Move a Project Zomboid save and preserve its server identity

Separate a server move from a version upgrade

A location change should first reproduce the working server on the same game build and mod combination. Combining a move with a major update makes missing characters, broken mods and save conversion harder to distinguish. Record the exact source version, branch, server profile name and installed Workshop/mod lists before stopping it.

Build 41 saves and mods must not be assumed compatible with Build 42. Early Build 42 releases also had explicit save compatibility breaks. Read the release notes covering your exact source and destination versions. A destination marked “latest” is not a promise that an older world can be opened safely.

Capture a complete stopped snapshot

Ask players to disconnect, stop the server normally and wait until it has exited. Download the complete active world-save directory, associated multiplayer/player data, the named account database and its server settings. Keep an untouched archive outside the service. Check that the download completed and that there is enough destination storage for both the imported data and a recovery copy.

Common layouts place worlds under `Saves/Multiplayer`, account databases under `db`, and named settings under `Server` within the game’s user-data root. Hosting and startup parameters can

change that root. Use the source configuration and logs to identify the actual directories instead of copying the first similarly named folder.

Preserve names and dependencies

A named server profile can have an INI file, sandbox Lua settings and spawn configuration alongside its matching database and save directories. Preserve the complete set and filename case. The public server title is different from the profile name used at launch; changing the profile can make the game select a different world and account database.

Install the matching game build and required mods on the destination before opening the copied world. Stop the destination, keep any existing destination data separately and upload the snapshot without merging unrelated saves. Do not copy only map chunks: player state, vehicles and other persistent systems need their associated data too.

Test the copy before returning to normal play

Start the destination once and confirm the log selects the intended server profile. Join with a compatible client and verify an existing character, familiar location, stored items and world changes. Test at least one ordinary player account as well as the administrator; administrator access can hide an account or whitelist problem.

Perform a normal save, stop and restart, then reconnect. Compare results with the source snapshot. If the server displays a fresh world or new characters, stop immediately and inspect the selected profile, save-root paths and upload nesting. Do not continue playing while deciding which copy should become authoritative.

Keep a clear rollback point

Once a newer game version writes to a save, do not assume an older binary can read it. Rollback means restoring the untouched pre-change data with its matching build, mods and configuration, rather than downgrading software over the converted copy.

Keep the source stopped while testing, and retain its data until the owner confirms the move. If a staged update is required, test each supported conversion on another copy first. Send support the versions, profile name and redacted errors; never send account databases or password-bearing

configuration through a public discussion.

Related guides

- Match Zomboid mods to the installed build
- Create a recoverable game-server backup

Further reading

- Project Zomboid dedicated-server wiki
- Official Build 42 compatibility warning
- Current official release notes

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