Проверки перед выпуском

Содержание

- Разделы окна
- Среда
- Тесты
- Окно
- Терминал
- Обслуживание
- Консольные меню
- Кодировка CMD
- Скриншоты

Разделы окна

Окно — шесть разделов: СЕТЬ, WSL И VM, ДРАЙВЕРЫ, АССОЦИАЦИИ, ЭКСПОРТ И ИМПОРТ, СИСТЕМА. Каждая кнопка окна — операция консольной команды `devops_cli.cmd`; что проверено в окне, то же даёт `devops_cli.cmd run --operation <id>`.

После изменений проверить:

- у каждого раздела есть подстрочник и внутренние страницы; пустых страниц нет;
- кнопки не подсвечиваются тревожным цветом только из-за `kind: dangerous`; опасная операция перед запуском спрашивает подтверждение и говорит, что изменит;
- СЕТЬ / ПОДКЛЮЧЕНИЕ: адаптеры в две колонки, физические и виртуальные; Wi-Fi-адаптер — на странице WI-FI; вход в общие папки (SMB) хранит только пары компьютер/пользователь, пароль вводится во внешней консоли `net use` и не появляется ни в окне, ни в журнале;
- СЕТЬ / WI-FI: профиль, к которому машина подключена сейчас, стоит в списке первым с подписью «— подключён сейчас» и сам не выбирается; импорт XML стоит рядом с экспортом, файл или папка, область `current` / `all`;

- СЕТЬ / HOSTS: `Определить точку` не меняет `hosts`, принимает только локальный или частный адрес и подставляет `ip_address` / `bitrix_ports`; имя узла и адрес по умолчанию пусты, порт `443`; состояние показывает записи `hosts`, разрешение имени с ними и без них, DNS-серверы, автоскан портов и итоговую проверку TCP; снятие подмены возвращает `hosts` побитово из `backup=hosts_prepatch_....bak` в служебном комментарии, а не пересобирает файл построчно;
- WSL И VM / WSL: состояние, возможности, обновление, список из сети, установка из сети и из файла (`.wsl`, tar, VHD/VHDX) в выбранную папку; выбора `Audion_WSL_Block_E/S` нет;
- WSL И VM / HYPER-V: состояние виртуализации и оптимизации — только чтение (hypervisorlaunchtype, возможности, VBS/Core Isolation, схема питания, исключения Defender, `.wslconfig`, размещение VHDX WSL); переключения спрашивают подтверждение, кладут копию BCD в `backup\virtualization` и прямо говорят о перезагрузке;
- АССОЦИАЦИИ / MICROSOFT: приложения расписаны по тому, что они перехватывают; галочками отмечены все, кто берёт картинки, музыку и видео, остальные — без галочек, но с возможностью удалить; Windows Media Player Legacy удаляется как компонент Windows;
- АССОЦИАЦИИ / ПОЛИТИКА: действия разделены на те, что меняют Windows, и безопасные (просмотр, открытие, выгрузка); проверка защиты показывает редакцию Windows и поддержку `DefaultAssociationsConfiguration`, на Home/Core включение по умолчанию отказывает; `UserChoice` не правится;
- ЭКСПОРТ И ИМПОРТ / СЕРТИФИКАТЫ: хранилища в две колонки, название — над разделителем; выгрузка PFX помечена `risk_level: secret_export`, импорт PFX и CA меняет хранилище сертификатов;
- ЭКСПОРТ И ИМПОРТ / AI: чаты Code видны без Desktop, Codex читает `.jsonl.zst`, Grok и Desktop-профили имеют явные состояния; фильтры сохраняют скрытые отметки; импорт и экспорт имеют независимый выбор; переключатели ЭКСПОРТ | ИМПОРТ и кнопка действия не прокручиваются; фильтры стоят в три строки; поиск чатов и его сброс видны над прокруткой списка, пустой результат объяснён; откат и память — в ИМПОРТЕ; категории не меняют выбор чатов и авторизации; оба переноса начинаются с плана, новый комплект имеет schema 3; пользовательский ИСТОЧНИК соблюдается, конфликты сохраняются, ошибочный импорт откатывается по журналу;
- СИСТЕМА / ОЧИСТКА: очистка Codex и Python с аудитом, пробным прогоном и полной очисткой; кэши — Windows Temp старше суток (нужны права администратора), сессии Claude Code в `%TEMP%\claude`, простоявшие сутки, кэш Premiere — и там, куда Premiere его перенёс;
- длинные подписи кнопок переносятся по словам, ряды кнопок при переносе не слипаются;

- полный текст операции — во всплывающей подсказке и в подтверждении, на странице — короткий;
- `wsl_online_distro_options()` даёт чистые имена дистрибутивов (`Ubuntu`, `Ubuntu-24.04`, `Debian`), а `wsl.exe --list --online` в терминале читается без кракозябр, даже когда WSL ещё не установлен.

Быстрая проверка списка дистрибутивов:

```
runtime\python.exe -c "from pathlib import Path; import sys; sys.path.insert(0,
str(Path.cwd())); import system_core.services.devops_tools as d;
print(d.wsl_online_distro_options(Path.cwd())[:8])"
```

Среда

```
runtime\python.exe system_core\doctor.py
```

Ожидается `[RESULT] Required environment looks good.`: пакет движка (`pyyaml`), PowerShell для скриптов, каждая операция манифеста находит свою службу, `.cmd` — UTF-8 без BOM и CRLF, `.sh` — LF.

PowerShell скриптов ищется в таком порядке, и раздел `[PowerShell]` его показывает:

1. `App\AudionPs.exe` — PowerShell 7 программы;
2. `pwsh.exe` из `PATH`;
3. встроенный `powershell.exe`.

Операции AppX и AppLocker намеренно идут встроенным Windows PowerShell 5.1.

Тесты

```
runtime\python.exe -m pytest -q
```

Ожидается: всё проходит. Две проверки очистки создают символическую ссылку; без режима разработчика Windows они пропускаются с явным `WinError 1314`. Тесты AI CLI Backup идут на `App\AudionPs.exe` — без собранного окна они пропускаются.

Окно

Сборка окна и его PowerShell — `install\Build-App.cmd` (нужен .NET 10 SDK). Сборка прогоняет тесты модели консольной команды и словаря окна, собирает окно во временную папку, проверяет его там (`--self-test --just-built`), сверяет, что формы отдают консольной команде, с записью параметров (`tests\test_window_values.py`) и только потом заменяет `App\`. Неудачная сборка оставляет рабочее окно на месте. После сборки убрать `Source\bin` и `Source\obj` (это делает и `install\Clean-Install-Cache.cmd`).

Проверка собранного окна без экрана:

```
install\verify_portable_env.cmd
```

Ожидается: среда в порядке, `devops_cli.cmd` читает манифест, `App\AudionPs.exe` запускается, окно проходит свою проверку и называет число пройденных. То же для одного окна — пункт `[3]` в `launcher_project.cmd`.

Запуск:

```
Start.exe
```

Ожидается один запрос UAC, затем окно с правами администратора. Ответ «нет» открывает окно без прав: значок окна это показывает, а операция, которой права нужны, спросит их у Windows сама. Без вопроса:

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

Второй запуск не открывает второе окно, а выводит вперёд первое.

Терминал

- вывод операции идёт живьём в цветах служб, журнал на диске — без кодов цвета;
- над терминалом: ЛОГИ, ОТЧЁТЫ, КОНФИГ, ПАПКИ; копирование всего текста, очистка, развёртывание на всё окно;
- под терминалом — операция, время и итог; точка активности пульсирует, пока операция идёт;

- строка команды: оболочка PS (`App\AudionPs.exe`) или CMD, рабочая папка, история и закреплённые команды; команда во время операции встаёт в очередь;
- дочерние программы не открывают своих консольных окон: их вывод идёт в терминал и журнал.

Обслуживание

```
init_folders.cmd
cleanup_project.cmd /DRYRUN /Y
cleanup_project.cmd /BACKUP /DRYRUN
```

Ожидается: `init_folders` выходит с кодом 0; очистка в пробном режиме показывает порождённое и скачанное (`runtime`, `wheelhouse`, `release`, `install\download`, `ripgrep`, `fzf.exe`, `Source\bin` и `obj`, журналы, отчёты, рабочие папки, снимки в `backup`) и ничего не удаляет; `App\` и исходники окна не трогаются. `/BACKUP /DRYRUN` показывает только снимки, при настоящем запуске спрашивает свой `Y/N/Q`.

Консольные меню

Меню без тяжёлых операций:

```
"0" | cmd /d /c launcher_project.cmd
"0" | cmd /d /c launcher_project_ru.cmd
"0" | cmd /d /c cli\launcher_wsl.cmd
"0" | cmd /d /c cli\launcher_bitrix.cmd
"0" | cmd /d /c cli\launcher_default_apps.cmd
"0" | cmd /d /c cli\launcher_association_defense.cmd
"0" | cmd /d /c cli\launcher_hardware.cmd
```

Ожидается: меню открывается, кириллица читается, выход по `0` без ошибки. Меню WSL, Bitrix и ассоциаций зовут операции манифеста через `system_core\cli_operation.py` и не повторяют логику служб.

Руками, потому что эти входы передают управление `Nuke.cmd` с запросом UAC:

```
cli\launcher_codex_nuke.cmd
cli\launcher_python_nuke.cmd
```

Ожидается меню встроенного `tools\...\Nuke.cmd`; разрушительные режимы по-прежнему требуют набрать подтверждение.

Кодировка CMD

Все `.cmd` — UTF-8 без BOM и строго CRLF. `.cmd` с одними LF перед выпуском исправить.

```
install\Check-CmdEncoding.cmd -Fix
```

Та же проверка встроена в сборку среды, установку без сети и `install\verify_portable_env.cmd`.

Скриншоты

После заметных правок окна — скриншоты разделов в `report\`: каждый раздел с открытой страницей, форма с раскрытыми полями, терминал и итог после короткого успешного запуска. Смысл — поймать настоящие сбои раскладки: не поместившиеся подписи, слипшиеся ряды, лишнюю прокрутку, невидимый итог.