Grant Rust owner or moderator access using Steam IDs

Rust administrator access is tied to a Steam account, not a display name or game-panel login. Grant the lowest role the person needs, then verify with that account. An owner has substantially more authority than an ordinary player.

Prepare and identify the player

Use a trusted game-panel server console or an already secured RCON connection. Never enter an owner command in public chat. Record the current administrator list privately and back up the active identity configuration before changing access. If you will edit files directly, stop the server first and retain the complete original file.

Ask the intended administrator to join. Run `users` in the server console and match the account carefully. Use its full Steam64 identifier. Similar display names and renamed accounts are common; a remembered name is not enough evidence. Have the person confirm their own profile before assigning a role.

Assign one role

The command forms below use a placeholder. Replace STEAM64 with the verified numeric account ID and use a short descriptive name:

```
moderatorid STEAM64 "Moderator name"
ownerid STEAM64 "Owner name"
```

Run only the command for the intended role. Rust distinguishes moderator authentication level 1 from owner authentication level 2. Giving everyone owner access to bypass a plugin error expands their control over the server and does not establish which permission was missing.

Reconnect and verify

1. Have the player disconnect completely and reconnect after the assignment.
2. Test a harmless administrative action appropriate to the role. Avoid testing item creation, player bans or destructive world commands on a live community.
3. Check the console for the result and confirm an unrelated ordinary player has not gained administrator behaviour.
4. Record who approved the access and when it should be reviewed.

Current Facepunch instructions state that these administrator changes are saved automatically. Do not repeatedly overwrite configuration files to force persistence. If access disappears after restart, investigate the active identity and any management process regenerating its configuration.

Separate Rust access from plugin permissions

Oxide plugins may require their own named permission even when the user is a Rust administrator. Read that plugin's documentation and inspect the user's Oxide permissions. Grant only the relevant permission. Conversely, adding a cosmetic chat title should not require making someone a Rust owner.

Recover from a wrong assignment

Keep the console available while testing so you do not depend on the newly authorised player to recover access. If the wrong account was authorised, stop further testing and remove that account using the current Facepunch administrator guidance or ask support to verify the saved administrator configuration. Restore a retained configuration only during a controlled stop, accounting for any legitimate access changes since the backup.

Support needs the service ID, timestamp, intended role and redacted console result. Do not publish the full administrator list, RCON password or another customer's account information.

Related guides

- Connect securely through RCON

- Manage Oxide groups and permissions

Further reading

- Facepunch: creating a Rust server
- uMod: plugin permissions

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