Guide contents

Pick the layer you need

Create the module shell

Build and test content in stages

Package a module other players can load

Sources / References

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

Offline copy from ModDocs. Source links remain clickable.

Mount & Blade II: Bannerlord modding

Bannerlord mods are organized as **modules**. A module can add or override game data, assets, scenes and code. TaleWorlds' modding documentation contains a module quick guide, editor references and API documentation. Build a minimal module before attempting a new campaign system or scene collection.

> **Quick start:** create a module with a unique identity, make one small data or asset change, load it in a test profile and verify it in game. Keep module dependencies explicit.

Pick the layer you need

Goal	Main documentation area	First milestone
Data change	XML editor and merging	One change without replacing unrelated data
New or overridden asset	Asset Management	Resource loads without missing references

Goal	Main documentation area	First milestone
Scene	Scene Editor	Traversable scene with correct entry points
Game behavior	API documentation	One isolated behavior change
Multiplayer mode	Multiplayer guides	Local server/test session first

These paths are not interchangeable. A data mod may need only a module and XML; a scene or model requires editor assets; a code mod requires compatible assemblies and careful versioning. Follow the documentation for your selected branch and game version.

Create the module shell

Use the official quick guide to create and register a module. Give it a unique name, version and dependency list. Keep sources in a working directory and do not edit Native or other base-game modules directly. Save an empty module first, confirm the launcher sees it, and start the game with the module enabled. That proves discovery and dependency resolution before content is added.

For a simple data edit, add the smallest valid data file and test one changed value. The XML editor guidance covers merges with Native; copying an entire base-game file for one adjustment can cause conflicts or discard later game updates. If your mod depends on another module, declare that relationship instead of relying on a particular manual load order.

Build and test content in stages

For a scene, create a small area, set its intended mission or scene type, and add required spawn points. TaleWorlds has separate documentation for scene editing and scene spawn points. Test in the actual mission context; a scene visible in the editor can still lack navigation, entry points or mission-specific requirements.

For an asset, import one mesh or texture, check its material and resource path, then verify that an in-game object references it. For code, build against the game version you target and test with only the new module and its declared dependencies. Inspect loader and game logs if initialization fails; an assembly mismatch is a different problem from a missing texture.

Package a module other players can load

Publish the module's own files and required original assets; do not bundle the game's base modules. Test the packaged copy from a fresh profile and remove loose development copies. Document required Bannerlord version, dependencies, whether it changes saves, and any known incompatibilities. TaleWorlds' Workshop instructions explain the distribution route when you are ready to publish.

This guide is **unfinished**. It describes module creation and testing but does not yet contain a locally verified custom scene, C# behavior, asset export or multiplayer example. The official TaleWorlds documentation is the version-specific reference for those deeper routes.

Sources / References

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

- modding documentation
- module quick guide
- API documentation