Restore Rust+ pairing after a server move without resetting players

If Rust itself works after a move but Rust+ reports the server offline, investigate the companion connection separately. Do not erase player records or regenerate the world to repair the app. You need the game console, access to the active identity folder and the allocation details supplied for your service.

Preserve the existing pairing state

Before changing anything, take a backup and record the identity in use. Facepunch identifies `companion.id` as the server's Rust+ identity and `player.tokens` as player pairing information. Removing tokens forces players to pair again. Keep these files private, and retain them with the server's consistent save set during an authorised migration.

If the move has already happened, compare filenames and backup dates with the old identity. Do not copy files from a different customer's server or merge two identities. When the expected files are missing, ask support to inspect the retained migration copy. Avoid restoring the entire world merely to replace a pairing identifier.

Check the companion listener

1. Run `app.info` in the server console. Record the companion address and port, excluding any tokens from screenshots or tickets.
2. Compare the reported port with your game-panel allocations. The Rust+ documentation explains that the companion listener needs TCP reachability and that `app.port` can select its port.
3. Ask support to provide or confirm a dedicated companion allocation when the reported port is not assigned to you. A client-side panel setting cannot grant use of an arbitrary host port.

4. If support supplies a replacement, set `app.port` to that allocation and schedule a normal restart. The documented port must be at least 10000; do not choose a number merely because it meets that minimum.
5. If registration or push notifications still fail, include the relevant error and timestamp in a ticket. Support should check outbound access to the documented Facepunch companion service and the specific inbound allocation.

Prove the repair

After startup, run `app.info` again. Confirm the expected listener and open Rust+ using an existing pairing. Test a simple read-only action such as viewing server information. Ask one previously paired player to check too. Record whether the app reaches the server and whether notifications work; these are different checks.

If existing pairing works, there is no reason to reset everyone. If only one phone has trouble while other pairings work, investigate that account or app before changing the server identity. A reachable game port alone does not prove that the companion TCP path is healthy.

Rollback and limits

Restore the previous `app.port` setting only if support confirms that allocation is still valid. Retain the backup and leave player data untouched. Companion details may evolve with Rust updates, so compare your console output with the current official documentation. Do not post pairing tokens, raw identity files or an unrestricted firewall rule in public support discussions.

Related help

- Find Rust server.cfg and the correct identity save folder
- Diagnose Rust game, query and RCON connection ports
- Schedule game server backups and check retention

Revision #1
Created 2026-11-09 10:42:41 UTC by CloudExa Hosting Admin
Updated 2026-11-09 10:42:41 UTC by CloudExa Hosting Admin