Cut: format matrix, test methods and FFmpeg pitfalls

Русский

> **About the interface in this document.** The cut points are found in the mpv player, which a button of the section opens, and are moved into the fields with buttons, or typed as a timecode on the keyboard.

A reference for the CUT section. Everything written here is **measured**, not taken from documentation: each cell of the tables is a real attempt to write with `-c copy` and to read the result back, each sync number is a measurement by frames and samples. The numbers were taken on **FFmpeg 9.0.1**, and taken again on 8.0.1 they matched cell for cell. Which build goes into the distribution is decided by the installer from the version of the NVIDIA driver (on most machines this is the 8.x branch), so both are checked.

The section in detail is in TRIM_EN.md, the decisions are in DECISIONS_EN.md. Here are the facts about trimming; the full set of measurements, including frame rate conversion and the limits of FFmpeg itself, is in MEASUREMENTS_RU.md (Russian).

1. What the section does

CUT copies the streams (`-c:v copy`) and does not re-encode the picture. From this come four properties that define everything else:

- the frame rate is taken from the file as it is: `ffprobe` gives the exact fraction (`24000/1001` for 23.976 and `24/1` for a true 24.000), and nothing is guessed; a camera shoots both;
- the start of a piece lands on a **keyframe** (otherwise there is nothing to start the stream with);
- the end is cut by **the number of frames**, by the exact fraction of the rate;

- the sound is copied as it is, except in two cases (channel selection and PCM in MXF), where it is rewritten sample for sample without a generation loss.

Mode	What stays	Points
Remove the start	everything after IN	one (IN)
Remove the end	everything before OUT	one (OUT)
Keep the middle	the piece between IN and OUT	two
Split	all parts, nothing is thrown away; the joins are on keyframes	N points give N + 1 parts, up to 30 points

The `+` rows in `KEEP` and `CUT OUT` are not new modes but separate runs of the same mode on the same take, each into its own file (`_keep01`, `_cut01`): everything measured here applies to each of them as to a single cut.

2. Container × video codec

Pairs «video + a sound that the container surely accepts», written with `-c copy`, then read back, and the codec compared.

Video	MP4	MOV	MKV	MXF	TS	M2TS
H.264	yes	yes	yes	yes	yes	yes
HEVC	yes	yes	yes	**no**	yes	yes
ProRes	**no**	yes	yes	yes	**no**	**no**
DNxHD/DNxHR	**no**	yes	yes	yes	**no**	**no**
MPEG-2	yes	yes	yes	yes	yes	yes
AV1	yes	**no**	yes	**no**	**no**	**no**
VP9	yes	**no**	yes	**no**	**no**	**no**

What follows from this in practice:

- **ProRes and DNxHR live in MOV, MKV and MXF.** MP4 does not take them: an attempt gives «Could not find tag for codec» and a zero-length file, so such a pair is rejected before the run.
- **HEVC does not go into MXF.** Camera HEVC (Canon, Sony) cannot be moved into MXF without re-encoding; that is a separate operation, not trimming.
- **AV1 and VP9 do not go into MOV.**

3. Container × audio codec

Sound	MP4	MOV	MKV	MXF	TS	M2TS
PCM 16 LE	yes	yes	yes	yes	**no**	**no**
PCM 16 **BE** (`twos`, Canon)	yes	yes	yes	**no**	**no**	**no**
PCM 24 LE	yes	yes	yes	yes	**no**	**no**
PCM 32 float	yes	yes	yes	**no**	**no**	**no**
AAC	yes	yes	yes	**no**	yes	yes
AC-3	yes	yes	yes	**no**	yes	yes
E-AC-3	yes	yes	yes	**no**	yes	yes
MP3	yes	yes	yes	**no**	yes	yes
FLAC	yes	**no**	yes	**no**	**no**	**no**
ALAC	yes	yes	yes	**no**	**no**	**no**
Opus	yes	**no**	yes	**no**	yes	yes

The main points:

- **MXF takes only PCM, and only little-endian.** This matters for Canon: the camera writes `pcm_s16be`, and on output to MXF the sound is rewritten into `pcm_s16le`: the same samples, in the byte order the container requires.
- **MPEG-TS does not take PCM.** Material from an AVCHD card with PCM sound cannot be moved into TS; MOV takes it as it is.
- **MOV does not take FLAC or Opus**, although MP4 takes both.

3.1. A sound file

A sound file does not copy the stream: it decodes it and cuts on the sample, so the container matrix above does not apply to it. What is written where:

Button	Codec	Container	Lossless
As is: PCM	the same PCM	`wav`, `w64`, `aif`, `aiff`, `caf`, `au`, `snd`, `mka`	yes
As is: FLAC, ALAC, WavPack, TTA	the same	its own or `mka`	yes
As is: MP3, AAC, Opus, Vorbis, AC-3, E-AC-3	the same, at the source bitrate	its own or `mka`	one generation
WAV	PCM at the source bit depth; float for a compressed source	`wav`, RF64 past 4 GB	yes
FLAC, ALAC	FLAC, ALAC	`flac`, `m4a`	up to 24 bits
M4A	AAC	`m4a`	no
MP3	LAME	`mp3`	no
Opus	libopus, at most 256 kbps	`opus`	no

MP3, M4A, FLAC, ALAC and MKA keep a cover; WAV and Opus do not.

Packet copying (exact cut turned off) is offered for MP3 in `mp3`, AAC in `m4a` and `m4b`, Vorbis in `ogg` and `oga`, Opus in `opus`, `ogg` and `oga`; for the rest it is not. What the join costs for each codec is in `TRIM_EN.md`; the measurements are in `MEASUREMENTS_RU.md` (Russian), section 12.3.

4. Cut accuracy

4.1 Frames

Checked on eight synthetic sources and one camera source. In every case the number of frames matched the plan exactly.

Source	Request	Frames	Difference
ProRes 25p	3 → 9	150	0
HEVC long-GOP 29.97	12 → 30	600	0 (after topping up by decodable frames)
H.264 all-intra 23.976	3 → 9	144	0
29.97 drop-frame	5 → 15	330	0
59.94 drop-frame	3 → 9	360	0
MPEG-TS (start 1.44 s)	2 → 8	300	0
4 PCM channels	2 → 8	150	0
Canon HEVC 4:2:2 10-bit	3 → 10	166	0
ARRI ProRes 4444 XQ 24.000	2 → 8	144	0
VFR (30 → 17.6)	4 → 16	by time	+1.07 s, reported

4.2 Sound sync

The mark is in both tracks at once: a white frame at the start of each second and a click at the same moment. The deviation is counted **against the source**, not as an absolute value.

Container	Change against the source
MP4	0…2 ms
MOV	0…5 ms
MKV	1…3 ms
MXF	0…5 ms (after the PCM rewrite; before it, 9…18)

On the camera file the sound of the piece is found in the source **word for word**: the piece starts 48 samples (1 ms, a fortieth of a frame) after the cut point.

4.3 Metadata

Checked on a file with rotation, two tracks and project tags:

What	MP4	MOV	MKV
Rotation (`rotation=90`)	yes	yes	yes
Track language (`rus`, `eng`)	yes	yes	yes
Project tags (title, artist)	yes	yes	yes
Camera tags (make, model, brands)	yes	yes	partly
Timecode as a `tmcd` track	yes	yes	**no** (a text tag)

Camera tags are carried by `-movflags use_metadata_tags`; without it the MP4 muxer writes its own `isomiso2mp41` and loses `make`, `model` and a brand like `mp42hvc1CAEP`.

---

5. FFmpeg pitfalls found by measurement

Each one took a separate investigation; none is obvious from the documentation.

No.	Pitfall	Symptom	How it is avoided
1	`-ss` before `-i` resets the time	`-to` after it counts from the new zero	the tail is passed as a duration/frames, not as an absolute point
2	`round(fps)` on fractional rates	4 minutes at 29.97 become 240.27 s	the rate lives as a `Fraction` from `ffprobe` to `ffmpeg`
3	Rounding the end «to the nearest»	a frame standing at 29.9966 is lost when 30.0 is requested	the end is rounded up
4	`-t` when copying	an overshoot of 1–3 frames on a stream with B-frames	the tail is set with `-frames:v`
5	`-frames:v` is exact only with an exact `-ss`	with `-ss` off the keyframe the piece is shorter by the gap	the seek always goes to the computed keyframe
6	HEVC with a B-pyramid	the muxer writes 2–4 frames fewer than requested	the shortfall is measured and the cut is repeated once with a correction

No.	Pitfall	Symptom	How it is avoided
7	`-avoid_negative_ts make_zero`	the video drifts 0.1 s from the sound	not used
8	`start_time` ≠ 0 (MPEG-TS 1.44 s)	all points shift by this amount	the offset is removed from the keyframes and added back on seek
9	`avg_frame_rate` on VFR	the frame count covers the wrong span (12 s → 16 s)	VFR is detected and cut by time, with a warning
10	`-n` on an existing file	«already exists», **exit code 0**	the target is checked before the run
11	PCM in MXF with `-c copy`	384 extra samples at the start (8 ms)	PCM is rewritten sample for sample (LE)
12	`-map_metadata 0` in MP4	camera tags and brands are lost	`-movflags use_metadata_tags`
13	`pan=mono	c0=c1` on mono	writes silence, exit code 0
14	MXF + compressed sound	«Could not write header», code 4294967274	rejected before the run with an explanation
15	Packet ≠ frame	the muxer writes 600 packets, the decoder gives 597: the trailing B-frames refer past the cut boundary	the shortfall is caught by the stream duration and topped up by decodable frames
16	Camera RAW in MXF	`codec_name` is empty, the sizes are zero, «dimensions not set», a zero-length file	rejected before the run: FFmpeg has no X-OCN or ARRIRAW decoder
17	A warning breaks the JSON parsing	`stderr` is mixed into `stdout`, and `json.loads` fails on the line «could not resolve file descriptor strong ref»	probes are read with separate streams

6. How all this is measured

The commands that produced the numbers above. Use the same ones for the next check; otherwise the results cannot be compared.

Frames. Count packets instead of decoding: the result is the same, the time is forty times shorter (0.14 s against 5.7 s on four minutes of 59.94).

```
ffprobe -v error -select_streams v:0 -count_packets -show_entries
stream=nb_read_packets -of csv=p=0 FILE
```

Keyframes around a point, without scanning the whole file:

```
ffprobe -v error -select_streams v:0 -skip_frame nokey -read_intervals
"START%+WINDOW" -show_entries frame=pts_time -of csv=p=0 FILE
```

Sync. You need a source with a mark in both tracks: a white frame and a click at one moment. Read the video through `signalstats` with `-fps_mode passthrough`; otherwise the filter duplicates frames to fit its own clock and shifts the times:

```
ffmpeg -v info -i FILE -map 0:v:0 -vf
"signalstats,metadata=print:key=lavfi.signalstats.YAVG" -fps_mode passthrough -f null
-
```

The sound is read by samples, not through `silencedetect`: the detector works in 20 ms windows, and the values to measure are a few milliseconds.

```
ffmpeg -v error -i FILE -map 0:a:0 -ac 1 -ar 48000 -f s16le -
```

Bit-exact check of copying. Find the sound of the piece inside the sound of the source: if it is there word for word, the sound was copied, and the position gives the exact offset of the cut.

Compatibility. Write one second with `-c copy` into the container and **read it back**. Choose the partner stream to suit the container: MXF does not take AAC, TS does not take PCM, and without this the cell shows the refusal of the partner, not of the codec under test.

Traps of the measurements themselves

Three mistakes that were already made once; they are worth not repeating:

1. **Comparing «the first flash found» with «the first click found».** If one stream opens with a mark and the other does not, the detectors describe different events, and the result is a «107 ms desync» that does not exist.

2. **Measuring sync with FFmpeg detectors.** `blackdetect` works in whole frames, `silencedetect` in windows; together they give an error of 40 ms, that is, more than the value being measured.
3. **Checking compatibility by the exit code.** A file can be written and not read back; TS «accepts» ProRes exactly this way.

6.1. Camera RAW: what the tool cannot do

Two formats that FFmpeg does not open **in any version**; checked on both 8.0.1 and 9.0.1:

File	What is inside	What ffprobe sees
Sony BURANO X-OCN LT	X-OCN in MXF OP1a	a stream without a codec name, `width=0`; the sound `pcm_s24le` is read
ARRI Alexa Mini `.mxf`	ARRIRAW in MXF	the same

An attempt at a plain copy gives «dimensions not set» and a zero-length file, so such sources are rejected before the run with a direct explanation. They can be cut in the vendor's own software (Sony Catalyst, ARRI Reference Tool) or after a transcode.

The same material in **MOV** is cut fully: ARRI ProRes 4444 XQ (2944×2160, 12-bit, 5 PCM tracks) went through frame for frame, together with the timecode 02:00:11:11 → 02:00:13:11.

---

7. What was not checked

The list is open, so that nobody (the author included) decides that the matrix is closed:

- cameras other than Canon HEVC and ARRI ProRes: Sony XAVC, Panasonic, BMPCC, GoPro with telemetry, AVCHD straight from the card;
- files **longer** than an hour (an hour is checked: 50.7 GB, a frame-accurate cut at the 58th minute and in the last seconds; see `MEASUREMENTS_RU.md` (Russian));
- running out of disk space in the middle of writing;
- a source on a network path or held by another program;

- two operations at once;
- 96 kHz and 32-bit float sound in real material;
- sound that runs past the last frame to the packet boundary: after the tail is topped up, it moves forward by 0.1 s and this is reported, but it was not checked by ear;
- sound files from a real recorder (poly-WAV, iXML), packet copying of AC-3, E-AC-3, DTS and AAC outside M4A.