# How to Contribute

We love Pull Requests! Your contributions help make LibGit2Sharp great.

## Getting Started

So you want to contribute to LibGit2Sharp. Great! Contributions take many forms from

submitting issues, writing documentation, to making code changes. We welcome it all.

But first things first...

\* Make sure you have a [GitHub account](https://github.com/signup/free)

\* Submit a ticket for your issue, assuming one does not already exist.

\* Clearly describe the issue including steps to reproduce when it is a bug.

\* Make sure you fill in the earliest version that you know has the issue.

\* Fork the repository on GitHub, then clone it using your favorite Git client.

\* Make sure the project builds and all tests pass on your machine by running

the `buildandtest.cmd` script on Windows or `buildandtest.sh` on Linux/Mac.

## LibGit2

LibGit2Sharp brings all the might and speed of libgit2, a native Git implementation, to the managed world of .Net and Mono.

LibGit2 is a git submodule referencing the [libgit2 project](https://github.com/libgit2/libgit2). To learn more about

submodules read [here](http://git-scm.com/book/en/v2/Git-Tools-Submodules).

To build libgit2 see [here](https://github.com/libgit2/libgit2sharp/wiki/How-to-build-x64-libgit2-and-LibGit2Sharp).

## Making Changes

Make sure you have the required .NET Core SDK and runtimes installed.

The easiest way to do this is run our `tools\Install-DotNetSdk.ps1` script.

Using the `-InstallLocality Machine` switch requires elevation but ensures

that Visual Studio will be able to load the solution even when launched from a shortcut.

Then proceed to:

\* Create a topic branch off master (don't work directly on master).

\* Implement your feature or fix your bug. Please following existing coding styles and do not introduce new ones.

\* Make atomic, focused commits with good commit messages.

\* Make sure you have added the necessary tests for your changes.

\* Run \_all\_ the tests to assure nothing else was accidentally broken.

## Submitting Changes

\* Push your changes to a topic branch in your fork of the repository.

\* Send a Pull Request targeting the master branch. Note what issue/issues your patch fixes.

Some things that will increase the chance that your pull request is accepted.

\* Following existing code conventions.

\* Including unit tests that would otherwise fail without the patch, but pass after applying it.

\* Updating the documentation and tests that are affected by the contribution.

\* If code from elsewhere is used, proper credit and a link to the source should exist in the code comments.

Then licensing issues can be checked against LibGit2Sharp's very permissive MIT based open source license.

\* Having a configured git client that converts line endings to LF. [See here.](https://help.github.com/articles/dealing-with-line-endings/).

# Additional Resources

\* [General GitHub documentation](http://help.github.com/)

\* [GitHub pull request documentation](https://help.github.com/articles/using-pull-requests/)