Guide contents

Understand package scope

Make a small XML change

Multiplayer, release and compatibility

Sources / References

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

Offline copy from ModDocs. Source links remain clickable.

Barotrauma content packages

Barotrauma's main modding format is the **content package**: XML files that register or alter items, creatures, jobs, missions, structures and other game content. This is distinct from modifying the game's source code. The official wiki explains the package manager, load order and Workshop route.

> **Quick start:** create a local content package through the game's Modding interface, add one small change, enable it in Settings, and test the specific object in a disposable campaign or editor. Restart if the game requests it. Keep the first package separate from a valued save or server.

Understand package scope

Content	Main format or tool	Verify
Items, jobs, creatures, missions	XML definitions in a content package	Asset appears and behaves in a test session
Submarines	In-game submarine editor and package	Sub loads, hulls connect, systems work
New graphics and audio	Files referenced from package XML	Paths resolve on another installation

Content	Main format or tool	Verify
Existing-content changes	Override or patch semantics for the chosen XML type	Only intended content is replaced

The official creation guidance recommends a new package rather than editing the installed game's `Content` folder. Updates may delete changes made directly to the game files. Create under the current local-mod folder supplied by your game build, and use the in-game manager to create or register the package where practical. The developer's modding-refactor note explains why modern local and Workshop copies are separated; some older wiki paths predate that refactor.

Make a small XML change

Start from an item or job category whose current XML schema is documented or visible in your installed game. Make a single small change under your own package, using unique identifiers for new content. Register the XML file as the appropriate content type in the package's metadata. For asset references, the developer notes that modern mods use `%ModDir%` to refer to files inside their own package. A path copied from an older mod may resolve on the author's computer but fail after installation.

Enable the package in **Settings**, apply the change and restart when prompted. The content package wiki describes the checkbox and drag handles for load order. Open the relevant editor or start a disposable session to look for the modified object. If it is absent, check whether the package itself is enabled before debugging the XML. Then check that the package lists the file, the XML parses, the identifier is unique or intentionally overrides another entry, and the asset path uses the current convention. Test with just your package and required dependencies to isolate conflicts.

Multiplayer, release and compatibility

Multiplayer clients need content compatible with the host. The official wiki says the host's package load order controls the session, and Workshop installation is the supported distribution route. Test with a second client before publishing a multiplayer

claim. Also test a fresh install of the package, not only the local development folder, because Workshop and local paths differ.

For a submarine, use the submarine editor and test its spawn, hull integrity, power, doors and docking. Merely saving it in the editor does not prove it can survive a normal mission. For a code mod, use a separate guide and loader: XML package instructions do not imply arbitrary C# code can be loaded as a content package.

This guide is **unfinished**: it lays out the modern package workflow but does not yet include a verified current-build XML item example, complete submarine publication test or dedicated-server configuration walkthrough. Use the official content-package reference for those version-specific fields.

Sources / References

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

- official wiki
- official creation guidance
- developer's modding-refactor note