WorldGuard ExtraFlags tasks

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

- Test regional flight with WorldGuard ExtraFlags before opening an area
- Check regional keep-inventory without changing the whole world rule
- Change regional walking speed without leaving players with a stale speed
- Fix an unknown ExtraFlags flag after a plugin or server update

Test regional flight with WorldGuard ExtraFlags before opening an area

A regional flight rule needs a safe exit as well as a working entry. Test the ExtraFlags fly flag near the ground before letting players cross the boundary at height.

Follow the procedure

1. Confirm the installed WorldGuard ExtraFlags plugin is aromaa's project and that its version supports your WorldGuard and server build. Check startup output for flag-registration errors.
2. Create or choose a small disposable test region over safe ground. Inspect its existing flags and any other plugin that enables flight for the test account.
3. Read the registered `fly` flag type in the maintainer source and use the installed WorldGuard flag command to apply the intended state to this test region.
4. Use a normal survival test account without global flight access. Enter the region and check whether the documented regional flight behaviour becomes available.
5. Leave at a low safe height, then test walking out, teleporting out and rejoining as relevant. Record whether another plugin restores or removes flight after the boundary transition.
6. Remove the trial flag or restore its previous value once the exit policy is verified. Only apply the reviewed rule to a public high-altitude area after its landing route is safe.

Check the result

The test account should gain and lose the intended regional flight behaviour predictably, with a safe landing during every tested exit route. A successful entry alone is insufficient. Record any dependence on another flight plugin or server game mode.

If the result is wrong

If flight persists or disappears unexpectedly, restore the previous flag and test the competing flight provider independently. Do not perform repeated high-altitude falls or grant global flight to mask the transition. Use a full supported restart if dependency changes are required; ExtraFlags does not support arbitrary plugin reloading.

References and related help

- WorldGuard ExtraFlags maintainer reference
- WorldGuard ExtraFlags maintainer reference
- WorldGuard ExtraFlags maintainer reference
- Add and test WorldGuard ExtraFlags in one region
- Protect a Minecraft spawn with WorldGuard regions

Check regional keep-inventory without changing the whole world rule

A region-specific death policy can conflict with world gamerules or other death plugins. Verify the ExtraFlags keep-inventory result using disposable items before promising players their possessions are protected.

Follow the procedure

1. Confirm the installed ExtraFlags build registers `keep-inventory` and read its Boolean flag type. Record the existing world keep-inventory gamerule and any grave, death-chest or inventory recovery plugin.
2. Choose a small isolated test region and save its current flags. Prepare a disposable non-operator account carrying only a counted set of ordinary items.
3. Apply the intended regional Boolean value through the supported WorldGuard flag command. Do not use an allow/deny value merely because some other flags are state flags.
4. Perform a controlled test death inside the region and record retained inventory, ground drops and any grave created. Check experience separately; `keep-exp` is a distinct flag.
5. Repeat outside the region with another disposable set if your world policy permits. Ensure the test demonstrates a regional difference rather than the world gamerule preserving everything.
6. Restore the test account’s ordinary state and document the exact death-plugin combination. Keep the region’s inventory and experience policies explicit for players.

Check the result

The inventory outcome should match the reviewed regional policy without producing duplicate dropped or grave items. Experience may follow a separate rule. Count retained and external items together so an apparently successful respawn does not conceal duplication.

If the result is wrong

If the result conflicts with a grave or death plugin, restore the previous regional value and consult that integration’s supported ordering. Do not test with valuable player inventories or change the whole world gamerule just to force a regional demonstration to pass.

References and related help

- WorldGuard ExtraFlags maintainer reference
- WorldGuard ExtraFlags maintainer reference
- WorldGuard ExtraFlags maintainer reference
- Add and test WorldGuard ExtraFlags in one region
- Protect a Minecraft spawn with WorldGuard regions

Change regional walking speed without leaving players with a stale speed

A regional speed effect should end predictably when a player leaves. Test entry, exit and reconnect behaviour so an event area does not leave a persistent movement change behind.

Follow the procedure

1. Confirm the installed ExtraFlags build provides the `walk-speed` flag and review its numeric type and supported range in the matching documentation or implementation.
2. Record the target region’s existing speed setting and identify other plugins that adjust movement speed. Use a short flat test route with a visible start and finish.
3. Apply a modest supported trial value to a disposable region through the WorldGuard flag command. Avoid extreme values and do not guess a range from another plugin’s speed command.
4. Have a non-operator account walk the same route before entry, inside the region and after exit. Keep sprinting, potion effects and game mode consistent.
5. Test the intended teleport and reconnect paths. Inspect whether another movement provider overwrites the state and whether the normal walking speed returns afterward.
6. Remove the trial value or restore the saved one and repeat the outside walk. Record the approved setting only after confirming a clean return to ordinary movement.

Check the result

The account should show the intended movement change inside the region and return to its normal state through the tested exit paths. A timing comparison is more useful than a subjective claim that movement feels faster. Keep external potion and speed effects out of the trial.

If the result is wrong

If the speed remains altered, restore the previous flag and use the supported movement provider to return the test account to its recorded normal state. Check plugin conflicts before increasing the value. Do not reset all player data or apply a global speed command to solve one regional transition.

References and related help

- WorldGuard ExtraFlags maintainer reference
- WorldGuard ExtraFlags maintainer reference
- WorldGuard ExtraFlags maintainer reference
- Add and test WorldGuard ExtraFlags in one region
- Protect a Minecraft spawn with WorldGuard regions

Fix an unknown ExtraFlags flag after a plugin or server update

An unknown flag usually means the extension did not register it, not that the region needs to be recreated. Inspect startup and dependency compatibility before changing protection data.

Follow the procedure

1. Record the exact unknown flag name and the versions of WorldGuard, WorldEdit, ExtraFlags and the server. Confirm you are using the intended aromaa project rather than a similarly named extension.
2. Open the panel console or startup log and find the first ExtraFlags error. A flag-registration error after a plugin reload has a different cause from a missing dependency or unsupported server API.
3. Compare the flag with the matching maintainer source. Some desired flags belong to a different project, and a name from an online example does not establish that your installed plugin provides it.
4. Preserve the region configuration and current plugin files before a correction. Do not delete or recreate a region merely because its extension flag cannot currently be recognized.
5. Arrange a controlled full restart with a documented compatible dependency set if needed. ExtraFlags explicitly rejects unsupported reload-time flag registration in its implementation.
6. Check startup for successful enablement, then inspect the existing region and test the specific flag on a disposable area before changing the public region.

   CloudExa game panel: view console logs
   CloudExa game panel reference. This image shows the panel, not the WorldGuard ExtraFlags interface.

Check the result

The intended extension should enable cleanly and register the documented flag, while existing region ownership and other protections remain intact. A recognized flag name is only the first check; test its practical behaviour with an ordinary account before considering the compatibility issue resolved.

If the result is wrong

If startup still fails, restore the previous matched plugin set and configuration and provide the first exception to the maintainer. Avoid repeated live plugin reloads, random forks or editing stored flag data by hand. Keep the region’s ordinary WorldGuard protections active while the extension is investigated.

References and related help

- WorldGuard ExtraFlags maintainer reference
- WorldGuard ExtraFlags maintainer reference
- Add and test WorldGuard ExtraFlags in one region
- Protect a Minecraft spawn with WorldGuard regions