Page contents

- Keep a clean game-data baseline
- Identify the metadata and entry point
- Choose append, override or owned file deliberately
- Package and preserve the result
- Coverage still needed
- Sources and provenance

Offline copy of Mod layout, loading and safe file changes .

Noita: mod layout, loading and safe file changes

Documentation status: In progress. **Source checked:** 2026-10-06. **In-game validation:** not recorded. This chapter extends the preserved first-mod guide. Nolla Games' modding repository identifies installed `tools_modding` files and example mods as the local reference; its online basics and Lua pages are largely TODO, so exact APIs must be checked against the installed game and the Noita Wiki's modding references.

Keep a clean game-data baseline

The developer's Getting started page points to `Noita/tools_modding` and the installed example mods. It describes a `data_wak_unpack.bat` workflow for examining editable game data. Treat the unpacked files as a reference copy: record the installed game build and keep a separate backup before testing replacements. The examples named by that page include `example`, `nightmare`, `starting_loadouts` and `translation_fi`; inspect the copies actually shipped with your build before adapting one.

Your new mod lives under `Noita/mods/<unique-folder>/`. The Noita Wiki basics separates `files/` for mod-owned assets from `data/` for paths that override game data. Begin with only `mod.xml`, `init.lua` and your own `files/` content. A `data/`

override changes the same virtual path used by the game and may conflict with another mod replacing it.

Identify the metadata and entry point

The mod.xml reference describes a single `<Mod/>` metadata element with `name` and `description`. Some attributes grant extra capabilities or declare a game mode or translation. Add those only for a feature that needs them; the reference warns that unrestricted Lua API access could change in the future. Do not copy another mod's identity or request broad access for a simple load check.

Use the first-mod guide to make `init.lua` print one startup message. Enable the mod in the game's mod menu, restart when prompted, and verify the message before adding gameplay code. If it does not appear, inspect XML syntax, folder location, enabled state, script path and the game's log. This is a load check, not proof that a spell, entity or biome is registered.

Choose append, override or owned file deliberately

Noita's developer modding basics outline names `ModLuaFileAppend`, `ModMagicNumbersFileAdd` and `ModMaterialsFileAdd` as distinct extension points. The community basics page describes Lua append and mirrored `data/` overrides. Before changing a vanilla path, locate its source in the installed data and follow any Lua/XML/asset references it loads. If an append point exists for the intended change, test that first. Use a mirrored replacement only when you understand what else the file contains and have a compatibility plan for another mod touching it.

For a controlled experiment, change one value, load with only your mod, and record the visible result. Then load with a second mod that affects the same system. Document which file or callback owns the result and whether load order matters. Restore the unmodified baseline between tests so a stale override does not masquerade as a script bug.

Package and preserve the result

Archive your mod folder, source notes, the Noita build used for testing, and a list of every overridden game path. Confirm a clean installation loads the folder without relying on your unpacked reference data. The archive should contain your own code and assets; keep the game-data extraction as a local development aid rather than placing copied game archives into a distributable package. The developer repository's publishing notes are an older reference; verify current Workshop steps in the installed tools and Steam interface before release.

Coverage still needed

Dedicated chapters are still needed for the entity-component model, actual spell/wand/perk examples, material and biome hooks, translation, sprites/audio, settings, load-order conflict tests and current Workshop publishing. The developer's Lua scripting page points to `tools_modding/lua_api_documentation.txt`; use that versioned local file for signatures. None of those gameplay systems has been independently tested for this chapter.

Sources and provenance

- Nolla Games modding repository, Getting started, Modding basics outline, and Lua scripting pointer — developer-maintained setup and local-reference pointers; checked 2026-10-06. Several online pages are explicitly incomplete.
- Noita Wiki basics and mod.xml — community-maintained folder and metadata detail; checked 2026-10-06. The mod.xml page is marked a stub.
- This chapter is original ModDocs prose; no game assets, extracted data or third-party document text are bundled.