# Contributing

## Issues

- Request features and report bugs using the [GitHub Issue Tracker](#).
- Please indicate the platform you are using fsnotify on.
- A code example to reproduce the problem is appreciated.

## Pull Requests

### Contributor License Agreement

fsnotify is derived from code in the [golang.org/x/exp](#) package and it may be included [in the standard library](#) in the future. Therefore fsnotify carries the same [LICENSE](#) as Go. Contributors retain their copyright, so you need to fill out a short form before we can accept your contribution: [Google Individual Contributor License Agreement](#).

Please indicate that you have signed the CLA in your pull request.

### How fsnotify is Developed

- Development is done on feature branches.
- Tests are run on BSD, Linux, macOS and Windows.
- Pull requests are reviewed and [applied to master](#) using [hub](#).
  - Maintainers may modify or squash commits rather than asking contributors to.
- To issue a new release, the maintainers will:
  - Update the CHANGELOG
  - Tag a version, which will become available through gopkg.in.

### How to Fork

For smooth sailing, always use the original import path. Installing with `go get` makes this easy.

1. Install from GitHub ( `go get -u github.com/fsnotify/fsnotify` )
2. Create your feature branch ( `git checkout -b my-new-feature` )
3. Ensure everything works and the tests pass (see below)
4. Commit your changes ( `git commit -am 'Add some feature'` )

Contribute upstream:

1. Fork fsnotify on GitHub
2. Add your remote ( `git remote add fork git@github.com:mycompany/repo.git` )
3. Push to the branch ( `git push fork my-new-feature` )
4. Create a new Pull Request on GitHub

This workflow is [thoroughly explained by Katrina Owen](#).

### Testing

fsnotify uses build tags to compile different code on Linux, BSD, macOS, and Windows.

Before doing a pull request, please do your best to test your changes on multiple platforms, and list which platforms you were able/unable to test on.

To aid in cross-platform testing there is a Vagrantfile for Linux and BSD.

- Install [Vagrant](#) and [VirtualBox](#)
- Setup [Vagrant Gopher](#) in your `src` folder.
- Run `vagrant up` from the project folder. You can also setup just one box with `vagrant up linux` or `vagrant up bsd` (note: the BSD box doesn't support Windows hosts at this time, and NFS may prompt for your host OS password)
- Once setup, you can run the test suite on a given OS with a single command `vagrant ssh linux -c 'cd fsnotify/fsnotify; go test'`.
- When you're done, you will want to halt or destroy the Vagrant boxes.

Notice: fsnotify file system events won't trigger in shared folders. The tests get around this limitation by using the /tmp directory.

Right now there is no equivalent solution for Windows and macOS, but there are Windows VMs [freely available from Microsoft](#).

## Maintainers

Help maintaining fsnotify is welcome. To be a maintainer:

- Submit a pull request and sign the CLA as above.
- You must be able to run the test suite on Mac, Windows, Linux and BSD.

To keep master clean, the fsnotify project uses the "apply mail" workflow outlined in Nathaniel Talbott's post ["Merge pull request" Considered Harmful](#). This requires installing [hub](#).

All code changes should be internal pull requests.

Releases are tagged using [Semantic Versioning](#).