

PackageConsole — QA Pack

README

Quality Assurance Pack — PackageConsole (WPF .NET 8)

This folder contains all documents to execute formal QA on PackageConsole. Start here.

Contents

- Test-Plan.md (scope, strategy, entry/exit criteria)
- Environment-Setup.md (how to prepare a clean test rig)
- Smoke-Checklist.md (15-minute gate)
- Test-Cases-*.md (feature-level detailed steps)
- Regression-Suite.md (stable always-run set)
- UAT-Script.md (business validation script)
- Bug-Report-Template.md (defect format)
- Release-Readiness-Checklist.md (go/no-go)
- Accessibility-Checklist.md, Security-Privacy-Checklist.md
- Performance-Test-Plan.md
- DB-Integrity-Checklist.md
- Test-Data/ (sample data guidance)

How to run the app (Debug)

- 1) Close any running instance
- 2) Build: ``dotnet build -c Debug``
- 3) Launch: ``bin/Debug/net8.0-windows/PackageConsole.exe``

Test identities

- Admin tester: account with Admin Mode access
- Standard tester: account without admin privileges

Key areas to validate

- INI Console (primary): editing UX, undo/redo, delete mode, permissions editors, config toggles, validation
- Package Dashboard: filters, Submitted By dropdown, Refresh Master, CSV export
- Add Package: creation, upgrade section numbering (UPGRADE1 → UPGRADE2 ...)
- Edit Existing INI: load, edit, save, diff behavior
- Admin Settings: paths, tooltips.json editing, central mirroring
- Feedback: submit/read
- SQLite over UNC: reliability via local temp copy pattern

Start with Smoke-Checklist.md, then follow Test-Plan.md.

PackageConsole — QA Pack

Test Plan

Test Plan — PackageConsole

Objective

Ensure the desktop app is functionally correct, stable, and ready for wider Quality Testing.

Scope

- In scope: UI workflows, SQLite data flows (local, central, master), file I/O (INI, CSV), admin-only features, role behavior, error handling.
- Out of scope: Cloud migration deliverables (separate track), non-Windows platforms.

Test Types

- Smoke testing (gate)
- Functional testing (feature test cases)
- Negative/error handling
- Regression testing
- Role-based testing (Admin vs Standard)
- Accessibility (basic keyboard focus/contrast)
- Security & privacy checks (local/UNC paths, PII)
- Performance (basic responsiveness, large INI)

Test Strategy

- Feature sheets with step-by-step cases (Test-Cases-*.md)
- Stable Regression-Suite.md run on each build
- Dual-identity passes (Admin/Standard)
- Data safety: use test BasePath/Central paths; never production shares

Entry Criteria

- Build succeeds; app launches on test rig
- Help content updated; screenshots available
- Admin paths configured to test locations

Exit Criteria

- 0 open Severity 1 (blocker) defects
- ≤ 2 open Severity 2 defects with workarounds approved
- All smoke/regression cases pass
- UAT-Script.md signed by requester

Risks & Mitigations

- SQLite on UNC can appear empty → app uses local temp copies; test via DB-Integrity-Checklist.md
- Concurrent edits → validate conflict handling via save validation and undo/redo

PackageConsole — QA Pack

- Path misconfig → use Admin Settings and Environment-Setup.md

Deliverables

- Executed checklists with results
- Defect tickets (Bug-Report-Template.md)
- UAT sign-off

Schedule

- Smoke: ~15–30 min
- Full feature pass: ~4–6 hours
- Regression on new build: ~60–90 min

PackageConsole — QA Pack

Environment Setup

Environment Setup — QA

Prereqs

- Windows 10/11
- .NET SDK 8.x installed
- Test account with Admin rights, and a Standard account

Initial build & run

- 1) Close any running app instance
- 2) Build: ``dotnet build -c Debug``
- 3) Run: ``bin/Debug/net8.0-windows/PackageConsole.exe``

Configure app for QA

- In Admin Settings:
 - BasePath: set to a QA-only folder (e.g., C:\QA\PackageConsole)
 - Central Share/Metadata path: point to a QA network/test folder (or local folder simulating UNC)
 - tooltips.json: ensure it loads from QA Metadata; save a tiny change to confirm write
- Place sample user SQLite DBs and a MasterDB in QA locations if needed by tests

Artifacts and logs

- Screenshots: capture key flows and failures
- CSV exports: store under QA evidence folder
- Logs: if logging is enabled, attach the log file(s) with time stamps

Resetting state

- Clear QA BasePath subfolders as needed (Backup before clearing)
- Replace sample DBs from a clean baseline

Performance data

- Prepare a large INI sample ($\geq 2,000$ keys across sections) for stress tests

PackageConsole — QA Pack

Smoke Checklist

Smoke Checklist — 15 minutes

Build & Launch

- ☐ Build succeeds in Debug
- ☐ App launches to main shell

Basic Navigation

- ☐ Sidebar loads pages without errors (Dashboard, INI Console, Add Package, Edit Existing INI, Admin Settings, Feedback)
- ☐ Help opens and screenshots render

INI Console (essentials)

- ☐ Sections and keys load for a known INI
- ☐ Edit a value and undo/redo works
- ☐ Toggle Delete Key Mode and remove a test key with confirmation
- ☐ Save applies changes and persists after reload

Dashboard

- ☐ Submitted By = All (Master) loads data
- ☐ Switch to a specific user; rows update
- ☐ CSV export works

Admin & Feedback

- ☐ Admin Settings page opens (as Admin) and can save a trivial path change then revert
- ☐ Feedback submit and view cycle works

Result

- ☐ PASS / [] FAIL — Notes:

PackageConsole — QA Pack

Test Cases INI Console

Test Cases — INI Console (Primary)

Legend: A=Admin, S=Standard

- 1) Load & Layout
 - Steps: Open INI Console with sample package; observe left/middle/right panes
 - Verify: Sections, Keys, Value editor visible; focus moves with selection; tooltips appear on hover
 - Role: A,S
- 2) Browse & Edit w/ Autocomplete
 - Steps: Select a key; start typing a known variable; pick from suggestions
 - Verify: Autocomplete list shows; chosen token inserts correctly; tooltip text displayed
 - Role: A,S
- 3) Config Gear (⚙️) Toggles
 - Steps: Open gear; toggle Show resolved variables, Show tooltips, Expand multi-line editors
 - Verify: Resolved preview updates; tooltips visible/hidden; multi-line expands
 - Role: A,S
- 4) Delete Key Mode (safety flow)
 - Steps: Enable Delete Key Mode; select a key; delete
 - Verify: Confirmation dialog; key removed; undo brings it back
 - Negative: Cancel maintains key
 - Role: A,S
- 5) Add Section/Key (+)
 - Steps: Use + to add a section; add keys; add row in foldperm/regperm/fileperm
 - Verify: New items appear; values editable; minus removes row
 - Role: A,S
- 6) Permissions Editors
 - Steps: In foldperm/regperm/fileperm, add row; set Identity/Rights/Inheritance; save
 - Verify: Validation for required fields; persisted on reload
 - Role: A,S
- 7) Undo/Redo
 - Steps: Make 3 edits across different keys; Ctrl+Z 3x; Ctrl+Y 3x
 - Verify: Edits revert/restore in order across session
 - Role: A,S

PackageConsole — QA Pack

- 8) Save with Validation

- Steps: Leave a required value blank; click Save
- Verify: Inline validation highlights; save blocked; fix and save succeeds
- Negative: Attempt to save with invalid format (e.g., bad path token) → message shown
- Role: A,S

- 9) Variable Resolution

- Steps: Enable resolved view; use variables referencing Admin Settings
- Verify: Preview shows computed value; changing Admin path updates preview
- Role: A,S

10) Large INI Performance

- Steps: Load large INI ($\geq 2,000$ keys); scroll and edit
- Verify: UI remains responsive; save completes $< 3s$ on test rig
- Role: A,S

11) Keyboard Shortcuts

- Steps: Ctrl+S (if supported), Ctrl+Z, Ctrl+Y, Ctrl+F search
- Verify: Actions performed; focus remains in editor
- Role: A,S

12) Error Handling & Recovery

- Steps: Simulate transient file lock; attempt save; retry
- Verify: Friendly error; no data loss; retry succeeds when lock released
- Role: A,S

PackageConsole — QA Pack

Test Cases Dashboard

Test Cases — Package Dashboard

- 1) Default Load
 - Steps: Open Dashboard; Submitted By = All (Master)
 - Verify: Rows load from Master; counts logical
- 2) Switch Submitted By
 - Steps: Change to a specific user database
 - Verify: Grid updates; only that user's packages appear
- 3) Refresh Master (Admin)
 - Steps: As Admin, click Refresh Master
 - Verify: Local build then copy to Central completes; Master view updates
- 4) CSV Export
 - Steps: Export current view
 - Verify: CSV downloads to chosen folder; header/rows match grid state and filters
- 5) Filters & Paging
 - Steps: Apply text filters; clear
 - Verify: Results refine/return; no stale filters
- 6) UNC/WAL Robustness
 - Steps: Ensure central DBs on QA share; open All (Master)
 - Verify: Data loads consistently (local temp copy approach works)
- 7) Error Handling
 - Steps: Break central path; open Dashboard
 - Verify: Clear error; instructions to fix Admin Settings

PackageConsole — QA Pack

Test Cases Add Package

Test Cases — Add Package

- 1) Create New Package
 - Steps: Fill App Name, Version, Vendor, etc.; generate INI
 - Verify: INI created; appears in user DB; required fields validated
- 2) Import Upgrade INI (fix verification)
 - Steps: Import INI containing upgrade section; import again another upgrade
 - Verify: Sections created sequentially UPGRADE1, UPGRADE2, ... not duplicated
- 3) Toolkit Copy (if enabled)
 - Steps: Use Copy Toolkit; select source; generate structure
 - Verify: Folders/files created under BasePath as expected
- 4) Error Handling
 - Steps: Leave mandatory field blank; try to proceed
 - Verify: Blocking validation; helpful message
- 5) Persistence
 - Steps: Save; navigate away; return
 - Verify: Package persists; INI view matches saved content

PackageConsole — QA Pack

Test Cases Edit Existing INI

Test Cases — Edit Existing INI

- 1) Load Existing
 - Steps: Choose an existing package; open editor
 - Verify: All sections/keys load; diffs vs original highlighted if applicable
- 2) Edit & Save
 - Steps: Change multiple keys; save
 - Verify: Only changed keys written; re-open shows changes
- 3) Undo/Redo & Delete
 - Steps: Make edits; delete a key in Delete Key Mode; undo
 - Verify: Behaviors consistent with INI Console rules
- 4) Validation Errors
 - Steps: Introduce invalid value; attempt save
 - Verify: Inline validation blocks save; message clear
- 5) Concurrency
 - Steps: Open same INI in two app instances; edit in one; refresh in the other
 - Verify: No silent overwrite; clear message to reload

PackageConsole — QA Pack

Test Cases Admin Settings

Test Cases — Admin Settings

- 1) Access Control
 - Steps: Open Admin Settings as Standard user
 - Verify: Access blocked/hidden
- 2) Paths Configuration
 - Steps: Set BasePath, Central/Metadata paths to QA folders; Save
 - Verify: Persisted; reopening shows values
- 3) tooltips.json Editing
 - Steps: Add/update a tooltip; save
 - Verify: Appears in INI Console tooltips; survives app restart
- 4) Central Mirroring / Master Refresh
 - Steps: Trigger master refresh or mirror (as applicable)
 - Verify: Local build → copy to Central succeeds; Dashboard reflects updates
- 5) Error Handling
 - Steps: Enter invalid path; save
 - Verify: Validation prevents save; helpful message

PackageConsole — QA Pack

Test Cases Feedback

Test Cases — Feedback

- 1) Submit Feedback
 - Steps: Open Feedback; enter text; submit
 - Verify: Entry stored and visible in list
- 2) View & Filter
 - Steps: Open Feedback list; filter/search (if available)
 - Verify: Items display correctly; time/user metadata sensible
- 3) Paths & Storage
 - Steps: Locate stored feedback file(s) under BasePath
 - Verify: Files created; readable; no PII beyond entered text
- 4) Error Handling
 - Steps: Disconnect path; try to submit
 - Verify: Graceful error shown; no crash; retry works after fix

PackageConsole — QA Pack

Regression Suite

Regression Suite — Always Run

- Launch app; navigate across all pages
- INI Console: edit → undo → redo → save cycle
- INI Console: Delete Key Mode on/off; delete and undo
- INI Console: permissions editors add/remove row; save
- Dashboard: Submitted By toggle; Refresh Master
- Dashboard: CSV export
- Add Package: create, upgrade import numbering (UPGRADE1→UPGRADE2)
- Edit Existing INI: load, edit, save
- Admin Settings: change a path; revert
- Feedback: submit and view
- Help window: opens; images render

PackageConsole — QA Pack

UAT Script

UAT Script — Business Validation

Participants

- Requester/Product Owner, QA lead, Admin tester, Standard tester

Preconditions

- Environment-Setup complete; paths point to QA folders

Scenarios

- 1) Create Package end-to-end
 - Add Package → create with required metadata → verify in Dashboard
- 2) INI authoring flow
 - Open INI Console → edit keys (with tooltips) → toggle config → save → reload
- 3) Upgrade management
 - Import upgrade INIs twice → verify UPGRADE1, UPGRADE2 sequential naming
- 4) Dashboard reporting
 - View All (Master) → CSV export → open file → contents match grid
- 5) Admin curation
 - Update a tooltip in Admin Settings → verify in Console
- 6) Feedback loop
 - Submit feedback → verify appears in list

Acceptance

- Each scenario: PASS/FAIL with comments
- Sign-off: Name, date, signature

Bug Report Template

Bug Report Template

Title

- Short, action-oriented summary

Environment

- App version/build:
- OS version:
- User role (Admin/Standard):
- Paths config (BasePath, Central):

Description

- What happened vs expected

Steps to Reproduce

- 1)
- 2)
- 3)

Actual Result

-

Expected Result

-

Evidence

- Screenshots / screen recording
- Logs (if available)
- Sample INI / CSV attachment

Severity & Priority

- Sev: 1/2/3/4
- Priority: P0/P1/P2

Workaround

- If any

Notes

- Related tickets/links

PackageConsole — QA Pack

Release Readiness Checklist

Release Readiness Checklist

Quality Gates

- ☐ All Smoke-Checklist items PASS
- ☐ Regression-Suite PASS
- ☐ 0 Sev-1 open; ≤ 2 Sev-2 (documented workarounds)
- ☐ UAT signed off

Docs & Help

- ☐ Help window updated; screenshots present
- ☐ Cloud Roadmap PDF present (optional)

Config

- ☐ Admin paths set appropriately for the target environment
- ☐ tooltips.json verified

Operations

- ☐ Build artifacts produced (Debug/Release as needed)
- ☐ Logging (if enabled) writing to expected location
- ☐ Backups for QA data taken

Sign-off

- QA Lead:
- Product Owner:
- Date:

PackageConsole — QA Pack

Accessibility Checklist

Accessibility Checklist (basic)

- ☐ Keyboard navigation works across pages (Tab/Shift+Tab)
- ☐ Focus visible on interactive elements
- ☐ Sufficient text contrast in key sections
- ☐ Tooltips readable; not truncated
- ☐ Images in Help have descriptive captions nearby
- ☐ No information conveyed by color alone

PackageConsole — QA Pack

Security Privacy Checklist

Security & Privacy Checklist

- ☐ Test data contains no real PII or secrets
- ☐ BasePath and Central paths point to QA-only locations
- ☐ App handles path errors gracefully; no crashes
- ☐ CSV exports contain only expected fields; stored securely
- ☐ Feedback content stored locally in QA path; review access
- ☐ Admin-only features inaccessible to Standard users

PackageConsole — QA Pack

Performance Test Plan

Performance Test Plan (lightweight)

Targets

- INI Console interactivity under large data
- Save operation latency
- Dashboard load under Master with many rows

Datasets

- Large INI ($\geq 2,000$ keys)
- Master DB with $\geq 5,000$ rows

Measurements

- Time to initial render of INI Console
- Time to save after 50 random edits
- Dashboard time to load All (Master)

Acceptance (reference rig)

- INI Console render $< 2s$
- Save 50 edits $< 3s$
- Dashboard load $< 2s$

Procedure

- 1) Prepare datasets (Environment-Setup)
- 2) Use a stopwatch or profiler; run each scenario 3 times; average
- 3) Record results and environment specs

PackageConsole — QA Pack

DB Integrity Checklist

DB Integrity Checklist (SQLite local/central/master)

Local User DB

- ☐ New package rows appear after Add Package
- ☐ Edits from INI Console persist and reload correctly

Master (All)

- ☐ Refresh Master (Admin) updates Master DB
- ☐ Dashboard counts logical; sampling matches source user DBs

Central/UNC Robustness

- ☐ With central DBs on QA share, data still loads (local temp copy approach)
- ☐ No empty grids when source DBs are populated

CSV Integrity

- ☐ Exported CSV matches current grid including filters

Error Handling

- ☐ Broken path yields clear error and no crash

PackageConsole — QA Pack

README

Test Data Guidance

INI Samples

- Small: 10–20 keys, 3–5 sections
- Large: $\geq 2,000$ keys across ≥ 50 sections (for performance)
- Include permissions sections (foldperm/regperm/fileperm) with multiple rows

Upgrade INIs

- Prepare two separate upgrade INIs to verify sequential UPGRADE1/UPGRADE2 creation

Master/User DBs

- Create 2–3 user DBs with distinct packages
- Prepare a Master DB baseline; capture expected row counts

Feedback

- Provide 3–5 sample entries with timestamps and users

Storage

- Keep all QA data under a dedicated QA BasePath and Central folder
- Store expected results (CSV/row counts) for quick comparison