Encoding Pipelines

Русский

Why Archive, Editing, Storage And Delivery Are Separate

These are the four modes of the ENCODE section. They look alike, because all of them build FFmpeg encode commands, but their goals differ.

- ARCHIVE - keep a master with minimal loss or without loss.
- EDITING - prepare a file for an editing program.
- STORAGE - get a compact high-quality file for a library.
- DELIVERY - make a file for viewing, sending or uploading.

If you choose by the codec alone, it is easy to go wrong. Choose by the purpose of the result.

Common Fields

Many encode sections repeat these fields:

- `Source file filter`;
- `Decoder`;
- `Encoder`;
- `Audio`;
- `MP3 / LAME`;
- `Audio bitrate`;
- `Container`;
- `Encoder preset`;
- `Resolution`;
- `Pixel format`;
- `CRF`;

- `CQ/QP`;
- `Encode tuning`;
- `Overwrite`;
- `Test on first file`;
- `Dry run`.

The tooltips over these fields explain their meaning without opening the documentation.

Archive

Purpose: long-term master formats.

Profiles:

- `FFV1 MKV + FLAC`;
- `x264 lossless MKV`;
- `FFV1 + x264 lossless`.

Targets:

- `FFV1`;
- `x264 lossless`.

When to choose it:

- you need to keep a technical master;
- quality matters more than size;
- the file will be the source for future transcodes;
- the material must survive repeated operations without accumulating loss.

By its purpose the container is usually MKV, because it holds FFV1/FLAC and service streams well.

A lossless archive either keeps the data of the source as it is or refuses - there is no quiet loss of precision:

- x264 lossless holds YUV of 8 and 10 bits; a 12-bit, RGB or alpha source is refused with a `[SKIP]` line that says FFV1 keeps it, and the run is not green. A pixel format chosen by hand is the person's choice and is not refused.
- FFV1 holds almost everything, up to 16 bits, RGB and alpha; a format not in its list is refused the same way.

- FLAC holds integer sound; float PCM (and 64-bit) is copied as it is - MKV takes it.

Editing Codecs

Purpose: an intermediate/mezzanine for an NLE.

Profiles:

- `ProRes 422 MOV`;
- `ProRes LT MOV`;
- `ProRes Proxy MOV`;
- `DNxHR HQ MXF`;
- `ProRes + DNxHR`.

Targets:

- `ProRes Proxy`;
- `ProRes LT`;
- `ProRes 422`;
- `ProRes HQ`;
- `DNxHR HQ`;
- `DNxHR HQX`.

When to choose it:

- the source is heavy to edit;
- you need an intra-frame codec;
- you need to hand the material over to editing;
- you need to leave long-GOP/H.264/H.265 for a while.

Recommendations:

- Proxy/LT - for light editing and drafts.
- ProRes 422/HQ - for a high-quality intermediate.
- DNxHR HQ/HQX - for Avid/Resolve/Windows workflows.
- MOV is more often convenient for ProRes, MXF for broadcast/NLE pipelines.

Storage

Purpose: compact high-quality files for a library or for transfer.

Profiles:

- `x264 CRF 14`;
- `x264 1080p CRF 14`;
- `HEVC Main10 CRF 14`;
- `SVT-AV1 CRF 14`.

Targets:

- `x264 / H.264`;
- `x265 / HEVC`;
- `AV1` (SVT-AV1 on the CPU).

When to choose it:

- you need a good file for a personal library;
- you need to reduce the size without strict upload requirements;
- readability on local hardware matters;
- a slower encode for the sake of quality is acceptable.

CRF:

- lower = larger and cleaner;
- higher = smaller and worse;
- `14` - a very high-quality starting point.

Delivery / Publishing

Purpose: files for viewing, uploading and transfer.

Profiles:

- `Review H.264`;
- `Upload H.264`;
- `Upload HEVC 10-bit`;
- `SVT-AV1 delivery`.

CPU/NVENC/AMF/QSV are chosen separately in the `Encoder` block, so that the profile describes the output scenario and not the hardware backend.

Targets:

- `x264 / H.264`;
- `x265 / HEVC`;
- `AV1` (SVT-AV1 on the CPU).

When to choose it:

- the file must play without special tools;
- you need an upload-friendly encode;
- you need to use a hardware encode;
- speed matters.

A hardware encode uses `CQ/QP`; a software encode usually uses `CRF`.

Decode Backend

`Decoder`:

- `Auto`;
- `CUDA`;
- `QuickSync`;
- `AMD/D3D11VA`;
- `dav1d AV1`.

The decode backend reads the input stream. It is not the same as the encode backend. For example, you can decode AV1 with dav1d and encode H.264 with x264 on the CPU.

`dav1d AV1` is not a general accelerator but a forced AV1 decoder (`-c:v libdav1d`). FFmpeg cannot read a source of another codec with it, so the preflight checks the codec of the first file of the Source and stops the operation with a clear message. For a mixed Source choose `CPU/Auto`.

Where The Frames Live: `-hwaccel_output_format`

A hardware decoder gives its frames either to video memory or to system memory. `-hwaccel_output_format` controls this, and the default behavior differs between the three stacks:

- `CUDA`: without an explicit format the frames go to system memory. NVIDIA recommends `-hwaccel cuda -hwaccel_output_format cuda` precisely to avoid this.
- `QuickSync`: by default FFmpeg keeps the frames in video memory and prints the warning `defaulting hwaccel_output_format to qsv ... DEPRECATED`. Any software filter after that fails with `Impossible to convert between the formats`.
- `AMD/D3D11VA`: by default the frames return to system memory; for a full GPU path AMD recommends `-hwaccel_output_format d3d11`.

So the program sets the format itself:

- The full GPU pipeline (`-hwaccel_output_format cuda` / `qsv` / `d3d11`) is turned on when the decoder and the encoder are of one vendor, `Pixel format` is left at `auto`, and there are no software filters: scaling, LUT or HDR-to-SDR. The terminal then shows the line `Decode frames stay in GPU memory ...`.
- If something needs the frames in system memory, for QuickSync the upload format is requested explicitly (`nv12`, and `p010le` for a 10-bit source); otherwise the filters do not work at all. CUDA and D3D11VA do not force a format in this case, so that the bit depth is left to FFmpeg.

Mixed pairs (for example CUDA decode and CPU encode) work fully: the hardware decoder speeds up reading, and the encode runs on the CPU.

Encode Backend

`Encoder`:

- `Auto`;
- `CUDA NVENC`;
- `QuickSync`;
- `AMD AMF`;
- `CPU`.

Recommendations:

- CPU - quality and predictability.
- NVENC/QSV/AMF - speed, especially for delivery/review.

- Auto - good if the profile already knows what it wants.

The encoder applies to all three target families: H.264, HEVC and AV1. The `AV1` target is encoded with SVT-AV1 on the CPU and with `av1_nvenc`, `av1_qsv` or `av1_amf` on the matching hardware. The quality controls are always the backend's own: `CRF` and `SVT-AV1 preset` for the CPU, `CQ/QP` and `NVENC preset` / `QSV preset` / `AMF quality` for hardware encoders.

Before a hardware encode, run `Hardware capabilities`.

Audio Mode

The common field `Audio`:

- `COPY`;
- `AAC override`;
- `MP3`;
- `FLAC`;
- `PCM 16-bit`;
- `PCM 24-bit`;
- `PCM float`;
- `No audio`.

For delivery, usually AAC. For archive, source or FLAC. For editing tasks, source or PCM.

Not every mode fits every container, and this is checked before the run:

- `MXF` takes only PCM 16-bit, PCM 24-bit or no audio;
- `MOV` cannot hold FLAC (`flac only supported in MP4`): for FLAC choose MP4 or MKV; for MOV, PCM fits.

A mode that is not available is grayed out in the form, and when the container is switched to MXF or MOV, an incompatible mode that is already selected moves to `PCM 24-bit` by itself. The backend checks the same, so the CLI and a profile cannot build an impossible pair either.

MP3 / LAME

With `Audio = MP3` the same scale of LAME presets appears as in the AUDIO section; it is one for the whole program:

- `VBR V0` (about 245 kbps) - the default, `-q:a 0`;

- `Insane 320` - `-b:a 320k`;
- `By bitrate` - takes the value of the `Audio bitrate` field.

With MP3 the `Audio bitrate` field is shown only for the third choice, because the first two do not read it. The value `384 kbps` is not available for MP3: the ceiling of MPEG-1 Layer III is 320 kbps, and LAME silently lowers any bigger number to it. The limit lives in the shared contract, so both the GUI and the CLI (`AUDION_MP3_LAME_PRESET`) encode MP3 the same way.

Interlaced Source

A source of two fields (25i, 29.97i, 50i) becomes progressive in EDITING, STORAGE, DELIVERY and COLOR by default. The field `Interlaced source` stands in the block «Advanced»:

- `Progressive x2` - the default: bwdif makes a frame of each field, the rate doubled, the motion whole (50i to 50p);
- `Progressive` - a whole frame of each frame, the rate kept (50i to 25p);
- `Keep fields` - the fields and their order stay, for an interlaced broadcast; x264 codes them as fields.

The folder of the settings says what was done: `x264_crf14_slow_progressive_2x`. A progressive source is not touched. ARCHIVE always keeps the fields - it is a lossless archive.

Pixel Format

Choices:

- auto;
- yuv420p;
- yuv420p10le;
- yuv422p;
- yuv422p10le.

In practice:

- yuv420p - maximum compatibility;
- yuv420p10le - HEVC/AV1 10-bit delivery/storage;
- yuv422p/yuv422p10le - editing/professional scenarios.

Pixel format applies to hardware encoders too: the selected format is translated into the encoder's own format (`yuv420p10le` → `p010le`, for QSV `yuv420p` → `nv12`). If the selected encoder physically cannot produce the requested format, the operation stops with an explicit message instead of silently writing another format:

- H.264 on NVENC/QSV/AMF is 8-bit only. For 10-bit choose HEVC/AV1 or CPU x264.
- 4:2:2 is not available on QSV/AMF; on NVENC only new hardware has it.

`auto` leaves the format negotiation to FFmpeg; this is the default behavior.

Encode Tuning

Flags:

- Grain;
- Film;
- Animation;
- Fast decode;
- Zero latency.

Do not turn tuning on “just in case”. It must match the material or a delivery requirement.

These are `tune` values for x264/x265, so the block is shown only when `Encoder = CPU`. NVENC, QSV and AMF have their own quality scale (`NVENC preset`, `QSV preset`, `AMF quality`) and do not read the tuning list.

Test On First File

`Test on first file` matters especially for encoding. It checks:

- whether the input is found;
- whether the backend works;
- whether the container is compatible;
- whether the audio mode is correct;
- whether the LUT/decode path is broken;
- whether there are enough rights for the Target.

For a large Source this is a required habit.