FFMEPG

and FFPLay

ffmpeg has many uses, and one area that is very useful for us is to be able to extract from either SPTS or MPTS transport stream (.ts) files to .mp4. There are a lot more players that can handle .mp4, and when testing UHD streams, most TVs will play .mp4 from a memory stick, not many will handle .ts (especially if it's MPTS).

Chris Duxbury has worked out a simple syntax we can use to extract our .ts files to .mp4 files (.mp4 is a valid container for both h.264 and HEVC steams).

FFMEPG

The following syntax will work for SPTS files (not sure about MPTS)

```
ffmpeg -probesize 50000000 -i source.ts -map 0:0 -map 0:1 -map 0:2 -c copy destination.mp4
```

The probesize bit tells ffmpeg to look over lots of file for the I picture, otherwise on long GOP it seems to give up sometimes. (like most decoders, ffmpeg can't start to decode until it finds an I frame)

Using -map 0:0 -map 0:1 -map 0:1 -map 0:2 tells ffmpeg to select/map specified input streams to output in that order.

Using -c copy mark all the streams to be just copied to the output

FFPLAY

FFPLAY comes bundled with the ffmpeg download. The ffplay is a simple to use command line app for playing the extracted streams that you have just created.

The simple use of fflplay is just:

```
ffplay filename.mp4
```

However, the issue with UHD streams is that they are larger than most monitors, so it is generally necessary to either rescale the output, or force it to fullscreen.

To replay a video in a window.

```
ffplay -i video.mpg -vf scale=800x600
This will replay the selected file at a window size of 800x600. Remember, currently (End 2016)
UHD streams play very poorly and you will only get 1 or 2 frames per second.
```

To replay a video full screen.

```
ffplay -i video.mpg -vs
This will replay the video at fullscreen resolution.
```

FFPROBE

FFPROBE allows the inspection of a file for some basic information. ffprobe is useful to analyse a .ts file before extraction to see what streams are present, the PIDs etc.

```
ffproble filename.ts
Input #0, mpegts, from 'AVP4000_HEVC_SDR.ts':
  Duration: 00:01:01.45, start: 63552.709811, bitrate: 24918 kb/s
  Program 1
```

```
    Metadata:
      service_name    : AVPUHD
      service_provider: Ericsson
    Stream #0:0[0x3e9]: Video: hevc (Main 10) ([36][0][0][0] / 0x0024), yuv420p1
0le(tv, bt709), 3840x2160 [SAR 1:1 DAR 16:9], 50 fps, 50 tbr, 90k tbn, 50 tbc
    Stream #0:1[0x7d1](eng): Audio: mp2 ([3][0][0][0] / 0x0003), 48000 Hz, stereo, s16p, 192 kb/s
```

In the above example we can see the following:

Duration: 00:01:01.45, start: 63552.709811, bitrate: 24918 kb/s The Program 1
The service_name is AVPUHD
The service_provider is Ericsson
The Stream #0:0[0x3e9] 0x3E9 is Dec 1001 is Video: hevc (Main 10) ([36][0][0][0] / 0x0024), yuv420p1
The Stream #0:1[0x7d1](eng) <color red>0x7d1 is Dec 2001<color> is Audio: mp2 ([3][0][0][0] / 0x0003), 48000 Hz, stereo, s16p, 192 kb/s.