3 · Benches & Access

Create fixed and portable benches, control access, manage props and integrate organization ownership.

Bench Types

The runtime normalizes benches into four main modes:

Type	Purpose
`craft`	Standard item crafting
`blueprint`	Blueprint drafting/creation
`stash`	Storage-focused bench
`weapon`	Weapon preview/component workflow

Older labels such as `general`, `illegal` or `military` can still be used as recipe organization concepts, but the current bench runtime normalizes the operational mode to the types above.

The bench type controls which interface and server validation path is used.

Fixed Benches

Fixed benches are persistent world benches stored in:

```
nord_crafting_fixed
```

They contain:

- Label.
- Prop/model.
- Coordinates and heading.
- Bench type.
- Access JSON.
- Assigned recipe/blueprint IDs.
- Stash configuration.
- Creator and creation timestamp.

Admin Studio provides placement mode for creating fixed benches in-world.

After changes, the bench cache is reloaded and synchronized to clients.

Portable Benches

Portable benches are enabled in:

```
Config.Portable = {
    Enabled = true,
    ItemName = 'craft_bench',
    Default = {
        label = 'Bancada Portátil',
        model = `prop_tool_bench02`,
        access = { type = 'public' },
        recipes = { ... },
        stash = { enabled = true, slots = 20, weight = 20000 },
        expiresMinutes = nil
    }
}
```

Placed portable benches are stored in:

```
nord_crafting_placed
```

Inventory requirement

If portable benches are enabled, the active inventory must contain the item named by `Config.Portable.ItemName`.

- With Nord Inventory, use the supported auto-install workflow and verify the item exists after setup.
- With other inventories, create/register the item manually.

Expiration

`expiresMinutes` can be used to create temporary placed benches. `nil` means no configured expiration from this default.

Access Control · Public, Job & Gang

Public

```
access = { type = 'public' }
```

Any player can use the bench, subject to normal distance and recipe validation.

Job

```
access = {
    type = 'job',
    jobs = { 'police', 'mechanic' },
    minGrade = 2
}
```

Supported on QBCore and ESX.

Gang

```
access = {
    type = 'gang',
    gangs = { 'ballas', 'lostmc' },
    minGrade = 1
}
```

Gang access is built into the QBCore bridge.

Normalization

The access resolver accepts common forms such as `list`, plural keys (`jobs`, `gangs`), singular keys, names and comma/space-separated strings.

Organization Access · Nord Crime / OpCrimes

```
Config.CrimeBridge = 'auto'
```

In automatic mode, Crafting prefers `nord_crime` when available and can fall back to `op-crime`.

Organization access

```
access = {
    type = 'org',
    orgId = 12
}
```

The player's organization state is checked server-side.

Nord Crime-specific modes

The bridge supports Nord Crime organization access patterns including:

- Direct organization ID.
- `zone_owner` mode.
- `org_type` mode.

Bench creation exports

Nord Crime or another compatible server resource can create organization benches using:

```
exports['nord_crafting']:CreateNordCrimeOrgBench(orgId, data)
exports['nord_crafting']:CreateOrgBench(orgId, data)
exports['nord_crafting']:AddOrgCraftingBench(orgId, data)
```

These aliases call the same organization bench creation implementation.

Bench Props & Placement

Nord Crafting loads bench prop catalogs from:

```
models/default_props.json
models/custom_props.json
```

The Admin Studio can also manage custom prop entries stored in:

```
nord_crafting_props
```

Placement

Admin placement spawns the selected model in-world and stores coordinates/heading when confirmed.

Target interaction

Spawned bench entities are registered through the target bridge when a supported target is active.

Restart behavior

On Crafting restart, the client clears spawned bench entities/cache and requests a fresh synchronization from the server, reducing stale world props after resource restarts.