Guide contents

Choose the correct source and compiler

Build a clean baseline

Make one content change and test it

Sources / References

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

Offline copy from ModDocs. Source links remain clickable.

Mount & Blade: Warband modding

Warband's **Module System** is a collection of source files compiled into the text files the game loads. The source files use Python syntax and a game-specific operation language; Warband does not execute those Python files itself. A scene-editing route also exists inside the game, but world-map editing is a separate workflow. The Module System overview and setup guide explain the difference.

> **Quick start:** copy the Native module to a uniquely named module, point a matching Module System at that copy, compile without edits, and confirm the copy starts in Warband. Then make one visible item or troop change in a `module_*` file and recompile.

Choose the correct source and compiler

Get a Module System release matching your Warband installation rather than an arbitrary tutorial download. The older wiki describes a 1.134-era source tree and Python 2.5/2.6; the later setup guide describes a Python 2.7 workflow. These are **historical, version-specific instructions**. Confirm the bundled build scripts and Python requirement of the exact Module System you choose. Do not install an old interpreter system-wide merely because a tutorial says to; an isolated build environment is safer.

The four major filename prefixes have different jobs:

Prefix	Role	Usual first-mod action
`module_*`	Editable content such as troops, items and scene declarations	Change a single entry
`header_*`	Operations and constants	Read for reference
`ID_*`	Generated numeric indices	Inspect when diagnosing IDs; do not hand-edit
`process_*`	Build scripts	Leave unchanged for a first mod

The distinction matters because editing generated text or `ID_*` output may appear to work briefly but will be overwritten at the next compile.

Build a clean baseline

Copy Warband's `Modules/Native` folder and rename it to your test module. In `module_info.py`, set `export_dir` to that folder; the setup guide shows the field and build sequence. Verify the path before running `build_module.bat`, because an incorrect export directory could overwrite another module. Keep the source tree outside the game's module folder and make a backup of both before the first build.

Compile **without edits**. If Python is not found, fix the isolated interpreter path and retry. If the build reports a file access error, check output folder permissions. A successful compiler exit only proves files were generated; launch Warband, select the new module, and start a disposable game to confirm that the baseline is playable.

Make one content change and test it

Open the relevant `module_*` file and change one easily identifiable property in a troop or item. Avoid adding a complex new faction as the first test, because it depends on many linked definitions. Recompile and check that the target output file changed in the copied module. Launch that module and inspect the result in a new game. If the

change is absent, first verify the menu selected the copied module and `export_dir` still points to it.

For scenes, use the game's edit mode and learn the difference between a scene and the campaign world map. The Warband modding guide describes edit-mode entry and module file relationships. Test collision, AI paths and entry points, not only the visual layout. World-map geometry needs a separate map editor and should be treated as another project.

Package the compiled module and original assets that your mod uses. Keep the Module System source separately for maintainers, and document the Warband version and third-party dependencies. This guide is **unfinished**: it covers the baseline and first edit, but a fully tested new troop, scene or multiplayer release path has not yet been written here.

Sources / References

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

- Module System overview
- setup guide
- Warband modding guide