Guide contents

Set up the editor

Choose the correct content layer

Verify in the game

Prepare a release

Sources / References

Documentation: In progress **Quick start: In progress**

Offline copy from ModDocs. Source links remain clickable.

Divinity: Original Sin 2 modding

Larian's **Divinity Engine 2** is the editor for Original Sin 2. It can make add-ons, levels, visual changes and story content. The Divinity Engine wiki provides separate setup, first-project and topic guides. Start with an **add-on**, which can extend the main campaign, rather than an unrelated level project if your aim is a small gameplay change.

> **Quick start:** create an add-on project, make one visible edit, publish it locally, enable it in the game and verify it on a test save. Do not overwrite the original campaign files.

Set up the editor

Install The Divinity Engine 2 for the game edition you own. Follow the Steam editor setup or the platform-specific setup that matches your installation. The editor asks for a game data path; point it at the installed Original Sin 2 data, not at an arbitrary mod folder. If it cannot load projects, the Larian FAQ covers common path and data problems.

Create a distinct project name and choose the correct project type. Larian's first add-on guide walks through this. Save the empty project, close the editor, reopen it and confirm the project still loads before editing a level or story file.

Choose the correct content layer

Aim	Useful starting guide	First test
Change the main campaign	Modding the main campaign	One visible change in a known level
Make a standalone area	Guide index and level tutorials	Enter and exit a tiny playable area
Change story behavior	Story guides	One event and its result
Change visuals or text	Visual and translation guides	Correct resource appears in game

For a campaign add-on, select the **Add-on** project type and target the existing story data as described by Larian's campaign guide. Open a known level, make a small change and publish the mod locally. Avoid changing several systems at once; otherwise a bad resource path or story condition can be difficult to isolate.

Verify in the game

Larian's first add-on walkthrough describes where an external mod is placed and how to check it in game. Enable only the new mod in a test profile, launch a test save and navigate to the changed content. Confirm it appears, then reload the save and confirm the change persists. If it is a level, test collision, entrances, exits and AI navigation. If it is a story change, test the trigger and at least one completion path.

When a project works in the editor but not the game, first check that it was published and enabled. Next inspect project type, target campaign, mod order and missing assets. The Larian FAQ notes that a level often needs AI grid generation for actors to navigate; a visually complete scene can still be unplayable.

Prepare a release

Test the published mod from a clean profile with the development project disabled or separated. Describe the required game edition and any dependent mods. Distribute your project's output rather than copied game data. Keep project sources for updates and a changelog for save compatibility.

This page is **unfinished**: it does not yet provide an end-to-end story script, custom character, asset-export workflow or multiplayer compatibility test. The Larian guide index is the source for those specialized paths.

Sources / References

External project and documentation links cited in this guide. Check the upstream release and version before following version-sensitive steps.

- Divinity Engine wiki
- Steam editor setup
- Larian FAQ
- first add-on guide
- Modding the main campaign
- Guide index