string>Fix 300-series RTC Bug</string>
  <key>Disabled</key>
  <false/>
  <key>Find</key>
</dict>
```

```
oAqTU1RBUwE=  
</data>  
<key>Repl ace</key>  
<data>  
oAqRCv8L//8=  
</data>  
</dict>
```

Fixes:

Wenn wir uns dann den Abschnitt Fixes ansehen, werden wir sehen, dass wir ein paar Dinge angekreuzt haben (es gibt 2 Seiten, also habe ich 2 Screenshots eingefügt):

- FixShutdown - dies kann bei einigen Boards helfen, die es vorziehen, neu zu starten, anstatt herunterzufahren. Manchmal kann es bei anderen Boards zu Problemen beim Herunterfahren führen (ironisch, nicht wahr?), wenn Sie also Probleme beim Herunterfahren haben, wenn dies aktiviert ist, sollten Sie es deaktivieren.
- Die übrigen Korrekturen helfen, IRQ-Konflikte usw. zu vermeiden, und sind nicht dafür bekannt, dass sie Probleme verursachen. Sie sind vielleicht nicht für jede Hardware notwendig, haben aber keine negativen Auswirkungen, wenn sie angewendet werden.

Drop Tables:

Wir haben DSDT in unserem Abschnitt über Patches schon leicht angerissen - und dies ist eine kleine Erweiterung davon. SSDT ist wie ein Unterabschnitt von DSDT. Der Abschnitt "Drop Tables" erlaubt es uns, bestimmte SSDT-Tabellen beim Laden auszulassen (wie ich bereits erwähnt habe, ist DSDT für Mac und PC unterschiedlich, und macOS kann ziemlich pingelig sein). Die beiden, die ich hinzugefügt habe, sind wie folgt:

- DMAR - dies verhindert einige Probleme mit Vt-d; das ist PCI-Passthrough für VMs, und nicht sehr funktionell (wenn überhaupt?) auf Hackintoshes.
- MATS - ab High Sierra wird diese Tabelle geparkt und kann manchmal nicht druckbare Zeichen enthalten, die zu einer Kernel-Panik führen können.

FixHeaders and PluginType:

Die einzigen anderen Dinge, die wir auf dieser Seite getan haben, sind diese beiden Kontrollkästchen zu aktivieren.

- FixHeaders - dies ist nur eine Verdoppelung unseres MATS-Tabellenabwurfs. Dieses Kontrollkästchen weist Clover an, Header zu bereinigen, um Kernel-Paniks im Zusammenhang mit nicht druckbaren Zeichen zu vermeiden.

- PluginType - dies injiziert einige DSDT-Daten, um X86PlatformPlugin zu laden - was uns einen Vorsprung bei der nativen CPU-Leistungsverwaltung verschafft. Diese Einstellung funktioniert allerdings nur auf Haswell und neueren CPUs.

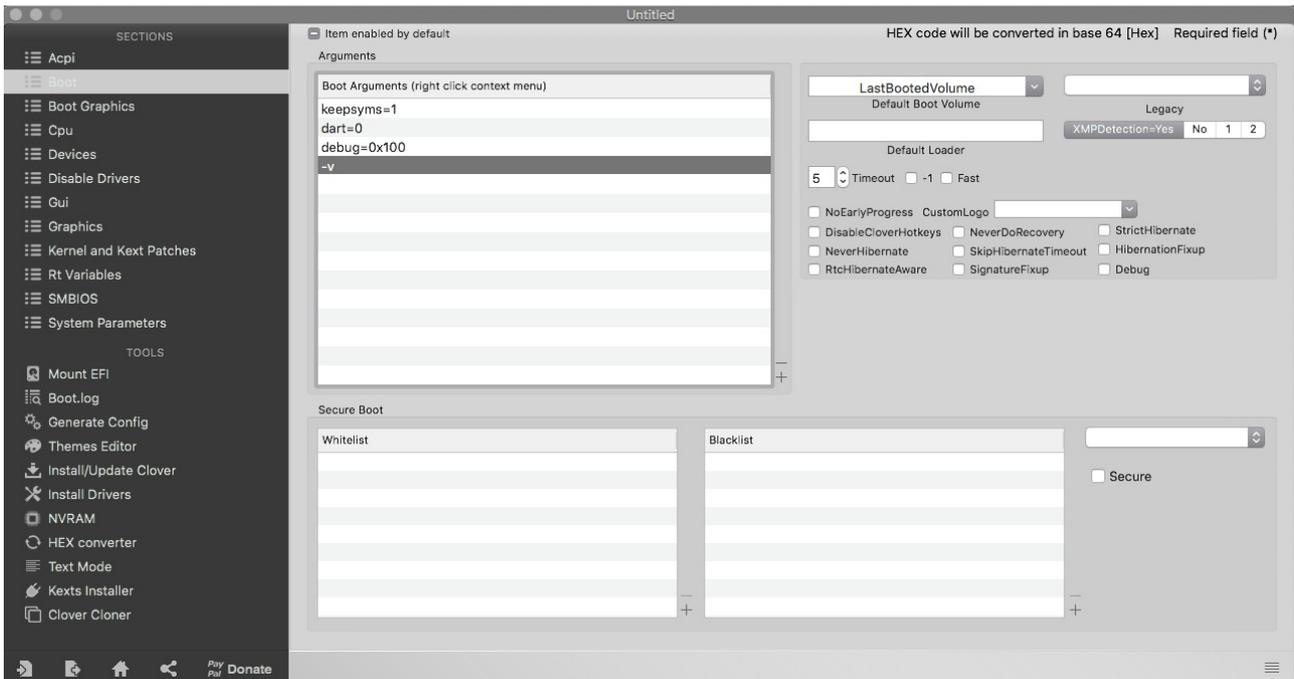
Boot

Wir brauchen hier nicht viel zu tun, aber wir werden ein paar Dinge ändern.

Raw XML

```
<key>Boot</key>
<dict>
  <key>Arguments</key>
  <string>keepsyms=1 dart=0 debug=0x100 -v</string>
  <key>DefaultVolume</key>
  <string>LastBootedVolume</string>
  <key>Timeout</key>
  <integer>5</integer>
</dict>
```

Clover Configurator Screenshots



Coffee Lake Boot CC Section

Erläuterung

Argumente:

Hier sind einige Boot-Argumente gesetzt:

-v - dies aktiviert den ausführlichen Modus, der den gesamten Text hinter den Kulissen anzeigt, der während des Bootvorgangs anstelle des Apple-Logos und des Fortschrittsbalkens vorbeiläuft. Dies ist für jeden Hackintosher von unschätzbarem Wert, da es Ihnen einen Einblick in den Boot-Prozess gibt und Ihnen helfen kann, Probleme, Problem-Kexts usw. zu identifizieren.

- dart=0 - dies ist nur ein zusätzlicher Schutz gegen Vt-d-Probleme.

- debug=0x100 - dies verhindert einen Neustart bei einer Kernel-Panik. Auf diese Weise können Sie (hoffentlich) einige nützliche Informationen sammeln und den Brotkrumen folgen, um die Probleme zu umgehen.

- keepsyms=1 - dies ist eine begleitende Einstellung zu debug=0x100, die dem Betriebssystem sagt, dass es bei einer Kernel-Panik auch die Symbole ausgeben soll. Das kann weitere hilfreiche Einblicke geben, was die Panik selbst verursacht.

DefaultBootVolume und Timeout:

Dies sind die einzigen anderen Einstellungen, die ich in diesem Abschnitt aktualisiert habe.

- DefaultBootVolume - dies verwendet NVRAM, um sich zu merken, welches Volume zuletzt von Clover gebootet wurde, und wählt dieses beim nächsten Booten automatisch aus.

- Timeout - dies ist die Anzahl der Sekunden, bevor das DefaultBootVolume automatisch gebootet wird. Sie können diesen Wert auf -1 setzen, um alle Timeouts zu vermeiden, oder auf 0, um die GUI komplett zu überspringen. Wenn Sie diesen Wert auf 0 setzen, können Sie beim Booten beliebige Tasten drücken, um die grafische Benutzeroberfläche wieder anzuzeigen, falls Probleme auftreten.

Boot Graphics

Hier steht nichts - nur die Standardeinstellungen. Sie könnten dies anpassen, wenn die Skalierung von Clover geändert werden muss, aber ich spiele damit nicht herum.

Cpu

Auch hier wird in den meisten Setups, mit denen ich gearbeitet habe, nichts verändert.

Devices

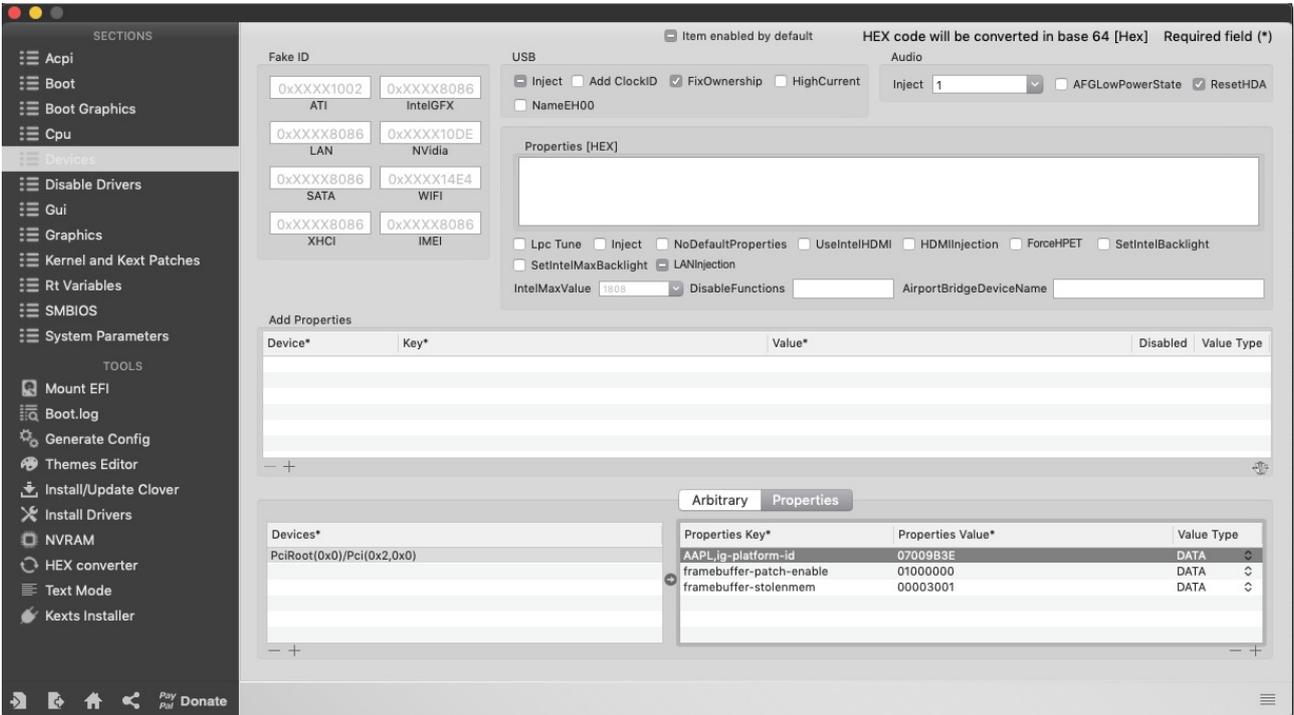
Hier werden wir einige Eigenschaften für WhateverGreen einfügen und einige grundlegende Audioeinstellungen vornehmen.

Raw XML

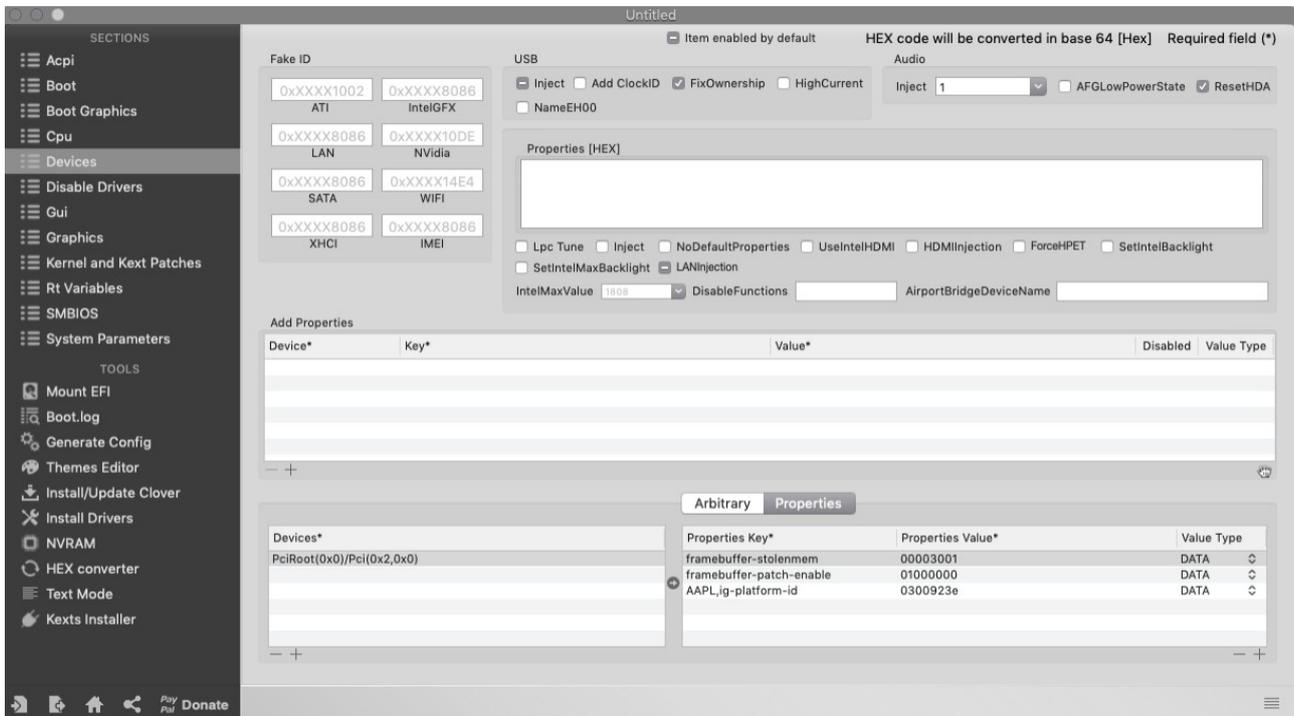
```
<key>Devices</key>
<dict>
  <key>Audio</key>
  <dict>
    <key>Inject</key>
    <integer>1</integer>
    <key>ResetHDA</key>
    <true/>
  </dict>
  <key>Properties</key>
  <dict>
    <key>PciRoot(0x0)/Pci(0x2,0x0)</key>
    <dict>
      <key>AAPL,ig-platform-id</key>
      <data>
        BwCbPg==
      </data>
      <key>framebuffer-patch-enable</key>
      <data>
        AQAAAA==
      </data>
      <key>framebuffer-stolenmem</key>
      <data>
        AAAwAQ==
      </data>
    </dict>
  </dict>
  <key>USB</key>
  <dict>
    <key>FixOwnership</key>
    <true/>
  </dict>
</dict>
```

</dict>
</dict>

Clover Configurator Screenshots



Coffee Lake Devices CC Section - iGPU



Coffee Lake Devices CC Section - iGPU Connectorless

Properties Key*	Properties Value*	Value Type
device-id	923e0000	DATA

Device-Id fake for i3-8100 UHD 630

Erläuterung

Gefälschte ID:

Dieser Abschnitt bleibt für unser Beispiel-Setup leer. In der Vergangenheit wurden fast unterstützte iGPUs (wie die HD 4400) hier zu einer unterstützten iGPU gefälscht, aber wir werden den Abschnitt "Clean Properties" dafür verwenden.

USB:

In diesem Abschnitt stellen wir sicher, dass Inject und FixOwnership ausgewählt sind, um Probleme mit dem Hängenbleiben an einer halb ausgedruckten Zeile irgendwo in der Nähe der Zeile Enabling Legacy Matching verbose zu vermeiden. Sie können das auch umgehen, indem Sie XHCI Hand Off im BIOS aktivieren.

Audio:

Hier haben wir unser Audio auf inject Layout 1 gesetzt - dies kann mit Ihrem Codec kompatibel sein oder auch nicht, aber Sie können auf der [AppleALC's Supported Codec Page](#) nachsehen.

Wir haben auch ResetHDA aktiviert, was den Codec zwischen den Neustarts des Betriebssystems in einen neutralen Zustand versetzt. Dies verhindert einige Probleme mit fehlendem Audio nach dem Booten zu einem anderen Betriebssystem und dann zurück.

Properties:

Dieser Abschnitt wird über Headkaze's [Intel Framebuffer Patching Guide](#) eingerichtet und wendet nur eine tatsächliche Eigenschaft an, nämlich die ig-platform-id. Um den richtigen Wert dafür zu erhalten, müssen wir uns die ig-platform-id ansehen, die wir verwenden wollen, und dann die Paare von Hex-Bytes vertauschen.

Wenn wir uns unsere ig-plat als 0xAABBCCDD vorstellen, würde unsere vertauschte Version wie 0xDDCCBBAA aussehen.

Die beiden ig-platform-id's, die wir verwenden, sind wie folgt:

- 0x3E9B0007 - dies wird verwendet, wenn die iGPU zur Ansteuerung eines Displays verwendet wird

- 07009B3E wenn hex-swapped
- BwCbPg== wenn die hex-swap-Version in base64 konvertiert wird

- 0x3E920003 - dies wird verwendet, wenn die iGPU nur für Rechenaufgaben verwendet wird und kein Display ansteuert

- 0300923E wenn hex-swapped
- AwCSPg== wenn die hex-swapped Version nach base64 konvertiert wird

Es ist erwähnenswert, dass Sie für 10.12 -> 10.13.5 die iGPU auf die gleichen Werte wie in der Kaby Lake-Anleitung fälschen müssten, da dies vor dem Auftauchen der nativen Coffee Lake iGPU war.

Wir fügen auch 2 weitere Eigenschaften hinzu, framebuffer-patch-enable und framebuffer-stolenmem. Die erste aktiviert das Patchen über WhateverGreen.kext, und die zweite setzt den minimalen gestohlenen Speicher auf 19 MB.

Ich habe auch einen weiteren Screenshot hinzugefügt, der eine gefälschte Geräte-ID für den UHD 630 des i3-8100 zeigt. Diese hat eine andere Geräte-ID als die UHD 630, die z.B. auf dem 8700k zu finden ist (3e918086 vs 3e928086).

Hierfür folgen wir einem ähnlichen Verfahren wie beim Hex-Swapping der ig-platform-id oben - aber dieses Mal arbeiten wir nur mit den ersten beiden Paaren von Hex-Bytes. Wenn wir uns unsere Geräte-ID als 0xAABB0000 vorstellen, würde unsere vertauschte Version wie 0xBBAA0000 aussehen. Wir machen nichts mit den letzten 2 Paaren von Hex-Bytes.

Die gefälschte Geräte-ID ist wie folgt aufgebaut:

- 0x3e920000 - das ist die Geräte-ID für die UHD 630, die auf einem 8700k zu finden ist
 - 923e0000 bei Hex-Swap
 - kj4AAA== wenn die hex-swapped Version in base64 konvertiert wird

Wenn Sie die rohe xml verwenden, würden Ihre Eigenschaften wie folgt aussehen (stellen Sie sicher, dass Sie immer noch die passende ig-platform-id für Ihr Setup verwenden):

```
<key>Properties</key>
<dict>
  <key>PciRoot(0x0)/Pci(0x2,0x0)</key>
  <dict>
    <key>device-id</key>
    <data>
      kj4AAA==
    </data>
    <key>AAPL,ig-platform-id</key>
    <data>
      BwCbPg==
    </data>
    <key>framebuffer-patch-enable</key>
    <data>
      AQAAAA==
    </data>
    <key>framebuffer-stolenmem</key>
    <data>
      AAAwAQ==
    </data>
  </dict>
</dict>
```

Rosa/violetter Farbstich?

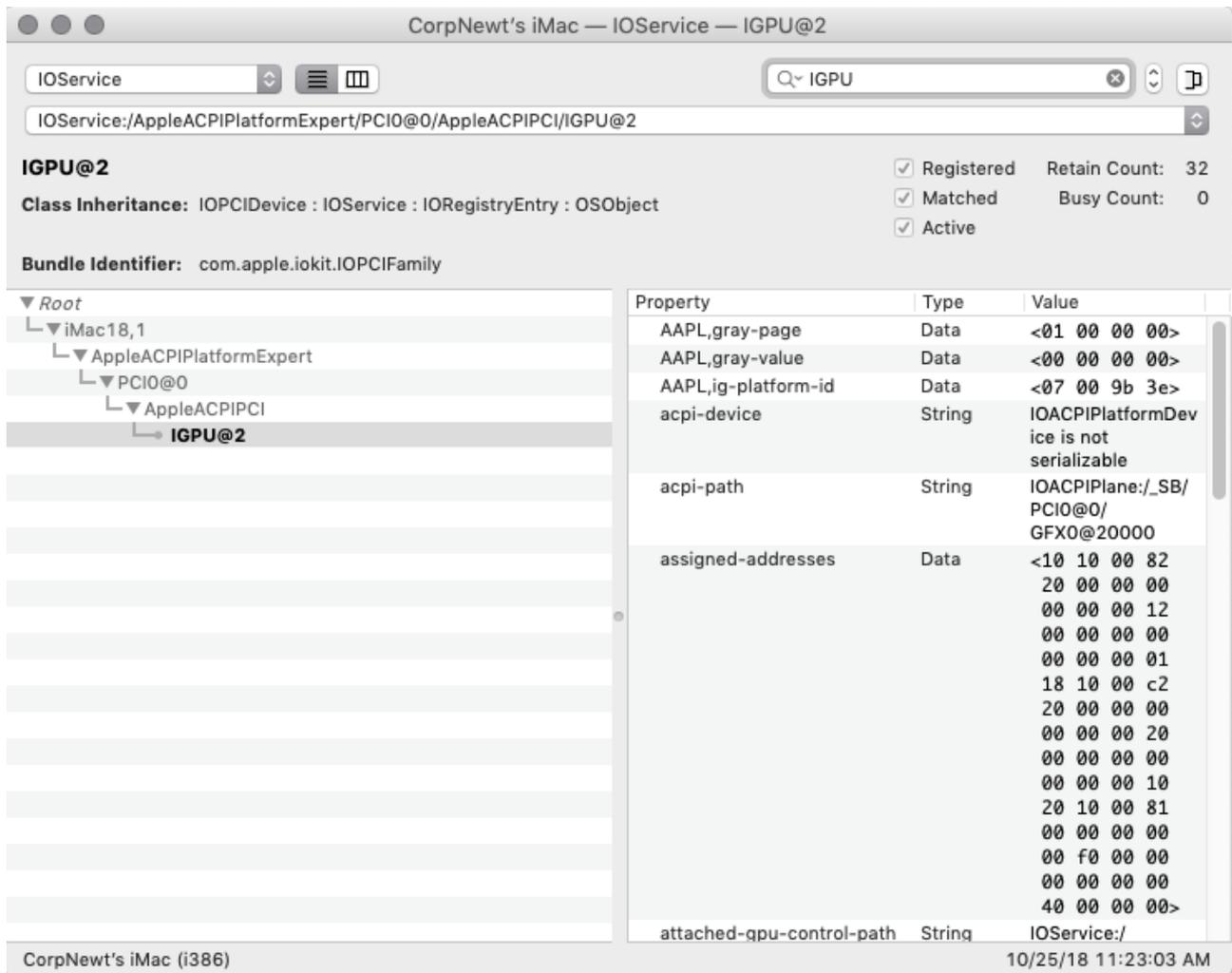
Ich habe ein paar Benutzer gesehen, die über einen rosa Farbton berichten, wenn sie HDMI mit der UHD 630 iGPU verwenden. Ich habe einige Experimente auf meinem eigenen System durchgeführt und konnte das Problem auf verschiedene Arten umgehen.

RGB erzwingen

Ich habe das Problem in einem [a reddit post](#), gesehen, und die Lösung war, einen benutzerdefinierten Override für Ihr Display anzuwenden, um es zu zwingen, RGB anstelle von YCbCr zu verwenden, wie [here](#). beschrieben. Ich habe ein [a script](#) geschrieben, das diese Methode umgeht und den Override automatisch anwendet, wenn es fertig ist. Das hat bei mir gut funktioniert, aber es fühlte sich nicht wie eine echte Lösung an. Was mich dazu brachte, etwas tiefer einzutauchen...

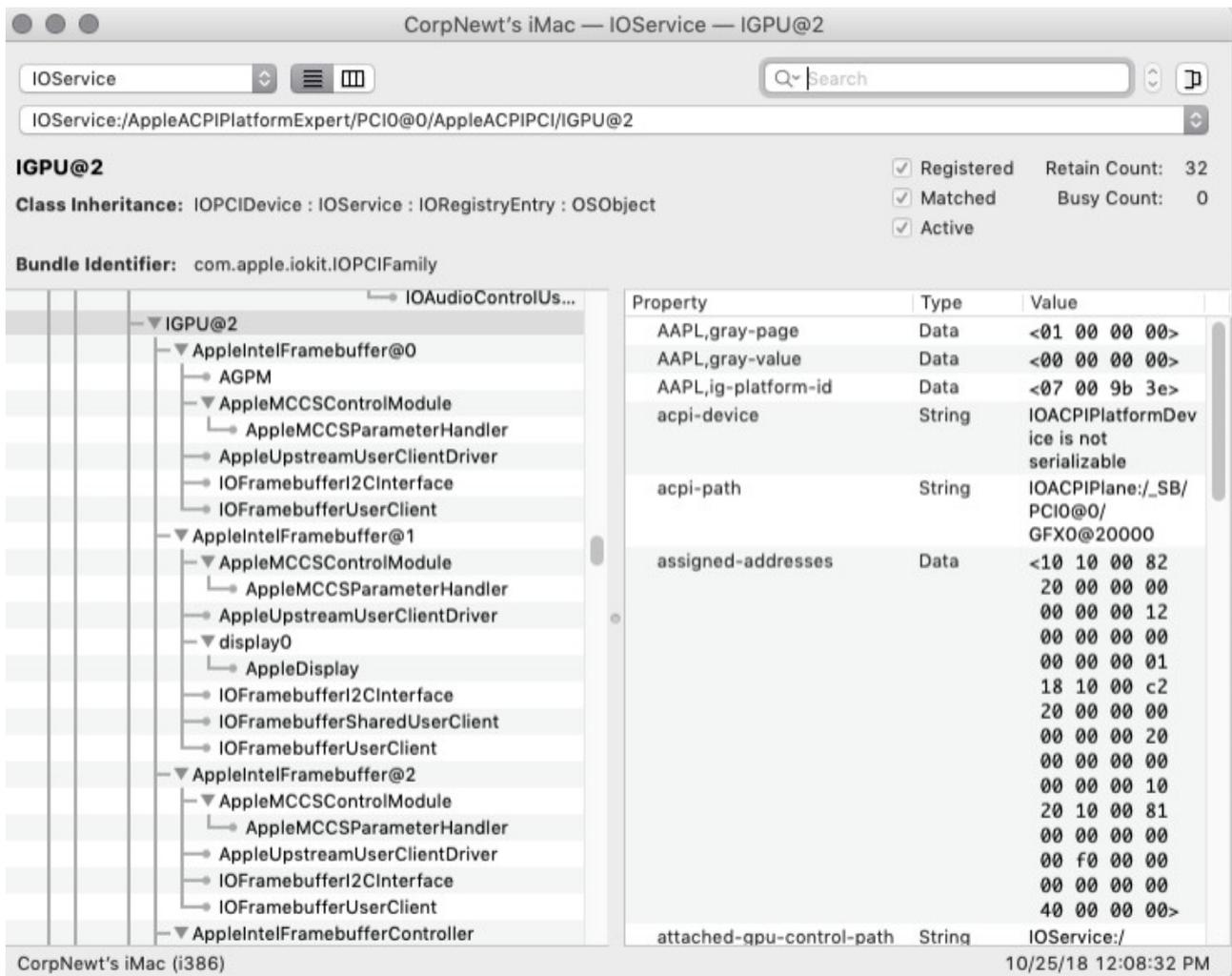
Konnektortypen in IOReg

Ich öffnete den IORegistryExplorer und gab in der Suchleiste IGPU ein (dies wird in ACPI manchmal als GFX0 bezeichnet, aber Lilu + WhateverGreen sollten es richtig umbenennen) und erhielt den folgenden Bildschirm:



Search for IGPU in IOReg

Sobald wir IGPU in IOReg gefunden haben, können wir unsere Suche löschen - dadurch werden alle Informationen rund um den IGPU-Bereich angezeigt, während wir unseren Platz behalten:



IGPU Selected With Search Cleared

Wie Sie im obigen Screenshot sehen können, hatte ich ein paar verschiedene AppleIntelFramebuffer-Verbindungen aufgelistet. Ich suche diejenige, die speziell mein Display ansteuert - die die Eigenschaft AppleDisplay hat. In meinem Fall war das AppleIntelFramebuffer@1. Wenn Sie das im linken Fenster ausgewählt haben, finden Sie die Eigenschaft connector-type, die in meinem Fall ursprünglich auf <00 04 00 00> gesetzt war. Der Anschlusstyp kann ein paar verschiedene Werte haben:

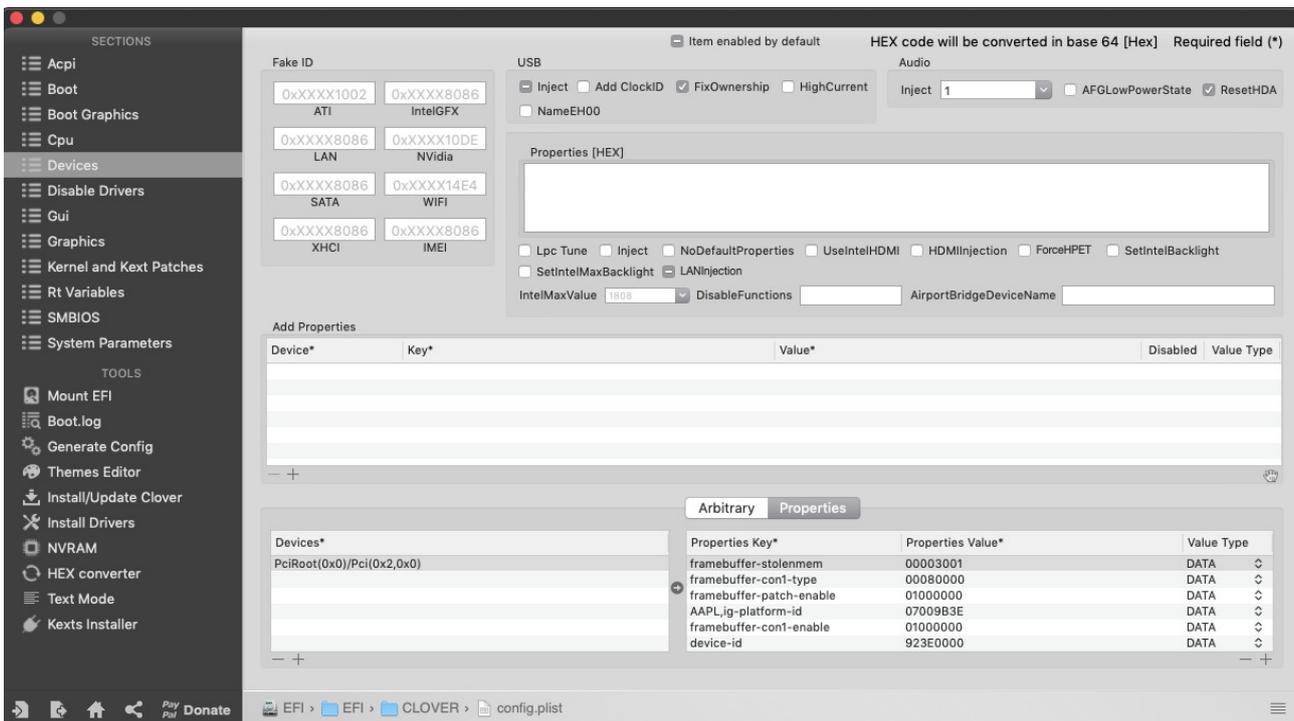
- <00 04 00 00> - this is DisplayPort
- <00 08 00 00> - this is HDMI
- <04 00 00 00> - this is Digital DVI
- <02 00 00 00> - this is LVDS (for laptops)
- <01 00 00 00> - this is just a Dummy port

Was mir in meinem Fall auffiel, war, dass mein HDMI-Anschluss als DisplayPort aufgeführt war - also konnte ich die Patching-Fähigkeiten von WhateverGreen nutzen, um den Anschlusstyp zu ändern.

Da sich mein falscher Anschluss unter AppleIntelFramebuffer@1 befand, ist dies Anschluss 1. Ich musste den Port-Patch in den Eigenschaften aktivieren und dann den Anschlusstyp auf HDMI setzen. Dazu habe ich die folgenden Properties-Einträge verwendet:

- framebuffer-conX-enable = 01000000
- framebuffer-conX-type = 00080000

Ich habe das conX in beiden Patches durch con1 ersetzt, um den Port, den ich ändere, wiederzugeben, und dann die Werte wie oben aufgeführt eingestellt.



```

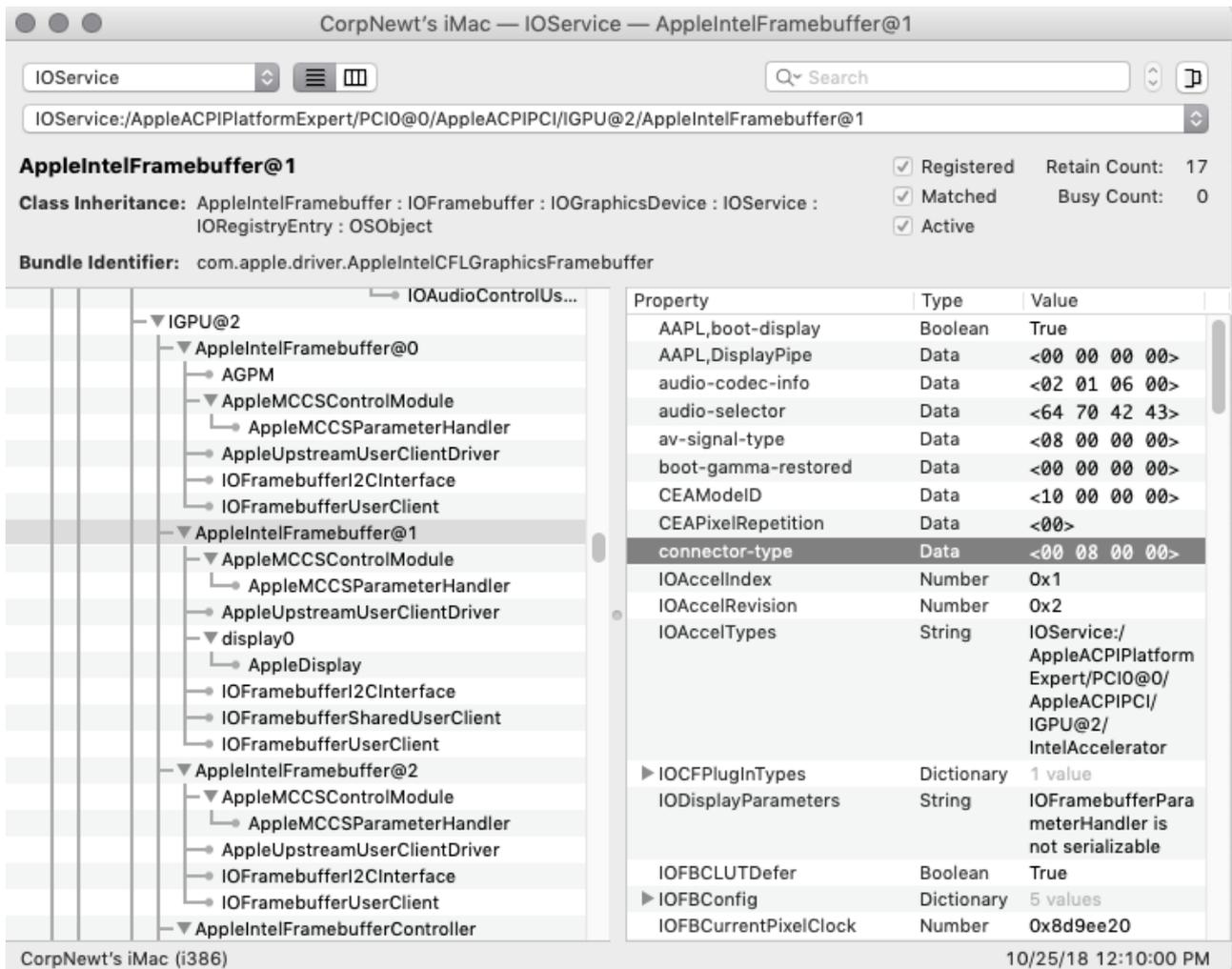
<key>Properties</key>
<dict>
  <key>PciRoot(0x0)/Pci(0x2,0x0)</key>
  <dict>
    <key>AAPL,ig-platform-id</key>
    <data>
      BwCbPg==
    </data>
    <key>device-id</key>
    <data>
      kj4AAA==
    </data>
    <key>framebuffer-con1-enable</key>
    <data>
      AQAAAA==
    </data>
    <key>framebuffer-con1-type</key>
    <data>
      AAgAAA==
    </data>
    <key>framebuffer-patch-enable</key>
    <data>

```

```

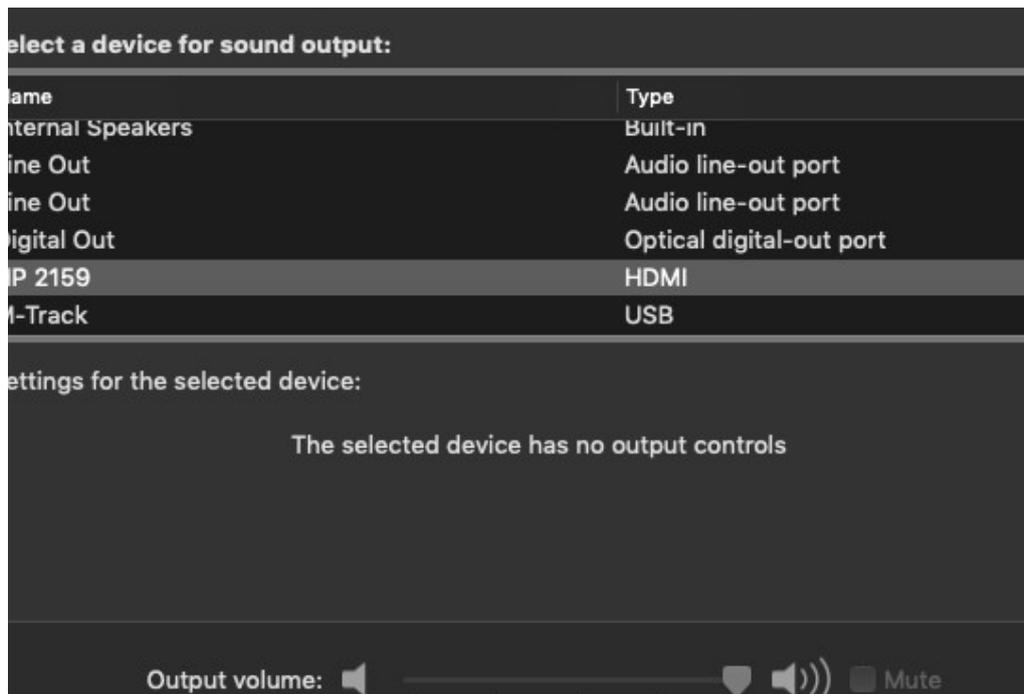
AQAAAA==
</data>
<key>framebuffer-stolenmem</key>
<data>
AAAQAQ==
</data>
</dict>
</dict>

```



IOReg -> IGPU -> AppleIntelFramebuffer@1 After Patching

Dadurch wurde auch HDMI-Audio für mich aktiviert.



Disable Drivers

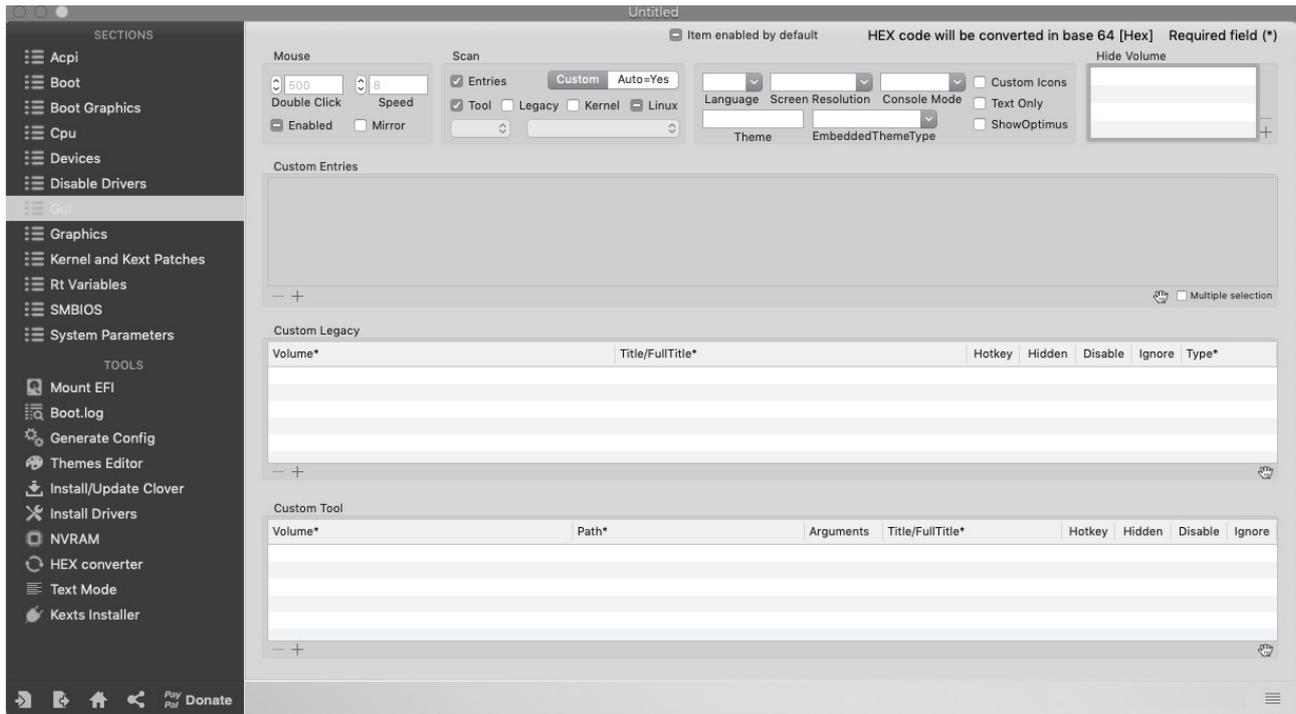
We have nothing to do here.

Gui

Raw XML

```
<key>GUI</key>
<dict>
  <key>Scan</key>
  <dict>
    <key>Entries</key>
    <true/>
    <key>Tool</key>
    <true/>
  </dict>
</dict>
```

Clover Configurator Screenshots



Coffee Lake Gui CC Section

Erläuterung

Scannen:

Die einzigen Einstellungen, die ich auf dieser Seite geändert habe, sind die Scan-Einstellungen. Ich habe "Benutzerdefiniert" ausgewählt und dann alles außer "Legacy" und "Kernel" angekreuzt. Damit werden nur einige der nicht bootfähigen Einträge in Clover weggelassen, um das Menü aufzuräumen.

Volumes ausblenden:

Ich habe hier nichts hinzugefügt, aber Sie können unerwünschte Volumes hier ausblenden. Sie können dies tun, indem Sie entweder den Namen des Volumes oder die UUID hinzufügen.

Um zusätzliche APFS-Einträge auszublenden, fügen Sie Folgendes zu dieser Liste hinzu:

- Preboot
- VM

Um alle Recovery-Partitionen auszublenden, fügen Sie Recovery zu dieser Liste hinzu.

Um die UUID eines zu versteckenden Laufwerks zu erhalten, können Sie den folgenden Terminalbefehl verwenden:

```
diskutil info diskXsY | grep -i "Partition UUID" | rev | cut -d' ' -f 1 | rev
```

Stellen Sie sicher, dass Sie diskXsY durch die tatsächliche Festplattennummer des Volumes ersetzen, das Sie ausblenden möchten.

Thema:

Wenn Sie ein neues Thema ausprobieren möchten (ich empfehle Ihnen clover-next-black), können Sie den entpackten Themenordner zum Verzeichnis /Volumes/EFI/EFI/CLOVER/themes hinzufügen und dann den Namen des Ordners in das Textfeld Thema eingeben, um ihn anzuwenden.

Graphics

In der Vergangenheit haben wir hier die iGPU eingerichtet, aber da wir dies bereits über Eigenschaften im Abschnitt Geräte getan haben, müssen wir hier nichts mehr konfigurieren. HINWEIS: Wenn Clover eine Intel iGPU erkennt, aktiviert es automatisch Intel Injection, wenn der Abschnitt Graphics in der config.plist nicht vorhanden ist. Um dies zu umgehen, können Sie die Injektion explizit deaktivieren, indem Sie das unten stehende Roh-XML verwenden, oder indem Sie die Schaltfläche "Inject Intel" einmal anklicken, um sie zu aktivieren, und einmal, um sie in CC zu deaktivieren.

Raw XML

```
<key>Graphics</key>
<dict>
  <key>Inject</key>
  <false/>
</dict>
```

Kernel And Kext Patches

Raw XML

```
<key>KernelAndKextPatches</key>
<dict>
  <key>KernelPm</key>
  <true/>
  <key>KextsToPatch</key>
  <array>
    <dict>
      <key>Comment</key>
      <string>Port limit increase</string>
      <key>Disabled</key>
      <false/>
      <key>Find</key>
      <data>
        g710///EA==
      </data>
      <key>InfoPlistPatch</key>
      <false/>
      <key>MatchOS</key>
      <string>10.12.x</string>
      <key>Name</key>
      <string>com.apple.driver.usb.AppleUSBXHCI</string>
      <key>Replace</key>
      <data>
        g710///Gw==
      </data>
    </dict>
  </array>
  <key>Comment</key>
</dict>
```

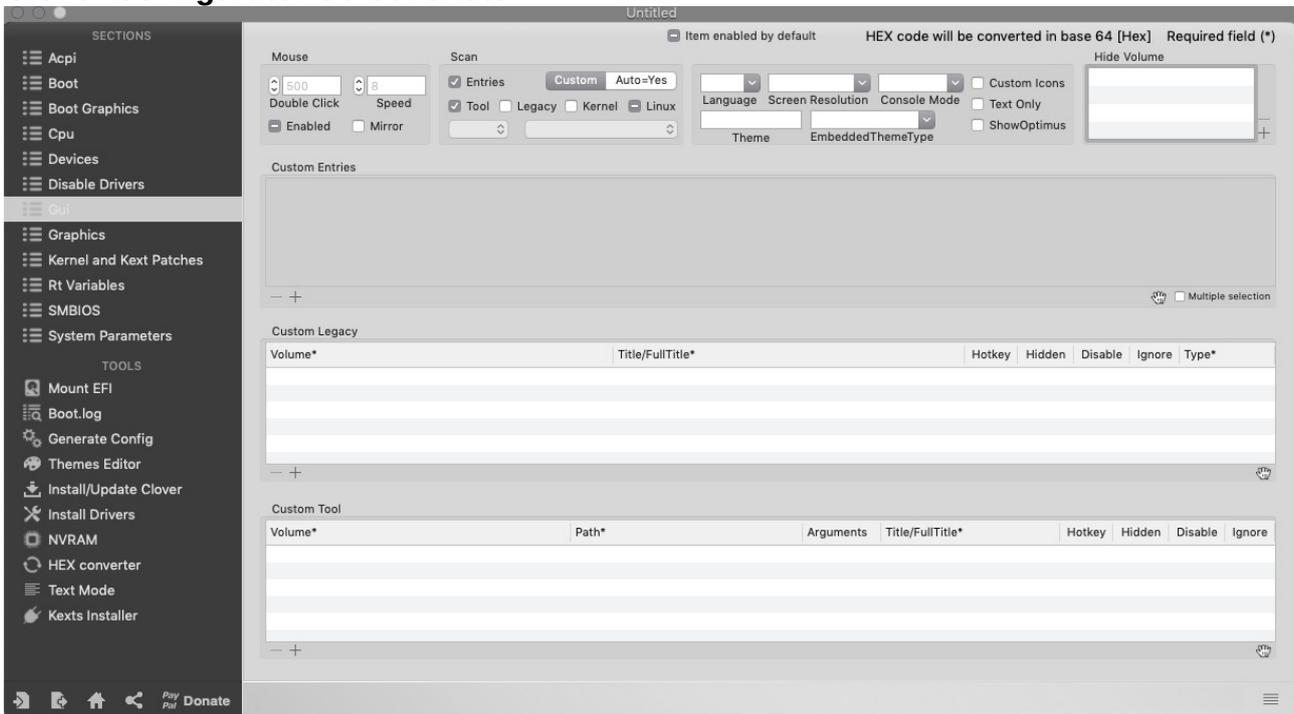
```
<string>Port limit increase (RehabMan)</string>
<key>Disabled</key>
<false/>
<key>Find</key>
<data>
g32IDw+DpwQAAA==
</data>
<key>InfoPlistPatch</key>
<false/>
<key>MatchOS</key>
<string>10.13.x</string>
<key>Name</key>
<string>com.apple.driver.usb.AppleUSBXHCI</string>
<key>Replace</key>
<data>
g32ID5CQkJCQkA==
</data>
</dict>
<dict>
<key>Comment</key>
<string>Port limit increase (PMHeart)</string>
<key>Disabled</key>
<true/>
<key>Find</key>
<data>
g/SPD4MDBQAA
</data>
<key>InfoPlistPatch</key>
<false/>
<key>MatchOS</key>
<string>10.14.0</string>
<key>Name</key>
<string>com.apple.driver.usb.AppleUSBXHCI</string>
<key>Replace</key>
<data>
g/sPkJCQkJCQ
</data>
</dict>
<dict>
<key>Comment</key>
<string>Port limit increase (Ricky)</string>
<key>Disabled</key>
<false/>
<key>Find</key>
<data>
g/SPD40PBAAA
</data>
<key>InfoPlistPatch</key>
<false/>
<key>MatchOS</key>
<string>10.14.x</string>
<key>Name</key>
<string>com.apple.driver.usb.AppleUSBXHCI</string>
<key>Replace</key>
<data>
g/sPkJCQkJCQ
</data>
</dict>
<dict>
<key>Comment</key>
<string>External Icons Patch</string>
<key>Disabled</key>
<false/>
<key>Find</key>
```

```

<data>
RXh0ZXJuYWw=
</data>
<key>InfoPlistPatch</key>
<false/>
<key>Name</key>
<string>AppleAHCIPort</string>
<key>Replace</key>
<data>
SW50ZXJuYWw=
</data>
</dict>
</array>
</dict>

```

Clover Configurator Screenshots



Coffee Lake KernelAndKextPatches CC Section

Erläuterung

In diesem Abschnitt haben wir ein paar Einstellungen aktiviert und einige Kext-Patches hinzugefügt.

Checkboxes:

Wir haben hier ein paar Checkboxes aktiviert:

- Apple RTC - dies stellt sicher, dass wir keinen BIOS-Reset beim Neustart haben.
- KernelPM - diese Einstellung verhindert das Schreiben auf MSR 0xe2, was eine Kernel-Panik beim Booten verhindern kann.

KextsToPatch:

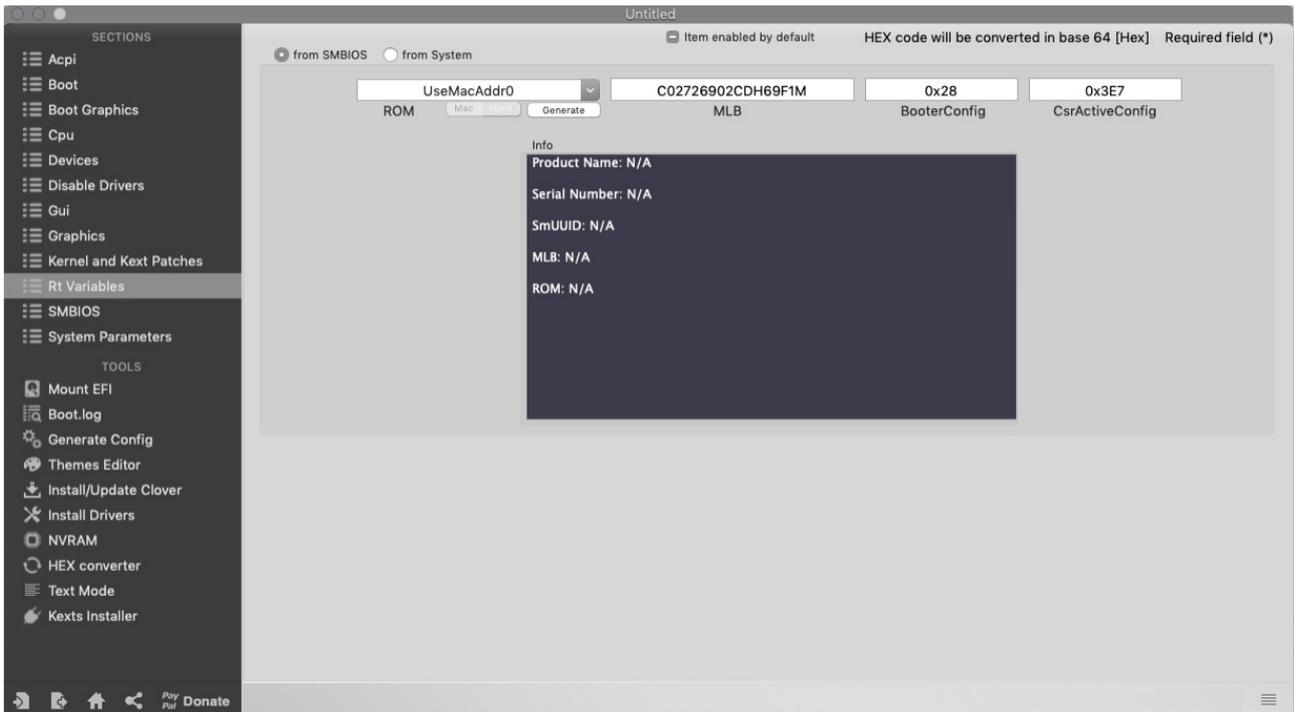
Wir haben hier 4 verschiedene Kexts zum Patchen hinzugefügt. Drei davon sind für die Erhöhung des USB-Port-Limits und der letzte dient als Fix für orangefarbene Icons - wenn interne Laufwerke hotplug-fähig sind und als externe Laufwerke behandelt werden. Sie werden feststellen, dass für jeden der USB-Port-Limit-Patches MatchOS-Werte gesetzt sind. Sie können alle Einträge für Betriebssystemversionen, die Sie nicht verwenden wollen, entfernen. Sie schaden nicht, wenn sie da sind, aber wenn Sie eine saubere, minimale Liste wollen, ist es nicht sinnvoll, sie zu haben.

RtVariablen und SMBIOS

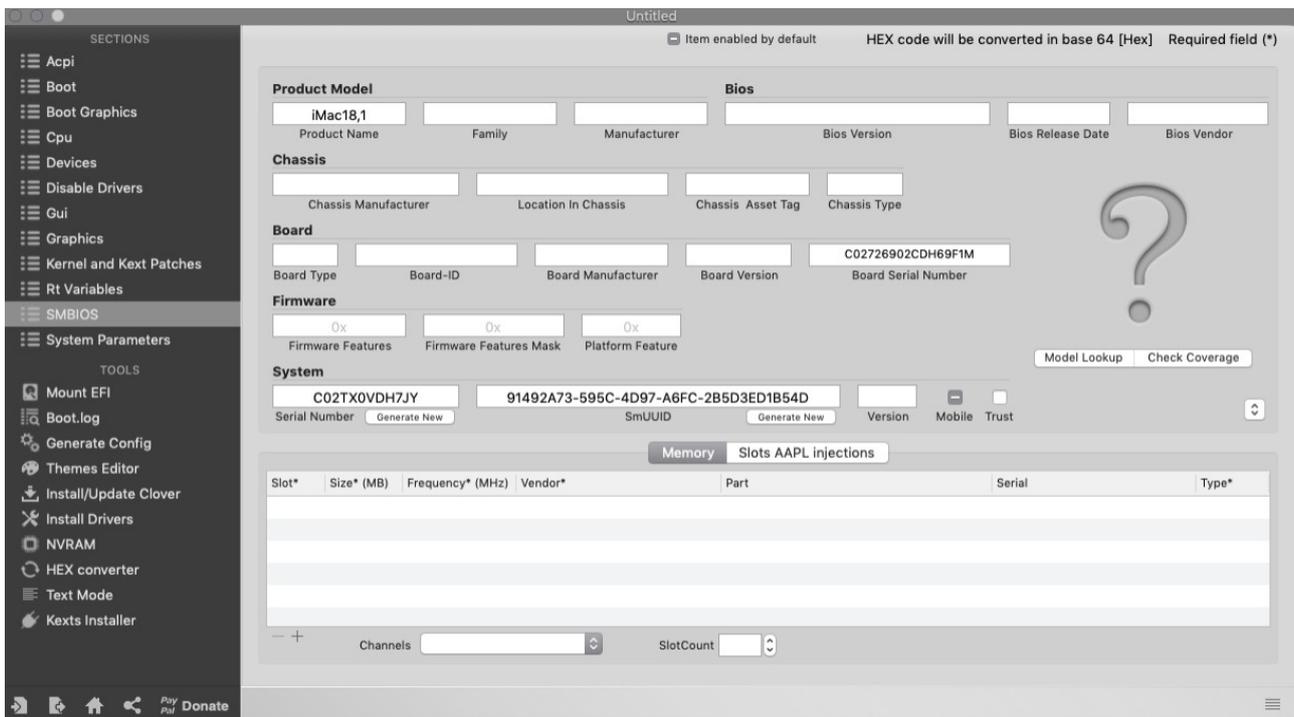
Raw XML

```
<key>RtVariables</key>
<dict>
  <key>BoooterConfig</key>
  <string>0x28</string>
  <key>CsrActiveConfig</key>
  <string>0x3E7</string>
  <key>MLB</key>
  <string>C02726902CDH69F1M</string>
  <key>ROM</key>
  <string>UseMacAddr0</string>
</dict>
<key>SMBIOS</key>
<dict>
  <key>BoardSerialNumber</key>
  <string>C02726902CDH69F1M</string>
  <key>ProductName</key>
  <string>iMac18,1</string>
  <key>SerialNumber</key>
  <string>C02TX0VDH7JY</string>
  <key>SmUUID</key>
  <string>91492A73-595C-4D97-A6FC-2B5D3ED1B54D</string>
</dict>
```

Clover Configurator Screenshots



Coffee Lake RtVariables CC Section



Coffee Lake SMBIOS CC Section

Erläuterung

Zum Einrichten der SMBIOS-Informationen verwende ich die [macserial](#) von acidanthera. Ich habe ein [python script](#) geschrieben, das es ebenfalls nutzen kann (und automatisch in der config.plist speichert, wenn es ausgewählt wird). Es gibt viele Informationen, die leer

gelassen werden, damit Clover die Leerstellen ausfüllen kann; das bedeutet, dass eine Aktualisierung von Clover die übergebenen Informationen aktualisiert und Sie nicht auch Ihre config.plist aktualisieren müssen.

Für dieses Coffee Lake-Beispiel habe ich das iMac18,1 SMBIOS gewählt - dies geschieht absichtlich aus Gründen der Kompatibilität. Es gibt zwei Haupt-SMBIOS, die für Coffee Lake verwendet werden:

- iMac18,1 - dies wird für Computer verwendet, die die iGPU für die Anzeige nutzen.
- iMac18,3 - dies wird für Computer verwendet, die eine dGPU für die Anzeige und eine iGPU nur für Rechenaufgaben verwenden.

Um die SMBIOS-Informationen zu erhalten, die mit macserial generiert wurden, können Sie es mit dem Argument -a ausführen (das Seriennummern und Board-Seriennummern für alle unterstützten Plattformen generiert). Sie können es auch mit grep parsen, um Ihre Suche auf einen SMBIOS-Typ zu beschränken.

Mit unserem iMac18,1-Beispiel würden wir macserial wie folgt über das Terminal ausführen:

```
macserial -a | grep -i iMac18,1
```

Das würde uns eine Ausgabe ähnlich der folgenden liefern:

Product	Serial	Board Serial (MLB)
iMac18,1	C02T8SZNH7JY	C02707101J9H69F1F
iMac18,1	C02VXBYDH7JY	C02753100GUH69FCB
iMac18,1	C02T7RY6H7JY	C02706310GUH69FA8
iMac18,1	C02VD07ZH7JY	C02737301J9H69FCB
iMac18,1	C02TQPYPH7JY	C02720802CDH69FAD
iMac18,1	C02VXYVH7JY	C02753207CDH69FJC
iMac18,1	C02VDBZ0H7JY	C02737700QXH69FA8
iMac18,1	C02VP0H6H7JY	C02746300CDH69FJA
iMac18,1	C02VLOW9H7JY	C02743303CDH69F8C
iMac18,1	C02V2NYMH7JY	C02728600J9H69FAD

Der Teil iMac18,1 wird nach SMBIOS -> Produktname kopiert.

Der Teil "Serial" wird nach SMBIOS -> Serial Number kopiert.

Der Teil "Board Serial" wird zu SMBIOS -> Board Serial Number sowie zu Rt Variables -> MLB kopiert.

Wir können eine SmUUID erstellen, indem wir uuidgen im Terminal ausführen (oder sie wird automatisch über mein GenSMBIOS-Skript generiert) - und diese wird nach SMBIOS -> SmUUID kopiert.

Wir setzen Rt Variables -> ROM auf UseMacAddr0, was einfach unsere Onboard-Mac-Adresse verwendet - diese sollte eindeutig genug sein, um nicht mit anderen in Konflikt zu geraten.

BooterConfig wird auf 0x28 gesetzt, und CsrActiveConfig wird auf 0x3e7 gesetzt, was SIP effektiv deaktiviert. Sie können eine Reihe von anderen Optionen wählen, um Abschnitte von SIP zu aktivieren/deaktivieren. Einige gängige sind wie folgt:

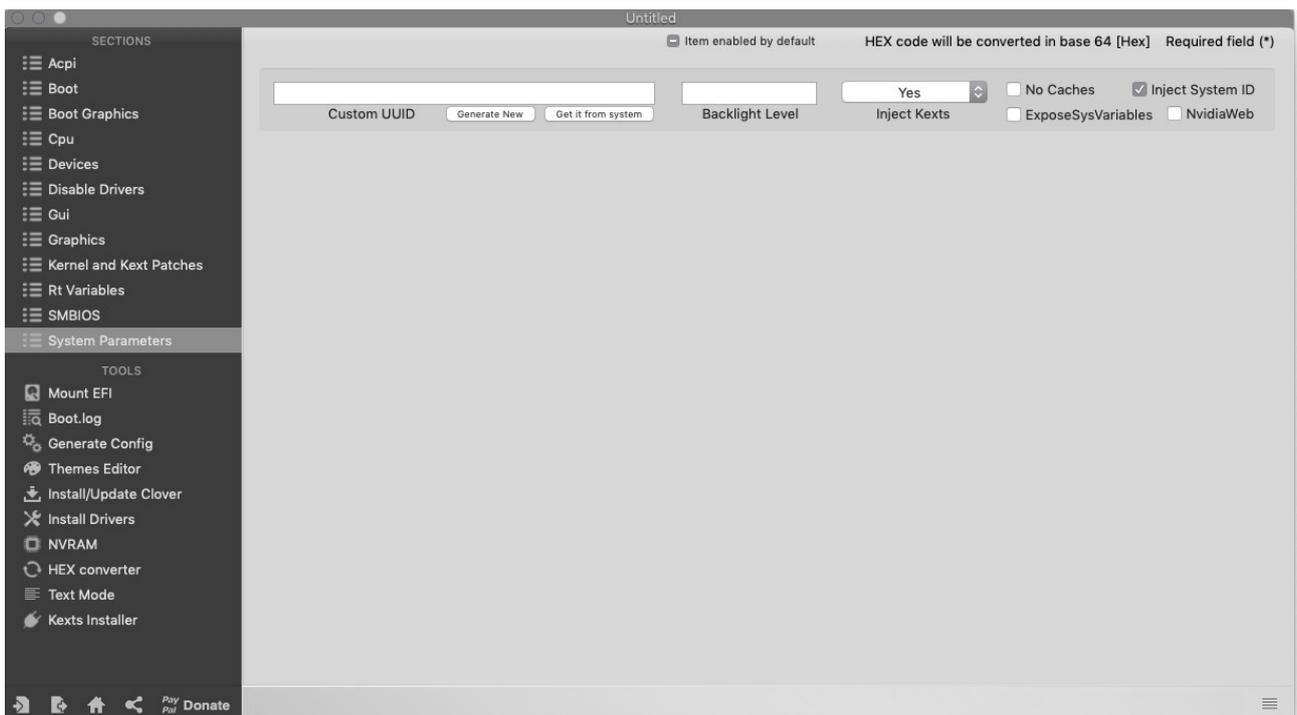
- 0x0 - SIP vollständig aktiviert
- 0x3 - Erlaubt unsignierte Kexts und das Schreiben auf geschützte fs-Speicherplätze
- 0x3e7 - SIP komplett deaktiviert

System Parameters

Raw XML

```
<key>SystemParameters</key>
<dict>
  <key>InjectKexts</key>
  <string>Yes</string>
  <key>InjectSystemID</key>
  <true/>
</dict>
```

Clover Configurator Screenshots



System Parameters CC Section

Erläuterung

Inject Kexts:

Diese Einstellung hat 3 Modi:

- Ja - damit wird Clover angewiesen, Kexts aus der EFI unabhängig davon zu injizieren.
- Nein - damit wird Clover angewiesen, keine Kexts von der EFI zu injizieren.
- Detect - Clover injiziert Kexts nur, wenn FakeSMC.kext nicht im Kext-Cache ist.

Wir setzen es auf Yes, um sicherzustellen, dass alle Kexts, die wir zuvor hinzugefügt haben, richtig injiziert werden.

InjectSystemID

Diese Einstellung weist Clover an, die SmUUID als System-ID beim Booten zu setzen - was wichtig für iMessage und so ist.

Saving

An dieser Stelle können Sie mit File -> Save die config.plist speichern. Wenn Sie Probleme haben, die Datei direkt im EFI zu speichern, können Sie sie auch auf dem Desktop speichern und dann einfach rüberkopieren. Ich lasse die [sample config.plist here](#) auch hier.