WorldGuard tasks

Task guides and troubleshooting for WorldGuard. Check the installed release before changing configuration or data.

- Give a WorldGuard plot member access without making them its owner
- Extend a WorldGuard region vertically without replacing its members
- Remove a WorldGuard greeting without confusing blank and inherited values
- Restrict entry to a WorldGuard region’s approved members
- Test WorldGuard chest access separately from block building
- Verify a WorldGuard no-PvP area at its actual boundary
- Find the correct WorldGuard region when names repeat across worlds
- Change a WorldGuard parent without losing track of inherited membership
- Limit a WorldGuard mob-spawn restriction to one region
- Review a WorldGuard blocked-command rule without relying on it for permissions

Give a WorldGuard plot member access without making them its owner

A region member can receive ordinary access without being granted the owner’s region-management powers. Confirm the intended role before adding someone to a shared build.

Follow the procedure

1. Stand in the correct world and inspect the plot with `/rg info REGION`. Record its owners, members, bounds and parent before changing the domain.
2. Confirm the invited player’s exact identity and the plot owner’s authorization. Check whether the player already belongs through a group entry or inherited parent membership.
3. Use the documented `/rg addmember REGION PLAYER` command for ordinary plot membership. Do not use addowner merely because the member cannot perform one unrelated action.
4. Have the invited non-operator player test an approved build action inside the plot and attempt the same action outside it. Keep the trial away from valuable blocks.
5. Check whether region management commands remain denied to the member under your permission policy. Membership and the server’s command permissions are separate controls.
6. Inspect the region again and record the new member. If the invitation is temporary, schedule a documented review rather than assuming region membership expires automatically.

Check the result

The invited player should receive the intended plot access while remaining unable to manage region ownership unless separately authorized. Verify both inside and outside behaviour. An owner account’s successful test cannot establish that the member role is limited correctly.

If the result is wrong

If access remains denied, inspect relevant flags, parent membership and other protection plugins before changing ownership. Remove only the newly added member if the invitation was mistaken. Do not remove all domain entries or raise region priority to fix an unidentified permission issue.

References and related help

- WorldGuard maintainer reference
- WorldGuard maintainer reference
- Protect a Minecraft spawn with WorldGuard regions
- Fix overlapping WorldGuard region flags with a small permission test

Extend a WorldGuard region vertically without replacing its members

A region selected too shallowly may protect the floor while leaving an upper build exposed. Redefine the bounds carefully so the region identity, membership and policy remain intact.

Follow the procedure

1. Inspect the existing region in the correct world with `/rg info REGION`. Save its bounds, flags, parent, priority, owners and members before selecting new geometry.
2. Use the documented WorldGuard selection command to load the region into your WorldEdit selection. Confirm the selected world and both horizontal corners.
3. Adjust only the intended vertical bounds with supported WorldEdit selection controls. Choose the actual required height range rather than assuming every world uses the same build limits.
4. Check the new volume and any neighbouring protected areas. A full-height expansion may protect more underground or aerial space than the plot owner intended.
5. Use the documented `/rg redefine REGION` command on the reviewed selection. Reinspect the region immediately and compare flags and domain entries with the saved values.
6. Test a disposable build action just inside the new upper boundary and just outside it using an ordinary player. Check a previous protected point to confirm old coverage remains.

Check the result

The region should cover the revised vertical volume while preserving its existing identity and recorded policy. Verify the geometry and player behaviour together. Seeing the same region name in a list does not prove the new selection was in the intended world.

If the result is wrong

If the bounds are wrong, reselect the saved original volume and redefine that same region. Do not delete and recreate it as a shortcut, since that risks losing domain and flag settings. Restore from a coordinated region backup only if the recorded configuration cannot be recovered directly.

References and related help

- WorldGuard maintainer reference
- WorldGuard maintainer reference
- Protect a Minecraft spawn with WorldGuard regions
- Fix overlapping WorldGuard region flags with a small permission test

Remove a WorldGuard greeting without confusing blank and inherited values

A blank region greeting and an unset greeting are different states. Unsetting can reveal an inherited message, so decide whether the region should inherit or deliberately show nothing.

Follow the procedure

1. Inspect the region’s current flags, parent and overlapping regions at the tested location. Record the exact greeting that appears and the account used to observe it.
2. Read the installed WorldGuard flag-command syntax. The documented command can unset a flag by omitting its value, while the empty-value option represents an explicit blank string where supported.
3. Choose the intended result: inherit the applicable greeting policy or suppress this region’s message with a supported blank value. Do not remove the parent merely to silence text.
4. Apply the narrow greeting change to the named region in the correct world. Keep the previous text privately so it can be restored exactly if needed.
5. Leave the region fully and re-enter with an ordinary player. Check both chat greeting and title-related flags if your server uses more than one greeting surface.
6. Inspect the stored flag value again and record whether it is unset or explicitly empty. Test another region to ensure its greeting remains unchanged.

Check the result

The player should see the intended message behaviour when crossing the boundary, and the stored state should match your inheritance decision. Testing while standing still inside the region may not trigger a new greeting. Check an actual exit and re-entry before reporting success.

If the result is wrong

If a message remains, identify its source region or another join-message plugin before clearing more flags. Restore the original greeting if the change suppresses useful messages elsewhere. Avoid editing region files while the running server may later save over them.

References and related help

- WorldGuard maintainer reference
- WorldGuard maintainer reference
- WorldGuard maintainer reference
- Protect a Minecraft spawn with WorldGuard regions
- Fix overlapping WorldGuard region flags with a small permission test

Restrict entry to a WorldGuard region’s approved members

A members-only area needs an entry policy as well as build protection. Check the region group and movement routes so the rule does not accidentally lock out its own members.

Follow the procedure

1. Inspect the target region, its members, owners and relevant parent. Confirm that the intended users actually resolve as members rather than relying on a similar permission-group name.
2. Read the entry flag and region-group documentation for your release. Choose the documented group restriction that applies the denial to nonmembers, not to everyone.
3. Apply the reviewed entry flag using the installed `/rg flag` syntax. Keep an administrator outside the area with a recovery path before testing access.
4. Use one ordinary member and one nonmember to walk across the boundary. Repeat from more than one side if the region has unusual geometry.
5. Test the teleport routes players are meant to use, including a warp or home if applicable. Check interactions with other plugins rather than assuming a walking test covers every transport route.
6. Remove any temporary test memberships and record the approved access list. Confirm ordinary movement outside the region remains unaffected.

Check the result

The member should enter through the approved routes while the nonmember is refused according to the documented policy. Verify an exit path as well. A protected build that still permits entry is a different policy and should not be mistaken for a malfunction.

If the result is wrong

If legitimate members are blocked, restore the previous entry flag and inspect domain membership and group selection. Do not grant bypass to all players or delete the region to release one account. Use the retained administrator recovery route while correcting the specific rule.

References and related help

- WorldGuard maintainer reference
- WorldGuard maintainer reference
- WorldGuard maintainer reference
- Protect a Minecraft spawn with WorldGuard regions
- Fix overlapping WorldGuard region flags with a small permission test

Test WorldGuard chest access separately from block building

A player may be able to open storage while being unable to build, or the reverse. Test container access directly instead of treating a successful block-protection check as complete spawn protection.

Follow the procedure

1. Identify the exact region and a disposable chest inside it. Record current flags, membership and any separate chest-lock plugin that may also control the container.
2. Read the chest-access and interaction flag descriptions for the installed WorldGuard release. Avoid setting the general build flag merely to solve a container-specific problem.
3. Prepare an ordinary nonmember, an approved member and a counted set of disposable chest items. Keep operator and bypass permissions off the test accounts.
4. Apply the intended narrow container policy and have each account try opening the chest, taking one trial item and returning it. Record each result separately.
5. Test a normal block action beside the chest and another container outside the region. This checks that the change has not accidentally widened building rights or global storage access.
6. Reinspect the region and confirm the final trial item count. Document whether a chest-lock plugin adds another ownership requirement beyond WorldGuard membership.

Check the result

Container access and build access should each match the intended policy, with no missing trial items. A successful chest-open animation alone does not prove withdrawals are allowed or denied. Test the actual permitted transaction with disposable contents.

If the result is wrong

If results conflict, inspect the effective region flags and other storage-protection plugins before granting broader access. Restore the previous chest-related flag when needed. Do not remove container ownership, disable all protections or test by taking valuable items from a real player’s storage.

References and related help

- WorldGuard maintainer reference
- WorldGuard maintainer reference
- Protect a Minecraft spawn with WorldGuard regions
- Fix overlapping WorldGuard region flags with a small permission test

Verify a WorldGuard no-PvP area at its actual boundary

A no-PvP area should protect the intended space without disabling combat elsewhere. Check the boundary with ordinary accounts and distinguish direct attacks from other damage sources.

Follow the procedure

1. Inspect the region’s bounds, relevant flags and world. Note any arena or combat plugin that may independently control damage or delay escape from combat.
2. Read the WorldGuard PvP flag description for your installed release. Apply the intended value to the named region rather than changing the whole world’s combat settings.
3. Use two consenting non-operator test accounts with no valuable items. Choose low-risk attacks and keep a safe recovery area available.
4. Test a direct player attack well inside the region, then outside its boundary. Repeat near the edge with each attacker and victim position recorded.
5. Check the specific projectile or combat interaction your players report. Do not infer protection against fire, fall damage or every plugin skill from one blocked melee hit.
6. Review any neighbouring regions and combat-plugin rules only if the observed result requires it. Record which attack routes were tested and what remains outside this policy.

Check the result

Direct PvP should follow the selected region policy in the tested positions while combat outside remains as intended. Keep the attacker and victim locations with the result. A blocked attack from an administrator account can be distorted by bypass or unrelated game-mode rules.

If the result is wrong

If damage still occurs, identify the source and controlling plugin before broadening flags. Restore the prior region setting if legitimate combat outside the intended area changes. Avoid treating no-PvP as universal invulnerability; use a separately reviewed damage policy where that is the actual

requirement.

References and related help

- WorldGuard maintainer reference
- WorldGuard maintainer reference
- Protect a Minecraft spawn with WorldGuard regions
- Fix overlapping WorldGuard region flags with a small permission test

Find the correct WorldGuard region when names repeat across worlds

WorldGuard region names belong to worlds, so the same label can describe different areas. Check the world explicitly before editing a spawn or plot with a familiar name.

Follow the procedure

1. Collect the affected player’s world and coordinates from an authorized source. A screenshot of a region name alone is insufficient when multiple worlds use the same naming scheme.
2. Use the documented world option with a read-only region query, such as the installed `/rg info` form using `-w WORLD`. Confirm syntax with command help on your release.
3. Compare the reported bounds with the player’s location. If necessary, list the relevant world’s regions rather than assuming the first result is the right one.
4. Inspect the same region name in any other likely world and record the differences. Keep all queries read-only until the intended target is certain.
5. For the eventual change, include the world explicitly where supported or move to the correct world and verify again. Record both world and region in the staff instruction.
6. Have an ordinary player repeat the reported action at the original coordinates after the reviewed change. Check another world’s similarly named region remains unchanged.

Check the result

The investigation should identify one unambiguous world-and-region pair whose bounds contain the affected location. A region with the right name but the wrong coordinates is not the target. Include the world in support notes and future maintenance commands.

If the result is wrong

If the region cannot be found, check world loading and spelling before creating a replacement. Do not define a new overlapping region merely because a query in the wrong world returned nothing. Restore any mistaken flag edit on the other world using its recorded previous value.

References and related help

- WorldGuard maintainer reference
- WorldGuard maintainer reference
- Protect a Minecraft spawn with WorldGuard regions
- Fix overlapping WorldGuard region flags with a small permission test

Change a WorldGuard parent without losing track of inherited membership

Changing a region’s parent can alter inherited members and policy even when its own member list stays the same. Review those effects before reorganizing a group of plots.

Follow the procedure

1. Inspect the child, current parent and proposed parent in the correct world. Record explicit owners, members, flags and priority for each relevant region.
2. Read the installed priority-and-inheritance documentation. Distinguish inherited values from entries stored directly on the child; a short member list may not describe effective access.
3. Write a small access matrix for an existing child member, a parent-only member and an ordinary nonmember. State which accounts should retain access after the change.
4. Rehearse the parent change on staging or a disposable region with comparable policy. Use the documented set-parent command and check that no inheritance cycle is introduced.
5. Test the selected build and interaction actions using the three ordinary accounts. Include a flag inherited from the parent so the test covers policy as well as membership.
6. Apply the reviewed relationship to production and reinspect the child. Record the previous parent and the observed access results for a precise rollback if required.

Check the result

The child should inherit only the intended parent policy and grant access to the expected accounts. Explicit entries should remain as recorded unless separately changed. A parent label changing successfully does not prove the resulting membership and flags are appropriate.

If the result is wrong

If unexpected users gain or lose access, restore the saved previous parent and retest. Do not compensate with broad owner grants or random priority increases. Review the inheritance chain and explicit child overrides before attempting the organizational change again.

References and related help

- WorldGuard maintainer reference
- WorldGuard maintainer reference
- WorldGuard maintainer reference
- Protect a Minecraft spawn with WorldGuard regions
- Fix overlapping WorldGuard region flags with a small permission test

Limit a WorldGuard mob-spawn restriction to one region

A spawn restriction can cover an event area without changing survival spawning across the world. Select the intended entity rule and verify the source of the mobs you are investigating.

Follow the procedure

1. Identify the target region, mob type and spawn source. Natural spawning, spawn eggs, spawners and plugin-created entities may not follow identical event paths.
2. Read the installed WorldGuard spawning flags and entity-set syntax. Choose a general spawning rule or a specific deny-spawn list according to the actual policy.
3. Record the current flag and any existing entity list before editing. Preserve intentional entries rather than replacing the list with only the newest reported mob.
4. Apply the reviewed narrow rule to the named region. Use the documented entity identifier for the installed server version instead of guessing from a display name.
5. Test the relevant spawn source inside and outside the region using a controlled disposable setup. Existing mobs should be counted separately from new spawn attempts.
6. Observe normal gameplay for the area and document unsupported plugin-created cases. Keep the outside world’s intended spawning behaviour unchanged.

Check the result

The selected new spawn attempts should follow the intended regional policy, while outside spawning remains unchanged. Existing entities may remain present after a spawn rule changes, so their continued presence alone does not demonstrate failure. Record the exact source and entity tested.

If the result is wrong

If mobs still appear, identify their spawn source and check integration with the responsible plugin. Restore the previous flag list if legitimate entities are blocked unexpectedly. Do not remove every existing entity or disable all world spawning as a diagnostic shortcut.

References and related help

- WorldGuard maintainer reference
- WorldGuard maintainer reference
- Protect a Minecraft spawn with WorldGuard regions
- Fix overlapping WorldGuard region flags with a small permission test

Review a WorldGuard blocked-command rule without relying on it for permissions

A regional command block can support an event rule, but it is not a substitute for securing the command itself. Check aliases and execution paths before describing it as complete protection.

Follow the procedure

1. List the exact player commands you intend to restrict inside the region, including known aliases and namespaced forms. Identify the plugin that actually owns each command.
2. Read the installed WorldGuard blocked-command flag description and syntax. Save the current list so adding one command does not remove an existing restriction.
3. Use a non-operator test player who ordinarily has the command permission. Apply the reviewed regional list and try the intended command inside the area.
4. Test the same command outside the region and inspect its documented aliases inside. Record routes WorldGuard does not intercept, such as a command handled by a proxy first.
5. Check the actual permission policy independently. Sensitive administrative actions must remain denied to unauthorized users regardless of their position or spelling of the command.
6. Document the regional rule as a convenience or gameplay control with its verified coverage. Route unsupported proxy restrictions to the responsible platform’s supported controls.

Check the result

The tested player command should be restricted in the chosen region and remain available outside where intended. Sensitive commands should still be permission-protected everywhere. A blocked

short alias alone does not prove the namespaced or proxy-owned command is covered.

If the result is wrong

If an alias bypasses the rule, adjust the documented list or command owner’s supported policy after review. Restore the previous flag value if unrelated commands stop working. Do not grant more command permissions to test a restriction or claim a region flag replaces backend and administrative security.

References and related help

- WorldGuard maintainer reference
- WorldGuard maintainer reference
- Protect a Minecraft spawn with WorldGuard regions
- Fix overlapping WorldGuard region flags with a small permission test