FastAsyncWorldEdit tasks

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

- Assign a finite FAWE edit limit to one builder group
- Test FAWE region restrictions with a claim owner and a visitor
- Investigate a FAWE edit that appears queued or unfinished
- Check FAWE disk history before relying on a later undo
- Limit FAWE schematic access for a shared building team
- Bound FAWE brush radius without removing a builder’s other tools
- Verify FAWE disallowed blocks with a non-admin account
- Inspect lighting and side effects after a FAWE terrain edit

Assign a finite FAWE edit limit to one builder group

A finite limit helps a builder correct a mistake before a huge edit affects the world. Define a named FAWE limit group for a real workload, then prove the limit applies to an ordinary builder account rather than an administrator with bypass permissions.

Work through the change

1. Record the largest legitimate edit the builder needs and create a small test region. Read the limit group settings generated by the installed FAWE release.
2. Add a named group with a bounded change limit and grant its documented fawe.limit group permission to the intended staff role. Keep administrator and bypass nodes out of the trial account.
3. Apply the configuration using the release-supported method in a planned window. Confirm the startup log accepts the limit group before testing an edit near the chosen limit.
4. Try one operation below the limit and one controlled operation above it on disposable blocks. Record which group FAWE selects when the player inherits more than one relevant permission.

Check the result

The permitted operation should finish and the oversized trial should stop according to the installed limit behaviour. Inspect the actual world after a rejected operation; a message alone does not establish that no partial changes occurred. Recheck with a fresh non-operator session.

Check the permission group actually assigned to the builder, including inherited grants. A limit configured under an unused group name will not govern their edit. Record the smallest successful edit and the deliberately rejected larger one so both sides of the allowance are demonstrated.

Recover or stop safely

Remove the trial group grant or restore its previous values if the result differs from the policy. Recover any partial test edit from history or the checkpoint. Do not grant an unlimited group just to suppress a confusing limit message.

Related guidance and maintainer reference

For the broader setup, read Replace WorldEdit with FastAsyncWorldEdit safely. Check the installed plugin version before applying syntax from the official FastAsyncWorldEdit documentation.

Test FAWE region restrictions with a claim owner and a visitor

A protection integration must distinguish someone who owns the work area from someone who merely stands inside it. Verify both identities on a test claim before giving builders access to fast editing commands across a populated server.

Work through the change

1. Record the installed protection plugin, FAWE version and region restriction setting. Select a disposable claim with a clearly marked inside block and outside block.
2. Grant the documented integration permission to a limited builder account. Make that account an owner or member only where your protection integration requires it; retain a visitor account with no build role.
3. As the builder, attempt a tiny edit inside the allowed area and a second edit crossing the boundary. Keep screenshots of the coordinates and resulting messages.
4. Repeat from the visitor account and inspect all control blocks. Test overlapping regions or plots separately, since the integration can combine ownership with an administrator bypass.

Check the result

The intended builder should edit the authorised area without modifying the outside control block. The visitor should not gain editing rights through proximity alone. A test as an operator can conceal a missing region rule, so finish with the exact player role you intend to deploy.

Use the same selection for the owner and visitor tests so different geometry cannot explain the result. Include one control block outside the claim. A plugin message alone is insufficient evidence if blocks across the boundary still changed during the operation.

Recover or stop safely

Revoke the trial editing permission if a boundary test changes protected blocks. Restore the small test region and review the integration setting rather than disabling protection globally. Keep the before-and-after role grants with the support evidence.

Related guidance and maintainer reference

For the broader setup, read Replace WorldEdit with FastAsyncWorldEdit safely. Check the installed plugin version before applying syntax from the official FastAsyncWorldEdit documentation.

Investigate a FAWE edit that appears queued or unfinished

An unfinished edit can reflect queue pressure, a stopped task, a limit or a protection failure. Repeating the same command may add more work and obscure the original operation. Preserve its timestamp and inspect progress before changing queue settings.

Work through the change

1. Write down the initiating player, command, selection volume and start time. Ask other builders to pause new edits while you examine the single affected operation.
2. Check server responsiveness and the FAWE log around that timestamp. Separate a queue warning from an exception or an explicit edit limit, and note whether blocks continue appearing.
3. Use the task or queue information supported by your installed release. Avoid copying undocumented cancellation commands from another FAWE generation into a live console.
4. If you must stop the task, use the documented control and then inspect the whole selected area. Capture a checkpoint of the current partial state before attempting a compensating edit.

Check the result

Your notes should establish whether the original operation is still progressing and which error, if any, stopped it. Compare block counts at fixed intervals instead of clicking the command again. Normal server ticks do not prove the asynchronous edit has completed.

Keep a second administrator from submitting unrelated edits during the observation window. Several concurrent jobs can make an apparently stationary queue harder to interpret. Record which account submitted the operation and its bounds so the next log review follows the correct piece of work.

Recover or stop safely

For a failed partial edit, choose a verified history entry or restore the world checkpoint in a maintenance window. Restore the previous queue configuration if a trial change worsens responsiveness. Do not assume a restart reverses the blocks FAWE already placed.

Related guidance and maintainer reference

For the broader setup, read Replace WorldEdit with FastAsyncWorldEdit safely. Check the installed plugin version before applying syntax from the official FastAsyncWorldEdit documentation.

Check FAWE disk history before relying on a later undo

Undo availability depends on the history configuration and retained files. A builder may remember an edit that the server can no longer recover through its current history. Test the recovery path using a small named operation before promising a rollback for a larger job.

Work through the change

1. Read the history settings from your installed FAWE configuration and note the storage location, retention controls and available disk space. Keep a separate completed world backup.
2. Make a distinctive tiny edit in a test area with a dedicated builder account. Record the time, world and player identity so you can distinguish it from earlier operations.
3. Use the documented history interface to locate that operation. Inspect the files or reported entry without manually changing their contents or moving the active history directory.
4. Perform a controlled undo and verify the actual blocks. If you need recovery after a restart, schedule that separate test on a staging copy rather than restarting a busy server for an assumption.

Check the result

The recorded test operation should be discoverable and reversible within the retention policy you observed. Disk history consumes storage and can contain useful audit information. An entry count alone does not prove every original block or entity state remains recoverable.

Compare the timestamp of the history record with the edit you intend to undo. An available history folder does not prove every operation was retained, and storage policies can remove older records. Keep a normal world backup for recovery beyond the history retention window.

Recover or stop safely

Return any history setting to its previous value after an unsuccessful experiment and preserve existing history files for diagnosis. Recover the test area from its backup if history fails. Keep world backups independent of FAWE retention so one storage problem does not remove both options.

> CloudExa game panel file manager
> The CloudExa file manager can help locate plugin configuration and saved diagnostic files. This is the hosting panel, not the plugin’s own interface.

Related guidance and maintainer reference

For the broader setup, read Replace WorldEdit with FastAsyncWorldEdit safely. Check the installed plugin version before applying syntax from the official FastAsyncWorldEdit documentation.

Limit FAWE schematic access for a shared building team

Schematic permissions can cover reading, writing and sharing different files. A builder who needs to paste an approved structure does not necessarily need to overwrite the team library or upload it to an external service. Test those capabilities separately.

Work through the change

1. Create a small approved schematic library and keep an offline copy. Record which builders should load existing files and which maintainers may save new library versions.
2. Consult FAWE and WorldEdit permission documentation for the installed release. Grant only the relevant load or save capabilities instead of an administrator or blanket WorldEdit permission.
3. With a non-operator builder, load one approved file and paste into a test plot. Then try the restricted save or share action against an expendable test name.
4. Check the library after the negative test. Confirm the original file size and name remain unchanged, and that the test account cannot access a sharing feature you intended to restrict.

Check the result

The builder should complete the approved paste while the restricted write or share operation fails. Inspect inherited permission groups if the account retains extra access. Folder names and permissions vary between releases, so an old library path should not become your only security assumption.

Agree which staff role can load shared schematics and which can save new ones. Those are separate responsibilities even when the interface feels similar. Keep an untouched known-good schematic for the permission trial, and avoid using a live customer build as an access-control test.

Recover or stop safely

Revoke the trial grants if the library becomes writable beyond policy. Restore any changed test file from the offline library copy and keep its version label. Do not replace the whole library with files from a player-provided archive without reviewing each entry.

Related guidance and maintainer reference

For the broader setup, read Replace WorldEdit with FastAsyncWorldEdit safely. Check the installed plugin version before applying syntax from the official FastAsyncWorldEdit documentation.

Bound FAWE brush radius without removing a builder’s other tools

A brush can affect more land with one click than a builder expects. Keep the brush radius tied to the role and test a boundary case in an empty area. Changing the brush limit is more precise than withdrawing every editing command from the team.

Work through the change

1. Record the current role grants and the largest brush the builder needs for approved work. Check the radius controls in the installed FAWE limit group configuration.
2. Create or adjust one finite limit group and assign it to a test builder without bypass grants. Keep the existing group definition available for rollback.
3. Bind a harmless small brush on disposable terrain, then attempt to choose a radius beyond the planned cap. Use the documented syntax for that brush type and installed release.
4. Inspect the actual affected footprint and repeat with another bound item. Verify that the builder can still make ordinary selections and perform the smaller authorised brush operation.

Check the result

The role should retain useful editing access while the oversized radius fails or clamps as the documented release specifies. A radius limit does not cap every other operation, so also retain a finite block-change limit and protection integration. Test as a normal builder after reconnecting.

Test a deliberately oversized radius on the disposable area as well as an allowed one. A successful small brush proves only that basic editing works. Record the rejection message so staff can distinguish the agreed size limit from a broken tool or missing plot permission.

Recover or stop safely

Restore the previous group settings if normal work stops functioning and remove the test brush binding. Undo the small trial operation while its history remains available. Investigate inherited unlimited or bypass roles before increasing any cap to satisfy an unexplained result.

Related guidance and maintainer reference

For the broader setup, read Replace WorldEdit with FastAsyncWorldEdit safely. Check the installed plugin version before applying syntax from the official FastAsyncWorldEdit documentation.

Verify FAWE disallowed blocks with a non-admin account

A disallowed block list can prevent accidental use of materials that create server or gameplay problems. An administrator bypass can make the list appear broken. Check one harmless denied material on a test copy before relying on the policy for other blocks.

Work through the change

1. Copy the current disallowed-block settings and note the configuration file used by your FAWE release. Pick one expendable material for the policy test rather than a dangerous block.
2. Check the documented relationship between the block list and the worldedit.anyblock permission. Remove that bypass from the trial builder while preserving other required edit permissions.
3. Attempt a small edit with the denied material and another edit with an allowed material. Keep both commands and the complete response in the change notes.
4. Inspect the world and reconnect the builder before repeating. If an unexpected grant remains, inspect inherited roles and temporary permissions instead of changing the list repeatedly.

Check the result

The denied test material should not appear, while the ordinary material should work within the same selection and region. That comparison separates a block policy from a general editing or region denial. Review command patterns and brushes that players actually use, not just one set command.

A player can receive a bypass through a parent group even when the direct role looks restricted. Save effective permissions from the actual test account and repeat after reconnecting. Keep the prohibited-material trial tiny so an unexpected grant cannot scatter dangerous blocks across the

world.

Recover or stop safely

Restore the test list entry and previous role grants when the check ends. Undo the expendable allowed edit. Keep the production list narrow and documented; removing all restrictions to diagnose one denied command can allow unrelated risky materials across the entire world.

Related guidance and maintainer reference

For the broader setup, read Replace WorldEdit with FastAsyncWorldEdit safely. Check the installed plugin version before applying syntax from the official FastAsyncWorldEdit documentation.

Inspect lighting and side effects after a FAWE terrain edit

A finished block placement can leave an area looking wrong because lighting or physics-related side effects need attention. Inspect the saved world from another client before assuming the edit failed and repeating a large terrain operation.

Work through the change

1. Record the FAWE build, affected coordinates and command. Keep builders out of the test area until you have compared its block layout with the intended result.
2. Reconnect or revisit the chunks and ask another player to view the same location. Compare client visual artefacts with server-side block and light conditions at fixed points.
3. Read the side-effect and queue guidance for the installed FAWE release. Trial only a documented update or relighting method on a compact section before applying it elsewhere.
4. Observe server responsiveness and inspect fluids, falling blocks and nearby redstone after the trial. These behaviours can matter even when the terrain now appears visually correct.

Check the result

The test section should retain the intended blocks and show consistent lighting to both clients. If a resource pack or client shader causes the display difference, server-wide relighting may add load without changing the underlying issue. Save before-and-after views from matching positions.

Inspect a test tunnel as well as an exposed surface. Lighting faults can be hidden at midday and obvious underground. Keep clients at comparable brightness settings, and record whether the anomaly survives a reconnect before deciding that stored light data needs repair.

Recover or stop safely

Undo only the demonstrated faulty test operation or restore the area checkpoint if the repair changes terrain. Return any trial side-effect setting to its earlier value. Avoid toggling several performance controls at once, because that removes your ability to identify which behaviour changed.

Related guidance and maintainer reference

For the broader setup, read Replace WorldEdit with FastAsyncWorldEdit safely. Check the installed plugin version before applying syntax from the official FastAsyncWorldEdit documentation.