# \*\*Contributors Wanted: Inquire Within\*\* #

It isn't wise to build something that can be a part of someone's ecosystem without help, and we are always open to contributions to help us fix issues or work on features

## Getting Started ##

### Making Changes ###

- Make sure you have the latest version

- Submit a ticket for the issue, using the GitHub Issue Tracker

- It's worth checking first to see if someone else has raised your issue. We might have responded to them, or someone might be fixing it.

- If you want to add a feature, as opposed to fixing something that is broken, you should still raise an issue.

- That helps us understand what you want to add, so we can give more specific advice and check it does not conflict with some work in progress on a branch etc.

- If someone else is already working on the suggestion, then hopefully you can collaborate with them.

- Add a comment to an issue if you pick it up to work on, so everyone else knows.

- If you have a defect please include the following:

- Steps to reproduce the issue

- A failing test if possible (see below)

- The stacktrace for any errors you encountered

### Submitting Changes ###

0. Fork the project

0. Clone your fork

0. Branch your fork. This is your active development branch

0. Use Test Driven-Development.

0. New features should have a test or

0. Defects should have a test

0. You might want to watch [this video](http://vimeo.com/68375232) to understand our preferred testing approach

0. This project uses [MSpec](https://github.com/machine/machine.specifications) and [FakeItEasy](https://github.com/FakeItEasy/FakeItEasy). So should you to contribute.

0. We separate the core library from add-ons. Consider if your change is really core, or could be shipped as an add-on. Let the user 'buy-in' to your feature over making them take it.

0. Make your tests pass

0. Try to follow the [Microsoft .NET Framework Design Guidelines] (https://github.com/dotnet/corefx/tree/master/Documentation#coding-guidelines) when writing your code

0. Providing [BDD] (http://dannorth.net/introducing-bdd/) style tests should provide for the need to use scenarios to test the design of your API

1. Use the coding style from [dotnet/corefx] (https://github.com/dotnet/corefx/blob/master/Documentation/coding-guidelines/coding-style.md)

2. You can use [codeformatter] (https://github.com/dotnet/codeformatter) if you can run VS2015 to automatically update your format

3. Ensure you update the template for your copyright if using codeformatter

0. Commit

0. Try to write a [good commit message](http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html).

0. Merge back into your fork

0. Push to your fork

0. Submit a pull request

0. Sit back, and wait.

0. Try pinging @BrighterCommmand on Twitter if you hear nothing

### Contributor Licence Agreement ###

To safeguard the project we ask you to sign a Contributor Licence Agreement. The goal is to let you keep your copyright, but to assign it to the project so that it can use it in perpetuity. It is still yours, but the project is not at risk from having multiple contributors holding the copyright, with anyone able to hold it to ransom by removing their grant of licence.

The process of signing works through GitHub.

To get started, <a href="https://www.clahub.com/agreements/iancooper/Paramore">sign the Contributor License Agreement</a>.

### Contributor Code of Conduct ###

Please note that this project is released with a Contributor Code of Conduct. By participating in this project you agree to abide by its terms.

The code of conduct is from [Contributor Covenant](http://contributor-covenant.org/)