Консольная команда

English · Руководство · Операции · Пресеты · ИИ-агент

Содержание

- Команды
- Запуск операции
- Кадр до и после
- Вывод
- Итог и коды выхода
- Куда ложится результат
- Остановка

`vs_cli.cmd` в корне программы запускает то же, что кнопки окна: пресет, профиль или службу, с теми же полями. Окно — лицо этой команды: всё, что оно запускает, оно запускает через неё, и первая строка каждого прогона в его журнале — готовая команда, которую можно повторить в консоли.

Команда нужна, когда пресет запускает не человек у окна: пакетный файл, другая программа, ИИ-агент, долгая ночь расчётов.

Команды

Команда	Что делает
`describe`	Программа, операции с полями, вариантами и умолчаниями, раскладка окна и что есть у машины для нейросетевых пресетов (`machine`: `rtx`, `mlrt`, `neural`) — одной строкой JSON.
`run`	Запустить операцию.
`frame`	Один кадр видео до и после пресета со значениями — двумя PNG.

Команда	Что делает
`list presets` / `encoders` / `profiles` / `paths`	Строки для чтения: пресеты с умолчаниями, 34 кодера, профили, папки программы.
`map`	Карта программы из её данных. `--out ФАЙЛ` пишет её в файл; `--docs` пишет справочные документы (`OPERATIONS_*` и таблицы этого и других руководств).
`components`	VapourSynth, плагины, MLRT, FFmpeg, драйвер видеокарты; `--latest` добавляет свежие выпуски (GitHub спрашивается не чаще раза в сутки), `--refresh` — спросить сейчас.
`plan --operation install_vapoursynth`	Что скачает установщик: файлы, откуда, размер. Ещё `install_vs_plugins`, `install_ffmpeg`, `install_mlrt`. Ничего не скачивается.
`form`	Сервер форм окна: запрос JSON строкой на вход, ответ строкой на выход.
`version`	Номер программы.

`--root ПАПКА` перед командой называет другую копию программы; `--language ru` или `en` — язык журнала (без него — язык окна).

Запуск операции

```
.\vs_cli.cmd run --operation film_35mm
.\vs_cli.cmd run --operation film_35mm --set stock=500T --set intensity=1.2 --encoder prores_422hq
.\vs_cli.cmd run --operation shadow_denoise_sota --set vs_accel=cpu --source D:\Footage --target D:\Clean
.\vs_cli.cmd run --operation profile_run --set profile=filmic_warm_35mm --source D:\Footage\take.mov
.\vs_cli.cmd run --operation deband_safe --dry-run
.\vs_cli.cmd run --operation doctor
```

Ключ	Что делает
`--operation ID`	Операция: пресет, `profile_run`, `profile_save` или служба.

Ключ	Что делает
`--source ПУТЬ`	Файл или папка — все видео в ней и в её подпапках; без ключа — `input\` программы.
`--target ПУТЬ`	Папка, в которую ложатся результаты; без ключа — `output\`.
`--set ИМЯ=ЗНАЧЕНИЕ`	Поле операции; сколько угодно раз. Поле, которое не названо, берёт своё умолчание.
`--encoder ИМЯ`	Кодер из `list encoders` — на карточки кодера; `--set` карточки ложится поверх.
`--params ФАЙЛ`	Поля JSON-объектом `{"имя": значение}`; `--set` ложится поверх.
`--dry-run`	Команды vspipe и FFmpeg для каждого файла Источника, ничего не запускается.
`--result ФАЙЛ`	Итог в JSON, пишется при любом исходе.
`--cancel-file ФАЙЛ`	Файл, появление которого останавливает прогон.
`--only mov,mxf`	Только эти виды файлов Источника.
`--live`, `--colours`	Для того, кто читает вывод трубой: полоса кадров служебными строками, коды цвета оставлены.

Относительные `--source` и `--target` считаются от корня программы; `--result`, `--params`, `--cancel-file` — от папки того, кто зовёт.

Поля, которые есть у каждого пресета, — обработка, карточки кодера и прогон:

Эти поля стоят на каждой странице: обработка и карточки кодера одни на всё окно - заданные на одной странице, они заданы на всех.

Поле	Вид	По умолчанию	Варианты	Что это
`vs_accel` Обработка	выбор	`auto`	`auto` Авто, `cpu` Процессор, `gpu` Видеокарта	Где работает BM3D. Авто: на видеокарте, когда VapourSynth видит рабочее устройство Vulkan, иначе на процессоре. Остальные фильтры всегда идут на процессоре.

Поле	Вид	По умолчанию	Варианты	Что это
`encode_family` Кодек	выбор	`x264`	`x264` x.264, `x265` x.265, `prores` ProRes, `dnxhr` DNxHR	Чем записывается результат.
`encode_backend` Кодер	выбор	`cpu`	`cpu` Процессор, `nvenc` NVIDIA, `qsv` QuickSync, `amf` AMF	Процессор или аппаратный кодер видеокарты. Аппаратный, не прошедший проверку на этой машине (СЕРВИС - Возможности оборудования), выключен. (видно, когда encode_family - одно из x264, x265)
`encode_quality` Качество	выбор	`14`	`14` 14, `17` 17, `21` 21	14 - почти без потерь, для архива; 17 - потери почти незаметны; 21 - веб, прокси, просмотр. (видно, когда encode_family - одно из x264, x265)
`encode_container` Контейнер	выбор	`mp4`	`mp4` MP4, `mkv` MKV	Файл, в который пишутся x.264 и x.265. (видно, когда encode_family - одно из x264, x265)
`prores_profile` Профиль ProRes	выбор	`lt`	`lt` LT, `422` 422, `hq` HQ	LT достаточно для грейдинга; HQ - для кеинга и повторных проходов. (видно, когда encode_family = prores)
`prores_wrapper` Обёртка	выбор	`mov`	`mov` MOV, `mxf` MXF	MOV для Resolve и Final Cut; MXF для Premiere и Avid. (видно, когда encode_family = prores)
`dnxhr_profile` Профиль DNxHR	выбор	`hq`	`hq` HQ, `hqx` HQX, `lb` LB, `sq` SQ	LB - низкий поток, SQ - стандарт, HQ - высокое качество, HQX - 10 бит. (видно, когда encode_family = dnxhr)
`overwrite` Перезаписать	флажок	`false`		Сделать результат заново там, где он уже есть. Выключено: готовый файл пропускается.

Куда ложится результат. `<Назначение>\<Палитра>\<пресет>_<кодер>\<папки Источника>\<имя>.<расш>`; профиль - `<Назначение>\Profiles\<профиль>_<кодер>\...`. Имя исходника сохраняется. Готовый результат пропускается, если не включено *Перезаписать*.

Аппаратный кодер, который эта машина не прошла (СЕРВИС → «Возможности оборудования»), заменяется процессором, и строка `[NOTE]` об этом говорит. `vs_accel` что-то меняет только у пресетов с BM3D и нейросетевых; у остальных принимается и ничего не делает.

Значение проверяется до прогона. Число вне диапазона, вариант, которого у поля нет, поле, которого нет у операции, `--dry-run` у операции, которая не умеет говорить свои команды, — неверный вызов: код 2 и причина, ничего не запускается. Значение никогда не подвигается в диапазон молча.

JSON-значение из PowerShell. Поле со значением JSON (`profile_values` у `profile_save`) передаётся через `--params ФАЙЛ`: PowerShell портит кавычки JSON по пути в `.cmd`.

Кадр до и после

```
.\vs_cli.cmd frame --operation polaroid --source D:\Footage\take.mov
.\vs_cli.cmd frame --operation mild_denoise --set strength=strong --source
D:\Footage\take.mov --at 0.25 --out D:\Look --max-width 0
```

Ключ	Что делает
`--operation ID`	Пресет или `profile_run` с `--set profile=ИМЯ`.
`--source ПУТЬ`	Видео или папка — её первое видео.
`--at ДОЛЯ`	Где в ролике: доля длины от 0 до 1; без ключа — 0,5 (середина).
`--side both|before|after`	Какие картинки: без ключа — обе.
`--out ПАПКА`	Куда кладутся `before.png` и `after.png`; без ключа — `._runtime\frame\`.
`--max-width N`	Кадр шире N уменьшается до N; без ключа — 1920, 0 — как есть (шумодав судят в 100 %).
`--set`, `--params`	Поля, как у `run`.

Исполняется тот же `.vpy` с теми же значениями, что у прогона; в Назначение ничего не пишется. Ответ — одна строка JSON: `before` и `after` с `path`, `frame`, `frames`, `width`, `height`, `fps`; `seconds`, `params` и `noise` у шумодава. Пресет, который меняет число кадров (RIFE), читается на той же доле длины. Если не выходит — нет движка, нейросетевой пресет без MLRT, нет видео, — `ok` равен `false`, `error` говорит почему, код 1; неверный вызов — код 2.

Вывод

Журнал — обычные строки:

```
[PREFLIGHT] OK engine: installed
[PREFLIGHT] OK Source: 3 video(s) in D:\Footage
[RUN] preset: film_looks/film_35mm, encoder h264_crf14, BM3D auto
[FILE] 1/3 take_01.mov
[PROGRESS] take_01.mov [##########--------------]  42% 105/250 18.40 fps ETA 00:08
Frame: 105/250 (18.40 fps)
[OK] take_01.mp4: 48,210,334 B in 13.6 s
[DONE] 3 done, 0 passed over, 0 failed
```

`[NOISE] mild_denoise: noise 5.34 / 3.44 / 2.75 (8-bit, Y/U/V) -> DFTTest sigma 57.1` — что шумодав измерил и что из этого сделал. `[SKIP]` — результат уже есть; `[FAIL]` — файл не вышел, следующий идёт дальше; `[DRY RUN]`, `[STOPPED]`, `[REFUSED]`, `[NOTE]` говорят то, что говорят. Тот, кто читает вывод трубой, получает ещё служебные строки `::vs::{...}` (`started`, `progress`, `activity`, `finished`); консоль — нет. Журнал каждого прогона — `logs\<время>_<операция>.log`, отчёт — `report\<время>_<операция>\run_report.json` и `.md`, рядом JSON каждого файла.

Итог и коды выхода

Код	Что значит
`0`	Готово.
`1`	Файл не вышел (остальные прошли).
`2`	Неверный вызов или отказ до начала: нет движка, нет видео в Источнике, нейросетевой пресет без MLRT, MLRT без карты RTX, Источник только из RAW камер (движок его не читает). Причину говорят строки `[PREFLIGHT] FAIL`; ничего не пишется.

Код	Что значит
`130`	Остановлено.

`--result ФАЙЛ` пишется при любом исходе:

```
{"operation": "film_35mm", "outcome": "done", "ok": true, "message": "Operation
finished.",
 "processed": 3, "skipped": 0, "failed": 0, "parameters": {"stock": "250D", "...":
"..."}, "seconds": 41.2,
 "log_file": "...", "source": "D:\\Footage", "target": "E:\\...\\output", "data":
{"files": ["..."], "encoder": "h264_crf14"}}
```

`outcome` — `done`, `failed`, `refused` или `stopped`. Файл — `.json` вне Источника.

Куда ложится результат

`<Назначение>\<Палитра>\<пресет>_<кодер>\<папки Источника>\<имя>.<расш>` — `Precision\`, `Film Looks\`, `Retro\`, `Restoration\`; профиль — `<Назначение>\Profiles\<профиль>_<кодер>\…`. Имя исходника сохраняется. Результат пишется как `<имя>.writing-<pid>-<метка>.<расш>` и получает своё имя, только когда дописан. Готовый результат пропускается, если не `--set overwrite=true`, так что второй прогон в то же Назначение даёт `done` с `processed` 0.

Кодеры и карточки, которым каждый соответствует:

Кодер	Карточки
`dnxhr_hq`	`encode_family=dnxhr`, `dnxhr_profile=hq`
`dnxhr_hqx`	`encode_family=dnxhr`, `dnxhr_profile=hqx`
`dnxhr_lb`	`encode_family=dnxhr`, `dnxhr_profile=lb`
`dnxhr_sq`	`encode_family=dnxhr`, `dnxhr_profile=sq`
`h264_amf_q14`	`encode_family=x264`, `encode_backend=amf`, `encode_quality=14`, `encode_container=mp4`
`h264_amf_q17`	`encode_family=x264`, `encode_backend=amf`, `encode_quality=17`, `encode_container=mp4`

Кодер	Карточки
`h264_amf_q21`	`encode_family=x264`, `encode_backend=amf`, `encode_quality=21`, `encode_container=mp4`
`h264_crf14`	`encode_family=x264`, `encode_backend=cpu`, `encode_quality=14`, `encode_container=mp4`
`h264_crf17`	`encode_family=x264`, `encode_backend=cpu`, `encode_quality=17`, `encode_container=mp4`
`h264_crf21`	`encode_family=x264`, `encode_backend=cpu`, `encode_quality=21`, `encode_container=mp4`
`h264_nvenc_q14`	`encode_family=x264`, `encode_backend=nvenc`, `encode_quality=14`, `encode_container=mp4`
`h264_nvenc_q17`	`encode_family=x264`, `encode_backend=nvenc`, `encode_quality=17`, `encode_container=mp4`
`h264_nvenc_q21`	`encode_family=x264`, `encode_backend=nvenc`, `encode_quality=21`, `encode_container=mp4`
`h264_qsv_q14`	`encode_family=x264`, `encode_backend=qsv`, `encode_quality=14`, `encode_container=mp4`
`h264_qsv_q17`	`encode_family=x264`, `encode_backend=qsv`, `encode_quality=17`, `encode_container=mp4`
`h264_qsv_q21`	`encode_family=x264`, `encode_backend=qsv`, `encode_quality=21`, `encode_container=mp4`
`h265_amf_q14`	`encode_family=x265`, `encode_backend=amf`, `encode_quality=14`, `encode_container=mp4`
`h265_amf_q17`	`encode_family=x265`, `encode_backend=amf`, `encode_quality=17`, `encode_container=mp4`
`h265_amf_q21`	`encode_family=x265`, `encode_backend=amf`, `encode_quality=21`, `encode_container=mp4`
`h265_crf14`	`encode_family=x265`, `encode_backend=cpu`, `encode_quality=14`, `encode_container=mp4`

Кодер	Карточки
`h265_crf17`	`encode_family=x265`, `encode_backend=cpu`, `encode_quality=17`, `encode_container=mp4`
`h265_crf21`	`encode_family=x265`, `encode_backend=cpu`, `encode_quality=21`, `encode_container=mp4`
`h265_nvenc_q14`	`encode_family=x265`, `encode_backend=nvenc`, `encode_quality=14`, `encode_container=mp4`
`h265_nvenc_q17`	`encode_family=x265`, `encode_backend=nvenc`, `encode_quality=17`, `encode_container=mp4`
`h265_nvenc_q21`	`encode_family=x265`, `encode_backend=nvenc`, `encode_quality=21`, `encode_container=mp4`
`h265_qsv_q14`	`encode_family=x265`, `encode_backend=qsv`, `encode_quality=14`, `encode_container=mp4`
`h265_qsv_q17`	`encode_family=x265`, `encode_backend=qsv`, `encode_quality=17`, `encode_container=mp4`
`h265_qsv_q21`	`encode_family=x265`, `encode_backend=qsv`, `encode_quality=21`, `encode_container=mp4`
`prores_422`	`encode_family=prores`, `prores_profile=422`, `prores_wrapper=mov`
`prores_422_mxf`	`encode_family=prores`, `prores_profile=422`, `prores_wrapper=mxf`
`prores_422hq`	`encode_family=prores`, `prores_profile=hq`, `prores_wrapper=mov`
`prores_422hq_mxf`	`encode_family=prores`, `prores_profile=hq`, `prores_wrapper=mxf`
`prores_lt`	`encode_family=prores`, `prores_profile=lt`, `prores_wrapper=mov`
`prores_lt_mxf`	`encode_family=prores`, `prores_profile=lt`, `prores_wrapper=mxf`

Остановка

`--cancel-file ФАЙЛ`: когда файл появляется, движок гасит vspipe и FFmpeg, убирает недописанный файл и оставляет готовые; итог — `stopped`, код 130. СТОП окна — этот файл.