PulseAudio built in Microphone Guide: The good, the nasty and the ugly

The PulseAudio Manager does not allow lower level configuration of composite devices such as video cameras with audio capture. Thus many audio capture applications do not work correctly when first installed. The fix is to configure either the alsa-base.conf or the sound.conf file both located in the /etc/modprobe.d directory. This guide provides an overview with in depth details on how to make the necessary configurations that will ultimately allow your microphone to work correctly.

First discover what is there and what is installed.

Find sound chip-set:

```
lspci | grep -i aud
00:14.2 Audio device: ATI Technologies Inc SBx00 Azalia (Intel HDA)
01:05.1 Audio device: ATI Technologies Inc RS780 Azalia controller
```

Find sound devices:

```
lsusb
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 046d:09b8 Logitech, Inc.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

In my case the audio is an Intel HDA device and the microphone input comes through the Bus 1 Dev 2 Logitech composite cam/mic bus.

Find installed sound modules:

```
lsmod | grep snd
snd_hda_codec_atihdmi     2727  1
snd_hda_codec_idt      55579  1
snd_hda_intel          24399  2
snd_hda_codec          86743  3 snd_hda_codec_atihdmi,snd_hda_codec_idt,snd_hda_intel
snd_hwdep               6392  1 snd_hda_codec
snd_seq                53791  0
snd_seq_device          6191  1 snd_seq
snd_pcm                80190  2 snd_hda_intel,snd_hda_codec
snd_timer              19892  2 snd_seq,snd_pcm
snd                    63968  12
snd_hda_codec_idt,snd_hda_intel,snd_hda_codec,snd_hwdep,snd_seq,snd_seq_device,snd_pcm,snd_timer
soundcore               6576  1 snd
snd_page_alloc          7559  2 snd_hda_intel,snd_pcm
```

So far so good as the HDA codec and snd modules are installed. Now let's look to see if the audio is alive and well. Running amixer will provide this information

```
amixer
Simple mixer control 'Master',0
  Capabilities: pvolume pswitch pswitch-joined penum
  Playback channels: Front Left - Front Right
```

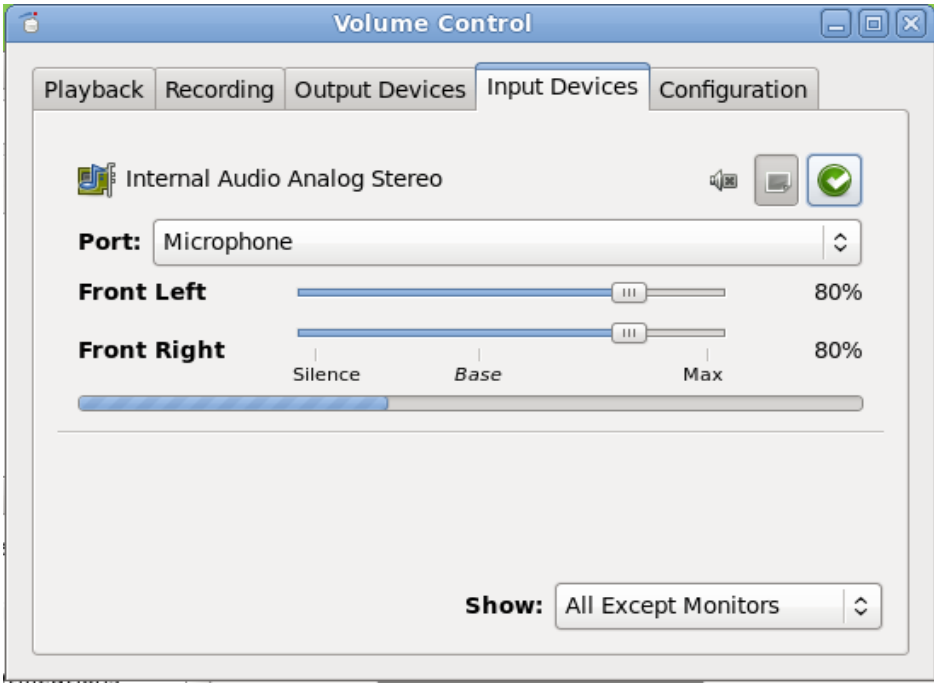
```
  Limits: Playback 0 - 65536
  Mono:
  Front Left: Playback 65536 [100%] [on]
  Front Right: Playback 65536 [100%] [on]
Simple mixer control 'Capture',0
  Capabilities: cvolume cswitch cswitch-joined penum
  Capture channels: Front Left - Front Right
  Limits: Capture 0 - 65536
  Front Left: Capture 56417 [80%] [on]
  Front Right: Capture 56417 [80%] [on]
```

The output of amixer shows that both a “playback” and “capture” devices are configured and operational. But even though all looks well, the built in microphone will not capture data at this point. Note the output volume is set to 100% and the input microphone volume is set to 80%.

Though this exercise is a nice review of how to examine the underlying hardware and software it is not really necessary to determine if the microphone is working. In most cases Linux has drivers for just about every sound device out there. Therefore, the above steps are only necessary after the “easy” way fails. So now let's move to PulseAudio Volume Control to see what the easy way reveals.

If the PulseAudio Volume Control is not installed then install the “pavucontrol” package and then launch it. Click on the Input Devices tab. In this case the microphone is part of the Internal Audio Analog Stereo. My laptop has two microphones on the bottom side of the front top lid. The screen capture below shows no active audio capture activity.

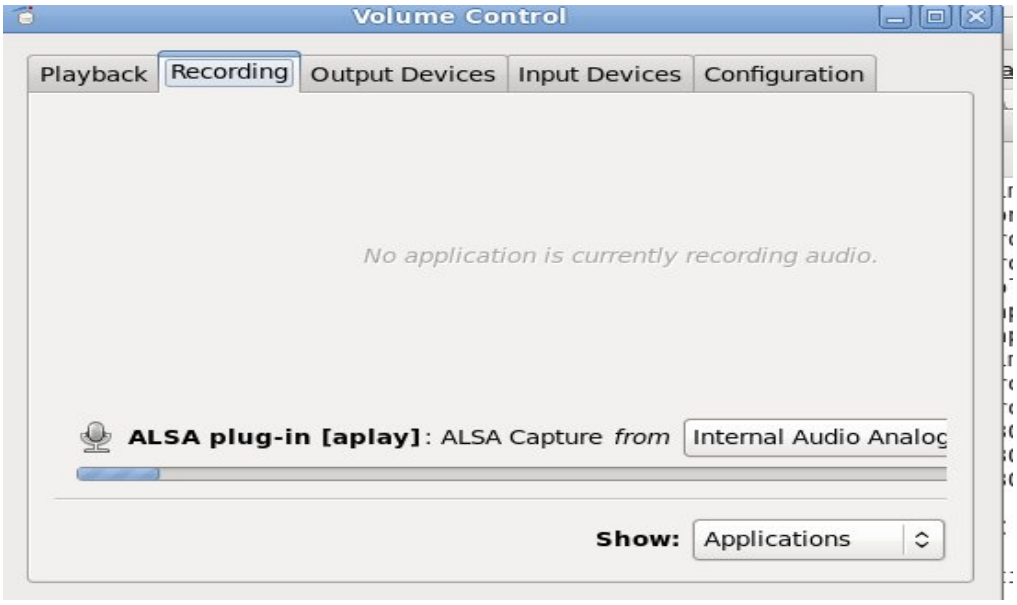
When the audio capture is working correctly the PA Volume Control will show an additional bar that indicates an input signal is present. It looks like the following.



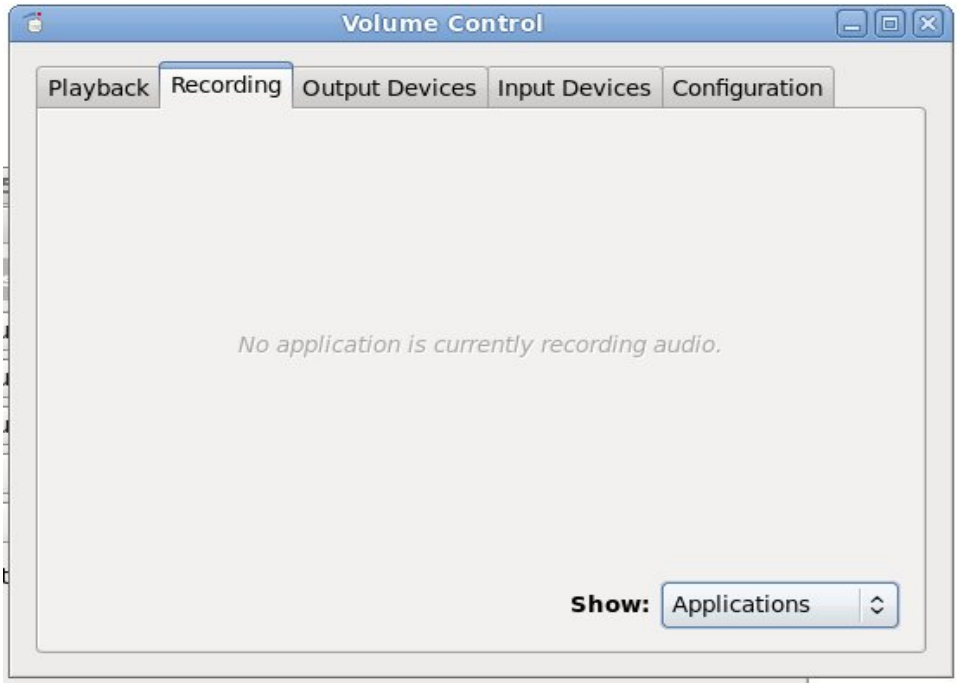
This screen capture shows the microphone audio with the sound level about 1/3 of scale. Note the microphone volume level of 80% is the same as the output of the “amixer” command. If you cannot get to this point don't bother to install any sound applications such as Skype, Audacity or the like. They will not work, at least not yet.

Similarly, when an application is receiving an audio input it will be shown under the Recording tab. The Input Devices tab will indicate that audio is being received by the microphone as the volume bar for the microphone will move with the level of the input sound. This only means the microphone is working correctly. It does not indicate that the application is actually getting the input sound.

When the audio capture software is configured correctly the Recording tab will display the active application as shown below. This indicates the application is actually receiving the audio input stream.



If the application is not receiving the input stream then there will be no application listed under the Recording tab as shown below.



A simple way to test the audio capture configuration is to use the “arecord” and “play” commands.

arecord -d 20 /tmp/test.wav

This command will capture 20 seconds of audio input from the microphone. Execute this command and say some nice things about you mother into the microphone. Also observe what is displayed under the Input Devices and Recording tabs.

Now to play the recorded data use this command.

aplay /tmp/test.wav

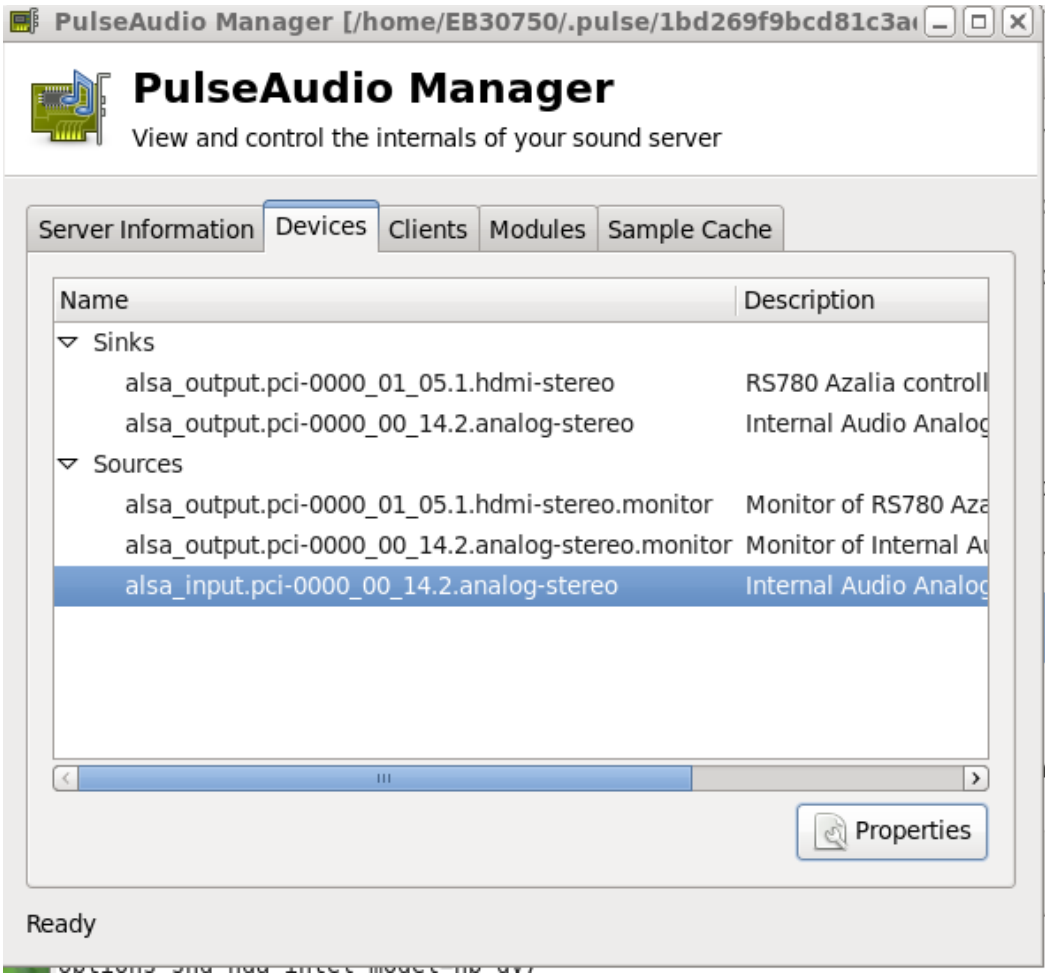
It should also be noted that when an application is “attempting” to capture audio the microphone icon will be displayed in the Gnome task bar as shown in the screen capture below. However, this does not mean that the application actually captured audio, only is attempting to. But if you got this far it means that the sound devices, drivers and modules are all working.

So if you tried this, how did you do? You most likely still struck out, right? Why? Well it gets political and almost nasty. It has to do with the PulseAudio Manager and it's interface to ALSA and the actual sound devices. Many, many Linux users despise PA and prefer to simply revert back to ALSA for sound. There are loads of instructions in forums on the web that have suggested many workarounds. Some of these work but not for the right reasons and they are actually configuring the sound manager redundantly.

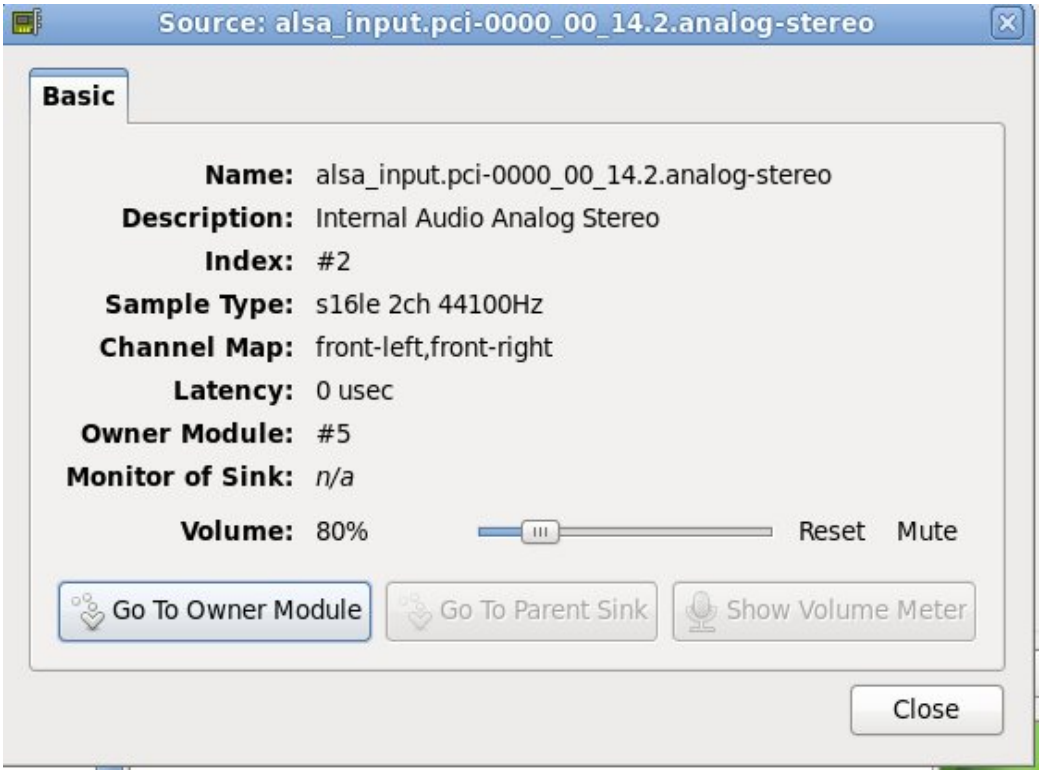
In today's multi-stream audio environment PA is a necessary evil. The sound device only has one input but there are many sound applications attempting to use it. Therefore, it was necessary to develop a “sound server” that converts many streams of audio to one stream that goes to the sound device. Also, many of the sound applications “take over” the sound device when they output directly to ALSA and then it no longer can be shared until the system is rebooted or the sound drivers restarted. Additionally, if a virtual machine is also running and sending its sound output to the host machine it can get really ugly using ALSA directly for audio. PA was created to orchestrate all of the applications' sound needs by providing a high level “service” that receives multiple audio streams and passes them through to the audio device in a consolidated manner. But there is just one little problem. PA does not go far enough to allow device

configuration at the driver level. To demonstrate this it is necessary to use the PulseAudio Manager. The PulseAudio Manager is installed by the ”paman” package.

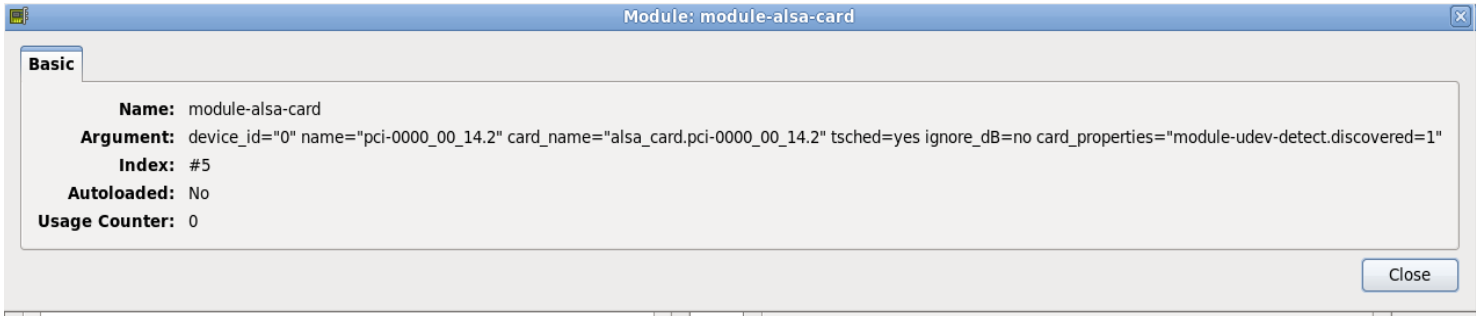
The following screen capture is displayed when the PA Manger is launched and the Devices tab accessed.



The “Internal Audio Analog Stereo” device is selected which represents the “source” of sound which is the built in microphone. By clicking the “Properties” button the following dialog box is displays additional configuration detail. Note the microphone volume level at 80% same a the “amixer” and PA Volume Control applications showed.



By clicking the “Go To Owner Module” button the actual device arguments are shown. Note that the device ID is “0” and the card name is “pci-0000.” But there is nothing in the argument list that ties card 0, slot 0 to the HDA device. This is the problem. The PA Manager lacks the capability (feature) to tie the input of the ALSA sound drivers to the the actual Intel HDA device. This is a PulseAudio bug not a Linux, Skype or any other audio capture application problem. In today's environment of user friendliness there is absolutely no reason to open a terminal and edit an “/etc/xxx.conf” file. But, if you want your built in microphone to work today with PA then you must do so.



There are one of two configurations that can be made that will “link” the sound device to the sound drivers. The first is by configuring (creating) the file /etc/modprobe.d/alsa-basic.conf as follows.

options snd_hda_intel model=hp-dv7
options snd-hda-intel position_fix=1 enable=yes

This file is read by ALSA on startup. It tells the drivers the model of the HDA device. However, this is redundant. As shown by the terminal commands earlier in this article the system already knows that the sound device is an hda_intel device. Creating this file as specified will make my built in microphone work but it is not the optimal solution since this method ties the device to the driver using information the system already knows.

The second configuration that represents the optimal solution is to simply “link” the sound device to the sound driver using an alias. This is accomplished by creating the file /etc/modprobe.d/sound.conf and configuring it as follows:

alias snd-card-0 snd-hda-intel
alias sound-slot-0 snd-hda-intel

(Remember, my sound chip-set is an snd_hda_intel. You will need to replace this with your chip-set reference as reported from terminal output shown earlier in this article.)

The alias lines tell the driver to use the HDA device as card 0, slot 0. This is the “MISSING” link. The PulseAudio Manager lacks the capability to make this link, and therefore, it is currently necessary to “open the hood” and create a “conf” file. The PA Manager should include a drop down menu or at least a line where additional command options can be added to the device. The concept of accessing “properties” implies that lower level configuration options are being accessed. This is totally different from the higher level “application” level configuration data such as volume level, port, etc.

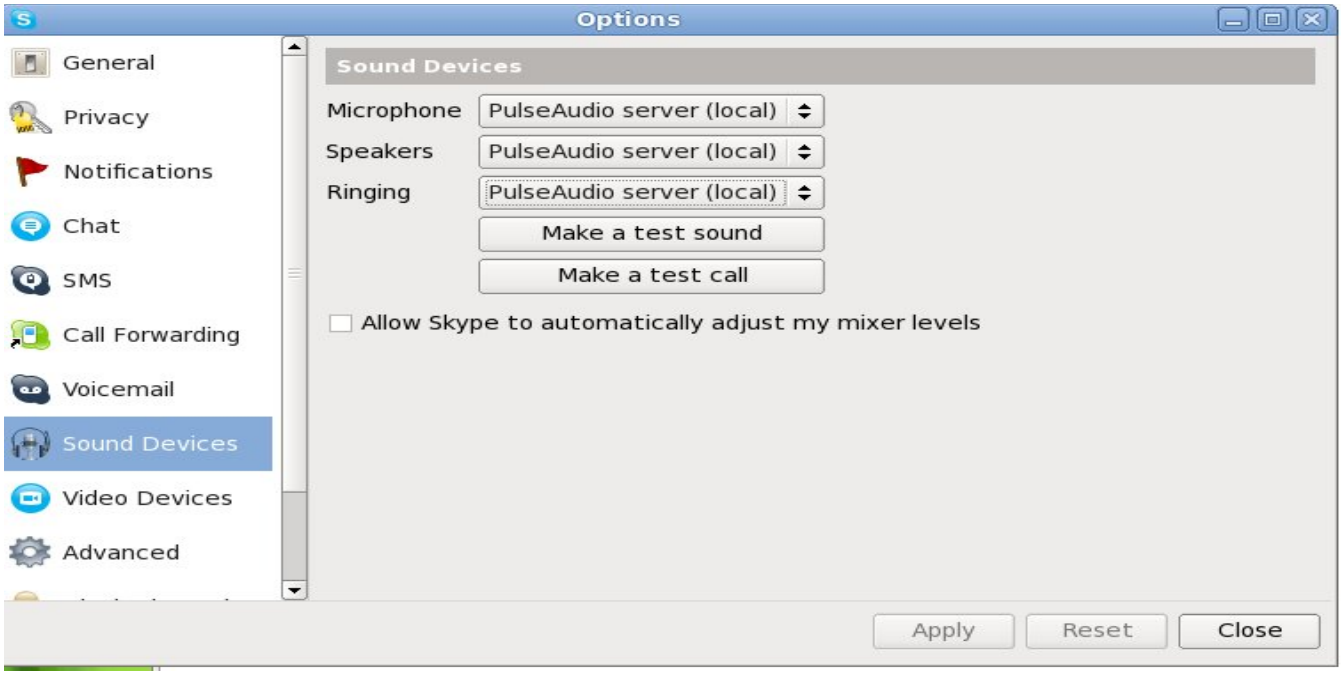
If you want the HDMI output audio to work then also add this line to the sound.conf file. I have not tested this but others have stated it is necessary.

options snd-hda-intel probe_mask=0xffff,0xfff2

And now for the ugly. The fact that the sound device must be configured using PCI bus terminology is quite scary. As driver software moves down to chip microcode the concept of a “bus” is completely lost. Eventually, the ALSA code will be replaced with processor microcode. When this occurs the PA Manager will need to access the chip registers directly where the sound configuration data will ultimately stored. This begins to cross over into the video realm where video is now part of the processor. This is the nasty part. There is no low level direct way to access video or sound via hardware addresses such as Windows and Apple do. This why Linux/Unix has been strapped for years with their ASCII bootup sequence. There can be no graphics until graphic (X) server is up and running and likewise there can be no sound until sound server is up and running. I believe there needs to be some “hefty” upstream rewrites for Linux/Unix to even have a chance to be a player much less remain competitive. I have an iPod that does all of this now. I loaded Skype and the microphone worked first time, every time with no “conf” files needed.

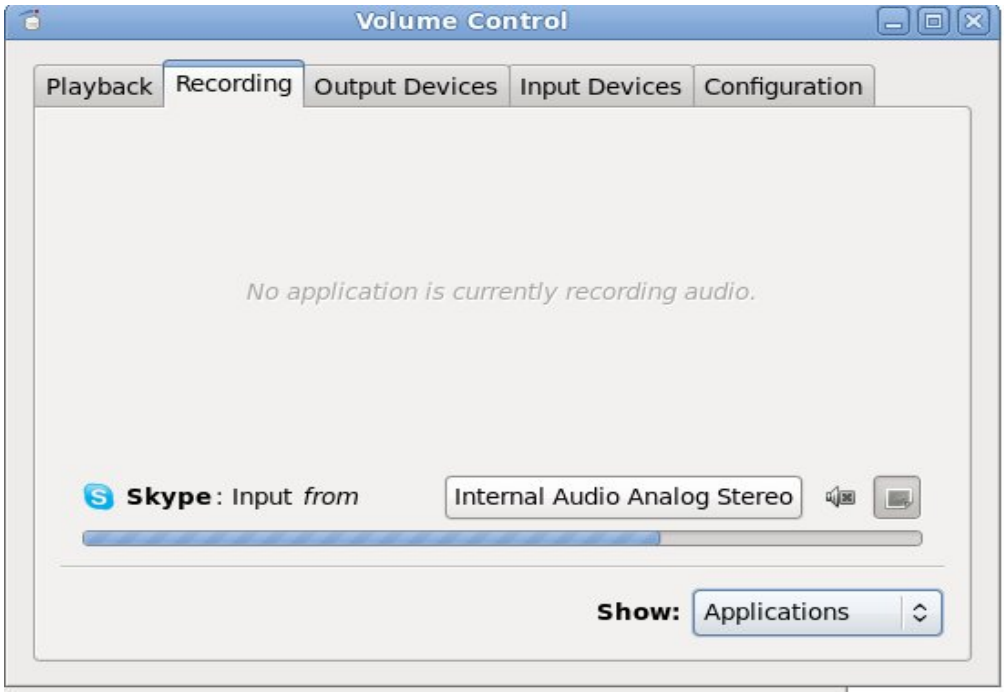
OK, we got passed the good, the nasty and the ugly part so let's continue. Making the assumption that you were able to create the sound.conf file with the proper device links and that you were able to capture and replay some nice words about your mother, let's look at some audio capture applications that use the built in microphone.

The first is Skype. Except for the microphone the 2.1.0.81 beta works out of the box after install. (Why does Fedora claim it is not supported? It's a much more user friendly package than PulseAudio) After adding the sound.conf file it works just fine including the built in microphone. (Please note this is only true for the i386 version. If Skype is installed on x86_64 then several 32 bit libraries must be installed via RPMs). The screen capture below shows the default configuration for Skype with PA installed. Not much you can do since “PulseAudio server (local)” is your only choice. However, I recommend that you uncheck “Allow Skype to automatically adjust my mixer levels.” If you are playing music or there is significant background noise, the microphone will pick this up and Skype will turn down the microphone level to a level where it will not pick up your voice when a phone call comes in.

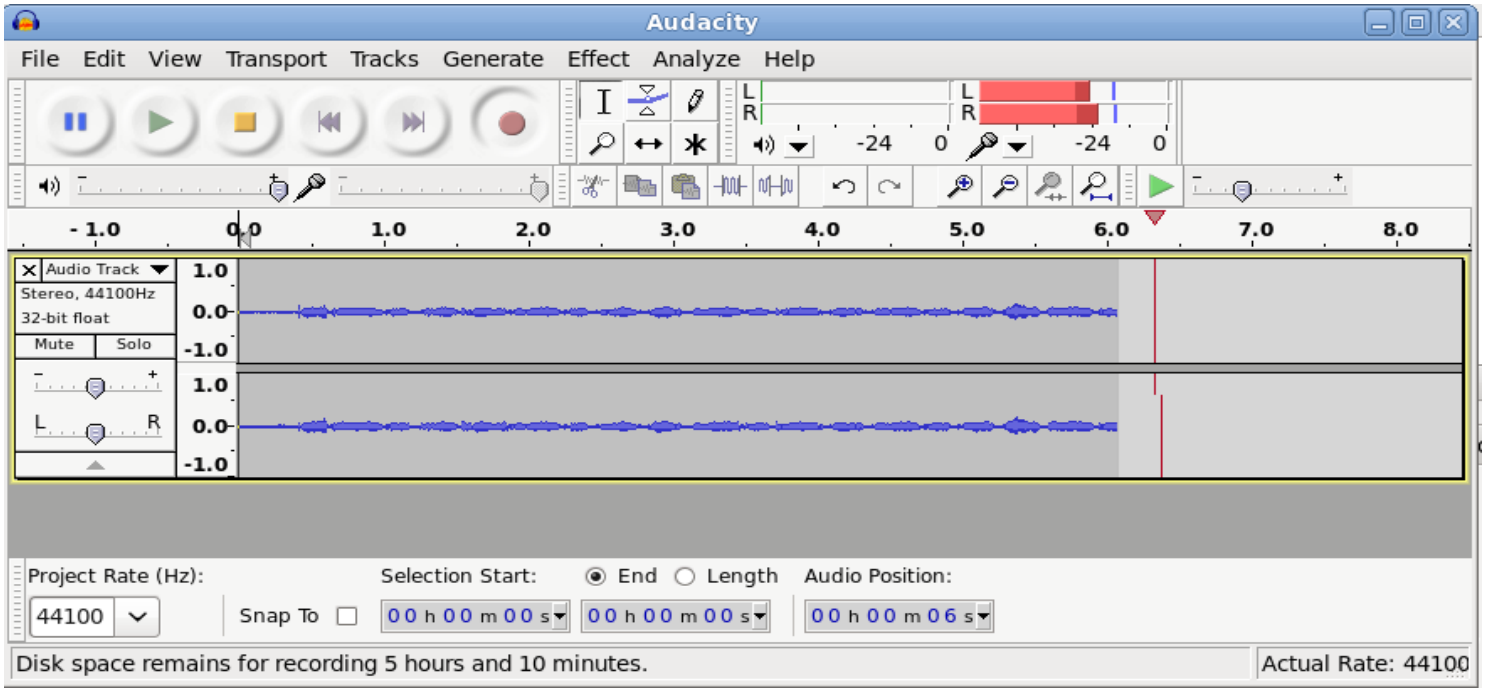


Also, on several forums it was stated that by unlocking the two microphone volume levels the microphone would work. I have tried both unlocking and locking these two volume controls together and it makes no difference. I have mine locked together, otherwise, the sound input might be distorted thus creating an echo affect at the other end.

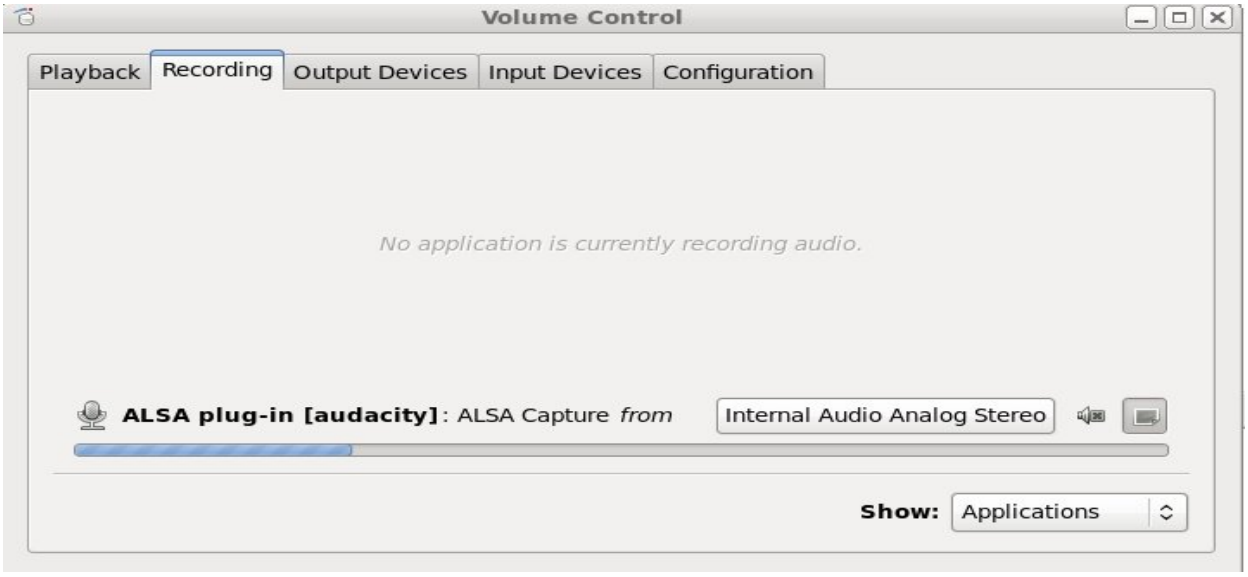
The screen capture below illustrates what the PA Volume Control, Recording tab shows during an active Skype call. The blue bar indicates the volume level that is being received.



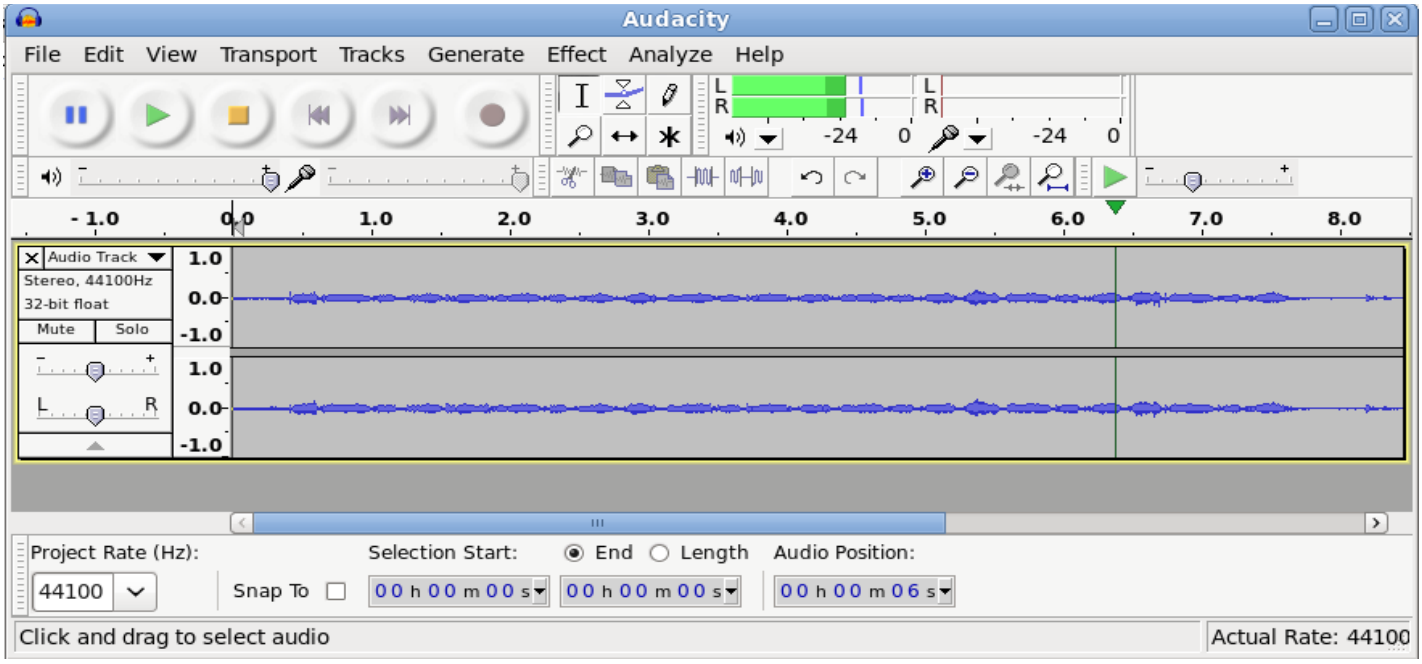
Now let's look at Audacity. The Audacity control panel shows the captured input from the built in microphone indicated by the red bars, one for each channel.



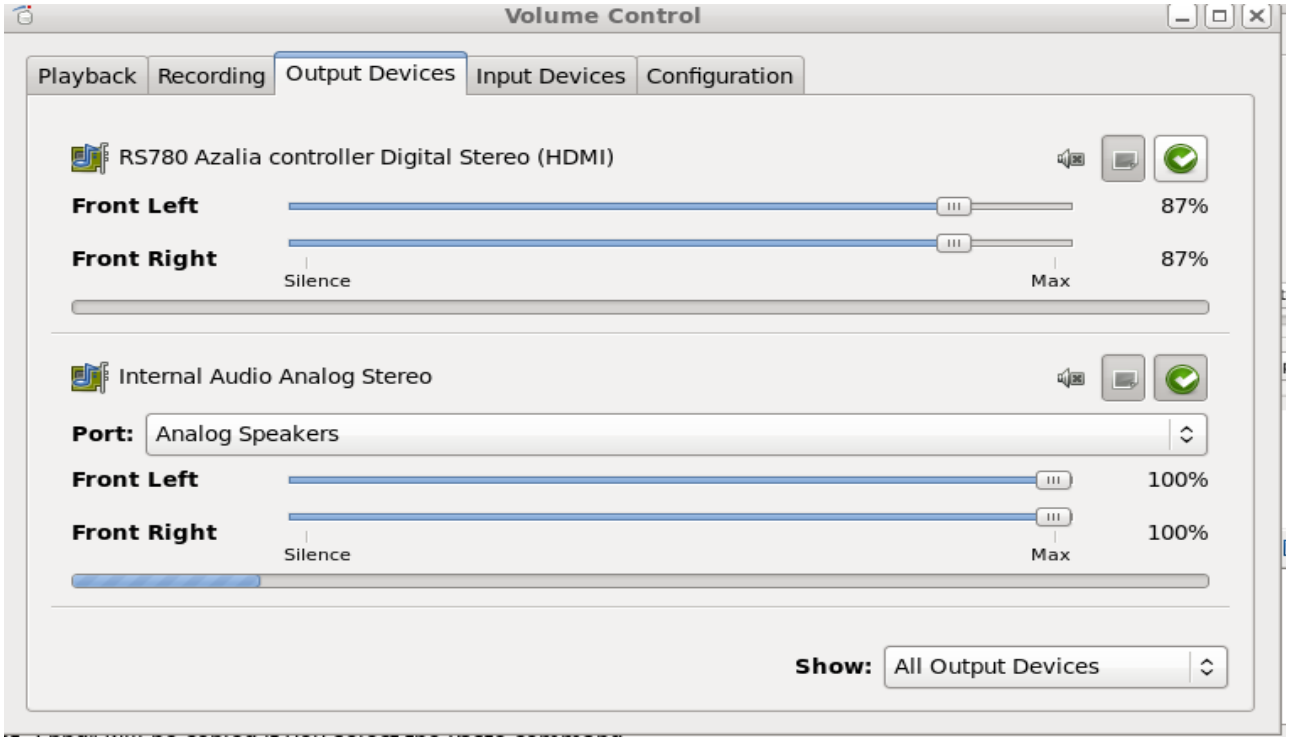
Concurrently, the PA Volume Control Recording tab displays the active audio capture by Audacity as shown in the screen capture below.



Finally, the next two screen captures show the input of the previous audio capture from the built in microphone being played back. First the Audacity control panel indicates playback by the green bars.



Second, the PA Volume Control Output Devices tab shows the output volume level changing during the Audacity playback. The playback device is the same as the microphone, "Internal Audio Analog Stereo", but the "Port" is Analog Speakers whereas for Input Devices the port is "microphone." Note the blue bar indicating sound output volume level.



I hope that this article helps many get their built in mics working and to understand how PulseAudio can be used successfully. The research and testing time behind this article is considerable on my part. This is not to mention the amount of time that others have spent whose research and knowledge I relied on. In the end hundreds, if not thousands of Linux users have spent millions of hours attempting to get their built in microphones working. This should have never been the case. Software with little or no "intelligence" is no longer acceptable. Software of this type is written by lazy programmers who like painters that believe that "cleanup" is not part of their job, just want to move on. For now I am sticking with Linux (Fedora 14 which has shown much improvement in the last release) but if there are anymore lapses like PA my next computer will a Mac. You can help direct this effort by reporting a "urgent" bug against the PA Manager. The upstream developers need to "listen up."

DISCLAIMER: I am not a Linux driver developer but do have experience in writing microprocessor operating systems that incorporate assembler boot and low level IO code. Therefore, I cannot claim that all of the technical terms or inferences relating to PulseAudio, ALSA or devices drivers are correct. However, my claims of user unfriendliness, unnecessary time consuming debugging and the lack of concise and accurate documentation to address built in microphones with Linux have merit.