# How to Contribute

Thanks for wanting to contribute to Caliburn.Micro. Contributions don't need to take the form of code, submitting issues and writing documentation is just as important.

## FAQ: I see an error when opening the solution

```error : The expression "[System.IO.Path]::GetDirectoryName('')" cannot be evaluated. The path is not of a legal form.```

This (unhelpfully) seems to be caused by not having the correct .NET Core SDK installed (see https://github.com/novotnyllc/MSBuildSdkExtras/issues/190).

To find the version of the SDK that Caliburn Micro requires go to src/global.json.

e.g.

```

{

"sdk": {

"version": "3.1.300"

},

"msbuild-sdks": {

"MSBuild.Sdk.Extras": "2.0.54"

}

}

```

Once this version listed in the global.json file is installed the solution will open with all the projects loaded. If the correct versions of the SDKs for Android, UAP and Xamarin are not installed there are error messages at build time that contain instructions on how to proceed.

## Raising Issues

Basic support is provided by the community on [Stack Overflow][http://stackoverflow.com/questions/tagged/caliburn.micro] including the core contributors.

The following information is incredibly helpful when raising issues:

- \*\*Platform(s)\*\* - Caliburn.Micro runs on a lot of platforms and is tightly coupled with their UI stacks, knowing which platform or platforms you're experiencing the issue on helps immensely.

- \*\*Steps\*\* - Steps on recreating the issue.

- \*\*Sample Project\*\* - Even better that reproduction steps is a sample project that highlights the issue.

## Getting started with the source

You can clone this repository locally from GitHub using the "Clone in Desktop"

button from the main project site, or run this command in the Git Shell:

`git@github.com:Caliburn-Micro/Caliburn.Micro.git Caliburn.Micro`

If you want to make contributions to the project, [forking the project](https://help.github.com/articles/fork-a-repo) is the easiest way to do this. You can then clone down your fork instead:

`git clone git@github.com:MY-USERNAME-HERE/Caliburn.Micro.git Caliburn.Micro`

Documentation of building the code is [currently available](http://caliburnmicro.com/documentation/build) on the Caliburn.Micro website.

### How is the codebase organised?

The Caliburn.Micro is broken up into three groups of assemblies.

- \*\*Core\*\* - This contains the building blocks of any MVVM solution such `PropertyChangedBase`, `Screen` & `EventAggregator`. It's a [Portable Class Library](http://msdn.microsoft.com/en-us/library/gg597391.aspx) designed to work on as many platforms as possible. However due to some limitations we create the Core assembly for Silverlight 5 and .NET 4.0 separately although they share code using linked files and compilation symbols.

- \*\*Platform\*\* - For each platform Caliburn.Micro runs on we have a platform assembly, this contains the code that makes direct use of the appropriate UI stack and any code that could not be made portable. Most of the code is shared between the various platform projects using linked files and compilation symbols.

## Making Changes

When you're ready to make a change,

[create a branch](https://help.github.com/articles/fork-a-repo#create-branches)

off the `master` branch. We use `master` as the default branch for the

repository, and it holds the most recent contributions, so any changes you make

in master might cause conflicts down the track.

If you make focused commits (instead of one monolithic commit) and have descriptive

commit messages, this will help speed up the review process.

### Submitting Changes

You can publish your branch from GitHub for Windows, or run this command from

the Git Shell:

`git push origin MY-BRANCH-NAME`

Once your changes are ready to be reviewed, publish the branch to GitHub and

[open a pull request](https://help.github.com/articles/using-pull-requests)

against it.

A few little tips with pull requests:

- prefix the title with `[WIP]` to indicate this is a work-in-progress. It's

always good to get feedback early, so don't be afraid to open the PR before it's "done".

- use [checklists](https://github.com/blog/1375-task-lists-in-gfm-issues-pulls-comments)

to indicate the tasks which need to be done, so everyone knows how close you are to done.

- add comments to the PR about things that are unclear or you would like suggestions on

Don't forget to mention in the pull request description which issue/issues are

being addressed.

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

\* Follow existing code conventions. Most of what we do follows standard .NET

conventions except in a few places.

\* Include unit tests that would otherwise fail without your code, but pass with

it.

\* Update the documentation, the surrounding one, examples elsewhere, guides,

whatever is affected by your contribution

# Additional Resources

\* [Caliburn.Micro documentation](http://caliburnmicro.com/documentation/)

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

\* [Stack Overflow](http://stackoverflow.com/questions/tagged/caliburn.micro)