

```
< dict>
  <key>Rtc8Allowed</key>
  <false/>
</dict>
```

true - die Regionslänge bleibt bei 8 Byte, falls sie es war, false - wird um 2 Bytes korrigiert, wodurch ein CMOS-Reset zuverlässiger verhindert wird. Wie von vit9696 recherchiert, sollte die Länge der Region noch 8 sein, da sie für die Speicherung des Ruhezustandsschlüssels wichtig ist. Und der Fix selbst ist nützlich. Nun, auf Desktops ist der Ruhezustand nicht unbedingt nötig, also denken Sie vielleicht darüber nach, das CMOS zu dumpen.

FixTMR

Entfernt auf ähnliche Weise einen Interrupt vom _TMR-Timer. Es ist veraltet und wird von MacOS nicht verwendet.

AddIMEI

Erforderlicher Patch für SandyBridge und höher, um ein Gerät IMEI in den device tree (Gerätebaum) hinzuzufügen, wenn es dort nicht bereits vorhanden war.

FixIntelGfx

Der Patch für die integrierte Intel-Grafik ist getrennt vom Rest der Grafikkarten, d. h. es ist möglich, eine Injektion für Intel und nicht für Nvidia zu setzen.

FixWAK

Fügt Return zur Methode _WAK hinzu. Es wird unbedingt gebraucht, aber aus irgendeinem Grund ist es in der DSDT oft nicht enthalten. Offenbar haben sich die Autoren an andere Normen gehalten. In jedem Fall ist dieser der Fix ist vollkommen sicher.

DeleteUnused

Löscht unbenutzte Diskettengeräte. Was meinen Sie, worüber Sie sich Sorgen machen müssen? Tatsächlich werden dadurch noch CRT- und DVI-Geräte entfernt - absolut notwendig Die IntelX3100-Startbedingung auf Dell-Laptops. Ansonsten gibt es einen schwarzen Bildschirm, verifiziert durch Hunderte von Benutzern.

FixADP1

Korrigiert das ADP1-Gerät (power supply (Stromversorgung)), was für einen ordnungsgemäßen sleep der Laptops notwendig ist, wenn sie an die Steckdose angeschlossen oder wenn sie von ihr getrennt werden.

AddPNLF

Fügt das PNLF-Gerät (Hintergrundbeleuchtung) ein, das für die korrekte Steuerung der Bildschirmhelligkeit erforderlich ist und es hilft seltsamerweise bei der Lösung von Schlafproblemen, auch bei Desktops.

FixS3D

In ähnlicher Weise löst dieser Patch das Schlafproblem.

FixACST

In einigen DSDTs gibt es ein Gerät oder eine Methode oder Variable namens ACST, aber dies ist der von MacOSX 10.8+ verwendete Name für die Steuerung des c-state. (Völlig logisch?).

Es führt zu einem Konflikt mit sehr unklarem Verhalten. Dieser Fix benennt alle Vorkommen von solchen Namen und wandeln sie in etwas Sicheres um. Ich kann nicht verstehen, wie Sie diese Serie von Patches ignorieren können! Braucht ihr Jungs nicht einen gut funktionierenden Computer?

AddHDMI

Fügt dem DSDT ein HDAU-Gerät hinzu, das dem HDMI-Ausgang der Graphik-Karte von ATI oder Nvidia. Da die Grafik-Karte separat von der Hauptplatine gekauft wird, ist es klar, dass das die native DSDT einfach nicht über ein solches Gerät verfügt. Zusätzlich wird das Gerät mit Eigenschaft hda-gfx=onboard-1 bzw. onboard-2 ausgestattet:

-1 wenn UseIntelHDMI=false

-2 wenn ein Intel-Port vorhanden ist, der Port 1 belegt hat.

FixRegions

Dies ist ein sehr spezielles Patch. Wenn die anderen Patches dazu gedacht waren, folgendes zu beheben BIOS.aml, um eine gute DSDT aus dem Nichts zu erzeugen, dann ist dieser Fix für endgültige Ausrichtung einer gut gemachten benutzerdefinierten DSDT.aml, aber für BIOS.aml ist es nutzlos. So sieht's aus.

Im DSDT gibt es z. B. Regionen, die eigene Adressen haben:

OperationRegion (GNVS, SystemMemory, 0xDE6A5E18, 0x01CD)

Das Problem ist, dass diese Regionsadresse dynamisch vom BIOS erzeugt wird und sie kann unterschiedlich von Last zu Last sein. Zuerst wurde es beim Wechseln des Gesamtspeichers bemerkt, dann beim Ändern der BIOS-Einstellungen, und auf meinem Computer hängt es auch von der Vorgeschichte der Bootens ab, z. B. von der Größe des belegten NVRAM. Es ist klar, dass in der benutzerdefinierten DSDT.aml diese Zahl fest vorgegeben ist und sollte daher nicht zutreffend sein. Die einfachste Beobachtung ist der Schlafmangel. Nach dem Korrigieren der Region erscheint der Schlaf, aber bis zum nächsten Neustart (Schicht). Dieser Fix fixiert alle Regionen in der benutzerdefinierten DSDT auf BIOS-DSDT-Werte und damit ist die Maske

```
<key>Fixes</key>
  <dict>
    <key>FixRegionen</key>
    <true/>
  </dict>
```

ist eine ausreichende Maske, wenn Sie eine gut gemachte DSDT mit allen mit den erforderlichen Korrekturen haben.

Es gibt noch einen weiteren Patch, der aber nicht DSDT, sondern alle ACPI-Tabellen im Allgemeinen betrifft. der Ort ist illegal.

FixHeaders

Es prüft die Header nicht nur von DSDT, sondern von allen ACPI-Tabellen im Allgemeinen und löst das Problem der chinesischen Zeichen in Tabellennamen, die MacOS nicht toleriert, und sofort Panik auslöst. Unabhängig davon, ob Sie ein Problem mit den Tabellen haben oder nicht, ist es sicherer, diese Korrektur zu aktivieren.

Patch-Auswahl

Wie entscheiden Sie, welche Patches notwendig, welche harmlos und welche gefährlich sind? Probieren sie es aus. Sie werden den Computer in keiner Weise zerstören. All dies geschieht nur während des Betriebes im Speicher, und das wird nach einem Neustart vergessen. Sie können einen Satz von Korrekturen testen, indem Sie Maske im grafischen Menü bearbeiten und dann das Ergebnisses durch Drücken von F5 - "Save DSDT- xxxx.aml speichern, korrigiert durch aktuelle Maske".

Sie können versuchen, mit der aktuellen Maske zu booten. Um den echten nicht zu beeinträchtigen, Das bereits im System vorhandene gepatchte DSDT kann im Menü DSDT-Name: BIOS.aml Wenn es keine solche Datei gibt, nimmt das System die OEM-DSDT aus dem BIOS und nimmt die Korrekturen vor, gemäß der eingestellten Maske. Im Falle eines Fehlers nach dem Neustart des Rechners wird die aktuelle Einstellungen verloren gehen, und die Standardeinstellungen wird wieder funktionieren. Die Maske 0xFFFFFFFF entspricht dem Einschalten aller Fixes, und wenn das System danach geladen wird, ist die Arbeit der Programmierer nicht umsonst. Aus der obigen Beschreibung haben Sie bereits entnommen, dass einige Fixes, die Sie einfach nicht brauchen (z. B. WIFI). Beginnend mit der Revision 1992 arbeiten, um Panik auf dem doppelten Patch zu verhindern, also scheuen Sie sich nicht, unnötige Fragen zu stellen Bits. Zwei Fixes würde ich zur Zeit nicht empfehlen: FIX_SATA, bit 0x0400, ist es besser, einen binären Patch des Kext zu verwenden, und FIX_SHUTDOWN, Bit 0x04, für das setzen von HaltEnabler=true funktioniert statt dessen fast immer, was korrekter funktioniert.

Auch der Patch AddMCHC_0008 ist gefährlich, manche Leute brauchen ihn unbedingt, und für manche ist streng kontraindiziert.

Um zu sehen, wie sich die DSDT-Patches auf das Ergebnis auswirken, und nehmen wir an, sie könnten nicht booten, dann können Sie den folgenden Schlüssel in den ACPI-Abschnitt der Config schreiben:

```
<key>DSDT</key>
<dict>
  <key>Debug</key>
  <true/>
```

Dann werden vor dem Start des Systems zwei Dateien im Ordner auf der Festplatte gespeichert

/EFI/CLOVER/ACPI/origin/

DSDT-or.aml.

DSDT-pa15.aml

origin ist Ihre DSDT, die entweder von der Festplatte geladen oder aus dem BIOS entnommen wird, bevor Patches eingespielt wurden.

patched - nachdem Patches eingespielt wurden. Da Sie beim Booten gescheitert sind, werden Sie immer mehrere Versuche unternehmen, und die gepatchten Dateien werden fortlaufend nummeriert, ohne dass die alten Informationen gelöscht werden. 15 in diesem Fall - dies ist der 15. Versuch des Startens in diesem Fall, sollte erfolgreich sein und wir müssen schauen, was das Problem beim 14. Versuch war.

Dennoch empfehle ich, ein doppeltes Patch zu vermeiden. Es kann auch eine Situation entstehen, in der Das doppelte Patchen tritt auf, weil die OEM-DSDT bereits die Methode _DSM hat, wenn wollen wir unsere eigene entwickeln. Sie müssen also die Bits in der Maske setzen DropOEM_DSM. Siehe [Hardware-Konfiguration](#) → [ACPI](#) → [DSDT](#) → [DropOEM_DSM](#). Nein, Sie müssen alle OEM-DSMs umbenennen.

Hand editing DSDT

Voraussetzungen

Sie müssen eine Art von Computer mit einem Betriebssystem haben, damit Sie eine Textdatei bearbeiten können. Oder vielleicht sind sie bereits an diesem Computer, sie haben macOS irgendwie installiert und möchten nun die DSDT verbessern.

Es gibt einen iasl-Kommandozeilen-Compiler für jedes Betriebssystem. Für Windows wird die iasl.exe verwendet, die eine Befehlszeile bietet, wie die Programme für MacOS. Sie haben die gleichen Funktionen, und liefert die gleichen Ergebnisse. Es ist umständlich, Texte in Windows zu bearbeiten, Notepad hat keine Syntaxhervorhebung und Zeilennummerierung, ist es besser, Notepad+ zu verwenden. Für MacOS gibt es viele Optionen, Xcode, und BBEdit, und andere. Eine neue Version des Compilers für Mac ist hier verfügbar Dsdt-Optimierung. Neuester Compiler.

Für Windows unter <https://acpica.org/Windows Binary Tools>

Sie sollten auch die ACPI-Sprachbeschreibung ACPI_6_2.pdf vorrätig haben. Es ist besser, eine neue Version zu besorgen, weil Sie mit der DSDT umgehen müssen, das sie aus dem frische BIOS erhalten haben (https://uefi.org/sites/default/files/resources/ACPI_6_2.pdf).

Creating a blank (Erstellen eines Rohlings)

Besorgen sie sich zunächst ein-Flash-Drive (USB-Stick), der bootfähig ist, auch wenn kein System vorhanden ist. Es ist wichtig, dass er im FAT32-Format vorliegt. Dort wird Clover darauf kopiert. Booten sie von diesem Flash-Laufwerk (USB-Stick) den Computer, um ins Clover Interface (Menue) zu kommen. Drücken Sie die Taste "O" (lateinischer Buchstabe O) oder wählen Sie das Symbol im Menü Optionen. Gehen Sie zu ACPI-> dort finden Sie DSDT Name: und geben sie BIOS.aml ein. Dies ist genau die DSDT, die Sie unter Windows über Aida erhalten würden. Abwärts gehen sie im Menü und wählen sie DSDT-Fixes -> dort können Sie Kästchen ankreuzen, entweder mit der Tastatur oder mit der Maus. Setzen Sie fast alle Kontrollkästchen, bei denen Sie sich nicht unwohl fühlen. Sie brauchen zum Beispiel kein Firewire und keinen Airport, IDE, wenn Sie wissen, dass Sie keine solchen Geräte haben. Alles, was Sie nicht kennen, sollten Sie besser einbauen. Kehren Sie aus diesem Untermenü zurück, und drücken Sie nacheinander F2, F4, F5.

Jetzt können sie den Computer ausschalten und das Flash-Laufwerk an dem Computer anstecken, mit dem sie die Dateien bearbeiten wollen. Die Dateien, an denen wir interessiert sind, befinden sich auf dem Flash-Laufwerk in den Ordnern:

```
EFI\CLOVER\ACPI\origin  
EFI\CLOVER\misc
```

Kopieren Sie diese Ordner in Ihren Arbeitsordner auf Ihrem Computer, in dem sich iasl-Programm befindet. Entweder mit MacOS oder mit Windows. Die Anweisungen sind die gleichen, abgesehen von Feinheiten wie der Schräglage(?)

Decompilation

Die resultierende DSDT.aml ist eine Binärdatei, die Sie nicht mit einem Texteditor bearbeiten (öffnen?) können. Versuchen sie es und schauen Sie sich das an. Von spezialisierten maciASL-Editoren rate ich kategorisch ab, DSDT-Editor und dergleichen, da der Zweck des Themas nicht darin besteht, den guten Onkel rehab(?), der das für sie macht. Es geht darum, mit eigenen Augen zu sehen, was passiert und wie es passiert.

Führen Sie die Befehlszeile aus: cmd.exe unter Windows, Terminal.app unter Mac, whatever unter Linux Ich weiß es nicht, ich glaube, es heißt "Bash".

```
> cd „ ihr ArbeitsOrdner“/origin
```

d. h. in dem gleichen Ordners, in den sie die vom Flash-Laufwerk erhaltenen Dateien kopiert haben. In den gleichen Ordner legen iasl.exe, wenn Sie auf Windows sind, oder installieren Sie iasl auf dem System, wenn Sie auf MacOS sind:

```
$ sudo cp ~/Downloads/iasl /usr/local/bin/
```

Jetzt können wir endlich dekompilein, d. h. die DSDT in eine lesbare Sprache bringen.

```
$ iasl -da SSDT*.aml DSDT-*.aml
```

Die Option -da bewirkt das Wunder, es wird in allen Dateien gleichzeitig nach Zeichen gesucht, so dass es am Ende keine unbekanntenen Zeichen mehr geben sollte. Dies gilt jedoch nur für den Idealfall. In der Praxis funktioniert nicht es, wir erhalten eine Menge Fehler wie AE_ALREADY_EXISTS

~~~~

```
Firmware Error (ACPI): Failure creating named object [IO1B], AE_ALREADY_EXISTS (20200925/ dswload-495) ACPI Error: AE_ALREADY_EXISTS, During name lookup/catalog (20200925/psobject-372) Could not parse external ACPI tables, AE_ALREADY_EXISTS
```

~~~~

Die andere Option erweist sich als erfolgreicher

```
iasl -e SSDT*.aml -d DSDT-*.aml
```

Die Ausgabe ist eine Warnung am Ende:

```
Parsing completed
```

```
Warning - Emitting ASL code "External (BNUM)"
```

```
This is a conflicting declaration with some other declaration within the ASL code
```

```
This external declaration may need to be deleted in order to recompile the dsl . file.
```

Das heißt, in der DSDT.dsl müssen wir die ad entfernen

```
Deklaration External(BNUM, UnknownObj) // Conflictswithlater declaration
```

Because there is an ad (Da es eine Ankündigung gibt?)

```
Field (GNVS, AnyAcc, Lock, Preserve)
```

```
{
```

```
...
```

```
BNUM, 8,
```

Das heißt, der Name wird hier bekannt gegeben, und Sie müssen ihn nicht extern benennen, oder gar einen unbekanntenen Namen nehmen. Aber dann sind da noch die Fehler

```
External (_PR_.CFGD, UnknownObj)
```

Dennoch existiert diese Variable und wird in einer der SSDTs beschrieben. Das Decompilat ist noch unvollständig. Jeder dieser Fälle muss einzeln behandelt werden.

Wir dekompilein die DSDT, die Clover gepatcht hat, und haben somit bereits eine Menge nützlicher Korrekturen. Auf ähnliche Weise können Sie die Originaldatei dekompilein

```
$ iasl -da SSDT*.aml DSDT.aml
```

Und um vergleichen zu können, was war und was nicht war. Und wir haben eine Datei mit dem langen Namen DSDT-1234567.dsl, (wahrscheinlich haben Sie andere Zahlen und Buchstaben). Dies ist die ursprüngliche Voreinstellung, die in DSDT.dsl umbenannt werden sollte, und das Original in DSDT-origin.dsl, bearbeiten und kompilieren in einem unendlichen, iterativen Prozess:

1. Mit einem Texteditor bearbeiten.
2. Kompilieren, DSDT.aml erhalten

```
Kompilierbefehl
```

```
$ iasl -ta DSDT.dsl
```

3. Testen Sie, d.h. legen Sie es in den Ordner EFI\CLOVER\ACPI\patched und starten Sie das System.
4. Zurück zu Punkt 1.

Leider kann die Kompilierung mit folgendem Fehler fehlschlagen:

Code:

```
Input file SSDT-10x.aml, Length 0x37F (895) bytes
ACPI: SSDT 0x0000000000000000 00037F (v02 PmRef          Cpu0Cst
00003001 INTL 20120913)
Pass 1 parse of [SSDT]
ACPI Error: [C3ST] Namespace lookup failure, AE_ALREADY_EXISTS
(20160729/dswload-462)
ACPI Exception: AE_ALREADY_EXISTS, During name lookup/catalog
(20160729/psobject-310)
Could not parse external ACPI tables. AE_ALREADY_EXISTS
```

Das bedeutet, dass die Beschreibung des C3ST-Symbols in zwei verschiedenen SSDTs zu finden ist, die letzte davon ist SSDT-10x.

In meinem Fall stellte sich heraus, dass es ähnlich wie SSDT-5x ist, mit dem Unterschied, dass 5x für die Version von ACPI1.0 ist und 10x für die ACPI2.0 Version. Und die Namen sind die gleichen! Ich habe etwas Ähnliches im Kernelprotokoll. Es ist BIOS-Fehler!

Außerdem befinden sich zwei identische Tabellen (Tische?) im Laptop. Ich bin dem nachgegangen, sie sind eigentlich beide im BIOS vorhanden, das ist nicht mein Fehler (Schuld?).

Was ist zu tun? Entfernen Sie die Zweitbesetzung vor dem Kompilieren, und in der realen Arbeit tun Sie es in der config Clover Drop extra Tabellen (Zusatztische?). Im Falle von Duplikaten fliegen beide weg.

Was ist zu reparieren

1. Syntaxfehler durch den Hersteller dieses Computers
2. Sinnvolle Fehler(?)
3. Tricks aus dem Internet. Zu diesem Zeitpunkt denkt jeder, es ist die einzige Sache, die getan werden muss. Nein, auch die ersten beiden Punkte sind wichtig.

Syntax-Fehler

Betrachtet man die Datei DSDT.dsl selbst, so sieht man die ersten Probleme in der Datei selbst

```
External (_SB_. PCI0. PEG0. VID_. LCD_, UnknownObj)
```

Es bedeutet, dass es irgendwo im Text einen Verweis auf ein Objekt gibt, das Objekt selbst aber nirgends zu finden ist. Wir sind auf der Suche nach ihr. Es ist befindet sich in SSDT-7. Es stellt sich heraus, dass es nicht von dort exportiert wurde. (Aus diesem Grund ist es dieser Laptop schlägt vor, dass dieses SSDT in ein generisches DSDT aufgenommen wird?). Durch einfaches Kopieren des Textes, der SSDT von der ersten geschweiften Klammer bis zur letzten im Ende von DSDT.dsl vor die letzte Klammer.

Zweiter Fehler mit fehlenden Zeichen

```
External (HNOT, MethodObj) // Warning: Unknown method, guessing 1 arguments
```

Eine Suche zeigt, dass die Methode in diesem design code (Konstruktion) erwähnt wird:

```
If (CondRefOf (HNOT))
{
```

```
    If (CondRefOf (HNOT))
    {
        HNOT (Arg0)
    }
    Else
    {
        Notify (GFX0, 0x80) // Status Change
    }
}
```

Dann ist es OK, wenn die Methode undefiniert ist, wird sie nicht ausgeführt. Es ist mir nicht klar, wie dann kompiliert wird. Es ist besser, sie zu löschen und nur Notify zu lassen...

Ich beginne mit dem Kompilieren der gerade empfangenen Datei (ich habe DSDT-1F2C3B4D.dsl in das einfachere DSDT1.dsl umbenannt als ersten Versuch)

```
$ iasl -ta DSDT1.dsl
1 errors, 14 warnings, 91 remarks, 109 optimisations
```

Der Fehler muss behoben werden, sonst gibt es kein Ergebnis. Sie sollten auch Warnungen beheben, indem Sie überlegen, womit der Compiler kämpft. Das ist immer noch eine gute Situation - ich bin beim ersten Kompilieren auf hunderte von Fehlern gestoßen.

In diesem Fall ist der Fehler unkritisch, in der Zeile

```
Name (_HID, "*pnp0c14")
```

Zeichenkettenformat ist ungültig, es wird nach Vorschrift (wissenschaftlich) wie folgt korrigiert :

```
Name (_HID, Eisald ("PNPOC14") /* Windows Management Instrumentation Device */)
```

Es muss korrigiert werden, sonst wird es nicht kompiliert, aber es hat keinen Einfluss auf den Betrieb, es ist nur für Windows von Bedeutung.

The wrinths are more critical (Die Lackierungen sind wirklich kritischer), hier sind Beispiele für Korrekturen.

- das war: - CreateDWordField (BUF0, _SB.PCI0._Y0F._LEN, MSLN) // _LEN: Länge

+ was wurde geändert: CreateQWordField (BUF0, _SB.PCI0._Y0F._LEN, MSLN) // _LEN: Länge

Der Hinweis stand im Kompilierungsprotokoll:

```
ResourceTag larger then field (size mismatch tag 64bit, Field 32 bit
```

Die Regel ist einfach: Tag ist das, was Sie brauchen, Feld ist das, was Sie reparieren müssen:

```
tag=1 => CreateBitField
```

```
tag=8 => CreateByteField
```

```
tag=16 => CreateWordField
```

```
tag=32 => CreateDWordField
```

```
tag=64 => CreateQWordField
```

Es muss korrigiert werden, damit im realen Betrieb der gesendete Wert nicht über den Nachbarwert hinauschießt das nächste Feld. Es gibt keine Möglichkeit, diese Störung abzufangen.

Diese Art der Warnung:

```
Not all control path return a value
```

Es ist ein Fehler der Logik, die Methode muss etwas zurückgeben, aber es stellt sich heraus, dass in einigen Fällen

wird nichts zurück gegeben. MacOS wird natürlich nicht darüber stolpern, aber erwarten Sie auch keine Angemessenheit. Und was soll ich dort schreiben? Return oder Return(Local0)? Es reicht, wenn der Compiler sich entspannt, aber im Allgemeinen sollten Sie sich die Logik ansehen.

Ähnliche Störung

Reserved method should not return a value

Der Code lautet

Code:

```
Method (_SRS, 1, Serialized)          _SRS: Set Resource Settings
{
    Return (BUF2) /* \_SB_.PCI0.A_CC.BUF2 */
}
```

Öffnen Sie die oben genannte pdf, ADCPI-Spezifikation, suchen Sie nach der _SRS-Methode, und lesen, was es tun soll. Die Rückgabe von Werten ist dort nicht vorgesehen.

Also machen wir es wie folgt neu

Code:

```
Method (_SRS, 1, Serialized)          _SRS: Set Resource Settings
{
    BuF2 = Arg0
}
```

Das macht mehr Sinn.

Und die berühmten Warnungen.

Name (_T_0, Null)

Verwendung des reservierten Compiler-Namens _T_0

Der neue iasl-Compiler handhabt diese Konstruktion perfekt, und es besteht keine Notwendigkeit, die tun.

Der alte Compiler erforderte die Umbenennung in TT_0

Meaning errors (Sinnvolle Fehler?)

Sorry, aber man muss zumindest ein bisschen Programmierer sein, um sie zu erkennen. Hinweise kann man nirgendwo erwarten.

In der ursprünglichen DSDT sehen wir:

External (CFGD, IntObj)

Und wir finden diese Variable in **SSDT** CpuPm. Und hier ist es an der Zeit, sich daran zu erinnern, dass wir diese Tabelle bereits haben, zusammen mit dieser Variable!

Sie sollte in DSDT kopiert werden. Sowie andere, die sich als nützlich erweisen könnten.

Code

```
+ Name (CFGD, 0x0066F6FF)
+ Name (PDC0, 0x80000000)
+ Name (PDC1, 0x80000000)
+ Name (PDC2, 0x80000000)
```

```
+ Name (PDC3, 0x80000000)
+ (PDC4, 0x80000000) Name
+ (PDC5, 0x80000000) Name
+ (PDC6, 0x80000000) Name
+ (PDC7, 0x80000000)
+ Name (SDTL, Zero)
```

Ein häufiger Fehler ist die OEM-Methode `_DSM`.

Es ist nicht der Fehler des Herstellers, es ist die Tatsache, dass er für Windows geschrieben hat, und wir wollen aber einen Hackintosh betreiben.

Beispiel

Code:

```
- Methode (_DSM, 4, Serialized) // _DSM: Device Specific Method
{
- Name (_T_0, Null) // _T_x: Emitted by ASL-Compiler
If (Arg0 == ToUID ("a5fc708f-8775-4ba6-bd0c-ba90a1ec72f8"))
{
While (One)
{
```

Sehen Sie die UUID? Sie stammt aus der Windows-Registrierung, bei dem MacOS gibt es so etwas nicht, und es läuft auch nichts damit.

Das wäre nur die halbe Miete, aber wenn es für die Winnds in Ordnung ist, `_DSM` sowohl für das Gerät als auch für den seine Bridge zu haben, verursacht es einen Absturz in MacOS.

Lasst uns alle anderen Leute `_DSM` töten!(?). Der einfache Weg:

Code:

```
- Methode (_DSM, 4, Serialized) // _DSM: Device Specific Method
+ Methode (ZDSM, 4, Serialized)
```

Die Methode selbst hat überlebt, aber es wird sich niemand mehr an sie wenden, und es wird keine Panik entstehen.

Hier ist ein kuriose Stück

Code:

```
OperationRegion (DXHC, SystemMemory, 0xFED1F418, 0x04) XHCD,
1
}
If ((OSYS < 0x07D6) && (OSYS > 0x03E8))
{
XHCD = One
Notify (XHC, Zero) // Bus Check
}
```

Nachdem ich mir die Adresse genauer angesehen habe, ist mir klar geworden, dass es sich um ein Function Disable Bit handelt.

Der Sinn der Operation ist es, USB3 für Systeme unterhalb von Windows Vista zu sperren. Ich denke, dieser Teil sollte ganz herausgeschnitten werden.

Das ist der Schwachsinn der Autoren.

Code:

```
If ((OSYS > 0x07D0) || (OSYS < 0x07D6))
```

Bereiche addieren sich und überschneiden sich mit beliebigen Werten

Es ist eher so, wie es sein sollte

Code:

```
If ((OSYS > 0x07D0) && (OSYS < 0x07D6))
```

Dann überschneiden sich die Bereiche. Aber dann müssen wir sehen, was in der Wenn und was in der Else Anweisung ist. I've got what's right in Then (Dann habe ich es richtig gemacht,) also das ursprüngliche Design

```
Code:
if ((OSYS > 0x07D0) || (OSYS < 0x07D6)) {
    Notify (PCI0, Arg1) }

Else {
    Notify (GFX0, Arg1) }
}
```

Ich habe es nur auf einen operator (Bediener?) reduziert.

```
Notify (PCI0, Arg1)
```

Der Punkt ist, dass dieses If prüft, ob das System WindowsXP ist und tut, Else was es dann für Systeme wie Windows7,8,10 macht.

Was ist der Unterschied? Optimus arbeitet in dem neuen Systemen. Unter **MacOS** benötigen wir eine first-class notification, für den gesamten Bus. Und in ähnlicher Weise müssen wir an anderer Stelle in der DSDT verfahren.

Ein kleiner Krimi zum Thema, vielleicht kommt ja jemand auf die richtige Idee.

Also, die Herausforderung. Der Laptop schläft und wacht auf, wenn er nicht eingesteckt ist. Aber dann ist die Batterie leer. Wenn er jedoch eingesteckt ist, wacht er sofort auf, eine Sekunde nachdem er eingeschlafen ist. Was ist denn los?

Genau bei der DSDT!

Also, auf der Suche nach dem, was den Unterschied zwischen **ACPI-Status in the outlet** (im Sockel?) und nicht macht und welches Gerät das verursacht.

```
Code:
Device(AC)
{
    Name (_HID, "ACPI0003"/* Power Source Device */) // _HID: Hardware-ID
```

Der Decompiler hat uns freundlicherweise mitgeteilt, dass dies die Stromversorgung ist.

Die Methode `_PSR` bestimmt, ob dieses Gerät eingeschaltet ist oder ruht.

Code:

```
Method(_PSR, 0, NotSerialized)    // _PSR: Leistung Quelle
{
    Lokal0 = EVG5 ()
    Lokal0 &= One
    If (Lokal0 != PWRS)
    {
        PWRS = Lokal0
        PNOT ()
    }

    Return (Lokal0)
}
```

Sie können es im Buch `ACPIspec.pdf` nachlesen, beliebige Jahresausgabe:

Ein Integer, der den Status der Stromquelle enthält

0 - Off-line (nicht am Netz)

1 - On-line

Das heißt, in meinem Fall gibt eine fest verdrahtete Methode ECG5() eine Zahl aus, in der das Bit 0 bedeutet eingesteckt, und speichert es in der PWRS-Variable, wenn der Wert dort anders ist.

Suchen Sie nun im gesamten Code, wo PWRS erwähnt wird, und finden Sie es in der Methode _PTS (Prepare To Schlaf), der für das Einschlafen verantwortlich ist.

Code:

```
Method (_PTS, 1, NotSerialized)    // _PTS: Prepare To Sleep
{
P80D = zero
P8XH (zero, Arg0)
PTS (Arg0)
Wenn (AOAC & one) {}
Wenn (Arg0 == 0x03)
{
If (PWRS == zero)
{
\_SB.PCI0.XHC.PMEB = zero
\_SB.PCI0.EHC1.PMEB = zero
\_SB.PCI0.EHC2.PMEB = zero
If (\_SB.PCI0.XHC.PMST == one)
{
\_SB.PCI0.XHC.PMST = one
}

If (\_SB.PCI0.EHC1.PMST == one)
{
\_SB.PCI0.EHC1.PMST = one
}

If (\_SB.PCI0.EHC2.PMST == one)
{
\_SB.PCI0.EHC2.PMST = one
}
}
}
```

Interessant hier! Voraussetzung: Wenn das Gerät nicht in die Steckdose gesteckt ist, dann tun Sie etwas in SSB2 und SSB3, und wenn sie es tun, müssen Sie nichts tun. Im Protokoll des unerwarteten Aufwachens sehen wir XHC, EHC2.

Die Ersteller dieser DSDT glauben also, dass ein Laptop nur eingesteckt bedient werden kann. Ist es nicht mit der Stromquelle verbunden, sollte es nicht arbeiten.

Durch Entfernen dieser Bedingung If (PWRS == Zero) habe ich die Lösung für mein Problem erhalten. Ich habe es jetzt so, daß der Laptop schläft und wacht auf, und das Netzteil stört ihn nicht.

Viel Erfolg bei der Erstellung einer minimal korrekten DSDT!

Native Speedstep

Richtiger ist es, von Power- und Prozessorfrequenzmanagement (PPFM) zu sprechen, indem EIST - Enhanced Intel Speedstep Technology, daher das russische Wort "Speedstep" genannt.

Eigentlich geht es in diesem Thema nicht so sehr um den Bootloader allein, sondern um die Konfiguration von Hackintosh im Allgemeinen, aber wie Clover einige Schritte dafür unternimmt, werden wir in einem separaten Kapitel beschreiben.

Clover macht nicht alles, was er machen muss, es ist auch etwas Handarbeit erforderlich. Was ist überhaupt der Zweck davon? Der Punkt ist folgender: Der Prozessor läuft im Leerlauf mit minimaler Frequenz bei minimaler Spannung, unter Last wachsen Drehzahl und Spannung. (Warum die Spannung? Und weil die Impulsfront steiler wird, und damit Pegel schneller, geht es schneller von Zustand 0 zu Zustand 1).

UPiP kann auf zwei Arten durchgeführt werden: durch ein spezialisiertes Dienstprogramm, wie z. B.

<http://www.coolbook.se/CoolBook.html> (nicht für moderne Systeme), oder GenericCPUPM, oder sie verstehen den nativen Speedstep, mit freundlicher Unterstützung von MacOS kann man das machen.

Die nächsten Schritte sind notwendig:

1. Die HPET muss zwingend im DSDT korrigiert werden, was erfolgreich durchgeführt wird durch Clover (Kleeblatt) an der Maske 0x0010.

2. Es muss ein korrekter Prozessorteil vorhanden sein, was von Clover unter dem Key GeneratePStates=Yes (und zusätzlich DropSsdT) macht

3. Als Beispiel für Ihr SMBIOS muss ein MacModell ausgewählt werden, für das die EIST-Technologie ist verfügbar. Es stellt sich heraus, dass dies nicht für alle Modelle gilt. K Für das Modell MacBook1.1 funktioniert Speedstep beispielsweise nicht, und für MacBook5.1 wird. Der Absatz 3 kann wie folgt umgedeutet werden: Konfigurieren Sie das Ihr Gerät so ähnlich wie möglich der realen Macs, aber lassen Sie uns seine Plattform-plist so festlegen, dass der Speedstep auftaucht.

Für jedes Modell gibt es eine eigene Plist, siehe hier

System/Library/Extensions/IOPlatformPluginFamily.kext/Contents/PlugIns/
ACPI_SMC_PlatformPlugin.kext/Contents/Resources/MacBook5_1.plist

Sehen Sie die Gemeinsamkeiten und Unterschiede der verschiedenen Listen, und korrigieren Sie Ihre in die richtige Richtung.

```
ConfigArray
<key>ConfigArray</key>
<array>
  <dict>
    <key>WWEN</key>
    <true/>
    <key>model</key>
    <string>MacBook4,1</string>
    <key>Restart-Aktion</key>
    <dict>
      <key>cpu-p-state</key>
      <integer>0</integer>
    </dict>
  </dict>
</array>
```

Dieser Neustart-Aktionsschlüssel bedeutet, in welchen P-State die CPU beim Neustart fallen soll. Nur mit diesem Key funktioniert der Ruhezustand und das Herunterfahren des Rechners!

CtrlLoopArray

```
<key>CtrlLoopArray</key>
<array>
  <dict>
    <key>Description</key>
    <string>SMC_CPU_Control_Loop</string>
    ...
    <key>PLimitDict</key>
    <dict>
      <key>MacBook4,1</key>
```

```
<integer>0</integer>
</dict>
```

Dieser PLimitDict-Schlüssel wurde bereits bei der Erzeugung von P-Zuständen erwähnt. Wiederholen wir es: Diese Einschränkung bewirkt die maximale Geschwindigkeit des Prozessors. 0 - die Geschwindigkeit ist maximal, 1- eine Stufe niedriger als das Maximum. Fehlt dieser Key hier jedoch, bleibt der Prozessor bei der Mindestfrequenz.

CStateDict

```
<key>CStateDict</key>
<dict>
  <key>MacBook4,1</key>
  <string>CSD3</string>
  <key>CSD3</key>
  <dict>
    <key>C6</key>
    <dict>
      <key>Einschalten</key>
      <true/>
```

Die Praxis zeigt, dass es besser ist, den gesamten Abschnitt zu entfernen, damit die Steuerung Spannungsversorgung genau nach PState, nicht nach Cstate funktioniert. Obwohl, es von der eigenen Meinung abhängt, vielleicht diese Option durchzuarbeiten.

Das Symptom ist, dass der Prozessor mit maximaler Frequenz arbeitet und nicht abfällt. Nach dem Entfernen des Abschnitts beginnt in der Häufigkeit zu variieren.

Mit Skylake-Prozessoren gibt es eine neue Methodik zur Verwaltung der CPU-Frequenz, in ihrem eigenen Namen, heißt SpeedShift es. Ich habe es noch nicht herausgefunden, wie es funktioniert, vorläufige Ergebnisse:

Im CPU-Bereich schreiben wir zwei Keys (der Erfinder ist goodwin_c)

```
<key>CPU</key>
<dict>
  <key>HWPEnable</key>
  <true/>
  <key>HWPValue</key>
  <string>0x30002a01</string>
```

Die erste Taste setzt eine 1 in das MSR-Register 0x770.

Die zweite Taste schreibt den angegebenen Wert in das MSR-Register 0x774.

Das Problem mit dem Sleep (Schlaf)

Was ist mit dem Schlafproblem? Wenn alle oben genannten Schritte durchgeführt wurden, wird der Computer ins Bett gehen und wie ein gehorsames Kind aufwachen. Das Wichtigste, was Sie dazu tun müssen hat Clover bereits getan: korrigierte FADT und FACS. Es bleibt nur noch die DSDT zu korrigieren, besorgen Sie sich einen guten Speedstep Kext,, und Sie werden glücklich sein.

Jedes Gerät kann eine gute Nachtruhe stören, auch ein stromloser PCI Gerät, oder teilweise arbeitende Geräte, zum Beispiel AppleHDA. Der Schlaf wird stark durch den NullCPUPM.kext gestört. Möglicherweise benötigen Sie keinen Speedstep, aber Sie sollten den NRET-Patch(?) auf diese Weise erstellen, um den nativen AppleCPUPM auszuführen zu können, und dann wird der NullCPUPM.kex nicht benötigt. Und für diejenigen, die keinen Prozessor haben, der es erlaubt, AppleCPUPM zu verwenden, können Sie SleepEnabler ausprobieren - manchmal hilft das, oder einen gepatchten Kernel.

DSDT hat eine Gruppe von _GPE-Methoden mit Benachrichtigungen pro Gerät, die aus ihrem Schlaf geweckt werden müssen. Der Computer selbst ist wach, aber es kann sich herausstellen, dass er Video/Netzwerk/Ton/Maus vergessen hat aufzuwecken. Siehe DSDT, lernen Sie die Theorie, wie es geht.

[Außerdem gab es ein Problem mit dem Ruhezustand beim UEFI-Boot auf einem 10.8-System.](#)

[Das Problem wurde in der Revision 1942 behoben. Änderung im OsxAptioFixDxe-Treiber: sleep/wakeup funktioniert auch in 10.8, sogar mit CsmVideoDxe.](#) Heutzutage ist dafür ProtectMemoryRegions zuständig.

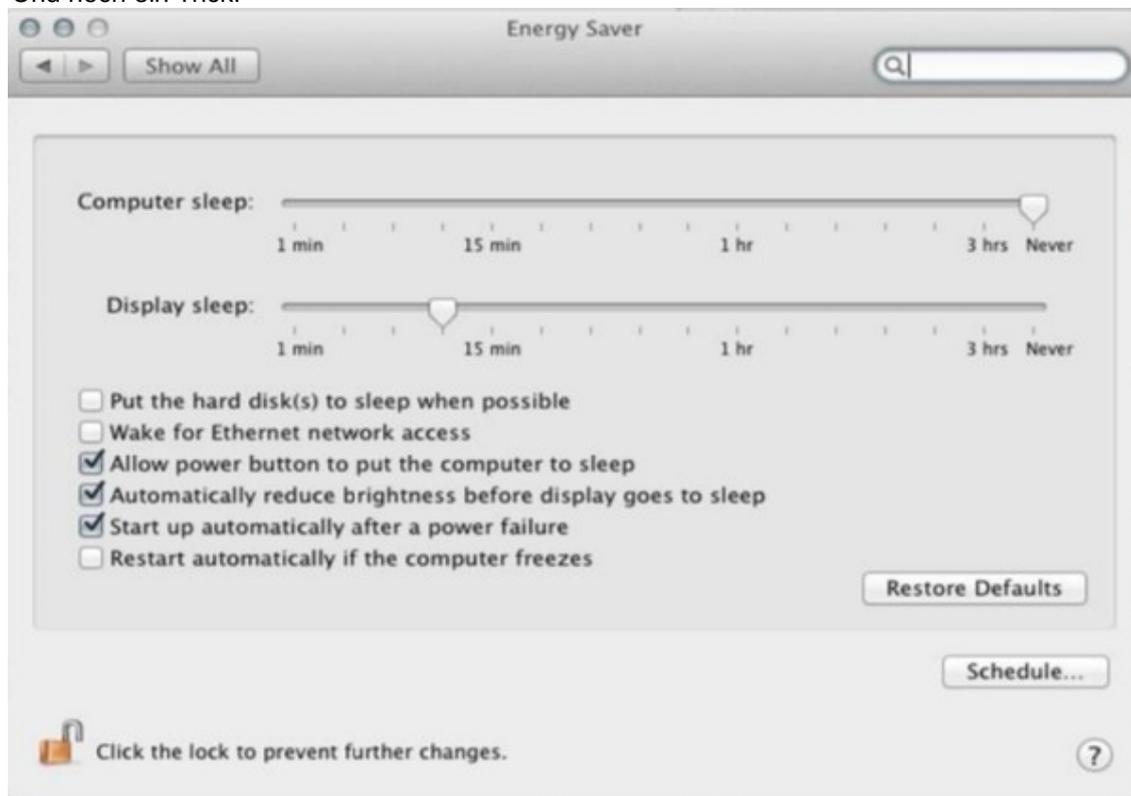
Nächster Trick für UEFI-Boot

```
<key>ACPI</key>
<dict>
  <key>HaltEnabler</key>
  <true/>
</dict>
```

Dies korrigiert den von UEFI-bios fehlinitalisierten Zustand des Chipsatzes. Für Legacy-Downloads macht CloverEFI alles richtig, ich habe dieses Problem dort nicht gesehen.

Symptome – der Computer geht nicht richtig in den Schlaf, Bildschirm geht aus, aber der Lüfter nicht.

Und noch ein Trick.



Ohne das Kontrollkästchen "Automatisch starten..." hatte ich keine Möglichkeit, nach einem Nickerchen aufzuwachen.

Hibernate

Man nennt es auch Tiefschlaf, aber im Allgemeinen ist es eher der klinische Tod. Der Punkt ist, dass das System scheinbar in den Ruhezustand versetzt wird, aber es speichert seinen Zustand in dem sleepimage auf der Festplatte, und schaltet den Computer einfach aus, damit er beim Wiedereinschalten nur seinen Zustand wieder herstellt und er ist aufgewacht. Für Laptops ist dies entscheidend. Wenn im normalen Ruhezustand der Computer nicht vollständig heruntergefahren wird und weiterhin Elektrizität verwendet, wenn auch weniger als im Arbeitszustand, aber immer noch spürbar für die Batterie wird sie während einer gewissen Ruhezeit vollständig entladen sein. Während des Hibernate-Ruhezustand wird die Batterie nicht verwendet, und nur durch seine Leckage entladen wird. Da der Zustand des Computers im Sleepimage gespeichert wird, gehen dennoch keine Daten verloren.

Hibernate funktionierte vor langer Zeit mit Chameleon, aber nur bis zur Version 10.7.2 (glaube ich), dann funktionierte diese Methodik nicht mehr, aufgrund einiger Änderungen im System. Mit Clover wurde es geschafft, einen Hibernate zu machen, aber unter folgenden Bedingungen:

- Laden Sie entweder CloverEFI (Legacy), InsydeEFI oder Phoenix 2.3.1. Im aktuellen Mit der Revision 2915+ wurde der Treiber OsxAptioFix2Drv eingeführt, mit dem hibernate mit AMI UEFI auf einem 10.9.1+ System funktioniert. Aber das System 10.7.5 bootet überhaupt nicht mit dieser Option. Ach, und was soll's!
- Das System ist entweder 10.7.5 oder 10.9.1+. Andere Systeme sind noch nicht aufgewacht.

- Mode 21 oder, noch besser, 29 oder sogar 57, obwohl Apple auf 25 beharrt.

```
sudo pmset -a hibernatemode 29
```

Ab Captain kann man das nicht mehr de 29 benutzen. Der Trick ist folgendermaßen:

- kopieren Sie /Library/Preferences/com.apple.PowerManagement.511CE201-1000-4000-9999-120361221216.plist auf dem Desktop. Es gibt mehrere solcher Dateien, wählen Sie die gewünschte aus, die mit der UUID in den Systeminformationen übereinstimmt. Sie bearbeiten diese Datei, zum Einfügen von 29

```
<key>Hibernate-Mode</key>
<integer>29</integer>
```

Und kopieren Sie die bearbeitete Datei mit Hilfe des Terminals zurück

```
sudo cp ~/Desktop/ com.apple.PowerManagement.511CE201-1000-4000-9999-120361221216.plist
/Library/Preferences/
```

Danach ist ein Neustart erforderlich, erst dann wird die Änderung wirksam.

- Wenn Sie jedoch ein echtes NVRAM laufen haben, können Sie Mod 25 ausführen, dann in der Konfiguration

```
<key>Boot</key>
<dict>
  <key>StrictHibernate</key>
  <true/>
```

- Für System 10.13 und höher müssen Sie Folgendes aktivieren:

```
<key>RtcHibernateAware</key>
<true/>
```

weil der Verschlüsselungsschlüssel in das CMOS geschrieben werden kann, und weil Clover sollte zusätzliche NVRAM-Variablen erzeugen. (über letzteres noch werden sie nachdenken müssen).

Es funktioniert wie folgt:

1. Stellen Sie Mod 29 (oder 25) ein, falls nicht bereits eingestellt. Es ist nicht notwendig, dies zu wiederholen.
2. Versetzen Sie den Computer in den Ruhezustand, entweder über das Menü oder durch Schließen des Deckels oder durch Drücken der Powerknopf, wenn dies so konfiguriert ist. Nach einer Minute wird der Computer vollständig heruntergefahren sein.
3. Um aufzuwachen, schalten Sie es einfach wie gewohnt ein. Sehen Sie den BIOS-Startbildschirm, geben Sie das Menü "Clover". Und hier sehen wir, dass unser System markiert ist (im Winterschlaf)



Bei anderen Systemen ist dies nicht der Fall. Wenn dieses Symbol angeklickt wird, wird das System mit dem markierten Bild in ein paar Sekunden gebootet. Sie können den Fortschritt am unteren Rand sehen, und das System schaltet sich ein. Es ist viel schneller als ein normaler Systemstart, insbesondere bei Laptops, und vor allem bei eine große Anzahl von offenen Anwendungen.

Es ist zu beachten, dass, wenn das Dateisystem des Datenträgers geändert wurde, nachdem sie z. B. von einem System, das von einer zweiten Partition geladen wurde, darauf zugreifen und es verändern, gibt es eine ernsthafte Gefahr der Beschädigung des Dateisystems, da im schlafenden System ein Cache mit anderen Strukturen ist. Für das System 10.9 wird diese Schwierigkeit automatisch überwunden, indem der Vergleich Änderungsdaten stattfindet. Dies funktioniert nicht im System 10.7.5, vergewissern Sie sich manuell, dass alles korrekt ist.

Sie können das Aufwachen aus dem Bild abbrechen, indem Sie die Leertaste auf diesem Symbol drücken, und indem Sie "Ruhezustand abbrechen" wählen.

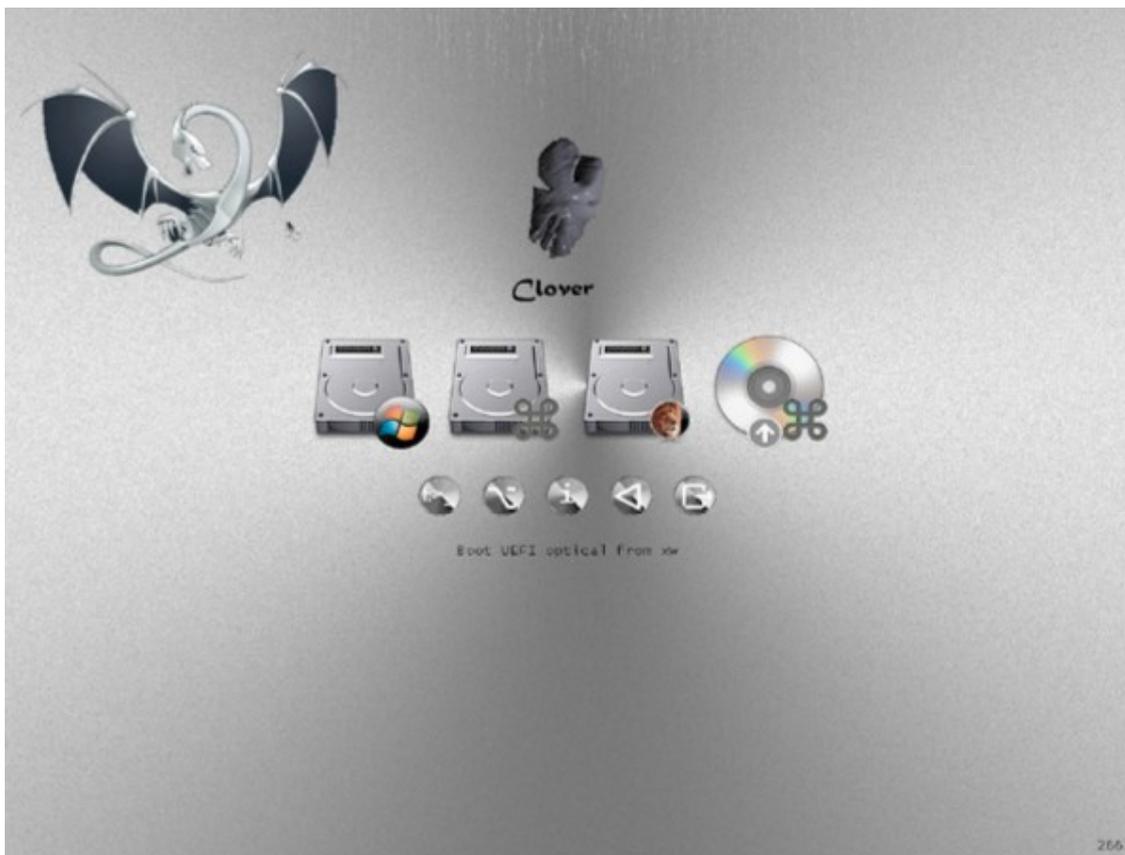
Wenn das System immer noch der Meinung ist, dass es aufwachen sollte, muss es in der Konfiguration:

```
<key>Boot</key>
<dikt>
  <key>NeverHibernate</key>
  <true/>
```

So verwenden Sie Clover

Die erste Begegnung

Booten Sie zunächst in die Clover-GUI und versuchen Sie, für den Anfang hier zu leben, Drücken Sie verschiedene Tasten, bewegen Sie die Maus herum.



Die obere Reihe der Schaltflächen sind die vorgesehenen Betriebssysteme, die gebootet werden können.

In diesem Bild gibt es zwei davon, Lion und Windows, wie Sie auf den Bildern sehen können. Realistisch gesehen, erinnere ich sie daran, dass Clover kein Bootloader für das Betriebssystem ist, sondern ein Manager ihrer eigenen Bootloader. Speziell für den Mac ist der Bootloader /System/Library/CoreServices/boot.efi. Und In diesem Fall für Windows, /EFI/microsoft/boot/bootmgfw.efi

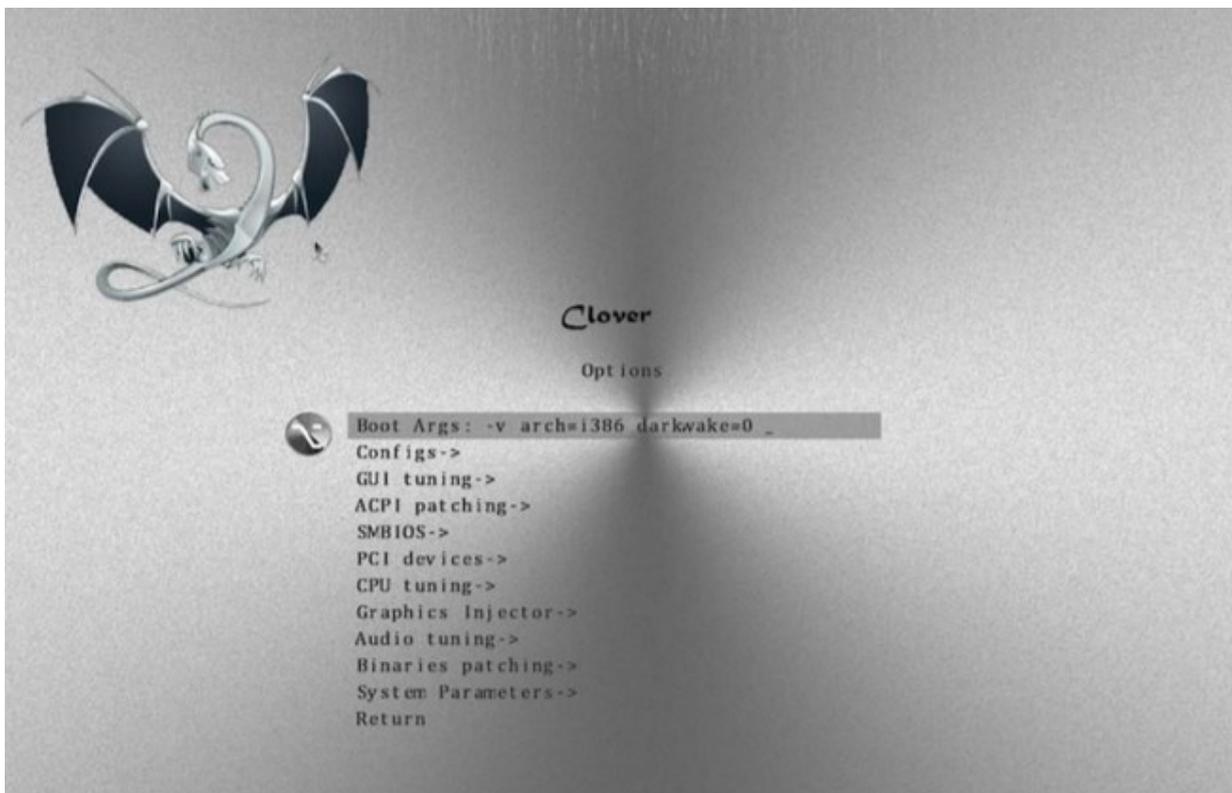
Untere Reihe der Schaltflächen - zusätzliche Funktionen: Befehlszeile (Shell), Menü Optionen, Informationen über den Lader und die Umgebung, starten Sie den Clover neu und beenden Sie ihn. Ausstieg wo? Zurück zur Umwelt. EFI, im UEFI-BIOS bzw. im CloverEFI.

Es ist sehr nützlich, für die erste Einführung F1 zu drücken (wer hätte das gedacht?!). Wenn in der Konfiguration eingestellt

```
<key>GUI</key>  
<dict>  
  <key>Language</key>  
  <string>ru:0</string>
```

wird die Hilfe auf Russisch sein. Die Befehlszeile ist so etwas wie DOS, mit der Möglichkeit Dateien zu kopieren und zu löschen. Wie und wofür, das würde den Rahmen dieses Buches sprengen. Dies ist Shell.efi mit seiner Hilfe.

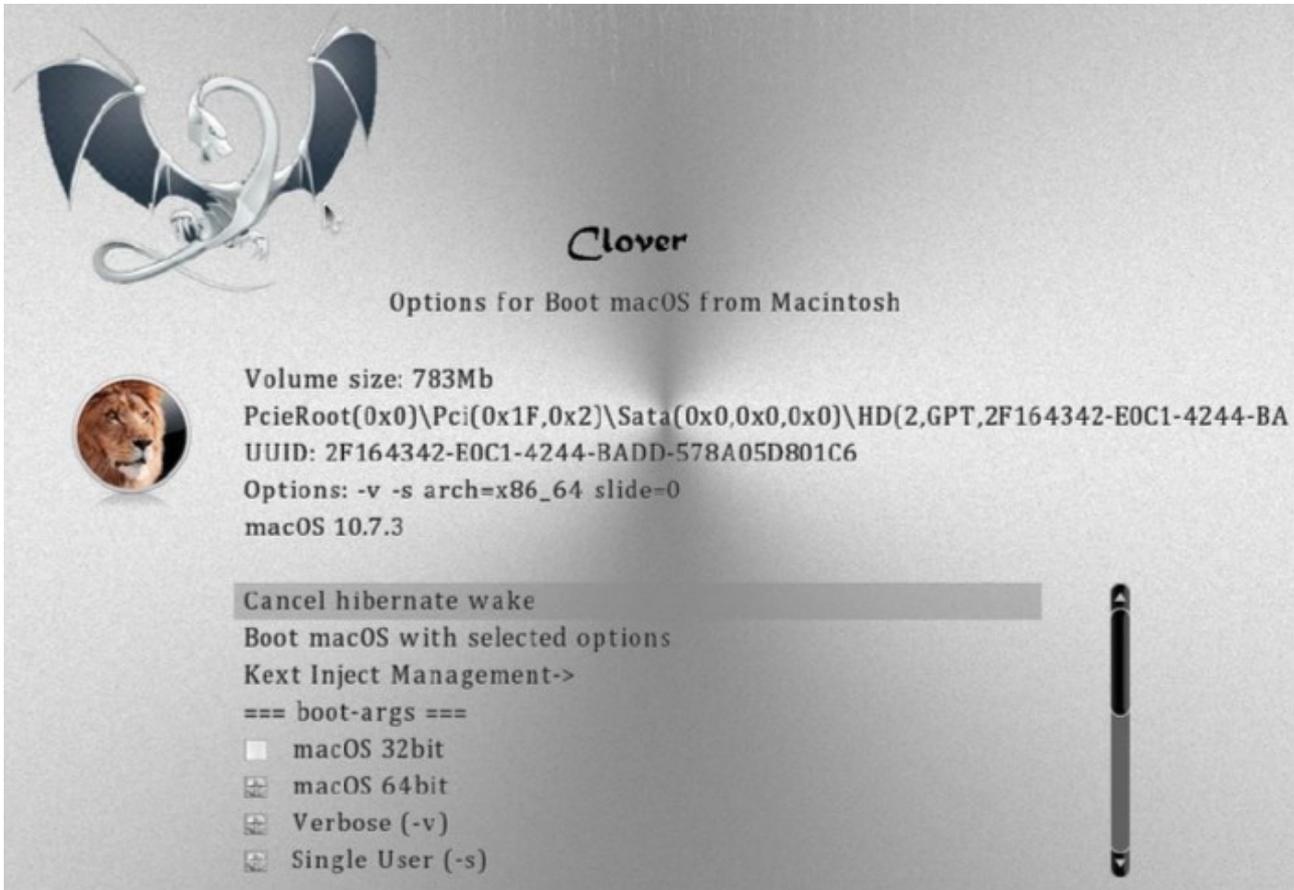
In den Optionsmenüs können Sie einige Einstellungen ändern, die sich auf Ihren Fortschritt auswirken der Systemlast.



Einige von ihnen sind in der Datei config.plist angegeben, aber es könnte durchaus sein, dass dort falsch steht, und um diese Datei noch nicht zu bearbeiten, können die Einstellungen bereits hier im Clover-Menue korrigieren (bei einer unvollständigen Kleeblatt config.plist).

Was genau zu ändern ist und wofür, das hängt von der Aufgabe ab, was sie genau erreichen wollen. Sehr nervige Anfragen wie "Ich habe eine xd Western Digital und Corsair-Speicher, helfen Sie mir beim Einrichten der config". Config wird nicht durch die Hardware konfiguriert, sondern durch das Ergebnis. Wenn Sie nicht sofort booten, versuchen Sie festzustellen, was das Problem sein könnte, und beheben Sie es in diesem Menü.

Mehrere spezielle Boot-Methoden können durch Drücken der Leertaste beim entsprechenden Boot-Icon erreicht werden . (ein erneutes ENTER bewirkt – Systemstart,drücken der SPACE-Taste - Aufruf des Zusatzmenüs für das Booten



Insbesondere ist dies die einzige Möglichkeit, einen unerwünschten Hibernaten (Überwinterung) zu verhindern.

Warum ist Clover so langsam beim Starten?

Einige können nicht einmal den Start abwarten und berichten, dass Clover nicht funktioniert. Schauen wir uns diese Frage einmal genauer an.

1. Das debug.log wird gesetzt:

```
<key>Boot</key>
<dict>
  <key>Debug</key>
  <true/>
</dict>
```

Ja, es ist eine sehr wertvolle Information, um heraus zu bekommen, was nicht funktioniert und warum. Aber bedenken Sie, dass es den Startvorgang wirklich verlangsamt. Wenn Clover auf dem USB installiert ist (um es sozusagen auszuprobieren, und wir nicht wissen, ob Clover überhaupt arbeiten kann, oder?!), dann kann ein Debug-Startup

10 Minuten dauern. Ja, wirklich. Dies liegt daran, dass das Debug-Protokoll öffnet, schließt und wird auf dem Flash-Laufwerk in jeder Zeile überschrieben. Damit ist sichergestellt, dass Sie Informationen über den Erfolg der Inbetriebnahme auch bei einem heftigen Bootproblem. Wenn sie also Debuggen wollen, warten sie! Oder setzen sie den Key auf `<false/>` für den ersten Versuch.

2. Es gibt zu viele Festplatten, Partitionen und Dateien auf ihrem System. Clover muss, um das Startmenü zu erstellen, alle Festplatten, alle Partitionen und alle Dateien auf ihnen durchforsten, um herauszufinden, welche Systeme zur Ausführung angeboten werden können. Na warte! Entweder den Scanvorgang abbrechen,

```
<key>GUI</key>
<dict>
  <key>Scan</key>
  <dict>
    <key>Entries</key>
    <false/>
    <key>Legacy</key>
    <false/>
    <key>Tool</key>
    <false/>
  </dict>
</dict>
```

und stellen Sie ein Menü manuell zusammen

```
<key>GUI</key>
<dict>
  <key>Custom</key>
  <dict>
    <key>Entries</key>
    <array>
      <dict>
```

Zugegeben, es bedarf einiger geistiger Anstrengung, um herauszufinden, was man dort schreiben soll. Entweder Belassen Sie den Bootloader-Scan vorerst, um wenigstens irgendwo zu booten.

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

3. Eine riesige Windows- oder Linux-Partition, und vielleicht auch noch mehrere. Die Windows-Partition wird mit dem NTFS-Treiber gescannt. Und in dieser Partition gibt es normalerweise eine Million Dateien, und unter ihnen suchen wir nach bootmgr.efi. Ich würde empfehlen, Windows so zu installieren, dass sich diese Datei auf der EFI-Partition befindet, und den NTFS.efi-Treiber komplett entfernt und somit keine Windows-Partitionen gescannt werden. Ähnlich verhält es sich mit der Linux-Partition und dem Treiber VBoxExt2.efi.

4. **Zu viele Treiber im Ordner** /EFI/CLOVER/drivers/

Ich sehe eine Situation voraus, in der Leute anfangen, ihre eigenen Treiber dieses Typs zu produzieren und bereit, sie auszuprobieren. Überlegen Sie an dieser Stelle, ob Sie die zusätzlichen Treiber benötigen Tastatur, Maus, LowMemoryFix... Der Punkt ist, dass, wenn Sie von usbb ausgehen, das Lesen all diese Dateien Zeit in Anspruch nehmen kann.

5. **Nicht unterstützte Maus.**

Leider werden nicht alle Mäuse von dem uns vorliegenden EFI-Treiber unterstützt. Schlecht kann es sein, dass sich die Maus auf dem Bildschirm nicht korrekt verhält, oder dass sie stark bremst. Verbieten Sie es zur Inspektion, oder sogar dauerhaft, wenn es falsch ist

```
<key>Mouse</key>
<dict>
  <key>Aktiviert</key>
  <false/>
  <key>Geschwindigkeit</key>
  <integer>0</integer>
</dict>
```

6. **Langsamer HFS+-Treiber.**

Im Lieferumfang des offiziellen Clover ist der Treiber VboxHFS.efi enthalten, der sauber lizenziert ist und links versteht, aber langsamer als Apples HFSplus.efi ist. Herunterladen wo-