# Contributing

Bond welcomes contributions from the community.

## Process

1. Make a proposal.

1. Implement the proposal and its tests.

1. Squash commits and write a good commit message.

1. Start a pull request & address comments.

1. Merge.

### Proposal

For things like fixing typos and small bug fixes, you can skip this step.

If your change is more than a simple fix, please don't just create a big

pull request. Instead, start by opening an issue describing the problem you

want to solve and how you plan to approach the problem. This will let us

have a brief discussion about the problem and, hopefully, identify some

potential pitfalls before too much time is spent.

### Implementation

\* Fork the repository on GitHub.

\* Start on a new topic branch off of master.

\* Instructions for getting Bond building and running the tests are in the

[README](https://github.com/microsoft/bond/blob/master/README.md).

\* Aim for each pull request to have one commit. If the commit starts to get

too large, consider splitting it into multiple, independent pull requests.

\* Some changes are more naturally authored in multiple commits. This is

fine, though we find this to be a rare occurrence. These sort of changes

are harder to review and interate on with GitHub's tooling.

\* Make sure that all the tests continue to pass.

\* The CMake `check` target will run the C++ tests for you.

\* The C# unit tests can be run from

[the command line](https://github.com/microsoft/bond/blob/56b5914a5bb41178521e01f6ce078d429e3e6b71/appveyor.yml#L214)

or from within Visual Studio.

\* Update the [changelog](https://github.com/microsoft/bond/blob/master/CHANGELOG.md).

### Squash commits

Before submitting your pull request, you'll probably want to squash any

intermediate commits you have into one commit. No "fix syntax error"

commits, please. :-)

Each commit should build and pass all of the tests. If you want to add new

tests for functionality that's not yet written, ensure the tests are added

disabled.

\* For Haskell tests, add the tests functions, but don't add a test case for

them.

\* For CTest tests, add the executable to the CMakeLists.txt file to ensure

that it builds, but do not include a `add\_test()` (or similar) entry for

it.

\* For C# NUnit tests, use the

[`[Ignore]`](http://www.nunit.org/index.php?p=ignore&r=2.6.4) attribute to

annotate the test cases to be ignored.

Don't forget to run `git diff --check` to catch those annoying whitespace

changes.

In your commit message, explain the reasoning behind the commit. The code

changes answer the question "What changed?", but the commit message answers

the question "Why did it need to change?".

Please follow the established Git

[convention for commit messages](https://www.git-scm.com/book/en/v2/Distributed-Git-Contributing-to-a-Project#Commit-Guidelines).

The first line is a summary in the imperative, about 50 characters or less,

and should \*not\* end with a period. An optional, longer description must be

preceded by an empty line and should be wrapped at around 72 characters.

This helps with various outputs from Git or other tools.

You can update messages of local commits you haven't pushed yet using `git

commit --amend` or `git rebase --interactive` with \*reword\* command.

### Pull request

Start a GitHub pull request to merge your topic branch into the

[main repository's master branch](https://github.com/microsoft/bond/tree/master).

(If you are a Microsoft employee and are not a member of the

[Microsoft organization on GitHub](https://github.com/microsoft) yet, please

contact the Bond Development team via e-mail for instructions before

starting your pull request. There's some process stuff you'll need to do

ahead of time.)

If you haven't contributed to a Microsoft project before, you may be asked

to sign a [contribution license agreement](https://cla.microsoft.com/). A

comment in the PR will let you know if you do.

When you submit a pull request, a suite of builds and tests will be run

automatically, and the results will show up in the "Checks" section of the

PR. If any of these fail, you'll need to figure out why and make the

appropriate fixes. If you think the failures are due to infrastructure

issues, please mention this in a comment, and one of the maintainers will

help.

The project maintainers will review your changes. We aim to review all

changes within three business days.

Address any review comments by adding commits that address the comments.

Don't worry about having fixup/tiny commits at this stage. They'll get

squashed together when the change is finally merged. Push these new commits

to your topic branch, and we'll review the edits.

### Merge

Once the comments in the pull request have been addressed, a project

maintainer will merge your changes. Thank you for helping improve Bond!

By default we'll perform a squash merge unless requested otherwise in the

PR.

# Reporting Security Issues

Security issues and bugs should be reported privately, via email, to the

Microsoft Security Response Center (MSRC) at

[secure@microsoft.com](mailto:secure@microsoft.com). You should receive a

response within 24 hours. If for some reason you do not, please follow up

via email to ensure we received your original message. Further information,

including the [MSRC

PGP](https://technet.microsoft.com/en-us/security/dn606155) key, can be

found in the [Security

TechCenter](https://technet.microsoft.com/en-us/security/default).

# Code of Conduct

This project has adopted the

[Microsoft Open Source Code of Conduct](https://opensource.microsoft.com/codeofconduct/).

For more information see the

[Code of Conduct FAQ](https://opensource.microsoft.com/codeofconduct/faq/)

or contact [opencode@microsoft.com](mailto:opencode@microsoft.com) with any

additional questions or comments.