

Audion Media Tools - Карта Проекта

Назначение

Audion Media Tools - portable GUI для медиа-операций на Windows. Он собирает FFmpeg, ffprobe, yt-dlp, Deno, 7-Zip, NiceGUI, старые CMD-пресеты и новый service layer в один управляемый инструмент.

Проект решает практические задачи:

- скачать видео или аудио через yt-dlp;
- извлечь, пересэмплировать, нормализовать и упаковать аудио;
- подготовить архивные master-копии;
- сделать монтажные mezzanine-файлы;
- сжать видео для хранения или доставки;
- сменить контейнер без перекодирования;
- пересчитать FPS;
- применить LUT или HDR-to-SDR;
- проверить portable tools, hardware backend и профили.

Основной Принцип

GUI не должен быть набором "магических кнопок". У каждой операции есть:

- профиль;
- явные параметры;
- подсказки над режимами и полями;
- dry-run preview через кнопку **Команда** ;
- лог;
- отчёт;
- preflight.

Если пользователь не уверен, что делает параметр, нормальный путь - задержать курсор на подсказке, затем нажать **Команда**, затем прогнать **Тест: первый файл**.

Рабочие Папки

Основные директории:

Source\	вход по умолчанию
Transcoded\	OUT по умолчанию
Download\	ссылки и результаты yt-dlp
LUTs\	LUT cache и .cube
Scripts\	CLI-пресеты и portable runner
Tools\	FFmpeg, yt-dlp, Deno, 7-Zip
config\	manifest, settings, profiles, path cache
logs\	журналы GUI и операций
report\	batch_report/audio_report/json
workspace\	временная рабочая зона

Source обходится рекурсивно для обычных batch media-операций. **OUT** повторяет относительную структуру подпапок Source.

Пример:

```
Source\Day1\SceneA\clip.mov
Source\Day2\SceneA\clip.mov
```

даёт результат вида:

```
Transcoded\<operation>\<suffix>\Day1\SceneA\clip.ext
Transcoded\<operation>\<suffix>\Day2\SceneA\clip.ext
```

Это защищает от плоского OUT, где одинаковые имена из разных папок могли бы конфликтовать. Если выбран внешний/кастомный OUT, он заменяет **Transcoded**, но структура **<operation>\<suffix>\<зеркало Source>** сохраняется.

GUI-Слои

Левая Область

Слева находится дерево операций и формы параметров. Основные страницы:

- `Скачивание / YouTube` ;
- `Аудиопотоки` ;
- `Архив` ;
- `Монтажные кодеки` ;
- `Хранение` ;
- `Доставка / публикация` ;
- `Упаковка / remix` ;
- `FPS / частота` ;
- `Цвет / LUT` ;
- `Диагностика` ;
- `Установка инструментов` .

Правая Область

Справа находится status/terminal/report control surface:

- статус операции;
- общий progress;
- терминал с stdout/stderr;
- dry-run preview;
- кнопки `Logs` , `Report` , `CONFIG` ;
- expanded terminal;
- ручная командная строка с Shell, CWD, историей и пинами.

Tooltips

Tooltips применяются над кнопками, режимами и параметрами.

Задержки:

- показ: `1000 ms` ;

- сккрытие: `100 ms`.

Технически подсказки хранятся в `data-audion-tooltip` и показываются CSS-слоем. Для крайних segmented-кнопок используется выравнивание по левому или правому краю, чтобы длинные русские тексты не обрезались.

Где подсказки особенно важны:

- `Режим аудио`;
- `Аудиопотоки`;
- `Режим LUFS`;
- `Ресемплер`;
- `Декодировать через`;
- `Кодировать через`;
- `Remux` source/target pair;
- LUT picker;
- cleanup-кнопки;
- terminal/history/cache controls.

Manifest И Service Layer

Главная карта UI лежит в:

```
config\tool_manifest.yaml
```

Там описаны:

- operation groups;
- labels EN/RU;
- descriptions;
- profile presets;
- fields;
- options;
- visibility conditions;
- service handlers.

Основная реализация:

```
system_core\ui_nicgui\app.py
system_core\services\media_service.py
Scripts\Common\script_runner.py
```

`app.py` строит GUI и собирает параметры. `media_service.py` валидирует, строит FFmpeg/yt-dlp команды, пишет отчёты и обновляет progress. `script_runner.py` сохраняет совместимость старых `Scripts\*.cmd` и поддерживает `AUDION_*` overrides.

Профили

Профиль - стартовая конфигурация операции. Он не обязан блокировать ручные изменения: пользователь может выбрать профиль, затем поменять поля ниже.

Пример:

- `M4A 384k LUFS -14` задаёт AAC/M4A, 384k, LUFS target и two-pass loudnorm;
- `Video copy + PCM 24/48 SoX` задаёт режим `Звук в видео`, MOV, PCM 24-bit, 48 kHz и libsoxr;
- `Upload HEVC 10-bit` задаёт доставочный HEVC 10-bit профиль.

Preflight

Перед медиа-операциями проект проверяет:

- Source существует;
- OUT существует или может быть создан;
- входные файлы найдены;
- выбранная LUT существует;
- remux-цель совместима с найденными файлами;
- нужный encoder доступен;
- hardware backend не заблокирован cache-ом.

Preflight не заменяет внимательность пользователя, но отсекает самые дорогие ошибки до запуска FFmpeg.

Отчёты

Обычные media batch-операции пишут:

```
report\<operation>\<timestamp>\batch_report.md  
report\<operation>\<timestamp>\batch_report.json
```

Аудио дополнительно пишет:

```
audio_report.md  
audio_report.json
```

Страница **Report** в GUI показывает последние операции и даёт открыть OUT, report, log, JSON или повторить команду.

Когда Использовать CLI

GUI - основной путь. CLI остаётся полезным, если нужно:

- запускать старые **Scripts*.cmd**;
- проверить portable runner;
- автоматизировать простую серию операций;
- сравнить GUI и script profile behavior.

Для CLI-слоя важны переменные **AUDION_***, например Source/OUT, recursive mode, CRF/CQ, backend и dry-run.

Не Цели Проекта

Проект не пытается:

- заменить DaVinci Resolve/Premiere/NLE;
- стать универсальным FFmpeg command builder;
- автоматически угадывать творческие решения;
- скрывать риск перезаписи;
- постоянно сканировать большие папки через ffprobe.

Сильная сторона Audion Media Tools - безопасные, повторяемые и понятные media-процедуры.