Dynmap tasks

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

- Find a Dynmap marker hidden by its layer or zoom settings
- Hide a consenting player’s Dynmap location and verify the public view
- Separate a stale Dynmap tile from a stale browser response
- Pause Dynmap rendering during a busy event and resume deliberately
- Compare a lower-detail Dynmap view before changing production map quality
- Diagnose a Dynmap view using the wrong world template

Find a Dynmap marker hidden by its layer or zoom settings

A missing marker may still exist in Dynmap’s marker data while its set is unchecked or hidden at the current zoom. Check the layer before recreating markers. Repeatedly adding the same location can leave duplicate icons that appear later when visibility changes.

Work through the change

1. Identify the expected marker and its marker-set ID. Read the existing marker and set listings through the installed dmarker help, recording labels separately from their internal identifiers.
2. Inspect the set’s hide and minzoom settings in the documented marker controls. Open the web layer selector and check whether that particular set is enabled in the current browser.
3. Zoom to the marker’s recorded coordinates and compare several zoom levels. Check the world name too; a correctly configured marker will not appear on an unrelated world’s map.
4. Change only the responsible set visibility setting through the documented updateset control if the default is wrong. Reload the public page in a fresh session and compare it with the browser that originally reported the problem.

Check the result

The existing marker should appear in its intended layer at the chosen zoom without adding a duplicate. A user can still hide a selectable layer locally, so document whether the change affects the default or the current browser selection. Confirm nearby unrelated marker layers retain their earlier behaviour.

Keep the set identifier in the maintenance notes. Labels can be renamed for visitors, while later command-based checks still need to target the correct internal collection.

Recover or stop safely

Restore the recorded set settings if the change exposes too much information. Do not delete the marker set to reset its display: deleting a set also deletes its contained markers. Preserve identifiers and coordinates before any later structural change to the marker collection.

Related guidance and maintainer reference

For the broader setup, read Install Dynmap and test a bounded world render. Check the installed plugin version before applying syntax from the official Dynmap documentation.

Hide a consenting player’s Dynmap location and verify the public view

Player visibility on a public map should be tested from the public page. Dynmap supplies hide and show controls, but another component can publish similar information independently. Agree what needs to be hidden before changing settings for a live player.

Work through the change

1. Ask a consenting test player to join and record the current marker and player-list behaviour. Check whether the installed version or another plugin applies persistent visibility rules on login.
2. Read the installed Dynmap command help for the hide control and its required permission. Apply the documented action to the test player without granting them map administration or unrelated command access.
3. Open the public map in a fresh browser and move the player between known locations. Inspect both the map icon and player list, then repeat after reconnecting to identify whether the state persists as intended.
4. Check third-party claim, chat and marker integrations separately. A static marker bearing the player’s name or a chat message can remain visible even when their live location is hidden.

Check the result

The tested public view should match the agreed location policy, including the reconnect behaviour you recorded. Do not promise that hiding a live icon erases information already observed by visitors. Keep test results tied to the installed plugin version and the components actually enabled.

Use a second consenting test account as a control. Its visible marker helps show that you changed the intended player policy rather than accidentally hiding every live location.

Recover or stop safely

Use the documented show control to restore the test player’s earlier state when appropriate. If another component republishes locations, change that integration through its own configuration rather than removing the terrain map. Preserve the privacy setting until the client understands which information is still public.

Related guidance and maintainer reference

For the broader setup, read Install Dynmap and test a bounded world render. Check the installed plugin version before applying syntax from the official Dynmap documentation.

Separate a stale Dynmap tile from a stale browser response

A build can remain outdated on one browser even when Dynmap has already rendered the new blocks. Compare the rendered state and the web response before scheduling a full world render. One small landmark is enough to isolate the failing layer.

Work through the change

1. Record the world, map name and coordinates of a distinctive changed block. Confirm the live world save includes it, then check Dynmap’s task state and console for render errors near the same time.
2. Use the installed documentation to request a small render around the affected location. Avoid a fullrender while investigating a single tile because it adds work unrelated to the observed problem.
3. Wait for the relevant work to finish and open the location in a fresh browser session. Compare a second device and inspect the response for the tile request if only one view remains outdated.
4. If a proxy or CDN serves the map, compare its cache policy and response age with the origin’s current data through the established administration path. Keep internal addresses and access tokens out of public screenshots.

Check the result

The changed landmark should appear consistently after the render and a fresh response. A successful plugin command confirms a request was accepted, not that every web cache has updated. Record whether the stale result affects one tile, one map or all served assets.

Compare the same zoom and coordinates on both clients. Different detail levels may request different rendered files and make a cache problem appear inconsistent.

Recover or stop safely

Leave world files unchanged while correcting the web delivery or specific render problem. Restore any temporary cache setting after the test and retain console evidence if rendering failed. Do not wipe the entire tile directory merely to clear one browser’s outdated response.

Related guidance and maintainer reference

For the broader setup, read Install Dynmap and test a bounded world render. Check the installed plugin version before applying syntax from the official Dynmap documentation.

Pause Dynmap rendering during a busy event and resume deliberately

Map rendering can compete with a large event for server resources. Dynmap provides rendering controls so you can defer background work without removing the map plugin. Record the paused state explicitly so yesterday’s emergency pause does not become tomorrow’s unexplained stale map.

Work through the change

1. Read the current rendering status using the installed Dynmap command documentation. Record active work, affected worlds and the reason for pausing before the event begins.
2. Use the documented pause control for the intended scope. Confirm the console acknowledges the change and inspect player response through the server’s existing monitoring instead of assuming every lag problem came from rendering.
3. Check the public map during the pause and tell staff which updates are deferred. Avoid adding full-render requests while work is intentionally paused because queued jobs can create a surprise burst later.
4. After the event, resume through the installed control and monitor a small interval of work. Compare map freshness, task progress and player response before leaving rendering unattended.

Check the result

The service should continue normal play during the pause, while deferred rendering resumes at the planned time. The public page may remain reachable with older imagery; that is different from proving live updates are working. Include the resumption check in the event’s maintenance notes.

Give the pause an owner and a planned review time. An intentionally stale map should not remain unexplained once the event and its resource pressure have ended.

Recover or stop safely

Pause again if the resumed work overwhelms available capacity and review the queue before changing performance settings. Keep existing tiles and world data intact. If staff cannot identify the current state, collect command output and logs rather than repeatedly restarting the server to reset rendering.

Related guidance and maintainer reference

For the broader setup, read Install Dynmap and test a bounded world render. Check the installed plugin version before applying syntax from the official Dynmap documentation.

Compare a lower-detail Dynmap view before changing production map quality

A detailed map can consume substantial storage and rendering time. Dynmap’s map definitions separate views and rendering choices, so evaluate a modest alternative before replacing an established public map. Keep existing tiles available while you measure the tradeoff.

Work through the change

1. Record the current world and map definitions, storage use and representative close-up screenshots. Back up the relevant configuration and check the installed documentation for available templates and render settings.
2. Prepare a distinct test view following the current world-and-template structure. Give it an unmistakable name and change only the quality-related setting you intend to evaluate, preserving source world identity.
3. Render a bounded representative area containing buildings, foliage and terrain. Measure generated storage and elapsed work, and compare the view on an ordinary client device at several zoom levels.
4. Decide whether the saving justifies the lost detail. Document whether existing tiles require regeneration for the chosen setting and schedule any wider work for a quiet period with sufficient free space.

Check the result

The comparison should show both the appearance difference and measured resource use for the same area. A smaller test render does not guarantee an identical saving across every terrain type. Keep the original public view until the new option has been accepted and its rollout is planned.

Include a representative dense build in the trial rather than only empty terrain. The visual compromise and storage cost are easiest to judge where players care about detail.

Recover or stop safely

Restore the original map configuration if the experiment is unsuitable. Retain old tile storage until its ownership and active references are understood. Do not delete a shared storage directory to recover test space; another map or world may still serve files from that location.

Related guidance and maintainer reference

For the broader setup, read Install Dynmap and test a bounded world render. Check the installed plugin version before applying syntax from the official Dynmap documentation.

Diagnose a Dynmap view using the wrong world template

A map that resembles the wrong dimension may have a configuration problem rather than corrupted terrain. Dynmap applies world and template settings together. Compare the effective world entry with the installed template structure before launching a large replacement render.

Work through the change

1. Record the affected public map name and its actual world identifier. Back up the world-and-template configuration and note recent world imports, renames or copied map entries.
2. Read the current maintainer documentation for template selection and world-specific overrides. Check that the intended world entry references the right template and that copied map definitions have distinct identifiers where required.
3. Compare a known landmark and dimension-specific feature in game with the web view. Correct one mistaken mapping or override at a time, validating the configuration through the plugin’s supported application method.
4. Render a bounded test area after the corrected definition loads. Check the map menu, orientation and visible terrain before scheduling any wider rerender needed by the configuration change.

Check the result

The map should identify and render the intended world under its chosen template, while other dimensions retain their previous definitions. A display label is not proof of source identity; inspect coordinates and landmarks. Keep the configuration diff with the test result so later world imports follow the same pattern.

Check all map menu entries after the correction. A copied template can introduce a duplicate or misleading name even when the newly tested terrain itself now looks correct.

Recover or stop safely

Restore the backed-up definition if the correction changes unrelated views or the plugin reports configuration errors. Preserve existing tiles while investigating map IDs and storage references. Do not regenerate the Minecraft world because a web renderer selected the wrong template or copied a label incorrectly.

Related guidance and maintainer reference

For the broader setup, read Install Dynmap and test a bounded world render. Check the installed plugin version before applying syntax from the official Dynmap documentation.