# DB-Authoritative Definitions & Precedence

# DB-Authoritative Definitions & Precedence

Why Admin Studio edits survive runtime registration and restart.

**Applies to:** `Nord Inventory v2.0.0`

# Database item persistence

Item-definition precedence is strict and restart-safe:

1. If `nord_inventory_custom_items` contains a row for an item name, that row owns the definition.
2. `shared/items.lua` / `shared/weapons.lua` are consulted only if no database row exists for that name.
3. Resource startup is read-only for item definitions. It never seeds, updates, replaces, or deletes DB item rows.
4. A disabled DB row remains authoritative and blocks file fallback.
5. The only way to restore the reference definition is an explicit Revert action, which deletes the DB override row.

This prevents edited base items from returning to file defaults after `restart nord_inventory`.

# Item definition precedence

Nord Inventory uses one deterministic rule:

1. **Database row exists for the item name** → the database definition is authoritative.
2. **No database row exists** → `shared/items.lua` / `shared/weapons.lua` is used as reference fallback.

Editing a reference/base item from Admin Studio creates or updates its database row. The reference file is never mutated. Reverting a base override removes the database row; only then does the reference definition become active again.

This prevents a base item from being simultaneously resolved from the file and the database.

# Restart behaviour

Resource/server startup is read-only for item definitions. Existing rows in `nord_inventory_custom_items` are never replaced, disabled, deleted, or reseeded from the Lua reference files during boot. Disabled DB rows still own their names and block reference fallback.

---

Revision #1
Created 2026-09-30 01:04:37 UTC by Riguita
Updated 2026-09-30 01:04:37 UTC by Riguita