Nord Forge — Official Documentation

Official English documentation for Nord Forge, including a complete FiveM resource installation guide covering server registration, Forge licensing, Keymaster setup, resource installation, dependencies, SQL and validation.

- 01 — Getting Started
  - What is Nord Forge?
  - Account Registration
  - Customer Quick Start
- 02 — Customer Portal
  - My Products
  - Forge Key
  - Updates & Release Information
- 03 — Servers & Licensing
  - Registering Your Server
  - server.cfg Configuration
  - Server Limits & Privacy
  - License Validation Flow
  - License Error Reference
- 04 — FiveM Resource Installation

    - Installation Overview
    - Step 1 — Get Your Forge Key
    - Step 2 — Register Your FiveM Server
    - Step 3 — Configure server.cfg
    - Step 4 — Download the Resource
    - Step 5 — Install the Resource
    - Step 6 — Install Dependencies
    - Step 7 — Database & SQL
    - Step 8 — Configure the Resource
    - Step 9 — Add the Resource to server.cfg
    - Step 10 — Start & Verify
    - Full Installation Example

- 05 — Marketplace & Purchases
    - Marketplace
    - Free Products
    - Manual Purchases
    - Tebex Purchases

- 06 — Developer Workspaces
    - Becoming a Developer
    - Workspace Roles
    - Workspace Lifecycle

- 07 — Products & Releases
    - Creating a Product
    - Creating Releases
    - Publishing a Product
    - Managing Existing Releases

- 08 — Protected Builds & FiveM Integration
    - Protected Customer Builds
    - Editable Files & escrow_ignore
    - Bundled FiveM License Resource

    - Licensing API Endpoints

- 09 — Tebex Bridge
    - Configuring Tebex
    - Product Tebex Setup

- 10 — Discord, Support & Documentation
    - Forge Discord Bot
    - Discord Workspace Setup
    - Support & Purchase Conversations
    - Workspace Documentation
    - Website Builder

- 11 — Administration & Deployment
    - Self-Hosted Installation
    - Important Environment Variables
    - Network Admin
    - Maintenance & Backups

- 12 — Troubleshooting & Reference
    - Customer Troubleshooting
    - Developer Troubleshooting
    - Production Checklist
    - FiveM Integration Quick Reference

01 — Getting Started

What Nord Forge is, how accounts work, and the basic customer workflow.

What is Nord Forge?

Nord Forge

Nord Forge is Nord Lab's platform for resource distribution, licensing, developer workspaces, releases, customer access, support, documentation and integrations.

The platform is built around two main experiences:

- **Customer Portal** — buy or receive products, manage your global Forge Key, register FiveM servers and download authorized releases.
- **Developer Workspaces** — create products, publish releases, manage licenses, collaborate with a team, configure Tebex, Discord, documentation and public pages.

Main capabilities

Nord Forge includes:

- Global Forge Keys for customers
- Manual FiveM server registration
- Product licenses linked to the customer's Forge Key
- Marketplace and public stores
- Free, manual and Tebex purchase flows
- Product and release management
- Protected customer builds for script products
- Clean developer downloads
- Workspace roles and permissions
- Documentation management
- Website Builder
- Discord Forge Bot integration
- Support and purchase conversations
- Audit logs, API keys and infrastructure records
- Platform administration and backups

> Access is authorization-based. Creating a Nord Forge account does not automatically grant access to every product.

Account Registration

Account Registration

New users register with an email address, username and password.

The password must contain at least **8 characters**.

Forge Key

When an account is created, Nord Forge automatically generates a unique global **Forge Key** for that user.

Example:

```
FG-XXXX-XXXX-XXXX-XXXX
```

This Forge Key is the customer's global Nord Forge license identity.

Products are assigned to this key through the licensing system.

First platform account

On a fresh Nord Forge installation, the **first registered user** automatically receives the global role:

```
network_admin
```

All later accounts are created as normal users.

Customer-first onboarding

A new account starts as a customer by default.

A developer workspace is only created when the user explicitly activates the developer flow or creates a workspace.

Customer Quick Start

Customer Quick Start

For a protected FiveM resource, use this workflow:

1. Create or sign in to your Nord Forge account.
2. Buy, claim or receive access to a product.
3. Open **My Products** and confirm the product is active.
4. Copy your global **Forge Key**.
5. Open **Servers** in the customer area.
6. Register your FiveM `sv_licenseKey` / Keymaster.
7. Add the required license convars to `server.cfg`.
8. Download the latest authorized release.
9. Install and start the resource.

Recommended `server.cfg` configuration:

```
setr nord_license "FG-XXXX-XXXX-XXXX-XXXX"
setr nord_keymaster "cfxk_xxxxxxxxxxxxxxxx"
sv_licenseKey "cfxk_xxxxxxxxxxxxxxxx"
```

> **Server registration is mandatory in the production configuration.** A valid Forge Key alone is not enough to run protected resources.

02 — Customer Portal

Customer products, downloads, updates, Forge Key and product access.

My Products

My Products

My Products is the customer's main product library.

A product appears here when:

- A free product has been claimed.
- A developer has granted access manually.
- A Tebex purchase has been successfully activated.
- A license has been assigned directly by a workspace.
- The logged-in user is a developer with access to that product's workspace.

Product information

The product details can include:

- Product name
- Description
- Store/workspace
- Latest release
- Changelog
- Update status
- Critical update flag
- Configuration change flag
- Database migration flag
- Compatibility notes
- License status

Downloads

Customer downloads are tied to the latest **published release with a valid file**.

A product cannot be distributed as a normal production product if there is no published release file.

Forge Key

Your Forge Key

Every Nord Forge user has one global Forge Key.

Example:

```
FG-XXXX-XXXX-XXXX-XXXX
```

The same Forge Key can have multiple products assigned to it.

What the Forge Key controls

Nord Forge validates:

- The Forge Key
- The requested product
- The product license status
- License expiration
- The registered FiveM server / Keymaster
- The resource identity

A product must be actively assigned to the Forge Key before validation can succeed.

Keep it private

Do not publish your Forge Key in:

- Public repositories
- Discord channels
- Screenshots
- Shared configuration files
- Support posts visible to other users

Treat it as a private server credential.

Updates & Release Information

Updates & Release Information

Nord Forge tracks the latest published release and the release most recently downloaded by the user.

When they differ, the customer can be shown that an update is available.

Releases can include additional update metadata:

- **Critical Update** — indicates an important update.
- **Configuration Required** — configuration may need to be reviewed or changed.
- **Database Migration Required** — the update requires a database migration.
- **Compatibility Notes** — additional information about supported versions or dependencies.

Always read the changelog and compatibility notes before updating a production server.

03 — Servers & Licensing

Mandatory server registration, Keymasters, license validation and common errors.

Registering Your Server

Registering Your Server

Protected Nord Forge resources require the FiveM server to be registered in the customer **Servers** area.

In production, Nord Forge has automatic server binding disabled by default.

Before registering a server

The account must have at least one **active product license**.

What to register

Register the server's FiveM `sv_licenseKey`, normally beginning with:

```
cfxk_
```

Nord Forge accepts either:

```
cfxk_xxxxxxxxxxxxxxxx
```

or a pasted configuration line such as:

```
sv_licenseKey "cfxk_xxxxxxxxxxxxxxxx"
```

Registration flow

1. Sign in to Nord Forge.
2. Make sure the required product is active in **My Products**.

3. Open **Servers**.
4. Add your server.
5. Enter its FiveM `sv_licenseKey`.
6. Optionally add a display name or hostname.
7. Save the server.
8. Configure the same Keymaster in `server.cfg`.
9. Start the protected resource.

> If the server has not been registered, protected resources are blocked with `server_not_registered_add_keymaster_in_forge`.

server.cfg Configuration

server.cfg Configuration

The recommended production configuration is:

```
setr nord_license "FG-XXXX-XXXX-XXXX-XXXX"
setr nord_keymaster "cfxk_xxxxxxxxxxxxxxxx"
sv_licenseKey "cfxk_xxxxxxxxxxxxxxxx"
```

`nord_license`

Contains the customer's global Forge Key.

```
setr nord_license "FG-XXXX-XXXX-XXXX-XXXX"
```

`nord_keymaster`

Contains the server Keymaster used by Nord Forge validation.

```
setr nord_keymaster "cfxk_xxxxxxxxxxxxxxxx"
```

`sv_licenseKey`

This is the normal FiveM server license key.

```
sv_licenseKey "cfxk_xxxxxxxxxxxxxxxx"
```

The value used for `nord_keymaster` should match the FiveM `sv_licenseKey` registered in Nord Forge.

Legacy Forge Key fallbacks

Protected loaders first use `nord_license`, while older integrations may support:

```
setr nord_forge_key "FG-XXXX-XXXX-XXXX-XXXX"
```

or:

```
setr forge_key "FG-XXXX-XXXX-XXXX-XXXX"
```

The recommended configuration is always `nord_license`.

Server Limits & Privacy

Server Limits & Privacy

The production configuration allows a limited number of active Keymasters per Forge account.

Default limit

The default platform limit is:

```
3 active servers / Keymasters per Forge Key
```

The server administrator can change this platform limit with:

```
NORD_FORGE_MAX_KEYMASTERS=3
```

One customer per active Keymaster

An active FiveM Keymaster cannot be registered to two different Forge customers at the same time.

If it is already linked elsewhere, Nord Forge returns:

```
keymaster_already_registered_to_another_forge_key
```

Raw Keymaster storage

Nord Forge does **not** store the raw `cfxk_` value.

The backend normalizes it and stores a cryptographic hash. The interface can use a masked display value instead of exposing the full key.

Removing a server

Removing a server changes its registration status to removed.

A protected resource running from that server will no longer validate until the server is registered and active again.

License Validation Flow

License Validation Flow

A protected resource validates against the Nord Forge licensing API.

The validation checks:

1. The Forge Key exists and belongs to an active account.
2. The product can be resolved.
3. The product is assigned to the customer.
4. The product license is active.
5. The license has not expired.
6. The FiveM Keymaster is registered and active.
7. The Keymaster belongs to the same Forge customer.

Session validation

Protected builds can request a temporary session used to unlock protected runtime code.

The current backend returns:

- A temporary decryption key
- A session TTL
- A heartbeat interval
- Release update information
- Active server slot information

Revoking or suspending product access causes later validation/session requests to fail.

License Error Reference

License Error Reference

These are the main licensing errors implemented by the current Nord Forge backend.

`invalid_forge_key`

The Forge Key does not exist or the account is not active.

`product_not_assigned`

The requested product is not assigned to this customer.

`license_inactive`

The product license exists but is not active.

`license_expired`

The product license has expired.

`missing_keymaster_use_server_cfg_str_nord_keymaster`

The protected resource did not receive a valid Keymaster.

Check:

```
setr nord_keymaster "cfxk_xxxxxxxxxxxxxxxx"
```

`server_not_registered_add_keymaster_in_forge`

The server is not registered in **Forge → Servers**.

Add the server manually before using the protected resource.

`keymaster_already_registered_to_another_forge_key`

The Keymaster is already active on another Nord Forge account.

`keymaster_blocked`

The server was previously registered but is currently removed or blocked.

`keymaster_limit_reached_max_3`

The Forge Key already has the maximum number of active servers under the default limit.

`product_not_found_for_...`

Nord Forge could not match the requested product/resource to an existing product.

04 — FiveM Resource Installation

Complete customer guide for registering a FiveM server, configuring the Forge license and installing protected Nord Forge resources.

Installation Overview

FiveM Resource Installation

This guide explains how to install a Nord Forge resource on a FiveM server from start to finish.

For protected Nord Forge resources, the installation requires **both**:

1. A valid Nord Forge product license.
2. A FiveM server registered in Nord Forge.

A valid Forge Key alone is **not enough**.

Complete installation flow

Follow these steps in order:

1. Sign in to Nord Forge.
2. Confirm the resource is available in **My Products**.
3. Copy your global **Forge Key**.
4. Register your FiveM server in **Forge → Servers**.
5. Add the Forge Key and Keymaster to `server.cfg`.
6. Download the latest resource release.
7. Extract the resource to your FiveM resources folder.
8. Install any required dependencies.
9. Import SQL files if the resource requires them.
10. Configure the resource.
11. Add `ensure resource_name` to `server.cfg`.
12. Restart the server.
13. Check the console for successful Nord Forge validation.

> Always register the FiveM server before starting a protected resource for the first time.

Step 1 — Get Your Forge Key

Step 1 — Get Your Forge Key

Every Nord Forge account has a global Forge Key.

Example:

```
FG-XXXX-XXXX-XXXX-XXXX
```

Where it is used

Protected Nord Forge resources use this key to identify the customer account and verify product access.

Add it to your FiveM `server.cfg`:

```
setr nord_license "FG-XXXX-XXXX-XXXX-XXXX"
```

Replace the example with your real Forge Key.

Important

Do not share your Forge Key publicly.

Never publish it in:

- GitHub repositories
- Discord channels
- Screenshots
- Public support messages
- Shared server configuration examples

Step 2 — Register Your FiveM Server

Step 2 — Register Your FiveM Server

Before using a protected resource, you must register the FiveM server in Nord Forge.

What you need

You need the FiveM server license key configured as:

```
sv_licenseKey "cfxk_xxxxxxxxxxxxxxxx"
```

Nord Forge uses this Keymaster to identify the FiveM server.

Register the server

In Nord Forge:

1. Sign in to your account.
2. Open **Servers**.
3. Click **Add Server**.
4. Enter the server Keymaster / `sv_licenseKey`.
5. Optionally enter a server name.
6. Save the server.

You can normally paste either:

```
cfxk_xxxxxxxxxxxxxxxx
```

or:

```
sv_licenseKey "cfxk_xxxxxxxxxxxxxxxxx"
```

Server must match

The Keymaster registered in Nord Forge must match the Keymaster used by the FiveM server.

If the server is not registered, protected resources can return:

```
server_not_registered_add_keymaster_in_forge
```

Default server limit

By default, one Forge account can have up to:

```
3 active registered servers
```

The platform administrator can change this limit.

Step 3 — Configure server.cfg

Step 3 — Configure `server.cfg`

Add the Nord Forge licensing configuration to your FiveM `server.cfg`.

Recommended configuration:

```
setr nord_license "FG-XXXX-XXXX-XXXX-XXXX"
setr nord_keymaster "cfxk_xxxxxxxxxxxxxxxx"
sv_licenseKey "cfxk_xxxxxxxxxxxxxxxx"
```

`nord_license`

Your global Nord Forge license key:

```
setr nord_license "FG-XXXX-XXXX-XXXX-XXXX"
```

`nord_keymaster`

The FiveM server Keymaster used by Nord Forge validation:

```
setr nord_keymaster "cfxk_xxxxxxxxxxxxxxxx"
```

`sv_licenseKey`

Your normal FiveM server license:

```
sv_licenseKey "cfxk_xxxxxxxxxxxxxxxx"
```

Important

These two values must refer to the same FiveM server:

```
setr nord_keymaster "cfxk_xxxxxxxxxxxxxxxx"
sv_licenseKey "cfxk_xxxxxxxxxxxxxxxx"
```

The same Keymaster must also be registered in **Nord Forge → Servers**.

Legacy Forge Key fallbacks

Some older resources may also support:

```
setr nord_forge_key "FG-XXXX-XXXX-XXXX-XXXX"
```

or:

```
setr forge_key "FG-XXXX-XXXX-XXXX-XXXX"
```

The preferred configuration is:

```
setr nord_license "FG-XXXX-XXXX-XXXX-XXXX"
```

Step 4 — Download the Resource

Step 4 — Download the Resource

Once the product is assigned to your account:

1. Open **My Products**.
2. Select the resource.
3. Review the latest version and changelog.
4. Check for configuration changes or database migrations.
5. Download the latest release.

Before updating an existing resource

Check whether the release contains:

- Critical update warnings
- Configuration changes
- Database migrations
- Compatibility notes
- New dependencies

Back up your current configuration and database before installing a major update.

Step 5 — Install the Resource

Step 5 — Install the Resource

After downloading the resource ZIP:

1. Extract the ZIP

Extract the resource package.

You should normally get a folder such as:

```
nord_inventory
```

or:

```
nord_crafting
```

2. Move it to the FiveM resources directory

Example:

```
resources/[nord]/nord_inventory
```

Recommended structure:

```
resources/
└── [nord]/
    ├── nord_inventory/
    ├── nord_crafting/
    └── nord_hud/
```

3. Do not rename protected resources unnecessarily

Some protected resources identify the product using the resource/build configuration.

Keep the original resource folder name unless the resource documentation explicitly says it can be changed.

4. Check `fxmanifest.lua`

Make sure the resource contains a valid FiveM manifest:

```
fxmanifest.lua
```

or, for legacy resources:

```
__resource.lua
```

Do not remove Nord Forge protection/bootstrap files from protected customer builds.

Step 6 — Install Dependencies

Step 6 — Install Dependencies

Some Nord Forge resources require additional FiveM resources.

Examples can include:

- Frameworks
- Database libraries
- Inventory systems
- Target systems
- Notification systems
- Voice systems
- UI libraries

Check the individual resource documentation

Every resource can have different requirements.

Before starting it, check:

- Dependencies
- Minimum supported versions
- Framework support
- Required start order

Start dependencies first

Example:

```
ensure oxmysql
ensure qb-core
ensure nord_inventory
```

The exact order depends on the resource.

If a Nord resource depends on another resource, the dependency should normally start first.

Step 7 — Database & SQL

Step 7 — Database & SQL

Some Nord Forge resources require database tables or migrations.

If the resource package contains SQL files, review them before starting the resource.

Example files:

```
install.sql
database.sql
migration.sql
```

Installation

Import the required SQL into the database used by your FiveM server.

Common database tools include:

- phpMyAdmin
- Adminer
- HeidiSQL
- MySQL/MariaDB CLI

Updates

If a Nord Forge release is marked as:

```
Database Migration Required
```

read the release notes before updating.

Always back up the database before applying a migration to a production server.

Step 8 — Configure the Resource

Step 8 — Configure the Resource

Open the resource configuration files and configure them for your server.

Depending on the product, this can include:

- Framework
- Locale
- Inventory
- Target
- Notifications
- Text UI
- Database
- Permissions
- Integrations

Common configuration locations include:

```
config.lua
config/
shared/config.lua
locales/
```

Protected Nord Forge builds are designed to keep supported configuration, locale and public files editable.

Do not edit protected runtime files

If a runtime file is protected, do not attempt to remove or bypass the protection.

Use the exposed configuration files and documented APIs instead.

Step 9 — Add the Resource to server.cfg

Step 9 — Add the Resource to `server.cfg`

Add the resource to your FiveM startup configuration.

Example:

```
ensure nord_inventory
```

For several Nord resources:

```
ensure nord_inventory
ensure nord_crafting
ensure nord_hud
```

Recommended order

Put required dependencies before the Nord resource.

Example:

```
ensure oxmysql
ensure ox_lib
ensure qb-core

ensure nord_inventory
```

The exact order depends on the resource documentation.

Licensing configuration should already exist

Make sure the license configuration is present before starting protected resources:

```
setr nord_license "FG-XXXX-XXXX-XXXX-XXXX"
setr nord_keymaster "cfxk_xxxxxxxxxxxxxxxxx"
sv_licenseKey "cfxk_xxxxxxxxxxxxxxxxx"
```

Step 10 — Start & Verify

Step 10 — Start & Verify

Restart your FiveM server after completing the installation.

You can also start the resource manually from the server console:

```
ensure nord_inventory
```

or:

```
restart nord_inventory
```

What to check

Verify that:

- The resource starts without Lua errors.
- Nord Forge license validation succeeds.
- The server is recognized as registered.
- Required dependencies are running.
- Database tables exist.
- The resource UI/features load correctly.

Forge validation problems

If the resource reports:

```
server_not_registered_add_keymaster_in_forge
```

register the server in Nord Forge.

If it reports:

```
missing_keymaster_use_server_cfg_setr_nord_keymaster
```

add:

```
setr nord_keymaster "cfxk_xxxxxxxxxxxxxxxxx"
```

If it reports:

```
invalid_forge_key
```

verify:

```
setr nord_license "FG-XXXX-XXXX-XXXX-XXXX"
```

Full Installation Example

Full Installation Example

This is a complete example for installing a protected Nord Forge FiveM resource.

1. Register the server in Nord Forge

Register:

```
cfxk_MY_SERVER_KEY
```

under:

```
Nord Forge → Servers
```

2. Configure `server.cfg`

```
sv_licenseKey "cfxk_MY_SERVER_KEY"

setr nord_license "FG-MY-FORGE-KEY"
setr nord_keymaster "cfxk_MY_SERVER_KEY"

ensure oxmysql
ensure qb-core
ensure nord_inventory
```

3. Install the files

Place the resource at:

```
resources/[nord]/nord_inventory
```

4. Configure it

Edit the supported configuration files inside the resource.

5. Import SQL

Import any required SQL or migration files included with the resource.

6. Restart the server

Start the FiveM server and verify that Nord Forge licensing succeeds.

Final checklist

- Product is in **My Products**
- Forge Key is active
- FiveM server is registered in Forge
- `nord_license` is configured
- `nord_keymaster` is configured
- `sv_licenseKey` matches the registered Keymaster
- Resource folder is installed correctly
- Dependencies are installed
- SQL is imported when required
- Resource is added to `server.cfg`
- Server console shows no licensing error

05 — Marketplace & Purchases

How free products, manual purchases and Tebex checkout work.

Marketplace

Marketplace

The authenticated marketplace lists products that are ready for distribution.

A product must meet all of these requirements before appearing normally:

- Product status is `published`.
- Store status is `published`.
- At least one published release exists.
- The published release has a file.

The marketplace can be filtered by search and product category.

Current product categories include:

- Scripts
- Bots
- MLOs
- Maps
- Other

Products already owned by the customer, or products from a developer workspace the user belongs to, are excluded from the normal marketplace purchase list.

Free Products

Free Products

Products with a price of `0` are delivered automatically.

When a customer claims a free product, Nord Forge:

1. Creates or reuses the active product license.
2. Associates the product with the customer's Forge Key.
3. Adds the product to **My Products**.
4. Records the free grant.
5. Sends a customer notification.

No manual seller approval is required for a normal free product.

Manual Purchases

Manual Purchases

Paid products using the **Manual** payment provider use a buyer/developer conversation flow.

Customer flow

1. Open the product.
2. Start the purchase.
3. A purchase conversation is created.
4. Discuss the purchase with the seller/workspace.
5. The workspace grants access after the payment or agreement is confirmed.

Access grant

When access is granted:

- The product license becomes active.
- The product is associated with the customer's Forge Key.
- The purchase thread is updated.
- The customer receives a notification.
- The product appears in **My Products**.

The product must have a published release file before access can be granted.

Tebex Purchases

Tebex Purchases

Products configured with the **Tebex** payment provider can use the Nord Forge Tebex Bridge.

Checkout flow

1. The customer starts checkout from Nord Forge.
2. Nord Forge creates or redirects to the configured Tebex checkout.
3. Tebex processes the payment.
4. The Tebex webhook is received by Nord Forge.
5. Nord Forge matches the payment to the product and customer.
6. The product is associated with the customer's Forge Key.
7. The product becomes available in **My Products**.

Product matching

Tebex products are linked using the configured **Tebex Package ID**.

Workspace owners should make sure the Package ID in Nord Forge matches the package configured in Tebex.

06 — Developer Workspaces

Developer onboarding, workspace ownership, members and permissions.

Becoming a Developer

Becoming a Developer

Nord Forge uses a customer-first account model.

A normal account does not automatically receive a developer workspace.

When the user activates the developer flow, Nord Forge creates:

- A developer workspace
- The user as workspace **Owner**
- A store associated with that workspace

If the user already owns or belongs to a non-customer workspace, the existing developer workspace can be reused.

Workspace Roles

Workspace Roles

The current team management flow exposes the following main roles.

Owner

The workspace owner has full workspace control.

Owner permissions include all workspace capabilities and owner-only actions.

Developer

Intended for product and release development.

Main permissions include:

- Workspace view
- Product management
- Download/release management
- Documentation management
- License viewing
- Analytics viewing

Support

Intended for customer support.

Main permissions include:

- Workspace view
- Support management
- License viewing
- Customer viewing

Viewer

Read-focused workspace access.

Main permissions include:

- Workspace view
- Documentation view
- Analytics view

Owner-only team management

The workspace owner controls invitations, member role changes, member removal and workspace ownership transfer.

Workspace Lifecycle

Workspace Lifecycle

Workspaces can be active or archived.

Archive

Archiving keeps the workspace and its data but removes it from normal active operation.

Restore

An archived workspace can be restored by its owner.

Delete protection

A workspace cannot be deleted normally while it still has customer/license relationships that require it to be preserved.

In that case, archive the workspace and resolve active customer access first.

Ownership transfer

Workspace ownership can be transferred to another authorized member through the workspace ownership flow.

07 — Products & Releases

Creating products, release requirements, publishing and customer distribution.

Creating a Product

Creating a Product

Developers with product management permission can create products inside a workspace.

New products are always created as:

```
draft
```

Nord Forge does not allow a new product to be published before a valid release exists.

Product information

A product can include:

- Name
- Slug
- Category
- Short summary
- Full description
- Price
- Framework
- Payment provider
- Tebex Package ID
- License type
- License duration
- Checkout status
- Publication status

Categories

Current categories are:

- `scripts`
- `bots`
- `mlos`
- `mapas`
- `outros`

License types

The current interface supports:

- Lifetime
- Monthly
- Subscription
- Trial

A lifetime product can use a duration of `0` days. Timed licenses use the configured duration to calculate expiration.

Creating Releases

Creating Releases

A release represents a downloadable version of a product.

Required file

Every new release requires a file.

Nord Forge rejects a release without an uploaded file.

Release fields

A release can include:

- Version
- Changelog
- File
- Status
- Critical update flag
- Configuration required flag
- Database migration required flag
- Compatibility notes

Release status

Releases support:

- `draft`
- `published`
- `archived`

Customer downloads use published releases with valid files.

Publishing a Product

Publishing a Product

A product can only be published when at least one valid published release with a file exists.

If a developer tries to publish too early, Nord Forge returns:

```
product_requires_release_before_publish
```

Recommended publishing flow

1. Create the product.
2. Keep it in Draft.
3. Complete the product information.
4. Upload the first release ZIP/file.
5. Add the changelog and compatibility information.
6. Publish the release.
7. Publish the product.
8. Confirm that it appears correctly in the marketplace/store.

This rule prevents products with no usable download from reaching customers.

Managing Existing Releases

Managing Existing Releases

Developers can update release metadata after creation.

Supported actions include:

- Change version
- Change changelog
- Change release status
- Update critical/config/database flags
- Update compatibility notes
- Replace the release file
- Delete releases when allowed

Replacing a release file updates the existing release instead of creating a fake version.

The customer library continues to resolve the latest published release with a valid file.

08 — Protected Builds & FiveM Integration

How protected customer builds work and how to use the bundled FiveM licensing resource.

Protected Customer Builds

Protected Customer Builds

Nord Forge can generate a protected customer build when a customer downloads a licensed **script** product.

Developer download

Developers/admins can use direct release downloads and receive the clean uploaded package.

Customer download

For protected script products, customer downloads can be transformed into a protected build.

The protected build:

- Keeps required public/configuration files editable.
- Protects runtime Lua code.
- Validates the customer's Forge Key online.
- Validates the registered FiveM server.
- Requests a temporary session key before protected code is unlocked.
- Can stop validating after product access is suspended or revoked.

Category behavior

The current product UI enables protected-build behavior for the **Scripts** category.

Bots, MLOs, maps and other asset categories are intended to be delivered clean rather than through the Lua script protection flow.

Editable Files & escrow_ignore

Editable Files & `escrow_ignore`

Nord Forge keeps common configuration and documentation files editable in protected customer packages.

Default public/editable content includes patterns such as:

- `fxmanifest.lua`
- `__resource.lua`
- `config.lua`
- `config/**`
- `shared/config.lua`
- `locales/**`
- `lang/**`
- `languages/**`
- `translations/**`
- Markdown and text documentation
- JSON
- SQL
- CFG
- INI files

`escrow_ignore`

During protected build generation, Nord Forge reads `fxmanifest.lua` or `__resource.lua` and honors `escrow_ignore` rules.

Example:

```
fx_version 'cerulean'
game 'gta5'

escrow_ignore {
    'config.lua',
    'shared/config.lua',
    'locales/*.lua',
    'data/*.json'
}
```

Files matched by `escrow_ignore` remain clean and editable.

Extra public patterns can also be configured at platform level with:

```
NORD_FORGE_PUBLIC_FILE_PATTERNS=
```

Bundled FiveM License Resource

Bundled FiveM License Resource

The source package includes:

```
integrations/fivem_resource/nord_forge_license
```

This resource provides a reusable FiveM-side Nord Forge license integration.

Default configuration

```
Config = {}
Config.ForgeApi = 'https://nord-forge.lunarocketeer.com'
Config.ForgeKeyConvar = 'nord_license'
Config.KeymasterConvar = 'nord_keymaster'
Config.AutoBind = false
Config.Debug = false
```

`AutoBind` is disabled by default, matching the production requirement that customers register servers manually in Forge.

Public exports

```
exports.nord_forge_license:ValidateProduct(product, callback)
exports.nord_forge_license:CreateSession(product, buildId, callback)
exports.nord_forge_license:IsValidated(product)
exports.nord_forge_license:GetSessionKey(product)
```

Console command

The integration also includes the `nordforge_check` server command for validation checks.

Licensing API Endpoints

Licensing API Endpoints

Nord Forge exposes licensing endpoints used by protected builds and integrations.

Validate

```
POST /api/license/validate
```

Validates the Forge Key, product and registered server.

Session

```
POST /api/license/session
```

Validates the license and creates a temporary protected-build session.

Register customer server

```
POST /api/license/keymasters
```

Authenticated customer endpoint used to register a Keymaster.

List customer servers

```
GET /api/license/keymasters
```

Returns active/removed Keymasters and slot usage.

Remove server

```
POST /api/license/keymasters/<id>/remove
```

Removes a Keymaster from active use.

> The licensing API should normally be used through the official Nord Forge integration rather than reimplemented unnecessarily.

09 — Tebex Bridge

Workspace Tebex setup, automatic activation and webhook behavior.

Configuring Tebex

Configuring Tebex

Tebex can be configured per developer workspace.

Users with product or license management permissions can access the Tebex Bridge configuration.

Workspace settings

The integration supports:

- Enabled/disabled state
- Tebex Headless/public token
- Webhook secret
- Store URL
- Success URL
- Cancel URL

Nord Forge exposes the workspace's webhook URL for use in Tebex.

Environment fallback

Platform-level fallback values can be provided with:

```
TEBEX_HEADLESS_TOKEN=
TEBEX_PUBLIC_TOKEN=
TEBEX_WEBHOOK_SECRET=
```

Per-workspace configuration should be preferred when multiple developer stores use the same Forge platform.

Product Tebex Setup

Product Tebex Setup

To sell a product through Tebex:

1. Open the product editor.
2. Set **Payment** to `Tebex`.
3. Enter the correct **Tebex Package ID**.
4. Configure the product license type and duration.
5. Make sure checkout is enabled.
6. Configure the workspace Tebex Bridge.
7. Publish a release with a file.
8. Publish the product.
9. Test checkout and webhook activation.

After a valid payment webhook, Nord Forge can automatically associate the product with the customer's Forge Key.

10 — Discord, Support & Documentation

Optional Discord integration, support workflows, docs and Website Builder.

Forge Discord Bot

Forge Discord Bot

Nord Forge includes an optional Discord bot integration.

The bot is intentionally decoupled from the core Forge platform.

If Discord or the bot process is offline, the following continue to work:

- Marketplace
- Licenses
- Releases
- Downloads
- Documentation

Supported bot features

- Release/changelog update notifications
- Release file replacement events
- Optional purchase-completed events
- Discord ticket creation
- Ticket panel publishing
- Bot heartbeat/status in Forge

Workspace connection

Workspace owners connect Discord from the infrastructure/Discord area using an OAuth bot invitation.

The platform administrator configures the single global Forge Bot application and token.

Discord Workspace Setup

Discord Workspace Setup

After the platform administrator configures the global Forge Bot:

1. Open the workspace.
2. Go to **Infrastructure → Discord**.
3. Click the bot invitation action.
4. Select the Discord server.
5. Complete the OAuth callback.
6. Configure the updates channel.
7. Configure the ticket panel channel.
8. Configure the ticket category.
9. Optionally configure customer/developer roles.
10. Save and send a test message.

The integration can cache Discord channels, categories and roles so configuration remains readable even if the bot becomes temporarily unavailable.

Support & Purchase Conversations

Support & Purchase Conversations

Customers can open product support from their authorized product.

Manual purchases also use a conversation thread between buyer and seller.

Depending on permissions, workspace users can:

- View purchase threads
- Reply to customers
- Handle support
- Grant product access
- Manage licenses

Workspace owner-only actions include closing or deleting purchase threads in the current flow.

Product access should only be granted once the corresponding product has a published release file.

Workspace Documentation

Workspace Documentation

Nord Forge includes an internal workspace documentation system.

Documentation records can be linked to a workspace and optionally to a product.

A documentation page can include:

- Category
- Slug
- Title
- Content
- Publication status

Developers with `docs.manage` permission can create and update documentation.

Published store content can expose documentation alongside products where supported by the public store interface.

Website Builder

Website Builder

Nord Forge includes a Website Builder for developer workspaces.

It supports draft and published pages and clean public URLs.

Examples:

```
/nord-admin
/nord-admin/about
/nord-admin/docs
```

A workspace website can contain multiple pages under one `siteSlug`.

The builder supports:

- Page list
- Visual content editing
- Page properties
- Draft/published state
- Light, dark or automatic site themes
- Store subdomain/public route support when DNS and reverse proxy are configured

Published websites are independent from the logged-in Forge UI theme.

11 — Administration & Deployment

Self-hosted deployment, environment variables, maintenance, backups and platform administration.

Self-Hosted Installation

Self-Hosted Installation

Nord Forge is a Python/Flask application with a separate optional Discord bot process.

Quick start

```
python3 -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
cp .env.example .env
python run.py
```

Put Nord Forge behind the reverse proxy/domain configured for the deployment.

Docker

The package also includes:

- `Dockerfile`
- `docker-compose.yml`

The Docker setup can run:

- `nord-forge` — web platform/API
- `nord-forge-bot` — optional Discord bot

Data and storage

Important paths include:

```
data/
storage/
backups/
```

Back up both the database and resource storage in production.

Important Environment Variables

Important Environment Variables

Core configuration includes:

```
NORD_FORGE_SECRET=change-me-super-secret
NORD_FORGE_CRYPTO_SECRET=change-me-build-encryption-secret
NORD_FORGE_DB=data/nord_forge.db
NORD_FORGE_STORAGE=storage
NORD_FORGE_PUBLIC_URL=https://your-forge.example.com
NORD_FORGE_PUBLIC_ROOT_DOMAIN=your-forge.example.com
```

Licensing

```
NORD_FORGE_MAX_KEYMASTERS=3
NORD_FORGE_ALLOW_KEYMASTER_AUTO_BIND=0
NORD_FORGE_PUBLIC_FILE_PATTERNS=
```

Keep automatic Keymaster binding disabled in normal production if customers must register servers manually.

Discord

```
DISCORD_CLIENT_ID=
DISCORD_CLIENT_SECRET=
DISCORD_PUBLIC_KEY=
DISCORD_INVITE_PERMISSIONS=117776
DISCORD_OAUTH_CALLBACK_URL=https://your-forge.example.com/api/discord/invite/callback
NORD_FORGE_BOT_TOKEN=
DISCORD_BOT_TOKEN=
FORGE_API_URL=https://your-forge.example.com
FORGE_BOT_POLL_SECONDS=8
```

Tebex fallback

```
TEBEX_HEADLESS_TOKEN=
TEBEX_PUBLIC_TOKEN=
TEBEX_WEBHOOK_SECRET=
```

Never commit production secrets to source control.

Network Admin

Network Admin

The first active account in a new installation receives the global `network_admin` role.

Network administrators can access platform-level administrative functions such as:

- Platform overview
- Maintenance mode
- User administration
- Workspace administration
- Update management
- Backups
- Global Discord Bot configuration
- Realtime/admin metrics

Workspace roles are separate from the global network administrator role.

Maintenance & Backups

Maintenance & Backups

Nord Forge includes platform maintenance and backup functionality.

Maintenance mode

When maintenance mode is enabled, normal users can be blocked from logging in while `network_admin` users retain administrative access.

Backups

The admin area includes backup listing, creation and download routes.

In production, also maintain external backups of:

- Forge database
- Product/release storage
- Environment configuration
- Reverse proxy configuration
- Any published site assets

Do not rely on a single local backup copy.

12 — Troubleshooting & Reference

Common problems, production checklist and quick technical references.

Customer Troubleshooting

Customer Troubleshooting

Product is not in My Products

Check that:

- The purchase or grant completed successfully.
- The license is active.
- You are signed into the correct Nord Forge account.

Resource says server is not registered

Add the server's `sv_licenseKey` in **Forge → Servers**.

Then confirm:

```
setr nord_license "FG-XXXX-XXXX-XXXX-XXXX"
setr nord_keymaster "cfxk_xxxxxxxxxxxxxxxx"
sv_licenseKey "cfxk_xxxxxxxxxxxxxxxx"
```

New server after migration

Register the new `cfxk_` value in Nord Forge before starting protected resources.

Remove the old server if you no longer need it.

Server limit reached

Remove an old/unused registered server or contact the Forge administrator if the configured platform limit needs to change.

Product license revoked or suspended

Contact the product workspace/developer. Server registration does not override an inactive product license.

Developer Troubleshooting

Developer Troubleshooting

Product cannot be published

A published release with a file is required.

Create the release first, then publish the product.

Release cannot be created

Every release upload requires a file.

Customer has a valid Forge Key but the product fails

Confirm that the product is actually assigned to the customer and that the license status is active.

Customer gets `product_not_found_for_...`

Confirm that the product slug/resource name used by the protected build matches the product in Forge.

Tebex payment does not activate

Check:

- Workspace Tebex Bridge is enabled.
- Token/store configuration is correct.
- Webhook secret is correct.
- Webhook endpoint is configured in Tebex.
- Product Tebex Package ID matches.
- The customer can be resolved to the correct Forge account.

Discord controls are empty

Confirm the global Forge Bot is configured and invited to the workspace's Discord server. Cached channel/role data may still appear when the bot is temporarily offline.

Production Checklist

Production Checklist

Use this checklist before releasing Nord Forge publicly.

Platform

- Change `NORD_FORGE_SECRET`.
- Change `NORD_FORGE_CRYPTO_SECRET`.
- Configure the public HTTPS URL.
- Configure database/storage paths.
- Verify backups.
- Verify reverse proxy and SSL.

Licensing

- Keep `NORD_FORGE_ALLOW_KEYMASTER_AUTO_BIND=0` if manual server registration is required.
- Confirm `NORD_FORGE_MAX_KEYMASTERS`.
- Test Forge Key validation.
- Test server registration.
- Test server removal.
- Test suspended/revoked product access.

Products

- Confirm products stay Draft without a release.
- Confirm a release file is required.
- Confirm customer download resolves the latest published release.
- Test protected script build generation.
- Confirm configuration/locales remain editable.

Payments

- Test free product delivery.
- Test manual purchase/grant flow.
- Test Tebex checkout and webhook activation.

Integrations

- Test Discord OAuth invite.
- Test update message.
- Test ticket panel.
- Verify Forge remains usable with the bot offline.

FiveM Integration Quick Reference

FiveM Integration Quick Reference

Required customer configuration

```
setr nord_license "FG-XXXX-XXXX-XXXX-XXXX"
setr nord_keymaster "cfxk_xxxxxxxxxxxxxxxx"
sv_licenseKey "cfxk_xxxxxxxxxxxxxxxx"
```

Register the server

Before starting a protected product:

```
Customer Portal → Servers → Add Server
```

Register the same `cfxk_` Keymaster used by the server.

Main exports

```
exports.nord_forge_license:ValidateProduct(product, callback)
exports.nord_forge_license:CreateSession(product, buildId, callback)
```

```
exports.nord_forge_license:IsValidated(product)
exports.nord_forge_license:GetSessionKey(product)
```

Main API endpoints

```
POST /api/license/validate
POST /api/license/session
POST /api/license/keymasters
GET  /api/license/keymasters
POST /api/license/keymasters/<id>/remove
```

Default server cap

```
3 active Keymasters per Forge Key
```