

Smoke-Test Checklist

Содержание

- [Runtime Imports](#)
- [Syntax](#)
- [Doctor](#)
- [CMD Encoding](#)
- [SH LF](#)
- [Cleanup / Init Check](#)
- [Project Upgrade Check](#)
- [NiceGUI Smoke](#)
- [Server Check](#)
- [Window Check](#)
- [Picker Check](#)
- [Layout Check](#)
- [Visual Smoke Screenshots](#)
- [Nested Menu And Fields Check](#)
- [PowerShell / CLI Window Check](#)
- [Long Operation / Navigation Check](#)
- [System Operations / Preflight Check](#)
- [NiceGUI ProcessPool Fallback](#)

Runtime Imports

```
runtime\python.exe -c "import nicegui, webview, yaml, rich; print('OK GUI imports')"
```

Если проекта ещё нет в portable runtime, используйте системный Python 3.12.

Syntax

```
runtime\python.exe -m py_compile system_core\ui_nicegui\app.py
system_core\ui_nicegui>window.py
```

Doctor

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

Если проекта ещё нет в portable runtime, используйте системный Python 3.12:

```
python system_core\doctor.py
```

CMD Encoding

Все `.cmd` должны быть:

```
UTF-8 without BOM + CRLF
```

ВАЖНО: ВСЕ `.CMD` ФАЙЛЫ ОБЯЗАТЕЛЬНО UTF-8 БЕЗ BOM И СТРОГО CRLF.

LF-only `.cmd` перед релизом исправить. После любого patch/edit проверять `BOM=False` и `LoneLF=0`.

Штатная проверка/ремонт для проекта:

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

Portable build не чинит `.cmd` во время выполнения: активный `.cmd` нельзя переписывать изнутри. Offline install, verify и release archive gate используют check-only проверку. Если она упала, сначала запустите отдельный пункт CMD ENCODING CHECK, затем повторите нужную операцию.

SH LF

Все `.sh` должны быть:

UTF-8 without BOM + LF

Штатная проверка/ремонт:

```
python system_core\core\sh_lf.py --fix  
python system_core\doctor.py
```

Это важно для WSL/bootstrap сценариев: CRLF в Linux shell files считается regression.

Cleanup / Init Check

- `install\init_folders.cmd` создаёт все управляемые рабочие папки проекта.
- `install\init_folders.cmd` не создаёт `.gitkeep` в `input\`, `output\` и вложенных `output\...`; эти папки должны быть пустыми для пользователя.
- `install\init_folders.cmd` не создаёт пустые файлы ключей, токенов или секретов.
- `cleanup_project.cmd` повторяет рабочие зоны проекта и чистит всё, что устанавливается или создаётся при использовании: runtime, wheelhouse, download/cache/release-артефакты, logs/report/workspace/input/output, caches и временные файлы.
- `cleanup_project.cmd` после чистки вызывает `install\init_folders.cmd`.
- `cleanup_project.cmd` не удаляет `config`, `docs`, `GitHub`, корневой `licenses\` целиком, launchers, install-скрипты, RU/EN языковые входы и исходный код вне явных cleanup-targets.
- После установленного окружения `cleanup_project.cmd /YES` возвращает шаблон к LEAN source-only состоянию: `runtime`, `wheelhouse`, `install\download` и рабочие папки пустые; `input\` и `output\` без `.gitkeep`, служебные placeholders только во внутренних зонах.

Project Upgrade Check

При портировании GUI проверьте архитектуру самого проекта:

- основные CLI launchers/скрипты представлены в manifest/tree и не потеряны;
- дубли presets сокращены до preset + controls там, где это безопасно;
- GUI-поля идут в `context.operation.parameters` или CLI args как runtime override, а не переписывают постоянный YAML;

- Doctor/capability detection объясняет missing binaries/modules/plugins/codecs/hardware;
- STATUS/probe/inventory показывает входные данные до запуска;
- interactive `pause`, `set /P`, stdin prompts и FZF-внутри-запуска не используются как основной GUI-путь;
- каждый запуск логирует operation id, cwd, итоговую команду/конфигурацию, redacted параметры, stdout/stderr и exit code;
- smoke matrix включает Doctor, GUI smoke, HTTP smoke, dry-run, короткий реальный run, STATUS/probe, cleanup cancellation и gitignore/heavy-folder sanity.

NiceGUI Smoke

```
runtime\python.exe system_core\ui_nicegui\app.py --smoke
```

Ожидается строка `OK nicegui shell`.

Эта проверка также вызывается из:

```
install\verify_portable_env.cmd
```

если `system_core\ui_nicegui\app.py` существует.

Server Check

Запустите на свободном тестовом порту:

```
runtime\python.exe system_core\ui_nicegui\app.py --host 127.0.0.1 --port 8099 --no-browser
```

Проверьте `http://127.0.0.1:8099/`.

Window Check

```
launcher_gui.cmd
```

Ожидается отдельное desktop-окно ruwebview. Браузер не должен открываться сам.

Ожидаемый стартовый размер окна: **1600x900**. Минимальный размер около **1180x720**.

Picker Check

Проверьте:

- **Добавить файл...** открывает Windows file picker и назначает один файл источником;
- picker строки источника выбирает папку без скрытой копии;
- источник и назначение доходят до backend как активные пути;
- **Сбросить** очищает незакреплённый кэш и возвращает проектные **input/output**;
- **Удалить** запрашивает подтверждение для внешнего источника и блокирует корни;
- повторяющиеся имена получают уникальные suffix;
- попытка добавить сам **input** не проходит.

Layout Check

На WUXGA **1920x1200** с Windows scale 150%:

- окно можно сжать до ноутбучного логического профиля без развала двух колонок;
- команды остаются слева, статус и терминал справа;
- терминал не уезжает под список команд;
- нет раннего перехода в вертикальную ленту.

На FullHD/4K:

- кнопки в одну строку;
- на экране команды **Назад** находится слева, основная кнопка запуска справа в той же строке;
- запуск не продублирован и не закопан ниже параметров;
- большие формы разбиты на логические секции с заголовками;
- родственные поля находятся в одном блоке, без пустых широких секций ради одного маленького control;
- терминал справа занимает большую часть высоты;
- status/progress не раздувают layout;
- **Cancel** виден только во время операции;

- **Logs** рядом с терминалом.
- под терминалом есть постоянный итоговый индикатор: серый в ожидании, синий во время выполнения, зеленый после успешного завершения, красный после ошибки.
- левая колонка не растягивается бессмысленно;
- терминал остается читаемым;
- нет пустых карточек ради декора.

Visual Smoke Screenshots

После заметных GUI/layout-правок сохраняйте smoke-скриншоты в проектной зоне отчётов, например:

```
report\gui_smoke_screenshots\
```

Минимальный набор:

- корневое меню с рабочими папками и терминалом;
- один главный экран команды с основными полями;
- тот же или похожий экран с раскрытым **Дополнительно**, если менялись редкие поля;
- экран TASK/длинной формы, если проект использует nested **operation_groups**;
- терминал и нижний статус после короткого успешного запуска.

Цель не в красивом альбоме, а в том, чтобы портирование ловило реальные UI-регрессии: не помещающиеся dropdown, слишком яркие рамки, дублирующие controls, ранний перенос в одну колонку, лишний scroll и невидимый финальный статус.

Nested Menu And Fields Check

Если проект использует **operation_groups** и **fields**:

- переходы по вложенному меню меняют только левую область команд;
- правый терминал, статус, прогресс и кнопки папок остаются на месте;
- leaf-команда сначала показывает финальный экран команды с **Назад** слева и запуском справа;
- **text**, **number**, **select**, **checkbox** и **checkboxes** отображаются корректно;

- `radio`, `profile_buttons` / `preset_buttons` отображаются корректно, если используются;
- `fields` сгруппированы по `group` / `ui_group` / `section` или по понятному auto-group fallback;
- `options_source` загружает динамические варианты, кэшируется и обновляется кнопкой `Обновить список`;
- `checkboxes` переносится на несколько строк и не ломает ширину окна;
- `min_selected: 1` блокирует запуск, если пользователь снял все флажки;
- выбранные значения видны в `context.operation.parameters` или в итоговой CLI-команде;
- пустые строки сохраняются, если они означают auto/default.
- схожие выборы на форме стоят рядом и читаются как один блок;
- маленький фиксированный single-choice не спрятан в dropdown без причины;
- нет двух почти одинаковых кнопок запуска для одного пользовательского результата;
- нет дублирующих controls избранного для одного и того же списка;
- нет оторванных команд без объекта: `В избранное`, `Сохранить`, `Проверить`, `Удалить` должны быть переименованы или визуальнo привязаны к единственному полю;
- второстепенные параметры не отодвигают основной запуск ниже видимой области.
- редкие параметры можно свернуть в `Дополнительно`, а состояние блока запоминается, если проект это поддерживает.

PowerShell / CLI Window Check

Если проект вызывает `pwsh.exe`, `powershell.exe`, Office COM helpers, ffmpeg или другой дочерний CLI:

- запуск через GUI не должен создавать всплывающие консольные окна на каждый файл;
- `-WindowStyle Hidden` не считается достаточной защитой;
- проверьте, что subprocess использует `run_process()` или `STARTUPINFO/SW_HIDE` и `CREATE_NO_WINDOW`;
- проверьте, что Windows system tools декодируются через byte-stream fallback, а не через `text=True, encoding="utf-8"`;
- проверьте, что ANSI-цвета в GUI-терминале рендерятся цветом, а не видимыми `\x1b[36m` / `[0m`;
- вывод дочерней команды должен идти в правый GUI-терминал или лог, а не в отдельное пользовательское CLI-окно;

- после завершения операции зеленый индикатор под терминалом остается видимым, даже если окно было неактивно.

Long Operation / Navigation Check

Проверьте жизненный цикл NiceGUI-слотов:

- запустите короткую, но не мгновенную операцию;
- пока она выполняется, перейдите в другой child screen или вернитесь на root;
- дождитесь завершения;
- статус под терминалом должен стать зеленым/красным по exit code;
- терминал и лог должны показать реальный итог;
- toast завершения/ошибки должен появиться, если окно живо;
- не должно быть `RuntimeError: The parent element this slot belongs to has been deleted.`;
- кнопки открытия источника, назначения, `LOGS`, `CONFIG`, `REPORT` не должны показывать toast;
- выбор нового пути, удаление или import могут показывать toast, если реально меняют состояние или пользователь отменил picker.

System Operations / Preflight Check

Если проект управляет WSL, сетью, дисками, Windows features, hosts, WinRE или adapter state:

- есть read-only `preflight_status` или аналогичный status snapshot;
- admin/elevated state виден в выводе;
- read-only/status операции работают без UAC;
- known admin-only операции либо запускаются elevated, либо явно запрашивают UAC;
- WSL labels различают `Список для установки` и `Список установленных`;
- WSL installation form имеет понятные поля `Дистрибутив`, `Файл образа`, `Папка установки`;
- Wi-Fi export с key=clear помечен как sensitive и имеет paired import;
- destructive prompt-heavy scripts открываются externally или перепроектированы как GUI-flow.

NiceGUI ProcessPool Fallback

В закрытых portable/sandbox окружениях NiceGUI может не создать multiprocessing process pool. GUI должен стартовать всё равно, если проект использует только обычные GUI-задачи и `run.io_bound`.

Проверка считается успешной, если `runtime\python.exe system_core\ui_nicegui\app.py --smoke` проходит, а серверный запуск не падает на `PermissionError` / `WinError 5`.

Для native picker dialogs PowerShell ищется в таком порядке:

1. `system_core\powershell\pwsh.exe`
2. `pwsh.exe` из `PATH`
3. встроенный `powershell.exe`

Наличие хотя бы одного варианта видно в секции `[GUI portability]` команды:

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