Guide contents

Choose the kind of change

Install REDkit and generate the depot

Find and edit game content

Write a script-only experiment

Test in editor and game

Cook, export and release

Current coverage and sources

Sources / References

Documentation: In progress Offline copy from ModDocs. Source links remain clickable.

The Witcher 3 modding with REDkit

The Witcher 3 REDkit is CD PROJEKT RED's PC editor for creating and packaging game content. The official user guide is the authority for version-sensitive controls. This library follows a project from installed game and uncooked depot through editing, script development, testing and release. ModDocs has not recorded an independent in-game run of these procedures.

Choose the kind of change

Decide whether the mod changes a script, an existing resource, or a world and quest. A script-only experiment can use the game's `mods` folder and a `.ws` file. REDkit projects are appropriate when the change needs editor assets, worlds, scenes or cooking. Keep the project and game installation distinct so you can identify authored files, generated files and base-game content. Record the game build, storefront, REDkit build and test save before editing.

Install REDkit and generate the depot

Install the PC game and obtain REDkit through Steam, GOG or Epic. Launch from the storefront; the official setup page documents `bin\x64_RedKit\editor.exe` as a fallback. On first launch choose the installed game path and a separate writable depot path. Generate the depot and wait for REDkit to report that it is valid. The depot contains uncooked resources for editing; it is not a mod release. Do not replace original game files with depot outputs.

Create a named mod project. REDkit distinguishes a workspace for project assets from a publish area for final output. Put both outside the original game and depot. Back up the project before large migrations and retain a manifest with version, source asset, intended change, dependencies and test result. See Editor, depot and world workflow for the detailed path to a controlled world test.

Find and edit game content

Use REDkit's Asset Browser to inspect an existing resource before adding or overriding it. Note its virtual path and the resources that refer to it. For world content, open a `.w2w` file through File > Open World or the Asset Browser under `virtaldepot > levels`. The editor initially shows limited terrain and vegetation; load the required group in Scene Explorer before deciding objects are absent. Change one defined asset or scene at a time, save, and inspect the first editor or cooking error before making another change. A placed object alone does not establish a working quest: dialogue, actors, conditions, transitions and journal state must be connected and tested as a workflow.

Write a script-only experiment

The official WitcherScript guide identifies `.ws` as the game scripting format. The first script mod procedure puts authored files under `mods/modYourName/content/scripts/local/`; the mod directory must begin with `mod`. Start with a uniquely named console-callable `exec function`, check for compilation errors on game launch and invoke it through the developer console. Keep this in a disposable test mod. An editor project and a directly installed script mod have different output and testing paths. See WitcherScript first mod.

Test in editor and game

For editor content, choose the correct `.redgame` game definition. The official editor playtest page distinguishes a streaming launch at the world origin from a loaded-layer launch at the editor camera. Check which mode you used; a missing object in an unloaded layer is not proof that the resource failed. Test the smallest repeatable action first, then record entry, exit, save/reload and repeat behavior where relevant.

After cooking, install the exported project in a separate test game or clean profile. Test with no other mods first, then with a documented conflict set. Verify the installed output itself; successful editor playback does not establish that the release archive contains every required file. Preserve the pre-test save and a copy of the previous mod version for rollback.

Cook, export and release

In REDkit's Publish tab, use **Save and publish mod project** after editor testing. The official publishing procedure describes project metadata, cooking, and separate export, local install and Workshop choices. Cooking gathers changed files and builds bundles, caches and metadata in packed output. Inspect the package rather than assuming every workspace file belongs in the release. Give the release a unique name and version, install instructions, game/REDkit version, dependency list, changed-resource list and known conflicts. Do not include the uncooked depot or an unmodified copy of the game. See Cooking, installation and release checks.

Current coverage and sources

These pages cover core REDkit setup, world inspection, a first script path, editor testing and export. Detailed quest graphs, dialogue, animation, localization, mesh/material import, DLC mods, API reference and historical version differences still need source-by-source chapters. Procedures are source-reviewed as of 2026-11-11; independent in-game validation is unrecorded. Official sources: REDkit user guide, REDkit introduction, starting the editor, working with the editor, modding guide, and WitcherScript.

Sources / References

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

- official user guide
- official setup page
- WitcherScript guide
- first script mod procedure
- editor playtest page
- official publishing procedure
- REDkit introduction
- working with the editor
- modding guide