Production Checklist

Production Checklist

Pre-release checklist for server owners and Nord Lab builds.

Applies to: `Nord Inventory v2.0.0`

Production Checklist

Use this checklist before replacing an inventory on a live server.

Tebex / Asset Escrow package

- Confirm the ZIP contains a single resource folder named `nord_inventory`.
- Confirm all owner settings are under `config/`; there must be no root `config.lua` or `server/config.lua`.
- Confirm `server/bridge/` and `client/bridge/` are listed in `escrow_ignore` as intended.
- Confirm `server/core/ecosystem.lua` is **not** in `escrow_ignore`.
- Confirm protected `server/core/`, `server/modules/`, `client/core/` and `client/modules/` files were not accidentally exposed.
- Confirm docs/SQL/changelog are not listed in the manifest `files` block, so clients do not download release documentation.
- Publish through Cfx.re/Tebex Asset Escrow before calling the distributed build encrypted/protected.

Files and dependencies

- Keep a copy of the currently deployed inventory resource and database.
- Start `oxmysql` before `nord_inventory`.
- Start any target/notification/TextUI providers in the order required by your configuration.

- Start `nord_inventory` before resources that call its registration exports.
- Confirm `fxmanifest.lua` and all listed files are present.

Database

- Let Nord create its tables automatically on a staging database, or import `sql/nord_inventory.sql` manually.
- Confirm `nord_inventory_custom_items`, `nord_inventory_disabled_reference_items`, audit and backup tables exist.
- Create an Admin Studio backup before a migration/import.
- Keep a full database backup outside Nord before the first production deployment.

Permissions

- Verify Developer/Admin/Owner ACE roles with test accounts.
- Confirm Developer cannot save/delete/import/restore through direct NUI/server event attempts.
- Confirm backup restore is restricted to Owner unless you intentionally change the permission bridge.
- If `Config.AdminAccess.UseAce = false`, verify only configured license fallback entries receive access.

Item registry

- Open System Health and resolve errors before launch.
- Review missing images and invalid/offline exports.
- Verify reference-item disable/restore behavior.
- Verify a DB-authoritative definition cannot be overwritten by `RegisterItem`.
- Verify runtime provider items disappear when their provider stops and re-register when it starts.

External integrations

For each native integration:

- Confirm the provider appears on **Integrations**.

- Confirm dependencies are online.
- Confirm item/provider ownership is what you expect.
- Test client and server external use handlers separately.
- Test return `false` and stopped-resource failure paths.
- Test item actions and metadata schema extensions.
- Stop/restart the provider and confirm cleanup/re-registration.

Player inventory

- Add/remove/transfer/split/give/drop items.
- Open Ground, stash, trunk, glovebox and item containers.
- Test overweight and slot-limit rejection.
- Test reconnect persistence.
- Test two players interacting with the same secondary inventory.

Lifecycle

- Test normal durability loss.
- Test durability repair items with missing, full and damaged targets.
- Test `GetDurability`, `SetDurability`, `AddDurability`, `RemoveDurability` and `RepairItem` from a trusted test resource.
- Test expiration on online and offline-loaded items.

Weapons

- Equip/unequip weapons.
- Verify serial/durability/ammo metadata.
- Verify manual and automatic magazine reload modes.
- Verify reserve ammunition remains inventory-authoritative.

Vehicles

- Test trunk and glovebox capacities by model/category.
- Test addon front/rear trunk placement.
- Test vehicles with no trunk.
- Test locked/unlocked/NPC vehicle behavior according to config.

Admin Studio

- Create/edit/delete/restore a database and reference item.
- Run an import preview before committing.
- Test templates.
- Create a backup, make a harmless change, restore it, then verify registry reload.
- Confirm audit records the administrative changes.

Static validation included with this package

The packaged build should pass:

- Lua compile/syntax validation for all `.lua` files.
- JavaScript syntax validation for `web/app.js`.
- JSON parsing and EN/PT locale-key parity.
- duplicate HTML-ID check.
- NUI callback and Nord client/server event mapping checks.

Static validation cannot reproduce a real FXServer, framework, OneSync, database driver, target provider or third-party resource. Run the checklist on a staging FXServer before swapping a live production server.

v2.0.0 Operations smoke test

- Confirm `nord_inventory_snapshots`, `nord_inventory_forensics_alerts` and `nord_inventory_placed_items` exist.
- Open all Operations tabs and confirm Ledger/Instance filters remain usable on your Admin Studio resolution.
- Run Anti-Dupe once; verify it records diagnostics but does not delete items.
- Create and delete a manual snapshot.
- Restore a disposable test snapshot and verify a pre-restore snapshot is created.
- Edit an online player's inventory and confirm the live client refreshes immediately.
- Test a simulated persistence failure on staging if possible; runtime state should roll back rather than diverge from SQL.
- Place a disposable persistent world item, restart the resource/server and verify it returns.
- Configure one test movement policy and verify server-side rejection for the blocked destination/action.

- Verify Recommended Scripts shows Nord Lab/Renovax Store and Documentation buttons and that only `http/https` URLs open.

Revision #1
Created 2026-09-30 01:04:35 UTC by Riguita
Updated 2026-09-30 01:04:35 UTC by Riguita