Nord Staff Integration

Nord Staff Integration

Automatic admin-access bridge plus shared item/provider APIs.

Applies to: `Nord Inventory v2.0.0`

Nord Staff Integration

Admin access bridge

Nord Inventory now exposes the exact access-export pattern detected automatically by current `nord_staff` builds:

```
exports['nord_inventory']:HasAdminAccess(playerId)
exports['nord_inventory']:GiveAdminAccess(playerId)
exports['nord_inventory']:RemoveAdminAccess(playerId)
```

No change is required in `nord_staff`. When **Dar acesso** is used, Nord Inventory grants the configured role and all server-side Admin Studio permission checks use it immediately. License-based grants are persisted automatically in `nord_inventory_admin_access`, so they survive resource/server restarts. Removing access removes only the Staff-managed grant; ACE and static license roles are never changed.

Configure the bridge in `config/admin.lua`:

```
Config.AdminAccess.NordStaffBridge = {
    Enabled = true,
    Role = 'admin', -- developer | admin | owner
    AllowedResources = { 'nord_staff' }
}
```

Nord Inventory exposes a canonical API intended for Nord Staff and other admin resources. The integration never uses a visible item label as the spawn ID.

Canonical identity

Every external item object follows:

```
{
    name = 'ammo-rifle2',       -- internal/spawn name
    id = 'ammo-rifle2',
    internalName = 'ammo-rifle2',

    label = 'Rifle Ammo Mk II', -- visible text only
    title = 'Rifle Ammo Mk II'
}
```

`name` is always the real Nord Inventory identifier.

Recommended Nord Staff exports

Integration information

```
local info = exports['nord_inventory']:GetIntegrationInfo()
```

Resolve any incoming item value

```
local internal = exports['nord_inventory']:ResolveItemName('mg_ammo')
-- ammo-rifle2
```

The resolver accepts canonical IDs, configured aliases, and unique labels.

Aliases are owner-editable in:

```
config/items.lua
```

under `Config.ExternalItemAliases`.

Item catalogue

```
local items = exports['nord_inventory']:GetItemCatalog()
```

Each entry includes canonical item identity plus `image`, `imageUrl` and `imageFile`.

Item image

```
local url = exports['nord_inventory']:GetItemImage('bandage')
local file = exports['nord_inventory']:GetItemImage('bandage', 'file')
local data = exports['nord_inventory']:GetItemImage('bandage', 'data')
```

Local item URLs use:

```
https://cfx-nui-nord_inventory/web/images/items/<file>
```

Remote HTTP/HTTPS item images are returned unchanged.

Inventory snapshot

```
local snapshot = exports['nord_inventory']:GetInventorySnapshot(playerId)
```

The snapshot is a slot-sorted array designed for Staff/Admin UIs. Every item uses the canonical internal `name`.

Add / remove

```
exports['nord_inventory']:AddItem(playerId, internalName, amount, metadata, slot)
exports['nord_inventory']:RemoveItem(playerId, internalName, amount, metadata, slot)
```

The Nord Staff bridge resolves the item through Nord before either operation.

Why this fixes Staff spawn errors

Older bridges commonly mixed:

```
name = item.label
id = item.name
```

That makes a Staff UI send a display title instead of the spawn identifier. The new contract keeps visible text in `label`/`title` and reserves `name` for the canonical Nord ID.

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