Internal & External Use Behavior

Internal & External Use Behavior

Choose one clear authority for item use.

Applies to: `Nord Inventory v2.0.0`

External item use exports

Nord Inventory has two mutually exclusive usable-item modes:

- `internal`: Nord handles duration, animation, hand prop, player effects and optional removal.
- `export`: Nord validates the item/slot transaction and delegates the gameplay action to another resource export.

The Admin Studio exposes this under **Item Management → Behavior → Use handler** with **Internal (Nord)** and **External export**.

OX-style item definition

A resource can register an item and point its use directly at one of its own exports. When the export name is bare, Nord automatically binds it to the resource calling `RegisterItem`.

```
exports.nord_inventory:RegisterItem('scanner', {
    label = 'Scanner',
```

```
    weight = 250,
    client = {
        export = 'UseScanner', -- automatically <calling_resource>:UseScanner
        remove = 0
    }
})

exports('UseScanner', function(item, slot)
    -- item.name, item.label, item.metadata, item.count, item.slot, ...
    -- Return false to cancel the transaction.
    return true
end)
```

The full resource name is still accepted:

```
client = { export = 'my_scanner:UseScanner', remove = 0 }
-- or
client = { export = 'my_scanner.UseScanner', remove = 0 }
```

Server export

```
exports.nord_inventory:RegisterItem('repair_kit', {
    label = 'Repair Kit',
    weight = 500,
    close = true,
    server = {
        export = 'UseRepairKit',
        remove = 1
    }
})

exports('UseRepairKit', function(source, item, slot, definition)
    -- Return false to cancel and prevent Nord from removing the item.
    return true
end)
```

The Nord-native descriptor is also supported:

```
exports.nord_inventory:RegisterItem('repair_kit', {
    label = 'Repair Kit',
    weight = 500,
    use = {
        mode = 'export',
        export = 'my_mechanic:UseRepairKit',
        side = 'server',
        remove = 1
    }
})
```

`mode = 'export'` may be omitted when `use.export` is present.

Register an external handler for an existing item

Compact form, bound to the calling resource:

```
exports.nord_inventory:RegisterUseExport('repair_kit', 'UseRepairKit', 'server')
```

Explicit resource form:

```
exports.nord_inventory:RegisterUseExport('repair_kit', 'my_mechanic', 'UseRepairKit',
'server')
```

Descriptor/string forms through `RegisterUsableItem` are also accepted:

```
exports.nord_inventory:RegisterUsableItem('repair_kit', 'UseRepairKit', 'server')
exports.nord_inventory:RegisterUsableItem('repair_kit', 'my_mechanic:UseRepairKit', 'server')
exports.nord_inventory:RegisterUsableItem('scanner', { export = 'UseScanner', side = 'client'
})
```

A normal server Lua callback remains supported:

```
exports.nord_inventory:RegisterUsableItem('repair_kit', function(source, item, definition,
slot)
    return true
```

```
end)
```

Register the item and handler in one call

The third `RegisterItem` argument can be a function, export string or export descriptor. The optional fourth argument selects `server` or `client` for string handlers.

```
exports.nord_inventory:RegisterItem('repair_kit', {
    label = 'Repair Kit',
    weight = 500,
    use = { remove = 1 }
}, 'UseRepairKit', 'server')
```

Bulk registration

`RegisterItems` registers multiple runtime items from one resource and applies the same caller-resource binding rules.

```
local ok, result = exports.nord_inventory:RegisterItems({
    scanner = {
        label = 'Scanner',
        weight = 250,
        client = { export = 'UseScanner', remove = 0 }
    },
    repair_kit = {
        label = 'Repair Kit',
        weight = 500,
        server = { export = 'UseRepairKit', remove = 1 }
    }
})
```

Transaction behavior

Nord validates that the player still owns the same item/slot before the external action is committed. If an export is unavailable, errors, or returns `false`, the use is cancelled and Nord does not remove the item.

`remove` stays server-authoritative:

- `remove = 0`: keep the item.
- `remove = 1`: remove one item after a successful external action.
- Higher values remove that amount, limited by the current stack.

External mode is exclusive: Nord does **not** also run its internal duration, animation, hand prop, hunger, thirst, health or armor effects. The external resource owns those effects.

Hands-only animations (`prop = false`)

An internal usable item can explicitly disable every hand-prop fallback while keeping its animation.

```
use = {
    duration = 2500,

    animation = {
        dict = 'missheistdockssetup1clipboard@idle_a',
        clip = 'idle_a',
        flag = 49
    },

    prop = false
}
```

`prop = false` is authoritative for the use action. Nord Inventory will not fall back to:

- `worldModel`;
- keyword prop rules;
- category prop rules;
- the generic fallback prop.

The item's `worldModel` can still be kept for ground/drop presentation.

The default `bandage` uses this mode so the player performs a hands-only bandaging animation with no object attached.

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