Operations & Forensics Center — v2.0.0

Operations & Forensics Center — v2.0.0

Nord Inventory v2.0.0 includes an admin-only Operations Center built around the existing server-authoritative inventory model.

Included systems

- **Ledger / Forensics** — searches `nord_inventory_transactions` by transaction id, identifier, item, inventory, UID, weapon serial or container UID. Movement transactions now preserve identity fields in their payload.
- **Anti-Dupe Center** — on-demand bounded scan of persisted + live cached inventories. UID, weapon serial and container UID are checked independently. A unique identity with multiple copies in one stack is also flagged.
- **Item Instance Explorer** — locates item instances across player, stash, container, vehicle and ground inventories and exposes owner, slot, storage key and sanitized metadata.
- **Inventory Manager** — inspects both connected and offline player inventories. Connected players are read from the authoritative live `InventoryCore` state; offline players are read from `nord_inventory_storage`. Every admin edit creates a safety snapshot before mutation when enabled.
- **Snapshots & Recovery** — manual and automatic per-inventory snapshots with owner-level restore permission. A pre-restore snapshot is always created first.
- **Persistent Placed Items** — players can deliberately place an item in the world. The drop, inventory payload and coordinates survive resource/server restarts.
- **Item Policies** — per-item or per-category movement restrictions for giving, dropping, throwing, persistent placement and destinations such as trunk, glovebox, stash, container, player and ground.

- **Dependency Graph** — reads the Nord provider registry and shows active provider/resource relationships in Admin Studio.
- **Equipment Slots 2.0** — special slots can now match item categories or explicit item-name allowlists. Existing servers retain the previous slot layout unless extra equipment slots are configured.

Permissions

The normal Nord Inventory permission bridge is used:

- `forensics` — Operations Center, ledger, instance explorer, anti-dupe and alert resolution.
- `offline_inventory` — Inventory Manager lookup, inspection, editing and manual snapshots. The permission name is kept for backwards compatibility.
- `restore_snapshots` — snapshot restore. Default minimum role is `admin`.
- `offline_inventory` — snapshot creation and deletion, alongside offline inventory management. Default minimum role is `admin`.

Nord Staff/ACE/database access continues to flow through the existing permission bridge.

Inventory Manager source of truth

The manager deliberately uses different sources depending on player state:

- **Online player:** `InventoryCore.GetPlayerInventory(source)` / live runtime state is authoritative. The UI never silently falls back to stale SQL for a connected owner.
- **Offline player:** `nord_inventory_storage` is authoritative.
- Online players are listed first and stale persisted rows for the same owner are de-duplicated.
- Legacy persisted player keys are redirected to the current canonical live inventory key when that owner is connected.
- Manual and automatic snapshots of connected players serialize the current live item table.
- Snapshot restore refreshes the connected player's inventory session immediately.
- Admin edits persist immediately; if the SQL transaction fails, the runtime mutation is rolled back in memory.
- While the Inventory Manager tab is open, selected live inventories receive event-driven updates through the normal inventory watcher path. The subscription is removed when the tab/Studio closes and also expires automatically.

Auto-SQL

The runtime creates these tables with `CREATE TABLE IF NOT EXISTS`:

- `nord_inventory_snapshots`
- `nord_inventory_forensics_alerts`
- `nord_inventory_placed_items`

The same schema is included in `sql/nord_inventory.sql` for manual installations. Existing inventory data is not deleted or rewritten by the Operations migration included with v2.0.0.

Item policies

Edit `config/operations.lua`. Missing properties mean **allow**, so updating from an older build does not block existing item movement.

```
Config.Operations.ItemPolicies = {
    ByName = {
        police_evidence = {
            allowGive = false,
            allowDrop = false,
            allowThrow = false,
            allowTrunk = false,
            allowGlovebox = false,
            allowedJobs = { police = 0 }
        }
    },
    ByCategory = {
        identity = {
            allowDrop = false,
            allowThrow = false
        }
    }
}
```

Supported policy fields: `allowGive`, `allowDrop`, `allowThrow`, `allowPersistentPlace`, `allowTrunk`, `allowGlovebox`, `allowStash`, `allowContainer`, `allowPlayer`, `allowGround`, `allowedJobs`.

Persistent placement

`Config.Operations.PersistentPlacedItems` controls the feature. Persistent items are intentionally different from ordinary drops:

- ordinary drops keep the configured lifetime;
- persistent placed items are database-backed;
- both sides are saved immediately when a persistent item is transferred to reduce crash-window duplication/loss;
- empty persistent placed inventories are removed from both the placed-item registry and inventory storage.

Scan safety

Forensics scans are on-demand and bounded by:

- `MaxLedgerRows`
- `MaxInstanceRows`
- `MaxStorageScanRows`
- `MaxOfflinePlayers`

The scanner overlays live cached inventories on top of database rows. That prevents stale database state and live state from being counted twice.

Equipment Slots 2.0

Existing category rules remain valid. An equipment slot may additionally use `items = { ... }`:

```
{
    key = 'phone',
    slot = 8,
    items = { 'phone', 'nord_phone' },
    label = 'Phone',
    icon = 'fa-mobile-screen'
}
```

Extra equipment slots are opt-in in `config/ui.lua` so an update does not unexpectedly reserve more player slots.

Optional clothing bridge

`Config.Operations.ClothingBridge` provides an opt-in hand-off for physical clothing items. It is disabled by default so existing servers keep their current appearance flow.

When enabled, an item whose category matches `Category` is treated as usable through the configured client export, for example:

```
ClothingBridge = {
    Enabled = true,
    Category = 'clothing',
    Resource = 'nord_appearance',
    ClientExport = 'UseInventoryClothingItem'
}
```

The target export receives `(itemData, slot)` and should return `false` to reject the equip action. Item-specific `use.mode = 'export'` definitions always take priority over this generic bridge.

Exports

Server exports added by the Operations module:

```
exports.nord_inventory:GetItemPolicy(itemName)
exports.nord_inventory:CreateInventorySnapshot(invKey, reason)
exports.nord_inventory:GetInventoryLedger(filters)
exports.nord_inventory:FindItemInstances(filters)
```

Production notes

The anti-dupe scan is diagnostic: it records alerts but never deletes items automatically. Snapshot restore requires an elevated permission and creates a safety snapshot first. Offline admin edits are audited through the existing Nord ecosystem audit system.

Revision #1
Created 2026-09-30 01:04:42 UTC by Riguita
Updated 2026-09-30 01:04:42 UTC by Riguita