Guide to patch AppleHDA for your codec

Started by Mirone, Jan 02 2014 02:00 AM

Mirone

Posted 02 January 2014 - 02:00 AM

. CHAPTER 1: GETTING CODEC INFORMATION.

Intro: This guide is for those who want to use AppleHDA and / or trying to learn how it works. It is divided in to several steps and we will address each of them throughout the guide.

. STEP ONE: Getting Information from the codec. There are other ways but in this guide we will explain how to get to Linux. Using Ubuntu or another distro. (Note: Use the latest version of Linux and Alsa drivers)

in terminal type:

```
cat /proc/asound/card0/codec#1 > ~/Desktop/codec_dump.txt
```

or

```
cat /proc/asound/card0/codec#0 > ~/Desktop/codec_dump.txt
```

or if you prefer use zhell script:

```
cd ~/Desktop && mkdir CodecDump && for c in /proc/asound/card*/codec#*; do f="${c/\/*card/card}"; cat "$c" > CodecDump/${f//\//-}.txt; done && zip
```

Save it in a safe place.

In OS X, we use the codec_dump for a graphical interface, convert it to decimal (I'll explain why later), get him codec verbs and correct them for OS X.

Download and install:Graphviz (http://www.graphviz.org/pub/graphviz/stable/macos/mountainlion/graphviz-2.34.0.pkg)
Download CodecGraph:codecgraph (http://helllabs.org/codecgraph/codecgraph-20120114.tar.gz) , unzip it to desktop, copy the generated codec_dump.txt linux folder to CodeGraph then type in termainal:

```
cd ~/Desktop/codecgraph
```

and then:

```
./codecgraph codec_dump.txt
```

. STEP TWO: Converting CodecGraph CodecDump and hexadecimal to decimal:
Download: convert_hex_to_dec (http://www.mediafire.com/?1ytbnh0zvfl) Credits Munky . Unzip it in the same folder used previously (codecgraph) - In the terminal:

```
cd ~/Desktop/codecgraph

chmod +x ./convert_hex_to_dec.rb

./convert_hex_to_dec.rb codec_dump.txt.svg > ~/Desktop/codecgraph/codec_dump_dec.txt.svg
```

* Replace "codec_dump.txt" with your user name and codec file (if you gave a specific name)

```
./convert_hex_to_dec.rb codec_dump.txt > ~/Desktop/codecgraph/codec_dump_dec.txt
```

* Replace "codec_dump.txt" with your user name and codec file (if you gave a specific name)

. STEP THREE: For our standard onboard sound card be recognized by OS X must add the device to our HDEF DSDT.aml. For this we use the following patches: HDEF or AZAL to HDEF in both DTGP method has to be present in our DSDT.aml. We use the DSDT Editor to apply the patches correctly http://olarila.com/f...c.php?f=7&t=646 (http://olarila.com/forum/viewtopic.php?f=7&t=646) The Patchs you find here http://olarila.com/forum/patches.php (http://olarila.com/forum/patches.php) This is a Audio Injection standard and should work for all codecs:

```
Device (HDEF)
        {
            Name (_ADR, 0x001B0000)
            Method (_DSM, 4, NotSerialized)
            {
                Store (Package (0x04)
                    {
                        "layout-id",
                        Buffer (0x04)
                        {
                            0x0C, 0x00, 0x00, 0x00
```

```
Method (DTGP, 5, NotSerialized)
    {
        If (LEqual (Arg0, Buffer (0x10)
                {
                    /* 0000 */    0xC6, 0xB7, 0xB5, 0xA0, 0x18, 0x13, 0x1C, 0x44,
                    /* 0008 */    0xB0, 0xC9, 0xFE, 0x69, 0x5E, 0xAF, 0x94, 0x9B
                }))
        {
            If (LEqual (Arg1, One))
            {
                If (LEqual (Arg2, Zero))
```

. CHAPTER 2: CORRECTION APPLEHDA.KEXT

For the AppleHDA Work correctly for our codec we make some modifications in the same plists.
So what are these changes?

If you look at it (click the right mouse button / Show Package Contents), you will find in the Content folder of the main Info.plist (not touching it) and a folder plugins. There are plugins that we have to fix, but not all just Info.plist from this one:

1-AppleHDAHardwareConfigDriver.kext
2-platform.xml and layout12.xml or Layout1.xml (in Folder Resources)
3-BinPatch

1-AppleHDAHardwareConfigDriver.kext

To edit this kext need to get configdata / CodecVerbs.
You can fix them in two ways, either manually or automatically using the script called Signal64 Verbit that performs the entire process automatically
requiring a few adjustments you can download it here: (http://www.insanelymac.com/forum/index.php?app=core&module=attach&section=attach&attach_id=137502) **verbit.zip** (http://www.insanelymac.com/forum/index.php?app=core&module=attach&section=attach&attach_id=137502) **2.65KB** 448 downloads
Unzip it in the same folder codecgraph used previously. then Run this command in Terminal:

```
cd /Users/your_user_name/Desktop/codecgraph
```

and then:

```
./Verbit codec_dump.txt> verbs.txt
```

. NOTE: You must delete the line codec_dump.txt: AFG Function Id: 0x1 (unsol 0)
Having done this codecgraph open the folder and you will see two files and another one called verbs.txt verbitdebug.txt.
verbit.texto is in our interest. For in him we find our configdata / CodecVerbs original and corrected tops underneath.
verbs.txt:

```
Verbs from Linux Codec Dump File: codec_dump.txt


Codec: Realtek ALC887-VD   Address: 0   DevID: 283904135 (0x10ec0887)

  Jack   Color  Description               Node     PinDefault          Original Verbs
-------------------------------------------------------------------------------------------------------
 ATAPI Unknown  SPDIF Out at Int ATAPI     17 0x11   0x99430140   01171c40 01171d01 01171e43 01171f99
   1/8   Black  Speaker at Ext Rear        18 0x12   0x411111f0   01271cf0 01271d11 01271e11 01271f41
   1/8   Green  Line Out at Ext Rear       20 0x14   0x01014010   01471c10 01471d40 01471e01 01471f01
   1/8   Black  Line Out at Ext Rear       21 0x15   0x01011012   01571c12 01571d10 01571e01 01571f01
```

verbitdebug.txt:

```
Blacklist:
411111f0 400000f0 CD at Int ATAPI
Removed Nodes: 0x12 0x1c 0x1f
Checking 71c Associations

Current Associations
 01171c40 = 4
 01471c10 = 1
 01571c12 = 1
 01671c11 = 1
 01771c14 = 1
 01871c50 = 5
 01971c60 = 6
 01a71c5f = 5
 01b71c20 = 2
 01d71c01 = 1  note: Changed 0 to 1
 01e71c30 = 3

  Used associations = 4 1 1 1 1 5 6 5 2 1 3
Unused associations = 7 8 9 a b c d e

Correcting duplicate associations

Checking if 4 already exists in: 1 1 1 1 5 6 5 2 1 3
  no duplicate found
Checking if 1 already exists in: 4 1 1 1 5 6 5 2 1 3
  duplicate found - Is this the first time we've seen this association?
  yes - ignoring
Checking if 1 already exists in: 4 1 1 1 5 6 5 2 1 3
  duplicate found - Is this the first time we've seen this association?
  no - replacing association with: 7
Checking if 1 already exists in: 4 1 7 1 5 6 5 2 1 3
  duplicate found - Is this the first time we've seen this association?
  no - replacing association with: 8
Checking if 1 already exists in: 4 1 7 8 5 6 5 2 1 3
  duplicate found - Is this the first time we've seen this association?
  no - replacing association with: 9
Checking if 5 already exists in: 4 1 7 8 9 6 5 2 1 3
  duplicate found - Is this the first time we've seen this association?
  yes - ignoring
Checking if 6 already exists in: 4 1 7 8 9 5 5 2 1 3
  no duplicate found
Checking if 5 already exists in: 4 1 7 8 9 5 6 2 1 3
  duplicate found - Is this the first time we've seen this association?
  no - replacing association with: a
Checking if 2 already exists in: 4 1 7 8 9 5 6 a 1 3
  no duplicate found
Checking if 1 already exists in: 4 1 7 8 9 5 6 a 2 3
  duplicate found - Is this the first time we've seen this association?
  no - replacing association with: b
Checking if 3 already exists in: 4 1 7 8 9 5 6 a 2 b
  no duplicate found
```

```
New 71c Associations
 Before     After
--------------------------------------------------
01171c40  01171c40  01171d01  01171e43  01171f99
01471c10  01471c10  01471d40  01471e01  01471f01
01571c12  01571c70  01571d10  01571e01  01571f01
01671c11  01671c80  01671d60  01671e01  01671f01
01771c14  01771c90  01771d20  01771e01  01771f01
01871c50  01871c50  01871d98  01871ea1  01871f01
01971c60  01971c60  01971d9c  01971ea1  01971f02
01a71c5f  01a71ca0  01a71d30  01a71e81  01a71f01
```

The codec used in the above example as you can see is a ALC887-VD was necessary to change only one of the external microphones.
To understand how these changes are made we need to go deeper.
So what is a codec verb?
For each input/output there is a sequence of 4 verbs.
Lets take an example from codec used before:

ts about NodeID(NID) 0x14 in hex, 20 in decimal:
The default verbs for it: 21471c10 21471d44 21471e01 21471f01
01471c10:
01471c10 = CAd (Codec Adress)
01471c10 = NID (NodeID)
01471c10 = Verb Commands like 71"c" then "d", "e" ,"f"
01471c10 = Verb Data
And the same for the rest.

I think Cad, NID and Verb Commands are pretty simple to figure out: first 2 you get them from linux dump, and commands are standard 71c, 71d, 71e, 71f.
Lets see about verb data:

- 71cXY X=Default Association Y=Sequence

. DEFAULT ASSOCIATION AND SEQUENCE: are used together by software to group Pin Complexes (and therefore jacks) together into functional blocks to support multichannel operation. Software may assume that all jacks with the same association number are intended to be grouped together, for instance to provide six channel analog output. The Default Association can also be used by software to prioritize resource allocation in constrained situations. Lower Default Association values would be higher in priority for resources such as processing nodes or Input and Output Converters. Note that this is the default association only, and software can override this value if required, in particular if the user provides additional information about the particular system configuration. A value of 0000b is reserved and should not be used. Software may interpret this value to indicate that the Pin Configuration data has not been properly initialized. A value of 1111b is a special value indicating that the Association has the lowest priority. Multiple different Pin Complexes may share this value, and each is intended to be exposed as independent devices.

. SEQUENCE: indicates the order of the jacks in the association group. The lowest numbered jack in the association group should be assigned the lowest numbered channels in the stream, etc. The numbers need not be sequential within the group, only the order matters. Sequence numbers within a set of Default Associations must be unique.

*My notes for this, if you look on the vanilla codec verb(no matter what codec) you will see that Sequence is always 0.
Why that, and why for us on our codec verbs is different? Simple b/c Apple dosen't have multichannel analog out! so you must change it to 0.

*About Default Association, Keep the default one, in order, and on those that are the same replace them with new one that dosen't exist already in the dump... the values you can use are: 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d and f.(see what does verbit if you still don't understand)

- 71dXY X=Color Y=Misc

. COLOR: indicates the color of the physical jack for use by software.

```
Unknown 0
Black 1
Grey 2
Blue 3
Green 4
Red 5
Orange 6
Yellow 7
Purple 8
Pink 9
Reserved A-D
```

. MISC: is a bit field used to indicate other information about the jack. Currently, only bit 0 is defined. If bit 0 is set, it indicates that the jack has no presence detect capability, so even if a Pin Complex indicates that the codec hardware supports the presence detect functionality on the jack, the external circuitry is not capable of supporting the functionality.
Reserved 3
Reserved 2
Reserved 1
Jack Detect Override 0

*Use 0 for Jack Detect - External Device(e.g. Headphones Mic etc)
Use 1 for Jack Detect Disabled - Internal Device(e.g. Internal speakers, Internal mic)

71eXY X=Default Device Y=Connection Type

. DEFAULT DEVICE: INDICATES: the intended use of the jack or device. This can indicate either the label on the jack or the device that is hardwired to the port, as with integrated speakers and the like.

Line Out 0
Speaker 1
HP Out 2
CD 3
SPDIF Out 4
Digital Other Out 5
Modem Line Side 6
Modem Handset Side 7
Line In 8
AUX 9
Mic In A

. CONNECTION TYPE: indicates the type of physical connection, such as a 1/8-inch stereo jack or an optical digital connector, etc. Software can use this information to provide helpful user interface descriptions to the user or to modify reported codec capabilities based on the capabilities of the physical transport external to the codec.

Unknown 0
1/8" stereo/mono 1
1/4" stereo/mono 2
ATAPI internal 3
RCA 4
Optical 5
Other Digital 6
Other Analog 7
Multichannel Analog (DIN) 8
XLR/Professional 9
RJ-11 (Modem) A

71fXY X=Port Connectivity Y=Location

. PORT CONNECTIVITY: indicates the external connectivity of the Pin Complex. Software can use this value to know what Pin Complexes are connected to jacks, internal devices, or not connected at all.
00b - The Port Complex is connected to a jack (1/8", ATAPI, etc.).
01b - No physical connection for Port.
10b - A fixed function device (integrated speaker, integrated mic, etc.) is attached.
11b - Both a jack and an internal device are attached. The Information provided in all other fields refers to the integrated device. The PD pin will reflect the status of the jack; the user will need to be queried to figure out what it is.

. LOCATION: INDICATES the physical location of the jack or device to which the pin complex is connected. This allows software to indicate, for instance, that the device is the “Front Panel Headphone Jack” as opposed to rear panel connections.

*Details:
Convert the 2 digit hex number to binary.
Pad the front with zero's to make it 8 dgits.

EXAMPLE:

0x02 = binary 10 = 00000010 8 digit binary

Reading the bits from left to right:

Port Connectivity bits 7:6
---
00 - Port is connected to a Jack
01 - No External Port -or- No physical connection for Port
10 - Fixed Function/Built In Device (integrated speaker, mic, etc)
11 - Jack and Internal device are attached

Location Part 1 - bits 5:4
---
00 - External on primary chassis
01 - Internal
10 - Separate chassis
11 - Other

Location Part 2 - bits 3:0

The meaning depends on Location Part 1

00 0000 N/A
00 0001 Rear
00 0010 Front
00 0011 Left
00 0100 Right
00 0101 Top
00 0110 Bottom
00 0111 Special (Rear panel)

A good APP for you to test in practice
is this: PinConfigurator (http://olarila.com/forum/download/file.php?id=3666)
Credits: sax-mmS (http://www.applelife.ru/threads/pin-configurator.18051/)

Thanks again to Signal64 for his VerbCheatSheet (http://rapidshare.de/files/48146006/VerbCheatSheet.rtf.html)
*All info came from High Definition Audio Specification (http://www.intel.com/content/www/us/en/standards/high-definition-audio-specification.html) Read it for more details!

. CONCLUSION:

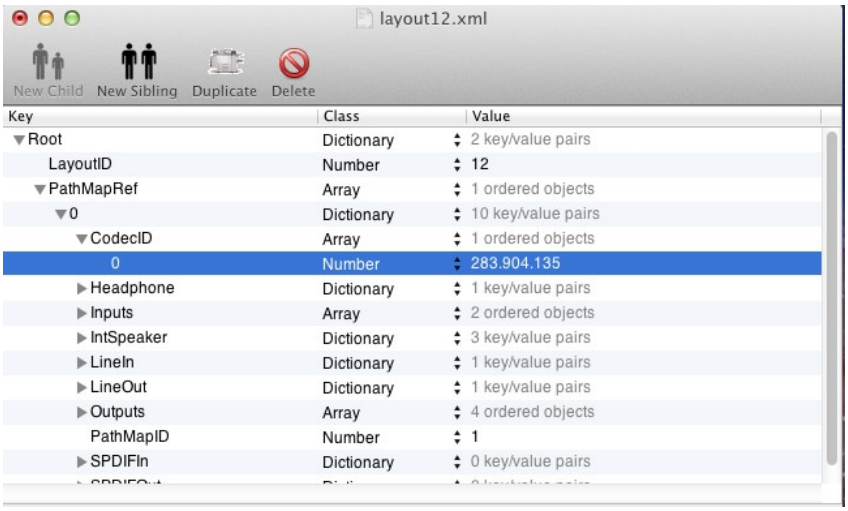
The script does a great job, still not perfect as it does not know what are our options ... perhaps a GUI application may be added in the future.
So that those options may be?
- you may want to setup mic as internal to get ambient noise reduction for it.
- some codecs have only line outs same trick can apply and you can set first output as speaker and get as an option software dsp for it.
- for front panel support you may need to transform back mic in line in.
And so on ...
Now that we got all necessary information about configdata / CodecVerbs we edit the info.plist of AppleHDAHardwareConfigDriver.kext.

Editing AppleHDAHardwareConfigDriver.kext (Info.plist)
(Click with the right mouse button / Show Package Contents) Open Plist with a Plist editor.
Navigate to the key IOKitPersonalities / HDA Hardware Config ResourceHDAConfigDefault.

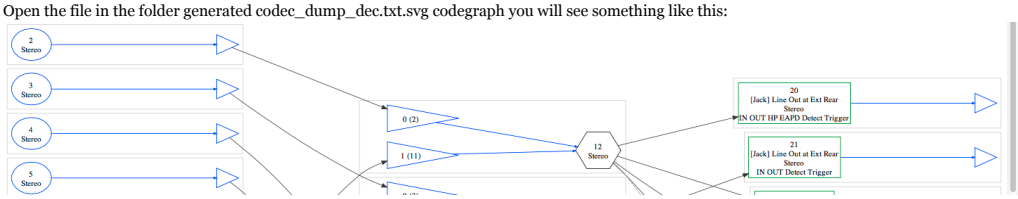
1-CodecID: Put the ID of your codec in Numbers 0x10ec0887 Decimal (Hexadecimal) -> 283 904 135 (Decimal)
2-configdata: Here we put our configdata / Verbs codec that previously corrected.
3-FuncGroup: Must always be 1.
4-LayoutID: By default is 12 but can be any other number provided that you also change the number of layout on your DSDT.aml
* NOTE: In the Mountain/Mavericks Lion kexts so we can use a layout that already has in itself no kext and others like layout887.xml as was used in the Lion kexts.

2- Editing and platform.xml layout12.xml or Layout1.xml (Folder Resources)
This is the trickiest part.
Within the Resources folder you will find several layouts.xml and a Platforms.xml.
You'll edit both according to the previous corrections ie in AppleHDAHardwareConfigDriver.kext use the layout12 so let's make the necessary edits on this.

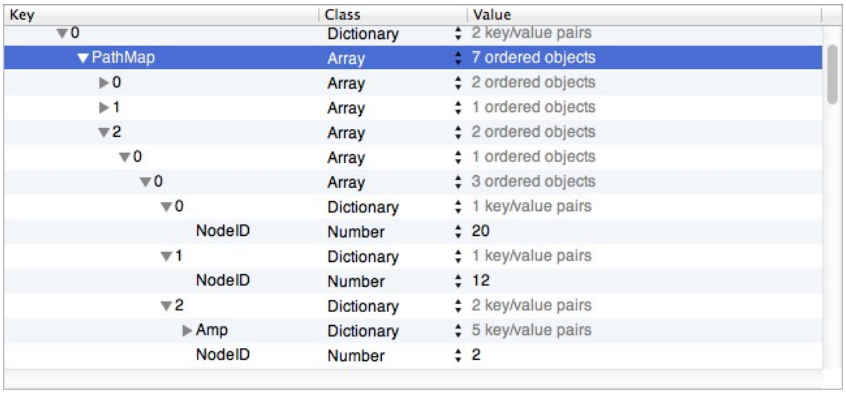
1-CodecID: Put the ID of your codec in Numbers 0x10ec0887 Decimal (Hexadecimal) -> 283 904 135 (Decimal)
2-PathMapID: Change to 1
* Note: Exclude MuteGPIO, SignalProcessing of all keys.



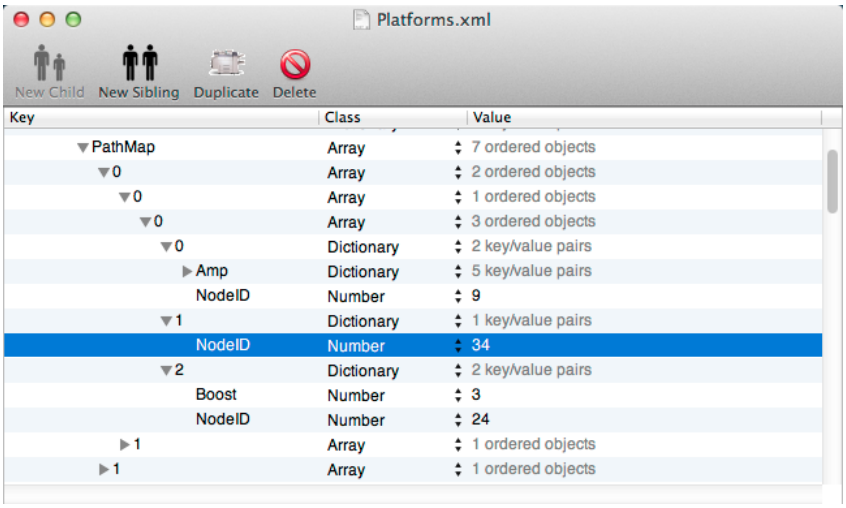
What we had to fix in Layout12.xml is already completed.
Now we edit the Platforms.xml is here that lies the pathmaps. Within Key Pathmaps there are other keys
which are numbered and each corresponds to a specific layout.xml eg use within the same layout12.xml has a key with pathmap Id = 1.
In the example we are using a codec ALC887-VD have 4 outputs and 2 audio inputs. Totaling 6 devices.
For output devices, the PathMap follows this pattern:
Pin Complex (NodeID) -> Audio Mixer-> Audio Output
So we have to find a node Pin Complex (NodeID), a node Audio Mixer and finally an audio output node.
For input devices, the PathMap follow this pattern:
Pin-Complex (NodeID)> Selector/Mixer- Audio Input> Audio
Here, we have to find a node Pin Complex (NodeID), an audio mixer / selector node and finally a node of audio input.
Open the file in the folder generated codec_dump_dec.txt.svg codegraph you will see something like this:



note 20 (Pin Complex) -> 12 (node Audio Mixer) -> 2 (Audio output node)
from there we have already formed our pathmap lineout to (in the case node20 is an internal speaker).
We will do the same for the rest of our pathmaps.
Note: Note that in the screenshot attached above you can see that the lines are continuous connections of nodes.



An example of Pathmaps to Mic (External):
note 24 (Pin Complex) -> 35 (node Audio Mixer) -> 9 (Audio output node)
Note: Usually starts unlike 9 -> 34 -> 24



The pathmaps codec for example were used as well:
9-34-24 = External Mic
9-34-25 = External Mic Front
8-35-26 = Line Out
2-12-20 = Internal Speakers
27-38-37 = Headphone
23-15-5 = Line Out
22-14-4 = Line Out
23-13-3 = Line Out
30-6 = Digital Output

3-BinPatch:
Starting with version 10.6.3 of Snow Leopard was also necessary to correct the binary file of Applehda.kext it is within the MacOs folder that sits inside the Kext to make this correction we will use the script done by bcc9.
patch-hda.zip (http://olarila.com/forum/download/file.php?id=2942)

Copy the folder to your desktop and already with Applehda.kext edits and installed in S / L / E type the following commands in Terminal:

```
cd ~/desktop/patch-hda
./patch-hda.pl 10ec0887
```

. NOTE: Open the folder hda-patch-hda-codecs.pl file inside it is all supported codecs for the script for more information: Script to patch AppleHDA binary for osx10.7/10.8/10.9 (http://www.insanelymac.com/forum/topic/284004-script-to-patch-applehda-binary-for-osx107108109/)

A tip if your codec is not supported by the script you can try to fix manually using a binary editor like HexFiend and replace one of the codecs used, by correction for you.
In the example above was corrected binary for ALC887-VD codec if you are correct for your codec is necessary to replace the 10ec0887 by the id of your codec.
In binary they are written from right to left ie 10ec0887 = 8708ec10.

. IMPORTANT:

The Mountain Lion kext / Mavericks have a new structure and the files are compressed in. Zlib to unpack and edit these files will use zlib.
This application is based on RevoGirl script, it makes the process of files. Zlib from AppleHDA.kext to decompress / compress easier. Just drag the file zlib. The application and it will unzip or drag the xml file. And it will compress to. Zlib.
So if you own an old kext from Lion can compress files using this APP.Download Zlib (http://olarila.com/files/Utils/Zlib.zip) .

Folder with all the necessary tools for editing and some kexts Applehda.kext several original versions of OS X: tools AppleHDA (http://www.sendspace.com/file/kemzwt)

. CREDITS: The King, bcc9, RevoGirl (RIP), EMlyDinEsH, Munky, TimeWalker and others.

pokenguyen

Posted 02 January 2014 - 04:21 AM

Very nice thread! I wonder if there is any difference if we use other layout files? I see most IDT AppleHDA use layout12, while Realtek AppleHDA use layout28.

Mirone

Posted 02 January 2014 - 04:30 AM

> Very nice thread! I wonder if there is any difference if we use other layout files? I see most IDT AppleHDA use layout12, while Realtek AppleHDA use layout28.

I see no difference in the number of Layout.
I think the important thing is the content of LayiutID.
I use LayoutID = 4

---

pokenguyen

Posted 06 January 2014 - 02:58 PM

Hi Mirone, when I edit Platforms.xml, I see some nodes have "Amp" and "Boost". Should I keep them or remove all?

Mirone

Posted 06 January 2014 - 03:29 PM

> Hi Mirone, when I edit Platforms.xml, I see some nodes have "Amp" and "Boost". Should I keep them or remove all?

boost for microphones and
AMP output and inputs for all
has a specific function is
for removing them should know why
to do this.
look here: High Definition Audio Specification (http://www.intel.com/content/www/us/en/standards/high-definition-audio-specification.html)

---

pokenguyen

Posted 06 January 2014 - 04:32 PM

> boost for microphones and
> AMP output and inputs for all
> has a specific function is
> for removing them should know why
> to do this.
> look here: High Definition Audio Specification (http://www.intel.com/content/www/us/en/standards/high-definition-audio-specification.html)

Thank you

---

crump

Posted 10 January 2014 - 02:27 AM

I got stuck reading there...
ts about NodeID(NID) 0x14 in hex, 20 in decimal:
The default verbs for it: 21471c10 21471d44 21471e01 21471f01
21471c10:
2471c10 = CAd (Codec Adress)
21471c10 = NID (NodeID)
21471c10 = Verb Commands like 71"c" then "d", "e" ,"f"
21471c10 = Verb Data
And the same for the rest.

would you mind explaining what is this coming from? Thank you.

Mirone

Posted 10 January 2014 - 11:07 PM

> I got stuck reading there...
> ts about NodeID(NID) 0x14 in hex, 20 in decimal:
> The default verbs for it: 01471c10 01471d44 01471e01 01471f01
> 21471c10:
> o471c10 = CAd (Codec Adress)

> 01471c10 = NID (NodeID)
> 01471c10 = Verb Commands like 71"c" then "d", "e" ,"f"
> 01471c10 = Verb Data
> And the same for the rest.
>
> would you mind explaining what is this coming from? Thank you.

what is your doubt?
since Verbit,
fixes
all verbs automatically for you.
Read the following below the part you
posted and will understand.

---

uzmi Posted 08 March 2014 - 03:16 PM

> **1-AppleHDAHardwareConfigDriver.kext**
>
> To edit this kext need to get configdata / CodecVerbs.
> You can fix them in two ways, either manually or automatically using the script called Signal64 Verbit that performs the entire process automatically
> requiring a few adjustments you can download it here: http://signal64.osx8...udio/verbit.zip (http://signal64.osx86.me/audio/verbit.zip)

Link not works

http://www.insanelym...ile/212-verbit/ (http://www.insanelymac.com/forum/files/file/212-verbit/)

Mirone,
Excelent works
Thanks

Mirone Posted 08 March 2014 - 05:35 PM

@uzmi
Broken link corrected.
Thanks for reporting.

---

realmadrid_cf Posted 10 March 2014 - 04:12 AM

i'm using clover bootloader, how to edit "config.plist"?

Mirone Posted 10 March 2014 - 04:16 AM

so if you go by injecting the audio clover.
using a DSDT has no need to stir in
config.plist.
delete the flag NPCI = 0x2000/3000

Mirone Posted 18 March 2014 - 10:17 AM

> @ **Mirone (http://www.insanelymac.com/forum/user/702532-mirone/)** Can you help me to patch apple Hda for VIA VT2020 for Asus maximus III gene board

OFF TOPIC!
look here: http://www.insanelym...patch-requests/ (http://www.insanelymac.com/forum/topic/293863-applehda-patch-requests/)

Revolutioner

Posted 30 March 2014 - 06:11 AM

mirone getting this only

> cat: /proc/asound/cardo/codec#0: No such file or directory

using linux mint Live USB ...

ємwe

Posted 09 April 2014 - 07:11 AM

can i to apply this method into all kind of audio codec..? like cmedia, ess and etc..?!

Mirone

Posted 09 April 2014 - 10:09 AM

You can try,
some codecs are not supported
Via codecs eg
some of them work and others do not.

---

trik82

Posted 18 April 2014 - 11:35 AM

can't get internal mic working with ALC269 (VC2??)

ConfigData 01471C10 01471D00 01471E17 01471F99 01571C20 01571D10 01571E21 01571F02 01271C30 01271D00 01271EA6 01271F99 01871C40 01871D10 01871EA1 01871F02

Paths
Line-in 8 - 35 - 24
Mic 9 - 34 - 18
Speaker 20 - 13 - 3
HDMI 21 - 12 - 2
Line-out 24 - 12 - 2

someone can help?

Attached Files

(http://www.insanelymac.com/forum/index.php?app=core&module=attach&section=attach&attach_id=139580) **codeco_dec.txt** (http://www.insanelymac.com/forum/index.php?app=core&module=attach&section=attach&attach_id=139580) **9.75KB** 13 downloads

---

Mirone

Posted 18 April 2014 - 02:07 PM

@trik82
Try this:
01471C10 01471D00 01471E13 01471F99 01571C20 01571D10 01571E21 01571F02 01271C30 01271D00 01271EA0 01271F90 01871C40 01871D10 01871E81 01871F02 01470C02.

---

trik82

Posted 18 April 2014 - 02:22 PM

> @trik82
> Try this:
> 01471C10 01471D00 01471E13 01471F99 01571C20 01571D10 01571E21 01571F02 01271C30 01271D00 01271EA0 01271F90 01871C40 01871D10 01871E81 01871F02 01470C02.

nope  i still can see 2 output devices but no inputs

---

Allan

Posted 22 April 2014 - 01:46 PM

> mirone getting this only
> using linux mint Live USB ...

try this:

```
cat /proc/asound/card0/codec#1 > ~/Desktop/codec_dump.txt
```