

iMessage & Co für den Hackintosh

Beitrag von „MacGrummel“ vom 20. Oktober 2016, 23:10



iMessage & Co für den Hackintosh

Von Grund auf

Du hast es geschafft, Dein neuer Hackintosh läuft, die Hardware wird komplett erkannt und arbeitet, von Lan über Bluetooth bis WLAN, der Sound ist auch da.

Und jetzt möchtest Du auch die Apple-Extras nutzen können: das Musik- und Film-Programm iTunes sowieso, Nachrichten mit iMessages, Bild-Chats und Telefonate mit FaceTime, AirPlay für Film-Übertragung, AirDrop, um kurz ein paar Daten vom iPhone rüber zu schieben, den Apple AppStore, Siri..



All diese Dienste (außer iTunes für Windows) laufen normal nur über einen modernen Apple-Rechner und oft auch direkt über Apples Server. Also müssen wir Apple vormachen, dass unsere Selbstbau-Rechner eigentlich aus Cupertino stammen. Dafür brauchen wir eine Vielzahl von jeweils einzig-artigen Nummern. Eine Serien-Nummer leuchtet ja sicher noch jedem ein, die gibt es dann nicht nur für das Gehäuse, sondern auch noch für das Board, die CPU usw., dann eine Kennung für das Lan-Netzwerk, das natürlich auch für WLAN, FireWire und Bluetooth, eine einzigartige universelle Rechner-Kennung, und so weiter.

Einige dieser Daten bringt unser Rechner von vornherein mit, die anderen müssen wir eben so erstellen, dass Apple nicht gleich merkt, dass wir mit einem Custom-Made-Rechner auf die Dienste zugreifen.

Was immer vorhanden ist sind die Kennungen der verschiedenen Netzwerk-Karten, deren MAC-Adresse. Sie besteht aus 6 oder 8 Blöcken á 2 Zeichen wie 40:6c:8f:c3:94:8e als Beispiel. Die bringen diese Karten jeweils mit und können nur schwer verändert werden.

Power Mac				
* Hardware	Active Dienste	Typ	Hardware	SSD-GeräteName
ATA	Bluetooth PAN	Ethernet	Ethernet	en0
Audio	Ethernet	Ethernet	Ethernet	en0
Bluetooth	FireWire	FireWire	FireWire	fw0
Brennen von Medien	PCI FireWire	FireWire	FireWire	fw1
Diagnose	Thunderbolt FireWire Slot 1	FireWire	FireWire	fw2
Drucker	Wi-Fi	AirPort	AirPort	en1
Ethernet-Karten				
Festplatte				
Fire-Channel				
FireWire				
Grafik/Displays				
Hardware-RAM				
Kamera				
Kartenleser				
NVMe/Express				
PCI				
Parallel-SCSI				
SAS				
SATA/SATA Express				
SPR				
Speicher				
Stromversorgung				
Thunderbolt				
USB				
* Software				

Wichtig ist für uns hier erstmal nur, dass die interne Ethernet-Netzwerk-Karte unter der internen Gerätenummer en0 läuft. Wenn das nicht so ist, funktioniert die ganze weitere Anmeldung nicht..

Zum Nachsehen: Apfelmenü oben links/Über diesen Mac/Systembericht/Netzwerk

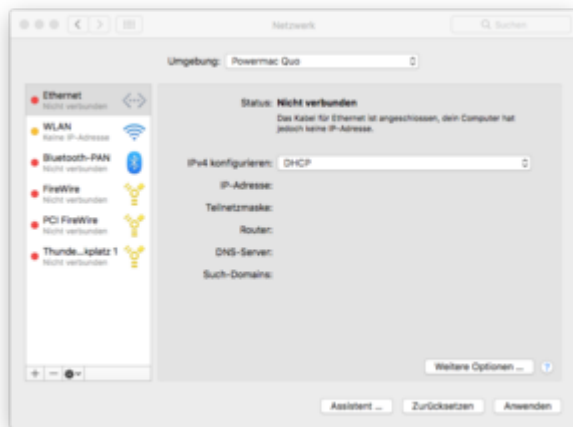
Also Kapitel 1: Ethernet-Netzwerk auf EN0 setzen

Kurzanleitung:

Deine Netzwerkkarte wird nicht als EN0 und Build in erkannt.



Der Fehler ist so alt wie der AppStore selbst und kommt zB. auch gelegentlich vor, wenn man bei einem bereits laufenden Rechner eine neue Systemplatte aufsetzt:



Lösch jetzt mal in den Systemeinstellungen -> Netzwerk alle Adapter und anschließend auch die Dateien /Library/Preferences/SystemConfiguration/NetworkInterfaces.plist und /Library/Preferences/SystemConfiguration/preferences.plist und starte den Rechner anschließend neu. Dann fügst Du als erstes den Ethernet Port wieder hinzu sofern noch nicht geschehen und das gute Stück sollte en0 sein.

Wenn wir richtig schön sauber und Schritt für Schritt unseren Rechner bearbeitet haben, ist dieses Kapitel nicht notwendig. Aber das kommt nur sehr selten vor, irgendwas hakt immer.

Für das weitere Vorgehen brauchen wir ein Paar kleine Tools:

[Clover Configurator](#)

[iMessageDebug.zip](#)

[ShowAllFiles.zip](#)

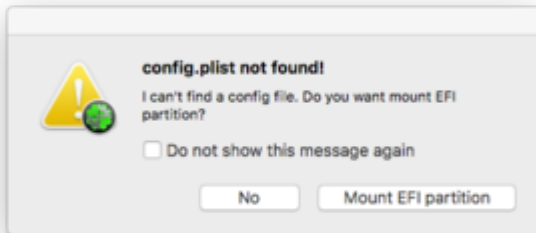
Ein paar Bordmittel wie Terminal und Text

Bei Ozmosis-Rechnern eine [rohe defaults.plist](#)

Und Zeit und Geduld

Für jedes weitere Bearbeiten möchte ich genau sehen, was ich mache. Also erstmal mit der kleinen Protokoll-App ShowAllFiles (ich packe das in die Dienstprogramme und ziehe es als Alias in die Dock-Zeile) genau was der Name sagt.

Dann brauchen wir für fast alles Weitere die sonst immer noch versteckte Startplatten-EFI-Partition und mounten die mit dem Configurator.



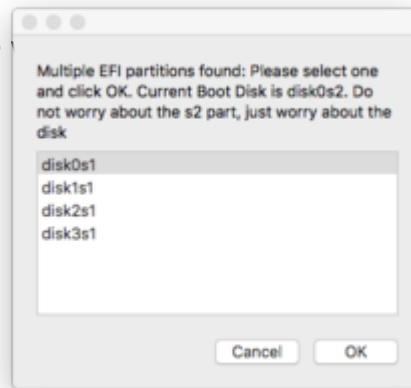
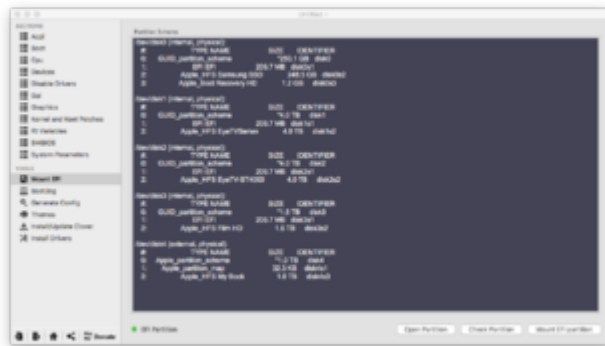
Dann werfe ich iMessageDebug an und sehe erstmal nach, was wir schon haben. Was hier auftaucht, sind meist die vorgegebenen Standard-Werte. Und es wäre doch peinlich, wenn unser neuer Rechner schon lange bei Apple angemeldet ist oder sogar gesperrt.



So ziemlich die gleichen Werte sollten dann auch auftauchen, wenn wir im Configurator die Seite Rt Variables öffnen und da auf Calculate gehen. Da wird nichts neu berechnet, sondern nur exakt gezeigt, was schon da ist.

Aber weiter: „Mount EFI“ unter Tools öffnen, kontrolliere die Partitionen und wähle die richtige aus, meist disk0s1.

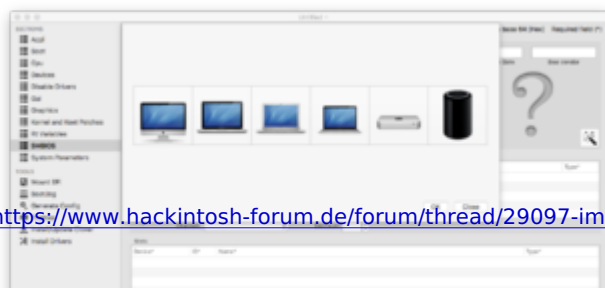
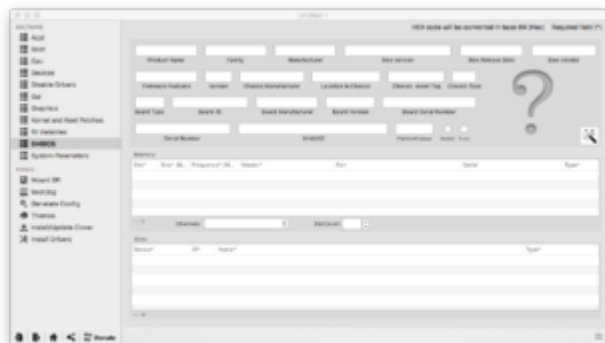
Da sollten jetzt unsere wichtigen Daten lagern, die



Wenn wir jetzt eine Clover-Installation vor uns haben, können wir recht leicht die vorhandenen Daten verändern: wir öffnen unsere Konfiguration (das ist von den Listen, die unten auftauchen, die Config.plist mit dem kürzesten Weg..)

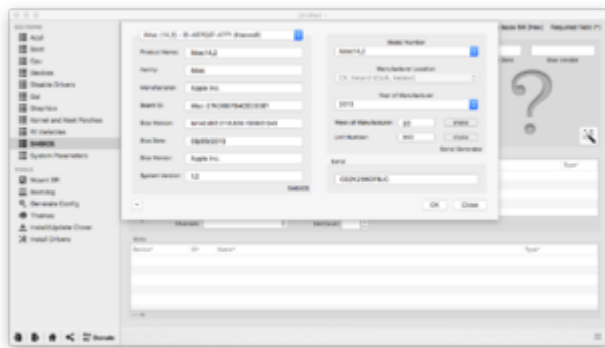


Dann öffnen wir die Seite SMBIOS und drücken da rechts auf den Zauberstab, da tauchen dann die verschiedenen Modelle auf.



Standard ist da inzwischen (leider) der iMac. Da suchen wir uns einen passenden Typen aus, 13,2 ist für Ivy Bridge gut, 14,2 für Haswell mit Nvidia-Grafik, 15,1 für Haswell mit ATI-Grafik, 16,1 für Broadwell und 17,1 für Skylake jeweils mit Intel- oder ATI-Grafik. Bei einem Skylake-Rechner mit Nvidia-Grafik sollte man besser auch den 14,2er auswählen.

Dann drückt man ein paar mal die shake-Tasten, bis einem die entstandenen Nummern gefallen. Mit einem OK bekommt man diese Werte dann in seine Liste.

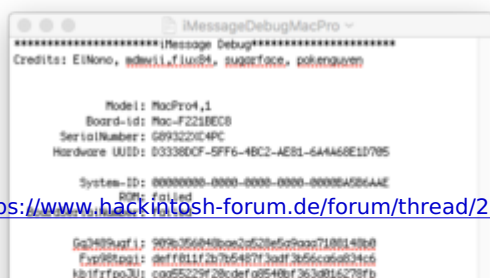


Auf der Seite Rt Variables in der oberen Zeile ist Platz für vier wichtige Werte, von denen drei nach Standard eingetragen werden: unter ROM geht man auf „UseMacAddr0“, bei BooterConfig kommt 0x28 und bei CsrActiveConfig 0x7f rein. Und bei MLB sinnvoller Weise die selbst generierte Seriennummer, die von 12 auf 17 Zeichen aufgefüllt wird.



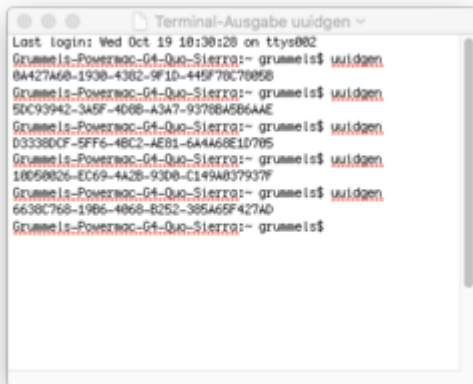
Jetzt fehlen noch zwei wichtige Werte, die aber direkt mathematisch aneinander hängen (sollten): die UUIDs für Hardware und System-ID.

Wenn wir Glück haben, hat unser Board diese UUIDs bereits. Dann sehen wir sie in dem eingangs gemachten iMessageDebug-Bericht.

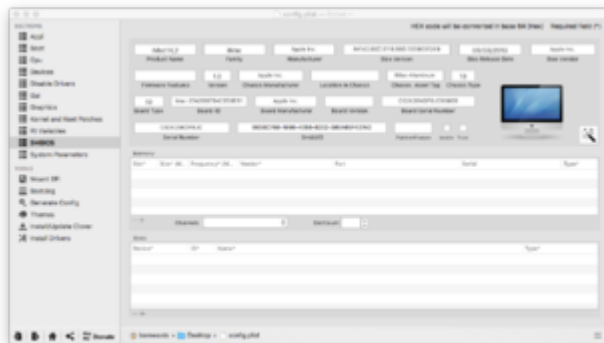


Sind die Felder Hardware UUID und System-UUID aber leer oder stehen dort überwiegend Nullen, ist es kein Problem, den entsprechenden Wert selbst zu generieren, EIN Wert reicht hier, den Rest macht unser Rechner allein:

Einfach das Dienstprogramm Terminal öffnen und dort uuidgen eingeben. Dann entsteht eine UUID, wenn man sich wirklich sicher sein will, dass diese UUID einmalig ist, gibt man den Befehl zwei, drei Mal ein, dann wird jedes Mal eine neue erstellt.



Diesen Wert tragen wir dann im Configurator in das Feld SmUUID auf der Seite SMBIOS ein.



Bei einer Neuinstallation mit Clover, bzw. dem darauf aufgebauten Uni-/Multibeast war es das jetzt eigentlich.

Nach einem Neustart sollten die neuen Daten auch vom iMessageDebug genau so ausgelesen werden, sonst darf man eben noch mal Schritt für Schritt kontrollieren..

Für die Freunde des Ozmosis-BIOS gibt es eine kleine Umleitung:

Spoiler anzeigen

Wenn der neue Rechner auf einer älteren System-Platte aufbaut

und es so auf Anhieb nicht klappt, kann es sinnvoll sein, noch folgende Daten zu löschen (vorher wieder ShowAllFiles..):

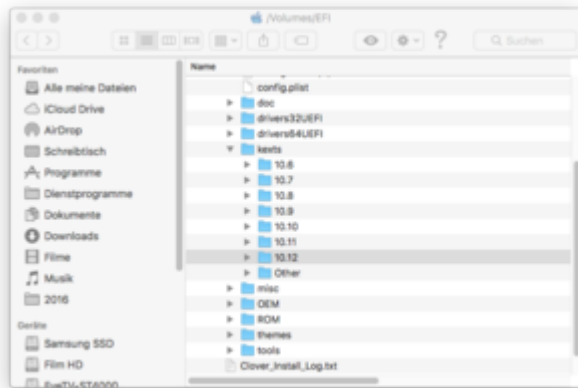
Spoiler anzeigen



Jetzt fehlt noch eine wichtige Kleinigkeit, die wir erst seit kurzer Zeit wieder auch am Hackintosh nutzen können:

iTunes-Filme mit DRM-Schutz.

Da für hat der findige Entwickler vit9696 einen speziellen Kext (=Treiber) gebaut und entwickelt ihn stetig weiter. Dieser [Shiki-Kext](#) kommt in unserem Clover-Fall in den entsprechenden kexts- Ordner auf unserer Start-EFI, wo wir vorher automatisiert oder nicht schon den Ethernet-Kext und ähnliches abgelegt haben, also auf der EFI /EFI/CLOVER/kexts/10.11 oder 10.12 und noch einmal dort in Other.



Bei Ozmosis-Installation sollte er in den Ordner EFI/EFI/Oz/Darwin)Extentions/Common gelegt werden, wirkt aber wieder erst nach einem Neustart mit den gedrückten Tasten alt, cmd, r und p und dem farbig aufblinkenden Bildschirm.

Je nach verwendeter Grafik sollte dann noch ein Terminal-Befehl eingegeben werden:

für Intel: `defaults write com.apple.AppleGVA forceIntel -boolean yes`

für Nvidia: `defaults write com.apple.AppleGVA forceNV -boolean yes`

und für AMD: `defaults write com.apple.AppleGVA forceATI -boolean yes`



Wichtig ist hier: VOR dem Anmelden bei den Apple-Diensten alles noch einmal, zweimal, dreimal genau kontrollieren.

```
iMessageDebug.txt -- Bearbeiten
*****iMessage Debug*****
Credits: E!Nano, mdui1, f1u84, sugarface, pokemayem

Model: iMac14,2
Board-Id: Mac-27AD0B704CE8661
SerialNumber: C82K28DF83C
Hardware UUID: 18050826-EC69-4A2B-9308-C149A837937F
System-ID: 6638C768-19B6-4868-B252-395A65F427AD
ROM: 486c8fc3948e
BoardSerialNumber: C82K28DF83C0000

Go340Puq1: 9635ea300914d8e9d67060e19be1bc0d2
Fyp98tpq1: 336eb28f42a2d79b74d77b411106504983
hb1frfpo3U: 166145809d8ef46ec84b3b8eadf5dd8ee
sync4ZioTNDw: 12618881e70f51f4ed1acb488cae9a5963
gbWPlatEctns: 8c8bb720f18f1a2c73d3d8c19a92ae952e
```

DA für wurde iMessageDebug ja gemacht. Und mit falschen Zahlen kann man sich hier wunderschön aussperren. Und darf ganz von vorn wieder anfangen.

Wenn es jetzt trotz aller richtig eingestellten Werte nicht gleich klappt: nicht verzweifeln! Das kann auch bei originaler Mac-Hardware so ordentlich klemmen, dass mancher Experte dran zweifelt! (http://www.macworld.com/articl...ntication.html#tk.rss_all)