Check Valheim BepInEx mods before updating a server

Build a compatibility list first

A Valheim modded server depends on several versions working together: the game, the BepInEx loader, individual plugins and their dependencies. “Newest” in one category is not proof of compatibility with the others. Read each mod author’s supported game build, loader major version and server/client requirements before changing a working installation.

Make a short inventory containing plugin name, version, download source, required libraries and whether clients also need it. Some plugins are client-only; others need matching client and server components. A server-side feature does not automatically work on consoles or other platforms where clients cannot install the required loader.

Protect the save and the working mod environment

Schedule a maintenance window, stop the server normally and download the world data, startup configuration and existing loader/plugin files. Keep configuration files alongside the matching plugin versions. A world backup alone may not be enough to reproduce modded items or game behaviour after a rollback.

Test on a separate copy where possible. Use a copied save or an explicitly separate test world identifier; do not use the production world as a compatibility experiment. A mod can change saved data even when the server appears to start successfully. Removing its DLL later may not undo those changes.

Install the right loader and a small test set

Choose a BepInEx package matching the game runtime, operating system and architecture, following the loader and Valheim mod-pack documentation for the exact release. Do not replace a working Valheim loader with an arbitrary generic or preview package. Hosted startup integration may also be required: uploading a loader does not guarantee the launch process actually loads it.

Plugin assemblies normally belong under `BepInEx/plugins`, with generated settings under `BepInEx/config`. Preserve any dependency or patcher layout supplied by the author. Avoid uploading a complete desktop game directory over the dedicated server. If the host’s startup needs adjustment, ask support rather than trying operating-system commands through the game console.

Read the loader log and verify gameplay

After a test start, inspect `BepInEx/LogOutput.log` when present, plus the game log. Confirm that the loader starts and that the intended plugin versions load. Missing dependencies, duplicate plugin identifiers and method or type errors point to different problems; save the first error rather than only the final crash line.

Add plugins in small dependency-aware groups. Join using a matching client and test the mod’s actual feature, saving and reconnecting. Compare memory and startup behaviour with the earlier baseline. If the loader never appears in logs, investigate startup integration before repeatedly rearranging plugins.

Return to a known working combination

When an update fails, stop the test, restore the previous plugin/configuration set and use the corresponding untouched world copy if the failed combination saved changes. Do not open a modified world in vanilla simply to see what survives. Keep the incompatible files separately for diagnosis.

For support, provide the game build, loader version, mod inventory and a redacted log excerpt. Remove passwords, access tokens and player identifiers where unnecessary. Report which single change introduced the fault; this is more useful than repeatedly updating every component at once.

Related guides

- Plan a safe game-server update
- Keep a complete Valheim world copy

Further reading

- BepInEx installation documentation
- BepInEx troubleshooting
- Valheim dedicated-server manual

Looking for another game? Browse game server setup and troubleshooting guides.

Revision #2
Created 2026-11-09 09:21:20 UTC by CloudExa Hosting Admin
Updated 2026-11-09 09:21:21 UTC by CloudExa Hosting Admin