EMBRACING OPEN SOURCE

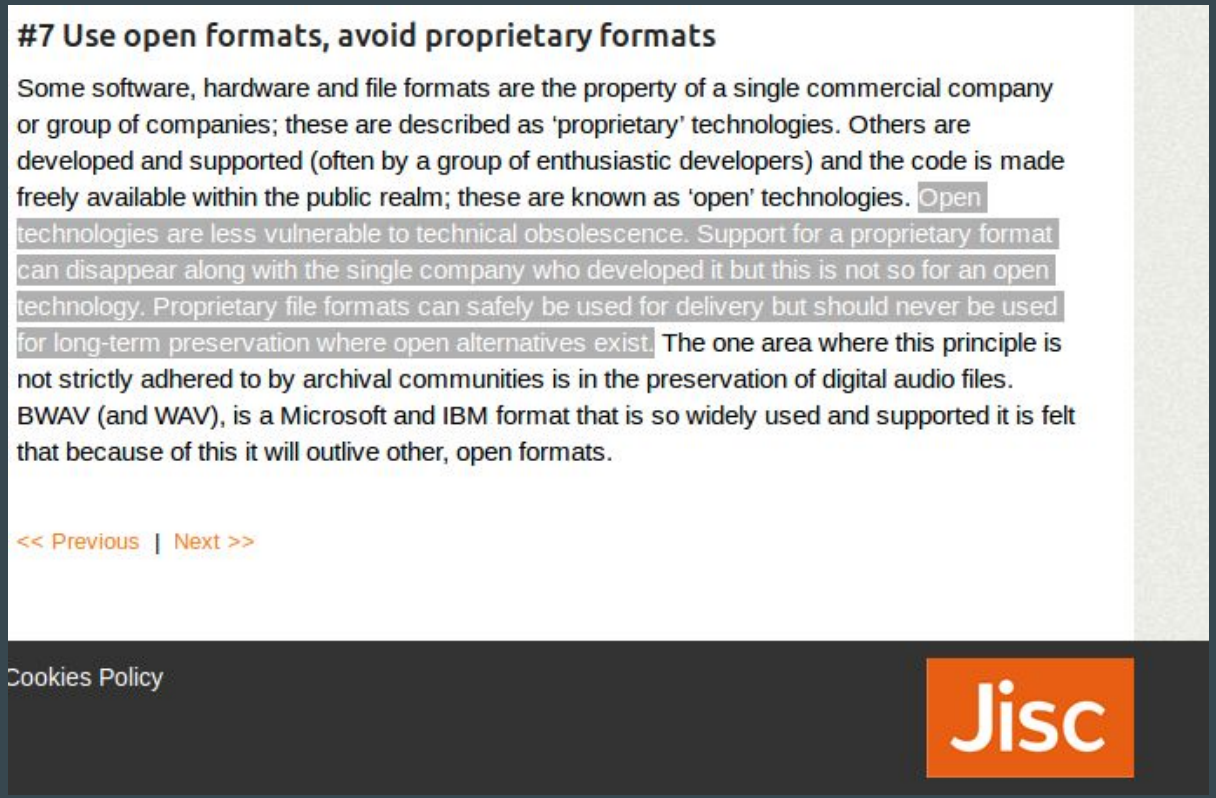
Kieran O'Leary - IFI Irish Film Archive
#FIAF2018 - Prague 2018-04-24
@kieranjol

Irish Film Institute

IFI Irish Film Archive

- Professional and amateur film and video
- Library
- Paper and artefact collections
- Born digital cinema and broadcast deposits
- 25% Funding from Irish Arts Council
- Other funding generated through partners such as The Irish Film Board, Broadcasting Authority of Ireland, The Arts Council

Why open formats?



#7 Use open formats, avoid proprietary formats

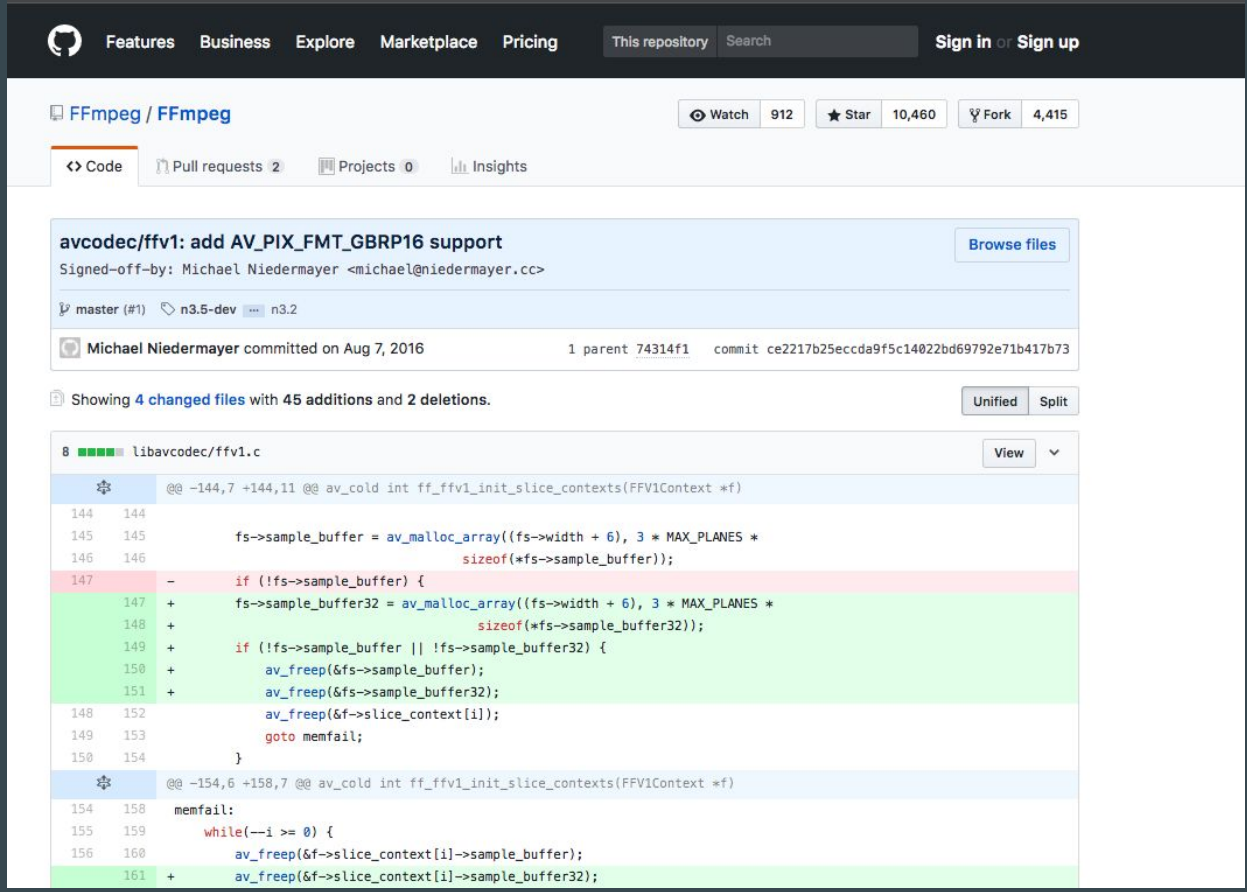
Some software, hardware and file formats are the property of a single commercial company or group of companies; these are described as 'proprietary' technologies. Others are developed and supported (often by a group of enthusiastic developers) and the code is made freely available within the public realm; these are known as 'open' technologies. Open technologies are less vulnerable to technical obsolescence. Support for a proprietary format can disappear along with the single company who developed it but this is not so for an open technology. Proprietary file formats can safely be used for delivery but should never be used for long-term preservation where open alternatives exist. The one area where this principle is not strictly adhered to by archival communities is in the preservation of digital audio files. BWAV (and WAV), is a Microsoft and IBM format that is so widely used and supported it is felt that because of this it will outlive other, open formats.

<< Previous | Next >>

Jisc

Why FFV1?

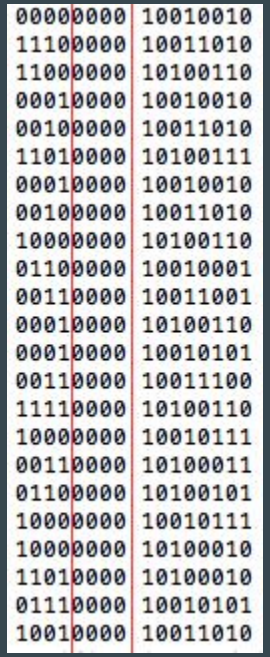
- Free/Open
- Self-descriptive (doesn’t rely on container for context)
- Actively standardised by IETF via CELLAR
- MediaConch conformance checker via PREFORMA
- Embedded checksums per frame/slice
- Helpful archive/FFmpeg community
- Potential to sponsor developments
- Lossless compression


Features Business Explore Marketplace Pricing
This repository Search
Sign in or Sign up
FFmpeg / FFmpeg
Watch 912 Star 10,460 Fork 4,415
Code Pull requests 2 Projects 0 Insights
avcodec/ffv1: add AV_PIX_FMT_GBRP16 support
Browse files
Signed-off-by: Michael Niedermayer <michael@niedermayer.cc>
master (#1) n3.5-dev ... n3.2
Michael Niedermayer committed on Aug 7, 2016
1 parent 74314f1 commit ce2217b25eccda9f5c14022bd69792e71b417b73
Showing 4 changed files with 45 additions and 2 deletions.
Unified Split
8 libavcodec/ffv1.c
View
@@ -144,7 +144,11 @@ av_cold int ff_ffv1_init_slice_contexts(FFV1Context *f)
fs->sample_buffer = av_malloc_array((fs->width + 6), 3 * MAX_PLANES *
sizeof(*fs->sample_buffer));
- if (!fs->sample_buffer) {
+ fs->sample_buffer32 = av_malloc_array((fs->width + 6), 3 * MAX_PLANES *
+ sizeof(*fs->sample_buffer32));
+ if (!fs->sample_buffer || !fs->sample_buffer32) {
+ av_freep(&fs->sample_buffer);
+ av_freep(&fs->sample_buffer32);
av_freep(&f->slice_context[i]);
goto memfail;
}
@@ -154,6 +158,7 @@ av_cold int ff_ffv1_init_slice_contexts(FFV1Context *f)
while(--i >= 0) {
av_freep(&f->slice_context[i]->sample_buffer);
+ av_freep(&f->slice_context[i]->sample_buffer32);

Early lessons learned from community

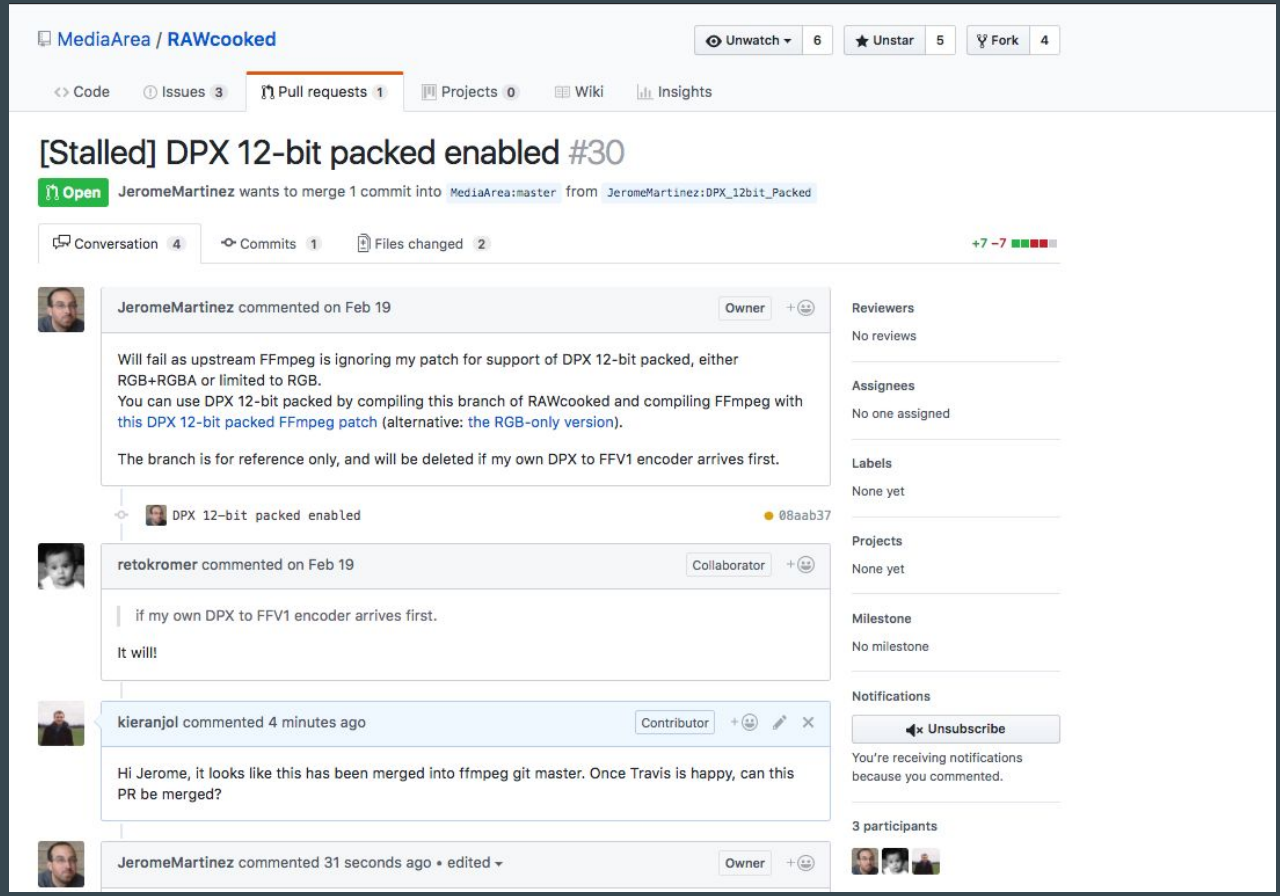
- Paid, professional support available
- Possible to engage in file format specification (CELLAR)
- New features can be added through financial support
- Avoid vendor-lock
- Rethink licensing
- Command line is to be embraced, not feared!

Padding


00000000 10010010
11100000 10011010
11000000 10100110
00010000 10010010
00100000 10011010
11010000 10100111
00010000 10010010
00100000 10011010
10000000 10100110
01100000 10010001
00110000 10011001
00010000 10100110
00010000 10010101
00110000 10011100
11110000 10100110
10000000 10010111
00110000 10100011
01100000 10100101
10000000 10010111
10000000 10100010
11010000 10100010
01110000 10010101
10010000 10011010

```
ffmpeg version git-2018-04-10-d64183e Copyright (c) 2000-2018 the FFmpeg developers
  built with Apple LLVM version 8.0.0 (clang-800.0.42.1)
  configuration: --prefix=/usr/local/Cellar/ffmpeg/HEAD-d64183e --enable-shared --enable-pthreads --enable-version3 --enab
le-hardcoded-tables --enable-avresample --cc=clang --host-cflags= --host-ldflags= --enable-gpl --enable-ffplay --enable-li
bass --enable-libfreetype --enable-libmp3lame --enable-libtesseract --enable-libx264 --enable-libxvid --enable-opencl --en
able-videotoolbox --disable-lzma --enable-libopenjpeg --disable-decoder=jpeg2000 --extra-cflags=-I/usr/local/Cellar/openjp
eg/2.3.0/include/openjpeg-2.3
  libavutil      56. 13.100 / 56. 13.100
  libavcodec     58. 17.100 / 58. 17.100
  libavformat    58. 11.101 / 58. 11.101
  libavdevice    58.  2.100 / 58.  2.100
  libavfilter     7. 14.100 /  7. 14.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  0.102 /  5.  0.102
  libswresample   3.  0.101 /  3.  0.101
  libpostproc    55.  0.100 / 55.  0.100
[dpx @ 0x7f8e99811000] Packing to 16bit required
[dpx_pipe @ 0x7f8e9900c800] Stream #0: not enough frames to estimate rate; consider increasing probesize
[dpx_pipe @ 0x7f8e9900c800] decoding for stream 0 failed
[dpx_pipe @ 0x7f8e9900c800] Could not find codec parameters for stream 0 (Video: dpx, none, 1600x1168 [SAR 1:1 DAR 100:73]
): unspecified pixel format
Consider increasing the value for the 'analyzeduration' and 'probesize' options
Input #0, dpx_pipe, from '6e0edc3b-7d89-4710-8b9a-66c8c990f9dc_1_00094787.dpx':
  Duration: N/A, bitrate: N/A
    Stream #0:0: Video: dpx, none, 1600x1168 [SAR 1:1 DAR 100:73], 24 tbr, 25 tbn, 24 tbc
At least one output file must be specified
ifi-mac-pro:12bitdpx admin$
```

```
ifi-mac-pro:12bitdpx admin$ ffmpeg -i 6e0edc3b-7d89-4710-8b9a-66c8c990f9dc_1_00094787.dpx -c:v ffv1 -level 3 -slicecrc 1 -
slices 24 -g 1 ffv1.mkv -f md5 -
ffmpeg version N-45206-gc116221 Copyright (c) 2000-2018 the FFmpeg developers
  built with Apple LLVM version 8.0.0 (clang-800.0.42.1)
  configuration: --prefix=/usr/local/Cellar/ffmpeg/HEAD-c116221 --enable-shared --enable-pthreads --enable-version3 --enab
le-hardcoded-tables --enable-avresample --cc=clang --host-cflags= --host-ldflags= --enable-gpl --enable-ffplay --enable-li
bass --enable-libfreetype --enable-libmp3lame --enable-libtesseract --enable-libx264 --enable-libxvid --enable-opencl --en
able-videotoolbox --disable-lzma --enable-libopenjpeg --disable-decoder=jpeg2000 --extra-cflags=-I/usr/local/Cellar/openjp
eg/2.3.0/include/openjpeg-2.3
  libavutil      56. 15.100 / 56. 15.100
  libavcodec     58. 19.100 / 58. 19.100
  libavformat    58. 13.100 / 58. 13.100
  libavdevice    58.  4.100 / 58.  4.100
  libavfilter     7. 17.100 /  7. 17.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  2.100 /  5.  2.100
  libswresample   3.  2.100 /  3.  2.100
  libpostproc    55.  2.100 / 55.  2.100
[dpx_pipe @ 0x7fae7400c800] Stream #0: not enough frames to estimate rate; consider increasing probesize
Input #0, dpx_pipe, from '6e0edc3b-7d89-4710-8b9a-66c8c990f9dc_1_00094787.dpx':
  Duration: N/A, bitrate: N/A
    Stream #0:0: Video: dpx, gbrp12le, 1600x1168 [SAR 1:1 DAR 100:73], 24 tbr, 25 tbn, 24 tbc
Stream mapping:
  Stream #0:0 -> #0:0 (dpx (native) -> ffv1 (native))
  Stream #0:0 -> #1:0 (dpx (native) -> rawvideo (native))
Press [q] to stop, [?] for help
[ffv1 @ 0x7fae74801e00] bits_per_raw_sample > 8, forcing range coder
Output #0, matroska, to 'ffv1.mkv':
  Metadata:
    encoder         : Lavf58.13.100
    Stream #0:0: Video: ffv1 (FFV1 / 0x31564646), gbrp12le, 1600x1168 [SAR 1:1 DAR 100:73], q=2-31, 200 kb/s, 24 fps, 1k t
bn, 24 tbc
    Metadata:
      encoder         : Lavc58.19.100 ffv1
Output #1, md5, to 'pipe:':
  Metadata:
    encoder         : Lavf58.13.100
    Stream #1:0: Video: rawvideo (G3[0][12] / 0xC003347), gbrp12le, 1600x1168 [SAR 1:1 DAR 100:73], q=2-31, 1614643 kb/s,
24 fps, 24 tbn, 24 tbc
    Metadata:
      encoder         : Lavc58.19.100 rawvideo
MD5=aa43e3ec1c2a8e4499b3bc796a35185b
frame=    1 fps=0.0 q=-0.0 Lq=-0.0 size=    4800kB time=00:00:00.04 bitrate=943754.0kbits/s speed=0.167x
video:15749kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
ifi-mac-pro:12bitdpx admin$ ffmpeg -i ffv1.mkv -f md5 -
```


MediaArea / RAWcooked
Unwatch 6
Unstar 5
Fork 4
Code
Issues 3
Pull requests 1
Projects 0
Wiki
Insights
[Stalled] DPX 12-bit packed enabled #30
Open
JeromeMartinez wants to merge 1 commit into MediaArea:master from JeromeMartinez:DPX_12bit_Packed
Conversation 4
Commits 1
Files changed 2
+7 −7
JeromeMartinez commented on Feb 19
Owner
Will fail as upstream FFmpeg is ignoring my patch for support of DPX 12-bit packed, either RGB+RGBA or limited to RGB.
You can use DPX 12-bit packed by compiling this branch of RAWcooked and compiling FFmpeg with this DPX 12-bit packed FFmpeg patch (alternative: the RGB-only version).
The branch is for reference only, and will be deleted if my own DPX to FFV1 encoder arrives first.
DPX 12-bit packed enabled
08aab37
retokromer commented on Feb 19
Collaborator
if my own DPX to FFV1 encoder arrives first.
It will!
kieranjol commented 4 minutes ago
Contributor
Hi Jerome, it looks like this has been merged into ffmpeg git master. Once Travis is happy, can this PR be merged?
JeromeMartinez commented 31 seconds ago • edited
Owner
Reviewers
No reviews
Assignees
No one assigned
Labels
None yet
Projects
None yet
Milestone
No milestone
Notifications
Unsubscribe
You're receiving notifications because you commented.
3 participants

[FFmpeg-devel] avcodec/dpx: Support for RGB 12-bit packed decoding

Submitted by Jerome Martinez on Feb. 14, 2018, 12:46 p.m.

Details

Message ID	133b22e5-2849-4cde-8da7-8332b4ae8a77@mediaarea.net
State	New
Headers	show

Commit Message

Jerome Martinez — Feb. 14, 2018, 12:46 p.m.

```
On 08/02/2018 11:28, Jerome Martinez wrote:
> Currently RGB and RGBA 12-bit are supported by DPX decoder only if
> component values are padded (packing "Filled to 32-bit words, method A").
> This patch adds decoding of RGB and RGBA 12-bit with no padding
> (packing "Packed into 32-bit words").
>
> As I have no file with line boundaries not aligned on 32-bit, I can
> not have good tests about the stride computing (so code about non
> aligned boundaries is theory) so I preferred to limit risks by
> decoding only if line boundaries are aligned on 32-bit words:
> - 8 pixels for RGB (8 pixels x 3 components x 12 bits = 288 bits = 9 x
> 32-bit words)
> - 2 pixels for RGBA (2 pixels x 4 components x 12 bits = 3 x 32-bit
> words)
>
> I think Little Endian parsing is fine thanks to the generic code about
> Big vs Little endian but I have no Little Endian test file so I also
> limited the decoding to Big Endian files.
>
> Would be happy to check with cases I was not able to check if someone
> provides files.
>
```

FFmpeg

python™

QC TOOLS

Drag a file on to Siegfried's anvil!

seq2ffv1.py

- Work in progress -more testing to be done.
- Recursively batch process image sequence folders and transcode to a single ffv1.mkv.
- Framemd5 files are generated and validated for losslessness.
- Whole file manifests are also created.
- Usage - `seq2ffv1.py parent_folder`

seq2prores.py

- Specific IFI workflow that expects a particular folder path:
- Recursively batch process image sequence folders with seperate WAV files and transcode to a single Apple Pro Res HQ file in a MOV container. PREMIS XML log files are generated with hardcoded IFI values for the source DPX sequence and the transcoded mezzanine file in the respective /metadata directory
- A whole file MD5 manifest of everything in the SIP are also created. Work in progress - more testing to be done.
- Usage - `seq2prores.py directory`
- seq2prores accepts multiple parent folders, so one can run `seq2prores.py directory1 directory2 directory3` etc

rawbatch.py

- Specific IFI workflow that expects a particular folder path:
- Recursively batch processes image sequence folders with seperate WAV files, generating PREMIS XML log files with hardcoded IFI values.
- A duplicate audio WAV file is created and sent to desktop as workhorse.
- A whole file MD5 manifest of everything in the SIP are also created. Work in progress - more

README.rst

Clairmeta

Clairmeta is a python package for Digital Cinema Package (DCP) probing and checking.

This project status is **Beta**, the following needs to be done for the release :

- Large scale tests on lots of DCPs (including D-Box, DVIs, OCAP, CCAP, ...)

Features

- DCP Probe : metadata extraction of the whole DCP, including all XML fields and MXF assets inspection.
- DCP Checker : advanced DCP validation tool, including (non exhaustive) :
  - SMPTE / Interop standard convention (naming, ...)
  - Integrity (MIME type, size, hash) of all assets
  - Foreign file identification
  - XSD Schema validation for XML files (VOLINDEX, ASSETMAP, CPL, PKL)
  - Digital signature validation (CPL, PKL)
  - Intra / Inter Reels integrity and coherence
  - Metadata match between CPL assets and MXF headers
  - Re-link VF / OV
  - Picture tests : FrameRate, BitRate, ...
  - Sound tests : Channels, Sampling, ...
  - Subtitle : Deep inspection of Interop and SMPTE subtitles
- DSM / DCDM Checker : basic image file sequence validation with some specific rules.

http://github.com/kieranjol/ifiscripts

http://ifiscripts.readthedocs.io/en/latest/

https://mediaarea.net/RAWcooked