GerbDiff

GitHub Action Setup Guide

Visual Gerber Diffs on Every Pull Request

Version 1.0 | April 2026

Free & Open to the Public

gerbdiff.com | ben@gerbdiff.com

Table of Contents

1 Overview

The **GerbDiff GitHub Action** runs on every pull request that touches your Gerber files and posts a layer-by-layer visual diff summary directly in the PR conversation. Reviewers see *what* changed on each copper, silkscreen, solder mask, and drill layer before ever opening a CAD tool.

What the Action does on each PR:

- Detects which Gerber files changed between the base and head commits.
- Parses and renders every layer on both refs using the same engine as the desktop app.
- Runs an XOR diff per layer, reporting change percentage and region count.
- Uploads full-resolution overlay PNGs (red = removed, cyan = added, white = unchanged) as a downloadable artifact.
- Posts a single PR comment with a summary table — subsequent pushes update the same comment in place.

Runtime: Runs on any GitHub-hosted or self-hosted Linux runner. Published to the GitHub Marketplace and free for everyone — no license, subscription, or token is required. A GerbDiff desktop license is not a prerequisite for using the Action.

> ***Privacy:*** *Parsing, rendering, and diffing all happen on your runner. Gerber files never leave your GitHub environment.*

Example PR Comment

Here is what a reviewer sees on a pull request that changes three layers of a six-layer board:

github-actions[bot] commented

GerbDiff Report

Comparing `a1b2c3d` ← `9f8e7d6` · **3** changed, **3** identical

Layer	Status	Change	Regions
Top Copper	● **Changed**	2.5%	10
Top Silkscreen	● **Changed**	2.2%	6
Bottom Copper	● **Changed**	1.1%	4
Top Solder Mask	● **Identical**	—	—
Bottom Solder Mask	● **Identical**	—	—
Board Outline	● **Identical**	—	—

Below the table, an expandable section links to the overlay artifact so reviewers can download full-resolution red/cyan images per layer. See Section 5 for a full walkthrough.

2 Prerequisites

Before adding the workflow, confirm the following about the repository that will host the Action:

Requirement	Details
Repository	GitHub-hosted (public or private). GitHub Enterprise Server is supported on v3.9+.
Gerber files tracked in Git	The Action compares files at two git refs. LFS-tracked Gerbers are fully supported.
Runner	**ubuntu-latest** is recommended and what we test against. The Action runs on Node.js 20 with prebuilt cross-platform rendering binaries, so self-hosted or non-Linux runners generally work, but are not part of our test matrix.
Permissions	The workflow needs **contents: read** and **pull-requests: write** on the default **GITHUB_TOKEN**.
Cost	Free. No GerbDiff license, API key, or secret is required in the workflow.

Supported File Extensions

By default the Action recognizes these extensions as Gerber or drill files:

```
gbr, gtl, gbl, gts, gbs, gto, gbo, gtp, gbp, gko, gm1, g2, g3, drl, xln, exc
```

You can override this list with the **gerber-extensions** input (Section 4). File-name conventions (e.g., ***.GTL** vs ***-F.Cu.gbr**) are auto-detected by the same layer-matching logic used in the desktop app.

3 Quick Start

Step 1: Create the workflow file

In your repository, create **.github/workflows/gerbdiff.yml** with the following contents:

```
name: GerbDiff

on:
  pull_request:
    paths: ['gerbers/**']

jobs:
  diff:
    runs-on: ubuntu-latest
    permissions:
      contents: read
      pull-requests: write
    steps:
      - uses: actions/checkout@v4
        with:
          fetch-depth: 0
      - uses: woodruffrb/gerbdiff@v2
        with:
          base-path: gerbers
```

*Replace **gerbers** with the subdirectory that contains your Gerber files. If your Gerbers live at the repository root, omit the **base-path** input.*

Step 2: Commit and open a PR

Commit the workflow on your default branch, then open a pull request that modifies any file under the path you configured. The Action triggers on the **pull_request** event and posts its report within 1–3 minutes of the PR opening.

Step 3: Review the comment

Open the Conversation tab of the PR. You will see a **GerbDiff Report** comment posted by **github-actions[bot]** with a layer-by-layer status table. Expand the **Full-resolution overlay images** section for instructions on downloading the artifact. See Section 5 for a walkthrough of the report.

That's it — three steps, no secrets, no license key. The Action will now run on every subsequent PR that touches the configured path.

4 Configuration Reference

Inputs

Input	Default	Description
token	`${{ github.token }}`	GitHub token used to post and update the PR comment. Override only if you need
gerber-extensions	`gbr,gtl,gbl,gts,...`	Comma-separated list of file extensions to treat as Gerber/drill files.
base-path	`(empty)`	Subdirectory containing Gerber files, relative to the repo root. Files outside this pa
diff-resolution	`1000`	XOR diff resolution in pixels on the long axis. Higher = more accurate, slower. 500
artifact-name	`gerbdiff-report`	Name of the uploaded artifact containing overlay PNGs.

Outputs

The Action exposes three outputs that downstream steps can read via **steps.<id>.outputs.<name>**:

Output	Type	Description
has-changes	`'true' | 'false'`	Whether any Gerber layer showed a visual difference.
changed-layers	`integer`	Number of layers with visual differences.
total-layers	`integer`	Total number of matched layers compared.

Required Permissions

Add the **permissions** block in your workflow to grant the Action what it needs. With default-restricted GITHUB_TOKEN permissions (recommended for organization repos), this block is required:

```
permissions:
  contents: read         # Read repository files
  pull-requests: write   # Post and update the PR comment
```

5 Understanding the PR Comment

Every run produces a single Markdown comment on the pull request. The comment has a hidden marker, so reruns (new pushes to the PR branch) update the same comment in place rather than cluttering the conversation.

Layer Status Icons

Icon	Status	Meaning
■	Identical	Layer present on both refs, no pixel differences detected.
■	Changed	Layer present on both refs with visual differences. Change % and region count are reported.
■	Added	Layer is new on the PR branch (no counterpart on base).
■	Removed	Layer was deleted on the PR branch (present on base only).

Reading the Summary Row

The one-line summary below the header reports counts in each category:

```
Comparing `7efaa10` <- `788d87c` | **2** changed, **8** identical
```

The first SHA is the base branch; the second is the PR head. Reruns on new pushes update both SHAs and refresh the table.

Change Percentage and Region Count

- **Change %:** Fraction of the layer's bounding-box pixels that differ between refs. 0.1–2% covers most real-world routing edits.
- **Regions:** Number of distinct clusters of changed pixels after spatial merging. A single rerouted trace is typically 1 region; a multi-net refactor might be 10+.

6 Working with Overlay Artifacts

For every layer with visual differences, the Action renders a full-resolution overlay PNG and uploads all of them as a single artifact attached to the workflow run.

Color Legend

Color	Meaning
Red	Features present on the base branch but removed or moved on the PR branch.
Cyan	Features added or moved to a new location on the PR branch.
White	Features unchanged — present on both refs at the same location.

Downloading the Artifact

- Open the **Actions** tab of your repository and click the GerbDiff workflow run for your PR.
- Scroll to the bottom of the run summary page. A section labeled **Artifacts** lists **gerbdiff-report** (or whatever you configured for **artifact-name**).
- Click the artifact name to download a ZIP containing one PNG per changed layer (e.g., **Top_Copper.png**, **Inner_3.png**).

Artifact retention is controlled by your repository's Actions settings (default: 90 days). Because overlays contain a rendered representation of your board, treat them with the same confidentiality as the source Gerbers.

7 Advanced Examples

Gating Merges on Clean Diffs

Use the **has-changes** output to require explicit approval whenever a PR touches the Gerbers. Add the GerbDiff job as a required status check in branch protection rules:

```
jobs:
  diff:
    runs-on: ubuntu-latest
    permissions:
      contents: read
      pull-requests: write
    steps:
      - uses: actions/checkout@v4
        with:
          fetch-depth: 0
      - id: gerbdiff
        uses: woodruffrb/gerbdiff@v2
        with:
          base-path: hardware/board-rev-a
      - name: Require hardware review on Gerber changes
        if: steps.gerbdiff.outputs.has-changes == 'true'
        run: |
          echo "::notice::Gerber layers changed - hardware review required"
```

Multiple Board Projects in One Repo

If your repo holds more than one board, run the Action once per project directory using a matrix strategy. Each project gets its own PR comment and artifact:

```
jobs:
  diff:
    runs-on: ubuntu-latest
    strategy:
      fail-fast: false
      matrix:
        board:
          - { path: hardware/main-board,    name: main }
          - { path: hardware/daughter-card, name: daughter }
    permissions:
      contents: read
      pull-requests: write
    steps:
      - uses: actions/checkout@v4
        with:
          fetch-depth: 0
      - uses: woodruffrb/gerbdiff@v2
        with:
          base-path: ${{ matrix.board.path }}
          artifact-name: gerbdiff-${{ matrix.board.name }}
```

Custom File Extensions

If your toolchain emits non-default extensions (e.g., KiCad's ***-F_Cu.gbr** is fine, but some fabs use **.pho** or **.art**), override the recognized list:

```
- uses: woodruffrb/gerbdiff@v2
  with:
    gerber-extensions: gbr,gtl,gbl,gts,gbs,gto,gbo,drl,pho,art
```

8 Troubleshooting

Issue: "This action must be run on a pull_request event"

Solution: The workflow fired on a trigger other than pull_request (e.g., push, schedule). The Action reads the PR number from pull_request event payload and cannot run on other events. Ensure on: pull_request is set in your workflow.

Issue: No comment appears on the PR

Solution: First check the Actions tab for the workflow run. Common causes: (1) the pull-requests: write permission is missing in the workflow, or the organization has restricted GITHUB_TOKEN; (2) the paths filter excluded the commit so the job never ran; (3) the base-path input did not match the location of your Gerber files.

Issue: "No Gerber file changes detected" but files did change

Solution: The Action filters by file extension. Verify your filenames end with one of the extensions listed in Section 2, or override with gerber-extensions. Also check that your base-path matches the directory containing the files.

Issue: Action fails with "fatal: bad object" or "no merge base"

Solution: This almost always means the checkout step did not fetch enough history. The Action compares the PR base and head commits, which requires full Git history. Always set fetch-depth: 0 on actions/checkout.

Issue: Action runs slowly on large boards

Solution: Render time scales with layer count and diff-resolution. Try lowering diff-resolution to 500 for faster (but less precise) diffs, or run the job on a larger runner class for big designs.

Issue: Comment keeps duplicating instead of updating

Solution: The Action finds existing comments by a hidden marker. If you manually edited the comment and removed the `` marker, a new comment will be posted on the next run. Delete the orphaned comment to resolve.

Issue: Git LFS files appear as text pointers

Solution: Add an explicit LFS pull after checkout: run `git lfs pull` in a step before the gerbdiff action, or set lfs: true on actions/checkout.

9 Support & Contact

Department	Contact
Technical Support	ben@gerbdiff.com
Sales & Licensing	ben@gerbdiff.com
Bug Reports	https://github.com/woodruffrb/gerbdiff/issues
Website	https://gerbdiff.com

When reporting an issue, please include a link to the failing workflow run and, if possible, the **Checkout** and **GerbDiff** step logs. We aim to reply within one business day.

Thank you for using GerbDiff. We are committed to supporting your team's PCB design review workflow.