Manage Project Zomboid administrators and whitelist access

Know which account you are changing

Project Zomboid has several separate access controls: the shared server password, individual multiplayer accounts, staff access levels and whitelist rules. Changing one does not automatically configure the others. Record the installed game build and active server profile before editing access. Build 42 introduced further multiplayer changes, so check command help on your server rather than relying on an old permissions chart.

Use the player’s exact multiplayer username, including spaces and case where relevant. Their Steam display name and hosting-panel email may be different. Never make a player an administrator merely to solve a connection or whitelist error; first determine which access check is refusing the account.

Keep an account recovery copy

Before a bulk access change, stop the server during a planned maintenance period and copy the account database with the matching named configuration and world save. Do not edit a live database file or replace it with a blank one. Keep the archive private because it contains account and player information.

For an established server, prefer its supported commands or administrator tools over direct database changes. The panel console is a privileged interface: anyone with access may be able to change accounts. Limit panel access separately from in-game staff roles and avoid sharing the owner’s credentials.

Grant a role and verify it

Use the server’s `help` command to inspect supported syntax. In-game administrator commands normally start with a slash; a dedicated-server console generally uses the command without that slash. For example, in-game `/setaccesslevel "ExactUser" "admin"` targets that specific account. Confirm that the account exists before assigning a role.

Choose the least powerful role needed for the job. The developer’s historical role list includes admin, moderator, overseer, gm and observer, but capabilities can change between builds. Read the installed version’s help and test the intended staff action. Do not promise a moderator can or cannot perform a particular action solely from a years-old role list.

Configure private entry without locking out players

For a closed server, review the active INI options governing open registration and automatic whitelist creation, including `Open` and `AutoCreateUserInWhiteList` where supported. A shared server password alone does not create individual accounts. Record the current settings before changing them and announce the entry process to players.

The documented `adduser` command can create a named account for whitelist access; consult local help for its arguments and supply a unique temporary password privately. Treat console and chat credentials as sensitive because they may appear in logs. Change passwords through the supported account workflow and never publish command transcripts containing them.

Test ordinary access and retain a way back

Reconnect as the staff account and confirm the selected role. Separately test a normal allowed account and an unlisted test account so you know the whitelist actually enforces the intended policy. Verify staff access after a normal restart if configuration changes required one.

If access fails, check the active profile, username, account existence and server password before making broader changes. Revert the recorded access settings if needed; do not wipe the database or world to reset permissions. For support, include the build, redacted command response and whether the problem affects one player or every player.

Related guides

- Back up Zomboid accounts and world data
- Manage game-panel access separately

Further reading

- Project Zomboid command reference
- Server options reference
- Developer explanation of access levels

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