

This document explains how we handle issue triage and scheduling in the Go project's issue tracker.

## Issue States

Any issue must be in one of the following states. Project contributors move issues from one state to another. The intent behind these explicit states is to describe the (minimum) next steps required to bring the issue to resolution. Issues may move between states in any order, as dictated by the needs of the particular issue.

### New

- The issue has been filed.
- May not be correctly formatted (title, etc).
- To transition from this state, someone must clean up the issue report and optionally CC people who might best investigate or fix it.

### Needs Investigation

- Is correctly formatted.
  - The title has a path prefix.
  - The body describes the issue.
- Has a milestone.
  - If the issue is not important, it is ok to set the milestone to **Backlog**. Otherwise, set it to the next upcoming release.
- Has the label **NeedsInvestigation**.
  - May also have the label **WaitingForInfo** if the investigator is waiting for more information from someone (e.g., the issue reporter).
- To transition from this state, someone must examine the issue and confirm that it is valid and not a duplicate of an existing issue.

### Needs Decision

- The issue is real, but we're not sure what action to take.
  - The issue can be addressed in Go 1.
  - Feedback is required from experts, contributors, and/or the community before a fix can be made.
  - Note that the majority of issues will never transition to this state, as most of the time the decision is an obvious "Yes, this should be fixed."
- Has a milestone.
- Has the label **NeedsDecision**.
  - May have the label **WaitingForInfo**.
  - May have the label **Blocked** if forward progress depends upon the resolution of another issue or the release of a future version of Go.

An accompanying comment should explain the blockage.

- Must not have the label **Go2**. (Those issues are handled separately.)
- To transition from this state, someone must decide how the issue is to be resolved.
  - If the decision is complicated, the issue may be given a **Proposal** label. The issue remains in this state until the proposal process is complete, and moves to **NeedsFix** if approved.

### **Needs Fix**

- The path to resolution is known, but the work has not been done.
- Has a milestone.
- Has the label **NeedsFix**.
  - May have the labels **Blocked** or **WaitingForInfo**.
- To transition from this state, someone must do the work to fix the issue.

### **Fixed**

- The issue is resolved. No further attention is required.
- The issue is closed.

Issues move from one state to another where appropriate. For example, a contributor may file an issue, assign it to themselves, and immediately apply the **NeedsFix** label. Or, an issue may go from **NeedsDecision** to **NeedsFix**, only to later move back to **NeedsDecision** as complexities arise.

An issue may be closed at any time, with a comment to indicate the reason for closure (“fixed by ...”, “duplicate of ...”, “working as intended”, etc).

At any state (except **New**) the issue may be assigned to someone. Unassigned issues are considered available for anyone to address.

## **Milestones**

Milestones describe the timeline for issue resolution.

- **Go1.x.y**
  - Must be fixed for release 1.x.y, or explicitly postponed to a later release.
- **Proposal**
  - Is a proposal and does not pertain to a specific release.
- **Soon**
  - Should be fixed soon, but is not included in or needed by a release.
- **Backlog**

Someone has plans to work on it, but it is not assigned to a release milestone yet.

- Unplanned

Might be fixed at some point, but nobody is planning to do it.

- Unreleased

Is not included in or needed by a release.

- Gccgo

For gccgo issues.

- Go2

Deferred until Go 2.

Additional milestones may be used to manage specific project work.