Policy Discussion for WG14
WG14 N3002

Title:	Issue Tracking Proposal for C
Author, affiliation:	Aaron Ballman, Intel
Date:	2022-06-09
Proposal category:	Committee Policies
Target audience:	WG14 members, external liaisons

Abstract: WG14 previously tracked issues against the C standard but stopped doing so in 2017 with the start of work on C2x. This proposes a replacement process so we can begin to track issues again.

Issue Tracking Proposal for C

Reply-to: Aaron Ballman (aaron@aaronballman.com)

Document No: N3002
Date: 2022-06-09

Summary of Changes

N3002

- Original proposal

1. Introduction and Rationale

During the May 2022 WG14 meeting, the committee discussed N2947 which laid out concerns about the lack of a procedure to track and resolve issues reported by committee members, implementers, or users. At that time, the committee agreed the issue was worth solving and asked for a concrete proposal laying out a potential new process.

2. Requirements

These are the key requirements to keep in mind as you consider the proposal:

2.1 Reporting Difficulty

Making a report as a committee member, member of a related committee like WG21 or WG23, or as an implementer should have as little friction as possible. Reporting issues as a general user of C should not be overly onerous but can have slightly more friction to reduce low-quality reports or spam.

2.2 Committee and Volunteer Burdens

We should not introduce significant burdens on the hosts of WG14 infrastructure (the committee website, git repository, etc) or on the volunteer maintaining the issue tracker. Additionally, the process should be such that the committee is not inundated with an intractable amount of (low-quality) reports.

2.3 Tracking

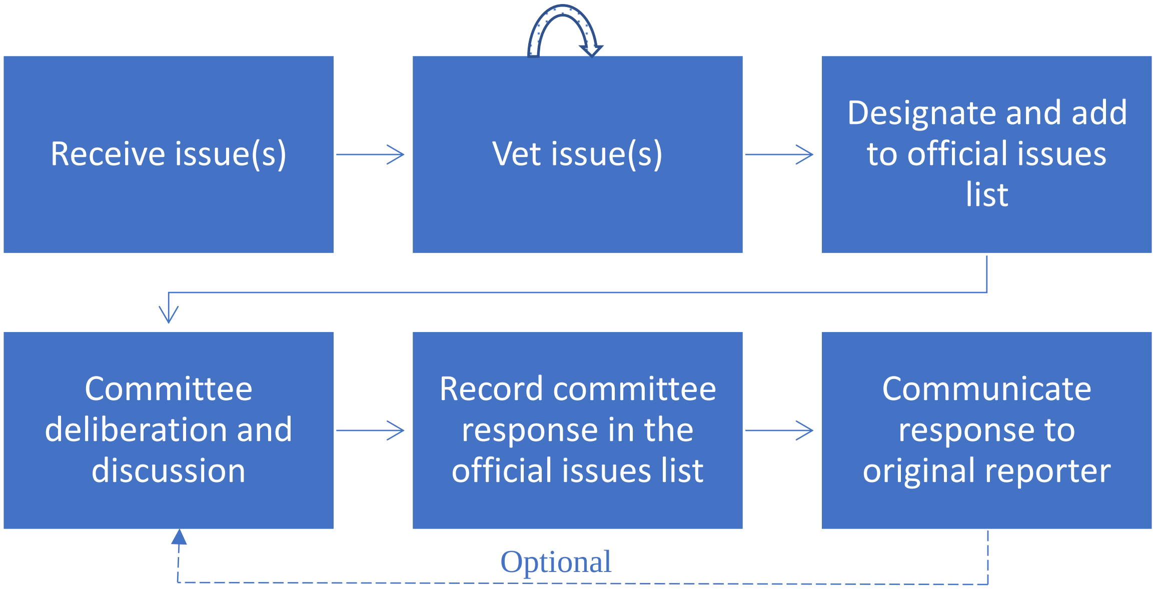
Any report which the committee deliberates on needs to be in a form that provides a historical trail for posterity. Transient services such as a Bugzilla or GitHub issues do not provide enough guarantees for that historical trail. It is reasonable to have additional tracking mechanisms beyond the historical paper trail (like a potential issues list tracked via a public issue tracker to ease collecting issues for the committee to consider).

2.4 Communication

Issue tracking is inherently a communication between the committee and some constituency. The state a report is in, the committee’s response to the report, the changes made to the standard in response to the report, next steps, and other such concerns should be clearly stated so that the reporter and the committee are communicating as effectively as possible. No issue should be resolved without some attempted form of communication with the original reporter.

3 Proposal

Based on these requirements, the following high-level workflow is proposed:


Receive issue(s)
Vet issue(s)
Designate and add to official issues list
Committee deliberation and discussion
Record committee response in the official issues list
Communicate response to original reporter
Optional

This workflow requires a committee member to be responsible for maintaining the issues and shuttling issues through the process; this individual is the *issues list maintainer*. This document assumes there is one such maintainer, but the workload can be shared amongst a group of maintainers so long as the results are consistent.

3.1 Receive Issues

The issues list maintainer is given wide latitude on how they want to manage their internal processes relating to receiving issues and communicating back to the original reporter. We recommend using an unofficial issues list, such as one provided by GitHub or Bugzilla, but it is not a requirement of this proposal to do so.

To reduce friction (especially for WG14 committee members), we propose accepting issues via direct email to the current issues list maintainer. We do not propose asking this individual to identify issues discussed on the reflectors as the discussions on the reflector are often ambiguous as to whether they're reporting an issue or simply discussing the standard without intending to spend plenary committee time on the discussion.

There is a choice as to whether the issues list maintainer uses an email address they provide or whether we have a stable email address which forwards to the issues list maintainer. If possible, we recommend using a stable email address. This allows us to share the work load of managing the issues list while having a single point of contact for people external to the committee and does not add confusion should an issues list maintainer be replaced in their role. Whatever address is used, the issues list maintainer email address should be listed on the committee's contacts page so that it is discoverable by both committee members and external constituents.

The issues list maintainer may support additional ways to receive issues, such as through a public issue tracker.

3.2 Vet Issues

The issues list maintainer is responsible for vetting issues, regardless of how they are received. The vetting performed is at the discretion of the issues list maintainer, but they should ensure that issues have an accurate title, sufficient description for the committee to consider, and are appropriate for the committee to consider (e.g., are not spam). Any time the issues list maintainer closes or rejects an issue by fiat, the reporter should be given an explanation as to why it is being rejected and what is required from the reporter before reopening it.

Any feature requests or enhancements to the C standard are rejected outright as the reporter is required to write a paper for these. It would be appropriate to give the reporter a link to the contributing page on the WG14 website for more details on that process [Contributing].

The vetting process may involve multiple iterations with the original reporter until the report is in a state where it can be discussed by the committee.

3.3 Designations

Issues have a lifecycle, and the issues list needs to track where a given report is in its lifecycle. As such, we propose using the following designations, which were the same set used by WG14 previously:

`Open`	The report is open and requires deliberation
`Closed`	The report is no longer open and has no corrigenda; represents reports that are not a defect or answers to clarification requests
`Review`	The report has a proposed response or a corrigendum, but it has not yet been accepted by the committee as final.
`<Standard Version>`	The report is no longer open and has a corrigendum; represents issues that have been resolved with changes to the standard. The standard version used is the standard which the issue is fixed in.

The issues list maintainer is responsible for ensuring that the designation in the issues list is updated as the committee processes the report. The committee is free to introduce new designations as a need arises, but the official issues list should have a legend explaining the different designations used, for historical context.

3.4 Add to Official Issues List

It is up to the issues list maintainer as to how they would like to maintain the official issues list and in what format they want to publish it. However, if the issues list maintainer writes scripts to help automate the process of maintaining the lists, it would be beneficial for those scripts to be committed into a version control repository so that future issues list maintainers can use the same process if they desire.

The official issues list should be published by the issues list maintainer as an N-numbered document before the meeting mailing deadline of each WG14 plenary meeting to ensure that it is put on the agenda for committee discussion. If there are no new reports, a new document does not need to be published; this situation should be communicated explicitly to the convener to ensure proper scheduling for the meeting if there are remaining open reports. At a minimum, each report in the list needs to contain a title, summary, designation, and committee response/proposed changes. It can contain other information if desired (such as additional comments from an unofficial issues list, reflector message IDs, etc). Each N-numbered document should contain information about all issues (both open and closed) for the version of the standard under development and be formatted so that the committee can easily navigate the document during discussion. Once the next version of the standard has been released, all closed items are removed for the first issues list to be discussed for the subsequent standard version.

3.5 Committee Deliberation and Discussion

Each meeting with open reports should schedule a portion of the week for discussions related to answering questions and resolving issues from the issues list. The scheduling for this is at the convener's discretion and likely should be based on how large the backlog of work is. It is anticipated that this should require a half of a day at most out of a week of meetings.

The convener (or someone they designate) will explicitly provide the issues list maintainer with any committee responses or status changes for each individual report discussed in the event the issues list maintainer is not present for the deliberations.

Some of this committee deliberation and discussion can potentially happen on the reflectors in advance of a meeting. The meeting will provide an official response to accepted reports, but at the issues list maintainer's discretion, reflector discussions can be used to come up with a proposed committee response that the full committee either modifies or ratifies. If such an approach is desirable to the committee, we could empanel a study group to perform these preliminary activities at regular intervals.

3.6 Record Committee Response

The issues list maintainer should ensure that all committee responses are faithfully reflected in the official issues list. Some amount of editorial correction is fine, but the issues list maintainer should not interpret or reword official committee responses or changes. The issues list maintainer may also give the report a new designation based on the committee response.

3.7 Communicate Response

Regardless of how the issues list maintainer records official committee responses, these responses need to be communicated to the original submitter. Closing the loop like this makes for a better working relationship with people outside of the committee, like implementers or sibling committee members, who may otherwise miss important information.

Once a response has been communicated to the reporter, the reporter may have additional information they would like the committee to consider which causes the report to be reopened. In this case, the report goes back into `Open` designation so it can be deliberated on again.

4 An Unofficial Issues List

The issues list maintainer may support additional reporting mechanisms beyond direct email. One such extremely common mechanism is using an issue tracker like GitHub or GitLab Issues, Bugzilla, JIRA, etc.

If an issue tracker is used to reduce friction for external constituents, we strongly recommend using GitHub Issues (not our internal committee GitLab) to house the **unofficial** issues list. This repository could potentially also be used to house the prepublication content of the official issues list that is eventually submitted to the WG14 document tracker, utilities for managing the lists, etc. Additionally, GitHub has a REST API for querying the issues list, which can be used to generate the official issues list, if desired. We propose using GitHub instead of GitLab because of the reach – the committee GitLab instance has restricted access which causes a burden for reporters to overcome, especially infrequent ones. Given that GitHub is already frequently used in our industry (so many committee members and implementers may already have accounts), it seems like the better option between the two.

As a precaution, we've taken the liberty of reserving `sc22wg14` as an organization on GitHub to ensure we have a reasonable home for an unofficial public issue tracker should we adopt something along the lines of this process (`wg14`, `isoc`, and `c` were already taken). Additional GitHub users can be trivially added to the `sc22wg14` organization, so this resource is not necessarily owned or controlled by any one individual.

GitHub has the notion of labels which can be added to an issue to provide a measure of categorization. These labels need not be the same ones used by WG14 for the official issues list; they can be used to communicate more information at a glance, to ease in generation of the official issues list, etc.

One potential approach to labelling that could be used is to categorize reports with an `issue` or `question` label. (Note, there should not be a categorization for enhancements or feature requests.) Additional labels could be introduced if there is benefit to further categorization, such as `core` and `library` labels to differentiate between core language issues and library issues. The following labels could then be used for status categorization:

`unvetted`	Automatically applied when a report is first opened, removed after the issues list maintainer accepts the report.
`unresolved`	Applied when the report has been accepted by the issues list maintainer and is not yet resolved by the committee. Can be used with both `issue` and `question` labels. This label should not be on closed issues, only open ones.
`nad`	"Not a Defect"; applied when the report is an `issue` and the committee determines it is an invalid issue and should be closed. A `question` should not be closed with this status.
`answer`	Applied when a report is a `question`, the committee has a response, and the report should be closed. Note, if the answer leads to further questions, this label can be replaced with `unresolved` to signal another round of discussion is requested.
`<standard version>`	Applied when a report is an `issue`, the committee has resolved it by modifying the next version of the standard, and the report should be closed. e.g., for an issue reported against C99 that was resolved in C23, the `c23` label would be applied.
`invalid`	Applied when a report is closed as being invalid (spam, misreport, unresponsive reporter, etc). Can be used with both `issue` and `question` labels.
`hold`	Applied when an `unresolved` report is on hold because further information is required from the reporter.

Additional labels can be created and used at the issues list maintainer's discretion, but reporters should be prevented from inventing new tags to reduce the chances for confusion.

Once a report has been finalized by the committee and is closed in the official issues list, it should be closed in GitHub as well. An issue which is closed can be easily reopened, if necessary, without losing previous content or designations. All closed issues should have the correct label applied as to why it was closed and any communications from the committee added as comments.

The issues list maintainer should strive to keep the unofficial issues and the official issues lists as in-sync as possible as it is anticipated that constituents will appreciate the searchability, ease of use, notifications on changes, and more public nature of the unofficial issues list. One potential way of recording the committee response is to add the information to the unofficial issue and then generate a new official issues document from it so that both lists are kept in-sync. This has an added potential benefit of automatically notifying the original reporter to the committee response. However, the mechanisms for this are ultimately at the discretion of the issues list maintainer and the official issues list is the Source of Truth.

If everything works in the process, the official issues list and the unofficial issues list should both have the same content, just provided through different mediums. However, in the event of a discrepancy between the official and unofficial issues lists, the official issues list has priority over the unofficial issues list.

If an unofficial issue tracking mechanism gains significant traction among critical constituents, it may be worth formalizing it as part of our regular issue processing to provide better continuity across issues list maintainerships, but this is not proposed currently.

5 Prior Art

WG21 allows people to report Core Issues (issues pertaining to the core language wording rather than the standard template library wording) via GitHub [CWG]. Additionally, both the Core and the Library groups allow people to submit bug reports via email [WG21]. Once a report comes in from the "public" interface, the chair (or their delegate) removes any obviously incorrect reports or reaches out for incomplete reports to get them completed, and eventually moves the issues onto the committee's official issue trackers [CWG Issue Tracker, LWG Issue Tracker]. This process has struck a good balance between the needs of implementers, users, and the committee without being an overwhelming amount of work to track. The proposed WG14 process was largely inspired by the processes used by WG21, but with consideration for the difference in committee sizes.

6 Straw Polls

Does WG14 want to adopt something along the lines of the official issues list workflow in N3002?

Does WG14 want to explicitly discourage the issues list maintainer from using a public, unofficial issues list?

Does WG14 want to empanel a study group (or some other group) to discuss and propose possible resolutions for issues in advance of committee meetings?

7 Acknowledgements

I would like to recognize the following people for their help in this work: Jens Maurer, Erich Keane, Elizabeth Andrews, and Tom Honermann.

8 References

[N2947] Issue Tracking in C. Ballman. https://www.open-std.org/jtc1/sc22/wg14/www/docs/n2947.pdf

[Contributing] C – Contributing. WG14. http://www.open-std.org/jtc1/sc22/wg14/www/contributing.html

[CWG] GitHub cplusplus/CWG. WG21. https://github.com/cplusplus/CWG/issues/new/choose

[CWG Issue Tracker] C++ Standard Core Language Active Issues. WG21. http://www.open-std.org/jtc1/sc22/wg21/docs/cwg_active.html

[LWG Issue Tracker] C++ Standard Library Issues List. WG21. https://cplusplus.github.io/LWG/lwg-toc.html

[WG21] How To Submit a New Issue / Defect Report. WG21. https://isocpp.org/std/submit-issue