# Contributing to ruby-git

Thank you for your interest in contributing to the ruby-git project.

This document gives the guidelines for contributing to the ruby-git project.

These guidelines may not fit every situation. When contributing use your best

judgement.

Propose changes to these guidelines with a pull request.

## How to contribute to ruby-git

You can contribute in two ways:

1. [Report an issue or make a feature request](#how-to-report-an-issue-or-make-a-feature-request)

2. [Submit a code or documentation change](#how-to-submit-a-code-or-documentation-change)

## How to report an issue or make a feature request

ruby-git utilizes [GitHub Issues](https://help.github.com/en/github/managing-your-work-on-github/about-issues)

for issue tracking and feature requests.

Report an issue or feature request by [creating a ruby-git Github issue](https://github.com/ruby-git/ruby-git/issues/new).

Fill in the template to describe the issue or feature request the best you can.

## How to submit a code or documentation change

There is three step process for code or documentation changes:

1. [Commit your changes to a fork of ruby-git](#commit-changes-to-a-fork-of-ruby-git)

2. [Create a pull request](#create-a-pull-request)

3. [Get your pull request reviewed](#get-your-pull-request-reviewed)

### Commit changes to a fork of ruby-git

Make your changes in a fork of the ruby-git repository.

Each commit must include a [DCO sign-off](#developer-certificate-of-origin-dco)

by adding the line `Signed-off-by: Name <email>` to the end of the commit

message.

### Create a pull request

See [this article](https://help.github.com/articles/about-pull-requests/) if you

are not familiar with GitHub Pull Requests.

Follow the instructions in the pull request template.

### Get your pull request reviewed

Code review takes place in a GitHub pull request using the [the Github pull request review feature](https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/about-pull-request-reviews).

Once your pull request is ready for review, request a review from at least one

[maintainer](MAINTAINERS.md) and any number of other contributors.

During the review process, you may need to make additional commits which would

need to be squashed. It may also be necessary to rebase to master again if other

changes are merged before your PR.

At least one approval is required from a project maintainer before your pull

request can be merged. The maintainer is responsible for ensuring that the pull

request meets [the project's coding standards](#coding-standards).

## Coding standards

In order to ensure high quality, all pull requests must meet these requirements:

### 1 PR = 1 Commit

\* All commits for a PR must be squashed into one commit

\* To avoid an extra merge commit, the PR must be able to be merged as [a fast forward merge](https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging)

\* The easiest way to ensure a fast forward merge is to rebase your local branch

to the ruby-git master branch

### Unit tests

\* All changes must be accompanied by new or modified unit tests

\* The entire test suite must pass when `bundle exec rake test` is run from the

project's local working copy

### Continuous Integration

\* All tests must pass in the project's [Travis CI](https://travis-ci.org/ruby-git/ruby-git)

build before the pull request will be merged

### Documentation

\* New and updated public methods must have [YARD](https://yardoc.org/)

documentation added to them

\* New and updated public facing features should be documented in the project's

[README.md](README.md)

### Licensing sign-off

\* Each commit must contain [the DCO sign-off](#developer-certificate-of-origin-dco)

in the form: `Signed-off-by: Name <email>`

## Licensing

ruby-git uses [the MIT license](https://choosealicense.com/licenses/mit/) as

declared in the [LICENSE](LICENSE) file.

Licensing is very important to open source projects. It helps ensure the

software continues to be available under the terms that the author desired.

### Developer Certificate of Origin (DCO)

This project requires that authors have permission to submit their contributions

under the MIT license. To make a good faith effort to ensure this, ruby-git

requires the [Developer Certificate of Origin (DCO)](https://elinux.org/Developer\_Certificate\_Of\_Origin)

process be followed.

This process requires that each commit include a `Signed-off-by` line that

indicates the author accepts the DCO. Here is an example DCO sign-off line:

```

Signed-off-by: John Doe <john.doe@hisdomain.com>

```

The full text of the DCO version 1.1 is below or at <http://developercertificate.org/>.

```

Developer's Certificate of Origin 1.1

By making a contribution to this project, I certify that:

(a) The contribution was created in whole or in part by me and I

have the right to submit it under the open source license

indicated in the file; or

(b) The contribution is based upon previous work that, to the

best of my knowledge, is covered under an appropriate open

source license and I have the right under that license to

submit that work with modifications, whether created in whole

or in part by me, under the same open source license (unless

I am permitted to submit under a different license), as

Indicated in the file; or

(c) The contribution was provided directly to me by some other

person who certified (a), (b) or (c) and I have not modified

it.

(d) I understand and agree that this project and the contribution

are public and that a record of the contribution (including

all personal information I submit with it, including my

sign-off) is maintained indefinitely and may be redistributed

consistent with this project or the open source license(s)

involved.

```