

strategies and tactics for high-volume digitisation and digital preservation

edward anderson
digital collections development manager
@anderson_edw



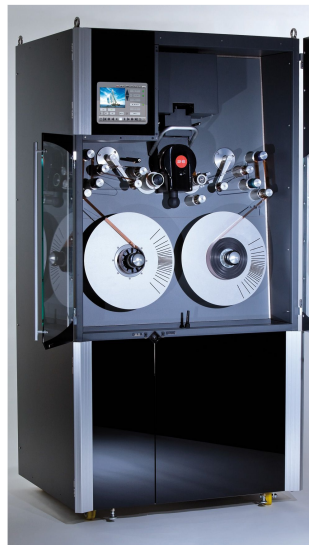
stephen mcconnachie
head of data and digital preservation
@mcnatch

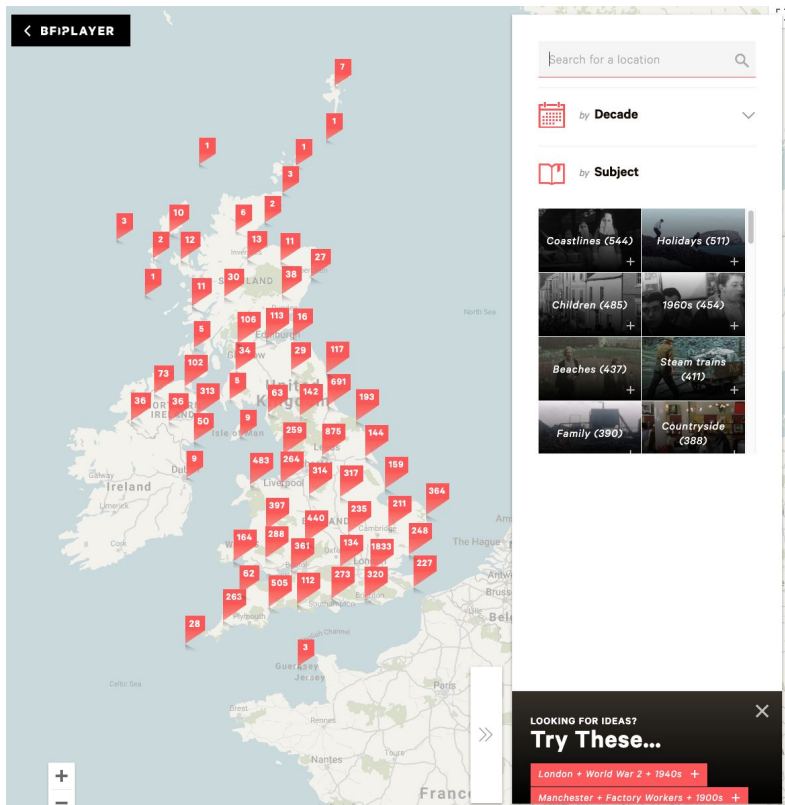
40,000	fiction films	19,000	posters
130,000	nonfiction films	1.6m	photographs
780,000	television progs	5,000	production + costume designs
21,000	scripts	45,000	library books
28,000	press books	600	special collections

BFI master film store

top secret location

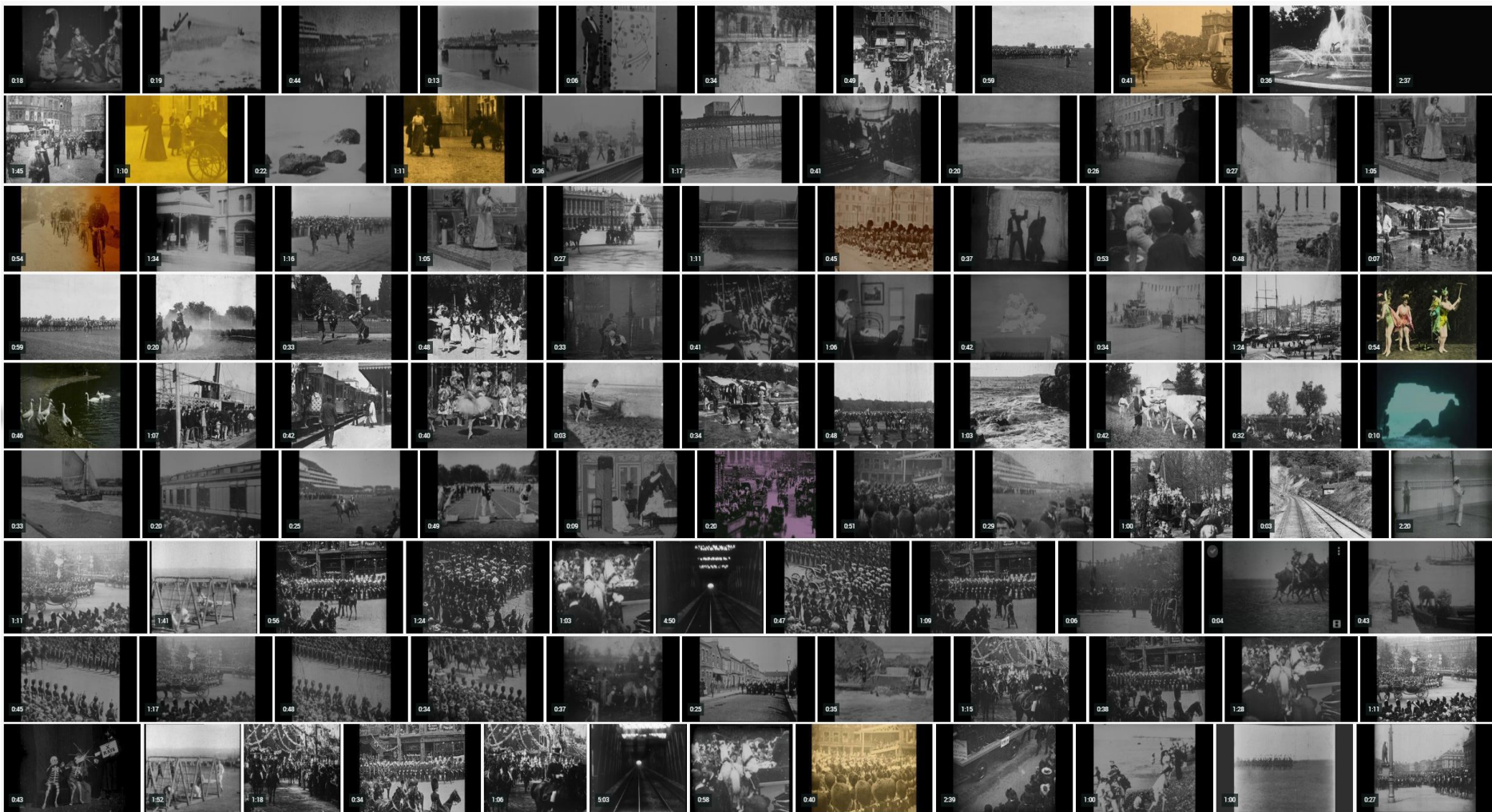






Unlocking Film Heritage project

- digitise + digitally preserve **10,000 films**
- from all of the uk's film archives
- including entirety of Victorian-era film
- make the films available online
- with a map interface (yay, geolocations!)
- procure, integrate + deploy digital preservation infrastructure
- consolidate + extend film digitisation capability



'We need Mr & Mrs in 200 years': why the BFI is saving 100,000 old TV shows

Gems of bygone era have already been digitised but now the BFI is hoping to preserve lesser known shows at risk of being lost



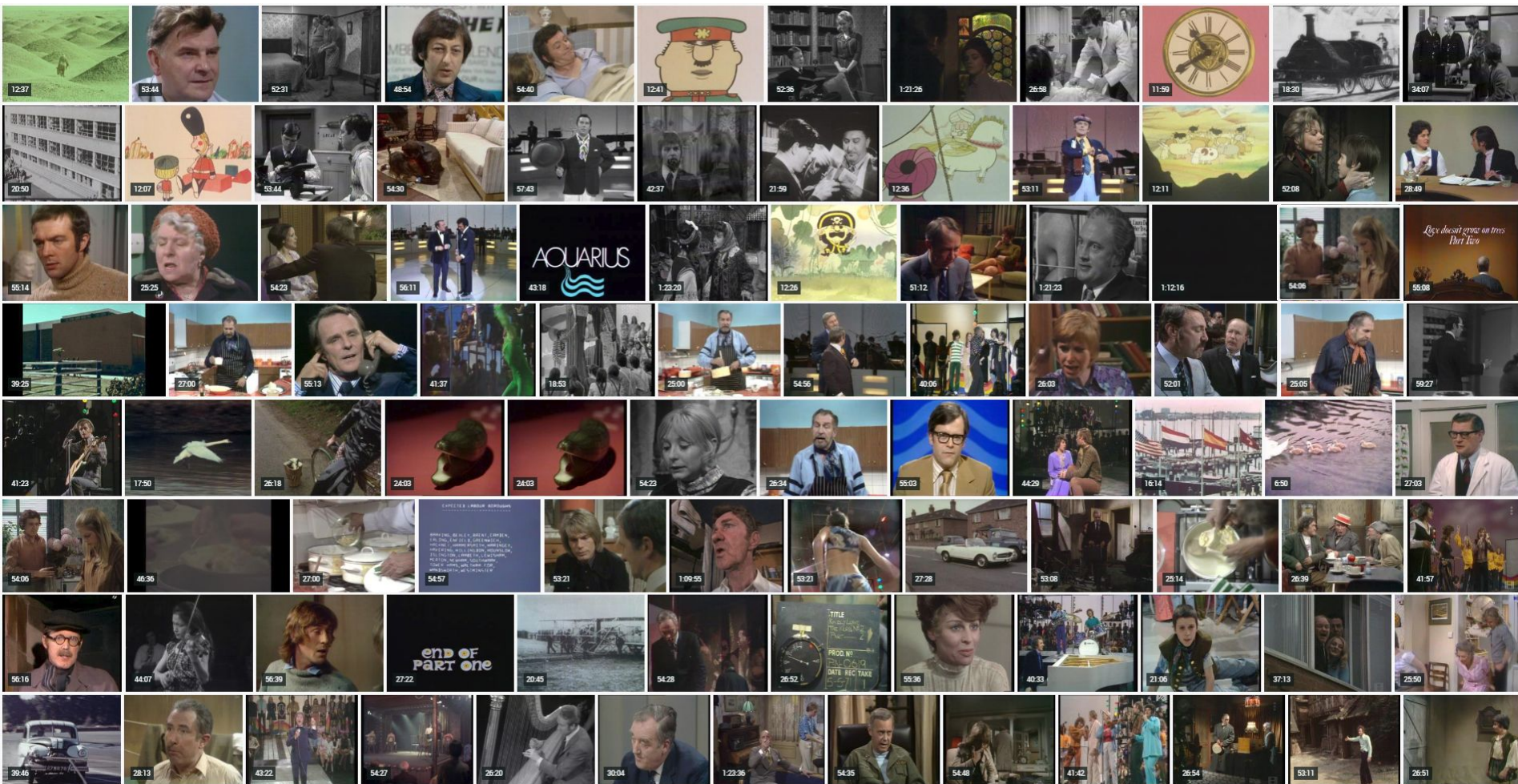
▲ Could they, should they be preserved? Clockwise from top left: The Basil Brush Show, Mr & Mrs, TV-AM, an earlier presenting line-up on TV-AM, Tiswas and the Bay City Rollers. Composite: Rex

An estimated 100,000 British television programmes - which could include early editions of TV-AM, children's series such as Tiswas, and fondly remembered, for some, episodes of Border TV's Mr & Mrs - are to be digitised to prevent them being lost forever.

The British Film Institute has announced the mass digitisation project to save programmes which are held on obsolete video formats, warning that there are only five or six years left to save those most at risk.

BFI2022 video digitisation project

- digitise + digitally preserve **100,000 works from videotape**
- broadcast and non-broadcast content
- from all of the uk's film archives
- consolidate + extend video digitisation capability

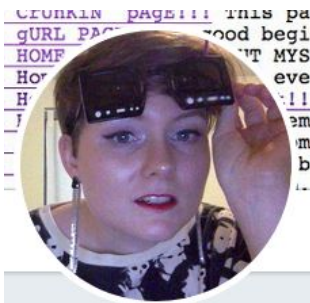


strategies and tactics
for digitisation and digital
preservation

~~strategies and tactics
for digitisation and digital
preservation~~

How on earth do you do
digital preservation of
10,000 films and 100,000
tv progs in 10 years?

1. community



→ **Ashley Blewer!** ➔

@ablwr



Erwin Verbruggen

@erwinverb



Kieran O'Leary

@kieranjol



Dave Rice

@dericed



Jérôme Martinez

@JeromeM78



Joanna White
@digitensions



Ben Turkus
@bturkus



Joshua Ng
@joshuatg



Callie Holmes
@cally_ho



Peter Bubestinger
@pjotrek_b



NO TIME TO WAIT #1, 2, 3, 4

<https://mediaarea.net/NoTimeToWait3>

Joanna White @digitensions · 29 Oct 2018

The community welcome at #ntw3 was overwhelming, thank you. I hope the connections made can turn into friendships that help build support networks for all struggling archivist-developers. Let's connect and find somewhere to talk and share our discoveries!



Joanna White on LinkedIn: "It's been an honour to participate in the No ... October 27, 2018: Joanna White posted images on LinkedIn [linkedin.com](#)"

👍 1 🔄 5 ❤️ 17 📌

No Time to Wait 3 Program			
A 2 day symposium on the intersection of open media, standardization & audiovisual preservation. October 25 and 26, 2018 @ the BFI https://mediaarea.net/NoTimeToWait3 Questions? Please contact us at info@mediaarea.net .			
Time	Speaker / Chair	Description	Type
Thursday, October 25			
8:30	Alessandra Luciano	No Time to Wait Introduction	
9:15		Standards, standards, standards	
9:30	Richard Wright	Review of Audiovisual Presentation	Presentation
9:55	Lars Gauselud	Presenting the Initial ISA TC 06	Presentation
10:20	Jim Jones	So Many Standards, So Little Time: Initial Findings and Analysis	Presentation
10:45	Dave Rice	Status of CELLAR	Lightning Talk
10:55	Sophie Buzi (chair)	How we FFV & How we Matroska. Panelists: Joanna White, Carl Eugen Hoyon, Dave Rice	Roundtable
11:25	Marin Wrigley, OPF	Preservation Action Registry	Presentation
11:50	Reto Komer	Beyond RGB: Multispectral imaging and FFV1	Lightning Talk
12:00		Lunch	
13:00	Peter Bubesinger-Stenel	Open Source & Long Term	Presentation
13:25	Everitha Samaras	Archiving VFX: A research project to preserve evidence of film digital visual effects production	Lightning Talk
FIRE ALARM! Periton our			
Break			
14:00	Jack McConchie, Tom Ensom	Developing a High Level Preservation Strategy for Virtual Reality Archives	Presentation
14:50	Stephen McConachie (chair)	Emerging Media: collecting, preservation and access challenges. Panelists: Erwin Verbruggen, Judy Wilcocks, Carlin Smith, Patricia Faloutsos	Panel
15:35	Ernesto Coto and Andrew Brown, Oxford University	Automated Tagging of Image and Video Collections using Face Recognition	Presentation
The Next Steps of Archives			
16:00	Kieran Kuntz	Supporting niche formats and niche hardware in open source software and operating systems	Presentation
16:25	Derek Butlerhuis	Every Solution is Wrong: Normalizing Ambiguous, Broken, and Parts-on-Head Crazy Media	Presentation
16:50	Ben Cartwright-Cox	Archiving DAB/DAB+ Feeds	Lightning Talk
17:00		Closing Comments	
18:00		Dinner	

Engage

AMIA-L Listserv

The AMIA-L list a forum for discussion among professionals interested in issues relevant to issues any and all involving all aspects of moving image materials and archives.



+ Performance Of Abel Gance's 1927 Masterpiece Napoleon, September 24, 2007 At The Coliseum In Rome Wi...

AMIA sponsors the AMIA-L list as a forum for discussion among professionals interested in issues relevant to issues involving all aspects of moving image materials and moving image archives, and to any related technologies or special interests of the profession. AMIA members and non-members are welcome, and subscription to AMIA-L is open to the public.

 [AMIA Code of Conduct](#)

 [Public Listserv Rules and Guidelines](#)

<https://amianet.org/engage/amia-l-listserv/>

Select List:

AMIA-L Association of Moving Image Archivists

Search For

2 String:

ffv1 matroska

2 Substring

☐
Advanced Search

2 Subject Contains:



2 Author's Address:

2 Since:

2 Until:

Search

LSV.UKY.EDU (48 Matches)

Subject	From	Date	Time
Re: Apple TN2162	Jim Lindner <jim@media-matters.net>	2019-01-19	09:38
Apple TN2162	Kevin Irelan <ganesh.irelan@syda.org>	2019-01-15	09:12
Re: Lossless compression flavor of the archive	AMIA-L <hdtv@earthlink.net>	2018-11-16	19:12
Re: Lossless compression flavor of the archive	Isaiah Beard <isaiah.beard@rutgers.edu>	2018-11-13	16:11
Lossless compression flavor of the archive	Kevin Irelan <ganesh.irelan@syda.org>	2018-11-13	15:15
Re: Lossless compression flavor of the archive [SEC=UNOFFICIAL]	Jim Lindner <jim@media-matters.net>	2018-11-13	04:38
Re: Lossless compression flavor of the archive [SEC=UNOFFICIAL]	Carl Flesichhauer <cflie.work@gmail.com>	2018-11-12	10:32
Re: DPX wrapped in .tar	Stephen McConnachie <stephen.mcconnachie@bfi.org.uk>	2018-08-15	12:13
Re: [SPAM] Re: [AMIA-L] [SPAM] [AMIA-L] Frv1--Try ffWorks, successor to IFFmpeg--link	Ted Langdell <preservation@tedlangdell.com>	2018-05-11	12:57
Re: [SPAM] [AMIA-L] Frv1--Try ffWorks, successor to IFFmpeg--link	Shai Drori <shayke888@gmail.com>	2018-05-10	23:43
Re: [SPAM] [AMIA-L] Frv1--Try ffWorks, successor to IFFmpeg--link	Preservation@tedlangdell.com <preservation@tedlangdell.com>	2018-05-10	19:31
Re: AVI and Pro Res	Jim Lindner <jim@media-matters.net>	2018-04-11	10:49
Re: AVI and Pro Res	David Crosthwait <david@dcvideo.com>	2018-04-09	18:07
Re: AVI and Pro Res	Carl Flesichhauer <cflie.work@gmail.com>	2018-04-07	10:50
Re: AVI and Pro Res	Isaiah Beard <isaiah.beard@rutgers.edu>	2018-04-06	15:01
Re: AVI and Pro Res	Stephen McConnachie <stephen.mcconnachie@bfi.org.uk>	2018-04-06	13:16



Home > Publications > IASA Special and Technical Publications

IASA-TC 06 Guidelines for the Preservation of Video Recordings

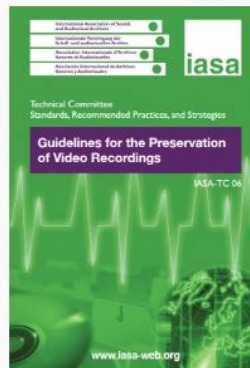
Archives hold original video recordings in a range of types, from media-dependent, carrier-based analogue videotapes to computer-file-based digital recordings. The appropriate preservation treatments for this array reflect the variation in the source recordings. For analogue videotapes, for example, *digitisation* is called for. Meanwhile, examples of digital file-based recording may require rewrapping into a fresh file "wrapper" or a combination of digital *transcoding* and *rewrapping*.

When complete, IASA-TC 06 will cover the full range of topics in the preceding paragraph, as well as providing advice on shooting ethnographic, documentary, and oral history video footage in a manner that maximizes its "preserve-ability".

IASA-TC 06 is being disseminated in phases. At a high level, there are two dissemination phases:

1. Initial edition, publicly accessible in early 2018: focus on the preservation of video recordings on conventional carriers. For the most part, this means the digitization of analogue videotapes, but it also encompasses the transfer of content from some types of digital videotapes.
2. Expanded edition, access planned for in 2019: discussion of the preservation of digital-file-based video formats and the transfer of digitally encoded recordings in videotape form (which may entail transcoding and/or rewrapping). Additional sections are anticipated that pertain to metadata and the production of new recordings in preservable formats.

This initial presentation on the IASA website is for *comment* and will support future revisions and improvements. Readers are asked to send notes to the chair of the IASA Technical Committee: Lars Gaustad (lars.gaustad@nb.no). Lars will share them with the guideline's authors and, from time to time, the authors will summarize the comments together with their responses either via the IASA blog (www.iasa-web.org/blog) or one of the forums (e.g. www.iasa-web.org/forums/public-forums/technical-issues).



Technical Committee
Standards, Recommended Practices, and Strategies

Guidelines for the Preservation of Video Recordings

IASA-TC 06

Part D. Planning, Setup,
and Workflows for Video Digitisation x

From IASA-TC 06, Edition 1
Version for comment, 2018

Version for comment, March 2018 D-1 Guidelines for the Preservation of Video Recordings

Table of Contents

D.1 Planning, Setup, and Workflows for Video Digitisation D-8

D.1.1 Introduction	D-8
D.1.1.1 Applicability to in-house and outsourced operations	D-8
D.1.1.2 Artisanal and high volume "factory" operations	D-8
D.1.1.3 Digitising in terms of three zones	D-8
D.1.1.4 Quality assurance, control, and critical control points	D-10
D.1.1.4.1 Quality assurance and quality control	D-10
D.1.1.4.2 Critical control points and quality control	D-11
D.1.1.5 Quality control: how much science, how much art?	D-11
D.1.1.6 Broadcast community quality control initiatives for file-based video	D-12
D.1.1.6.1 The EBU quality control activity insight into QC dimensions	D-12
D.1.1.6.2 The DPP and digital-format program interchange	D-13
D.1.2 Planning, preparing collection materials for digitisation, and related logistics	D-14
D.1.2.1 Description and background for planning, collection preparation, and related logistics	D-14
D.1.2.1.1 Project planning and the selection of materials for digitisation	D-14
D.1.2.1.2 Assessing condition and preparing source materials for digitisation	D-15
D.1.2.1.3 Barcodes and management systems	D-15
D.1.2.1.4 Shipping logistics when digitising is outsourced	D-16
D.1.2.1.5 Manifests to support collection movement	D-16
D.1.2.2 Critical control point for planning, collection preparation, and related logistics	D-16



Kieran O'Leary
kieranjol

Unfollow

Block or report user

IFI Irish Film Archive
Dublin, Ireland.
<https://twitter.com/kieranjol>

Overview Repositories 72 Projects 0 Stars 284 Followers 55 Following 90

Pinned

IFiscripts

Detailed documentation is available here:
<http://ifiscripts.readthedocs.io/en/latest/index.html>

Python ★ 24 🍴 21

ffv1-frontend

Python ★ 4

LibraryOfCongress/bagit-python

Work with Bagit packages from Python.

Python ★ 115 🍴 65

amiaopensource/ffmpegprovisr

Repository of useful FFMpeg commands for archivists!

HTML ★ 148 🍴 31

MediaArea/MediaInfo

Convenient unified display of the most relevant technical and tag data for video and audio files.

Pascal ★ 263 🍴 53

premisviewer

Transforms PREMIS XML into a human readable report via python/xml

Python ★ 2

kieranjol / IFiscripts

Watch 12 Unstar 24 Fork 21

Code Issues 55 Pull requests 5 Projects 0 Wiki Insights

Branch: master IFiscripts / makeffv1.py

Find file Copy path

kieranjol makeffv1 - adds proper RGB framemd5 pix_fmt, fixes #229

1a59e14 on 26 Dec 2018

1 contributor

Executable File 300 lines (292 sloc) 11.8 KB

Raw Blame History

```
1 #!/usr/bin/env python
2
3 # Written by Kieran O'Leary, with a major review and overhaul/cleanup by Zach Kelling aka @Zeekay
4 # Makes a single ffv1.mkv
5
6 import subprocess
7 import sys
8 import filecmp
9 import os
10 import shutil
11 import csv
12 import time
13 import itertools
14 import getpass
15 from glob import glob
16 try:
17     from ififuncs import set_environment
18     from ififuncs import hashlib_manifest
19     from ififuncs import make_mediatrace
20     from ififuncs import make_medainfo
21     from ififuncs import get_medainfo
22     from ififuncs import append_csv
23     from ififuncs import create_csv
24     from ififuncs import generate_log
25     from ififuncs import get_ffmpeg_fmt
26 except ImportError:
27     print '*** ERROR - IFIFUNCS IS MISSING - *** \n'
28     'Makeffv1 requires that ififuncs.py is located in the same directory'
29     'as some functions are located in that script -'
30     'https://github.com/kieranjol/IFiscripts/blob/master/ififuncs.py'
31     sys.exit()
32
33 def read_non_comment_lines(infile):
34     # Adapted from Andrew Dalke - http://stackoverflow.com/a/8304087/2188572
35     for lineno, line in enumerate(infile):
36         #if line[:1] != "#":
37             yield lineno, line
38
39 def get_input():
40     if len(sys.argv) < 2:
41         print 'IFI FFV1.MKV SCRIPT'
42         print 'USAGE: PYTHON makeffv1.py FILENAME'
43         print 'OR'
```





Kieran O'Leary

kieranjol

Unfollow

Block or report user

IFI Irish Film Archive

Dublin, Ireland.

<https://twitter.com/kieranjol>

Overview Repositories 72 Projects 0 Stars 284 Followers 55 Following 90

Pinned

[IFiscripts](#)

Detailed documentation is available here:
<http://ifiscripts.readthedocs.io/en/latest/index.html>

Python ★ 24 🍴 21

[ffv1-frontend](#)

Python ★ 4

[LibraryOfCongress/bagit-python](#)

Work with Bagit packages from Python.

Python ★ 115 🍴 65

[amiaopensource/ffmpegprovisr](#)

Repository of useful FFMpeg commands for archivists!

HTML ★ 148 🍴 31

[MediaArea/MediaInfo](#)

Convenient unified display of the most relevant technical and tag data for video and audio files.

Pascal ★ 263 🍴 53

[premisviewer](#)

Transforms PREMIS XML into a human readable report via python/xml

Python ★ 2



GitHub

```

274 elif len(checksum_mismatches) > 1:
275     print 'NOT LOSSLESS'
276     append_csv(
277         csv_report_filename,
278         (
279             output, 'NOT LOSSLESS',
280             source_video_size, ffv1_video_size, compression_ratio
281         )
282     )
283     generate_log(log, 'makeffv1.py Not Lossless.')
284     hashlib_manifest(filenoext, manifest, source_parent_dir)
285     if filecmp.cmp(fmd5, fmd5ffv1, shallow=False):
286         print "YOUR FILES ARE LOSSLESS YOU SHOULD BE SO HAPPY!!!"
287     else:
288         print "The framemd5 text files are not completely identical."
289         " This may be because of a lossy transcode,"
290         " or a change in metadata, most likely pixel aspect ratio."
291         " Please analyse the framemd5 files for source and output."
292
293 def main():
294     video_files, csv_report_filename = get_input()
295     make_ffv1(video_files, csv_report_filename)
296
297
298 if __name__ == "__main__":
299     main()

```



Ashley Blewer

audiovisual preservation training



Independent study paths

Computers (Understanding)

- Computers
- Fixity
- Networks
- Storage
- Databases

Computers (Talking to)

- Command Line
Interface
- (Bash) Scripting
- Git 100
- Git 101
- Metadata

Digital Preservation

- Digital Preservation
- OAIS
- Fixity
- Storage
- Digital Forensics

FFmpeg

- FFmpeg
- FFplay
- FFprobe
- FFV1
- FFmpeg + Art
- FFmpeg +
Preservation

Media Analysis

- Video
- MediaInfo
- MediaConch
- QCTools
- FFprobe
- (Other) Analysis
Tools

All of 'em

Scripting (Bash)

Command Line
Interface

Computers

Data Wrangling

Databases

Digital
Forensics

Digital
Preservation

FFmpeg + Art

Minimum* Viable Station Documentation

Link to this document: bit.ly/mindigit

Please contribute! Contact [Ashley Blewer](#) to request write-access to this document.

- [Blog post for doc context](#)
- [Minimum Viable Station as Diagram](#)
- [Minimum Viable Station Recipes!](#)

Table of Contents

[Open examples](#)

[Hardware](#)

[Computer](#)

[Analog-to-digital converter](#)

[Time Base Corrector \(TBC\)](#)

[Capture Software](#)

[Disk Imaging Software \(for Optical Media\)](#)

[Packaging/Fixity](#)

[Quality Control Software](#)

[Metadata Software](#)

[Decks](#)

[Cables](#)

[Film](#)

[Storage](#)

[Misc.](#)

[License](#)

Minimum Viable Workstation... Recipes!

Shopping lists for workstations at every budget!

Please contribute! Contact Ashley Blewer to request write-access to this document.

For more details for each component of the workflow, please see [the original Minimum Viable Station document](#).

Don't forget you will need to also budget for small incidentals like cleaning supplies, tables/storage/chairs, WD-40, headphones, and power cables.

Table of Contents

[Ain't Got No \(\\$\)](#)

[Preservation-grade Bells and Whistles \(\\$\\$\)](#)

[Saving High School Mixtapes \(\\$\)](#)

Ain't Got No (\$)

Description:

"If you build it, they will come." Sometimes you have to get started with what you have and upgrade later.

Setup:

- Your Computer (Any OS, laptop or desktop) | free or around \$300 (minimally, could easily be more)
- Instead of a computer, you could consider using a Raspberry Pi (\$30)
- VHS Deck from your Friend's Parent's Attic (Consumer-grade) | free or [around \\$30](#)
- Video8/Hi-8 Camera from your Friend's Parent's Attic (Consumer-grade) | free or [around \\$150](#)
- MiniDV Camera from your Friend's Parent's Attic (Consumer-grade) | free or [around \\$130](#)
- [ElGato Video Capture](#) (includes hardware, cable, and software) | \$75

Total (with found/borrowed equipment): \$75

Total (with dedicated equipment): around \$685

Preservation-grade Bells and Whistles (\$\$)

Digital Tape Preservation Strategy: Preserving Data Or Video

By Kara Van Malssen | December 2, 2009 | Papers and Presentations

By David Rice and Chris Lacinak – December 2, 2009

Abstract: This paper examines preservation philosophies and strategies applied to large scale video collections that are both *born-digital* and *tape-based*. Technically and philosophically different approaches may be applied to migrating born-digital, tape-based content with decisions ranging from deck selection and choice of output to specifications of the resulting file. At the core of this is the distinction between migrating digital video as an audiovisual signal versus migrating it as data.

Introduction

In trying to conceptualize the issues around the migration of born-digital tape-based content we're challenged to separate our normal associations with videotape and the video/audio signals from the fact that the content is stored digitally on the tape. It looks and acts like a legacy analog videocassette in many ways, but some of the underlying technology is different. This is why we may discuss the "migration", and not "digitization", of the content to the file-based domain. The content in this specific collection is already digital, born in a compressed DV codec.

Profile of DV tapes (miniDV and DVCam)

Composition: DVCam and miniDV tapes are a metal-evaporated tape made with a prioritization to compact size and low cost. The lubricated tape is extremely thin, chemically complex, and susceptible to drop-outs, errors, and data loss. The tape is also finicky as damaged or deteriorated portions of the tape may play back with varying degrees of accuracy from one playback to another or from one deck to another through error correction and concealment strategies.

The Case of DV tape

With DV tape we are once again faced with a physical format that is dependent on certain hardware for playback. However, DV is a published standard that is well documented and is supported by both major hardware and software manufacturers.

Still the conflation of a digital signal and a tape-based video carrier brings about confusion on how to approach migration. The hardware for playback offers multiple types of signal outputs. The selection of output identifies a key question. Should the information on the tape be treated purely as audiovisual information similar to legacy video formats or should it be treated as stream of data? The fact is that DV tape contains a much greater amount of information beyond the audiovisual signal.

A DV stream may contain:

- Video (NTSC or PAL)
- Audio (48, 44.1, or 33 kHz; 2 or 4 track)
- Metadata from the Source Tape
 - TimeCode
 - Closed Captioning (as auxiliary data, not video line 21)
 - Camera Metadata (iris, gain, white balance, etc.)
 - Original Recording Date and Time
- Metadata from Device Read (Occurrences during playback)
- "Status" data ([video error concealment](#), [audio errors](#), etc.)

Selecting the firewire output of a DV deck treats the DV tape as data and migrates all of the above data in its native format. Selecting the Serial Digital Interface (SDI) output from a DV deck treats the DV tape like a legacy videotape discarding a great deal of metadata, decompressing the video and blocking information on whether or not the deck used error concealment or produced a drop-out.

You may be asking, why does this other metadata matter?

1. **Authenticity:** DV attaches metadata to every recorded frame that can tie that frame to its place in the production chain. Inconsistencies in the timestamp, timecode, and camera information that occur during filming or editing can be identified and tracked.
2. **Error identification:** As addressed above, errors or concealments that occur during playback can be permanently encoded into the migrated datastream. A bit-for-bit transfer captures metadata that describes the error-related processes within the deck. This data can be analyzed for causes and possible solutions to determine if a second transfer is warranted or to identify potential issues in the workflow. Further information about different kinds of errors can be found under the *DV Analyzer: Case Studies* heading [here](#).
3. **Efficiency:** Because so much information about the digital transfer itself is encoded in the stream, that data can be leveraged to perform targeted, efficient analysis and quality control with tools such as Live Capture Plus and DV Analyzer. During a high-throughput migration of content, quality control must be selective and focused. Putting on headphones and viewing every sample may not be practical or affordable. Tools that solely analyze the decoded audiovisual playback may play a useful role but may also under- or over-report on errors through estimations rather than responding to quality control information that is explicitly documented in the DV stream output of a FireWire cable. If the tools managing the migration of the content can provide information about the work being done, especially the parts of the migration pertinent to errors, we can narrow our quality control to areas where the errors are known.

Products



Aviary

An audiovisual content publishing platform, Aviary provides controlled-access to intended users with search, navigation, and playback across audio and video archives, collections, and files, pinpointing search results with playback exactly where a search term is found.

[LEARN MORE](#)

Catalyst Inventory Solution

Catalyst is a solution for creating better inventories and managing digitization workflows. It transforms unknowns into quantitative data for better decision making, more effective resource allocation, item access and management, and planning and strategizing.

[LEARN MORE](#)

AVCC

AVCC makes collaborative, efficient item-level cataloging of audiovisual collections possible over the web. Built-in reporting on critical data enables users to uncover hidden collections, support preservation reformatting, and expand access to content.

[LEARN MORE](#)

Exactly

Exactly is an easy-to-use and in-demand application for securely and remotely sending and receiving digital data. Exactly allows recipients to create customized metadata templates for senders, incorporates data integrity measures, and notifies users when data reaches its destination.

[LEARN MORE](#)

Fixity

Fixity is a simple, powerful and popular application for automated monitoring and reporting on the data integrity of stored files. Fixity provides data for comparative analyses and sends status updated and reports automatically.

[LEARN MORE](#)

MDQC

MDQC stands for Metadata Quality Control, and it allows users to create rules on embedded metadata, scan a set of files, and report on the conformance of each file against the user-defined rules. MDQC greatly minimizes the time needed to QC large batches of files.

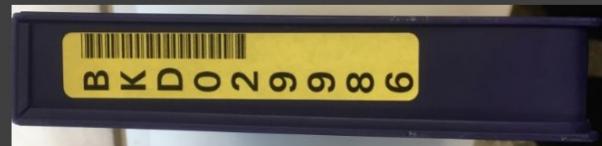
[LEARN MORE](#)

BBC Archives

Securing the past for the future

Steve Daly
Head of Technology
BBC Information and Archives

1: FILENAMES



BKD029986_79054025255fb1a26e4bc422aef54eb4.mxf

2: MODULARITY



1. community
2. open source tools + resources



**A complete, cross-platform
solution to record, convert and
stream audio and video.**



Download

Converting video and audio has never been so easy.

```
$ ffmpeg -i input.mp4 output.avi
```

<https://ffmpeg.org/>

Discover more

```
stephen@BK-CI-WEB:~$ tldr ffmpeg
# ffmpeg
```

Video conversion tool.

- Extract the sound from a video and save it as MP3:

```
ffmpeg -i video.mp4 -vn sound.mp3
```

- Convert frames from a video or GIF into individual numbered images:

```
ffmpeg -i video.mpg|video.gif frame_%d.png
```

- Combine numbered images (frame_1.jpg, frame_2.jpg, etc) into a video or GIF:

```
ffmpeg -i frame_%d.jpg -f image2 video.mpg|video.gif
```

- Quickly extract a single frame from a video at time mm:ss and save it as a 128x128 resolution image:

```
ffmpeg -ss mm:ss -i video.mp4 -frames 1 -s 128x128 -f image2 image.png
```

- Trim a video from a given start time mm:ss to an end time mm2:ss2 (omit the -to flag to trim till the end):

```
ffmpeg -ss mm:ss -to mm2:ss2 -i video.mp4 -codec copy output.mp4
```

- Convert AVI video to MP4. AAC Audio @ 128kbit, h264 Video @ CRF 23:

```
ffmpeg -i input_video.avi -codec:a:aac -b:a:128k -codec:v:libx264 -crf 23 output_video.mp4
```

- Remux MKV video to MP4 without re-encoding audio or video streams:

```
ffmpeg -i input_video.mkv -codec copy output_video.mp4
```

- Convert MP4 video to VP9 codec. For the best quality, use a CRF value (recommended range 15-35) and -b:video MUST be 0:

```
ffmpeg -i input_video.mp4 -codec:v:libvpx-vp9 -crf 30 -b:video 0 -codec:a:libopus -vbr on -threads number_of_threads output_video.webm
```

tldr ffmpeg

NAME

ffmpeg - ffmpeg video converter

SYNOPSIS

ffmpeg [global options] [{input file options} -i input url] ... [{output file options} output url] ...

DESCRIPTION

ffmpeg is a very fast video and audio converter that can also grab from a live audio/video source. It can also convert between arbitrary sample rates and resize video on the fly with a high quality polyphase filter.

ffmpeg reads from an arbitrary number of input "files" (which can be regular files, pipes, network streams, grabbing devices, etc.), specified by the "-i" option, and writes to an arbitrary number of output "files", which are specified by a plain output url. Anything found on the command line which cannot be interpreted as an option is considered to be an output url.

Each input or output url can, in principle, contain any number of streams of different types (video/audio/subtitle/attachment/data). The allowed number and/or types of streams may be limited by the container format. Selecting which streams from which inputs will go into which output is either done automatically or with the "-map" option (see the Stream selection chapter).

To refer to input files in options, you must use their indices (0-based). E.g. the first input file is 0, the second is 1, etc. Similarly, streams within a file are referred to by their indices. E.g. "2:3" refers to the fourth stream in the third input file. Also see the Stream specifiers chapter.

As a general rule, options are applied to the next specified file. Therefore, order is important, and you can have the same option on the command line multiple times. Each occurrence is then applied to the next input or output file. Exceptions from this rule are the global options (e.g. verbosity level), which should be specified first.

Do not mix input and output files -- first specify all input files, then all output files. Also do not mix options which belong to different files. All options apply ONLY to the next input or output file and are reset between files.

- To set the video bitrate of the output file to 64 kbit/s:

```
ffmpeg -i input.avi -b:v 64k -bufsize 64k output.avi
```

- To force the frame rate of the output file to 24 fps:

```
ffmpeg -i input.avi -r 24 output.avi
```

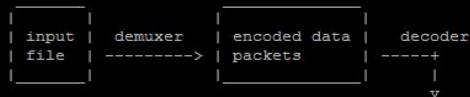
- To force the frame rate of the input file (valid for raw formats only) to 1 fps and the frame rate of the output file to 24 fps:

```
ffmpeg -r 1 -i input.m2v -r 24 output.avi
```

The format option may be needed for raw input files.

DETAILED DESCRIPTION

The transcoding process in ffmpeg for each output can be described by the following diagram:



ffmpeg man

FFmpeg FAQ

Table of Contents

- 1 General Questions
 - 1.1 Why doesn't FFmpeg support feature [xyz]?
 - 1.2 FFmpeg does not support codec XXX. Can you include a Windows DLL loader to support it?
 - 1.3 I cannot read this file although this format seems to be supported by ffmpeg.
 - 1.4 Which codecs are supported by Windows?
- 2 Compilation
 - 2.1 error: can't find a register in class 'GENERAL_REGS' while reloading 'asm'
 - 2.2 I have installed this library with my distro's package manager. Why does configure not see it?
 - 2.3 How do I make pkg-config find my libraries?
 - 2.4 How do I use pkg-config when cross-compiling?
- 3 Usage
 - 3.1 ffmpeg does not work; what is wrong?
 - 3.2 How do I encode single pictures into movies?
 - 3.3 How do I encode movie to single pictures?
 - 3.4 Why do I see a slight quality degradation with multithreaded MPEG* encoding?
 - 3.5 How can I read from the standard input or write to the standard output?
 - 3.6 -f jpeg doesn't work.
 - 3.7 Why can I not change the frame rate?
 - 3.8 How do I encode Xvid or DivX video with ffmpeg?
 - 3.9 Which are good parameters for encoding high quality MPEG-4?
 - 3.10 Which are good parameters for encoding high quality MPEG-1/MP EG-2?
 - 3.11 Interlaced video looks very bad when encoded with ffmpeg, what is wrong?
 - 3.12 How can I read DirectShow files?
 - 3.13 How can I join video files?
 - 3.14 How can I concatenate video files?
 - 3.14.1 Concatenating using the concat *filter*
 - 3.14.2 Concatenating using the concat *demuxer*
 - 3.14.3 Concatenating using the concat *protocol* (file level)
 - 3.14.4 Concatenating using raw audio and video
 - 3.15 Using -f lavfi, audio becomes mono for no apparent reason.
 - 3.16 Why does FFmpeg not see the subtitles in my VOB file?
 - 3.17 Why was the `ffmpeg -sameq` option removed? What to use instead?
 - 3.18 I have a stretched video, why does scaling does not fix it?
 - 3.19 How do I run ffmpeg as a background task?
 - 3.20 How do I prevent ffmpeg from suspending with a message like *suspended (tty output?)*
- 4 Development
 - 4.1 Are there examples illustrating how to use the FFmpeg libraries, particularly libavcodec and libavformat?
 - 4.2 Can you support my C compiler XXX?
 - 4.3 Is Microsoft Visual C++ supported?
 - 4.4 Can you add automake, libtool or autoconf support?
 - 4.5 Why not rewrite FFmpeg in object-oriented C++?
 - 4.6 Why are the ffmpeg programs devoid of debugging symbols?
 - 4.7 I do not like the LGPL, can I contribute code under the GPL instead?
 - 4.8 I'm using FFmpeg from within my C application but the linker complains about missing symbols from the libraries themselves.
 - 4.9 I'm using FFmpeg from within my C++ application but the linker complains about missing symbols which seem to be available.
 - 4.10 I'm using libavutil from within my C++ application but the compiler complains about 'UINT64_C' was not declared in this scope
 - 4.11 I have a file in memory / a API different from *open/*read/*libc how do I use it with libavformat?
 - 4.12 Where is the documentation about ffv1, msmpeg4, asv1, 4xm?
 - 4.13 How do I feed H.263-RTP (and other codecs in RTP) to libavcodec?
 - 4.14 AVStream.r_frame_rate is wrong. It is much larger than the frame rate.
 - 4.15 Why is make_fate not running all tests?
 - 4.16 Why is make_fate not finding the samples?

ffmpeg Documentation

Table of Contents

- 1 Synopsis
- 2 Description
- 3 Detailed description
 - 3.1 Filtering
 - 3.1.1 Simple filtergraphs
 - 3.1.2 Complex filtergraphs
 - 3.2 Stream copy
- 4 Stream selection
 - 4.1 Description
 - 4.1.1 Automatic stream selection
 - 4.1.2 Manual stream selection
 - 4.1.3 Complex filtergraphs
 - 4.1.4 Stream handling
 - 4.2 Examples
- 5 Options
 - 5.1 Stream specifiers
 - 5.2 Generic options
 - 5.3 AVOptions
 - 5.4 Main options
 - 5.5 Video Options
 - 5.6 Advanced Video options
 - 5.7 Audio Options
 - 5.8 Advanced Audio options
 - 5.9 Subtitle options
 - 5.10 Advanced Subtitle options
 - 5.11 Advanced options
 - 5.12 Preset files
 - 5.12.1 ffpreset files
 - 5.12.2 avpreset files
- 6 Examples
 - 6.1 Video and Audio grabbing
 - 6.2 X11 grabbing
 - 6.3 Video and Audio file format conversion
- 7 See Also
- 8 Authors

1 Synopsis

`ffmpeg [global_options] [{input_file_options} -i input_url] ... [{output_file_options} output_url] ...`

2 Description

`ffmpeg` is a very fast video and audio converter that can also grab from a live audio/video source. It can also convert between arbitrary sample rates and resize video on the fly with a high quality polyphase filter.

`ffmpeg` reads from an arbitrary number of input "files" (which can be regular files, pipes, network streams, grabbing devices, etc.), specified by the `-i` option, and writes to an arbitrary number of output "files", which are specified by a plain output url. Anything found on the command line which cannot be interpreted as an option is considered to be an output url.

Each input or output url can, in principle, contain any number of streams of different types (video/audio/subtitle/attachment/data). The allowed number and/or types of streams may be limited by the container format. Selecting which streams from which inputs will go into which output is either done automatically or with the `-map` option (see the Stream selection chapter).



making ffmpeg easier

FFmpeg is a powerful tool for manipulating audiovisual files.

Unfortunately, it also has a steep learning curve, especially for users unfamiliar with a command line interface.

This app helps users through the command generation process so that more people can reap the benefits of FFmpeg.

Table of Contents

- About this resource
- Show/hide all recipes
- FFmpeg basics
- Advanced FFmpeg concepts
- Change container (rewrap)
- Change codec (transcode)
- Change video properties
- Change/view audio properties
- Join/trim/create an excerpt
- Work with interlaced video
- Overlay timecode or text on a video
- Create thumbnails or GIFs
- Create a video from image(s) and audio
- Use filters or scopes
- View or strip metadata
- Preservation tasks
- Generate test files
- Use OCR
- Compare similarity of videos
- Something else
- Similar tools: tips & tricks
- CDDA (Audio CD) Ripping Tools
- ImageMagick
- flac tool

About ffmprovisr

Making FFmpeg Easier

FFmpeg is a powerful tool for manipulating audiovisual files. Unfortunately, it also has a steep learning curve, especially for users unfamiliar with a command line interface. This app helps users through the command generation process so that more people can reap the benefits of FFmpeg.

Each button displays helpful information about how to perform a wide variety of tasks using FFmpeg. To use this site, click on the task you would like to perform. A new window will open up with a sample command and a description of how that command works. You can copy this command and understand how the command works with a breakdown of each of the flags.

This page does not have search functionality, but you can open all recipes (second option in the sidebar) and use your browser's search tool (often ctrl+f or cmd+f) to perform a keyword search through all recipes.

Tutorials

For FFmpeg basics, check out the program's [official website](#).

For instructions on how to install FFmpeg on Mac, Linux, and Windows, refer to Reto Kromer's [installation instructions](#).

For Bash and command line basics, try the [Command Line Crash Course](#). For a little more context presented in an ffmprovisr style, try [explainshell.com!](#)

License



This work is licensed under a [Creative Commons Attribution 4.0 International License](#).

Sister projects

[Script Ahoy](#): Community Resource for Archivists and Librarians Scripting

[The Sourcecaster](#): an app that helps you use the command line to work through common challenges that come up when working with digital primary sources.

[Cable Bible](#): A Guide to Cables and Connectors Used for Audiovisual Tech

Learn about FFmpeg basics

🔗 Basic structure of an FFmpeg command

🔗 Streaming vs. Saving

Learn about more advanced FFmpeg concepts

🔗 Codec defaults

🔗 Filtergraphs

🔗 Stream mapping

Change codec (transcode)

 Transcode to deinterlaced Apple ProRes LT


 Transcode to an H.264 access file

 Transcode from DCP to an H.264 access file

 Transcode your file with the FFV1 Version 3 Codec in a Matroska container

 Convert DVD to H.264

 Transcode to an H.265/HEVC MP4

 Transcode to an Ogg Theora

 Convert WAV to MP3

 Generate two access MP3s (with and without copyright)

 Convert WAV to AAC/MP4

Change video properties

- 🔧 Transform 4:3 aspect ratio into 16:9 with pillarbox
- 🔧 Transform 16:9 aspect ratio video into 4:3 with letterbox
- 🔧 Flip video image
- 🔧 Transform SD to HD with pillarbox
- 🔧 Change display aspect ratio without re-encoding
- 🔧 Convert colorspace of video
- 🔧 Modify image and sound speed
- 🔧 Synchronize video and audio streams
- 🔧 Clarify stream properties
- 🔧 Crop video
- 🔧 Change video color to black and white

Change or view audio properties

- 🔧 Extract audio without loss from an AV file
- 🔧 Combine audio tracks
- 🔧 Inverses the audio phase of the second channel
- 🔧 Calculate Loudness Levels
- 🔧 RIAA Equalization
- 🔧 Reverse CD Pre-Emphasis
- 🔧 One Pass Loudness Normalization
- 🔧 Two Pass Loudness Normalization
- 🔧 Fix A/V sync issues by resampling audio

Work with interlaced video

- 🔧 Upscaled, pillar-boxed HD H.264 access files from SD NTSC source
- 🔧 Deinterlace video
- 🔧 Inverse telecine
- 🔧 Set field order for interlaced video
- 🔧 Identify interlacement patterns in a video file












Overlay timecode or text

- 🔧 Create opaque centered text watermark
- 🔧 Overlay image watermark on video
- 🔧 Burn in timecode
- 🔧 Embed subtitles

Create thumbnails or GIFs

- 🔧 Export one thumbnail per video file
- 🔧 Export many thumbnails per video file
- 🔧 Create GIF from still images
- 🔧 Create GIF from a video

Preservation tasks

-  Batch processing (Mac/Linux)
-  Batch processing (Windows)
-  Check decoder errors
-  Check FFV1 fixity
-  Create MD5 checksums (video frames)
-  Create MD5 checksums (audio samples)
-  Create MD5 checksum(s) for A/V stream data only
-  Get checksum for video/audio stream
-  QCTools report (with audio)
-  QCTools report (no audio)
-  Read/Extract EIA-608 Closed Captioning

Preservation tasks

 Batch processing (Mac/Linux)

 Batch processing (Windows)

Create PowerShell script to batch process with FFmpeg

As of Windows 10, it is possible to run Bash via [Bash on Ubuntu on Windows](#), allowing you to use [bash scripting](#). To enable Bash on Windows, see [these instructions](#).

On Windows, the primary native command line program is **PowerShell**. PowerShell scripts are plain text files saved with a .ps1 extension. This entry explains how they work with the example of a PowerShell script named "rewrap-mp4.ps1", which rewraps .mp4 files in a given directory to .mkv files.

"rewrap-mp4.ps1" contains the following text:

```
$inputfiles = ls *.mp4
foreach ($file in $inputfiles) {
    $output = [io.path]::ChangeExtension($file, '.mkv')
    ffmpeg -i $file -map 0 -c copy $output
}
```

`$inputfiles = ls *.mp4`

Creates the variable `$inputfiles`, which is a list of all the .mp4 files in the current folder.

In PowerShell, all variable names start with the dollar-sign character.

`foreach ($file in $inputfiles)`

Creates a loop and states the subsequent code block will be applied to each file listed in `$inputfiles`.

`$file` is an arbitrary variable which will represent each .mp4 file in turn as it is looped over.

{

Opens the code block.

`$output = [io.path]::ChangeExtension($file, '.mkv')`

Sets up the output file: it will be located in the current folder and keep the same filename, but will have an .mkv extension instead of .mp4.

`ffmpeg -i $file`

Carry out the following FFmpeg command for each input file.

Note: To call FFmpeg here as just "ffmpeg" (rather than entering the full path to ffmpeg.exe), you must make sure that it's correctly configured. See [this article](#), especially the section 'Add to Path'.

`-map 0`

retain all streams

`-c copy`

enable stream copy (no re-encode)

`$output`

The output file is set to the value of the `$output` variable declared above: i.e., the current file name with an .mkv extension.

}

Closes the code block.

Note: the PowerShell script (.ps1 file) and all .mp4 files to be rewrapped must be contained within the same directory, and the script must be run from that directory.

Execute the .ps1 file by typing `.\rewrap-mp4.ps1` in PowerShell.

Modify the script as needed to perform different transcodes, or to use with ffprobe. :)

Link to this command: https://amaiaopensource.github.io/ffmpegprovisr/index.html#batch_processing_win

ffprobe Documentation

Table of Contents

- 1 Synopsis
- 2 Description
- 3 Options
 - 3.1 Stream specifiers
 - 3.2 Generic options
 - 3.3 AVOptions
 - 3.4 Main options
- 4 Writers
 - 4.1 default
 - 4.2 compact, csv
 - 4.3 flat
 - 4.4 ini
 - 4.5 json
 - 4.6 xml
- 5 Timecode
- 6 See Also
- 7 Authors

1 Synopsis

`ffprobe [options] [input_url]`

2 Description

`ffprobe` gathers information from multimedia streams and prints it in human- and machine-readable fashion.

For example it can be used to check the format of the container used by a multimedia stream and the format and type of each media stream contained in it.

If a url is specified in input, `ffprobe` will try to open and probe the url content. If the url cannot be opened or recognized as a multimedia file, a positive exit code is returned.

`ffprobe` may be employed both as a standalone application or in combination with a textual filter, which may perform more sophisticated processing, e.g. statistical processing or plotting.

Options are used to list some of the formats supported by `ffprobe` or for specifying which information to display, and for setting how `ffprobe` will show it.

`ffprobe` output is designed to be easily parsable by a textual filter, and consists of one or more sections of a form defined by the selected writer, which is specified by the `print_format` option.

Sections may contain other nested sections, and are identified by a name (which may be shared by other sections), and an unique name. See the output of sections.

Metadata tags stored in the container or in the streams are recognized and printed in the corresponding "FORMAT", "STREAM" or "PROGRAM_STREAM" section.

3 Options

All the numerical options, if not specified otherwise, accept a string representing a number as input, which may be followed by one of the SI unit prefixes, for example: 'K', 'M', or 'G'.

If 'i' is appended to the SI unit prefix, the complete prefix will be interpreted as a unit prefix for binary multiples, which are based on powers of 1024 instead of powers of 1000. Appending 'B' to the SI unit prefix multiplies the value by 8. This allows using, for example: 'KB', 'MiB', 'G' and 'B' as number suffixes.

Options which do not take arguments are boolean options, and set the corresponding value to true. They can be set to false by prefixing the option name with "no". For example using "nofoo" will set the boolean option with name "foo" to false.

```
{
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from
'N-502753_01of01_DVS_1_20150819.195200_to_20150819.210759.program8500.
mp4':
  Metadata:
    major_brand      : mp42
    minor_version    : 0
    compatible_brands: isomiso2mp41
    creation_time    : 2015-09-02T11:40:58.000000Z
    encoder          : FFmpeg 0.7

  Duration: 01:01:43.16, start: 0.000000, bitrate: 1021 kb/s
    Stream #0:0(und): Video: mpeg4 (Simple Profile) (mp4v /
0x7634706D), yuv420p, 512x288 [SAR 1:1 DAR 16:9], 900 kb/s, 25 fps, 25
tbr, 25 tbn, 25 tbc (default)
      Metadata:
        creation_time    : 2015-09-02T11:40:58.000000Z
        handler_name     : VideoHandler
        Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 48000
Hz, stereo, fltp, 116 kb/s (default)
      Metadata:
        creation_time    : 2015-09-02T11:40:58.000000Z
        handler_name     : SoundHandler
    "streams": [
    {
      "index": 0,
      "codec_name": "mpeg4",
      "codec_long_name": "MPEG-4 part 2",
      "profile": "Simple Profile",
      "codec_type": "video",
      "codec_time_base": "1/25",
      "codec_tag_string": "mp4v",
      "codec_tag": "0x7634706d",
      "width": 512,
      "height": 288,
      "coded_width": 512,
      "coded_height": 288,
      "has_b_frames": 0,
      "sample_aspect_ratio": "1:1",
      "display_aspect_ratio": "16:9",
```



FrameMD5

 Search[Login](#) | [Preferences](#) | [Help/Guide](#) | [About Trac](#) | [Register](#)[Wiki](#)[Timeline](#)[View Tickets](#)[Search](#)[Tags](#)[wiki: framemd5 Intro and HowTo](#)[+0](#)[Start Page](#)[Index](#)[History](#)

Introduction

Within digital preservation environments, the generation and verification of checksums against digital files can aid the confirmation or denial of digital authenticity over time. A checksum mismatch is an alert that a file under care has changed from a prior state; potentially triggering retrieval of backups, review of hardware, or migration of content. Generally, if a given checksum algorithm is applied to a file, then as long as the same checksum can be regenerated from the file then the data is verified, else a mismatched checksum reveals a digital change. Further details such as the whereabouts, extent, or significance of the change in data are not revealed by the checksum mismatch but only that the data examined now is not the same as the data examined before.

The FFmpeg [⇒ framemd5](#) format and [⇒ framecrc](#) format as used to decode input audiovisual data to produce one checksum per frame. These formats facilitate testing functions such as verifying that an adjusted decoder maintains intended results or that an FFmpeg decoder decodes a stream to the same data as another decoder.

By producing checksums on a more granular level, such as per frame, it is more feasible to assess the extent or location of digital change in the event of a checksum mismatch. By decoding a file and processing the decoded data to generate a framemd5 document, each decoded audio and video frame is documented according to its timestamp, digital size, and MD5 checksum. For the first three frames of video, the framemd5 output could be:

```
#tb 0: 1001/24000
0, 0, 0, 1, 518400, 5bc19af1a75adb8bda9d79390981a0ea
0, 1, 1, 1, 518400, bb485b0d6fd001358aa7dbe76031ff4d
0, 2, 2, 1, 518400, 30dc414cd4487dd58b0d16a5ddaafba35
```

In this output the columns refer to the stream number, counting from zero, (column 1), the decoding and presentation timestamps (column 2 and 3), the samples duration (column 4), the size of the data checksummed in bytes14 (column 5), and the MD5 checksum for that data.

Storing a framemd5 file along with each audiovisual file does not replace the function of a traditional whole-file checksum. It is still possible for a file to be changed in a way that would result in a mismatch for a future whole-file checksum analysis, but not create any difference between a stored framemd5 output and a newly created framemd5 output. This could occur when embedded metadata is edited but the stored audiovisual data remains the same.

For audiovisual data, storing both a whole-file checksum and a framemd5 output enables greater awareness of digital change in managed files, a more strategic and aware response to change, and the ability to verify lossless transcoding. If an audiovisual file is found to have a mismatch between a newly generated whole-file checksum and one generated previously, indicative of digital change, then comparison between a stored framemd5 document and a newly generated one could facilitate in pinpointing the digital change as it affects audiovisual presentation if at all.

How to Create a framemd5

A framemd5 report can be generated with this command:

```
ffmpeg -i MOVIE.mov -f framemd5 MOVIE.framemd5
```

For this example the output is:

```
#tb 0: 1001/30000
0, 0, 0, 1, 1669440, 1fb241f71b9b14abdf88ad5034b6dc21
0, 1, 1, 1, 1669440, 38310375ae195c17019e26da9d99e3d0
0, 2, 2, 1, 1669440, c154e232f7f5cb74a60afc06e11cabae
0, 3, 3, 1, 1669440, 508b0d017ffa6f4694541762ed5fae6a
0, 4, 4, 1, 1669440, 36f5da2bceef097350585e91f748d1a
0, 5, 5, 1, 1669440, d36fd15efdf503c1ef25640d890917b3
0, 6, 6, 1, 1669440, 31b7232bf8e6fd2337e2beddc480dc42
0, 7, 7, 1, 1669440, 7ab5486e5999d86dd016ae0b8df13a70
0, 8, 8, 1, 1669440, 47b2a83dd6801d2c2bd414f57af8eff5
0, 9, 9, 1, 1669440, b883d73e78c230b220f311e8fb34e6ee
0, 10, 10, 1, 1669440, 4171860688591526ad3c9c3780eb044f
0, 11, 11, 1, 1669440, ad8df2d43442eb45155300965e4f59d0
0, 12, 12, 1, 1669440, 9bf60490424ebc2b5209d5d2ba3398d9
0, 13, 13, 1, 1669440, 7184bf36a237199e68afe9b51ef232e5e
0, 14, 14, 1, 1669440, 905b35a7638b53566cd5235d1edfcd0
0, 15, 15, 1, 1669440, e0f3577df7cbe6420d712be67abc1733
0, 16, 16, 1, 1669440, d20aa192b1a8da3ffa26d16464ef4ef5
0, 17, 17, 1, 1669440, 84bf9143b1ed133fa60dd04fcdcf6d2e
0, 18, 18, 1, 1669440, 118784efb0da45b148d763857a616ec6
0, 19, 19, 1, 1669440, d86e92e1046c5190b582fc527c36c69
0, 20, 20, 1, 1669440, cd37e29476412d8ff2a7effdbb538d60
0, 21, 21, 1, 1669440, 78fda53e3b2e88029fc42b347c4045fc
0, 22, 22, 1, 1669440, 3f4718d7d93899497c314a7b65ec2f95
0, 23, 23, 1, 1669440, 3650ecff2013c0bac2d8a9006972f842
0, 24, 24, 1, 1669440, de9b78e46be1ed555dfbd16d73773dd4
0, 25, 25, 1, 1669440, 3ab9ab618d930b79e9f2396d95de5ca9
0, 26, 26, 1, 1669440, e40524fab40c44811a8d21b641b4af16
0, 27, 27, 1, 1669440, d44cc0cfea82fb7b14a9b62c713c9500
0, 28, 28, 1, 1669440, 29f6ca7e17a378f939a4b4153bb258de
0, 29, 29, 1, 1669440, 7db13c711801b7a90b17e3a891035088
```

This output reflects the default handling of framemd5 where each frame is decoded (to rawvideo for video or pcm_s16le for audio) and then the checksum is generated from that decoded data.

Reconsidering the Checksum for Audiovisual Preservation

This article was authored by Dave Rice and initially published in the [IASA Journal number 29](#) under a [CC-BY-ND](#) license.

Reconsidering the Checksum for Audiovisual Preservation:

Detecting digital change in audiovisual data with decoders and checksums

Firstly, what are checksums for?

A checksum is small data value computed from a given amount of data, such as a file or bit-stream, for the purpose of facilitating the future ability to detect changes in that given data. The generation and verification of checksums for digital archival holdings is a central principle of digital preservation and enable archivists to trust that data held within an archive is the same data that was received by the archive. Although checksum wrangling is typically a behind-the-scenes process within digital storage systems and repositories, these values are worth a closer look. The checksum value is generally expressed in hexadecimal representation (aka base 16) comprised of the numbers 0 through 9 and the letters A through F. Several checksums algorithms, such as CRC32, MD5, and SHA-1, have been introduced offering varying degrees of processing efficiency, security, and collision resistance¹. As an example, the CRC32 checksum value for a text file that contains the ASCII characters “checksum” would be de6fd9a and the MD5 value for the same file would be 226190d94b21d1b0c7b1a42d855e419d. If the text file changed, whether through manipulation, bit rot, or data corruption, then further evaluations of the file would produce a different checksum value. The mismatch of a newly calculated checksum and a stored checksum produced earlier is an alert that data under care has been changed.

Verifying Lossless Transcoding

The use of lossless codecs alone does not guarantee that the resulting encoded lossless audiovisual file could be used to reconstruct the original audiovisual data. A preservation-suitable lossless audiovisual encoding should decode to the same data that the original source would decode to, meaning that each pixel, frame, and timing decoded from the lossless version should be the same as the decoded original.

An original uncompressed digital audiovisual file called “uncompressed.mov” could produce this framemd5 output (the first four video frames are listed):

```
0, 0, 0, 1, 518400, 6a6b8be7dbac428b86669992cc740d10
0, 1, 1, 1, 518400, 6a6b8be7dbac428b86669992cc740d10
0, 2, 2, 1, 518400, 2eb3a9a53f42d5c6b21e3b6b7e267414
0, 3, 3, 1, 518400, e67d03e253f3997ae2db46aa28d8c749
```

Let’s imagine that an archive decides to transcode the uncompressed file to a lossless codec in order to reduce storage requirements. This ffmpeg command generates a lossless ffv1 encoding from the original file and copies the audio data as-is.

```
ffmpeg -i uncompressed.mov -map 0 -c:v ffv1 -c:a copy lossless.mov
```

Generating a framemd5 report on the “lossless.mov” should produce the same output because both files, although utilizing different codecs, both decode to identical audiovisual presentations. If the two files do not decode to produce identical framemd5 documents then it is likely that the transcoding from the uncompressed codec to the lossless codec was not truly mathematically lossless¹⁵.

[MediaArea ▾](#)[MediaInfo ▾](#)[Projects ▾](#)[Support Us! ▾](#)[Log in ▾](#)

MediaInfo

Convenient unified display of the most relevant technical and tag data for video and audio files.

Designed, led and fully developed by MediaArea.

[Learn more](#)

MediaConch

Implementation checker, policy checker, & reporter.

Funded by the [PREFORMA](#) project co-funded by the [European Commission](#); designed, led and fully developed by MediaArea.

[Learn more](#)

MediaTrace

Technical report that expresses the binary architecture of a file.

Designed in collaboration by MediaArea and [MoMA](#); led and developed by MediaArea.

[Learn more](#)

QCTools

Helps users analyze and understand their digitized video files through use of audiovisual analytics and filtering.

Funded by the [National Endowment for the Humanities](#) and the [Knight Foundation](#); designed and led by Dave Rice and the [Bay Area Video Coalition](#); developed by MediaArea, [Fabio Utzig](#), [Alexander Ivash](#).

[Learn more](#)

BWF MetaEdit

Supports embedding, validating, and exporting of metadata in Broadcast WAVE Format (BWF) files. It supports the [FADGI \(Federal Agencies Digitization Guidelines Initiative\)](#) for embedded metadata in the text and INFO chunks.

Initially funded by the [Library of Congress](#) and [FADGI](#); designed and led by [AVPreserve](#), developed by MediaArea.

[Learn more](#)

AVI MetaEdit

Supports embedding, validating, and exporting of metadata in AVI (Standard and OpenDML) files. This tool can also enforce file structure and metadata recommendations and specifications from U.S. National Archives, Microsoft, and IBM.

Initially funded by [NARA \(National Archives and Records Administration\)](#) supported by the [FADGI](#); designed and led by [AVPreserve](#); developed by MediaArea.

[Learn more](#)

MOV MetaEdit

Supports the embedding and editing of metadata in MOV (Apple QuickTime) or MP4 (ISO/IEC 14496-14 a.k.a. MPEG-4 Part 14) files. It is currently focused on Universal Ad ID metadata but could be expanded on request.

Sponsored by [Ad-ID](#); designed, led and fully developed by MediaArea.

[Learn more](#)

DV Analyzer

Technical quality control and reporting tool that examines DV streams in order to report errors in the tape-to-file transfer process, such as video error concealment information, invalid audio samples, timecode inconsistency, inconsistent use of arbitrary bits in video DIF blocks, and DIF structural problems.

Initially funded, designed and led by [AVPreserve](#); developed by MediaArea.

[Learn more](#)

MediaArea

MediaArea is an open source software company focused on digital media analysis. We develop and support tools that aid developers in integrating digital media file investigation into their work.

We do mostly open source code, without rejecting proprietary code on request.

[MediaInfo](#) is our flagship product with 4 000 downloads per day, we also currently work on [MediaConch](#), and we worked in the past on [QCTools](#), [BWF MetaEdit](#) and some other [projects](#).

The team

Each member of the team is unique and commits code under their own name. Their work is visible via their respective accounts listed below.

Jérôme Martinez (Digital Media Analysis Specialist)

Technical design, implementation of the bytestream/bitstream analyzer, extraction of metadata



Dave Rice (Archivist)

Communication with memory institutions, definition of tests, documentation



Guillaume Roques (Senior Developer)

Database management, automation, performance optimization, shell, website maintenance



Ashley Blewer (Archivist)

Research and development, community engagement, and design



Maxime Gervais (Junior Developer)

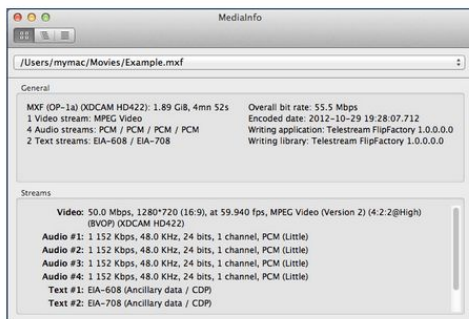
Release manager, scripting, bug fixes



Peter Bubestinger-Steindl (Digital Media Specialist, Archivist)

Training, setup, support, A/V file analysis and repair, data recovery, communication with memory institutions





MediaInfo

MediaInfo is a convenient unified display of the most relevant technical and tag data for video and audio files.

Download MediaInfo

Version 18.12, Graphical User Interface with installer, Mac App Store

Other versions (packaging, OS, interface...) are also [available](#)

See [change log](#)

You can also use [MediaInfoOnline](#) to test MediaInfo without installing any software on your computer.

Please **donate** to support further development

The MediaInfo data display includes:

- Container: format, profile, commercial name of the format, duration, overall bit rate, writing application and library, title, author, director, album, track number, date, duration...
- Video: format, codec id, aspect, frame rate, bit rate, color space, chroma subsampling, bit depth, scan type, scan order...
- Audio: format, codec id, sample rate, channels, bit depth, language, bit rate...
- Text: format, codec id, language of subtitle...
- Chapters: count of chapters, list of chapters...

MediaInfo analytics include:

- Container: MPEG-4, QuickTime, Matroska, AVI, MPEG-PS (including unprotected DVD), MPEG-TS (including unprotected Blu-ray), MXF, GXF, LXF, WMV, FLV, Real...
- Tags: Id3v1, Id3v2, Vorbis comments, APE tags...
- Video: MPEG-1/2 Video, H.263, MPEG-4 Visual (including DivX, XviD), H.264/AVC, H.265/HEVC, FFV1...
- Audio: MPEG Audio (including MP3), AC3, DTS, AAC, Dolby E, AES3, FLAC...
- Subtitles: CEA-608, CEA-708, DTVCC, SCTE-20, SCTE-128, ATSC/53, CDP, DVB Subtitle, Teletext, SRT, SSA, ASS, SAMI...

MediaInfo features include:

- Read many [video and audio file formats](#)
- View information in different formats (text, sheet, tree, HTML...)
- Customise these viewing formats
- Export information as text, CSV, HTML...
- Graphical user interface, command line interface, or library (.dll/.so/.dylib) versions available
- Integrate with the shell (drag 'n' drop, and Context menu)
- Internationalisation: display any language on any operating system
- Localisation capability (for which volunteers are needed - please [contact us!](#))



MediaConch

MediaConch is an extensible, open source software project consisting of an implementation checker, policy checker, reporter, and fixer that targets preservation-level audiovisual files (specifically [Matroska](#), Linear Pulse Code Modulation (LPCM) and FF Video Codec 1 ([FFV1](#))) for use in memory institutions, providing detailed and batch-level conformance checking via an adaptable and flexible application program interface accessible by the command line, a graphical user interface, or a [web interface](#).

Download MediaConch

Version 18.03.2. for Ubuntu

Other versions (packaging, OS, interface...) are also [available](#)                 

See [ChangeLog](#) or [very latest snapshots](#)

You can also use [MediaConchOnline](#) to test MediaConch without installing any software on your computer.

Please **donate** to support further development

Check by file upload
Check online files
Check server files

Policy
Choose a policy
Display
Media/Conch Html
Verbosity
Default level
Check files

Results

Apply a policy to all results
Choose a new policy to apply
Close all results

Show 10 entries
Search:

Files	Implem	Policy	MediaInfo	MediaTrace	Status
ffv1_test_gcdfmt-yuv44p10le...	Valid	Matroska is well described?			
ffv1_test_gcdfmt-yuva422p_co...	Valid	Matroska is well described?			
ffv1_test_gcdfmt-yuva44p_co...					
veraPDF test suite 6-1-10-0...	Not valid	Matroska is well described?			
train1.tif	Not valid	Matroska is well described?			
buggy_header.pdf	Not valid	Matroska is well described?			

Showing 11 to 16 of 16 entries
Previous
1
2
Next

QCTools



QCTools is a software tool that helps users analyze and understand their digitized video files through use of audiovisual analytics and filtering.

Funded by the [National Endowment for the Humanities](#) and the [Knight Foundation](#); designed and led by Dave Rice and the [Bay Area Video Coalition](#); developed by MediaArea, [Fabio Utzig](#), [Alexander Ivash](#).

MediaArea was involved in the initial development and provides binaries for all platforms.

Download QCTools

Version 1.0, for macOS

Other versions (packaging, OS, interface...) are also available

See [ChangeLog](#) or very latest snapshots

Please **donate** to support further development

QCTools graphing features:

QCTools offers a variety of graphing features including:

- YUV Values
- Temporal Outliers (TOUT)
- Vertical Line Repetitions (VREP)
- Broadcast Range (BRNG)
- Crop Width and Height (CropW and CropH)
- Peak Signal to Noise Ratio (PSNRf)
- Mean Square Error (MSEf) differences per frame.
- ...[Learn more...](#)

QCTools playback filters:

The QCTools preview window is intended as an analytical playback environment that allows the user to review video through multiple filters simultaneously. The playback window includes two viewing windows which may be set to different combinations of filters.

- Histogram
- Waveform
- Waveform Target
- Line Select
- Vectorscope
- Vectorscope Target
- Extract Planes Equalized
- Extract Planes UV Equalized
- Bit Plan
- Bit Plane Noise
- Value Highlight
- Saturation Highlight
- Vectorscope
- ...[Learn more...](#)

1. community
2. open source tools + resources
3. open, standards-based
archival file formats

ffv1 + matroska

codec + container

an open, standards-based solution for moving
image digital preservation

FFV1 and Matroska reading list

DOWNLOAD HERE (PDF)

This is a collection of web links for further reading on the combination of FFV1 (codec) and Matroska (container), as an emerging open and standards-based digital preservation solution for the digitisation of video and film materials. This combination is going through a standardisation process via the Internet Engineering Task Force (IETF) mandated by the Preforma project, and is already in use in many archives as the preservation solution for video-source digitisation. It is also under active development as a lossless preservation solution for film scans, via the RAWcooked initiative.

Collated for FIAF by Stephen McConnachie, Head of Data, BFI, March 2018.

FFV1 and Matroska reading list

A collection of web links for further reading on the combination of FFV1 (codec) and Matroska (container), as an emerging open and standards-based digital preservation solution for the digitisation of video and film materials. This combination is going through a standardisation process via the Internet Engineering Task Force (IETF) mandated by the Preforma project, and is already in use in many archives as the preservation solution for video-source digitisation. It is also under active development as a lossless preservation solution for film scans, via the RAWcooked initiative.

FFV1 in Wikipedia, includes list of archives known to be using FFV1:

- <https://en.m.wikipedia.org/wiki/FFV1>

MATROSKA in Wikipedia:

- <https://en.m.wikipedia.org/wiki/Matroska>

CASE STUDY: University of Indiana MDPI video digitisation project, file format discussion, by Mike Casey:

- <https://mdpi.iu.edu/doc/MDPIwhitepaper.pdf>

PREFORMA / No Time To Wait: Open Source and FFV1 MKV for archives (blogs, videos, interviews, slides):

- <https://www.beeldengeluid.nl/en/knowledge/blog/tools-trade>
- <https://mediaarea.net/MediaConch/2016/07/26/No-Time-To-Wait-Preservation-FFV1-Matroska-Symposium/>
- <https://github.com/preforma/notmetowait>
- <https://blogs.loc.gov/thesignal/2015/09/improving-technical-options-for-audiovisual-collections-through-the-preforma-project>

FFV1 as preservation choice - implementation risk assessments, by Peter Bubestinger:

- http://www.av-rd.com/knowhow/video/risk_assessment.html
- <https://mediaarea.net/MediaConch/interviews/InterviewLewetzBubestinger.html>
- http://download.das-werkstatt.com/pb/fsfe/presentations/2015/the_ffv1_story.pdf
- <https://www.scarf.be/?q=nl/content/interview-met-peter-bubestinger-en-herman-lewetz-%C3%B6sterreichische-mediathek>
- http://download.das-werkstatt.com/pb/mthk/info/video/FAQ-digital_video_archiving.html

Library of Congress blog: comparing formats for video digitisation

- <https://blogs.loc.gov/thesignal/2014/12/comparing-formats-for-video-digitization/>

FFMPEG: FFV1 encoding reference:

- <https://trac.ffmpeg.org/wiki/Encode/FFV1>

FFMPEG: FFmpegprovisr to MKV using FFV1 with FrameMD5:

- https://amiaopensource.github.io/ffmpegprovisr/#create_FFV1_mkv

MediaConch: conformance checker for FFV1 / Matroska:

- <https://mediaarea.net/MediaConch/documentation/FAQ.html>

MKVToolNix: toolset for manipulating Matroska files, including adding and extracting attachments, editing metadata headers, adding chapters with annotations:

- <https://mkvtoolnix.download/>

Library of Congress file formats for video digitisation:

- http://www.digitizationguidelines.gov/guidelines/Video_reformatting_compare.html?locir=blogsig
- http://www.digitizationguidelines.gov/guidelines/FADGI_VideoReFormatCompare_pt5_20141202.pdf

New South Wales Govt: preservation file formats assessed:

- <https://web.archive.org/web/20131110105550/http://www.records.nsw.gov.au/recordkeeping/government-recordkeeping-manual/guidance/guidelines/guideline-22/digital-audio-and-video-file-formats-form-formats-and-codecs-for-digital-video-preservation>

FIXITY: FrameMD5 article, by Dave Rice:

- <http://dericed.com/papers/reconsidering-the-checksum-for-audiovisual-preservation>

FILM SCANS: FFV1 MKV as solution for film and video sources, FIAF JFP article by Reto Kromer:

- http://www.fiafnet.org/images/storyUpload/Publications/Journal-Of-Film-Preservation/Matroska-and-FFV1_Kromer_JFP96.pdf

FILM SCANS: FFV1 MKV solution discussed by Kieran O'Leary from IFI:

- <https://kieranjol.wordpress.com/2016/10/07/introduction-to-ffv1-and-matroska-for-film-scans/>

ffv1 + matroska are open

328 commits1 branch11 releases8 contributors

Branch: masterNew pull requestFind FileClone or download

dericed and Michael Niedermayer bump version to ffv1-08_v4-05Latest commit cbe76a3 on Feb 26

g!ghignore	upgrading to a working group document	2 years ago
Makefile	bump version to ffv1-08_v4-05	23 days ago
README.md	add link to version 4	6 months ago
ffv1.md	Replaced NBSPs with spaces.	a month ago
pdf_backmatter.md	pdf_backmatter.md - updates dead links	2 years ago
pdf_frontmatter.md	move pdf frontmatter to separate header file	3 years ago
r!fc_frontmatter.md	bump version to ffv1-08_v4-05	23 days ago
style.css	CSS stylesheet now locally available	3 years ago

README.md

FFV1 Specification

Introduction

This repository manages the development of specification documents for FFV1, a lossless intra-frame video codec. The goals of this specification effort are defined within the charter of the [cellar working group](#) of the [Internet Engineering Task Force](#). Information within this repository should be considered in draft form. The most recent official version of the specification for FFV1 versions 0, 1 and 3 may be found at <https://datatracker.ietf.org/doc/draft-ietf-cellar-ffv1/> and for FFV1 version 4 at <https://datatracker.ietf.org/doc/draft-ietf-cellar-ffv1-v4/>. An HTML rendering is also available via <https://ffmpeg.org/~michael/ffv1.html>.

Formatting

The FFV1 specification was initially written in lyx. In July 2015 the formatting of the specification was transitioned to Markdown. Propering PDF and HTML rendering has been tested with requires pandoc version 1.13.2.1 and higher.

The Markdown version of the FFV1 specification may also be converted into XML, HTML and text formats as an IETF RFC draft. Producing the RFC formats requires mmark version 1.3.4 or higher and xml2rfc version 2.5.1 or higher.

A Makefile is provided that can produce the PDF and RFC outputs.

Version Handling

The ffv1.md file is currently intended to be used to produce documentation for both FFV1 versions 0, 1, and 3 as well as FFV1 version 4. Lines containing (v4) will be suppressed from the version 0,1,3 outputs while lines containing (v3) will be suppressed from the version 4 outputs.

FFV1 Video Codec Specification

by Michael Niedermayer michaelni@gmx.at

- 1 Introduction
- 2 Notation and Conventions
 - 2.1 Definitions
 - 2.2 Conventions
 - 2.2.1 Arithmetic operators
 - 2.2.2 Assignment operators
 - 2.2.3 Comparison operators
 - 2.2.4 Mathematical functions
 - 2.2.5 Order of operation precedence
 - 2.2.6 Range
 - 2.2.7 NumBytes
 - 2.2.8 Bitstream functions
- 3 General Description
 - 3.1 Border
 - 3.2 Median predictor
 - 3.3 Context
 - 3.4 Quantization
 - 3.5 Colospace
 - 3.5.1 YCbCr
 - 3.5.2 JPEG2000-RCT
 - 3.6 Coding of the sample difference
 - 3.6.1 Range coding mode
 - 3.6.2 Huffman coding mode
- 4 Bitstream
 - 4.1 Configuration Record
 - 4.1.1 reserved for future use
 - 4.1.2 configuration_record_crc_parity
 - 4.1.3 Mapping FFV1 into Containers
 - 4.2 Frame
 - 4.3 Slice
 - 4.4 Slice Header
 - 4.4.1 slice_x
 - 4.4.2 slice_y
 - 4.4.3 slice_width
 - 4.4.4 slice_height
 - 4.4.5 quant_table_index_count
 - 4.4.6 quant_table_index
 - 4.4.7 picture_structure
 - 4.4.8 sar_num
 - 4.4.9 sar_den
 - 4.4.10 reset_contexts
 - 4.4.11 slice_coding_mode
 - 4.5 Slice Content
 - 4.5.1 primary_color_count
 - 4.5.2 plane_pixel_height
 - 4.5.3 slice_pixel_height
 - 4.5.4 slice_pixel_y
 - 4.6 Line
 - 4.6.1 plane_pixel_width
 - 4.6.2 slice_pixel_width
 - 4.6.3 slice_pixel_x
 - 4.7 Slice Footer
 - 4.7.1 slice_size
 - 4.7.2 error_status
 - 4.7.3 slice_crc_parity

3.7.2. RGB

JPEG2000-RCT is a Reversible Color Transform that codes RGB (red, green, blue) planes losslessly in a modified YCbCr color space [ISO.15444-1.2016]. Reversible Pixel transformations between YCbCr and RGB use the following formulae.

$Cb=b-g$

$Cr=r-g$

$Y=g+(Cb+Cr)>>2$

$g=Y-(Cb+Cr)>>2$

$r=Cr+g$

$b=Cb+g$

Exception for the JPEG2000-RCT conversion: if bits_per_raw_sample is between 9 and 15 inclusive and alpha_plane is 0, the following formulae for reversible conversions between YCbCr and RGB MUST be used instead of the ones above:

$Cb=g-b$

$Cr=r-b$

$Y=b+(Cb+Cr)>>2$

$b=Y-(Cb+Cr)>>2$

$r=Cr+b$

$g=Cb+b$

Background: At the time of this writing, in all known implementations of FFV1 bitstream, when bits_per_raw_sample was between 9 and 15 inclusive and alpha_plane is 0, GBR planes were used as BGR planes during both encoding and decoding. In the meanwhile, 16-bit JPEG2000-RCT was implemented without this issue in one implementation and validated by one conformance checker. Methods to address this exception for the transform are under consideration for the next version of the FFV1 bitstream.

When FFV1 uses the JPEG2000-RCT, the horizontal lines are interleaved to improve caching efficiency since it is most likely that the RCT will immediately be converted to RGB during decoding. The interleaved coding order is also Y, then Cb, then Cr, and then if used Alpha.

As an example, a frame that is two pixels wide and two pixels high, could be comprised of the following structure:

P	P	P	P
Y[1,1]	Cb[1,1]	Cr[1,1]	Y[2,1]
Y[1,2]	Cb[1,2]	Cr[1,2]	Y[2,2]
Y[1,1]	Cb[1,1]	Cr[1,1]	Y[2,1]
Y[1,2]	Cb[1,2]	Cr[1,2]	Y[2,2]

In JPEG2000-RCT, the coding order would be left to right and then top to bottom, with values interleaved by lines and stored in this order:

$Y[1,1] Y[2,1] Cb[1,1] Cb[2,1] Cr[1,1] Cr[2,1] Y[1,2] Y[2,2] Cb[1,2] Cb[2,2] Cr[1,2] Cr[2,2]$

3.8. Coding of the Sample Difference

Instead of coding the n+1 bits of the Sample Difference with Huffman or Range coding (or n+2 bits, in the case of RCT), only the n (or n+1) least significant bits are used, since this is sufficient to recover the original sample. In the equation below, the term "bits" represents bits_per_raw_sample+1 for RCT or bits_per_raw_sample otherwise:

$coder_input = [(sample_difference + 2^{(bits-1)}) \& (2^{bits} - 1)] - 2^{(bits-1)}$

<https://github.com/FFmpeg/FFV1>

ffv1 is lossless <and> compressed

```
#tb 0: 1001/30000
0,      0,      0,      1, 1669440, 1fb241f71b9b14abdf88ad5034b6dc21
0,      1,      1,      1, 1669440, 38310375ae195c17019e26da9d99e3d0
0,      2,      2,      1, 1669440, c154e232f7f5cb74a60afc06e11cabae
0,      3,      3,      1, 1669440, 508b0d017ffa6f4694541762ed5fae6a
0,      4,      4,      1, 1669440, 36f5da2bceef0973550585e91f748d1a
0,      5,      5,      1, 1669440, d36fd15efdf503c1ef25640d890917b3
0,      6,      6,      1, 1669440, 31b7232bf8e6fd2337e2beddc480dc42
0,      7,      7,      1, 1669440, 7ab5486e5999d86dd016ae0b8df13a70
0,      8,      8,      1, 1669440, 47b2a83dd6801d2c2bd414f57af8eff5
0,      9,      9,      1, 1669440, b883d73e78c230b220f311e8fb34e6ee
0,     10,     10,      1, 1669440, 4171860688591526ad3c9c3780eb044f
0,     11,     11,      1, 1669440, ad8df2d43442eb45155300965e4f59d0
0,     12,     12,      1, 1669440, 9bf60490424ebc2b5209d5d2ba3398d9
0,     13,     13,      1, 1669440, 7184bf36a237199e68afe9b51ef23e5e
0,     14,     14,      1, 1669440, 905b35a7638b53566cd5235d1dedfdc0
0,     15,     15,      1, 1669440, e0f3577df7cbe6420d712be7abc1733
0,     16,     16,      1, 1669440, d20aa192b1a8da3ffa26d16464ef4ef5
0,     17,     17,      1, 1669440, 84bf9143b1e1d33fa60dd04fddcf6d2e
0,     18,     18,      1, 1669440, f18784efb0da45b418d763857a616ec6
0,     19,     19,      1, 1669440, d86e92e1046c5190b9582fc527c36c69
0,     20,     20,      1, 1669440, cd37e29476412d8ff2a7effdbb538d60
0,     21,     21,      1, 1669440, 78fda53e3b2e88029fc42b347c4045fc
0,     22,     22,      1, 1669440, 3f4718d7d93899497c314a7b65ec2f95
0,     23,     23,      1, 1669440, 3650ecff2013c0bac2d8a9006972f842
0,     24,     24,      1, 1669440, de9b78e46be1ed555dfbd16d73773dd4
0,     25,     25,      1, 1669440, 3ab9ab618d930b79e9f2396d95de5ca9
0,     26,     26,      1, 1669440, e40524fab40c44811a8d21b641b4af16
0,     27,     27,      1, 1669440, d44cc0cfea82fb7b14a9b62c713c9500
0,     28,     28,      1, 1669440, 29f6ca7e17a378f939a4b4153bb258de
0,     29,     29,      1, 1669440, 7db13c711801b7a90b17e3a891035088
```

50gb v210 mov



ffmpeg



22gb ffv1 mkv

ffv1 + matroska are standards-based



Codec Encoding for LossLess Archiving and Realtime transmission (cellar)

About	Documents	Meetings	History	Photos	Email expansions	List archive	Tools »
WG	Name	Codec Encoding for LossLess Archiving and Realtime transmission					
	Acronym	cellar					
	Area	Applications and Real-Time Area (art)					
	State	Active					
	Charter	charter-ietf-cellar-01 Approved					
	Dependencies	Document dependency graph (SVG)					
	Additional URLs	<div>- Wiki</div> <div>- Issue tracker</div>					
Personnel	Chairs	<div><input checked="" type="checkbox"/> Michael Richardson</div> <div><input checked="" type="checkbox"/> Tim Terriberry</div>					
		<div><input checked="" type="checkbox"/> Alexey Melnikov</div>					
	Area Director	<div><input checked="" type="checkbox"/> Alexey Melnikov</div>					
Mailing list	Address	cellar@ietf.org					
	To subscribe	https://www.ietf.org/mailman/listinfo/cellar					
	Archive	https://mailarchive.ietf.org/arch/browse/cellar/					
Jabber chat	Room address	xmpp:cellar@jabber.ietf.org?join					
	Logs	https://jabber.ietf.org/logs/cellar/					

Charter for Working Group

The preservation of audiovisual materials faces challenges from technological obsolescence, analog media deterioration, and the use of proprietary formats that lack formal open standards. While obsolescence and material degradation are widely addressed, the standardization of open, transparent, self-descriptive, lossless formats remains an important mission to be undertaken by the open source community.

FFV1 is a lossless video codec and Matroska is an extensible media container based on EBML (Extensible Binary Meta Language), a binary XML format. There are open source implementations of both formats, and an increasing interest in and support for use of FFV1 and Matroska. However, there are concerns about the sustainability and credibility of existing specifications for the long-term use of these formats. These existing specifications require broader review and formalization in order to encourage widespread adoption.

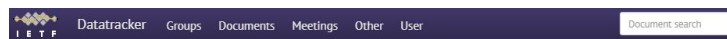
There is also a need for a lossless audio format to complement the lossless video codec and container format. FLAC is a lossless audio codec that has seen widespread adoption in a number of different applications including archival applications. While there are open source implementations of the codec, no formal standards for either the codec itself or its use in container formats currently exist. Review and formalization of the FLAC codec standard and its use in Matroska container formats is needed for wider adoption.

Using existing work done by the development communities of Matroska, FFV1, and FLAC, the Working Group will formalize specifications for these open and lossless formats. In order to provide authoritative, standardized specifications for users and developers, the Working Group will seek consensus throughout the process of refining and formalizing these standards. Initial specifications can be accessed here:

Specifications:

- FFV1: <https://mediaarea.net/temp/ffv1.html>
- Matroska: <http://matroska.org/technical/specs/index.html>
- EBML: <http://matroska-og.github.io/libebml/specs.html>
- FLAC: <https://xiph.org/flac/format.html>

<https://datatracker.ietf.org/wg/cellar/about/>



Document	Type	Active Internet-Draft (cellar WG)
	Last updated	2019-02-06
	Stream	IETF
	Intended RFC status	(None)
	Formats	plain text xml pdf html bibtex
Stream	WG state	WG Document (wg milestone: Dec 2018 - Submit specification...)
	Document shepherd	No shepherd assigned
IESG	IESG state	I-D Exists
	Consensus	Unknown
	Boilerplate	
	Telechat date	
Responsible AD	(None)	
	Send notices to (None)	

[Email authors](#) [Email WG](#) [IPR](#) [References](#) [Referenced by](#) [Nits](#) [Search lists](#)

cellar

Internet-Draft

Intended status: Standards Track

Expires: August 10, 2019

M. Niedermayer

D. Rice

J. Martinez

February 6, 2019

FFV1 Video Coding Format Version 4
draft-ietf-cellar-ffv1-v4-04

Abstract

This document defines FFV1, a lossless intra-frame video encoding format. FFV1 is designed to efficiently compress video data in a variety of pixel formats. Compared to uncompressed video, FFV1 offers storage compression, frame fixity, and self-description, which makes FFV1 useful as a preservation or intermediate video format.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 10, 2019.

ffv1 + matroska are by archivists for archivists

Wednesday, 10 April 2019

WEBSITE PROJECT PARTNERS TENDER ACTIVITIES OPEN SOURCE PORTAL COMMUNITY DOWNLOAD CONTACTS

BLOG HOME PREFORMA WEBSITE

Username

Password

MEDIA PARTNER

DIGITAL CULTURE

RELEVANT NEWS from Digitalmeetsculture



a project co-funded by the EU in the framework of CEF Connecting Europe Facility programme

50s in Europe Kaleidoscope (2018-2020)



registration is still open for the event to take place in Boston and Cambridge, Massachusetts, 24-27 September

Going to IPRES2018, the 15th International Conference on Digital Preservation



Cyprus, 29 October – 3 November 2018

7th EUROMED 2018

Further standardisation of Matroska and FFv1

IETF 93 in Prague, VideoLAN Dev Days in Paris and much more
Posted on: 30 September 2015

[Share / Save](#) [f](#) [t](#) [v](#) [+](#)

[Leave a comment](#)

Here are a few updates about the efforts that PREFORMA is conducting towards the standardisation of Matroska and FFv1 thanks to MediaArea, one of the PREFORMA suppliers that is working in the prototyping phase.



Tessa chairing 35 minutes of conference time at IETF 93 in Prague on the subject of whether the IETF should proceed with efforts to further standardize Matroska and FFv1. Attendance at IETF allowed MediaArea and PREFORMA to establish relationships with IETF members/directors who could assist in coordination of standardization efforts within IETF guidelines and procedures. Representatives from both formats participate in the meeting and subsequent online discussion were carried out with the IETF DISPATCH working group.

Representatives of cultural heritage organizations that use Matroska and FFv1, such as the Austrian Mediathek and the UK National Archives also participated in discussions on these formats directly with IETF representatives. Tessa Fallon has worked with Matroska, FFv1, and Flac (at the request of IETF) communities to draft and propose a working group charter which is currently under consideration by the IETF's IESG. Prior to the approval and establishment of the working group, much of the work is focused on building relationships and offline coordination and planning.

Dave Rice gave a [presentation](#) in Bern, Switzerland discussing the PREFORMA project and discussed it at VideoLAN Dev Days, an "unconference" held in Paris, France. The discussion at VDD included many Matroska users such as representatives from YouTube, Obencoder, and other open media groups on the standardization effort and there was a productive conversation on merging the standardization efforts from Google/YouTube on webM (a Matroska derivative) with the PREFORMA-facilitated IETF work on the Matroska specification. This has led to itemized listserv discussions of proposals to advance the container towards making it more self-descriptive and future-ready; see: <http://lists.matroska.org/pipermail/matroska-devel/2015September/thread.html>.



CONTACTS

Project Coordinator: Borje Justrell

Technical Coordinator: Antonella Fresca

Communication Coordinator: Claudio Frandini

Scientific Coordinator: Nicola Ferro

Project website: www.preforma-project.eu

COORDINATOR



TECHNICAL COORD.



PARTNERS



MediaArea

MediaInfo

Projects

Support Us!

Log in

Tags: [MediaConch](#) [Presentations](#) [No Time to Wait](#)

No Time to Wait: Standardizing FFV1 and Matroska for Preservation



Photo credit: CC BY-SA Erwin Verbruggen

Introduction

No Time to Wait!: Standardizing FFV1 & Matroska for Preservation was a symposium intentionally overlapping with Internet Engineering Task Force's 96th meeting, held in Berlin. No Time to Wait! was held on 18-20 July, 2016 and hosted by Deutsche Kinemathek, Zuse Institute Berlin, and MediaArea. The symposium was designed to bring together audiovisual archivists and audiovisual format designers with a focus on the standardization of a preservation-grade audiovisual file format combination package. The structure of this symposium was contingent heavily on the CELLAR working group and the initial meeting of this working group at IETF and first rounds of RFCs submitted to the organization.

Why these formats?


After introductions, the first talk of the symposium was from Erwin Verbruggen (Netherlands Sound & Vision), who gave a summary of the PREFORMA project. And with it, the insight and history into the decision-making behind the selection of these open formats and how they compared against other potential options to use in the development of a conformance checker for preservation-grade audiovisual formats. As obvious by this symposium, Matroska and FFV1 (and LPCM) were chosen by PREFORMA. The Internet Engineering Task Force working group, formed last year, adopted Matroska and FFV1 but chose to focus on FLAC.

Steve Lhomme was able to attend the symposium and the IETF meeting, which is amazing because he is one of the founding developers of the Matroska format. His continual input on the CELLAR listserv and during the conference was absolutely invaluable. By the end of the symposium, Steve also had a thorough understanding of the unique needs of archivists and he was happy to assist in the required mapping work to ensure his format is suitable for this use case.

Fun fact: Steve original came up with the Matroska format because he was trying to catch Jacques Chirac, at the time President of France, lying on television. The origins of this format seem very archivally-minded, even if that context was not known or considered at the time.

ffv1 + matroska are by archivists for archivists

Archivists' Guide to Matroska




#amia17 @dericed
@av_morgan #matroska
2017-11-29

MATROŠKA

ffv1 + matroska are by archivists for archivists

 MKVToolNix – Matroska tools for Linux/Unix and Windows

News Downloads Source code Documentation Troubleshooting Blog FAQ Bugs Authenticity Links Community RSS 2.0

Support development!  

About MKVToolNix

MKVToolNix is a set of tools to create, alter and inspect [Matroska](#) files under Linux, other Unices and Windows.

Since May 1st 2003, the Matroska libraries themselves and my Matroska tools are officially available. I urge you to download the software, use it and report any errors you encounter to me. I'm always grateful for good [bug reports](#) because I make mistakes and you guys normally find them. Your feedback is important to me :)



MKVToolNix

mkvtoolnix-gui -- a GUI for [mkvmerge\(1\)](#) including a chapter and a header editor

1. Synopsis

```
mkvtoolnix-gui [[configuration-file-name.mxcfg] | [source-file.ext] | [[matroska-file.mkv]] | [[chapter-file.ext]] | [[matroska-file.mkv]]]
```

2. Description

mkvtoolnix-gui(1) is a Qt^(tm) based GUI for *mkvmerge(1)*. It also implements *mkvinfo(1)*'s and *mkvpropedit(1)*'s functionality and will evolve to cover *mkvextract(1)* as well. All settings (e.g. source files, track options etc) can be saved and restored.

mkvpropedit -- Modify properties of existing *Matroska*^(tm) files without a complete remux

1. Synopsis

```
mkvpropedit [options] {source-filename} {actions}
```

2. Description

This program analyses an existing *Matroska*^(tm) file and modifies some of its properties. Then it writes those modifications to the existing file. Among the properties that can be changed are the segment information elements (e.g. the title) and the track headers (e.g. the language code, 'default track' flag or the name).

mkvinfo -- Print information about elements in *Matroska*^(tm) files

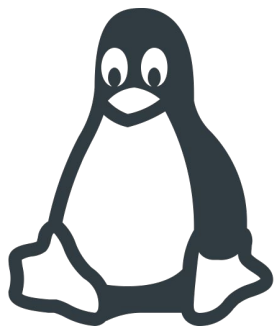
1. Synopsis

```
mkvinfo [options] {source-filename}
```

2. Description

This program lists all elements contained in a *Matroska*^(tm). The output can be limited to a list of tracks in the file including information about the codecs used.

1. community
2. open source tools + resources
3. open, standards-based
archival file formats
4. automation





API website

[Getting started](#)[API](#)[Downloads](#)[Documentation](#)[Demos](#)[Resources](#)

API Browser

[Search ▶](#)[Write ▶](#)[Session ▶](#)[Select ▶](#)[Lock ▶](#)[Utilities ▶](#)[Images ▶](#)

Welcome

News

The Adlib API is a command library that can be used to interact with Adlib databases. Using this API you can easily build your own applications through one of the three available methods: Url requests, Adlib.Data Windows dll or jQuery plugin.

This website offers extensive documentation, including an overview of all [API commands ▶](#) and many examples on how to use them. You will also find links to other relevant [documentation ▶](#), downloads of sample code, [demos ▶](#) and [resources ▶](#). A [forum ▶](#) is also available to help you and share information with other Adlib API users.

Try the Adlib API now!

The following Url request is an example of a Boolean search to retrieve *paintings* by *Bernadino Fungai* or *William Etty*

```
http://test2.adlibsoft.com/api/wwwopac.ashx?database=c
```

[run example ▶](#) [select text ▶](#)

© Axiell ALM Netherlands, 2010-2018.

<http://api.adlibsoft.com/>

Search docs

Introduction

Imagen documentation

Applications

Webhooks

Authentication

Endpoints

Core

REST API

GET core/v1/application/

MediaControlCentre

REST API

GET imcc/v1/import/record/

PUT imcc/v1/import/record/

PUT imcc/v1/import/record/

Resource URL

Resource Information

Additional Request Headers

Parameters

Response

Examples

Create Record Example

Ingest Image Example

Update Records Example

Update By Search Example

DELETE imcc/v1/import/record/

GET imcc/v1/import/mappings/

GET imcc/v1/profiles/

GET imcc/v1/workflows/

DELETE imcc/v1/media/

Standard error

StorageService

REST API

GET storage/v1/file/

GET storage/v1/urls/

Standard error

[Docs](#) » [MediaControlCentre](#) » [PUT imcc/v1/import/record/](#)

PUT imcc/v1/import/record/

Submit record(s) to add/update, in user defined XML form. IMCC will transform this into ImagenRecord(s) using the specified XSL (eXtensible Stylesheet Language) mapping file and add to the database. Records may also be updated with this method. This call will expect the body of the call to have Content-Type "text/xml" or "application/xml".

Resource URL

`https://{MEDIACONTROLCENTRE_HOST}:{PORT}/imcc/v1/import/record/`

Resource Information

Response formats	XML
Requires authentication?	Yes (user context only)
Required scope	modifyUserData

Additional Request Headers

X-Imagen-Pathprefix

optional

The path to the directory where the media you want to ingest is stored. If specified then IMCC will add this to the paths specified in the media object elements, if not specified these paths will be used as is.

The X-Imagen-Pathprefix value should be a fully qualified path and can be a UNC path, local path, HTTP URL or HTTPS URL.

For example, if the Media Element OriginalFile is "koala.jpg" and the X-Imagen-Pathprefix is "\\testing\myshare\" then StorageService will look for the file in "\\testing\myshare\koala.jpg", but if the Media Element OriginalFile is "01234.mpg" and the X-Imagen-Pathprefix is "\\testing\myvideos\record_" then StorageService will look for the file in "\\testing\myvideos\record_01234.mpg".

Parameters

Optional parameters should be supplied as query parameters.

Mapping

required

SDK Design Best Practices

Introduction

The purpose of this page is to provide BlackPearl client developers with guidance and tips on how to build an integration to BlackPearl using our [Software Development Kits \(SDKs\)](#). We strongly recommend you use our SDKs instead of using direct connection to BlackPearl's http-based Application Program Interface (API). For additional assistance, please post questions to our [Google Group](#). The intended audience for this page is someone developing code with the BlackPearl SDKs or APIs.

Do NOT Use Traditional Amazon S3 PUT and GET Commands

A BlackPearl client should **not** use the traditional Amazon S3 [PUT Object](#) and [GET Object](#) commands to move files to and from BlackPearl. The client should use the [Spectra S3 \(formerly DS3\) Bulk Commands](#). The BlackPearl SDKs use these bulk commands. Use of traditional S3 PUT and GET commands will result in poor performance and/or errors when BlackPearl is used with a tape library. We will only [certify](#) integrations using the Spectra S3 commands, and we will only support certified integrations.

Group Files into "Jobs" When Archiving or Restoring

The Spectra S3 bulk commands described above group files into "jobs" when they are transferred. Grouping the files significantly increases the performance of the BlackPearl environment, especially when archiving or restoring with tape. **Every effort should be made to put as many files into a job as possible to maximize performance. Single-file jobs should be avoided.**

When transferring files in a job, multiple files or file parts must be uploaded in parallel to achieve maximum performance. See the "Bandwidth and Performance" section below.

Determine SDK Approach — Helper Classes

The architecture and workflows of the existing software being integrated to BlackPearl will help determine how best to use the SDKs. Some of the SDKs include "Helper" classes that simplify file movement and optimize performance. **The client should use the Helper classes** if they are available and will fit in the architecture/workflow.

To determine if the Helper classes can be used, first identify which Software Development Kit (SDK) will be used. The SDKs are available in five languages – Java, .NET/C#, Python, C, and Go.

Recent Posts

- [Using the New BlackPearl "Staging Objects" Feature](#)
- [BlackPearl 5 Simulator Now Available](#)
- [Upcoming BlackPearl 5.0 Release Requires Client Recertification](#)
- [BlackPearl Eon Browser 2.1 Official Release](#)
- [Eon Browser 2.1 Beta Available for Testing](#)

Archives

- [June 2018](#)
- [May 2018](#)
- [March 2018](#)
- [February 2018](#)
- [December 2017](#)
- [September 2017](#)
- [May 2017](#)
- [April 2017](#)
- [February 2017](#)
- [December 2016](#)
- [November 2016](#)
- [July 2016](#)
- [June 2016](#)
- [May 2016](#)
- [March 2016](#)
- [February 2016](#)
- [January 2016](#)
- [December 2015](#)
- [November 2015](#)
- [October 2015](#)
- [September 2015](#)
- [August 2015](#)
- [July 2015](#)
- [June 2015](#)
- [March 2015](#)
- [February 2015](#)

Why should I care about BPMN?

Business Process Model and Notation (BPMN) is the global standard for process modeling and one of the most important components of successful Business-IT-Alignment.

More and more organizations are using BPMN and in more and more universities BPMN is taught as a subject. These are the reasons:

- Standard** BPMN is not owned by a certain enterprise but by an institution (OMG), which is already established through other world-wide standards, e.g., UML. The standard is supported by many software products; you are less dependent on any particular vendor's products.
- Simplicity** The principle behind BPMN is rather simple which is why you can start working with this notation very quickly.
- Power of expression** If necessary, you can describe precisely how a process functions with BPMN. However, this is more difficult than only roughly describing the process. This way of precise modeling is possible, but not mandatory.
- Implementation in IT** BPMN has been primarily developed to support technical implementation of processes ("Process Automation"). The more important the IT is in a company, the more helpful the use of BPMN becomes.

A simple flow in BPMN

Let's begin our BPMN tutorial with a rather simple process diagram:

Hover over orange symbols for explanation



This diagram shows a simple process triggered by someone being hungry. The result is that someone must shop for groceries and prepare a meal. After that, someone will eat the meal and have his or her hunger satisfied.

Best Practice: Naming Conventions

When naming tasks, we try to adhere to the object-orientated design principle of using the [verb] + [object] pattern. We would say "acquire groceries," for example, not "first take care of shopping for groceries."

Events refer to something that has already happened regardless of the process (if they are catching events) or as a result of the process (if they are throwing events). For this reason, we use the [object] and make the [verb] passive in voice, so we write "hunger noticed." BPMN does not require you to model start and end events for a process - you can leave them out - but if you model a start event, you must model an end event for each path. The same is true for end events, which require start events. We always create our models with start and end events for two reasons: first, that way it's possible to determine the process trigger, and second, you can describe the final status of each path end. We only sometimes abandon this practice with sub-processes. More on this later.

Business Process Model and Notation

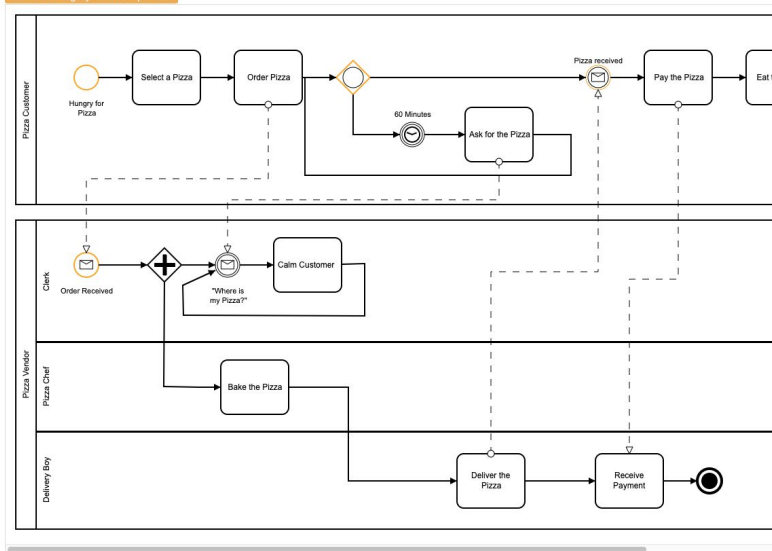


a graphical representation for modelling business processes for humans, systems, scripts, logistics

Pizza Collaboration

This example is about Business-To-Business-Collaboration. Because we want to explicitly model the interaction between a pizza customer and the vendor, we have classified them as "participants", therefore providing them with dedicated pools:

Hover over orange symbols for explanation

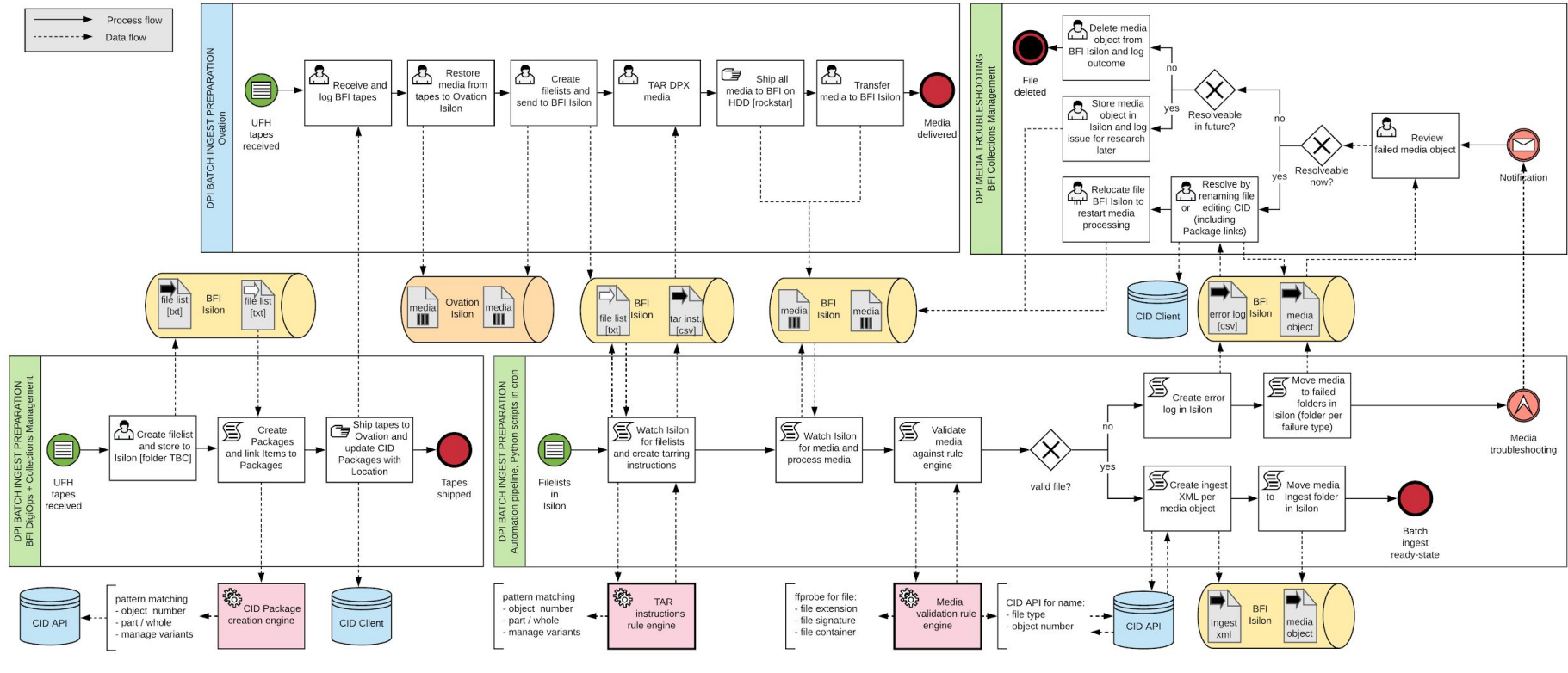


Please note that there are no default semantics in this type of modeling, which means you can model collaboration diagrams to show the interaction between business partners, but also zoom into one company, modeling the interaction between different departments, teams or even single workers and software systems in collaboration diagrams. It is totally up to the purpose of the model and therefore a decision the modeler has to make, whether a collaboration diagram with different pools is useful, or whether one should stick to one pool with different lanes, as shown in the previous chapter.

BFI legacy data ingest BPMN

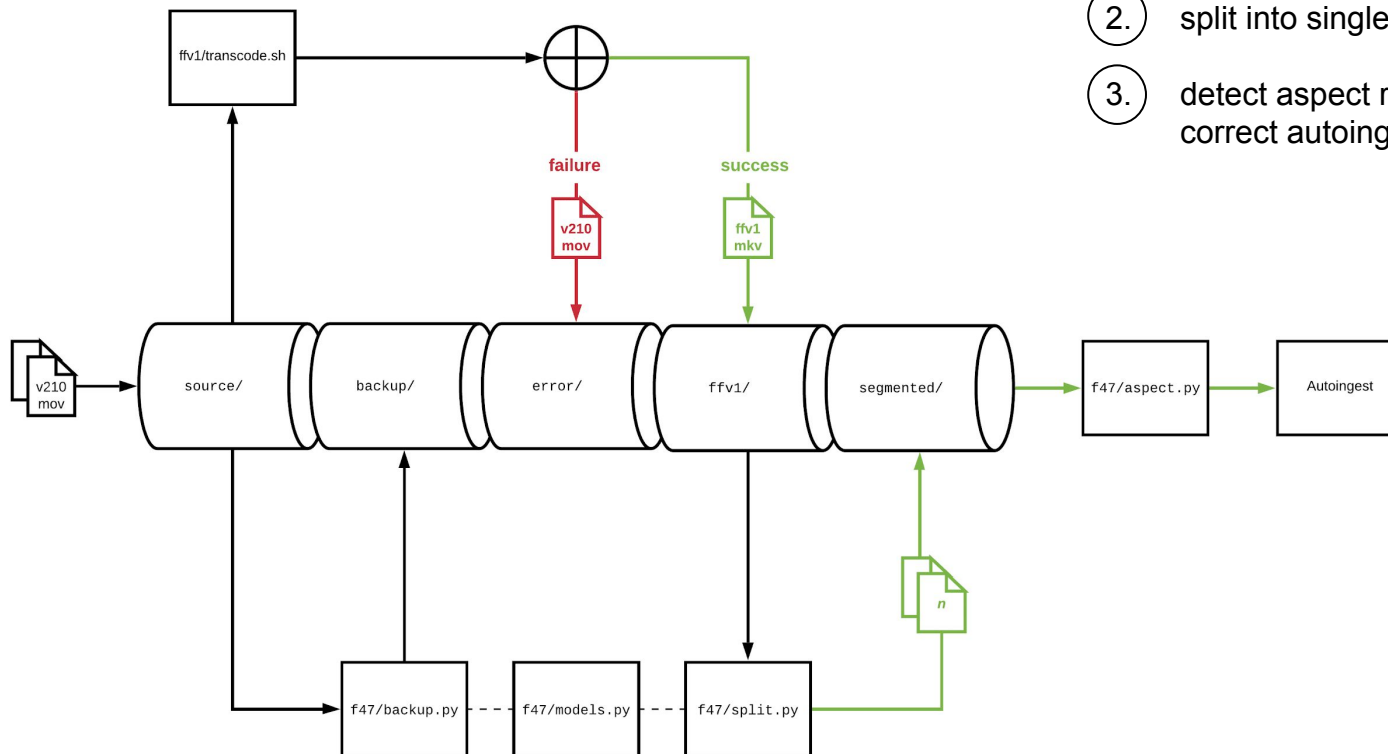
DPI BATCH INGEST PREPARATION V.4

Stephen McConnachie and Edward Anderson | April 7, 2019



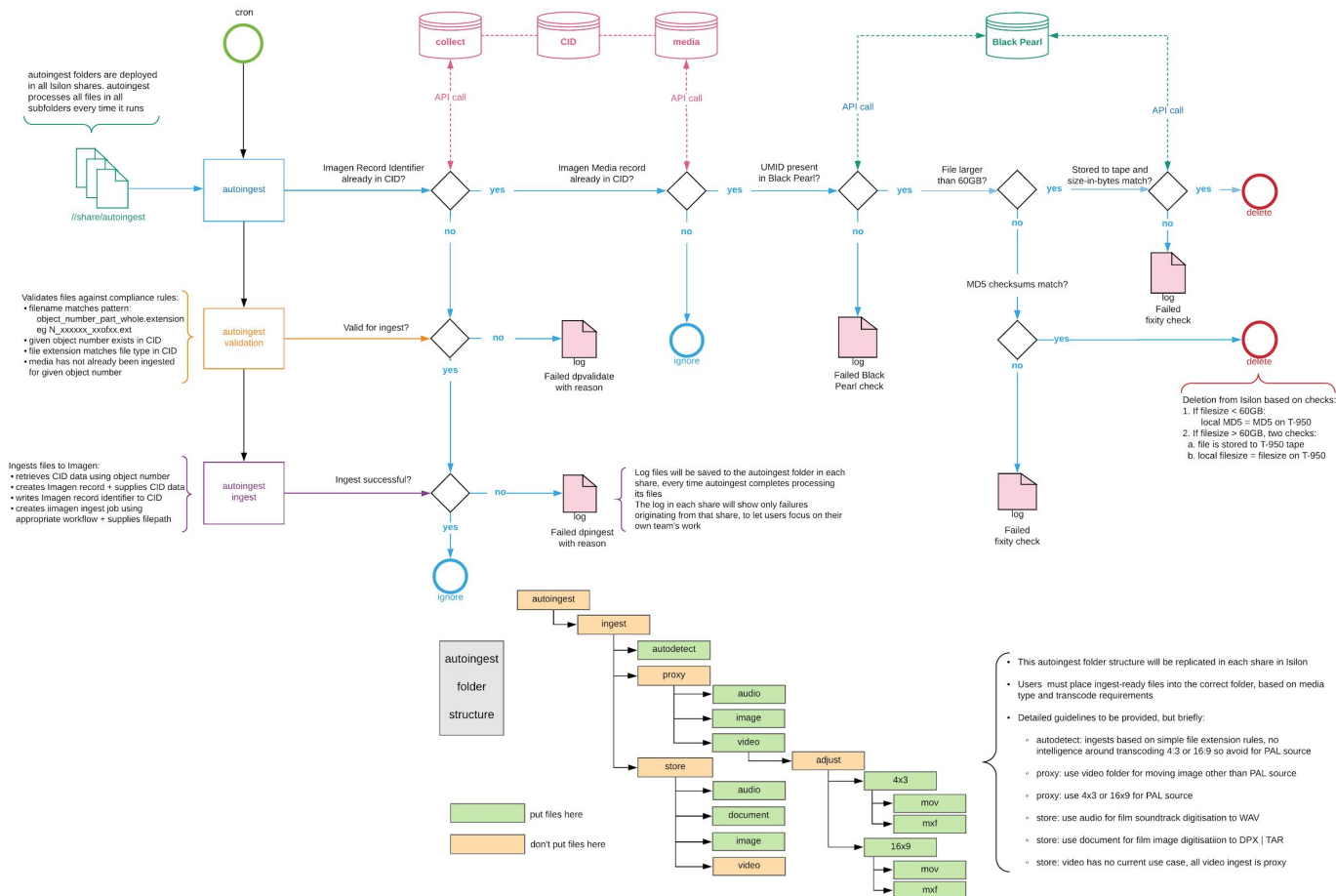
video digitisation | post-capture processing | bash + python + ffmpeg + ffprobe

1. transcode to ffv1 matroska with framemd5 validation
2. split into single programmes
3. detect aspect ratio and triage into correct autoingest workflow



autoingest

1. validate file against policies
2. ingest using correct transcode workflow
3. confirm bit-perfect storage to data tape x 2
4. delete file from network storage
5. log actions and errors at every stage



Mediatheque at BFI Southbank

The new BFI Southbank Mediatheque has reopened with thousands more titles and a fresh new look.



The Mediatheque is evolving, with constantly growing content, a completely new digital platform and new facilities.

Thanks to continuing digitisation work at the BFI National Archive, the content available on the new Mediatheque service is constantly growing in both depth and breadth.

Beautiful South: Lambeth and Southwark on screen



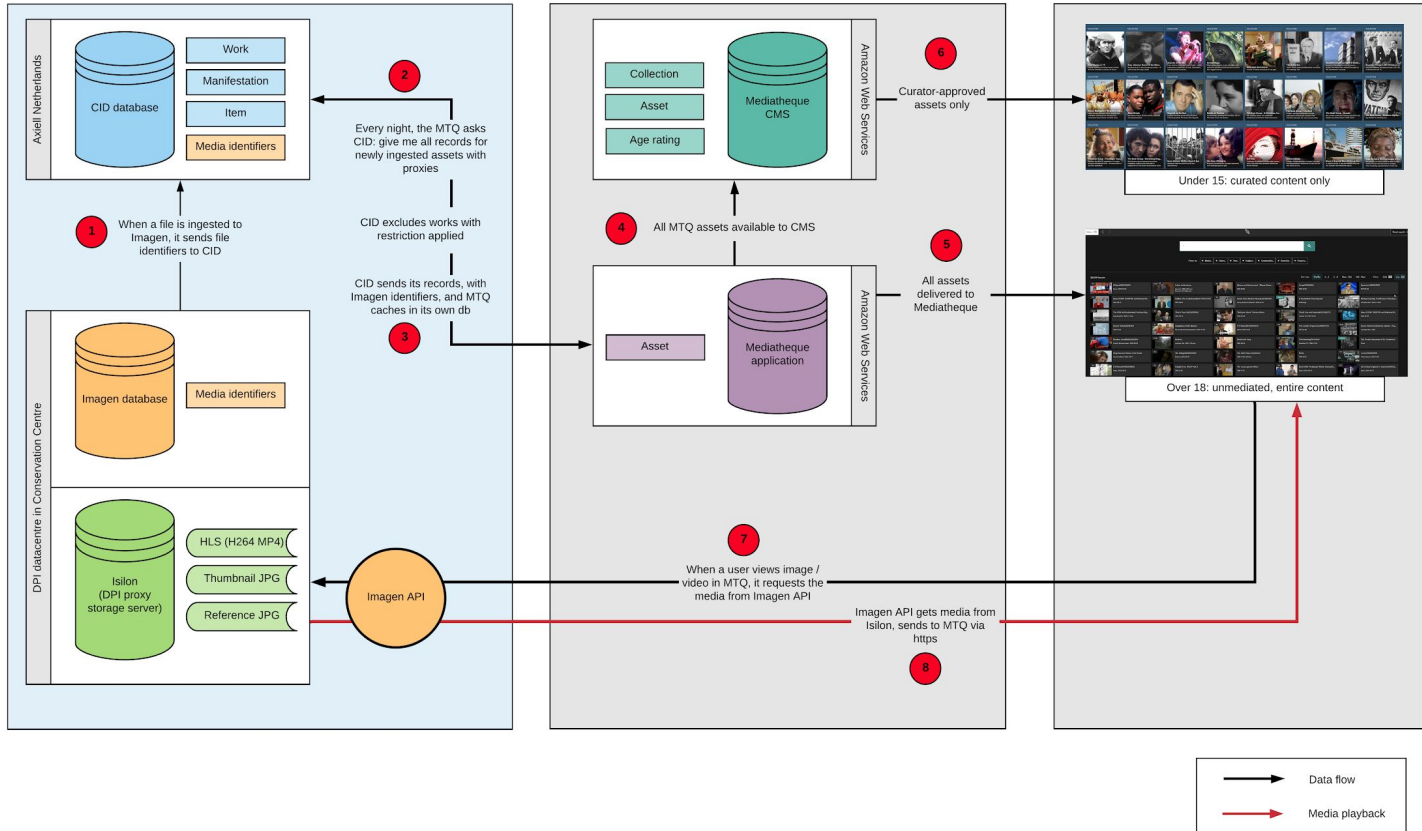
As part of the Mayor's Story of London project we explore two vibrant boroughs on BFI Southbank's own doorstep.

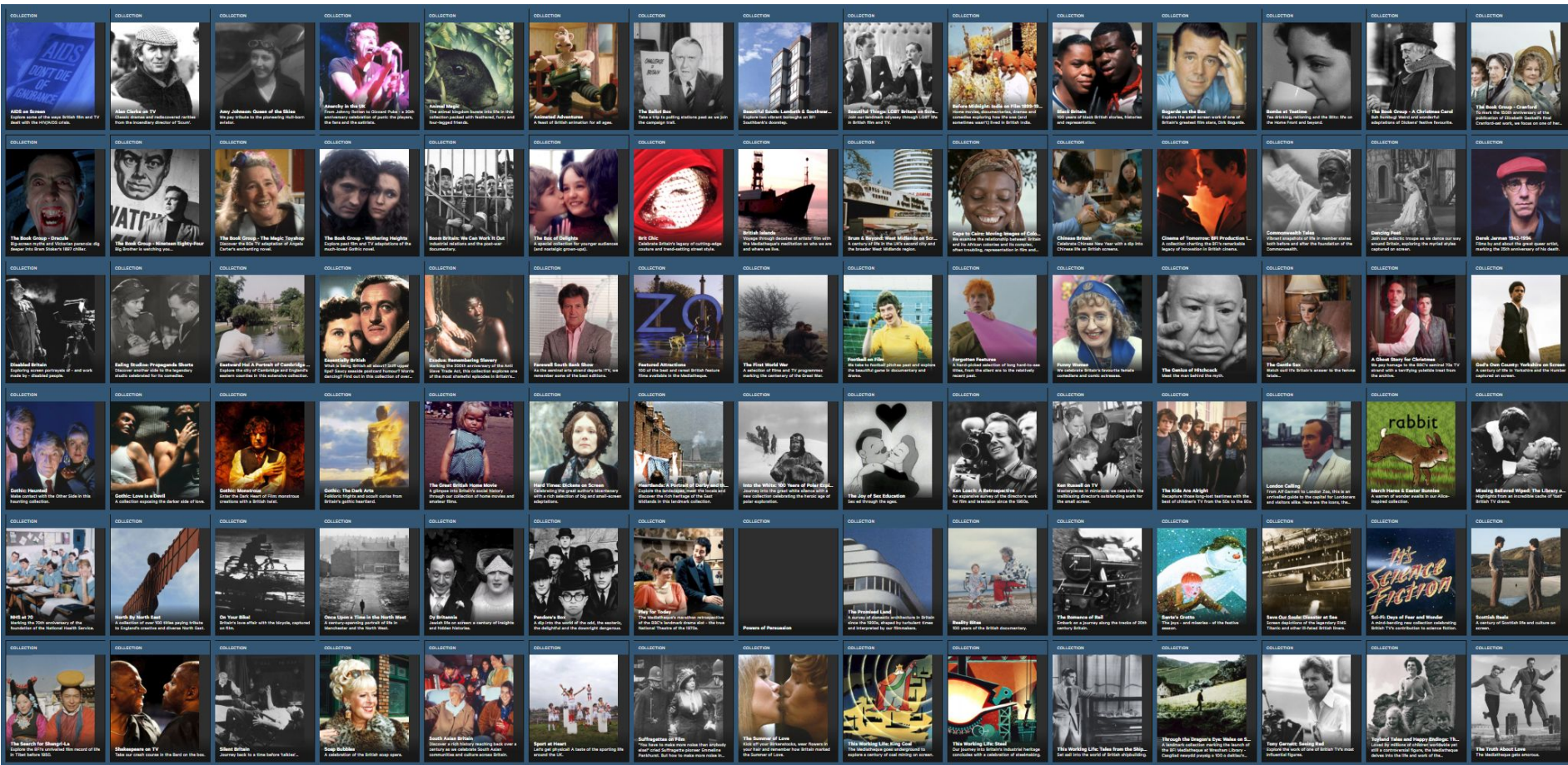
[Read more >](#)

Related links

[Mediatheque Films around the UK](#)

bfi mediatheque - how it works





Filter by

+ Media...

+ Genre...

+ Year...

+ Subject...

+ Cast/credits...

+ Duration...

+ Country...

61,037 Results

Sort by:

Shuffle

A - Z

Z - A





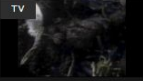
























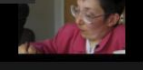


New - Old

Old - New

View:

Grid

List

<div>TV</div> <div></div> <div>Britain's Biggest Mosque</div> <div>Documentary / 2016-05-26</div>	<div>TV</div> <div></div> <div>Peter Lilley MP</div> <div>1993-12-05</div>	<div>TV</div> <div></div> <div>Fortress W1</div> <div>1996-11-13</div>	<div>TV</div> <div></div> <div>News 8.6.94 10.05 P.M. and National We...</div> <div>1994-06-08</div>
<div>TV</div> <div></div> <div>The World within Us</div> <div>1988-04-27</div>	<div>TV</div> <div></div> <div>5 News Tonight[30/04/2014]</div> <div>News / 2014-04-30</div>	<div>TV</div> <div></div> <div>ITV News[14/12/2013]</div> <div>News / 2013-12-14</div>	<div>TV</div> <div></div> <div>NewsTalk Live[12/09/2013]</div> <div>News / 2013-09-12</div>
<div>TV</div> <div></div> <div>Hollyoaks Omnibus[10/11/2013]</div> <div>Soap opera / 2013-11-10</div>	<div>FILM</div> <div></div> <div>Houses in the Town</div> <div>Documentary / 1951 / 18 mins</div>	<div>TV</div> <div></div> <div>ITV News[08/09/2015](1)</div> <div>News / 2015-09-08</div>	<div>TV</div> <div></div> <div>The Godless Society</div> <div>1996-10-08</div>
<div>TV</div> <div></div> <div>News 9.6.93 10.00 P.M. and National We...</div> <div>1993-06-09</div>	<div>TV</div> <div></div> <div>The Agenda[05/11/2012]</div> <div>Current affairs / 2012-11-05</div>	<div>TV</div> <div></div> <div>Liberal Democrat Party Political Broadc...</div> <div>Party political broadcast / 2012-09-24</div>	<div>TV</div> <div></div> <div>A Better Read[29/08/82]</div> <div>1982-08-29</div>
<div>FILM</div> <div></div> <div>Look after Yourself! An Explanatory Tap...</div> <div>Documentary / 1983 / 21 mins Director: Stan Parkinson</div>	<div>TV</div> <div></div> <div>5 News[01/05/2013]</div> <div>News / 2013-05-01</div>	<div>TV</div> <div></div> <div>The Age of Revolutions 1776-1848</div> <div>1985-08-21</div>	<div>TV</div> <div></div> <div>The Chief[05/05/95]</div> <div>1995-05-05</div>
<div>TV</div> <div></div> <div>Autumn</div> <div>Documentary / 2015-06-15</div>	<div>SERIES</div> <div></div> <div>Yus My Dear</div> <div>Comedy</div>	<div>TV</div> <div></div> <div>Britain's Jihadi Brides</div> <div>Documentary / 2015-04-08</div>	<div>TV</div> <div></div> <div>Edgar's Other Woman</div> <div>Sitcom / 1977-03-24</div>
<div>TV</div> <div></div> <div>Channel 4 News[30/08/2014]</div> <div>News / 2014-08-30</div>	<div>TV</div> <div></div> <div>Flight Part 1</div> <div>Children's / 1971-10-18</div>	<div>TV</div> <div></div> <div>Unforgotten[26/01/2017]</div> <div>Police drama / 2017-01-26</div>	<div>TV</div> <div></div> <div>European Figure Skating Championship...</div> <div>Sport / 1972-01-14</div>
<div>TV</div> <div></div> <div>Crime Monthly[17/12/93]</div> <div>1993-12-17</div>	<div>TV</div> <div></div> <div>Access! All Areas</div> <div>1997-08-31</div>	<div>TV</div> <div></div> <div>Ayrshire Part 1</div> <div>1993-06-27</div>	<div>TV</div> <div></div> <div>ITV News at Ten[26/07/2012]</div> <div>News / 2012-07-26</div>

Questions ?

‘Not a question really more of a comment’ ?



stephen mcconnachie
head of data and digital preservation
@mcnatch