

Tvheadend - Feature #6079

Run script on tune request

2021-08-07 19:12 - Sean Micklem

Status:	New	Start date:	2021-08-07
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			

Description

When tuning certain channels I need to send some specific control commands to another piece of equipment, and another set of commands when I stop watching. I can do this in a recording profile using pre-processor and post processor commands that call shell scripts, but that only works when recording, not when tuning a live channel. I need to be able to run one script when the channel is tuned and an second script when playing stops, and the scripts are channel specific (especially the one at the start of tuning). I found the thread at <https://tvheadend.org/issues/6039> where it was suggested to the OP that this issue be created but he never did, so I am doing so because I'm in a somewhat similar situation to his (however the device I'm using can be programmed to produce a transport stream (.ts) so I don't need to use .m3u8 files or anything like that). There was some talk in that thread about a really kludgy solution but I followed the links and could not figure out what was being suggested (and I don't think it was applicable to what I am trying to do anyway), and in any case it would be much easier to be able to do this from within Tvheadend itself.

The options to send these pre- and post-tuning commands could be associated with the channel, or with the mux (or I suppose even the service), though from a user's perspective it would probably make more sense for it to be an advanced option for the channel, but in my situation there is a 1:1 relationship between mux, service, and channel (there is only one service and channel per mux) so it really would not matter.

History

#1 - 2021-08-07 20:08 - saen acro

Have you try to use Spawn profile to send commands.
It's not real solution but is a workaround.

#2 - 2021-08-07 21:55 - Sean Micklem

saen acro wrote:

Have you try to use Spawn profile to send commands.
It's not real solution but is a workaround.

For one thing I can't seem to find any instructions on how to do that, but also it appears that's only available in the unstable branch, and I was making a feature request for a future stable version. I know very little about Linux, so the thought of running unstable software and possibly running into issues I can't resolve is quite frightening. Even doing normal kernel updates on this system (which I only do for security reasons) scares the hell out of me. So, I stick to stable versions of Tvheadend.

#3 - 2021-08-07 22:09 - Sean Micklem

Also, it appears that the Spawn profile is really intended for a completely different purpose, and I don't see any way at all that it could be adapted to send a command, then tune a channel, then when streaming the channel is ended, send a different command. I could guess that if someone really knew what they were doing (or if someone could explain how to use that profile) you could MAYBE do one of those, but I need to be able to do both. But really that seems intended for use in transcoding a stream, which is not what I'm doing, at least not as far as I know. My system absolutely cannot transcode video in real time; it can transcode audio but I'm not doing that in this case.

#4 - 2021-08-08 09:40 - saen acro

Unstable is enough stable ;)
You can **PIPE** and use stream in bash script.

#5 - 2021-08-08 12:23 - Sean Micklem

Well I thought of that, but there is a problem. Let's say that in a mux in an IPTV network, instead of calling the command directly I call a bash script, e.g. `pipe:///home/user/my_script.sh`, and then in `my_script.sh` I put the actual call to whatever I would have gone to directly from the pipe command (for example a call to `ffmpeg`). This works. And I can add bash commands or calls to other scripts (with an `&` following to put them in the background) prior to making that call to whatever delivers the actual stream. So far, so good. But what I cannot do is put a command **AFTER** whatever delivers the stream, because it never gets executed. Once Tvheadend stops the stream it apparently kills the process forcefully, which I suppose is desirable so that streams don't continue to play even when there is nothing receiving them. So I had the thought to launch another script as a background task, that would continuously keep checking to see if the pid of the parent script still existed, and if not then it would run the commands I wanted to run.

The problem is that when Tvheadend closes the stream it somehow also closes that spawned secondary script. It does not matter if I use & to put it in the background, or nohup prior to the command. So, running something before the stream begins playing is a relative piece of cake, but running something after it stops appears to be impossible.

I can run my secondary script manually and it runs fine, and watches the first script (the one called from the pipe:// URL) and when that script ends it runs what it should. But for whatever reason, I can't launch that second script from within the first script, because if I do then it will end the moment the stream is shut down. I have read several pages that say that using & to run the script in the background OR using nohup at the start of the line should prevent that, but it doesn't. I was up all night trying to make this work, and it would have been more productive to beat my head against a rock, at least then at some point I would have knocked myself unconscious and got some sleep!

#6 - 2021-08-08 16:31 - Dave Pickles

When you wake up, you could try invoking your watchdog program using 'at'. Something like (untested)

```
at -f /path/to/watchdog.sh -t now
```

Depending on distribution you may need to install the package containing 'at'.

#7 - 2021-08-08 18:47 - Sean Micklem

Dave Pickles wrote:

When you wake up, you could try invoking your watchdog program using 'at'. Something like (untested)

```
at -f /path/to/watchdog.sh -t now
```

Depending on distribution you may need to install the package containing 'at'.

Thanks for the suggestion, preliminary tests show that a script launched with "at" does survive the closure of the stream (your syntax was wrong but I was able to figure that out, you did say it was untested). This is getting unbelievably kludgy though, which is why I would still very much prefer that this capability be added to Tvheadend if that is possible. But at now least it feels like this might be workable if not elegant.