Preserve a 7 Days to Die save when moving or changing worlds

Distinguish the map from the saved game

A 7 Days to Die installation can store a generated map separately from the ongoing saved game that uses it. The map describes terrain and locations; the save records the particular playthrough and player progress. Copying only one side can produce a missing-world error or an apparently fresh game. Record both the selected world and the game/save name before moving data.

For random generation, the seed and size influence map creation. They are not a backup of an existing playthrough: regenerating a map from a seed does not restore buildings, containers, vehicles or characters. Generation can also change between game versions, so a seed alone is not a reliable substitute for preserving the original map files.

Find the active data roots

The developer’s Windows guidance places user data under `%APPDATA%/7DaysToDie`, including `GeneratedWorlds` and `Saves`. Hosted Linux paths can differ, particularly where the launch command or configuration specifies a custom user-data directory. Check the active settings and log rather than treating a desktop path as the hosted destination.

Within the save data, preserve the entire relevant world/game folder and its contents, including player and region data. Include the matching generated-world folder when that map is not part of the standard server installation. If the files are outside your accessible file area, ask support to copy them; no root login is needed for ordinary hosted game administration.

Move a stopped, consistent copy

Ask players to leave and stop the source cleanly before downloading. Keep an untouched archive of the map, save and server configuration. Record the exact game build and mods as well as total copy size. A copy made while the server writes data can mix different points in time even when the transfer finishes without errors.

Stop the destination, retain any existing destination data separately and upload the complete copy into the correct data root. Preserve folder names and case. Select the imported world through `GameWorld` and its existing saved game through `GameName`. Do not merge two worlds' region files or rename arbitrary internal files to make the names look similar.

Create a new world without sacrificing the old one

If you want to try another seed, use a clearly different world/save selection in a separate test instance or isolated copied data directory. Confirm storage and memory capacity first. Keep the original files and configuration offline and untouched. Do not delete the current save to force generation or reuse the only production save location for an experiment.

Before generation starts, check the installed version's documented world names and generation settings. A typo in a world or game name may select a new playthrough instead of the intended one. Stop if logs show unexpected generation while you are trying to import an established game.

Verify and retain rollback data

Join after the move and check a known base, game day and existing player progress. Save normally, stop, restart and reconnect to confirm persistence. Only then invite everyone back. Keep the original source stopped while testing to avoid progress diverging between copies.

For a version change, read the release notes and test a duplicate first. If conversion fails, restore the pre-change map, save, configuration and matching software as a set. Keep the failed copy for diagnosis and send support redacted logs rather than deleting evidence.

Related guides

- Edit 7 Days server configuration carefully
- Keep an independent save backup

Further reading

- Developer save-location and migration guide
- Official server property reference

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