

**Erledigt**

# **SMBIOS mit OZ 1479 richtig setzen - jeder Rechner ein Unikat**

**Beitrag von „Griven“ vom 20. Juni 2015, 02:33**

Das Thema SMBIOS ist spätestens seit der Einführung von FaceTime, iMessage und iCloud wieder ein brandaktuelles Thema, denn will man diese Dienste nutzen ist es unabdingbar ein passendes SMBIOS zu erstellen das den Rechner zu einem Unikat macht aber wie macht man das unter Ozmosis 1479m und wo kommen die Werte her die da eingetragen werden müssen? Hier ein kleiner Überblick wie Ihr Euren Hackintosh zu einem Unikat macht.

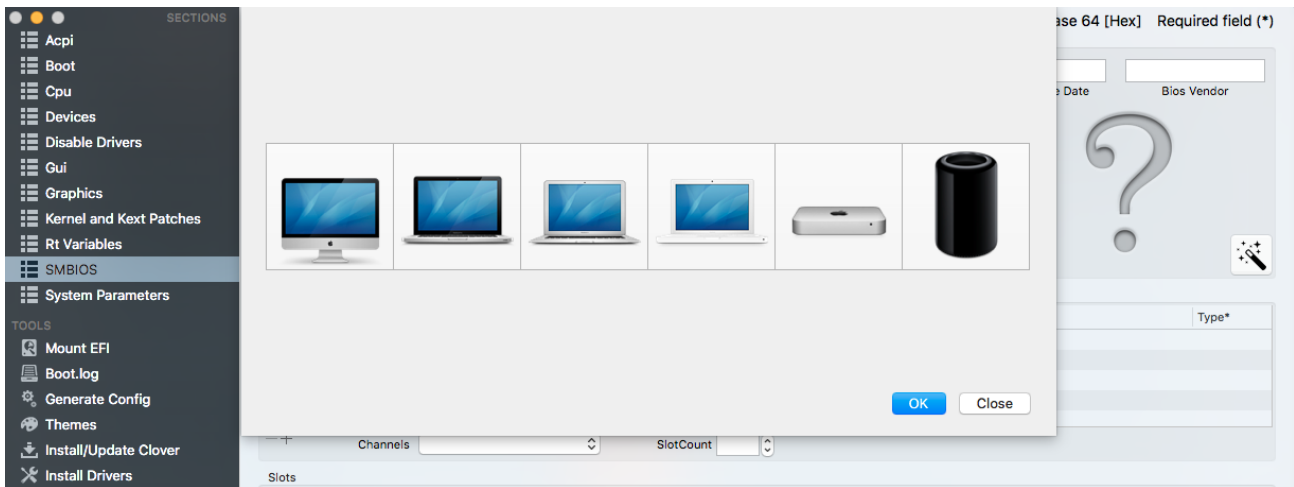
## **Grundsätzliches:**

Das SMBIOS ist mehr als reine Kosmetik denn es hat weit mehr Aufgaben als die Anzeige unter "über Diesen Mac" zu befeuern vielmehr dient es dazu einen Computer eindeutig zu identifizieren und ganz nebenher beeinflusst es auch einen ganzen haufen Dinge innerhalb von OS-X die maßgeblich dazu beitragen ob Euer Rechner gut läuft oder eben auch nicht. Das SMBIOS nimmt zum Beispiel ganz erheblichen Einfluss darauf wie zum Beispiel das PowerManagement geregelt ist oder ob Euer Hackintosh bei kompatibler WLAN/Bluetooth Karte Handoff beherrscht oder nicht usw. . Ausgestattet mit diesem Wissen sollte es klar sein, dass man ein SMBIOS wählt das möglichst nahe an der verbauten Hardware ist sprich es macht keinen Sinn einen MacPro6.1 zu wählen weil die Tonne so schön stylisch ist wenn in der Kiste dann am ende ein Core2Duo steckt und umgekehrt macht es genau so wenig Sinn stoisch immer nur MacPro3.1 zu wählen weil das ja das kompatibelste Model ist. Die Zeiten der reinen Kosmetik sind spätestens seit Mavericks vorbei...

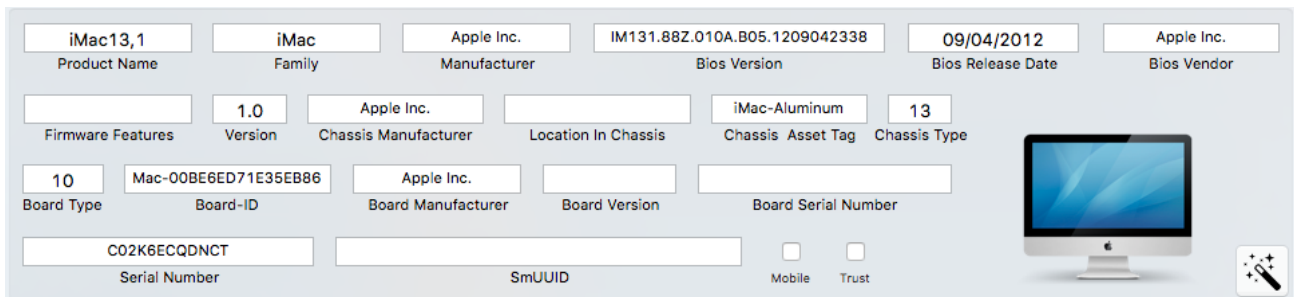
## **Woher die Werte nehmen?**

Um an passende Werte für den Hackintosh zu kommen bietet sich aktuell am ehesten eine aktuelle Version das CloverConfigurators an ( [klick](#) ) denn dessen SMBIOS Generator liefert verlässlich brauchbare Werte. Einmal geladen und gestartet ignorieren wir einfach die Meldung das keine config.plist gefunden wurde und quittieren das Angebot die EFI Partition zu mounten mit einem NO. Anschließend klicken wir zielstrebig links auf den Punkt SMBIOS und in dem sich dann öffnenden Bereich auf den Zauberstab & voilà schon haben wir alles, was wir brauchen





Aber hier erzähle ich Euch sicher auch nichts Neues. Wählt das passende System mit Bedacht aus (Für Desktops geeignet sind am ehesten iMac und MacPro). Mein System1 läuft als iMac 13.1 passend zu seiner Konfiguration (Z7X Chipsatz, IvyBridge CPU). Einmal gewählt und mehrfach auf "Shake" geklickt habt Ihr fast alle Werte, die nötig sind.



### **SMBIOS über den NVRAM setzen...**

Die bei weitem einfachste und vielfach auch problemloseste Variante das SMBIOS bei OZ zu setzen führt über den NVRAM. Seit OZ1479m ist der NVRAM bei nahezu allen Mainboards beschreibbar demnach empfiehlt es sich diesen Weg zu gehen. Damit man den NVRAM aus OS-X heraus beschreiben kann wird zunächst das Terminal gestartet und anschließend der folgende Befehl gefolgt vom eigenen PW eingegeben:

Code

1. `sudo -s`

ist alles richtig gelaufen quittiert OS-X Eure Eingabe mit

Code

1. bash-3.2#

Einmal das Terminal so vorbereitet öffnet Ihr Euch den Texteditor und erstellt eine neue Datei in die Ihr das folgende hinein kopiert:

Code

1. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BaseBoardAssetTag="Base Board Asset Tag#"
2. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BaseBoardSerial=
3. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BiosDate=
4. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BiosVersion=
5. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BoardVersion=
6. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ChassisAssetTag=
7. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:EnclosureType=
8. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareFeatures=2.147.488.791
9. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareFeaturesMask=3.221.487.415
10. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareRevision=65.546
11. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareVendor=Apple
12. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:HardwareAddress=
13. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:HardwareSignature=
14. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:Manufacturer="Apple Inc."
15. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProcessorSerial=
16. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProductFamily=
17. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProductId=
18. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProductName=
19. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:SystemSKU="System SKU#"
20. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:SystemSerial=
21. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:SystemVersion=1.0

Alles anzeigen

wenn auch das erledigt ist geht es daran die Werte zu komplettieren hierbei gilt überall da, wo nachdem dem "=" schon etwas steht lassen wir die Finger von bei allen anderen Werten kommen entweder die Werte aus dem CloverConfigurator rein oder an den Stellen wo die Werte nicht aus dem Configurator kommen gibt es einen Verweis darauf wo sie herkommen. Also los geht's...

Code

1. BaseBoardAssetTag="Base Board Asset Tag#"

2. BaseBoardSerial= <- Serial Number
3. BiosDate= <- Bios Release Date
4. BiosVersion= <- Bios Version
5. BoardVersion= <- Board Type
6. ChassisAssetTag= <- Chassis Asset Tag
7. EnclosureType= <- Chassis Type
8. FirmwareFeatures=2.147.488.791
9. FirmwareFeaturesMask=3.221.487.415
10. FirmwareRevision=65.546
11. FirmwareVendor=Apple
12. HardwareAddress= <- Mac Adresse von EN0 aus Systeminformationen -> Netzwerk -> Ethernet im Format xx:xx:xx:xx:xx:xx
13. HardwareSignature= <- eine beliebige UUID -> Terminal -> uuidgen eingeben und kopieren
14. Manufacturer="Apple Inc."
15. ProcessorSerial= <- Serial Number
16. ProductFamily= <- Family
17. ProductId= <- Board-ID
18. ProductName= <-Product Name
19. SystemSKU="System SKU#"
20. SystemSerial= <- Serial Number
21. SystemVersion=1.0

Alles anzeigen

Habt Ihr alles zusammen könnte das zum Beispiel (**NICHT** kopieren, ist ein Beispiel) so aussehen:

Code

1. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BaseBoardAssetTag="Base Board Asset Tag#"
2. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BaseBoardSerial=C02K6ECQDNCT
3. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BiosDate=09/04/2012
4. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BiosVersion=IM131.88Z.010A.B05.1209042338
5. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:BoardVersion=10
6. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ChassisAssetTag=iMac-Aluminum
7. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:EnclosureType=13
8. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareFeatures=2.147.488.791
9. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareFeaturesMask=3.221.487.415
10. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareRevision=65.546

11. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:FirmwareVendor=Apple
12. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:HardwareAddress=40:0b:23:1c:35:10
13. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:HardwareSignature=B5D37D4E-90EC-4DE6-B8D7-6373D34EFE31
14. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:Manufacturer="Apple Inc."
15. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProcessorSerial=C02K6ECQDNCT
16. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProductFamily=iMac
17. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProductId=Mac-00BE6ED71E35EB86
18. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:ProductName=iMac13,1
19. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:SystemSKU="System SKU#"
20. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:SystemSerial=C02K6ECQDNCT
21. NVRAM 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:SystemVersion=1.0

Alles anzeigen

Habt Ihr alles zusammen muss nur noch Zeile für Zeile ins Terminal kopiert werden und anschließend der Rechner neu gestartet werden und das war es schon, Euer System besitzt ab jetzt eine eindeutige Definition mit der es eigentlich kein Problem mehr sein sollte alle Apple Dienste zu nutzen.

### **Was aber wenn sich der NVRAM nicht beschreiben lässt?**

Obwohl die 1479m eigentlich die NVRAM Sperre bei allen Boards aufheben sollte kommt es immer wieder vor, dass sich einzelne Firmwares doch weigern den NVRAM zu beschreiben aber keine Panik auch hier lässt sich das SMBIOS selbstverständlich ändern nur eben nicht über den NVRAM. Damit es in dem Fall mit einem perfekten SMBIOS klappt müssen die Werte in eine .plist Datei eingetragen werden die dann auf die /EFI Partition wandert. Im großen und ganzen sind die Vorbereitungen die selben Sprich ihr könnt Euch zum ermitteln der Werte weitestgehend an den oberen Teil der Anleitung halten.

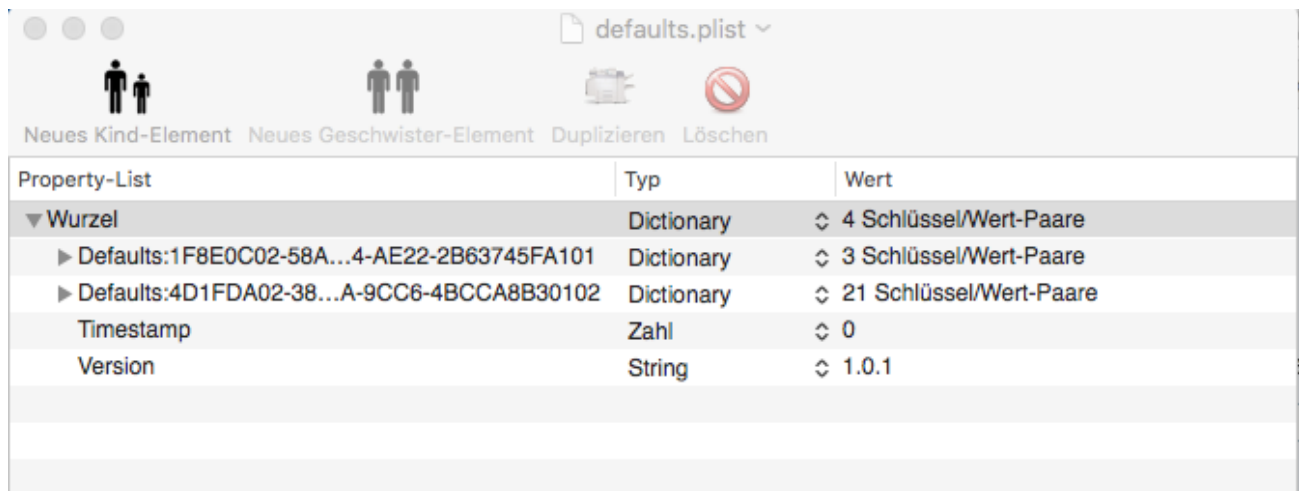
### **Was brauche ich um die defaults.plist zu erstellen bzw. zu bearbeiten?**

Neben der OzmoisDefaults.plist (-> hier aus dem Ozmosis Paket extrahiert [defaults.plist.zip](#) ) benötigt Ihr noch einen plist Editor Eurer Wahl oder das Programm textedit aus OS-X. Als plist Editor eignet sich entweder der in Xcode enthaltene Apple eigene plist Editor oder aber PlistEdit Pro welches sich als kostenlose trial herunterladen lässt. Egal was Ihr verwendet es unterscheidet sich lediglich in der Handhabung, die eingetragenen Werte sind überall die selben.

### **Alles zusammen, was ist zu tun?**

Wenn wir alles zusammen haben geht es darum die Werte in die defaults.plist einzutragen.

Hierzu öffnen wir zunächst die heruntergeladene plist Datei im PlistEditor (ich benutze PlistEdit Pro) zum bearbeiten. Direkt nach dem Öffnen der defaults.plist fällt auf, dass die Struktur den NVRAM Befehlen oben nachempfunden ist.



Für unser Vorhaben ist nur der 2. Knoten (Defaults:4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102) interessant denn analog zum NVRAM beherbergt auch die defaults.plist die Informationen zum SMBIOS unter dieser Adresse was ersichtlich wird, wenn wir den Knoten einfach mal aufklappen.

▼ Defaults:4D1FDA02-38...A-9CC6-4BCCA8B30102	Dictionary	↕ 21 Schlüssel/Wert-Paare
BaseBoardAssetTag	String	↕ Base Board Asset Tag#
BaseBoardSerial	String	↕
BiosDate	String	↕
BiosVersion	String	↕
BoardVersion	String	↕
ChassisAssetTag	String	↕
EnclosureType	Zahl	↕ 13
FirmwareFeatures	Zahl	↕ 2.147.488.791
FirmwareFeaturesMask	Zahl	↕ 3.221.487.415
FirmwareRevision	Zahl	↕ 65.546
FirmwareVendor	String	↕ Apple
HardwareAddress	String	↕

Da die einzelnen Punkte genau so benannt sind, wie bei den NVRAM Befehlen ist der Rest eigentlich simples Copy&Paste sprich einfach die Werte jeweils passend zu den jeweiligen Punkten in die defaults.plist einfügen und anschließend speichern. Einmal erzeugt und gespeichert müssen wir Ozmosis jetzt nur noch davon überzeugen unsere Defaults auch zu

verwenden. Hierzu binden wir uns zunächst mal die EFI Partition unserer ersten Platte ein (hier sucht OZ per default nach diesen Daten). Dazu einfach ein Terminal öffnen und folgenden Befehl eingeben

Code

1. `diskutil list`

und erhalten anschließend in etwa die folgende Ausgabe

Code

1. `diskutil list`
2. `/dev/disk0 (internal, physical):`
3. `#: TYPE NAME SIZE IDENTIFIER`
4. `0: GUID_partition_scheme *128.0 GB disk0`
5. `1: EFI EFI 209.7 MB disk0s1`
6. `2: Apple_HFS yose1 127.0 GB disk0s2`
7. `3: Apple_Boot Recovery HD 784.2 MB disk0s3`
8. `/dev/disk1 (internal, physical):`
9. `#: TYPE NAME SIZE IDENTIFIER`
10. `0: GUID_partition_scheme *60.0 GB disk1`
11. `1: EFI EFI 209.7 MB disk1s1`
12. `2: Apple_HFS capitain 59.2 GB disk1s2`
13. `3: Apple_Boot Recovery HD 650.0 MB disk1s3`

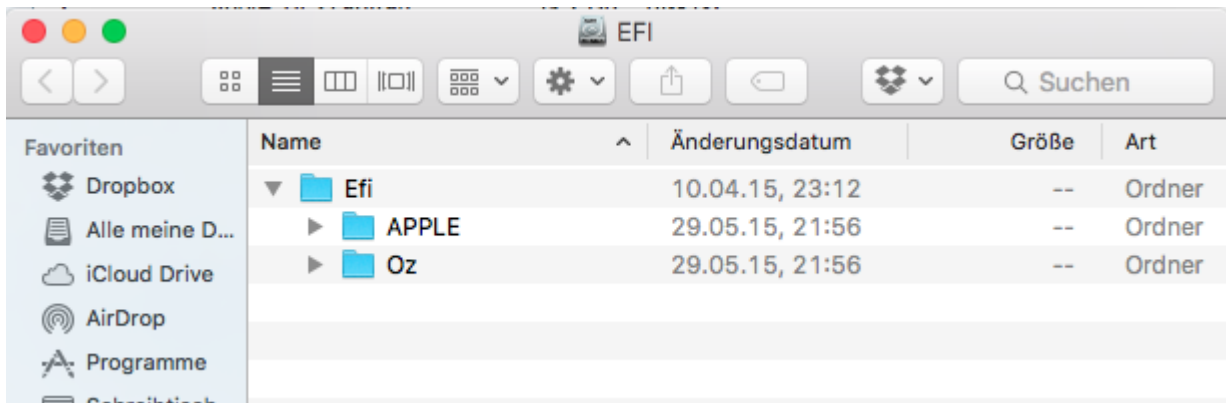
Alles anzeigen

Die erste Platte trägt den Namen `disk0` und die auf ihr enthaltene EFI Partition den Namen `disk0s1`. Damit wir die EFI Partition nun ins System einbinden können geben wir dann die folgenden Befehle ein

Code

1. `mkdir /Volumes/efi`
2. `sudo mount -t msdos /dev/disk0s1 /Volumes/efi`

und schon ist die Partition eingebunden. Als letzten Schritt müssen wir nur noch unsere eben erstellte `defaults.plist` in das passenden Verzeichnis kopieren. Öffnet Ihr die eben eingebundene EFI Partition gibt es hier einen Ordner `EFI` der 2 Unterordner enthält nämlich einen Namens `Apple` und einen anderen Namens `OZ`...



Die defaults.plist wird nun einfach direkt in den OZ Ordner eingefügt und im Anschluss der Rechner heruntergefahren. Ist der Rechner heruntergefahren schalten wir ihn wieder ein und gehen direkt ins Bios. Um die neuen Werte aus der default.plist nun zu übernehmen drücken wir im Bios die Tasten CMD+ALT+P+R gleichzeitig woraufhin sich der Bildschirm erst Rot dann Orange und letztlich grün färbt und der Rechner neu startet voilà das wars schon Ihr habt erfolgreich ein SMBIOS gesetzt.