

So you want to construct your own DSDT (Differentiated System Description Table)

It is my hope that this short document outlining a method of DSDT construction is both entertaining and instructional.

While I strive for perfection, a character flaw for sure, alas it is not attainable so please drop me a note for corrections or suggestions for improvement.

I have written this down but it is not really my own. So many intelligent, patient and definitely obsessive individuals have gone before dropping tidbits here and there, writing code and applications, researching devices, motherboards, kernel extensions and the rest to make this document possible that I simply can take no credit for it.

Most of what we do to make our machines function near-normal is the hard work of countless individuals and while it may seem less than enough, I would like to say 'Thank You'.

You guys rock!

REMOVE ANY DSDT from your boot sequence; this can be accomplished on a temporary basis by adding **DSDT=/Extra/xxxx.aml** to your boot string.

Boot into Windows. If your Bios needs it, update it now; as of this writing the A12 Bios is current for the XPS L702x. Once the update is complete, boot into the Mac OS **overriding your DSDT** if you have one specified in the com.apple.Boot.plist file; if you do not prevent the old DSDT from loading, it will overwrite your newly installed Bios update and the updates will be lost to your Mac OS (should this occur, boot into Windows and then back into the Mac OS).

Run **/Developer/Applications/Utilities/IORegistryExplorer** and save to your desktop.

Download **DSDTSE**: <http://www.osx86.es/?p=610>

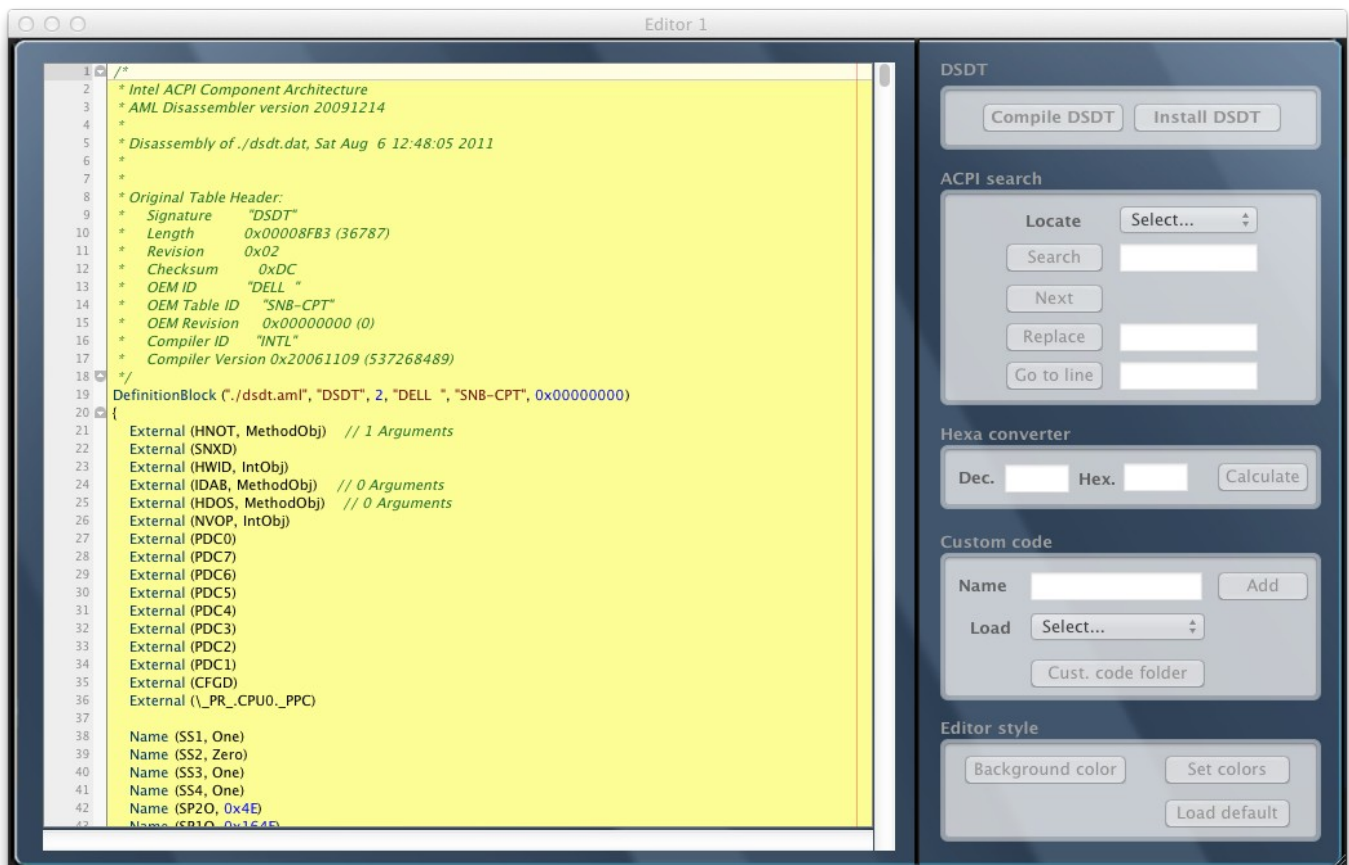
Download **P4V Client** (Macintosh): http://www.perforce.com/downloads/complete_list

Download **iASLME** (compiler): <http://cvad-mac.narod2.ru/iaslme/>

Start the **DSDTSE** application.



This is the opening dialog of the DSDTSE application. The button toward the bottom says 'Extract DSDT'; press it once to open the window shown on the next page.



Press the window <CLOSE> button, press <SAVE>, override the default output location to **Desktop** and override the default file name to **dsdt_vanilla_Ann** where **Ann** is your Bios level (A10, A12 or whatever).

The next two pages show the **iASLME** compiler dialog window (3 views) and the **Finder** window showing the compiler's output **Sessions** folder.

Start the **iASLME** application; drop your vanilla DSDT from the desktop onto the DOCK ICON of the compiler to get an initial result. There should now be a **Sessions** folder on your desktop; copy your vanilla DSDT source into the subfolder (date/time) of the **Sessions** folder containing the dsdt_vanilla_Ann.aml file so you have a record of source and output for every version built. If necessary, you can always rebuild the source (.dsl) from the binary (.aml) using **DSDTSE**.

The first **iASLME** window shown below is the result from the compilation; note it shows 1 error.

The second window is the same dialog scrolled back to the beginning showing some stats about the machine's environment as well as the current boot's kernel flags. This is one place that we can verify we have successfully overridden the DSDT; note it says **DSDT=/Extra/xxxx.aml** was a kernel flag so our override is in effect.

The third window is scrolled back toward the bottom and shows the 1 error's messages.

iaslMe

^ (_SRS)

Elapsed time for major events

Miscellaneous compile statistics

```

18891 : Parse nodes
3638  : Namespace searches
 930  : Named objects
 312  : Control methods
 650  : Memory Allocations
3682571 : Total allocated memory
    0 : Constant subtrees folded

```

ASL Input: /Users/Dad1525/Desktop/dsdt_vanilla_A12.dsl - 10016 lines, 316494 bytes, 3954 keywords

Compilation complete. 1 Errors, 4 Warnings, 12 Remarks, 11 Optimizations

All done.

Enjoy ...

Quit

iaslMe

```

-- iaslMe v1.31 @ cVad 2011 --
      Mac OS X Lion ready
- (De)Compiling ASL <=> AML files -
- ASL Optimizing Compiler version 20110527-64 [May 27 2011] inside -
      cvad-mac.narod2.ru
      www.applelife.ru

```

```

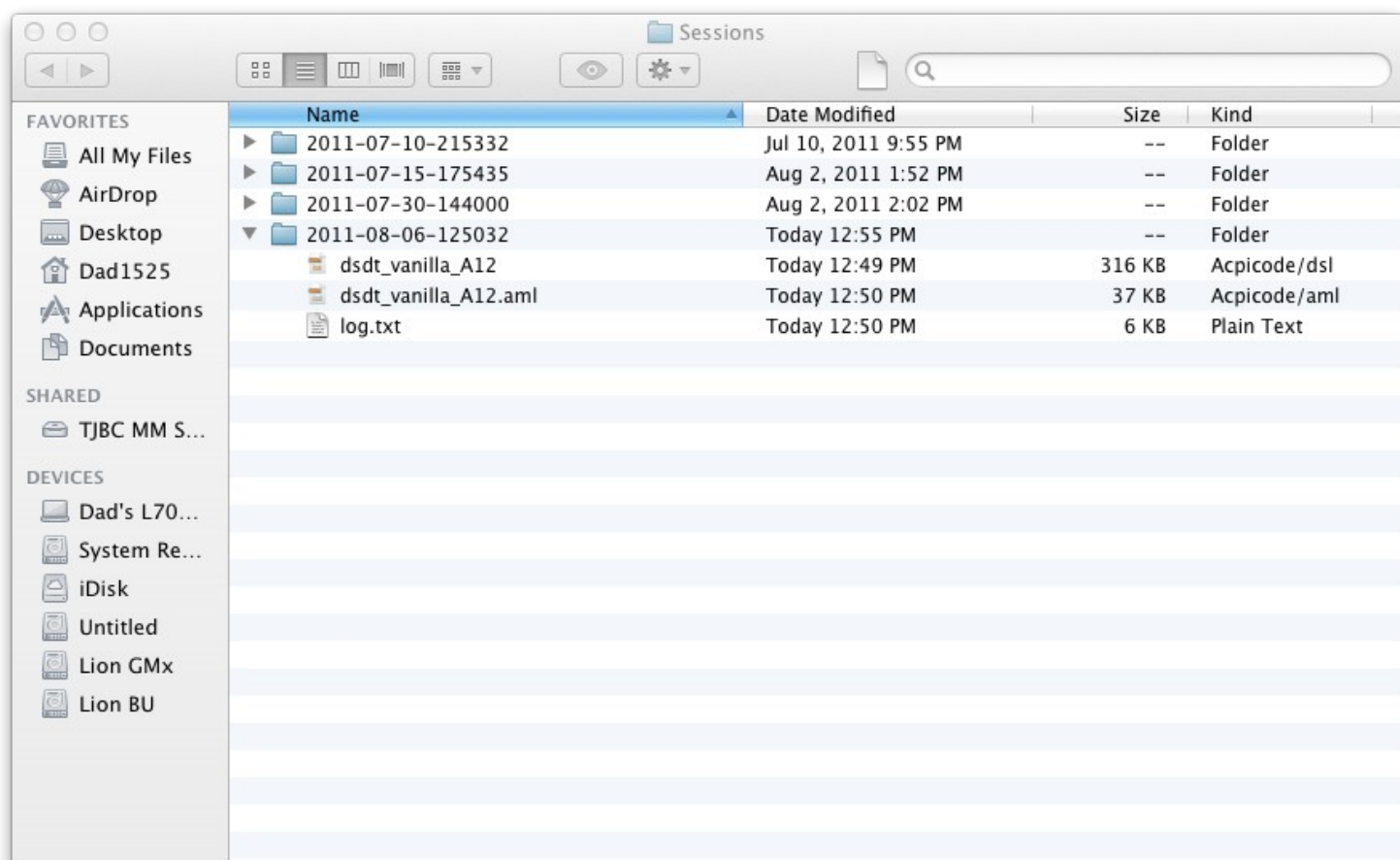
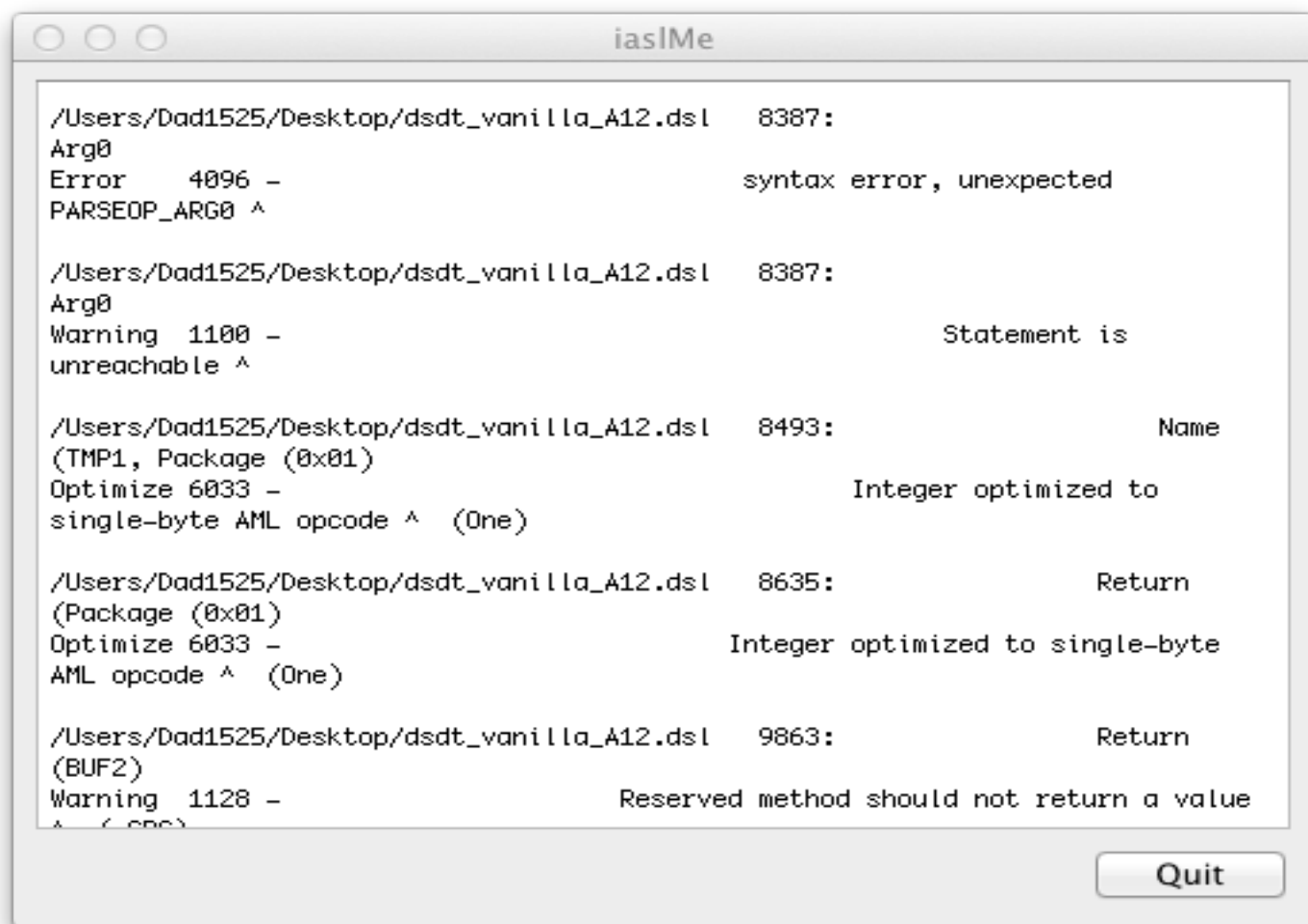
ProductName:  Mac OS X ProductVersion: 10.7 BuildVersion: 11A511
Kernel:       Darwin Kernel Version 11.0.0: Sat Jun 18 12:56:35 PDT 2011
Bootargs:     boot-uuid=6C3CF812-3962-349F-9CC5-7FE075B26000 rd=*uuid
busratio=20 DSDT=/Extra/xxxx.aml
Model ID:     MacBookPro8,3
CPU TYPE:     Intel(R) Core(TM) i7-2630QM CPU @ 2.00GHz
Core:         1995(1995) MHz x 20.1(20.1) Bus: 99 MHz FSB: 396 MHz
RAM:          8192 Mb
SwapUsage:    total = 64.00M used = 0.00M free = 64.00M (encrypted)
HibernateMode: 0
User:         Dad1525 on Dads-L702x-MacBookPro.local

```

Start working: 2011-08-06 12:50:31 -0500

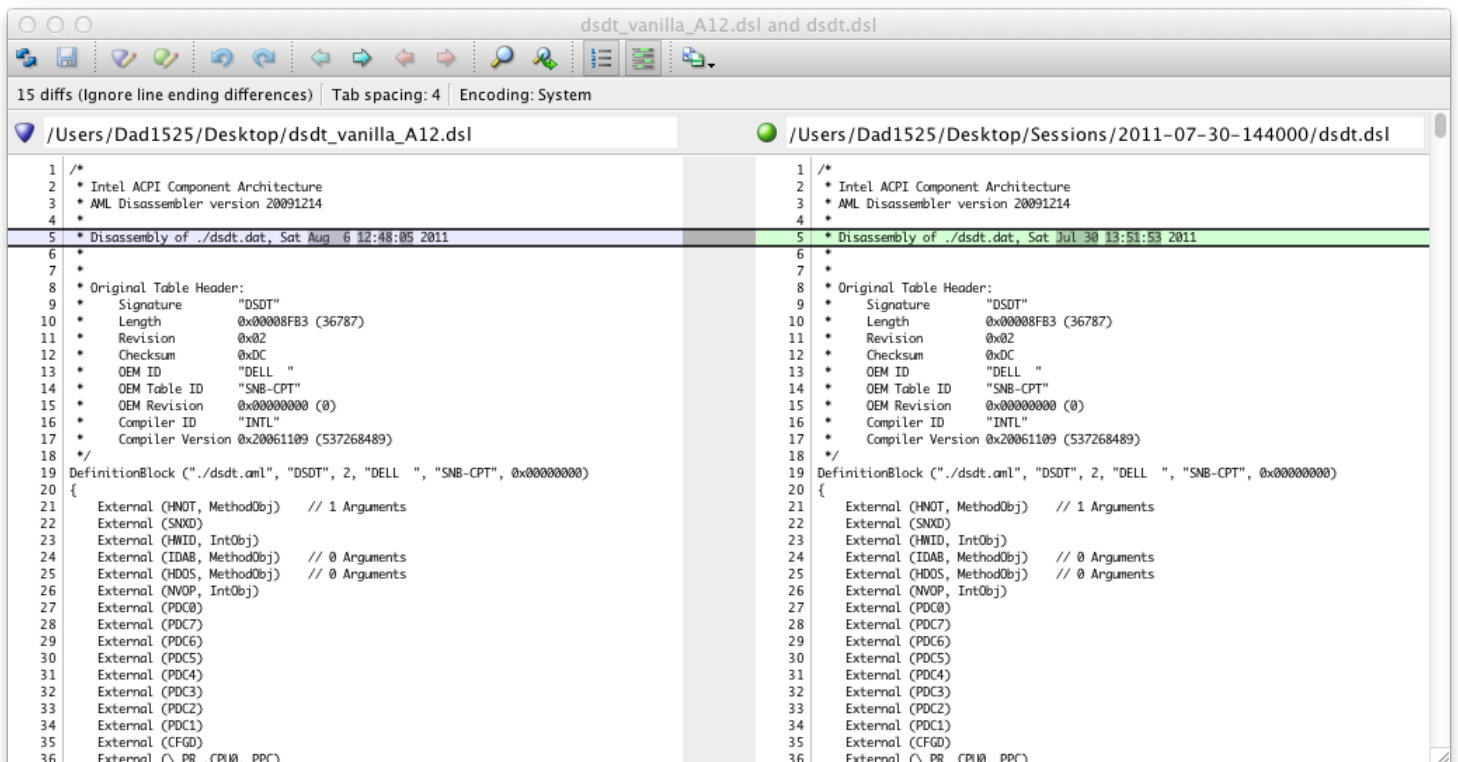
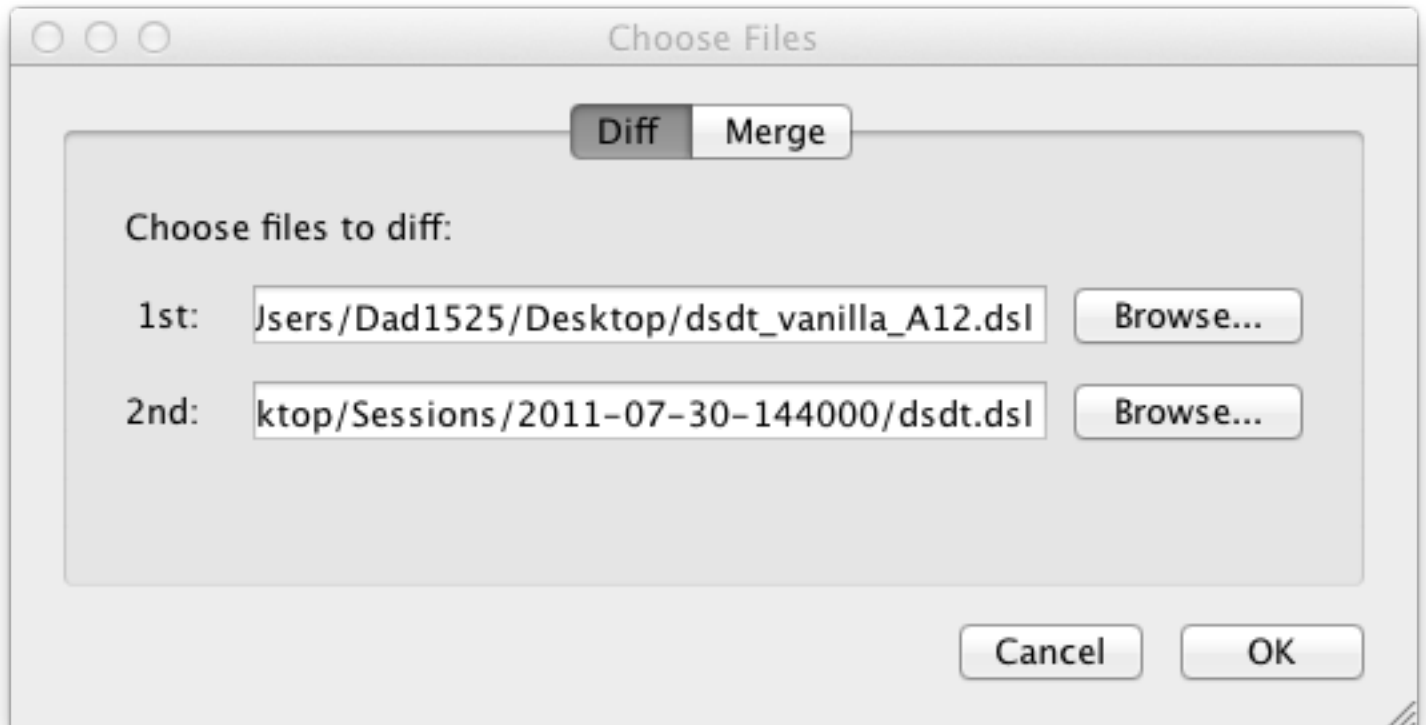
Detected ... MAC OS X "Lion".

Quit



Using **p4merge**, open your vanilla DSDT and my A12 out of the Steve's Update and observe the differences. There maybe several hundred; relax ... all but about 10 of those are cosmetic and we won't be doing anything with them.

NOTE: You are opening these files for potential edit; if your intent is to actually modify them, I would copy the originals first and open the copies. If you accidentally modify your original source, you can introduce some very difficult-to-find errors.



Once open, note that **p4merge** indicates how many differences there are (upper left) and displays two windows with the differences high-lighted; the left window is represented by a bluish-purple triangle and the right window is represented by a green circle. There is a row of icons at the top of the screen; five from the right is the search icon. Move focus to the window you wish to search, press the search icon, enter the search term and press 'Next' or 'Previous' to locate. An unfortunate anomaly removes the window indicator while actually searching so you must remove focus from the search window to see which window was searched; I move the search window so I don't lose it behind the editor window.

In the side-by-side editor display, note the differences are highlighted. This high-lighting has two-levels; the first high-lights both the line that is different and the actual difference in that line; the second just high-lights the line. To modify this setting, press the icon second from the right.

The comments at the top of the code itself are informational, placed there by the disassembly process, and need no modification.

The next screen, shown below, has been advanced to the line number of our compiler error. In this case, unused arguments have been left in the code; this compiler will ignore them. However, as zero errors are better than one (can help to avoid confusion later), just comment them out. Commenting is often better than removing; you just never know.

To edit the source displayed in the left window, press the icon third from the left, a bluish-purple triangle with a pencil.

```
8373     }
8374
8375     Method (_DSM, 4, NotSerialized)
8376     {
8377         If (LEqual (BRID, One))
8378         {
8379             If (LEqual (Arg0, Buffer (0x10))
8380             {
8381                 /* 0000 */ 0xF8, 0xD8, 0x86, 0xA4, 0xDA, 0x0B, 0:
8382                 /* 0008 */ 0xA7, 0x2B, 0x60, 0x42, 0xA6, 0xB5, 0:
8383             })
8384             {
8385                 P8XH (Zero, 0xF5)
8386                 Return (NVOP)
8387             }
8388             Arg0
8389             Arg1
8390             Arg2
8391             Arg3
8392         }
8393     }
8394
8395     If (LEqual (Arg0, Buffer (0x10))
8396     {
8397         /* 0000 */ 0x75, 0x0B, 0xA5, 0xD4, 0xC7, 0x65, 0xF7,
8398         /* 0008 */ 0xBF, 0xB7, 0x41, 0x51, 0x4C, 0xEA, 0x02,
8399     })
8400     {
8401         Store (0x42, NVDR)
8402         Return (NV3D (Arg0, Arg1, Arg2, Arg3))
8403     }
8404
8405     Return (0x80000001)
8406 }
8407 }

8473     }
8474
8475     Method (_DSM, 4, NotSerialized)
8476     {
8477         If (LEqual (BRID, One))
8478         {
8479             If (LEqual (Arg0, Buffer (0x10))
8480             {
8481                 /* 0000 */ 0xF8, 0xD8, 0x86, 0xA4, 0xDA, 0x0B, 0:
8482                 /* 0008 */ 0xA7, 0x2B, 0x60, 0x42, 0xA6, 0xB5, 0:
8483             })
8484             {
8485                 P8XH (Zero, 0xF5)
8486                 Return (NVOP)
8487             }
8488             Arg0
8489             Arg1
8490             Arg2
8491             Arg3
8492         }
8493     }
8494
8495     If (LEqual (Arg0, Buffer (0x10))
8496     {
8497         /* 0000 */ 0x75, 0x0B, 0xA5, 0xD4, 0xC7, 0x65, 0xF7,
8498         /* 0008 */ 0xBF, 0xB7, 0x41, 0x51, 0x4C, 0xEA, 0x02,
8499     })
8500     {
8501         Store (0x42, NVDR)
8502         Return (NV3D (Arg0, Arg1, Arg2, Arg3))
8503     }
8504
8505     Return (0x80000001)
8506 }
8507 }
```


A third display opens to show the results of any actions taken; before we start editing, let's examine a couple of things. First, note that in the third display, the left hand bar shows all differences in this comparison indicated by the dash-line; to move quickly, we can grab the white current-line indicator and use it as a gross positioning to the next difference. Note also that each difference has a triangle and a circle on the right; those give us the option of including or excluding the difference(s) of the indicated source into the copy being edited.

The screenshot shows a software interface for comparing two DSDT files. The top window is titled "dsdt_vanilla_A12.dsl and dsdt.dsl" and displays two side-by-side files. The left file is located at "/Users/Dad1525/Desktop/dsdt_vanilla_A12.dsl" and the right file is at "/Users/Dad1525/Desktop/Sessions/2011-07-30-144000/dsdt.dsl". Both files contain ACPI code, including a table header and a definition block. The bottom window shows a single file at "/Users/Dad1525/Desktop/dsdt_vanilla_A12.dsl" with a list of differences on the left margin. The differences are indicated by a dash-line and a white current-line indicator. The code in the files is as follows:

```
1 /*
2 * Intel ACPI Component Architecture
3 * AML Disassembler version 20091214
4 *
5 * Disassembly of ./dsdt.dat, Sat Aug 6 12:48:05 2011
6 *
7 * Original Table Header:
8 * Signature "DSDT"
9 * Length 0x00008FB3 (36787)
10 * Revision 0x02
11 * Checksum 0xDC
12 * OEM ID "DELL "
13 * OEM Table ID "SNB-CPT"
14 * OEM Revision 0x00000000 (0)
15 * Compiler ID "INTL"
16 * Compiler Version 0x20061109 (537268489)
17 */
18
19 DefinitionBlock ("./dsdt.aml", "DSDT", 2, "DELL ", "SNB-CPT", 0x00000000)
20 {
21     External (HN0T, MethodObj) // 1 Arguments
22     External (SN0D)
23     External (HWID, IntObj)
24     External (IDAB, MethodObj) // 0 Arguments
25     External (HD0S, MethodObj) // 0 Arguments
26     External (NW0P, IntObj)
27     External (PDC0)
28     External (PDC7)
29     External (PDC6)
30     External (PDC5)
31     External (PDC4)
32     External (PDC3)
33     External (PDC2)
```

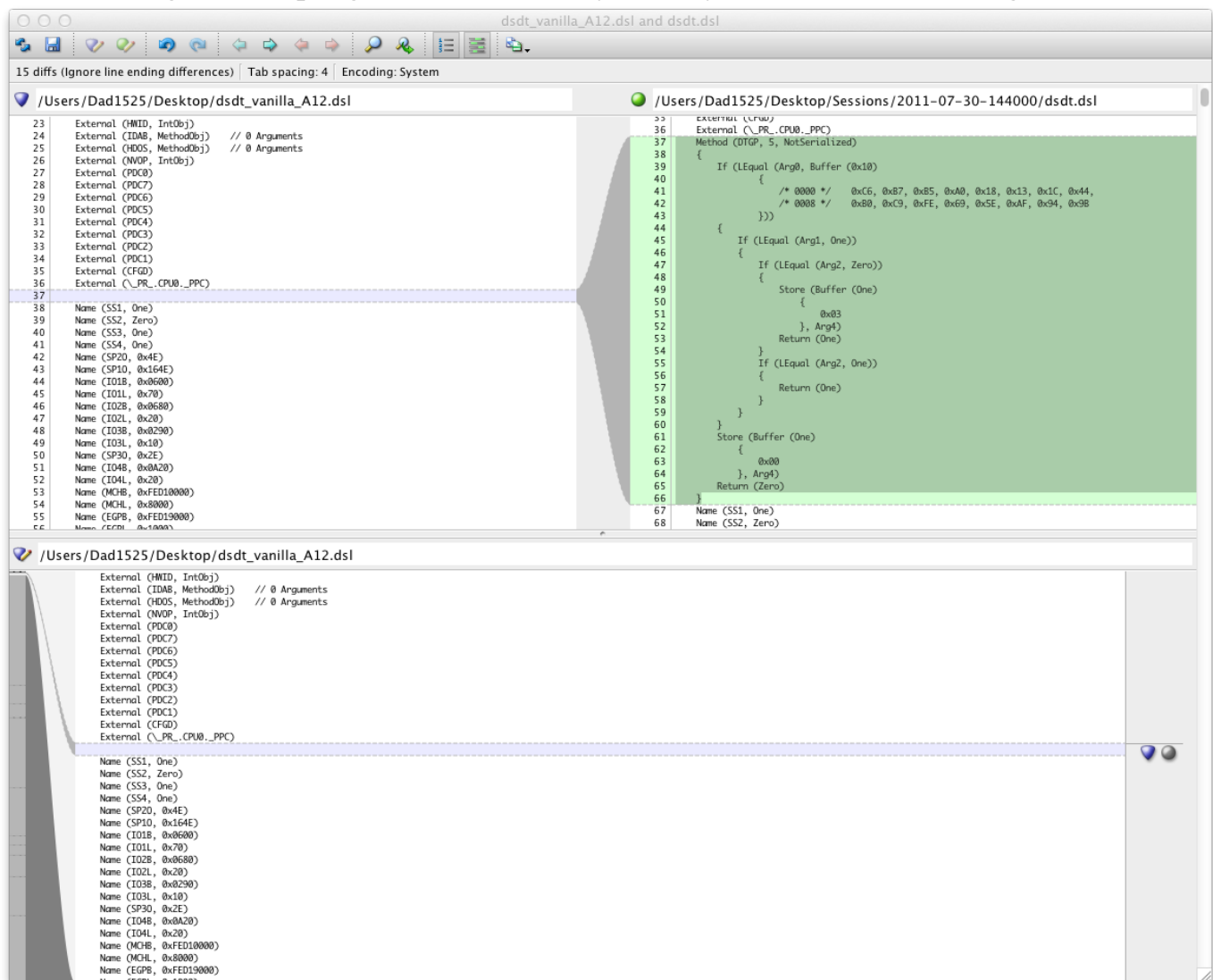
The differences between the two files are as follows:

- Line 5: "Disassembly of ./dsdt.dat, Sat Aug 6 12:48:05 2011" vs "Disassembly of ./dsdt.dat, Sat Jul 30 13:51:53 2011"
- Line 16: "Compiler Version 0x20061109 (537268489)" vs "Compiler Version 0x20061109 (537268489)"
- Line 19: "DefinitionBlock ("./dsdt.aml", "DSDT", 2, "DELL ", "SNB-CPT", 0x00000000)" vs "DefinitionBlock ("./dsdt.aml", "DSDT", 2, "DELL ", "SNB-CPT", 0x00000000)"
- Line 21: "External (HN0T, MethodObj) // 1 Arguments" vs "External (HN0T, MethodObj) // 1 Arguments"
- Line 22: "External (SN0D)" vs "External (SN0D)"
- Line 23: "External (HWID, IntObj)" vs "External (HWID, IntObj)"
- Line 24: "External (IDAB, MethodObj) // 0 Arguments" vs "External (IDAB, MethodObj) // 0 Arguments"
- Line 25: "External (HD0S, MethodObj) // 0 Arguments" vs "External (HD0S, MethodObj) // 0 Arguments"
- Line 26: "External (NW0P, IntObj)" vs "External (NW0P, IntObj)"
- Line 27: "External (PDC0)" vs "External (PDC0)"
- Line 28: "External (PDC7)" vs "External (PDC7)"
- Line 29: "External (PDC6)" vs "External (PDC6)"
- Line 30: "External (PDC5)" vs "External (PDC5)"
- Line 31: "External (PDC4)" vs "External (PDC4)"
- Line 32: "External (PDC3)" vs "External (PDC3)"
- Line 33: "External (PDC2)" vs "External (PDC2)"

This is the first difference we need to include in our new DSDT. The **DGTP** method is the 'injector' for the **_DSM** methods which we are going to add to our DSDT to enable the MAC OS to recognize our devices. Every DSDT must have this as it is referenced in a number of modifications; it can be most anywhere as long as you don't change the 'scope'. I have placed it first, many others place it last in the DSDT; physical position in the source does not matter, 'scope' does.

Methods? Scope? For this limited discussion, let's think of 'methods' as boxes and 'scope' as a location; a shelf in a particular closet. E.G.: The **HDEF** method is in a box on shelf **PCI0** in closet **_SB** so **/_SB.PCI0** is the scope (location) of the box **HDEF**. If we were to place the **DTGP** box there, other methods on other shelves and/or in other closets would not have access as they would not know where to look. As you go through the DSDT, you will recognize many boxes of the same name. 'Scope' makes them different by pointing to their location; by pointing to the shelf they're on and the closet they're in.

OK, we have exhausted my 'extensive' knowledge of this so it is time to move on. Just do not change the scope; got it? Besides, if you do, you'll know soon enough.



```
15 diffs (Ignore line ending differences) Tab spacing: 4 Encoding: System

/Users/Dad1525/Desktop/dsdt_vanilla_A12.dsl
23 External (HWID, IntObj)
24 External (IDAB, MethodObj) // 0 Arguments
25 External (HDOS, MethodObj) // 0 Arguments
26 External (NWOP, IntObj)
27 External (PDC0)
28 External (PDC7)
29 External (PDC6)
30 External (PDC5)
31 External (PDC4)
32 External (PDC3)
33 External (PDC2)
34 External (PDC1)
35 External (CFG0)
36 External (\_PR_.CPU0_PPC)
37
38 Name (SS1, One)
39 Name (SS2, Zero)
40 Name (SS3, One)
41 Name (SS4, One)
42 Name (SP20, 0x4E)
43 Name (SP10, 0x164E)
44 Name (IO1B, 0x0600)
45 Name (IO1L, 0x70)
46 Name (IO2B, 0x0680)
47 Name (IO2L, 0x20)
48 Name (IO3B, 0x0290)
49 Name (IO3L, 0x10)
50 Name (SP30, 0x2E)
51 Name (IO4B, 0x0A20)
52 Name (IO4L, 0x20)
53 Name (MCHB, 0xFED10000)
54 Name (MCHL, 0x8000)
55 Name (EGPB, 0xFED19000)
56 Name (CEPB, 0x1000)

/Users/Dad1525/Desktop/Sessions/2011-07-30-144000/dsdt.dsl
33 External (HWID)
36 External (\_PR_.CPU0_PPC)
37 Method (DTGP, 5, NotSerialized)
38 {
39     If (LEqual (Arg0, Buffer (0x10))
40     {
41         /* 0000 */ 0xC6, 0x07, 0x05, 0xA0, 0x1B, 0x13, 0x1C, 0x44,
42         /* 0008 */ 0xB0, 0xC9, 0xFE, 0x69, 0x5E, 0xAF, 0x94, 0xB8
43     })
44     {
45         If (LEqual (Arg1, One))
46         {
47             If (LEqual (Arg2, Zero))
48             {
49                 Store (Buffer (One)
50                 {
51                     0x03
52                 }, Arg4)
53                 Return (One)
54             }
55             If (LEqual (Arg2, One))
56             {
57                 Return (One)
58             }
59         }
60     }
61     Store (Buffer (One)
62     {
63         0x00
64     }, Arg4)
65     Return (Zero)
66 }
67 Name (SS1, One)
68 Name (SS2, Zero)
```

The following three images demonstrate how a little research can help you weed out unnecessary changes. Making and/or leaving unnecessary DSDT modifications in the source not only makes future modifications more difficult but it can also lead to obscure, intermittent errors and/or flakey device operation.

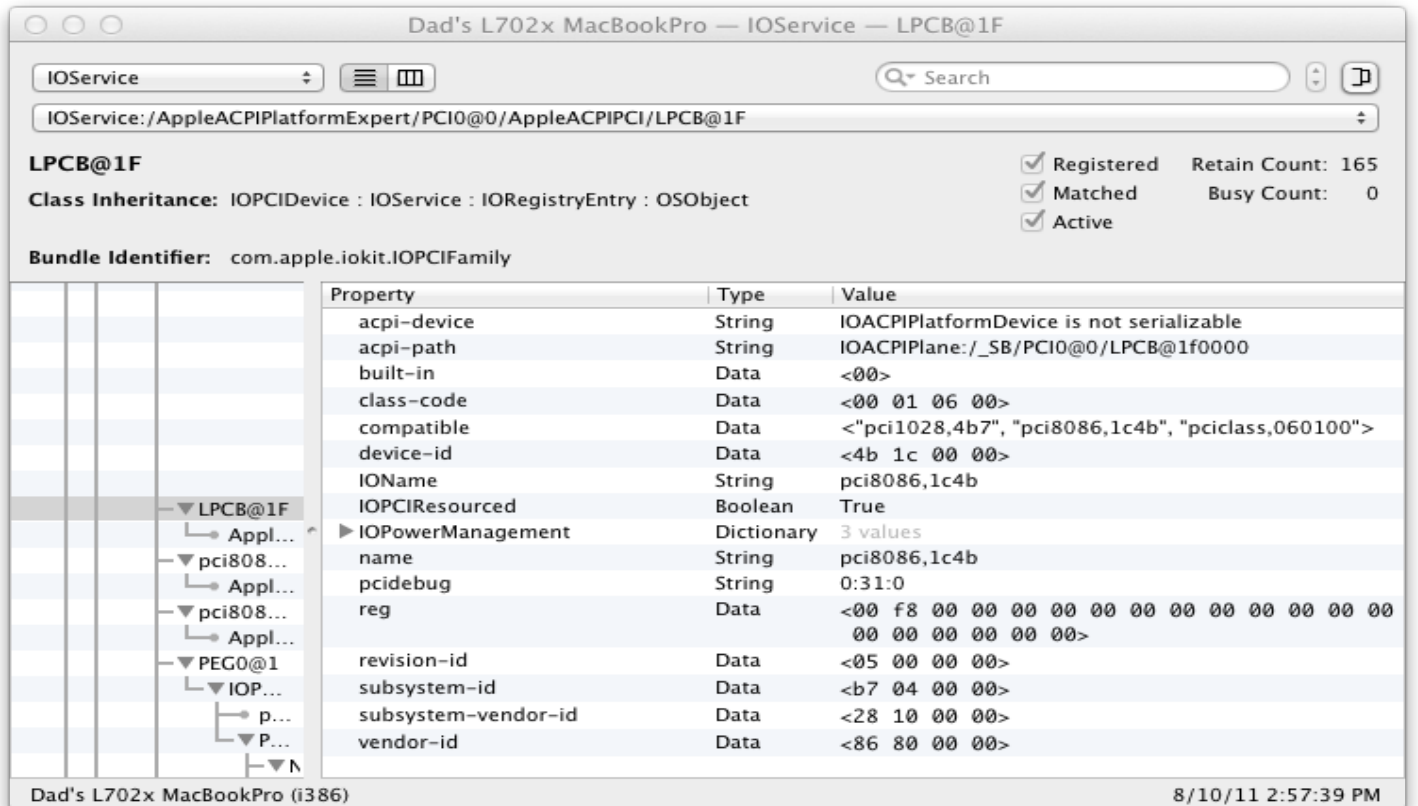
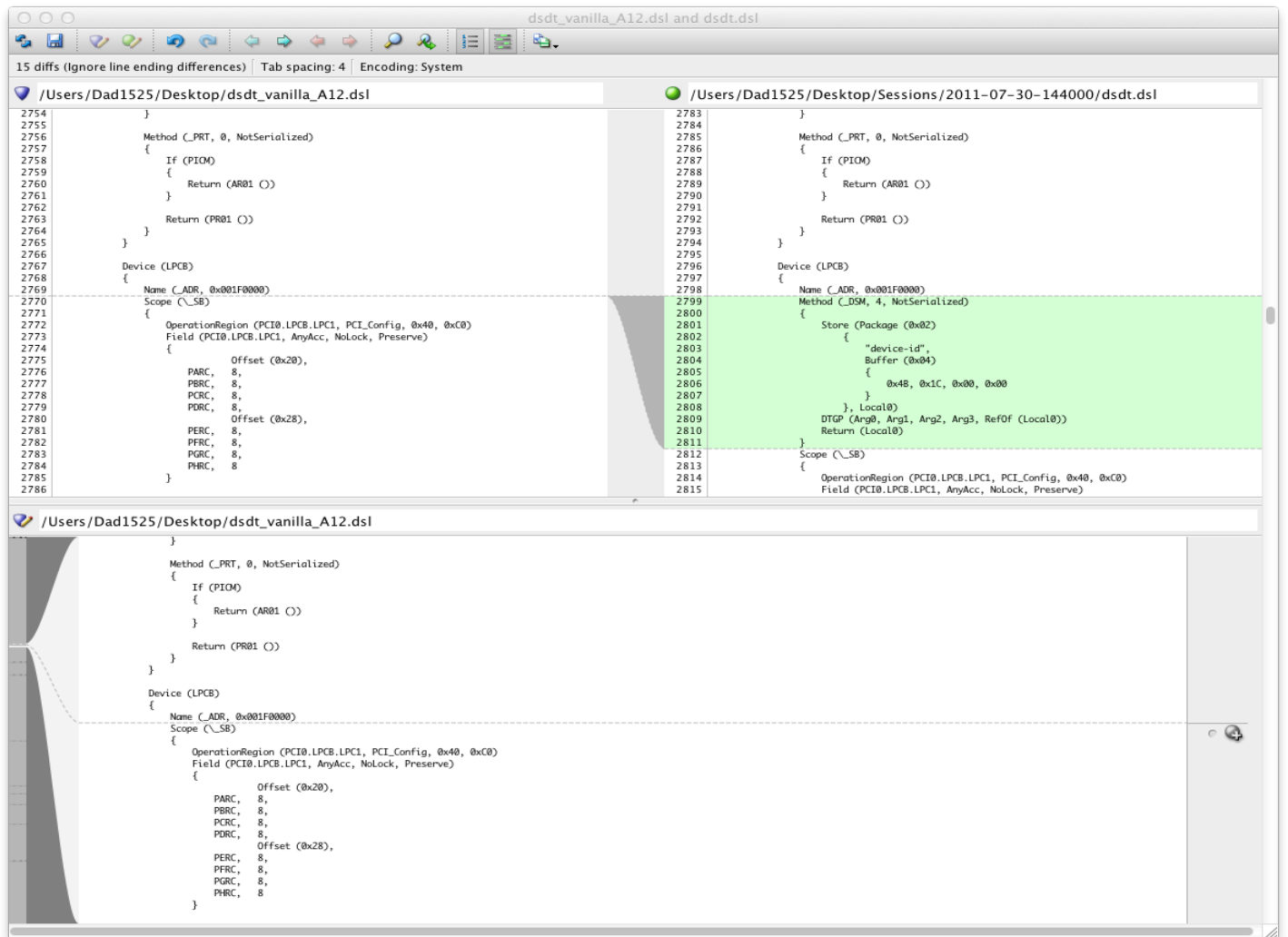
The first image indicates a change that might need to be made in order to support the **LPCB** device for the XPS L702x. This change was introduced from the DSDT used by the doomed L701x which died on day 20 of Dell's 21-day return policy. The change to prompt the MAC OS to support this device consists of injecting a **Property** named *device-id* with a value of 0x00001c4b (Data <4b 1c 00 00>) into the **LPCB** box on the **PCI0** shelf in the **_SB** closet.

OK, so technically the **LPCB** is a *device* not a *method*; let's just go ahead and think of *devices* as somewhat bigger boxes in which one or more smaller *method* boxes are occasionally kept ...

Anyway, in the second (and third) image, we are checking the IORegistryExplorer output which was previously saved (I know you did *not* skip this step, right? Well good for you!) and it indicates that the **LPCB** device is currently being recognized by the MAC OS and that the **Property** which we are injecting, *device-id*, is already present. As this device is already supported, we should not choose this modification.

I must admit here that this change is still in the DSDT's I have previously posted. I simply hadn't taken the time to check until now; sigh. Oh well, better late than never ...

The third image simply demonstrates a way to get to the information we need without paging through a lot of unneeded stuff trying to find the **LPCB** device. Just type the term for which you are looking into the IORegistryExplorer Search box and you get a clean result with unwanted information suppressed. Either way works but the second will reveal duplicates to you more clearly, when they exist.



Dad's L702x MacBookPro — IOService — LPCB@1F

IOService

Q- LPCB

IOService:/AppleACPIPlatformExpert/PCI0@0/AppleACPIPCI/LPCB@1F

LPCB@1F

Class Inheritance: IOPCIDevice : IOService : IORegistryEntry : OSObject

Bundle Identifier: com.apple.iokit.IOPCIFamily

▼ Root

└─ MacBookPro8,3

└─ AppleACPIPlatformExpert

└─ PCI0@0

└─ AppleACPIPCI

└─ **LPCB@1F**

Registered

Matched

Active

Retain Count: 41

Busy Count: 0

| Property | Type | Value |
|---------------------|------------|---|
| acpi-device | String | IOACPIPlatformDevice is not serializabl |
| acpi-path | String | IOACPIPlane:/_SB/PCI0@0/LPCB@1f000 |
| built-in | Data | <00> |
| class-code | Data | <00 01 06 00> |
| compatible | Data | <"pci1028,4b7", "pci8086,1c4b", "pcicss,060100"> |
| device-id | Data | <4b 1c 00 00> |
| IOName | String | pci8086,1c4b |
| IOPCIResourced | Boolean | True |
| ▼ IOPowerManagement | Dictionary | 3 values |
| ChildrenPowerState | Number | 0x2 |
| CurrentPowerState | Number | 0x2 |
| MaxPowerState | Number | 0x2 |
| name | String | pci8086,1c4b |
| pcidebug | String | 0:31:0 |
| reg | Data | <00 f8 00> |
| revision-id | Data | <05 00 00 00> |
| subsystem-id | Data | <b7 04 00 00> |
| subsystem-vendor-id | Data | <28 10 00 00> |
| vendor-id | Data | <86 80 00 00> |

Dad's L702x MacBookPro (i386)

8/6/11 1:18:03 PM

This is the final example; it illustrates a type difference which you will run into and will, almost without exception, NOT include in your new DSDT. You will note that the changed line contains an address and that this address is the only difference in the line. Most often, this is the result of different machines and or/devices included with those machines. This is why you really want to start with your vanilla DSDT source (.dsl) and add the changes you need instead of just installing a binary (.aml) from another person.

It is likely that blindly including this change in your DSDT would result in a Kernel Panic quite possibly rendering your MAC OS useless until corrected. **Should this occur, boot into Windows and then back into MAC OS while overriding your DSDT with the boot parm `DSDT=/Extra/xxxx.aml` and then back out the change immediately.**

While unlikely, it is actually possible to damage your machine via improper DSDT edits, a power anomaly, for example, so these edits should not be taken lightly.

```
dsdt_vanilla_A12.dsl and dsdt_vanilla.dsl
6 diffs (Ignore line ending differences) | Tab spacing: 4 | Encoding: System

/Users/Dad1525/Desktop/dsdt_vanilla_A12.dsl
132 name (NVUW, zero)
133 OperationRegion (QWVS, SystemMemory, 0xBF79EF98, 0x000A)
134 Field (QWVS, AnyAcc, Lock, Preserve)
135 {
136     OVRS, 8,
137     WFI1, 16,
138     WFI2, 16,
139     WIFS, 8,
140     BEMQ, 8,
141     QCKS, 8,
142     BSBF, 8,
143     MSRC, 8
144 }
145
146 OperationRegion (QWVS, SystemMemory, 0xBF741E18, 0x01AF)
147 Field (QWVS, AnyAcc, Lock, Preserve)
148 {
149     OSYS, 16,
150     SMIF, 8,
151     PRM0, 8,
152     PRM1, 8,
153     SCIF, 8,
154     PRM2, 8,
155     PRM3, 8,
156     LCKF, 8,
157     PRM4, 8,
158     PRM5, 8,
159     P800, 32,
160     LTDS, 8,
161     PWRS, 8,
162     DBG5, 8,
163     THOF, 8,
164     ACT1, 8,
165 }

/Users/Dad1525/Desktop/Sessions/2011-07-10-215332/dsdt_vanilla.dsl
132 name (NVUW, zero)
133 OperationRegion (QWVS, SystemMemory, 0xBF79EF98, 0x000A)
134 Field (QWVS, AnyAcc, Lock, Preserve)
135 {
136     OVRS, 8,
137     WFI1, 16,
138     WFI2, 16,
139     WIFS, 8,
140     BEMQ, 8,
141     QCKS, 8,
142     BSBF, 8,
143     MSRC, 8
144 }
145
146 OperationRegion (QWVS, SystemMemory, 0xBF742E18, 0x01AF)
147 Field (QWVS, AnyAcc, Lock, Preserve)
148 {
149     OSYS, 16,
150     SMIF, 8,
151     PRM0, 8,
152     PRM1, 8,
153     SCIF, 8,
154     PRM2, 8,
155     PRM3, 8,
156     LCKF, 8,
157     PRM4, 8,
158     PRM5, 8,
159     P800, 32,
160     LTDS, 8,
161     PWRS, 8,
162     DBG5, 8,
163     THOF, 8,
164     ACT1, 8,
165 }

/Users/Dad1525/Desktop/dsdt_vanilla_A12.dsl
name (NVUW, zero)
OperationRegion (QWVS, SystemMemory, 0xBF79EF98, 0x000A)
Field (QWVS, AnyAcc, Lock, Preserve)
{
    OVRS, 8,
    WFI1, 16,
    WFI2, 16,
    WIFS, 8,
    BEMQ, 8,
    QCKS, 8,
    BSBF, 8,
    MSRC, 8
}
OperationRegion (QWVS, SystemMemory, 0xBF741E18, 0x01AF)
Field (QWVS, AnyAcc, Lock, Preserve)
{
    OSYS, 16,
    SMIF, 8,
    PRM0, 8,
    PRM1, 8,
    SCIF, 8,
    PRM2, 8,
    PRM3, 8,
    LCKF, 8,
    PRM4, 8,
    PRM5, 8,
    P800, 32,
    LTDS, 8,
    PWRS, 8,
    DBG5, 8,
    THOF, 8,
    ACT1, 8,
}
```

Well that is all for now. I hope you learned a little and make speedy progress toward that perfect Hackint0sh desktop or laptop.

Good luck.

Steve

The End