

## Tvheadend - Feature #4443

### support vaapi for transcoding

2017-06-19 11:53 - C vH

<b>Status:</b>	Fixed	<b>Start date:</b>	2016-05-28
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Törnblom	<b>% Done:</b>	100%
<b>Category:</b>	Transcoding	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	4.4		
<b>Description</b>			
This was similar requested a few times but this is a different approach.			
<u>current situation:</u>			
Tvh supports only nvenc (for Nvidia GPUs) for hw accelerated transcoding, QSV is supported but <u>has no real world usability</u> (you need to patch/recompile your kernel with outdated patches + sign in for SDK ...) and was widely dropped in favor of vaapi that is supported by every Intel and AMD CPUs that are not older then ~6+ years that include an gpu.			
So basically every low end CPU like the popular J1900,N3150,J3455 that are even at commercial NAS systems (Synology...) would be able to handle easily 2-3 Streams at the same time, the latest generation could even transcode H265 in real time without an problem.			
Vaapi is completely integrated at the latest ffmpeg, Linux distributions support it also widely (because you only need the GPU driver) and even Tvh ffmpeg is getting build with --enable-vaapi.			
What is missing:			
Vaapi is used via command line switches, you can't just build ffmpeg with support for it and it works. <a href="#">full howto</a>			
<pre>ffmpeg -hwaccel vaapi -vaapi_device /dev/dri/renderD128 -i "input file" -vf 'format=nv12,hwupload' -map 0:0 -map 0:1 -threads 8 -aspect 16:9 -y -f matroska -acodec copy -b:v 12500k -vcodec 264_vaapi "output file"</pre>			
<u>possible solutions:</u>			
- Tvh offers a way to specify the command line that is used for transcoding (user could tune it their self)			
- Tvh adds a profile for vaapi (this could be problematic due the different systems with maybe different paths to dri render)			
- redo the transcoding profiles or "merge" the codecs branch from here <a href="https://github.com/lekma/tvheadend/tree/codecs">https://github.com/lekma/tvheadend/tree/codecs</a> that is sadly not finished (but works) and was intended to pred to Tvh			
<b>Subtasks:</b>			
Feature # 3831: VAAPI Encoding via FFmpeg.			<b>Rejected</b>

### History

#### #1 - 2017-06-19 14:33 - Jaroslav Kysela

Unfortunately, it seems that lekma has no time to work on the transcoding code and I have no ETA to port his changes to latest tvh. There's no such thing like command-line if you like to use the ffmpeg library directly - you have to build/use all things which the command line ffmpeg tool does.

The problem with the current tvh code is missing: "-vf 'format=nv12,hwupload'" and the /dev/dri device passing.

#### #2 - 2017-06-19 14:43 - C vH

To be honest I have no real knowledge how ffmpeg gets called from Tvh, I thought Tvh calls ffmpeg with some command line - there it would be relatively easy to adjust the used command line to call ffmpeg. No idea if this works this way but at least would be some workaround till a proper solution is at the table.

#### #3 - 2017-06-19 21:46 - saen acro

Then what about [Gstreamer](#)

#### #4 - 2017-07-06 10:36 - Gustavo Sánchez

C vH wrote:

This was similar requested a few times but this is a different approach.

current situation:

Tvh supports only nvenc (for Nvidia GPUs) for hw accelerated transcoding, QSV is supported but has no real world usability (you need to patch/recompile your kernel with outdated patches + sign in for SDK ...) and was widely dropped in favor of vaapi that is supported by every Intel and AMD CPUs that are not older then ~6+ years that include an gpu.

So basically every low end CPU like the popular J1900,N3150,J3455 that are even at commercial NAS systems (Synology...) would be able to handle easily 2-3 Streams at the same time, the latest generation could even transcode H265 in real time without an problem.

Vaapi is completely integrated at the latest ffmpeg, Linux distributions support it also widely (because you only need the GPU driver) and even Tvh ffmpeg is getting build with --enable-vaapi.

What is missing:

Vaapi is used via command line switches, you can't just build ffmpeg with support for it and it works. [full howto](#)

```
ffmpeg -hwaccel vaapi -vaapi_device /dev/dri/renderD128 -i "input file" -vf 'format=nv12,hwupload' -map 0:0 -map 0:1 -threads 8 -aspect 16:9 -y -f matroska -acodec copy -b:v 12500k -vcodec 264_vaapi "output file"
```

possible solutions:

- Tvh offers a way to specify the command line that is used for transcoding (user could tune it their self)
- Tvh adds a profile for vaapi (this could be problematic due the different systems with maybe different paths to dri render)
- redo the transcoding profiles or "merge" the codecs branch from here <https://github.com/lekma/tvheadend/tree/codecs> that is sadly not finished (but works) and was intended to pred to Tvh

Hello, when implementing vaapi support, take in mind that is possible to use vaapi without X11 installed, there is no need to compile with libva\_x11. libva\_drm is usable and you can compile only with this library when using a headless server.

You could compile vaapi when QSV is not present in your system. Current configuration script only active vaapi when you select QSV and always try to compile with libva\_x11

Is easy to change this behavior, changing configure script and Makefile.ffmpeg but when you run tvheadend it fails to open h264\_vaapi codec. FFMPEG and a especial Plex version with HW transcode are running fine on my headless server, so permission and installation is OK.

I am not familiarized with ffmpeg lib use, so I can not help too much. As beta tester but there are many people as me. I look at lekmark implementation but differs so much respect to mainline, so I could not include changes from one version to other.

Lekmark version is working great on my server, but is relative old and sometimes crash. Quality is good and performance and power usage is great !!!! On a Celeron J1900 you could transcode a Full HD to a 360p, 240p, 120p with an extra use of 1w. Linux system + tvheadend streaming = 9w, Linux system + tvheadend vaapi transcode and streaming = 10w !!!! And cpu use is around 12~20%

Hope you develop a great vaapi support for tvheadend

Sorry my bad english.

Regards, Gustavo

#### #5 - 2017-08-14 16:09 - Christian Brauwera

This would be really nice.

I was playing around on a Synology DS916+ with the ffmpeg\_vaapi Version. Its working great with 3 Streams ... Have the ability of setup everything in TVH would be fantastic.

#### #6 - 2017-08-23 11:44 - Jaroslav Kysela

Just a little note: We have a workaround now in latest tvh - the 'MPEG-TS Spawn' profile. You can run an external program which converts MPEG-TS to wanted output (using ffmpeg with vaapi support for example).

#### #7 - 2017-08-23 13:07 - saen acro

- File `sb_sys_analyzer_linux.py` added

Jaroslav Kysela wrote:

Just a little note: We have a workaround now in latest tvh - the 'MPEG-TS Spawn' profile. You can run an external program which converts MPEG-TS to wanted output (using ffmpeg with vaapi support for example).

This commit have a bug and cannot make package

```
CC      libavutil/hwcontext.o
CC      libavutil/hwcontext_cuda.o
CC      libavutil/hwcontext_vaapi.o
libavutil/hwcontext_vaapi.c: In function 'vaapi_frames_get_constraints':
```

```

libavutil/hwcontext_vaapi.c:160:15: error: implicit declaration of function 'vaQuerySurfaceAttributes' [-Werror=implicit-function-declaration]
    vas = vaQuerySurfaceAttributes(hwctx->display, config->config_id,
    ^~~~~~
ccl: some warnings being treated as errors
common.mak:60: recipe for target 'libavutil/hwcontext_vaapi.o' failed
make[4]: *** [libavutil/hwcontext_vaapi.o] Error 1
make[4]: Leaving directory '/opt/tvheadend/build.linux/ffmpeg/ffmpeg-3.3.3'
Makefile.ffmpeg:600: recipe for target '/opt/tvheadend/build.linux/ffmpeg/ffmpeg-3.3.3/.tvh_build' failed
make[3]: *** [/opt/tvheadend/build.linux/ffmpeg/ffmpeg-3.3.3/.tvh_build] Error 2
make[3]: Leaving directory '/opt/tvheadend'
Makefile:780: recipe for target '/opt/tvheadend/build.linux/ffmpeg/build/ffmpeg/lib/libavcodec.a' failed
make[2]: *** [/opt/tvheadend/build.linux/ffmpeg/build/ffmpeg/lib/libavcodec.a] Error 2
make[2]: Leaving directory '/opt/tvheadend'
debian/rules:11: recipe for target 'override_dh_auto_build' failed
make[1]: *** [override_dh_auto_build] Error 2
make[1]: Leaving directory '/opt/tvheadend'
debian/rules:5: recipe for target 'build' failed
make: *** [build] Error 2
dpkg-buildpackage: error: debian/rules build gave error exit status 2

```

to work need to have some checking tools something as intel one, and then to make ffmpeg with corresponding variables.

### #8 - 2017-08-23 15:22 - Jaroslav Kysela

It seems that your libva is really obsolete. The vaQuerySurfaceAttributes() was added in Version 1.2.0 released in 25.Jun.2013. <https://github.com/01org/libva/blob/master/NEWS>. If you don't enable qsv (--enable-qsv) in tvh's configure, this code is not used.

### #9 - 2017-08-23 15:50 - saen acro

- File putty.log added

I just do

```

git clone https://github.com/tvheadend/tvheadend.git
cd tvheadend
./Autobuild.sh

```

ubuntu 17.04 4.10.0-32-generic

Creating configuration files ...

```

WARNING: Option --enable-demuxer=mpeg did not match anything
WARNING: Option --enable-demuxer=mp4 did not match anything
WARNING: Option --enable-muxer=dvd did not match anything

```

after second try

... cut ...

```

In file included from src/main.c:70:0:
src/libav.h:27:34: fatal error: libavfilter/avfilter.h: Няма такъв файл или директория
#include <libavfilter/avfilter.h>
                             ^
compilation terminated.
Makefile:642: recipe for target '/opt/tvheadend/build.linux/src/main.o' failed
make[3]: *** [/opt/tvheadend/build.linux/src/main.o] Error 1
make[3]: Leaving directory '/opt/tvheadend'
Makefile:99: recipe for target 'ffmpeg_all' failed
make[2]: *** [ffmpeg_all] Error 2
make[2]: Leaving directory '/opt/tvheadend'
debian/rules:11: recipe for target 'override_dh_auto_build' failed
make[1]: *** [override_dh_auto_build] Error 2
make[1]: Leaving directory '/opt/tvheadend'
debian/rules:5: recipe for target 'build' failed
make: *** [build] Error 2
dpkg-buildpackage: error: debian/rules build gave error exit status 2

```

### #10 - 2017-08-24 10:45 - C vH

Jaroslav Kysela wrote:

You can run an external program which converts MPEG-TS to wanted output (using ffmpeg with vaapi support for example).

could you show a basic example how to use it ? As you could use transcoding for an FullHD stream even at an RPi2/3 with ffmpeg these days this is pretty interesting smile.png

#### #11 - 2017-08-24 13:36 - Jaroslav Kysela

C vH wrote:

Jaroslav Kysela wrote:

You can run an external program which converts MPEG-TS to wanted output (using ffmpeg with vaapi support for example).

could you show a basic example how to use it ? As you could use transcoding for an FullHD stream even at an RPi2/3 with ffmpeg these days this is pretty interesting smile.png

Just create new streaming profile - select 'MPEG-TS Spawn' type and write the command line to the appropriate field. The spawned task will have MPEG-TS stream on stdin (ffmpeg -i pipe:0 - and tvh expects the output on stdout while is 'pipe:1' for ffmpeg). Note that the output can be anything - just specify the correct mime type for HTTP clients.

#### #12 - 2017-08-25 01:05 - Ricardo Rocha

Since i am not an expert not even close with ffmpeg i will wait that Jaroslav or maybe C vH put a real comand line with ffmpeg example.

Jaroslav i didnt understand "Note that the output can be anything - just specify the correct mime type for HTTP clients." Could you be more specific.

Thanks

#### #13 - 2017-08-25 16:25 - Jaroslav Kysela

There are a lot of examples for the MPEG-TS input: [https://tvheadend.org/projects/tvheadend/wiki/Custom\\_MPEG-TS\\_Input](https://tvheadend.org/projects/tvheadend/wiki/Custom_MPEG-TS_Input) . The only difference is that your should replace INPUTLINK to 'pipe:0' which means stdin (standard input of the unix task) where TVH sends the received stream. The mime should match with the used muxer, so if you keep contents as MPEG-TS, it should be "video/mp2t", for matroska "video/x-matroska" etc. Google will help.

Also, the command line does expect the command, so remove pipe:// from the examples in the provided link.

#### #14 - 2017-08-26 09:09 - Martin Walter

Very, very nice, Jaroslav! This allows HW transcoding on all Synology devices, which has been a frequently requested feature in that community. Is this for TVH 4.4 onwards or are you going to backport it into TVH 4.2.4 as well?

#### #15 - 2017-08-26 09:55 - Christian Eiden

So this means something like this:

```
ffmpeg -hwaccel vaapi -vaapi_device /dev/dri/renderD128 -i pipe:0 -vf 'format=nv12,hwupload' -sn -c:v h264_vaapi -tune zerolatency -f mpegts pipe:1
```

(not working yet as I'm not very firm with ffmpeg)

#### #16 - 2017-08-26 10:32 - Christian Eiden

So this means something like this:

```
ffmpeg -hwaccel vaapi -vaapi_device /dev/dri/renderD128 -i pipe:0 -vf 'format=nv12,hwupload' -sn -c:v h264_vaapi -tune zerolatency -f mpegts pipe:1
```

not working yet as I'm not very firm with ffmpeg ... what am I doing wrong? A CLI call on a recordings is working ...

```
sudo ffmpeg -hwaccel vaapi -vaapi_device /dev/dri/renderD128 -i 'Black Hawk Down.ts' -vf 'format=nv12,hwupload' -sn -c:v h264_vaapi -tune zerolatency -f mpegts 'test.ts'
```

#### #17 - 2017-08-26 11:08 - Jaroslav Kysela

I suggest to do tests with:

```
unset DISPLAY
cat {INPUT_TS_FILE} | {FFMPEG_CMD_LINE_FOR_TVH_SPAWN_PROFILE} > {OUTPUT_TS_FILE}
```

Change {} to your requirement.

#### #18 - 2017-08-26 13:02 - Christian Eiden

Ok, I've been playing with this for a while now and get the following results:

I created a profile "spawn" with the command

```
ffmpeg -hwaccel vaapi -vaapi_device /dev/dri/renderD128 -i pipe:0 -vf 'format=nv12,hwupload,scale_vaapi=w=1920:h=1080' -sn -c:v h264_vaapi -qp 20 -bf 2 -c:a aac -tune zerolatency -f matroska pipe:1
```

Testing this on the CLI works fine with the following command:

```
cat "/media/hdd/tv/Cool Runnings.ts" | DISPLAY="" ffmpeg -hwaccel vaapi -vaapi_device /dev/dri/renderD128 -i pipe:0 -vf 'format=nv12,hwupload,scale_vaapi=w=1920:h=1080' -sn -c:v h264_vaapi -qp 20 -bf 2 -c:a aac -tune zerolatency -f matroska pipe:1 > "/media/hdd/tv/test.mkv"
```

Running it inside TVH gives me the following logs:

```
...
Aug 26 12:58:08 tv tvheadend[740]: spawn: [AVFilterGraph @ 0x5644e8f82d00] No such filter: 'format=nv12,hwupload,scale_vaapi=w=1920:h=1080'
Aug 26 12:58:08 tv tvheadend[740]: spawn: Error reinitializing filters!
...
```

Anything I'm missing?

#### #19 - 2017-08-26 16:43 - Jaroslav Kysela

Remove ' characters. The argument split / escaping is a bit different in tvh and in shell. TVH does not use " or "" for argument contents. It does only space split and \ escaping.

Thinking more about this - you're not first who hits this issue, so tvh should be more friendly for this ;-)

#### #20 - 2017-08-26 16:56 - Jaroslav Kysela

Jaroslav Kysela wrote:

Remove ' characters. The argument split / escaping is a bit different in tvh and in shell. TVH does not use " or "" for argument contents. It does only space split and \ escaping.

Thinking more about this - you're not first who hits this issue, so tvh should be more friendly for this ;-)

Added to v4.3-361-gaa2f92720 .

#### #21 - 2017-08-26 19:51 - Christian Eiden

Works, thanks!

#### #22 - 2017-08-27 17:24 - Chris Peltier

So in v4.3-361-gaa2f92720 " can be used for argument contents? I've been following this thread with interest and I'm a new user of TVH. I've got my ffmpeg vaapi working and tested at the command line and v4.3-361-gaa2f92720 running however htsp clients won't connect to the profile. Is it the case where ffmpeg and htsp are mutually exclusive?

#### #23 - 2017-08-27 18:52 - Jaroslav Kysela

For HTSP protocol only 'htsp' and 'transcode' profiles can be used.

```
4.3-361 arguments: 'this is one argument' or "this is one argument" or this\ is\ one\ argument
before (only):    this\ is\ one\ argument
```

#### #24 - 2017-09-01 19:27 - Jaroslav Kysela

- Target version set to 4.4

Marking as resolved. I did some tests on Intel hardware with v4.3-413-g4598008b9 . There's merged lekma's code with my extensions. The last my extension is that you can select the DRI device for decoding/encoding when you have multiple ones (like me with NVidia Optimus technology - missing fw for nouveau so only Intel DRI was tested).

Please, create a new bug report for a specific issue.

#### #25 - 2017-09-01 19:28 - Jaroslav Kysela

- Status changed from New to Fixed

#### #26 - 2017-09-01 20:18 - C vH

thank you for integration, will publish some test builds (LibreELEC) and try to discover problems !

#### #27 - 2017-09-01 21:01 - Alfred Zastrow

I'm getting a compiling error on debian buster:

```
src/transcoding/codec/codecs/libs/vaapi.c:24:10: fatal error: drm/drm.h file not found #include <drm/drm.h>
```

Which package is missing?

#### #28 - 2017-09-01 21:21 - Mark Clarkstone

Alfred Zastrow wrote:

I'm getting a compiling error on debian buster:

```
src/transcoding/codec/codecs/libs/vaapi.c:24:10: fatal error: drm/drm.h file not found #include <drm/drm.h>
```

Which package is missing?

I have the same issue on Stretch, Jaroslav already knows. He told me to install kernel headers. I've tried that - no luck - so I'm rebuilding my kernel to make sure it's there - will report back.

#### #29 - 2017-09-01 22:19 - saen acro

If Intel 01.org drivers installed there is some bug with search files  
If nVidia not shure.

p.s.

01.org drivers is impossible to be uninstalled if GUI installed. [#4548](#)

#### #30 - 2017-09-01 22:21 - Jaroslav Kysela

I'm on fedora and the drm/drm.h is directly in kernel-headers package, but the same file is copied in libdrm-devel (as libdrm/drm.h). I added a check to configure in v4.3-414-g994de3d9f, so you can give a try.

#### #31 - 2017-09-01 22:35 - Alfred Zastrow

I've changed the #include statement to libdrm/drm.h and it comiles fine. But at the time being I don't get a picture back from a recording:  
journal says:

```
Sep 01 22:28:44 tvh tvheadend7570: subscription: 004D: "DVR: Tatort: Frühstück für immer" subscrib
ing on channel "Das Erste HD", weight: 300, adapter: "Adapter0", network: "Astra_19_2E", mux: "11493
.75H", provider: "ARD", service: "Das Erste HD", profile="testprofile"
Sep 01 22:28:44 tvh tvheadend7570: transcode: 0007: 01:H264: > Using profile h264_vaapi
Sep 01 22:28:44 tvh tvheadend7570: transcode: 0007: 02:MPEG2AUDIO: > Passthrough
Sep 01 22:28:44 tvh tvheadend7570: transcode: 0007: 02:MPEG2AUDIO: > Using profile copy
Sep 01 22:28:44 tvh tvheadend7570: transcode: 0007: 03:MPEG2AUDIO: > Filtered out
Sep 01 22:28:44 tvh tvheadend7570: transcode: 0007: 06:AC3: ==> Filtered out
Sep 01 22:28:45 tvh tvheadend7570: libav: Using deprecated struct vaapi_context in decode.
```

```
Sep 01 22:28:45 tvh tvheadend7570: libav: The filters 'Parsed_format_1' and 'Parsed_yadif_
2' do not have a common format and automatic conversion is disabled.
```

```
Sep 01 22:28:45 tvh tvheadend7570: transcode: 0007: 01:H264: [h264 => h264_vaapi]: filters
: failed to config filter graph
```

#### #32 - 2017-09-02 09:15 - Fer De Montanaro

I have a problem with vaapi and AMD GPU.

With mesa/libav with AMD hardware, is necessary to declare two environment variables:

Extra notes for AMD hardware supporting VCE:

If you have a supported GCN+ AMD GPU running on Linux with the mesa driver stack, you may be able to use the AMD VCE Block via VAAPI with an example such as the one shown below:

```
DRI_PRIME=1 LIBVA_DRIVER_NAME=radeonsi ffmpeg -hwaccel vaapi -vaapi_device /dev/dri/renderD128 -hwaccel_output
_format vaapi \
-framerate 30 -video_size 1920x1200 -f xllgrab -i :0.0 -f pulse -ac 2 -i 1 \
-vf 'format=nv12,hwupload' -threads 8 \
-vcodec h264_vaapi -bf 0 -acodec pcm_s16le output.mkv
```

I declare in /etc/profile the variables, but tvheadend send and error because not found the encoder device, if i declare the variables in the command box in tvheadend other error appear:

```
tvheadend[828]: pass: Unable to start pipe 'DRI_PRIME=1 LIBVA_DRIVER_NAME=radeonsi ffmpeg -hwaccel vaapi -vaapi_device /dev/dri/renderD128 -i pipe:0 -vf 'format=nv12,hwupload' -sn -c:v h264_vaapi -qp 20 -bf 2 -c:a aac -tune zerolatency -f mpegts pipe:1' (wrong executable?)
```

How set the variables to tvheadend find the encoder?

Thanks

**#33 - 2017-09-02 09:18 - Jaroslav Kysela**

@Alfred : Unset the 'deinterlace' flag in the codec settings (the hardware decoder deinterlaces the input anyway and it's settings for the software deinterlacer). I'll look into this later. And, again, report bugs separately.

**#34 - 2017-09-02 09:24 - Jaroslav Kysela**

@Fer De Montanaro: You're probably trying 'MPEG-TS Spawn' profile. You cannot define the environment variables in this way. I may add new field later, but for tests, use a script. Also, you can use the direct vaapi support in the recent tvh code (where you can select the rendering device).

**#35 - 2017-09-02 20:07 - Alfred Zastrow**

Jaroslav Kysela wrote:

@Alfred : Unset the 'deinterlace' flag in the codec settings (the hardware decoder deinterlaces the input anyway and it's settings for the software deinterlacer). I'll look into this later. And, again, report bugs separately.

Thanks, it works now.

**#36 - 2017-09-02 20:51 - Fer De Montanaro**

Thanks for the response.

Where select the rendering device? I not found the option.

Yesterday i build tvh: 4.3-414~g994de3d9f (and yes, all dependencies satisfied, includes vaapi and drm.h) drm.h is not included in kernel headers in Debian, installing libdrm-dev from repo install this.

**#37 - 2017-09-02 21:36 - Jaroslav Kysela**

Fer De Montanaro wrote:

Thanks for the response.

Where select the rendering device? I not found the option.

Create vaapi\_h264 or vaapi\_hevc. The rendering device field is there.

**#38 - 2017-09-03 00:34 - Fer De Montanaro**

I found the setting. But not work, because libav reports:

```
tvheadend[1485]: libav: Failed to initialise VAAPI connection: -1 (unknown libva error).
tvheadend[1485]: vaapi: failed to create a context for device: /dev/dri/renderD128
```

another error:

```
tvheadend[1485]: Invalid frame dimensions 0x0
```

I currently delete all git download, and re-download the last version: 4.3-425~g3cceb22

Not compile:

```
CC      src/libav.o
src/libav.c:248:13: error: 'libav_va_info_callback' defined but not used [-Werror=unused-function]
static void libav_va_info_callback(const char *msg)
      ^~~~~~
src/libav.c:243:13: error: 'libav_va_error_callback' defined but not used [-Werror=unused-function]
static void libav_va_error_callback(const char *msg)
      ^~~~~~
cc1: all warnings being treated as errors
Makefile:684: fallo en las instrucciones para el objetivo '/root/instalables/tvheadend/build.linux/src/libav.o'
```

```
'
make[3]: *** [/root/instalables/tvheadend/build.linux/src/libav.o] Error 1
make[3]: se sale del directorio '/root/instalables/tvheadend'
Makefile:99: fallo en las instrucciones para el objetivo 'ffmpeg_all'
make[2]: *** [ffmpeg_all] Error 2
make[2]: se sale del directorio '/root/instalables/tvheadend'
debian/rules:11: fallo en las instrucciones para el objetivo 'override_dh_auto_build'
make[1]: *** [override_dh_auto_build] Error 2
make[1]: se sale del directorio '/root/instalables/tvheadend'
debian/rules:5: fallo en las instrucciones para el objetivo 'build'
make: *** [build] Error 2
dpkg-buildpackage: fallo: debian/rules build devolvió un estado de salida de error 2
```

### #39 - 2017-09-03 08:18 - saen acro

Fer De Montanaro wrote:

I found the setting. But not work, because libav reports:

[...]

another error:

[...]

I currently delete all git download, and re-download the last version: 4.3-425~g3ccebae22

Not compile:

[...]

Same Error on Ubuntu 16.04 x86 with Atom D410

### #40 - 2017-09-03 10:10 - Jaroslav Kysela

Fixed in v4.3-427-gb580b6b0e .

### #41 - 2017-09-04 06:09 - Christian Eiden

Will there be vaapi enabled builds available in bintray?

### #42 - 2017-09-05 02:20 - Ricardo Rocha

Jaroslav:

first i have been playing a lot with ffmpeg so probably i could have done something wrong...

i tried to use a ffmpeg from source so i can be able to use the vp9\_vaapi with spawn profile...

but just when i was about to start my tests with spawn i am getting this:

```
2017-09-05 01:02:18.414 subscription: 0007: "HTTP" unsubscribing from "RTP 1", hostname="192.168.1.115", client="VLC/2.2.6 LibVLC/2.2.6"
2017-09-05 01:02:18.440 http: 192.168.1.115: using ticket D5166248C31C457C8C0522C9297F1ACF8E931444 for
/stream/channel/ad2c611d844dfd8a310edb9f45960e9c
2017-09-05 01:02:18.445 spawn: Executing "/usr/bin/ffmpeg-10bit"
2017-09-05 01:02:18.448 spawn: ffmpeg version N-86925-g4ff94558f2-static http://johnvansickle.com/ffmpeg/ Copyright (c) 2000-2017 the FFmpeg
developers
2017-09-05 01:02:18.448 spawn: built with gcc 6.4.0 (Debian 6.4.0-2) 20170724
2017-09-05 01:02:18.448 spawn: configuration: --enable-gpl --enable-version3 --enable-static --disable-debug --disable-ffplay --disable-indev=ndio
--disable-outdev=ndio --cc=gcc-6 --enable-fontconfig --enable-frei0r --enable-gnutls --enable-gray --enable-libass --enable-libfreetype
--enable-libfribidi --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrwb --enable-libopenjpeg --enable-libopus
--enable-librtmp --enable-libsoxr --enable-libspeex --enable-libtheora --enable-libvidstab --enable-libvo-amrwbenc --enable-libvorbis --enable-libvpx
--enable-libwebp --enable-libx264 --enable-libx265 --enable-libxvid --enable-libzimg
2017-09-05 01:02:18.448 spawn: libavutil 55. 69.100 / 55. 69.100
2017-09-05 01:02:18.448 spawn: libavcodec 57.102.100 / 57.102.100
2017-09-05 01:02:18.448 spawn: libavformat 57. 76.100 / 57. 76.100
2017-09-05 01:02:18.448 spawn: libavdevice 57. 7.100 / 57. 7.100
2017-09-05 01:02:18.448 spawn: libavfilter 6. 96.100 / 6. 96.100
2017-09-05 01:02:18.448 spawn: libswscale 4. 7.101 / 4. 7.101
2017-09-05 01:02:18.448 spawn: libswresample 2. 8.100 / 2. 8.100
2017-09-05 01:02:18.448 spawn: libpostproc 54. 6.100 / 54. 6.100
2017-09-05 01:02:18.448 spawn: Unrecognized option 'vaapi_device'.
2017-09-05 01:02:18.448 spawn: Error splitting the argument list: Option not found
```

is this version installed with tvheadend? because i saw it's missing vaapi so that option will never work on command line.

is it possible use spawn profile with own compiled ffmpeg and not that one "/usr/bin/ffmpeg-10bit"?

thanks

**#43 - 2017-09-05 08:36 - Jaroslav Kysela**

I don't understand your question. You can run any command to process the stream. It seems that ffmpeg-10bit does not have vaapi support or you passed a wrong argument. Test your command on command-line as first, as noted in comment number 17.

**#44 - 2017-09-05 09:28 - Ricardo Rocha**

I guess i did that but i will confirm.

When tvheadend installs it also install own ffmpeg right? Is that one "/usr/bin/ffmpeg-10bit"?

If i have a custom ffmpeg binary is it possible to make spawn profile use it? I ask this because tvh always try to use that one on /usr/bin/.

Thanks

**#45 - 2017-09-05 10:43 - Jaroslav Kysela**

No, tvh uses own compiled static ffmpeg library, but it does not create any ffmpeg executable. This binary is probably from <https://www.johnvansickle.com/ffmpeg/> as printed to log.

**#46 - 2017-09-05 12:18 - C vH**

ffmpeg from <https://www.johnvansickle.com/ffmpeg> is known to have no vaapi/nvenc capabilities (as they won't work static linked)

**#47 - 2017-10-06 04:02 - Brandon Shermer**

Jaroslav Kysela wrote:

For HTSP protocol only 'htsp' and 'transcode' profiles can be used.

[...]

You said HTSP clients? So I can use an HDHomeRun through the IPTV setup and use a VAAPI transcoding profile for mobile clients correct?

I plan to install the 4.2.3 package for the Synology 916+ NAS. Also, where do I get the proper FFMPEG to install?

**#48 - 2017-10-06 04:11 - Brandon Shermer**

Also noticed that this is implemented in 4.4 versions. Either way would this eventually work for the HDHomeRun using the IPTV transcoding to a mobile phone using the TVHClient?

**#49 - 2017-11-09 23:49 - Remi Os**

**First remove libva1 and i965 packages**

Remove intel-graphics-update-tool

<https://01.org/linuxgraphics/forum/graphics-update-tool-discussions/how-remove-drivers-installed-intel-graphics-update-tool>

a)

```
dpkg-query -f '${binary:Package}\n' -W | \
while read pkg; do apt-cache madison $pkg | head -n 1; done | \
grep -F 01.org | tee updated-packages.txt
```

b) remove the 01.org repository from your configuration:

```
sudo rm -i /etc/apt/sources.list.d/intellinuxgraphics.list
sudo apt-get update
```

remove the lines with i915-4.6.3-4.4.0-dkms and intel-graphics-update-tool from updated-packages.txt

c)

```
apt-get install --reinstall $(while read pkg x; do echo $pkg/xenial; done < updated-packages.txt)
```

Remove libva1 and i965 packages

Check with apt list --installed

```
Apt-get remove libva1 i965-va-driver
```

**Build platform: Ubuntu 16.04LTS.**

Build the dependency chain first.

apt-get install autoconf automake libdrm-dev apt-get m4 libtool libevent perl pkg-config yasm

#### 1. cmrt:

```
git clone https://github.com/01org/cmrt
cd cmrt
./autogen.sh
./configure
time make -j$(nproc) VERBOSE=1
sudo make -j$(nproc) install
sudo ldconfig -vvvv
```

#### 2. libva:

```
git clone https://github.com/01org/libva
cd libva
./autogen.sh
./configure
time make -j$(nproc) VERBOSE=1
sudo make -j$(nproc) install
sudo ldconfig -vvvv
```

#### 3. intel-hybrid-driver:

```
git clone https://github.com/01org/intel-hybrid-driver
cd intel-hybrid-driver
./autogen.sh
./configure
time make -j$(nproc) VERBOSE=1
sudo make -j$(nproc) install
sudo ldconfig -vvv
```

#### 3. intel-vaapi-driver:

```
git clone https://github.com/01org/intel-vaapi-driver
cd intel-vaapi-driver
./autogen.sh
./configure --enable-hybrid-codec
time make -j$(nproc) VERBOSE=1
sudo make -j$(nproc) install
sudo ldconfig -vvvv
```

#### 4. Libva-utils (vainfo):

```
git clone https://github.com/01org/libva-utils
cd libva-utils/
./autogen.sh
./configure
make -j$(nproc) VERBOSE=1
make -j$(nproc) install
ldconfig -vvvv
```

When done, test the VAAPI supported featureset by running vainfo:

```
vainfo
```

The output on my current testbed is:

```
libva info: VA-API version 1.0.0
libva info: va_getDriverName() returns 0
libva info: Trying to open /usr/local/lib/dri/i965_drv_video.so
libva info: Found init function __vaDriverInit_1_0
libva info: va_openDriver() returns 0
vainfo: VA-API version: 1.0 (libva 2.0.1.pre1)
vainfo: Driver version: Intel i965 driver for Intel(R) CherryView - 2.0.1.pre1 (2.0.0-6-g4997665)
vainfo: Supported profile and entrypoints
VAProfileMPEG2Simple      : VAEntrypointVLD
VAProfileMPEG2Simple      : VAEntrypointEncSlice
VAProfileMPEG2Main        : VAEntrypointVLD
VAProfileMPEG2Main        : VAEntrypointEncSlice
VAProfileH264ConstrainedBaseline: VAEntrypointVLD
VAProfileH264ConstrainedBaseline: VAEntrypointEncSlice
VAProfileH264Main         : VAEntrypointVLD
VAProfileH264Main         : VAEntrypointEncSlice
VAProfileH264High         : VAEntrypointVLD
VAProfileH264High         : VAEntrypointEncSlice
VAProfileH264MultiviewHigh : VAEntrypointVLD
VAProfileH264MultiviewHigh : VAEntrypointEncSlice
VAProfileH264StereoHigh   : VAEntrypointVLD
VAProfileH264StereoHigh   : VAEntrypointEncSlice
VAProfileVC1Simple        : VAEntrypointVLD
```

```

VAProfileVC1Main      : VAEntrypointVLD
VAProfileVC1Advanced  : VAEntrypointVLD
VAProfileNone         : VAEntrypointVideoProc
VAProfileJPEGBaseline : VAEntrypointVLD
VAProfileJPEGBaseline : VAEntrypointEncPicture
VAProfileVP8Version0_3 : VAEntrypointVLD
VAProfileVP8Version0_3 : VAEntrypointEncSlice
VAProfileHEVCMedia    : VAEntrypointVLD

```

### Making a usable FFmpeg build to test the encoders:

```
mkdir -p ~/ffmpeg_sources
```

Include extra components as needed:

(a). Build and deploy nasm: Nasm is an assembler for x86 optimizations used by x264 and FFmpeg. Highly recommended or your resulting build may be very slow.

```

cd ~/ffmpeg_sources
wget wget http://www.nasm.us/pub/nasm/releasebuilds/2.14rc0/nasm-2.14rc0.tar.gz
tar xzvf nasm-2.14rc0.tar.gz
cd nasm-2.14rc0
./configure
make -j$(nproc) VERBOSE=1
make -j$(nproc) install
make -j$(nproc) distclean

```

```

cd ~/ffmpeg_sources
wget http://download.videolan.org/pub/x264/snapshots/last\_x264.tar.bz2
tar xjvf last_x264.tar.bz2
cd x264-snapshot*
./configure --enable-static --disable-openc1
make -j$(nproc) VERBOSE=1
make -j$(nproc) install VERBOSE=1
make -j$(nproc) distclean

```

```

sudo apt-get install cmake mercurial
cd ~/ffmpeg_sources
hg clone https://bitbucket.org/multicoresware/x265
cd ~/ffmpeg_sources/x265/build/linux
cmake -G "Unix Makefiles" -DENABLE_SHARED:bool=off ../..source
make -j$(nproc) VERBOSE=1
make -j$(nproc) install VERBOSE=1
make -j$(nproc) clean VERBOSE=1

```

```

wget http://downloads.xiph.org/releases/ogg/libogg-1.3.2.tar.gz
tar xvzf libogg-1.3.2.tar.gz
cd libogg-1.3.2/
./configure
make -j4
make -j4 install
make -j4 distclean

```

```

cd ~/ffmpeg_sources
wget -O fdk-aac.tar.gz https://github.com/mstorsjo/fdk-aac/tarball/master
tar xzvf fdk-aac.tar.gz
cd mstorsjo-fdk-aac*
autoreconf -fiv
./configure --disable-shared
make -j$(nproc)
make -j$(nproc) install
make -j$(nproc) distclean

```

```

cd ~/ffmpeg_sources
git clone https://github.com/webmproject/libvpx/
cd libvpx
./configure --enable-runtime-cpu-detect --enable-vp9 --enable-vp8 \
--enable-postproc --enable-vp9-postproc --enable-multi-res-encoding --enable-webm-io --enable-vp9-highbitdepth --enable-onthefly-bitpacking
--enable-realtime-only \
--cpu=native --as=yasm
time make -j$(nproc)
time make -j$(nproc) install
time make clean -j$(nproc)
time make distclean

```

```

cd ~/ffmpeg_sources
wget -c -v http://downloads.xiph.org/releases/vorbis/libvorbis-1.3.5.tar.xz
tar -xvf libvorbis-1.3.5.tar.xz

```

```
cd libvorbis-1.3.5
./configure --enable-static
make -j$(nproc)
make -j$(nproc) install
make clean -j$(nproc)
make distclean
```

#### OPTIONAL

```
cd ~/ffmpeg_sources
git clone https://github.com/FFmpeg/FFmpeg -b master
cd FFmpeg
./configure \
--pkg-config-flags="--static" \
--prefix="/apps/ffmpeg/dyn" \
--extra-cflags="-I/apps/ffmpeg/dyn/include" \
--extra-ldflags="-L/apps/ffmpeg/dyn/lib" \
--bindir="/apps/ffmpeg/dyn/bin" \
--enable-debug=3 \
--enable-vaapi \
--enable-libvorbis \
--enable-libvpx \
--enable-gpl \
--cpu=native \
--enable-opengl \
--enable-libfdk-aac \
--enable-libx264 \
--enable-libx265 \
--enable-nonfree\
---extra-libs=-lpthread
make -j$(nproc)
make -j$(nproc) install
make -j$(nproc) distclean
hash -r
```

#### TVHEADEND

```
git clone https://github.com/tvheadend/tvheadend.git
cd tvheadend/
./Autobuild.sh -t precise-amd64
(build autodetects x264 x265 and vaapi)
cd ..
dpkg -i tvheadend_4.*.deb
```

make a streaming profile using h264\_vaapi and use intel\_gpu\_top to see if GPU is used

#### #50 - 2018-02-12 10:24 - Acid Rain

Does this tutorial also work with an Intel Xeon E3-1225-v3 CPU with iGPU and OpenMediaVault v4.x (Debian Stretch 9.x)? (actually I am using Debian 7.x and I don't want to touch the system if transcoding won't be better than now ^^)

#### #51 - 2018-07-31 15:29 - Malte Deiseroth

I just wanted to confirm that this feature is indeed working. I tested on Arch linux using the aur git (version 4.3.something) package and it worked out of the box. Thx for the awesome work.

#### #52 - 2018-09-11 00:17 - Paul M

The ffmpeg in ubuntu 16.04 is too old to have vaapi support, but I found a convenient PPA for it:

<https://launchpad.net/~jon-hedgerows/+archive/ubuntu/ffmpeg-backports>

using a sample command like, I can transcode a .TS file to .MKV and without vaapi it reports up to about 0.8 (which I think means 80% of real time). With vaapi, it reports at least 3.0 times, which means I am getting a gain of 3.7 times performance using the vaapa with my core i3-2100 (sandy bridge). I would expect an even better gain with a more modern Core-i processor with higher end Intel HD GPU!

#### Files

sb_sys_analyzer_linux.py	19.6 KB	2017-08-23	saen acro
putty.log	252 KB	2017-08-23	saen acro