Audion DevOps Tools

English · Руководство

Содержание

- Зачем это сделано
- Принципы
- Что внутри
- Дальше
- Техническая часть
  - Запуск
  - Обслуживание
  - Проверка
  - Устройство
  - Рабочие папки
  - Основания в документации Microsoft

Портативная оболочка для точных операций над Windows: подсистема Linux, виртуализация, драйверы и железо, диски, сеть, ассоциации файлов, ключи и сертификаты.

Зачем это сделано

Есть класс задач, для которых нет хорошего инструмента. Не «ускорить Windows» и не «вырезать лишнее» — а сделать одну точную вещь и знать, что именно ты сделал: поставить дистрибутив Linux из файла, переключить машину между Hyper-V и сторонним гипервизором, запретить обновлению подменять драйвер видеокарты, снять слепок ассоциаций до и после установки программы, выгрузить сертификат вместе с закрытым ключом.

Каждая такая операция делается разными средствами: где-то `wsl.exe`, где-то `bcdedit`, где-то политика в реестре, где-то команды PowerShell. Все они документированы, но разбросаны, и половина требует помнить, что откатить, если пойдёт не так.

Программа собирает их в одном месте — с явными параметрами, метками риска, копией перед изменением и живым журналом.

Это не чистильщик системы и не сборник твиков. Ассоциации не подделываются обходом защиты Windows, ключи реестра не правятся напрямую там, где есть документированный механизм. Программа оборачивает штатные средства администрирования, а не воюет с системой.

Принципы

Оборачиваем документированное. Подсистема Linux — через `wsl.exe`. Профили Wi-Fi — через `netsh wlan`. Ассоциации — через механизм развёртывания и политику для всей машины. Сертификаты — через средства PowerShell поверх системных хранилищ. Хэши пользовательского выбора и защита от их подмены не обходятся вручную.

Копия до изменения. Загрузчик, ассоциации, файл `hosts`, состояние сети, хранилище драйверов — всё, что меняется, сначала снимается в `backup\`.

Метка риска у каждой команды. Мягкие тона кнопок различают щадящее, обычное и сильное действие; опасное требует подтверждения. Парные сценарии — резерв и восстановление, выгрузка и загрузка, запрет и снятие запрета — показаны рамками, чтобы обратный ход был на виду.

Интерфейс намеренно технический. Кнопки короткие, подписи компактные, а полный контекст — что именно произойдёт, чем рискуете, как откатить — раскрыт в подсказке. Место экономится на подписи, а не на объяснении.

Неподдерживаемая редакция — не цель. Домашние выпуски Windows не гарантируют работу групповых политик; программа показывает редакцию и по умолчанию запрещает применение там, где оно не сработает.

Ничего не тянется из соседних папок. Все средства лежат внутри проекта. Исторические папки читаются только для сравнения или восстановления — и только по прямой просьбе.

Что внутри

раздел	о чём
Подсистема Linux	возможности и обновление, установка из сети и из файла, состояние, резерв, клонирование, перенос, удаление, регистрация дисков

раздел	о чём
Виртуализация	состояние и диагностика без изменений, режим Hyper-V или сторонних машин, совместимость, переключатели с копией загрузчика и предупреждением о перезагрузке
Сеть	диагностика, резерв и восстановление состояния, прокси, адаптеры, вход в общие папки, профили Wi-Fi, быстрые режимы
Узлы и Bitrix	подмена адреса на локальный, определение точки, состояние службы имён и портов, побитовое возвращение файла `hosts` из копии
Ассоциации	слепок и сверка умолчаний, защита политикой, встроенные приложения Microsoft — убрать, вернуть, держать удалёнными, запрет переустановки, отслеживание смены
Железо и драйверы	запрет драйверов из обновлений, ограничения для видеокарты, резерв хранилища драйверов, звук по HDMI и DisplayPort, опись дисков, среда восстановления, мастер SSD
Ключи	выгрузка и загрузка ключей SSH и сертификатов, сборка всех доступов в одну папку с описью, переезд на новую машину
Обслуживание	резерв памяти командных помощников, очистка кэшей, ярлыки, встроенный поиск по файлам

Дальше

- Руководство — работа по разделам, порядок действий для опасных подсистем.

Техническая часть

Запуск

```
Start.exe
```

`Start.exe` один раз спрашивает права администратора — они нужны развёртыванию, файлу `hosts`, сетевым адаптерам, работе с дисками и настройке подсистемы Linux. Ответ «нет» открывает окно без прав: это видно по значку прав в строке машины, щелчок по нему перезапускает окно с правами, а операция, которой права нужны, спросит их у Windows сама.

Без вопроса о правах:

```
set AUDION_GUI_NO_ELEVATE=1
Start.exe
```

Каждая кнопка окна — операция консольной команды, и её можно запустить из командной строки:

```
devops_cli.cmd describe                          разделы, операции и поля (JSON)
devops_cli.cmd run --operation dns_status        операция, вывод — в консоль
devops_cli.cmd check --operation <операция>      можно ли запустить и о чём спросить
devops_cli.cmd options --operation <операция> --field <поле>
devops_cli.cmd version
```

Отдельные входы для одного раздела:

```
cli\launcher_wsl.cmd
cli\launcher_bitrix.cmd
cli\launcher_default_apps.cmd
cli\launcher_association_defense.cmd
cli\launcher_hardware.cmd
```

Они идут через тот же слой описаний и служб, что и окно, — поведение совпадает.

Обслуживание

```
init_folders.cmd                 создать недостающие рабочие папки
cleanup_project.cmd              деликатная очистка
cleanup_project.cmd /DRYRUN /Y   показать план, ничего не удаляя
```

Очистка оставляет скрипты, настройки, документацию, лицензии, собранное окно и сами папки. Удаляет порождённое и скачанное: рантайм, склад пакетов, сборки, загрузки, журналы, отчёты, содержимое рабочих папок и кэши.

Проверка

```
install\verify_portable_env.cmd              среда, консольная команда, собранное
окно
runtime\python.exe -m pytest -q              тесты
runtime\python.exe system_core\doctor.py     только среда
```

Окно собирается из `Source\` командой `install\Build-App.cmd`; нужен .NET 10 SDK.

Устройство

```
Start.exe                       запуск окна
App\                            окно и PowerShell программы (AudionPs.exe)
Source\                         исходники окна
devops_cli.cmd                  консольная команда
config\tool_manifest.yaml       дерево операций, поля, метки риска
config\window_layout.yaml       разделы и страницы окна
system_core\cli.py              консольная команда: операции манифеста
system_core\core\jobs.py        запуск операций, разбор вывода
system_core\services\           системные действия
system_core\cli_operation.py    операции для меню cli\
tools\                          средства проекта
backup\                         слепки и данные для отката
logs\                           журналы операций
```

Граница ответственности. Окно не содержит системной логики: оно собирает параметры, показывает риск и подтверждение и запускает операцию консольной командой. Системные действия живут в службах, модулях и скриптах проекта, а описания операций — в `tool_manifest.yaml`.

Рабочие папки

Над терминалом две строки. **ИСТОЧНИК** — откуда читают импорты: ключи, сертификаты, профили Wi-Fi, копии; пока не выбрана другая — папка `input`. **НАЗНАЧЕНИЕ** — куда ложатся выгрузки, резервные копии и отчёты; пока не выбрана другая — папка `output`. Обе строки помнят прежние и закреплённые папки.

Основания в документации Microsoft

- Политика умолчаний приложений
- Ассоциации через развёртывание

- Профили Wi-Fi
- Команды подсистемы Linux

Выгрузки ключей SSH, сертификатов с закрытой частью и паролей Wi-Fi — сведения чувствительные: полученные архивы хранятся как секреты.