

Erledigt

Clover richtig installieren, ein kleines HowTo

Beitrag von „Griven“ vom 20. Januar 2013, 02:30

Da aktuell immer mehr von Euch anfangen mit Clover zu experimentieren und dabei recht schnell feststellen, das Clover um einiges komplizierter zu installieren ist als Chameleon oder Chimera versuche ich mal eine kleine Anleitung zu schreiben, wie sich [Clover installieren](#) und konfigurieren lässt und wobei man dabei achten muss.

Was wird benötigt:

- [Clover Installationpaket](#)
- USB Stick
- PLIST Editor (entweder XCODE oder [plistedit pro](#))

Was ist Clover eigentlich und welchen Vorteil habe ich durch Clover?

Anders als beispielsweise Chameleon oder Chimera ist Clover weniger ein in Bootloader im eigentlichen Sinne sondern vielmehr eine EFI Simulation die in Echtzeit funktioniert und es so möglich macht OSX mit seinem eigenen Bootloader zu starten. Einfach ausgedrückt hängt sich Clover so in den Bootprozess ein, dass es nach dem BIOS Post die Hardware analysiert und aus der gefundenen Hardware ein vollständig Apple kompatibles EFI Bios erzeugt. Hier ein Vergleich der Bootprozesse mit Chameleon/Chimera und Clover:

- Chimera/Chameleon: Bios->boot0->boot1->BootLoader->mach_kernel
- Clover: BIOS->boot0->boot1->BOOT->CLOVERX64.efi->Apple's boot.efi->mach_kernel

Während also Chameleon/Chimera Apples eigenen Bootloader umgehen um den Kernel zu starten gibt Clover nach kompletter EFI Simulation die Kontrolle an den AppleBootloader zurück was gleich mehrere Vorteile hat wobei der bedeutendste wohl die Tatsache ist, dass Clover OSX ermöglicht Informationen aus dem EFI zur Laufzeit zu bekommen was keiner der anderen Bootloader ermöglicht. Apple legt diverse Informationen zur Hardware im NVRAM des MACs ab was zur Laufzeit in Kombination mit dem aus dem EFI gelesenen Informationen passiert um im Anschluss die dort gespeicherten Informationen für verschiedenen Dienste und Services zu benutzen. Dieser Mechanismus funktioniert analog auch auf Hackintosh Systemen allerdings nur soweit, wie der Bootloader die nötigen Informationen zur Verfügung stellt. Bei Chimera oder Chameleon und deren derivaten sind einige, wenige dieser Werte so zu sagen unveränderbar hardcodiert vorhanden viele von Ihnen werden aber nicht bereit gestellt, weil sie für den Betrieb bisher nicht notwendig waren und genau diese Lücke schließt Clover damit, dass es OSX nicht einfach über einen alternativen Loader startet sondern anstelle dessen lieber dafür sorgt, dass OSX ein Umfeld vorfindet das so nah an einem echten MAC ist, das apples eigener Loader Einsatz finden kann.

Wie Installiere ich Clover?:

Zunächst auf keinen Fall direkt auf die Festplatte sondern zum testen und ausprobieren immer erst mal auf einen USB Stick um ein bootfähiges System in der Hinterhand zu haben, falls was schief geht. Den USB Stick bereiten wir uns mit dem Festplattendienstprogramm für die Installation von Clover vor indem wir ihm im GUID Layout partitionieren und anschließend MAC OS Extended (Journaled) formatieren. Die Größe des Sticks spielt fast keine Rolle, er sollte nur nicht kleiner als 512MB sein damit sich neben Clover noch einige Tools auf ihm unterbringen lassen. Wenn der Stick bereit ist kann das Installations Programm gestartet werden.

Als Ziel wählt Ihr natürlich Euren USB Stick aus. Wichtig ist, dass die Standardinstallation in

vielen Fällen nicht funktioniert und Ihr jetzt über anpassen die zu Eurem System passenden Einstellungen wählen müsst. Je nach dem ob Ihr ein Board mit BIOS oder UEFI einsetzt müsst Ihr die entsprechenden Optionen wählen:

- Für klassisches Bios: BIOS MBR boot0hfs und je nachdem ob Ihr den 64Bit oder 32Bit Kernel starten möchtet für 2. die entsprechende EFI Variante (32 od. 64Bit)
- Für UEFI Mainboards: UEFI Boot und je nachdem ob Ihr den 64Bit oder 32Bit Kernel starten möchtet für 2. die entsprechende EFI Variante (32 od. 64Bit)

Beachtet bitte, dass die unter 2. gewählte Option das Verhalten von OSX entscheidend beeinflussen kann. Habt Ihr an dieser Stelle 32Bit gewählt wird ein 32Bit EFI emuliert was unter anderem dazu führt, dass ML nicht startet oder installiert werden kann, da es nicht auf MAC´s mit 32Bit EFI läuft und Lion den 64Bit Mode verweigert da auch dieser auf 32Bit EFI Systemen normalerweise nicht lauffähig ist. Solange es keine zwingende Notwendigkeit gibt auf 32Bit EFI zu gehen solltet Ihr die 64Bit Variante einsetzen. Wenn Ihr alles eingestellt habt klickt Ihr nun nur noch auf Installieren, gebt Euer Kennwort ein und wartet bis der Installer seine Arbeit erledigt hat. Die Installation des Sticks ist damit abgeschlossen. Bitte jetzt das System nicht mit dem Stick booten, denn noch sind wir nicht fertig...

Clover für den ersten Start konfigurieren:

Von Chameleon und dessen Derivaten kennen wir die beiden Dateien SMBIOS.plist und org.chameleon.Boot.plist beide gibt es unter Clover nicht mehr, ihre Inhalte werden zusammengefasst in der config.plist im Verzeichnis EFI untergebracht. Ich benutze zum bearbeiten von plist Dateien bevorzugt PlistEdit Pro es geht aber auch jeder einfache Text oder plist Editor. Anders als die beiden aus Chameleon oder Chimera bekannten Dateien bietet die config.plist von Clover jede Menge Möglichkeiten die Konfiguration zu beeinflussen. Folgende Blöcke stehen zur Verfügung:

- ACPI
- GRAPHICS
- PCI
- Pointer
- RtVariables
- SMBIOS
- SystemParameters

Auf Diese Blöcke werden wir jetzt den Inhalt unserer beiden vorhandenen plist Dateien im folgenden verteilen.

Block ACPI:

Analog zu den Einstellungen in der org.chameleon.Boot.plist kommen in diesem Block unsere Einstellungen bezüglich der Erzeugung von P und C States sowie der Name unserer ggf. eingesetzten DSDT unter wobei wir in diesem Arbeitsschritt unsere dsdt.aml sofern vorhanden auch gleich auf den Stick kopieren...

In meinem Fall sieht der Block wie folgt aus:

Code

1. `<key>ACPI</key>`
2. `<dict>`
3. `<key>DsdtName</key>`
4. `<string>dsdt.aml</string>`
5. `<key>GenerateCStates</key>`
6. `<string>Yes</string>`
7. `<key>GeneratePStates</key>`
8. `<string>Yes</string>`
9. `</dict>`

und die DSDT liegt hier:

Warum auch immer, aber es ist nötig die DSDT sofern sie denn vorhanden und nötig ist sowohl im ROOT als auch im /EFI/ACPI/patched Verzeichnis abzulegen damit Clover sie zuverlässig findet.

Block Graphics:

Ihr bootet mit GraphicsEnabler=Yes und/oder gebt einen bestimmten Framebuffer in der org.chameleon.Boot.plist vor? Falls ja gehören diese Einstellungen in den Block Graphics in die config.plist. Exemplarisch kann das so aussehen (ATI Karte, die GE=Yes benötigt und der Redwood Familie angehört (HD55XX)):

Code

1. `<key>Graphics</key>`
2. `<dict>`

3. <key>FBName</key>
4. <string>Redwood</string>
5. <key>GraphicsInjector</key>
6. <string>Yes</string>
7. </dict>

Bei NVidia Karten muss der Framebuffer Name nicht angegeben werden, hier reicht der 2. Key (GraphicsInjector).

Block PCI:

Dieser Block ist ziemlich wichtig da sich hier die PCIRootUID und die Device Properties festlegen lassen. Der volle Satz an Schlüsseln für den Bereich sieht wie folgt aus:

Code

1. <dict> <key>StringInjector</key>
2. <string>No</string>
3. <key>DeviceProperties</key>
4. <string>7a0000001000000010000006e000000020000002010c00d041030a000000001010600001
5. <key>PCIRootUID</key>
6. <string>0</string>
7. <key>HDAInjection</key>
8. <string>Detect</string>
9. <key>USBInjection</key>
10. <string>Yes</string>
11. <key>LpcTune</key>
12. <string>Yes</string>
13. </dict>

Alles anzeigen

Zu befüllen ist das Ganze natürlich wieder analog zu Euren Einstellungen in der org.chameleon.Boot.plist. Je nachdem was dort in Richtung USBFixes und PCIRoot eingestellt gehört hierher übertragen. In Meinem Fall sieht das Ganze dann so aus. Zunächst org.chameleon.Boot.plist:

Code

1. <key>device-properties</key>
2. <string>4b0000001000000010000003f0000000100000002010c00d041030a0100000001010600041

und analog dazu in der config.plist

Code

1. `<key>PCI</key>`
2. `<dict>`
3. `<key>DeviceProperties</key>`
4. `<string>4b0000001000000010000003f0000000100000002010c00d041030a0100000001010600041`
5. `<key>PCIRootUID</key>`
6. `<string>0</string>`
7. `<key>StringInjector</key>`
8. `<string>No</string>`
9. `</dict>`

StringInjector muss in dem Fall den Wert No haben, das ansonsten Clover selbst die DeviceStrings generiert und einsetzt was Erfahrungsgemäß bei Systemen, die mit Chameleon und Device Strings aus EFI Studio installiert wurden zu Problemen führt.

Block Pointer:

Kann und darf ignoriert werden, da er lediglich angibt ob sich das EFI BootMenu von Clover mit der Mouse bedienen lassen soll oder eben nicht und falls die Bedienung mit der mouse erwünscht ist wie sich diese zu verhalten hat. Bei mir funktioniert das Ganze nicht, da ich eine Magic Mouse an Bluetooth einsetze, die auf Bios Ebene noch nicht verfügbar ist.

Block RtVariables:

Jetzt wird es spannend, denn zusammen mit der Bergziege (MountainLion) hat Apple damit begonnen Dienste die bisher den iOS Geräten vorbehalten waren auch auf die MAC's zu schaffen. Einer der prominentesten Vertreter dieser Dienste und zugleich auch der Grund dafür, warum sich plötzlich alle Welt für Clover interessiert dürfte wohl Messages sein. Neue Dienste bedeuten bei Apple auch immer neue Herausforderungen für die Hackintosh Community um diese auch auf den Hackie's nutzen zu können. Echte MAC's haben ein VRAM in dem diverse Informationen aus dem EFI gespeichert sind, welche dem klassischen Hackintosh fehlen einfach weil die Bootloader sie nicht bereitstellen, die Rede ist von RtVariablen die in den NvRam geschrieben werden beim Start des MAC's und die Apple nutzt um diverse Authentifizierungen zu ermöglichen (iCloud, Messages etc.). Neben der Seriennummer der MAC's und dessen Mainboards befinden sich hier auch noch die MAC

Adresse der Netzwerkkarte und diverse andere Informationen. Clover erlaubt es uns diese Variablen zu setzen...

Code

1. `<key>RtVariables</key>`
2. `<dict>`
3. `<key>MLB</key>`
4. `<string>xxxxxxx123456</string>`
5. `<key>ROM</key>`
6. `<data>`
7. `1d7d9f7e`
8. `</data>`
9. `</dict>`

In den Key "MLB" Kommt Eure Seriennummer aus "Über Diesen MAC" die Ihr dann einfach auf 17 Stellen auffüllt. Den Key ROM könnt Ihr setzen, müsst es aber nicht bindend, da Clover das normalerweise automatisch erledigt. Für den KEY MLB liest man im Netz aktuell vielfach die Aussage, das es reicht hier irgendwelche 17 Zeichen einzugeben um Messages zu nutzen, ich rate davon ab und eher dazu die wirkliche Seriennummer zu verwenden und auf 17 Zeichen aufzufüllen. Bisher prüft Apple nicht ob die Seriennummer gültig ist, aber das kann sich schnell ändern.

Block SMBIOS:

Wer hätte es Gedacht, hier kommen die Informationen aus der bestehenden SMBIOS.plist rein wobei Clover hier ein Stück weiter ins Detail geht als es Chameleon und dessen Derivate tun. Keine Panik, die Informationen lassen sich leicht beschaffen. Macht man es sorgfältig sieht dieser Block wie folgt aus:

Code

1. `<key>SMBIOS</key>`
2. `<dict>`
3. `<key>BiosReleaseDate</key>`
4. `<string>05/03/10</string>`
5. `<key>BiosVendor</key>`
6. `<string>Apple Inc.</string>`
7. `<key>BiosVersion</key>`
8. `<string>MP31.88Z.006C.B05.0802291410</string>`
9. `<key>Board-ID</key>`

10. <string>Mac-F42C88C8</string>
11. <key>BoardManufacturer</key>
12. <string>Apple Inc.</string>
13. <key>BoardSerialNumber</key>
14. <string>C020xxxxxxxxxxxx</string>
15. <key>Family</key>
16. <string>MacPro</string>
17. <key>Manufacturer</key>
18. <string>Apple Inc.</string>
19. <key>ProductName</key>
20. <string>MacPro3,1</string>
21. <key>SerialNumber</key>
22. <string>YM812345678</string>
23. <key>SmUUID</key>
24. <string>6669A757-1CB8-5350-A16F-7A74DA1F41EB</string>
25. <key>Version</key>
26. <string>3.0</string>
27. </dict>

Alles anzeigen

Ich habe mir die Informationen aus ChameleonWizzard, Champlist und evermac.com zusammen gesucht ;). Die vielen "x" müsst Ihr natürlich ersetzen...

Block SystemParameters:

Ihr benutzt in der org.chameleon.Boot.plist bestimmt Bootparameter etwa Darkwake=0 etc. alles, was bei Euch als Bootflags zum Einsatz kommt gehört hier her. Zudem gehört hier mindestens folgende Einstellung rein um die im Block SMBIOS und RtVariables vorgenommen Einstellungen auch wirklich an OSX zu übergeben:

Code

1. <key>SystemParameters</key>
2. <dict>
3. <key>InjectSystemID</key>
4. <string>Yes</string>
5. </dict>

Vollständig sieht der Block bei mir so aus:

Code

1. <key>SystemParameters</key>
2. <dict>
3. <key>InjectSystemID</key>
4. <string>Yes</string>
5. <key>LegacyBoot</key>
6. <string>PBR</string>
7. <key>boot-args</key>
8. <string>-v darkwake=0 arch=x86_64 Kernel=mach_kernel</string>
9. <key>prev-lang:kbd</key>
10. <string>en:0</string>
11. </dict>

Alles anzeigen

Bastelt man das alles zusammen erhält man am Ende ein Gesamtkunstwerk, dass als config.plist im Verzeichnis EFI auf dem USB Stick seinen Platz findet. Meine volle config.plist sieht wie folgt aus, Ihr dürft sie gerne kopieren und an Eure Gegebenheiten anpassen:

XML

1. <?xml version="1.0" encoding="UTF-8"?>
2. <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3. <plist version="1.0">
4. <dict>
5. <key>ACPI</key>
6. <dict>
7. <key>DsdtName</key>
8. <string>dsdt.aml</string>
9. <key>GenerateCStates</key>
10. <string>Yes</string>
11. <key>GeneratePStates</key>
12. <string>Yes</string>
13. </dict>
14. <key>CPU</key>
15. <dict>
16. <key>Turbo</key>
17. <string>Yes</string>
18. </dict>
19. <key>Graphics</key>

20. <dict>
21. <key>FBName</key>
22. <string>Redwood</string>
23. <key>GraphicsInjector</key>
24. <string>Yes</string>
25. </dict>
26. <key>PCI</key>
27. <dict>
28. <key>DeviceProperties</key>
29. <string>
 4b00000001000000010000003f0000000100000002010c00d041030a0100000001010600041c010106
 </string>
30. <key>PCIRootUID</key>
31. <string>0</string>
32. <key>StringInjector</key>
33. <string>No</string>
34. </dict>
35. <key>Pointer</key>
36. <dict>
37. <key>Speed</key>
38. <string>8</string>
39. </dict>
40. <key>RtVariables</key>
41. <dict>
42. <key>MLB</key>
43. <string>YMxxxxxxxxxxxx</string>
44. <key>ROM</key>
45. <data>
46. 1d7d9f7e
47. </data>
48. </dict>
49. <key>SMBIOS</key>
50. <dict>
51. <key>BiosReleaseDate</key>
52. <string>05/03/10</string>
53. <key>BiosVendor</key>
54. <string>Apple Inc.</string>
55. <key>BiosVersion</key>
56. <string>MP31.88Z.006C.B05.0802291410</string>
57. <key>Board-ID</key>
58. <string>Mac-F42C88C8</string>
59. <key>BoardManufacturer</key>

```

60. <string>Apple Inc.</string>
61. <key>BoardSerialNumber</key>
62. <string>C0xxxxxxxxxxxxxxxx</string>
63. <key>Family</key>
64. <string>MacPro</string>
65. <key>Manufacturer</key>
66. <string>Apple Inc.</string>
67. <key>ProductName</key>
68. <string>MacPro3,1</string>
69. <key>SerialNumber</key>
70. <string>YMxxxxxxxx</string>
71. <key>SmUUID</key>
72. <string>xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxxxx</string>
73. <key>Version</key>
74. <string>3.0</string>
75. </dict>
76. <key>SystemParameters</key>
77. <dict>
78. <key>InjectSystemID</key>
79. <string>Yes</string>
80. <key>LegacyBoot</key>
81. <string>PBR</string>
82. <key>boot-args</key>
83. <string>-v darkwake=0 arch=x86_64 Kernel=mach_kernel</string>
84. <key>prev-lang:kbd</key>
85. <string>en:0</string>
86. </dict>
87. </dict>
88. </plist>

```

Alles anzeigen

Die jeweils durch "x" ersetzten Werte findet Ihr heraus, wenn Ihr "Über diesen MAC" benutzt und hier dann "Weitere Informationen" -> "Systembericht". Die SmUUID entspricht hierbei dem Wert der unter Hardware-UUID angezeigt wird.

Nötige Kexte kopieren:

Zumindest auf meinen beiden Systemen war es notwendig die FakeSMC.kext und deren evtl. genutzten Plugins ins Verzeichnis /kexts/10.8/ auf dem Stick zu kopieren damit diese geladen wurden. Aus welchem Grund auch immer hat OSX das laden aus /S/L/E mit Clover verweigert, was nach kurzer Benutzung des Hacks natürlich zu Kernel Paniken geführt hat. Es kann, aber **ausdrücklich muss nicht**

, ohne diesen Schritt gehen. Probiert hier einfach aus, was für Euch funktioniert und was nicht.

Wenn Ihr nun alles zusammen habt könnt Ihr Euren Hackintosh mit dem Stick booten und solltet in etwa von einem Botscreen empfangen werden, der diesem hier ähnlich sieht:

Wenn Ihr alles richtig gemacht habt und Euer Hacki mit Clover bootet, dann steht im Normalfall der Nutzung von Messages auch nichts mehr im Wege und die Probleme gehören der Vergangenheit an zumindest war es bei mir so....