# Contributing to OpenThread

We would love for you to contribute to OpenThread and help make it even better than it is today! As a contributor, here are the guidelines we would like you to follow.

- [1 Code of Conduct](#code-of-conduct)

- [2 Bugs](#bugs)

- [3 New Features](#new-features)

- [4 Contributing Code](#contributing-code)

- [4.1 Initial Setup](#initial-setup)

- [4.2 Contributor License Agreement (CLA)](#contributor-license-agreement--cla-)

- [4.3 Submitting a Pull Request](#submitting-a-pull-request)

## Code of Conduct

Help us keep OpenThread open and inclusive. Please read and follow our [Code of Conduct](CODE\_OF\_CONDUCT.md).

## Bugs

If you find a bug in the source code, you can help us by [submitting a GitHub Issue](https://github.com/openthread/openthread/issues/new). The best bug reports provide a detailed description of the issue and step-by-step instructions for predictably reproducing the issue. Even better, you can [submit a Pull Request](#submitting-a-pull-request) with a fix.

## New Features

You can request a new feature by [submitting a GitHub Issue](https://github.com/openthread/openthread/issues/new).

If you would like to implement a new feature, please consider the scope of the new feature:

- \_Large feature\_: first [submit a GitHub Issue](https://github.com/openthread/openthread/issues/new) and communicate your proposal so that the community can review and provide feedback. Getting early feedback will help ensure your implementation work is accepted by the community. This will also allow us to better coordinate our efforts and minimize duplicated effort.

- \_Small feature\_: can be implemented and directly [submitted as a Pull Request](#submitting-a-pull-request).

## Contributing Code

The OpenThread Project follows the "Fork-and-Pull" model for accepting contributions.

### Initial Setup

Setup your GitHub fork and continuous-integration services:

1. Fork the [OpenThread repository](https://github.com/openthread/openthread) by clicking "Fork" on the web UI.

Setup your local development environment:

```bash

# Clone your fork

git clone git@github.com:<username>/openthread.git

# Configure upstream alias

git remote add upstream git@github.com:openthread/openthread.git

```

### Contributor License Agreement (CLA)

Contributions to this project must be accompanied by a Contributor License Agreement. You (or your employer) retain the copyright to your contribution; this simply gives us permission to use and redistribute your contributions as part of the project. Head over to <https://cla.developers.google.com/> to see your current agreements on file or to sign a new one.

You generally only need to submit a CLA once, so if you've already submitted one (even if it was for a different project), you probably don't need to do it again.

### Submitting a Pull Request

#### Branch

For each new feature, create a working branch:

```bash

# Create a working branch for your new feature

git branch --track <branch-name> origin/master

# Checkout the branch

git checkout <branch-name>

```

#### Create Commits

```bash

# Add each modified file you'd like to include in the commit

git add <file1> <file2>

# Create a commit

git commit

```

This will open up a text editor where you can craft your commit message.

#### Upstream Sync and Clean Up

Prior to submitting your pull request, you might want to do a few things to clean up your branch and make it as simple as possible for the original repo's maintainer to test, accept, and merge your work.

If any commits have been made to the upstream master branch, you should rebase your development branch so that merging it will be a simple fast-forward that won't require any conflict resolution work.

```bash

# Fetch upstream master and merge with your repo's master branch

git checkout master

git pull upstream master

# If there were any new commits, rebase your development branch

git checkout <branch-name>

git rebase master

```

Now, it may be desirable to squash some of your smaller commits down into a small number of larger more cohesive commits. You can do this with an interactive rebase:

```bash

# Rebase all commits on your development branch

git checkout

git rebase -i master

```

This will open up a text editor where you can specify which commits to squash.

#### Coding Conventions and Style

OpenThread uses and enforces the [OpenThread Coding Conventions and Style](STYLE\_GUIDE.md) on all code, except for code located in [third\_party](third\_party). Use `script/make-pretty` and `script/make-pretty check` to automatically reformat code and check for code-style compliance, respectively. OpenThread currently requires [clang-format v6.0.0](http://releases.llvm.org/download.html#6.0.0) for C/C++ and [yapf v0.29.0](https://github.com/google/yapf) for Python.

As part of the cleanup process, you should also run `script/make-pretty check` to ensure that your code passes the baseline code style checks.

#### Push and Test

```bash

# Checkout your branch

git checkout <branch-name>

# Push to your GitHub fork:

git push origin <branch-name>

```

This will trigger continuous-integration checks using GitHub Actions. You can view the status and logs via the "Actions" tab in your fork.

#### Submit Pull Request

Once you've validated that all continuous-integration checks have passed, go to the page for your fork on GitHub, select your development branch, and click the pull request button. If you need to make any adjustments to your pull request, just push the updates to GitHub. Your pull request will automatically track the changes on your development branch and update.