GSit tasks

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

- Let players opt out of GSit click interactions
- Separate GSit command sitting from right-click seat permissions
- Diagnose a GSit region denial without opening claim protection
- Check GSit pose compatibility after a Minecraft update

Let players opt out of GSit click interactions

Clicking a stair and clicking another player are different GSit interactions. Give players clear control over the behaviour your server permits, and verify which toggle affects the specific complaint before changing global sitting settings.

Prepare the test

Choose a safe flat test area with two ordinary players. Separate sitting, clicking a seat and clicking another player because they can follow different permissions and protection checks.

Make the change

1. Choose two consenting ordinary players and a safe flat test area. Record what happens when one clicks a supported seat and when one clicks the other player, avoiding combat, edges or an active public event.
2. Check the installed help for `/sit toggle` and `/sit playertoggle`. Compare the relevant `GSit.SitToggle` and `GSit.PlayerSitToggle` permissions with the affected account’s effective permissions, including inherited denies and world contexts.
3. Use one toggle and repeat only its corresponding interaction, then test the other separately. Explain the observed scope to the player rather than promising that any one toggle disables every GSit command or interaction.
4. Reconnect and repeat the agreed checks to establish how the installed release preserves the preference. Leave voluntary command-based sitting available if your policy permits it, and verify the second player’s own preference independently.

Check the result

Each player should understand which click interaction is enabled and how to change it through an allowed command. The other player’s settings and ordinary gameplay should remain unaffected by the test account’s preference change.

If the test fails

If the toggle is unavailable, repair the specific intended toggle permission instead of granting `GSit.*` or operator status. If behaviour still differs, inspect global configuration and protection rules before removing other movement or seat plugins.

Use your community rules alongside the plugin controls. A technical toggle is helpful, but it does not replace consent rules for player stacking or staff handling of repeated unwanted interactions.

References and related help

Check the official GSit documentation against your installed release before using a command or setting from another version.

- Set up GSit poses and restrict player interactions
- Create Minecraft ranks and inheritance with LuckPerms

Separate GSit command sitting from right-click seat permissions

A player who can use `/sit` may still be unable to right-click a stair. Those paths have distinct permissions and can also meet different protection checks, so test each action explicitly before broadening the rank.

Prepare the test

Choose a safe flat test area with two ordinary players. Separate sitting, clicking a seat and clicking another player because they can follow different permissions and protection checks.

Make the change

1. Choose a safe unclaimed comparison area and one intended seat block. Record whether the affected player can run `/sit` and whether the same player can right-click the supported seat while holding an appropriate item state.
2. Inspect effective `GSit.Sit` and `GSit.SitClick` permissions through your permission manager. Compare the exact account, world and group context with an ordinary account that works, keeping operator bypasses out of the comparison.
3. Correct only the permission that contradicts the server’s rule. Test the command and click action again before changing allowed materials or region flags, so you know whether the permission edit fixed its intended path.
4. Repeat inside the area where the complaint occurred. If both actions now work outside but one remains blocked inside, compare protection settings for that region rather than adding a broad world bypass to the player.

Check the result

The intended command and click routes should each follow your permission policy in the relevant area. Other poses, player stacking and administration actions should remain controlled by their separate permissions.

If the test fails

Restore an ineffective permission edit and inspect the seat material, available space and protection response. Do not grant a wildcard or remove claim protection simply because an administrator can sit on a block that an ordinary player cannot.

Keep the before-and-after permission results in a small support note. A reproducible difference between command, click and region is more useful than describing the entire plugin as unavailable to one rank.

References and related help

Check the official GSit documentation against your installed release before using a command or setting from another version.

- Set up GSit poses and restrict player interactions
- Create Minecraft ranks and inheritance with LuckPerms

Diagnose a GSit region denial without opening claim protection

Protected areas can intentionally reject sitting or posing. Prove which action is denied and which protection integration owns the decision before changing region permissions that also protect blocks, vehicles or other interactions.

Prepare the test

Choose a safe flat test area with two ordinary players. Separate sitting, clicking a seat and clicking another player because they can follow different permissions and protection checks.

Make the change

1. Use an ordinary test account and record the region, action and message shown. Repeat the same sitting or pose action just outside the protected area, keeping the player, seat material and nearby space otherwise comparable.
2. Identify whether the location uses WorldGuard, GriefPrevention, PlotSquared or another integration supported by the installed GSit release. Inspect overlapping region priorities and membership before assuming the most visible region is the only rule.
3. For WorldGuard, compare the documented GSit action flags with the relevant underlying interaction requirements. Review the sit, playersit, pose or crawl action that actually failed rather than enabling every flag across the world.
4. Apply the narrow intended region policy and retest as an ordinary player. Separately confirm that protected building and unrelated interaction restrictions remain effective, using harmless checks that do not modify another player’s claim.

Check the result

The chosen GSit action should work only where your area policy permits it, with claim protection still intact. Check both a region member and a non-member if the intended behaviour depends on membership.

If the test fails

Restore the original flags if the experiment broadens access or fails to change the action. Investigate another overlapping rule or integration before granting a global bypass or setting build access to allow for everyone.

The maintainer documents that some GSit WorldGuard actions also depend on ride, interact or build permissions. That dependency is a reason to review the full policy, not an instruction to open all three flags globally.

References and related help

Check the official GSit documentation against your installed release before using a command or setting from another version.

- Set up GSit poses and restrict player interactions
- Find Minecraft logs and send useful troubleshooting details

Check GSit pose compatibility after a Minecraft update

Working sitting does not prove that every pose or crawl feature supports the same client and server combination. Capture the failing pose precisely after an update instead of reinstalling the server or disabling all GSit functionality.

Prepare the test

Choose a safe flat test area with two ordinary players. Separate sitting, clicking a seat and clicking another player because they can follow different permissions and protection checks.

Make the change

1. Record the Minecraft server build, GSit version and affected client version. Compare them with the maintainer’s current release information; pose support can have different requirements from ordinary seat interaction and should not be inferred from an old tutorial.
2. Test a simple `/sit` in a safe area, then the specific failing pose using the installed command help. Use an ordinary player account and note whether the command is denied, accepted without a visual change or accompanied by a console error.
3. Compare a client matching the server’s native protocol with a translated client if your network uses version-translation plugins. Keep permission and region contexts identical so a protocol difference is not confused with an access rule.
4. If a compatible plugin update is required, preserve the current jar and configuration and test it on staging first. Check its dependencies and release notes before applying the narrow update during a planned restart.

Check the result

Supported poses should behave consistently for the client combinations you intend to support. Record any documented limitation and keep working seat features available instead of promising universal pose support from one successful administrator test.

If the test fails

Restore the previous compatible plugin build if a trial update introduces new faults, following its documented configuration compatibility rules. Send the exact pose, versions and redacted error to support rather than deleting player data or rebuilding the world.

A console capture can establish when an update-related error occurs. Include the short searchable error text and reproduction steps alongside a screenshot so the report remains useful when diagnosing a specific pose.

> CloudExa game panel console and logs view for collecting startup and diagnostic evidence
> Collect the relevant timestamped log text in the panel, then remove private data before sharing it with support.

References and related help

Check the official GSit documentation against your installed release before using a command or setting from another version.

- Set up GSit poses and restrict player interactions
- Find Minecraft logs and send useful troubleshooting details