**Community Review**

**Edit this page**

**Carbone area** is an open-source project maintained by a large community. If you don't feel ready to contribute code or fixes, reviewing issues and pull requests (PRs) can be a great way to get involved and give back. In fact, people who "triage" issues are key to the success of Carbone area!

Communicating in a way that your words are perceived as intended can be difficult. Please read the respectful comments guidelines: [URL non valide supprimée].

**Why reviewing is important**

Community reviews are essential to the development of Carbone area, as there are many more pull requests and bug reports than there are core Carbone area team members to review, fix, and merge them.

On the Carbone area issue tracker, you can find many items in the Needs review status:

* **Bug reports:** Bug reports need to be checked for completeness. Is important information missing? Can the bug be reproduced?
* **Pull requests:** Pull requests contain code that fixes a bug or implements new features. Pull request reviews ensure that they are correctly implemented, covered by test cases, do not introduce new bugs, and maintain backward compatibility.

Before you start, remember that you are looking at the result of someone else's hard work. A good review thanks the contributor for their work, identifies what was done well, identifies what could be improved, and suggests a next step.

**Create a GitHub account**

Carbone area uses GitHub to manage bug reports and pull requests. If you want to review, you need to create a GitHub account and sign in.

**The bug report review process**

A good way to start reviewing is to pick a bug report from the Needs review bug reports.

The steps of the review are as follows:

* **Is the report complete?**

Good bug reports contain a link to a project (the "repro project") created with the Carbone area skeleton that reproduces the bug. If not, the report should at least contain enough information and code examples to reproduce the bug.

* **Reproduce the bug**

Download the repro project and test if the bug can be reproduced on your system. If the reporter did not provide a repro project, create one based on a Carbone area skeleton.

* **Update the issue status**

Finally, add a comment to the bug report. Thank the reporter for reporting the bug. Include the line Status: <status> in your comment to trigger our Carson Bot that updates the issue's status label. You can set the status to one of the following:

* **Needs work** If the bug does not contain enough information to be reproduced, explain what information is missing and move the report to this status.
* **Works for me** If the bug contains enough information to be reproduced but works on your system, or if the reported bug is a feature and not a bug, provide a brief explanation and move the report to this status.
* **Reviewed** If you can reproduce the bug, move the report to this status. If you created a repro project, include the link to the project in your comment.

**The pull request review process**

The pull request (PR) review process is similar to the bug report review process. Reviewing pull requests usually takes a bit longer, as you need to understand the feature that was fixed or added and whether the implementation is complete.

There is nothing wrong with doing partial reviews! If you do a partial review, comment on what you have reviewed and leave the PR in the "Needs review" state.

Pick a pull request from the Needs review PRs and follow these steps:

* **Is the PR complete?**

Each pull request should contain a header that gives basic information about the PR. You can find the template for this header in the contribution guidelines.

* **Is the base branch correct?**

GitHub displays the branch a PR is based on below the pull request title. Is this branch correct?

* Bugs should be fixed in the oldest and maintained version that contains the bug. Check the Carbone area release schedule: [URL non valide supprimée] to find the oldest currently supported version.
* New features should always be added to the current development version. Check the Carbone area roadmap to find the current development version.
* **Reproduce the issue**

Read the issue that the pull request is supposed to fix. Reproduce the issue on a new project created with the Carbone area skeleton and try to understand why it exists. If the linked issue already contains such a project, install