This page describes how we track issues in the `vscode` repository.

Popular queries

- Global Inbox
- Bugs to be Verified - VS Code only
- Verification Needed - VS Code only

Inbox tracking and Issue triage

New issues or pull requests submitted by the community are initially triaged by an automatic classification bot. Issues that the bot does not correctly triage are then triaged by a team member. The team rotates the inbox tracker on a weekly basis.

A mirror of the VS Code issue stream is available with details about how the bot classifies issues, including feature-area classifications and confidence ratings. Per-category confidence thresholds and feature-area ownership data is maintained in .github/classifier.json.

💡 The bot is being run through a GitHub action that runs every 30 minutes. Give the bot the opportunity to classify an issue before doing it manually.

Inbox Tracking

The inbox tracker is responsible for the global inbox containing all **open issues and pull requests** that

- are neither **feature requests** nor **test plan items** nor **plan items** and
- have **no owner assignment**.

The **inbox tracker** may perform any step described in our issue triaging documentation but its main responsibility is to route issues to the actual feature area owner. If you're not sure who to assign, try looking at the working areas document.

Feature area owners track the **feature area inbox** containing all **open issues and pull requests** that

- are personally assigned to them and are not assigned to any milestone
- are labeled with their feature area label and are not assigned to any milestone. This secondary triage may involve any of the steps described in our issue triaging documentation and results in a fully triaged or closed issue.

💡 Use the bot commands (like `/needsMoreInfo`, `/extCpp`, etc) to your advantage. They offer a wide range of canned responses from needing more info, to closing as a result of specific extensions.

💡 The github triage extension can be used to assist with triaging — it provides a "Command Palette"-style list of triaging actions like assignment, labeling, and triggers for various bot actions.

💡 The inbox notebook can also assist with triaging.

Ongoing Issue Management

The details can be found in our issue triaging documentation.

Planning

During the iteration planning process we use the following sources as input:

- Review feature requests with many reactions. Issues we plan to work on during an iteration are assigned to the current milestone.

Filing bugs as a development team member

When team members files a bug they perform steps of the inbox tracker for the issue they filed. Therefore bugs filed by the development team do not need to be triaged by the global inbox tracker.

Verification

Issues need to be verified.

Verification is a service that you request from others either implicitly with the `bug` -label or explicitly with the `verification-needed` -label. Find issue that are to be verified with these queries

- bugs to be verified, VS Code
- verification needed, VS Code
- bugs to be verified, all GitHub projects
- verification needed, all GitHub projects

Follow the these rules:

1. Query for issues that are to be verified
2. Start with issues you created (filter by `Author`) but didn't close
3. Pick an item
    - Start with setting `verified` -label (prevents duplicate verifications)
    - Verify the issue
    - If you cannot verify the issue due to missing or hard-to-understand repro steps, add a `verification-steps-needed` label and remove the `verified` label
    - If the issue still shows, add the `verification-found` -label and remove the `verified` label
    - Go back to #3

Author Verification

In some cases, such as when a bug is particularly hard or time-consuming to reproduce, it can be desirable to allow the initial, community member, author of the bug report to verify an issue. This can be achieved by adding the `author-verification-requested` label, which launches a workflow where the author is be pinged when the relevant patch is released, and then asked to verify the issues themselves.

Issues must be closed with a reference to a commit SHA in order for the bot to accurately ping users when the Insiders release with their fix is released.

Consistent labels across vscode repositories

Visual Studio Code consists of multiple repositories and we should use consistent work flows and labels across all our repositories.

To establish consistent labels across all our repositories use the Label Manager tool.

Consistent milestones across vscode repositories

To enable planning across repositories all the Visual Studio Code related repositories need to define the same milestones.

Iteration Planning

We use issues for iteration plans. Iteration plans have a label `iteration-plan` with `tasks` [ ] for the different items. The individual items are tracked in existing issues (bugs, feature requests). If there is no existing issue then a new issue with the label `plan-item` is created. All our iteration plans can be found here