

Erledigt

SSDT in die DSDT integrieren und Power Nap Nutzen.

Beitrag von „thomaso66“ vom 24. Mai 2014, 12:54

Hier mal eine Anleitung wie man sich eine SSDT erstellt und diese dann in die DSDT Integrieren kann, und damit [Power Nap](#) in der Systemeinstellung freizuhalten, je nach System Definition sie dazu den Link, oder am ende der Anleitung.

Benötigte Programme/Tools

maciASL gibt es hier im [DLC](#) oder ein anderer DSDT Editor eurer Wahl.

ssdtPRGen.sh von Piker-Alpha

Im Terminal geben wir dieses ein, das läd die aktuelle Revision und speichert sie im Benutzer Verzeichnis.

Code

```
1. curl -o ~/ssdtPRGen.sh https://raw.githubusercontent.com/Piker-Alpha/ssdtPRGen.sh/master/ssdtPRGen.sh
```

Danach müssen wir das Shell Strippt noch ausführbar machen (+x), dieses geht mit dem Terminal Befehl

Code

```
1. chmod +x ~/ssdtPRGen.sh
```

Die Hilfe Information läst sich via aufrufen

Code

1. ~/ssdtPRGen.sh -h

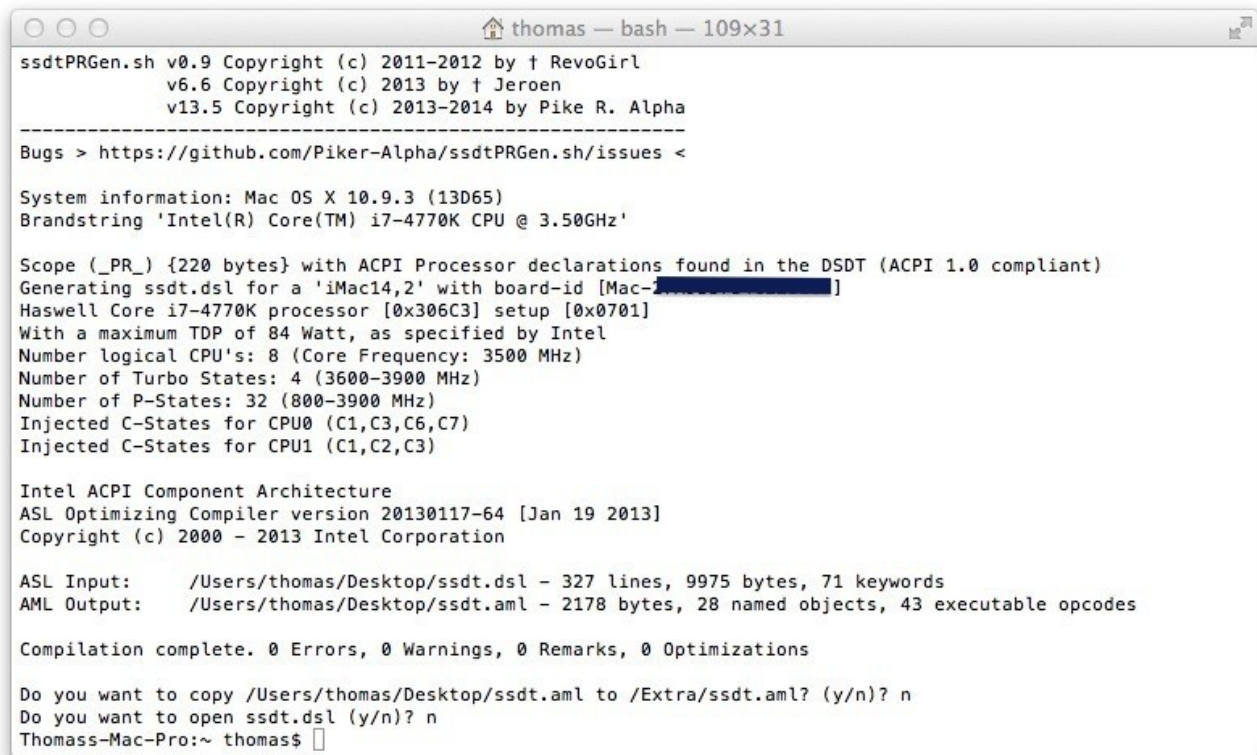
für diejenigen Interessant die Übertakten möchten, für uns an dieser stelle nicht.

So nun erstellen wir uns unsere eigene SSDT passend zum Rechner in dem wir das Skript einfach Starten

Code

1. ~/ssdtPRGen.sh

Da es Root rechte erfordert geben wir nun unser Passwort ein, die beiden fragen beantworten wir einfach mit n (nein) danach sollte sich auf dem Desktop/Schreibtisch eine ssdt.aml und eine ssdt.dsl befinden.



```
thomas -- bash -- 109x31
ssdtPRGen.sh v0.9 Copyright (c) 2011-2012 by † RevoGirl
              v6.6 Copyright (c) 2013 by † Jeroen
              v13.5 Copyright (c) 2013-2014 by Pike R. Alpha
-----
Bugs > https://github.com/Piker-Alpha/ssdtPRGen.sh/issues <

System information: Mac OS X 10.9.3 (13D65)
Brandstring 'Intel(R) Core(TM) i7-4770K CPU @ 3.50GHz'

Scope (_PR_) {220 bytes} with ACPI Processor declarations found in the DSDT (ACPI 1.0 compliant)
Generating ssdt.dsl for a 'iMac14,2' with board-id [Mac-1422000000000000]
Haswell Core i7-4770K processor [0x306C3] setup [0x0701]
With a maximum TDP of 84 Watt, as specified by Intel
Number logical CPU's: 8 (Core Frequency: 3500 MHz)
Number of Turbo States: 4 (3600-3900 MHz)
Number of P-States: 32 (800-3900 MHz)
Injected C-States for CPU0 (C1,C3,C6,C7)
Injected C-States for CPU1 (C1,C2,C3)

Intel ACPI Component Architecture
ASL Optimizing Compiler version 20130117-64 [Jan 19 2013]
Copyright (c) 2000 - 2013 Intel Corporation

ASL Input:      /Users/thomas/Desktop/ssdt.dsl - 327 lines, 9975 bytes, 71 keywords
AML Output:     /Users/thomas/Desktop/ssdt.aml - 2178 bytes, 28 named objects, 43 executable opcodes

Compilation complete. 0 Errors, 0 Warnings, 0 Remarks, 0 Optimizations

Do you want to copy /Users/thomas/Desktop/ssdt.aml to /Extra/ssdt.aml? (y/n)? n
Do you want to open ssdt.dsl (y/n)? n
Thomass-Mac-Pro:~ thomas$
```

Nun öffnen wir unsere DSDT.aml und die ssdt.aml im DSDT Editor unserer Wahl ich nutze wie eingangs schon geschrieben maciASL

In der SSDT.aml brauchen wir den Kopf nicht sondern nur den Score (_PR.CPU0) bis Score (_PR.CPU7) im meinen Fall siehe Bilder

```

9 * Signature          5307
10 * Length            0x00000882 (2178)
11 * Revision          0x81
12 * Checksum         0xB0
13 * OEM ID           "APPLE "
14 * OEM Table ID     "CpuPm"
15 * OEM Revision     0x00013500 (79104)
16 * Compiler ID     "INTL"
17 * Compiler Version 0x20130117 (538116375)
18 */
19 DefinitionBlock ("IASLOHTIAs.aml", "SSDT", 1, "APPLE ", "CpuPm", 0x00013500)
20 {
21     External (\_PR_.CPU7, DeviceObj)
22     External (\_PR_.CPU6, DeviceObj)
23     External (\_PR_.CPU5, DeviceObj)
24     External (\_PR_.CPU4, DeviceObj)
25     External (\_PR_.CPU3, DeviceObj)
26     External (\_PR_.CPU2, DeviceObj)
27     External (\_PR_.CPU1, DeviceObj)
28     External (\_PR_.CPU0, DeviceObj)
29
30     Scope (\_PR.CPU0) ← von hier
31     {
32         Method (_INI, 0, NotSerialized)
33         {
34             Store ("ssdtPRGen version.....: 13.5 / Mac OS X 10.9.3 (13065)", Debug)
35             Store ("target processor.....: i7-4770K", Debug)
36             Store ("running processor.....: Intel(R) Core(TM) i7-4770K CPU @ 3.50GHz", Debug)
37             Store ("baseFrequency.....: 800", Debug)
38             Store ("frequency.....: 3500", Debug)
39             Store ("busFrequency.....: 100", Debug)
40             Store ("logicalCPUs.....: 8", Debug)
41             Store ("maximum TDP.....: 84", Debug)
42             Store ("packageLength.....: 32", Debug)
43             Store ("turboStates.....: 4", Debug)
44             Store ("maxTurboFrequency.....: 3900", Debug)
45             Store ("machdep.xcpm.mode.....: 1", Debug)
46         }
47     }
48     Name (APLF, Zero)
49     Name (APSN, 0x04)
50     Name (APSS, Package (0x20)
51     {

```

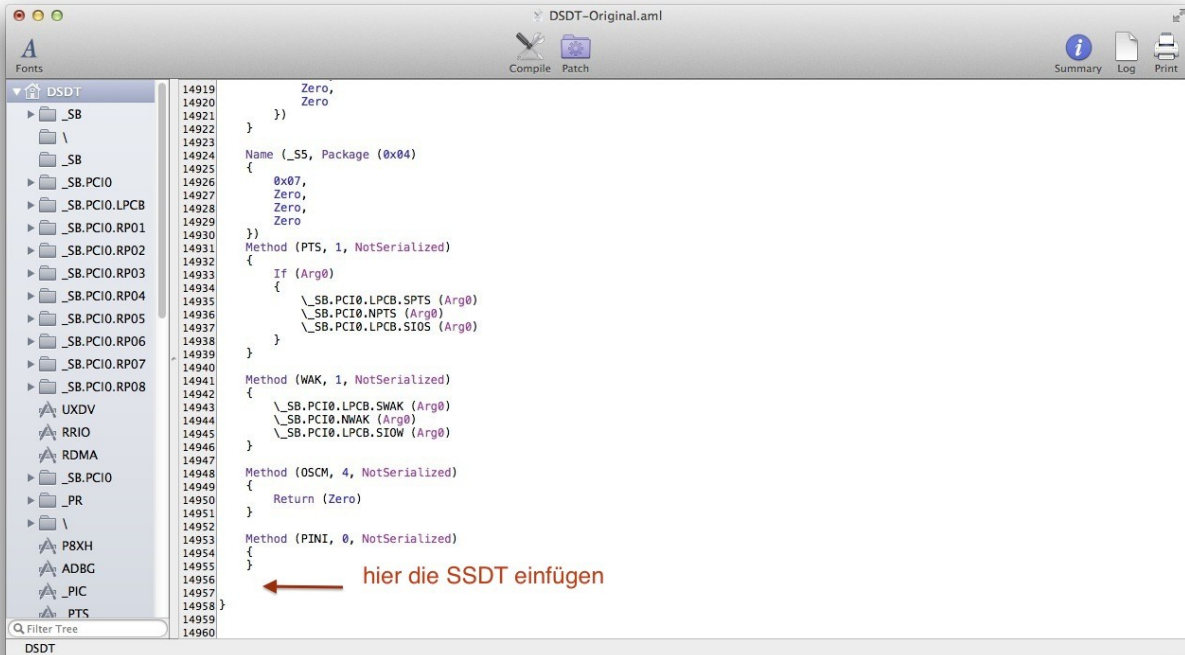
```

583     Method (APSS, 0, NotSerialized)
584     {
585         Store ("Method _PR_.CPU5.APSS Called", Debug)
586         Return (\_PR.CPU0.APSS)
587     }
588     Method (ACST, 0, NotSerialized)
589     {
590         Return (\_PR.CPU1.ACST ())
591     }
592 }
593
594
595     Scope (\_PR.CPU6)
596     {
597         Method (APSS, 0, NotSerialized)
598         {
599             Store ("Method _PR_.CPU6.APSS Called", Debug)
600             Return (\_PR.CPU0.APSS)
601         }
602
603         Method (ACST, 0, NotSerialized)
604         {
605             Return (\_PR.CPU1.ACST ())
606         }
607     }
608
609     Scope (\_PR.CPU7)
610     {
611         Method (APSS, 0, NotSerialized)
612         {
613             Store ("Method _PR_.CPU7.APSS Called", Debug)
614             Return (\_PR.CPU0.APSS)
615         }
616
617         Method (ACST, 0, NotSerialized)
618         {
619             Return (\_PR.CPU1.ACST ())
620         }
621     }
622 }
623
624 }

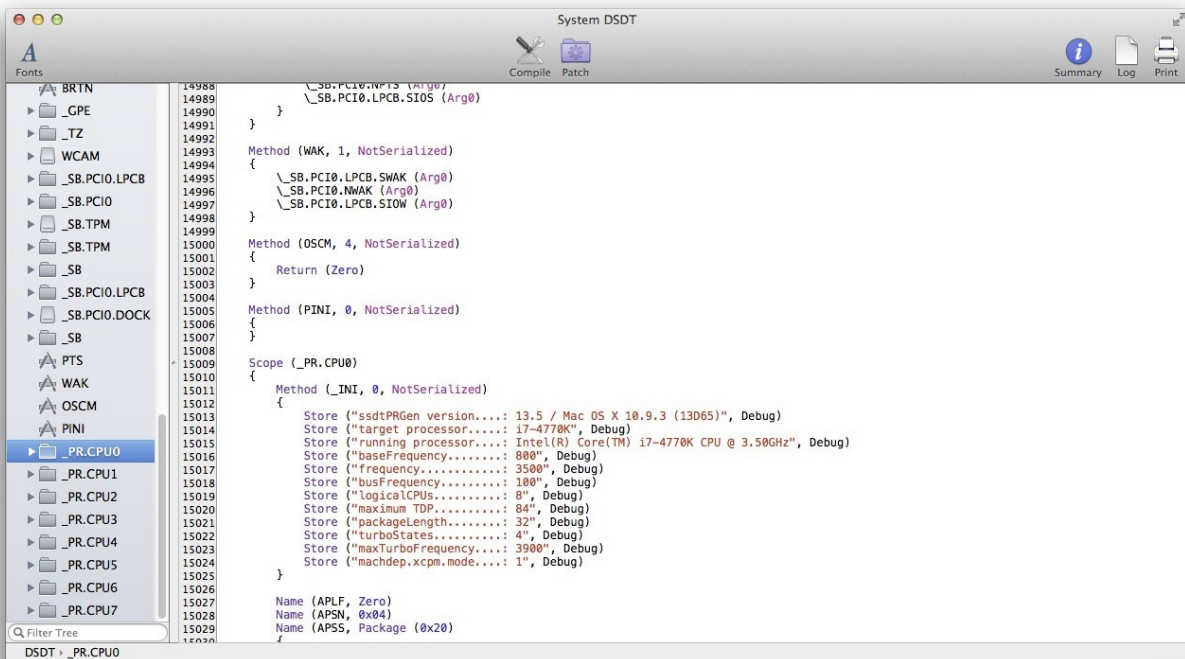
```

In unser DSDT gehen wir nun ganz an Ende, denn hier muss die SSDT eingefügt werden, siehe

Bild



nach dem einfügen sollte das dann so aussehen

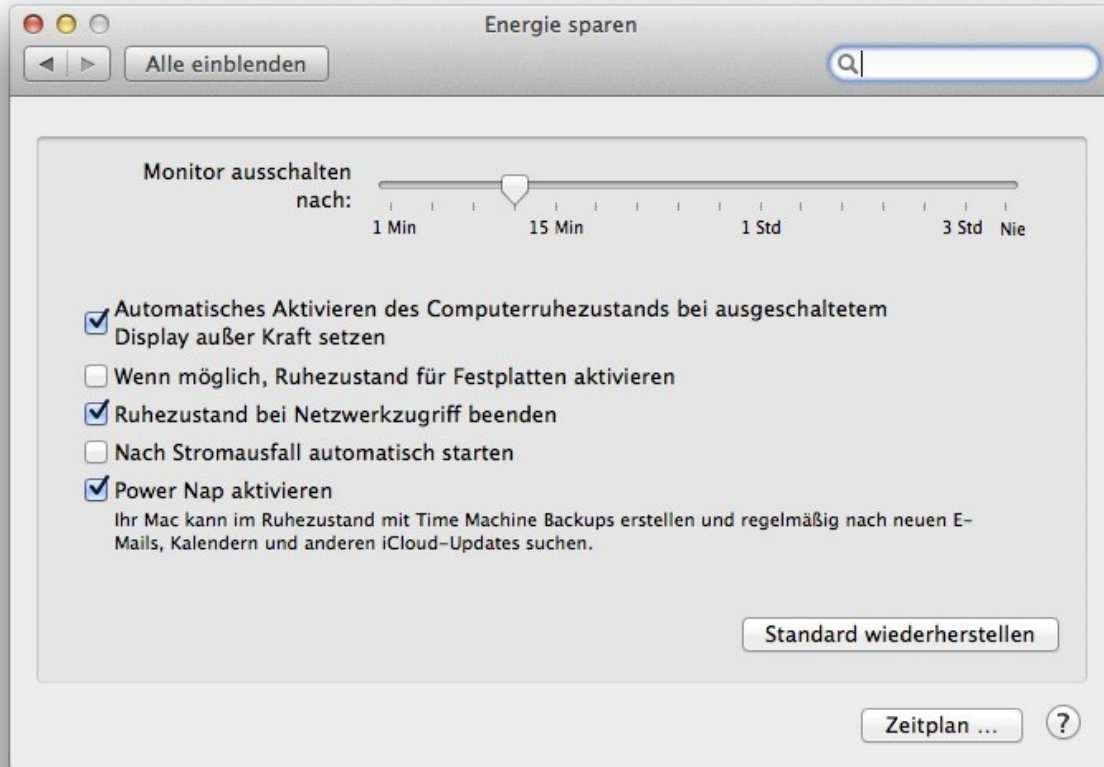


So das war es schon, wenn wir alles Richtig gemacht haben, lässt sie sich einfach als DSDT.aml abspeichern und wie gewohnt einsetzen.

Der ein oder andere fragt sich warum sollte ich mir die Arbeit machen? Lauft doch auch so ohne SSDT, das ist Richtig nur mit lauft es halt noch besser 😄 und man kann dann auch in der System Einstellung unter Energie Power Nap wie beim echten Mac Nutzen, sollte man seinen Rechner mit der Passenden Systemdefinition betreiben, meine beiden Rechner laufen als iMac 14,2 moglich ist es laut der oben verlinkten Seite mit folgenden Modellen.

Modell
MacBook Air (Anfang 2014)
MacBook Air (Mitte 2013)
MacBook Air (Mitte 2012)
MacBook Air (Mitte 2011)
MacBook Air (Ende 2010) mit OS X Mountain Lion 10.8.2 oder neuer
MacBook Pro (Retina, 13", Ende 2012 und neuer)
MacBook Pro (Retina, Mitte 2012)
MacBook Pro (Retina, 15", Anfang 2013 und neuer)
Mac mini (Ende 2012)
iMac (21,5", Ende 2012 und neuer)
iMac (27", Ende 2012 und neuer)
Mac Pro (Ende 2013)

So sieht das dann in der System Einstellung aus.



Wer dieses nun in sein BIOS Integrieren möchte dazu habe ich ja hier schon eine sehr einfache [Anleitung](#) geschrieben, bietet sich für Ozmosis Nutzer auf jeden fall an.

So das war es auch schon wieder, vielleicht hilft es ja dem ein oder anderen Leser.