How to Submit Your First Pull Request

So you've decided to contribute to the Swift toolchain, welcome! Maybe this is your first time contributing to an open source project, or maybe you are an experienced open source contributor who is excited about Swift, or maybe you are somewhere in-between. Regardless of your background, we are excited to have you contribute and improve the developer experience for Swift programmers all over the globe. :sparkles:🧑🏾👱🏼🎓👩🏽‍💻🧑🏻💻👩🏿‍💻:sparkles:

This document provides a high-level overview of different parts of the contribution process.

How do I pick something to work on?

In case you don't have something specific you'd like to work on, such as implementing something for a Swift Evolution pitch, you could start off by working on a bug labeled `StarterBug` on Swift JIRA. If the bug hasn't been assigned to someone, check the comments in case someone has already started working on it. If not, feel free to assign it to yourself and start working on it!

Getting Help

Usually, Starter Bugs try to provide some instructions to help you get started. In case those are missing, please ask the bug reporter for more detailed steps and they will be happy to help.

Once you start working on the bug, you will inevitably end up having a lot of questions. Don't be afraid to ask for help! The codebase is large and wrapping your head around it will take time. For example, you might have questions like:

- Where can I find documentation on X?
- I'm seeing a cryptic error E when trying to build the compiler. How do I fix it or work around it?
- I'm seeing very long build times even for incremental builds. How do I speed up iteration time?
- I'm not sure how to implement X. Any suggestions on where I should start?
- What is the difference between types T1 and T2? They look very similar.
- Should I split my new X into a separate file?
- Should I create a new test file or update an existing test?
- How should I test that I actually fixed this bug?
- Test X is failing and I can't understand why. What might be going wrong here?
- Test X is failing in CI but passing locally. Any tips for debugging?
- I made some change but that seems to be not getting picked up. What should I do to fix it?
- I need to update the CMake but I'm not familiar with CMake. Could you give me more guidance?
- How do I do X in git?

Some of these already have answers in our FAQ. Maybe you have a question that's not on this list. That's fine. We're here to help. There are a couple of options to ask for help:

- Development category on the Swift forums: Prefer using the forums for broad questions, such as those related to building the toolchain, or understanding how something works at a high-level. Since more people are likely to see and be able to answer your question, the question is likely to get an answer sooner. Another benefit of asking in public is that the answers you receive will be helpful to bystanders too.
- Bug report/Pull request comments: Prefer asking in the bug report/pull request when the question involves additional context specific to the bug report/pull request.

These are suggestions, not rules. For example, it's okay if you ask a broad question in a bug report or a pull request.

When asking for help, prefer giving as much information as possible, while highlighting the parts that you think are important.

Remember that the Swift Code of Conduct applies whenever you are participating in the Swift project.

I didn't get a response from someone. What should I do?

It's possible that you ask someone a question in a bug report/pull request and you don't get a response as quickly as you'd like. Maybe they are juggling several tasks and the discussion with you accidentally slipped by. Maybe they are on vacation or on leave for some reason. If you don't get a response within a week, it's okay to politely ping them using an `@` mention with a reminder. If you don't get a response for 2-3 weeks in a row, please ping someone else.

Working on a change

Please see our Getting Started guide to understand how to build the code, make changes, run tests and debug issues.

Submitting a pull request

Tidying up

Alright! You've implemented a change and would like to submit it. Double-check that you've tidied your Git history, such as squashing work-in-progress commits, and that your commit messages provide context. For example, if a commit fixes a bug, then include a "Fixes SR-NNNNN" with the bug number in the commit message.

Next, format your changes using `clang-format`.

Pushing and creating a pull request

Assuming you followed the steps in our Getting Started guide, you should now be able to push your latest changes to GitHub using `git push`.

Next, create a pull request (PR). Usually, if you navigate to https://github.com/apple/swift right after pushing your change, GitHub will show a helpful "Compare & Pull Request" button.

Compare & Pull Request button in GitHub UI

Asking for code review

If you had an active discussion with someone on how to implement your change, you can `@` mention them in the PR description and ask for code review. If you directly implemented the change without any guidance from anyone else, `@` mention someone from GitHub's suggested reviewers. If GitHub doesn't make any suggestions, ask the Code Owner based on the component for your change. Please ask someone though! We don't want you to get stuck because you were not sure who to ask for code review.

At the beginning, contributors are not able to run the continuous integration (CI) bot, which builds the project and runs tests. Please ask your code reviewer(s) to invoke the bot for you.

Responding to code review comments

During the process of code review, someone might suggest changes or have questions about the implementation. If something is unclear, such as someone using a technical term you don't recognize, check our Lexicon or ask someone instead of trying to figure out everything by yourself. Code review does not need to be a one-way street. It is also a good opportunity for you to ask highly contextual questions on topics that you struggled with or were unable to understand.

While making changes based on code review, if you are comfortable with rebasing, prefer rebasing and force-pushing for small patches (say < 100 lines). For larger patches, you can add fixup commits (`git commit --fixup ...`) addressing the suggestions and rebase after it the patch has been approved to clean up the history.

When you push again and want the tests to be re-run, please ask the reviewer to invoke `swift-ci` for you.

At the end, once the tests are passing, the pull request is approved by the reviewer, and you are satisfied with your changes, ask your reviewer to merge your changes. :tada:

I can't finish the contribution I started. :frowning_face:

That's totally okay! There is no shame in that. You only have limited time and energy in a day. If you can, leave a comment on the bug report/pull request that you will not be able to continue and unassign yourself from the bug on JIRA. Don't worry about trying to explain *why* you aren't able to contribute further. We understand. Unanticipated things come up all the time and you should do what *works for you*.

This point also applies if you don't have time right now but hope to get to something in the near future. Please don't feel sad or apologetic!

I submitted and merged my first pull request. What now?

Awesome! You could try fixing a few more Starter Bugs until you feel some level of comfort working with the codebase. You could also start looking at other bugs which interest you and you think you might be able to tackle. Don't forget to ask for help if you need directions or you get stuck!

Once you've made multiple substantial contributions, you can ask for commit access, which will allow you to pick reviewers, trigger the CI bot and merge changes.