6 · Admin Studio

Operate the complete in-game management interface for recipes, benches, analytics, levels, props, blueprints and updates.

- Admin Studio Overview
- Recipe Manager
- Bench Manager & Usage Analytics
- Levels & Runtime Options
- Prop Manager
- Blueprint Manager
- Update Checker & Security

Admin Studio Overview

Open the Crafting Admin Studio with:

```
/craftbench
```

or through the `nord_staff` integration.

The current interface covers:

- Dashboard/system overview.
- Recipes.
- Benches.
- Bench Manager and usage analytics.
- Props.
- Levels.
- Blueprints.
- Runtime Options.
- Update status.

Admin actions are validated server-side through the Crafting admin manager.

Recipe Manager

The recipe manager can:

- Create custom recipes.
- Edit database-backed custom recipes.
- Delete custom recipes.
- Import recipe data.
- Search and filter recipe sources.
- Configure output item/amount/time.
- Configure ingredients.
- Enable blueprint requirements.
- Enable level requirements.
- Configure minimum level.

Default config recipes are presented as defaults, while database recipes are treated as custom entries.

The server protects the edit/delete path so Admin Studio does not silently rewrite the static Lua config recipe file.

Bench Manager & Usage Analytics

Admin Studio can request an overview of registered benches and usage data.

The interface supports:

- Search by bench/model/ID.
- Edit bench configuration.
- Teleport an admin to a bench location.
- Copy/inspect coordinates.
- Review top benches by usage.
- Review usage series/history where data is available.

Craft and stash activity is tracked by bench ID so administrators can understand which benches are actually being used.

Levels & Runtime Options

The Levels tab manages progression tiers stored in `nord_crafting_levels`.

The Options panel persists global runtime values in `nord_crafting_settings`.

Current runtime options

- Level System.
- Base XP per Craft.
- Level-Based Failure.
- Minimum failure percentage.
- Maximum failure percentage.
- Debug Logs.
- Disable updater logs.
- Disable all resource logs.
- OpCrimes toggle.
- Discord Logs.
- Nord VIP integration.

The server applies runtime options back to the active `Config` table, so the change can affect live behavior without rewriting the original configuration file.

Prop Manager

The Prop Manager combines default and custom bench prop catalogs.

Custom prop records are stored in:

```
nord_crafting_props
```

Admin actions include:

- Create prop entry.
- Update prop entry.
- Delete prop entry.
- Load merged prop list into the bench creation UI.

Static prop catalogs are also shipped in:

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

Blueprint Manager

Admin Studio can create and delete blueprint definitions stored in the database.

A blueprint definition can include:

- ID.
- Label.
- Drafting materials.
- Output/reference item fields.
- Maximum uses.
- Image.

The blueprint list is synchronized to the NUI and is also used by blueprint bench drafting flows.

Before publishing a new blueprint-gated recipe, verify:

1. Blueprint definition exists.
2. `blueprint` inventory item exists.
3. Draft materials are valid items.
4. Required recipe `blueprint_type` matches the definition ID exactly after normalization.

Update Checker & Security

Nord Crafting contains a GitHub release checker for private/public repository workflows.

Console command:

```
nordcrafting:update
```

The Admin Studio can also request update status.

Recommended token handling

The updater supports providing the GitHub token through a server convar:

```
set nord_crafting_github_token "YOUR_TOKEN"
```

Do not publish a live GitHub access token inside documentation, screenshots, public ZIPs or shared config examples.

If a token has previously been distributed in plaintext, rotate/revoke it in GitHub and move the replacement to a private server-side secret/convar.

Download behavior

The updater can download the latest release ZIP into the resource directory, but the current flow still expects the server owner to extract/apply it and restart the resource/server manually.