

Ozmosis Bios für macOS High Sierra erstellen

Beitrag von „Fredde2209“ vom 4. November 2017, 22:41

Nun ja, die obige Anleitung ist ja schön und gut, aber leider nicht mehr wirklich aktuell. September 2014 ist vorbei und auch Ozmosis hat sich entwickelt. Ein bisschen... Also hier eine Ergänzung dazu, wie ihr Schritt für Schritt vorgehen solltet, wenn ihr euer eigenes Ozmosis Bios für macOS High-Sierra erstellen wollt.

Vorab ist zu sagen:

Ozmosis funktioniert nur bis maximal Haswell, also B97, H97 und Z97. Allerdings ist momentan eher darauf zu setzen, dass B97 und H97 nicht funktionieren, da einfach zu wenig Platz im Bios ist, um alle nötigen Dateien unterzubringen. Bei Asus und AsRock kommt es vereinzelt dennoch vor, dass genügend Platz vorhanden ist, also versucht euch ruhig selbst dran!

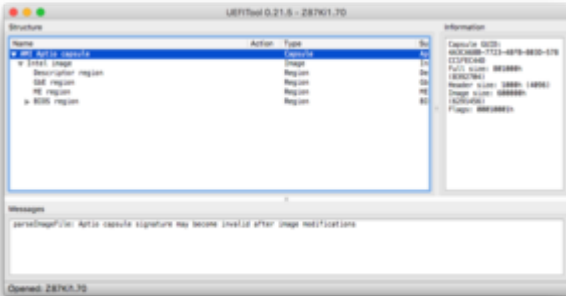
Skylake, KabyLake, CoffeLake, ... braucht ihr nicht probieren, der "neue" Aptio 5 Standart ist nicht mit Ozmosis kompatibel.

Und wie geht dat jetzt?

Je nach Hersteller des Mainboards, müssen vor dem injizieren von Ozmosis noch einige Vorbereitungen getroffen werden. Asus und AsRock haben zum Beispiel gern mal das MSR2 Register gesperrt, wodurch der Start von macOS verhindert werden kann. macOS möchte dort anscheinend gerne rein schreiben und wenn dieses nicht möglich ist, landen wir leider nicht im Installer, wie gewünscht. Ich empfehle dennoch so ziemlich jedes Bios (auch MSI, etc.) einmal durch diesen Vorgang zu jagen. Es schadet nicht!

AsRock:

Hierbei gibt es noch eine Besonderheit! AsRock baut ihr Bios manchmal in eine "Aptio Capsule". Andere Hersteller genauso, allerdings lässt sich bei AsRock das Bios nicht flashen, sobald etwas in dieser Capsule verändert wird. Daher muss das Bios dort erstmal raus. Dafür öffnen wir zunächst mal das UEFITool und schieben unser Bios einfach mal darein (Beispiel: Z87 Killer):



Dabei klicken wir jetzt mit der rechten Maustaste auf diese Capsule (im Bild ausgewählt) und klicken auf "Extract Body". Damit wird der Inhalt, der in diesem "Ordner" ist extrahiert wird. Das ist immer ein "Intel Image". Dann sollte sich bereits ein Fenster zum Speichern öffnen. Dort wählt ihr einfach einen Ordner aus und speichert es dort ab.

Weiterer Verlauf für AsRock, Asus, MSI, ...:

So, das wäre schonmal geschafft. Weiter geht es dann jetzt mit dem entsperren des MSR2 Registers. Dafür brauchen wir das Tool "UEFIPatch". Dieses liegt in einem Ordner mit einer patches.txt Datei, welche es auf jeden Fall benötigt - Diese also bitte nicht löschen! In diesen Ordner schiebt ihr auch euer ROM hinein. Bei mir liegt dieser Ordner "UEFIPatch_0" jetzt unter "~/Documents/Ozmosis-HighSierra/". Daher verschiebe ich mein Rom also nach ~/Documents/Ozmosis-HighSierra/UEFIPatch_0 und navigiere mit dem Terminal dort hin. Also im Terminal einfach eingeben:

Code

1. cd Documents/Ozmosis-HighSierra/UEFIPatch_0/

Tipp: Die Tab-Taste hilft dir mit der Auto-Vervollständigung 😊

Da UEFIPatch nun eine Terminal-Anwendung ist, müssen wir es auch so aufrufen. Um das Tool jetzt anzusprechen wird der Befehl ./UEFIPatch benutzt. Wenn ihr die Datei umbenannt hättet, wäre es dementsprechend anders, aber immer mit ./ vorher. Dahinter könnt ihr einfach den Namen eures ROMs schreiben (auch wieder die Autovervollständigung nutzen). Es sollte sich dann ergeben ./UEFIPatch ROMName . Bei mir wäre das dann:

Code

1. ./UEFIPatch Z87Killer.rom

ENTER drücken - fertig. Wenn es fertig ist, steht im Terminal, welche Patches er angewendet hat. Das ist für uns wohl unwichtig, es kann jedoch auch sein, dass keine Patches angewendet werden müssen, da das MSR2 Register nicht gesperrt war. Dann steht einfach (wie in meinem Beispiel) im Terminal: No patches can be applied to input file

Falls doch welche angewendet wurden, wird ein neues ROM in diesem Ordner liegen. Dieses heißt wie das originale, mit .patched hinten dran.

Nun aber zu Ozmosis!

Wie [@kuckkuck hier](#) schon passend aufgelistet hat, sind folgende Dateien wirklich notwendig zum Boot von Ozmosis:



Davon werdet ihr auch alle soweit im Anhang finden, allerdings entwickelt sich das Dateisystem APFS noch immer. Da es also sein kann, dass die alte apfs.ffs Datei nicht mehr mit neueren Versionen kompatibel ist, geht es jetzt erstmal darum, die neuste apfs.efi Datei zu bekommen und in das ffs-Format umzuwandeln. Dafür braucht ihr erstmal den macOS High Sierra installer, der Version, für welche ihr euer Bios erstellt. Ich habe gerade nur den 10.13.0 Installer zur Hand, deshalb nehme ich einfach diesen für mein Beispiel.

Die apfs.efi Datei befindet sich in /usr/standalone/i386. Und wie komme ich da jetzt hin? Ganz einfach. Beim Installieren von macOS wird unter anderem eine Ordnerstruktur einfach auf die Festplatte verschoben. Und in dieser Ordnerstruktur befindet sich eben auch die apfs.efi. Zunächst müsst ihr euch die Paketinhalte des Installers anzeigen lassen. Darin befindet sich dann eine BaseSystem.dmg Datei. Diese einfach per Doppelklick mounten und in den Ordner /usr/standalone/i386 navigieren. Darin liegt sie, die apfs.efi! 😊 Also erstmal da raus kopieren, dass wir damit arbeiten können. Den Installer brauchen wir jetzt nicht mehr!

Viel mehr brauchen wir jetzt Kext2FFS. Mit dem Skript lassen sich sowohl Kexte, als auch .efi Dateien in das .ffs Format umwandeln. Das Tool findet ihr unten im Anhang.

Die apfs.efi könnt ihr nun einfach in den Ordner "Efi" legen das Skript via Doppelklick ausführen. In dem Ordner FFS/Efi/Compress findet ihr danach die komprimierte apfs.ffs Datei. Leider ist sie immer noch sehr groß, aber das können wir blöderweise nicht ändern. Deshalb ist es auch nicht immer möglich B97/H97 Boards mit Ozmosis für High Sierra auszurüsten.

Wenn das geschafft ist, muss der Ganze Ramsch jetzt nur noch ins Bios rein. Dabei hilft uns das OZMTool. Damit wir es von überall im Terminal erreichen können (und nicht immer, wie beim UEFITool mit dem Terminal dort hin navigieren müssen) gibt es einen lokalen Ordner, wo wir die Datei platzieren können. Da der Ordner standardmäßig versteckt ist könnt ihr entweder per CMD + Shift + . die versteckten Ordner anzeigen lassen, oder ihn einfach mit dem Terminal öffnen. Man gewöhnt sich dran und wenn man weiß, wo die Dinge sind es oft einfach schneller. Das Ganze geht dann so:

Code

1. open -a Finder /usr/bin/

/usr/bin/ ist der Ordner in den unsere OZMTool Terminal-Anwendung muss, damit sie von überall erreicht werden kann. Einfach hineinkopieren und den Kopiervorgang authentifizieren (Password eingeben).

Nun wie oben auch bereits beschrieben mit dem folgenden Befehl die Dateien in das Bios injizieren:

Code

1. OZMTool --ozmcreate -i /Pfad/zum/orgBios -o /Pfad/zum/endBios -a 1 --compressDxe -f /Pfad/zu/OzFiles/

Hierbei muss das ./ nicht mehr benutzt werden, da Ozmosis nicht mehr in dem aktuellen Ordner liegt.

Warum jetzt diese ganzen Sachen dabei??

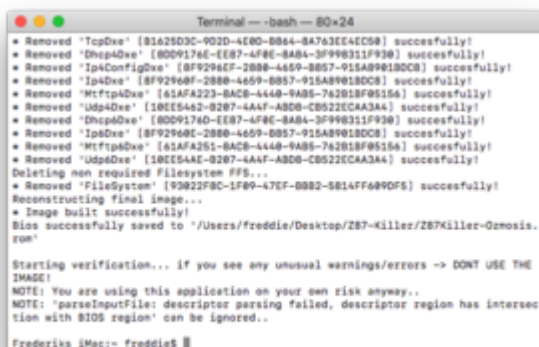
--compressDxe komprimiert die CoreDxe Datei, das Herzstück unseres Bios. Das spart nochmal zusätzlich ein bisschen Platz.

-a 1 löscht die Netzwerk Support Dateien des Bios. Diese werden eh selten benutzt (BSP: Wake on LAN) und können somit für extra Platz im Bios sorgen.

Und warum "-f /Pfad/zu/OzFiles/" nach hinten??

Wenn man öfter solche Biose baut kommt es vor, dass bei manchen eben weniger rein passt. Wenn man jetzt alle Dateien immer in einem Ordner hat, muss man halt immer etwas löschen, was auf Dauer sehr nervig werden kann. Daher habe ich zum Beispiel immer mehrere Ordner. Einmal mit allen Dateien, einmal ohne Sensoren und Injectoren usw. und einmal wirklich nur mit dem nötigsten (siehe Bild oben). So kann ich dann einfach mit den Pfeiltasten nach oben die letzte Eingabe aufrufen und direkt den Ozmosis Pfad anpassen. So muss ich nicht immer bis vorne den "Cursor" verschieben. Ist praktisch und die Anordnung der Parameter juckt das OZMTool nicht. Wenn bei euch also auch nicht alle Dateien rein passen könnt ihr einfach nur die, die oben auf dem Bild beschrieben sind im Ordner drin lassen (oder Stück für Stück Ozmosis Bestandteile, die nicht notwendig sind löschen) und erneut probieren, ob es nun klappt.

Wenn es geklappt hat, sollte am Ende im Terminal-Fenster stehen:



```
Terminal -- bash -- 80x24
* Removed 'TcpxDxe' [81628D9C-902D-4E8D-8B64-8A763EE4C58] successfully!
* Removed 'Dhcp4Dxe' [80D9174E-EE87-4F8E-8AB4-3F998311F938] successfully!
* Removed 'Ip4ConfigDxe' [8F9296EF-2888-4659-8857-915A89818DC8] successfully!
* Removed 'Ip4Dxe' [8F92968F-2888-4659-8857-915A89818DC8] successfully!
* Removed 'MfFtp6Dxe' [41AFA223-8AC8-4448-9A85-762818F85156] successfully!
* Removed 'Udp4Dxe' [18E15442-B287-4A4F-ABD8-C8522ECAA344] successfully!
* Removed 'Dhcp6Dxe' [80D9174D-EE87-4F8E-8AB4-3F998311F938] successfully!
* Removed 'Ip6Dxe' [8F92968E-2888-4659-8857-915A89818DC8] successfully!
* Removed 'MfFtp6Dxe' [41AFA251-8AC8-4448-9A85-762818F85156] successfully!
* Removed 'Udp6Dxe' [18E1544E-B287-4A4F-ABD8-C8522ECAA344] successfully!
* Deleted non required filesystem FFS...
* Removed 'Filesystem' [92822F8C-1F89-47EF-B882-5814FF6890F5] successfully!
Reconstructing final image...
* Image built successfully!
Bios successfully saved to '/Users/freddie/Desktop/Z87-Killer/Z87Killer-Ozmosis.rom'

Starting verification... if you see any unusual warnings/errors -> DONT USE THE IMAGE!
NOTE: You are using this application on your own risk anyway..
NOTE: 'parseInputFile: descriptor parsing failed, descriptor region has intersection with BIOS region' can be ignored..

Frederika iMac:~ freddie$
```

(Bios successfully saved to /Pfad/zum/endBios/)

Wenn es nicht ging sollt dort folgendes stehen:

```
Terminal --bash-- 80x24
* Current file: "DADE1002-1B31-4FE4-8567-26FCEFC7B275_DisablerKext.ffs"...
* File not existent, inserting at the end of volume
Compressing CORE_DXE to save space...
* File was injected compressed successfully!
Deleting network BIOS stuff (PXE) to save space...
* Removed 'TcpDxe' [81625D3C-902D-4E8D-8B64-8A763EE4EC58] successfully!
* Removed 'DhcpDxe' [80D9176D-EE87-4F8E-8AB4-3F998311F938] successfully!
* Removed 'Ip4ConfigDxe' [8F9296E9-2888-4659-8857-915A8981BDC8] successfully!
* Removed 'Ip4Dxe' [8F9296E9-2888-4659-8857-915A8981BDC8] successfully!
* Removed 'WtftpDxe' [43AFA251-8ACB-4448-9AB5-762B1BF05156] successfully!
* Removed 'Udp4Dxe' [10EE5462-8287-4AAE-ABD8-CB522ECA344] successfully!
* Removed 'Dhcp6Dxe' [80D9176D-EE87-4F8E-8AB4-3F998311F938] successfully!
* Removed 'Ip6Dxe' [8F9296E9-2888-4659-8857-915A8981BDC8] successfully!
* Removed 'Wtftp6Dxe' [43AFA251-8ACB-4448-9AB5-762B1BF05156] successfully!
* Removed 'Udp6Dxe' [10EE5462-8287-4AAE-ABD8-CB522ECA344] successfully!
Deleting non required Filesystem FFS...
* Removed 'FileSystem' [93022F8C-1F89-47CF-B082-5B1AFF6890F5] successfully!
Reconstructing final image...
reconstructVolume: 8C8CE578-BA3D-4F1C-9935-896185C32D03: root volume can't be gr
om
ERROR: Image exploded...
! Program exited with errors !
Status code: 17
```

(ERROR: Image exploded)

Bei Asus Roms sollte am Ende das ROM-Image wieder umbenannt werden, zu dem original Namen! Manchmal streikt Flashback nämlich, wenn der Name anders ist 🤔

Was ist danach noch zu tun??

Noch ganz war es das noch nicht. Funktionieren sollte es, dennoch solltet ihr

1. bei euch die Sensoren, sowie auch FakeSMC aktuell halten. Dies könnt ihr genauso mit dem Kext2FFS Tool. Dabei die Kexte also einfach nach Kexts legen und nach dem Ausführen die .ffs Dateien aus dem Ordner Ffs/Kext/Compress entnehmen. Die aktuelle FakeSMC könnt ihr euch super mit [diesem Tool](#) von [@Sascha 77](#) ziehen 😊

2. die defaults.plist anpassen. Diese ist eine, die wir immer in unsere ROMs einbauen. Sie klappt zum starten prima, aber mit iMessage oder so ist leider nicht. Eine Anleitung findet ihr dazu hier in unserem Wiki: <https://www.hackintosh-forum.de/jeder-Rechner-ein-Unikat/>

Ach ja, und darf ich noch Werbung für @kuckkucks Hackintosh-Forum Theme machen? 😊 Das könnt ihr einfach statt des OzmosisHorizontalThemes in euren Ozmosis-Files Ordner legen:
[Neue Themes / Ozmosis GUI / BootMenü / UserInterfacehttps://www.hackintosh-forum.d...jeder-Rechner-ein-Unikat/](https://www.hackintosh-forum.de/jeder-Rechner-ein-Unikat/)