07 — Products & Releases

Creating products, release requirements, publishing and customer distribution.

- Creating a Product
- Creating Releases
- Publishing a Product
- Managing Existing Releases

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.