Remux, FPS И LUT

Remux

Remux меняет контейнер без перекодирования, где это возможно.

Remux теперь работает от цели вывода. Пользователь не выбирает формат источника руками: Source обходится рекурсивно, формат каждого файла и аудиокодеки читаются автоматически. Несовместимые файлы пропускаются с причиной в логе.

Видеоконтейнеры вывода:

```
MP4 MOV MKV MXF
```

Аудиоконтейнеры вывода:

```
M4A ALAC MKA AAC AC3 EAC3 DTS FLAC WAV MP3 OPUS OGG
```

Пайплайн:

```
Video Audio Video&Audio
```

Все три режима являются чистым `copy`: меняется только карта потоков (`-map`) и контейнер, а кодеки не трогаются.

- `Video` оставляет только видеотреки (`-map 0:v? -an`).
- `Audio` отделяет аудиотреки (`-map 0:a? -vn`) в выбранный аудиоконтейнер.
- `Video&Audio` копирует видео и аудио вместе (`-map 0:v? -map 0:a?`).

Same-container copy тоже разрешён, если меняется состав потоков: например `MP4 -> MP4 / Video` создаёт MP4 без аудио, а `MP4 -> MP4 / Video&Audio` делает чистую переупаковку в suffix-папку.

Для audio-only режима контейнер проверяется по реальным аудиокодекам. AAC stereo/mono из MP4/MOV/MKV может уйти в M4A, AC3/EAC3/DTS - в родной файл кодека, FLAC/MP3/OPUS - в свои контейнеры, ALAC - в M4A. MKA остаётся универсальным контейнером для сложных или многодорожечных случаев.

Для video и video+audio режимов проверяется кодек, а не только пара контейнеров. `MOV -> MP4` - разрешённая пара, но она законна для H.264 и невозможна для ProRes: у MP4-муксера нет тега для этого кодека. Такой файл пропускается с причиной вида `MP4 does not support copy of prores`, а не отдаётся FFmpeg на верную ошибку. Практические ограничения:

- ProRes и DNxHR - в MOV, MXF, MKV; в MP4 нельзя;
- HEVC - в MP4, MOV, MKV; в MXF нельзя;
- AV1 и VP9 - в MP4, MKV, WebM; в MOV и MXF нельзя;
- в MXF из аудио идёт только PCM 16/24-bit;
- в режиме `Video&Audio` так же проверяется аудиокодек.

Таблицы говорят, что заведомо работает, и отказ выдаётся только по измеренному кодеку. Кодек, которого в таблицах нет, отдаётся FFmpeg: глухой запрет стоит пользователю рабочей конвертации, а лишний неудачный запуск стоит одной строки в логе. `MKV -> MP4` со звуком - обычная рабочая пара и ничем не ограничена; субтитры и data-потоки до муксера не доходят, их снимает карта потоков (`-sn -dn`).

Если FFmpeg всё же обрывается на записи, пустой файл в OUT не остаётся: нулевой результат удаляется со строкой `[CLEAN]` в логе.

Результат пишется в suffix-папку с рекурсивным зеркалом Source, а не в имя файла. Пример: `Transcoded\Remux\remux_mkv_to_mp4_video_audio\Day1\clip.mp4`.

Когда Выбирать Remux

Выбирай remux, если:

- video/audio streams уже подходят;
- нужно сменить контейнер или состав потоков без encode;
- нужно MP4 faststart;
- нужно переупаковать MKV/MOV/MXF;
- нельзя терять качество на повторном encode.

Не выбирай remux, если нужно:

- поменять codec;
- поменять resolution;
- применить LUT;
- изменить FPS;
- нормализовать аудио;
- сделать downmix.

Для этого нужны encode/audio/FPS/LUT-разделы.

Совместимость Контейнеров

Не все codec/container пары безопасны. GUI показывает только целевые контейнеры, а backend проверяет каждый файл отдельно и пропускает несовместимые случаи.

Практически:

- MP4 хорошо подходит для H.264/H.265/AAC.
- MOV удобен для ProRes/PCM workflows.
- MKV гибкий для archive/storage.
- MXF нужен для broadcast/NLE procedures.
- WEBM с Opus обычно безопаснее вести в OPUS/OGG/MKA для audio-only или MKV для video/video+audio.

FPS / Частота

Раздел `FPS / частота` не просит вручную вводить исходный FPS. Он читается из видеопотока.

Поля:

- `Режим пересчёта`: Varispeed или Conform.
- `Целевая частота`: **16, 18, 23.976, 24, 25, 29.97, 30, 48, 50, 59.94, 60**. Дробные записаны точными дробями (`24000/1001`, `30000/1001`, `60000/1001`); частота, которую не удалось прочитать, отклоняется, а не подменяется на умолчание — раньше любое неизвестное значение молча превращалось в 23.976. 120/144/240 не добавлены намеренно: это не геймерский инструмент. 16 и 18 — для немого кино и любительских 8 мм.

- **Исходная частота — любая**, вплоть до скоростной съёмки: 8000 fps → 24 прошли без потери единого кадра (2000 кадров, ровно 83,333 с).
- `Профиль вывода`: H.264 MP4, H.265 MP4, ProRes MOV/MXF, DNxHR MOV/MXF.
- Для H.264/H.265 доступен AAC bitrate.
- Для ProRes/DNxHR доступны PCM 16/24/32-float в Apple MOV; MOV сохраняет выбранную рабочую частоту вплоть до 176.4/192 кГц.
- MXF принимает PCM 16/24, но muxer используемой сборки FFmpeg финализирует аудио только в 48 кГц; PCM 32-float в MXF отключён.
- Oversampling аудио: COPY, 2x 88.2/96 или 4x 176.4/192 через SoX/libsoxr. Для MXF это внутренняя рабочая частота с обязательным финальным ресемплингом в 48 кГц.
- CLI FPS-мастер предлагает те же шесть профилей, разрядности и `x1/x2/x4`; CLI и GUI строят команду общим `system_core\core\fps_contract.py`.

Varispeed

Varispeed меняет скорость воспроизведения. Длительность меняется, движение остаётся "чистым" без интерполяции кадров.

Используй, когда:

- это осознанный conform по скорости;
- аудио тоже должно растянуться/сжаться;
- нужно перевести material timing.

Conform

Conform меняет интерпретацию/выходную частоту без того же смысла, что varispeed. Выбирай его, когда workflow требует именно conform-выход, а не speed change.

Перед batch-операцией проверь один файл и внимательно посмотри duration.

LUT / Цвет

Раздел `Цвет / LUT` применяет LUT и grading-пресеты.

Основные элементы:

- `LUT` picker;

- кнопки цветовых профилей, которые являются единственным выбором режима окна;
- `Gamma до LUT`;
- output targets;
- encode/decode backend;
- audio/container/resolution/pixel format/CRF/CQ.

LUT Cache

LUT выбирается из `LUTs\`. GUI умеет:

- выбрать LUT;
- добавить LUT files;
- добавить LUT folder;
- открыть LUT folder;
- закрепить часто используемый LUT.

Если поле пустое, проект использует `LUTs\active.cube` или первый найденный LUT-файл, если это предусмотрено текущей логикой.

Цветовые Операции

Доступно:

- `Только LUT`;
- `Предэкспозиция + LUT`;
- `Pregamma + LUT`;
- `HDR to SDR Hable`.

`Только LUT` применяет LUT без pre-gamma helper.

`Предэкспозиция + LUT` и `Pregamma + LUT` используют технически один параметр `eq=gamma=<число>` до LUT, но разные стартовые значения и творческий смысл.

`HDR to SDR Hable` применяет tonemap-style transform для SDR-выхода.

Служебный `script_preset` хранится скрыто и передаётся в backend выбранной профильной кнопкой. Отдельного дублирующего select `Цветовая операция` в GUI нет.

18% Gray Preview

Для pregamma/pre-expose GUI показывает 18% gray preview:

- слева исходный серый;
- справа результат до LUT;
- подпись показывает, станет серый светлее, темнее или останется нейтральным.

Это не замена просмотру итогового видео, но хороший быстрый индикатор направления коррекции.

LUT Path Escaping

Windows-пути к LUT могут содержать:

- диск `C:`;
- пробелы;
- апострофы;
- запятые;
- точки с запятой;
- скобки.

GUI и scripts используют общий helper для FFmpeg filtergraph escaping. Если LUT работает через GUI, он должен работать и через соответствующий wrapper.

Output Targets

Цветовые цели:

- x264 / H.264;
- x265 / HEVC;
- SVT-AV1;
- ProRes 422;
- ProRes LT.

Для проверки цвета лучше начать с короткого H.264 preview, а не сразу с тяжёлого ProRes/AV1 batch.

Рекомендуемая Процедура LUT

1. Выбери Source и OUT.
2. Выбери LUT.
3. Выбери profile.
4. Проверь 18% gray preview, если доступен.
5. Нажми `Команда`.
6. Включи `Тест: первый файл`.
7. Оцени короткий результат.
8. Запускай batch.

Типовые Ошибки

LUT Не Применился

Проверь:

- выбран ли `.cube`;
- существует ли файл;
- нет ли старого/не того pinned LUT;
- что dry-run содержит LUT filter.

Цвет Слишком Тёмный Или Светлый

Проверь `Gamma до LUT`. Для разных LUT стартовое значение может требовать ручной правки.

FPS Результат Не Той Длительности

Сравни Varispeed и Conform. Это не косметические режимы, они по-разному влияют на timing.