Subject: Re: [mythtv-users] My experience with MythTV annoyances, 0.20.1 version
From: "Michael T. Dean" <mtdean@thirdcontact.com>
Date: Thu, 21 Jun 2007 12:59:20 -0400
To: Discussion about mythtv <mythtv-users@mythtv.org>

On 05/14/2007 12:47 AM, Yeechang Lee wrote:

> * Old: Analog output to a 2.1 speaker system. New: SP/DIF output to a new 5.1 speaker system and receiver.

...

> * The move to 5.1 digital audio revealed a problem with the audio in about half of the recordings I make from analog (sub-100) cable channels (<URL:http://www.gossamer-threads.com/lists/mythtv/users/241814#241814>). Before Mike Dean jumps in, let me stress again that this has nothing to do with ALSA configuration files (and if he or anyone else can prove me wrong, I'd very much welcome it) or lack thereof.

You have a broken ALSA configuration (among other things) preventing proper playback.

<pet-peeve>How can someone say their configuration is not broken when they are unable to get something to work? In that situation, the only way I can see to know for sure that the configuration is not broken is to try every single possible configuration. Since that's completely impossible, what benefit does saying, "It doesn't work, but I */know/* my configuration is correct," provide? Lack of evidence does not constitute evidence.</pet-peeve>

I'm assuming you're trying to use AC-3 passthrough (and, as you mentioned in the other thread, you have some 32kHz AC-3). I'm also assuming you've properly configured your sound card for IEC958 output (available through ALSA's built-in PCM names iec958 and spdif). I'm further assuming (and this is probably going too far in my assumptions) that your sound card supports a "discrete" IEC958 input (many (most?) don't) */and/* supports 32kHz AC-3 passthrough (and many don't--they'll always slurp in data at 48kHz causing Myth to try to catch up on video playback). If all these assumptions are correct, then everything should just work--if you have a receiver that properly autodetects rate/format and adjusts the IEC958 status appropriately (this one, however, */is/* a pretty good assumption--I don't think consumers would have accepted digital audio if they needed to know how everything works; rather they want it to just work and receivers were designed to handle "broken" audio input so this would be the case).

If these assumptions are */all/* correct, you can set your audio passthrough device to "ALSA:iec958:AES0=0x06,AES1=0x82,AES2=0x00,AES3=0x03" (or "ALSA:spdif:AES0=0x06,AES1=0x82,AES2=0x00,AES3=0x03") and Myth should play a 32kHz AC-3 stream without issue (whether your receiver can is a whole other question, though). Of course, this will completely break playback of 48kHz AC-3 streams (but you could just go into setup and replace the AES3 stanza with "AES3=0x02" for 48kHz or "AES3=0x00" for 44.1kHz). If you try this, I can almost guarantee, though that it won't work. Why, because your sound card probably doesn't support AC-3 at any rate other than 48kHz...

If your sound card does not support a discrete IEC958 input (i.e. the sound processing chip pulls samples from another input--such as one of the PCM inputs--like most (all?) of the nForce/Via/Realtek-based cards) you'll either need a bunch of extra plumbing for something like this to work--if you're lucky--or a new sound card that actually supports 32kHz passthrough (as "passthrough" is a bit of a loose definition when the data is pulled from anything other than a discrete IEC958 input).

I don't know of any specific sound cards that pull data from non-IEC958 inputs but respect the IEC958 rate set on the IEC958 input--all I've ever heard of assume an AC-3 rate that matches the PCM input's rate (and since the same boards that leave off traces to the discrete IEC958 input tend to use fixed-48kHz PCM input with software (driver) rate conversion, they don't support AC-3 at any rate other than 48kHz). The IEC958 rate setting is then just as useless as the sound processing chip's not-connected-to-anything IEC958 input. After all, generally, the manufacturer-supplied drivers (on Windows) will completely hide the disconnected discrete input, so why would the card respect its settings.

But, wait! It works in Windows! Well, the Windows drivers are probably decoding and re-encoding the AC-3 stream on the fly (although some drivers use "hardware" decoders/encoders on the card, if present, but still they're re-encoding)...

As a matter of fact, it's probably Windows and Microsoft and Intel you have to thank for your broken sound card. Even though the IEC958 specification explicitly allows 32kHz, 44.1kHz, and 48kHz rates, the AC'97 specifications required that audio hardware support 96kHz 20-bit stereo and 48kHz 20-bit stereo or 48kHz multichannel recording and playback to be compliant. The successor to AC'97 (Intel High Definition Audio (HD Audio/IHD)), released in 2004, allows 192kHz 32-bit stereo or 96kHz 32-bit 8-channel audio. TTBOMK (and I'll admit my Windows knowledge is minimal), Windows XP didn't provide support for IHD, so until Vista was released vendors had little reason to support it, either, so most sound cards out there today are still AC'97 cards (although some have started to appear with IHD support). But my Creative SoundBlaster Live! (or later) allows me to pass in audio at any rate without using the plug plugin to do rate conversion, so wouldn't that violate AC'97? No. AC'97 simply specifies a means of linking a CODEC chip (has nothing to do with binary formats, like MP3) to a digital controller chip using 5 wires. Vendors can do anything they want before or after those connections, like include a DSP that resamples to 48kHZ (as in the Live!/Audigy/Extigy series) or use software resampling in drivers. The point is that because AC'97 requires a 48kHz digital link, resampling was required for any other rate, and since most vendors were unwilling to add additional hardware, they used driver-based resampling. Since they could do all that without enabling the dedicated IEC958 input, there was little reason to do so: once the architecture was in place for rate conversion of PCM, extending it to decode/re-encode AC-3 was easier (and likely cheaper).

But, wait! It works in MPlayer/xine! Well, MPlayer and xine are "smarter" than users (they've taken the approach that most users don't know--or even care to know--what's happening; they just want to watch their video with sound). If the user specifies passthrough, but the media player is told by the sound card that it requires a 48kHz audio stream, it will decode the AC-3--regardless of your passthrough settings. So, how come I get multi-channel audio? Well, it could be your receiver or it could be your sound card configuration causing you to think you have multi-channel audio when in fact it's plain old stereo (replicated to multiple channels). There are so many possibilities that I can't tell you unless I can see your entire configuration. (And, really, I have my own configuration to worry about.) In other words, if you ever find an app you can use to output a 32kHz AC-3 stream from your sound card such that your receiver shows 32kHz AC-3 on its little LCD, then talk to me and we can get into the details.

So, if your sound card (and/or driver) does not support 32kHz AC-3 passthrough, the */only/* thing MythTV could do to allow AC-3 passthrough to work for a recording with a 32kHz AC-3 stream is to decode it and re-encode it to a 48kHz AC-3 stream (complete with generational loss). Now, this may eventually be possible as a side-effect (or just another part of) #1104, but it's not currently possible.

One of these days, I may write the code to allow MythTV to automatically choose the right IEC958 status bits and send them to the receiver. This would allow users whose sound cards support 32kHz or 44.1kHz AC-3 and a discrete IEC958 input to properly specify the audio type/rate, but it won't do anything for those whose sound cards support only 48kHz AC-3 (for that, you'll have to talk to someone else, like Dr. Mark Spieth of #1104, who might be interested in adding AC-3 re-encoding support). However, I will admit that I have a few higher priority things to do first--such as build and set up my 5.1 (or 7.1) channel speaker system. (In other words, it might take a bit as it's pretty low priority for me--especially now since I don't have a surround speaker system.) However, once I write the code it will do properly what #1105 (http://svn.mythtv.org/trac/ticket/1105) and #1598 (http://svn.mythtv.org/trac/ticket/1598) and #1608 (http://svn.mythtv.org/trac/ticket/1608) tried to do. In the meantime, those with auto-detecting receivers (and extremely nice sound cards) should have it easy (and, like I said, most receivers auto-detect just fine--the patch would just minimize the pops/clicks as the receiver switches modes). Then again, I may find a way to get an HDMI 1.3+ audio connection so I can send 8 channels of 192kHz 24-bit uncompressed PCM audio via a digital link (and avoid the whole AC-3 generational loss). (Note, this probably helps you to infer the kind of timeline I'm talking about for writing this code.)

Of course, if you're one of the lucky ones who has a sound card on which the manufacturer decided to connect the discrete IEC958 input (and that supports 32kHz passthrough), there's no reason any useful program should ever force a user to use a cryptic configuration string like "ALSA:iec958:AES0=0x06,AES1=0x82,AES2=0x00,AES3=0x03". Instead, wouldn't it be nice to be able to put that information somewhere where it could be reused in any application using a nice, simple-to-remember human-readable-and-understandable name? */That/* is the entire reason for an ALSA configuration file. Therefore, it makes no sense to say, "You don't need an ALSA configuration file." Really, I don't need Myth, either. I could just type a bunch of cryptic commands at the console that create cron/at jobs to execute the commands required to set up capture cards and record programs (I'm guessing it would take about 42 lines of Perl code   . Myth--like an ALSA configuration file--is meant to make the user experience a little bit more enjoyable.

So you could, instead, put the above configuration information into an ALSA configuration file. Something like the bit at the bottom of this post would allow you to specify "ALSA:passthru32k" for a 32-kHZ AC-3 passthrough device. Note, however, that I doubt this will "fix" issues with 32kHz passthrough for very many people since most are probably having issues due to lack of support from the card.

Show me a sound configuration problem and I'll show you a sound configuration problem that can be fixed with ALSA configuration files... Unfortunately, people hate what they don't understand, so they love to say, "It's not an ALSA configuration file problem," or "This [ALSA configuration file] is not required on Fedora Core 5 and 6 systems," or "If you are using a version of ALSA newer than 1.0.12 -- <supposedly authoritative source> states that for most uses an .asoundrc is not necessary."

I've given up on trying to fix the completely broken Wiki page on Configuring Digital Sound (http://www.mythtv.org/wiki/index.php/Configuring_Digital_Sound) because people think that just because something worked for them, they can make an authoritative statement about proper configuration. I also wholly agree with Bruce Markey's dislike of people's posting his words to the wiki as the Configuring Digital Sound page was created by copy/pasting my words from one of my posts directed at a specific set of users having a specific issue with 5.1-channel analog audio on the NFORCE2 and explaining why it wasn't working by explaining what they were doing incorrectly with the ALSA configuration files. When the info was copied, the words were attributed to me--which was perfectly reasonable at the time because I actually said them. Then, a lot of people started changing the page--including words attributed to me--without changing the attribution. Since I never really looked at the wiki for the first 6 months, I was "credited" with a */lot/* of inaccurate information until I finally took my name out of the page (and fixed some of the inaccuracies and removed the "how to understand and write an ALSA configuration" part (leaving a link to my post) since no one read it--instead just copying the example and proclaiming it either works or doesn't). But, I haven't really touched the page since then, and doing so isn't worth my time as anything I correct would tend to be uncorrected again--if not where I corrected it, in some other comment somewhere else on the page or on some other page (i.e. someone makes a new page to say what they want because their "opinion" differs from the existing page).

So, basically, I've been ignoring the discussion of sound configuration for quite some time because it's a very complex issue when you realize the vast number of sound cards out there and, although they use a limited number of chips--after all, there aren't that many ALSA drivers--those chips are hooked up in */very/* different ways, requiring very different configurations, especially when you factor in the various permutations of sound card/receiver combos (which can also have a big effect on perceived usability). And, you may have noticed, I can get a bit spun-up over it, too. Unfortunately, it seems most don't care to learn how things work or even how their equipment works (so they can figure out their own proper configuration) but instead want a configuration they can read off a wiki page. And, what's worse, once they figure out something that works for their system and audio files, they spread it around as "universally" authoritative configuration help.

The only reason I responded to this e-mail was your insistence that it's not an ALSA configuration issue and your mentioning me as the guy who always says that sound can be fixed with the ALSA configuration files. I'll admit that you were pretty much right on the second--I always say ALSA configuration issues can be fixed with ALSA configuration files--and I stand by my opinion (and would actually go so far as to say ALSA configuration issues */should/* be fixed with ALSA configuration files). Wow. You're still reading this? It just so happens that most sound issues (and many playback issues) are caused by broken ALSA configurations. Some, however, may be caused by users trying to get their hardware to do things it can't do (like play 32kHz AC-3 audio via passthrough)--or, put another (less accusatory) way, because of broken/incapable sound cards.

The reason I took forever to respond to the e-mail is because, well, it took forever to find enough free time to write it (just like--I'm sure you're thinking--it took forever to read it). And, it was very low priority because I had been assured that your issue had nothing to do with ALSA configuration , and besides, I've been accused of being an ALSA-configuration-file zealot and don't want to feed the rumors.

Mike "extra-wordy ALSA-configuration-file zealot" Dean

Example ALSA conf fragment for a sound card that pulls IEC958 data from a PCM input where the "digital-hw" device is set up to properly output PCM via the digital output.

This fragment relies on the rest of the ALSA configuration file at http://www.mythtv.org/wiki/index.php/Configuring_Digital_Sound#Setting_up_ALSA.27s_.asoundrc.2C_Properly and should be appended to the end of the config file. The only devices that should be used by the user are "passthru32k", "passthru44k", and "passthru48k". Using this fragment, the user does not even have to worry about setting the IEC958 settings on the card. Note that the passthru32k and passthru44k are likely to work the same as passthru48k (i.e. the sound card most likely only allows 48kHz AC-3 passthrough and will ignore the rate specified in the IEC958 status bits, so the user should specify "ALSA:passthru48k" as Myth's "Passthrough output device" just so the name is more "correct" as to the sound card's capabilities). This fragment will */only/* work with sound cards without a discrete IEC958 input that pull IEC958 input from a PCM input (which likely only support 48kHz passthrough)!

```
# Use the ALSA device name "passthru32k" for 32kHz encoded audio (i.e. AC-3,
# DTS, ...) passthrough.
pcm.passthru32k {
  type plug
  slave.pcm "passthru:AES0=0x06,AES1=0x82,AES2=0x00,AES3=0x03"
}

# Control device (mixer, etc.) for the card
ctl.passthru32k {
  type hw
  card 0
}

# Use the ALSA device name "passthru44k" for 44.1kHz encoded audio (i.e.
AC-3,
# DTS, ...) passthrough.
pcm.passthru44k {
  type plug
  slave.pcm "passthru:AES0=0x06,AES1=0x82,AES2=0x00,AES3=0x00"
}

# Control device (mixer, etc.) for the card
ctl.passthru44k {
  type hw
  card 0
}

# Use the ALSA device name "passthru48k" for 48kHz encoded audio (i.e. AC-3,
# DTS, ...) passthrough.
pcm.passthru48k {
  type plug
  slave.pcm "passthru:AES0=0x06,AES1=0x82,AES2=0x00,AES3=0x02"
}

# Control device (mixer, etc.) for the card
ctl.passthru48k {
  type hw
  card 0
}

# Generic passthrough device which accepts IEC958 status bits and sets
IEC958
# controls for proper passthrough on a card without a discrete IEC958 input
# connection that instead pulls non-audio data from PCM
pcm.passthru {
  @args [ AES0 AES1 AES2 AES3 ]
  @args.AES0 {
    type integer
  }
  @args.AES1 {
    type integer
  }
  @args.AES2 {
    type integer
  }
  @args.AES3 {
    type integer
  }
  type hooks
  slave.pcm "digital-hw"
  hooks.0 {
    type ctl_elems
    hook_args [
      {
        name "IEC958 Playback Switch"
        lock true
        preserve true
        value true
      }
      {
        name "IEC958 Playback AC97-SPSA"
        lock true
        preserve true
        value 0
      }
      {
        name "IEC958 Playback Source"
        lock true
        preserve true
        value 0
      }
      {
        name "IEC958 Playback Default"
        lock true
        preserve true
        value [ $AES0 $AES1 $AES2 $AES3 ]
      }
    ]
  }
}

# Control device (mixer, etc.) for the card
ctl.passthru {
  type hw
  card 0
}
```