# Contribution Guidelines

\* [Prerequisites](#prerequisites)

\* [Definition of trivial contributions](#definition-of-trivial-contributions)

\* [Code](#code)

\* [Code style](#code-style)

\* [Dependencies](#dependencies)

\* [Unit tests](#unit-tests)

\* [Contributing process](#contributing-process)

\* [Get buyoff or find open community issues or features](#get-buyoff-or-find-open-community-issues-or-features)

\* [Set up your environment](#Set-up-your-environment)

\* [Prepare commits](#prepare-commits)

\* [Submit pull request](#Submit-pull-request)

\* [Respond to feedback on pull request](#respond-to-feedback-on-pull-request)

\* [Other general information](#other-general-information)

\* [Acknowledgement](#acknowledgement)

## Prerequisites

By contributing to Cake, you assert that:

\* The contribution is your own original work.

\* You have the right to assign the copyright for the work (it is not owned by your employer, or

you have been given copyright assignment in writing).

\* You [license](https://github.com/cake-build/cake/blob/develop/LICENSE) the contribution under the terms applied to the rest of the Cake project.

\* You agree to follow the [code of conduct](https://github.com/cake-build/cake/blob/develop/CODE\_OF\_CONDUCT.md).

## Definition of trivial contributions

It's hard to define what is a trivial contribution. Sometimes even a 1 character change can be considered significant.

Unfortunately because it can be subjective, the decision on what is trivial comes from the maintainers of the project

and not from folks contributing to the project.

What is generally considered trivial:

\* Fixing a typo.

\* Documentation changes.

\* Fixes to non-production code - like fixing something small in the build code.

What is generally not considered trivial:

\* Changes to any code that would be delivered as part of the final product.

This includes any scripts that are delivered, such as the PowerShell bootstrapper.

Yes, even 1 character changes could be considered non-trivial.

## Code

### Code style

Normal .NET coding guidelines apply.

See the [Framework Design Guidelines](https://msdn.microsoft.com/en-us/library/ms229042%28v=vs.110%29.aspx) for more information.

### Dependencies

The assemblies `Cake.Core` and `Cake.Common` should have no dependencies except the .NET BCL library.

Dependencies for the Cake executable is acceptable in specific circumstances.

If you want to introduce a dependency to the Cake executable, make sure you bring it

up in an issue or a pull request, so it can be properly discussed.

### Unit tests

Make sure to run all unit tests before creating a pull request.

Any new code should also have reasonable unit test coverage.

## Contributing process

### Get buyoff or find open community issues or features

\* Through GitHub, or through the [Gitter chat](https://gitter.im/cake-build/cake) (preferred),

you talk about a feature you would like to see (or a bug), and why it should be in Cake.

\* If approved through the Gitter chat, ensure an accompanying GitHub issue is created with

information and a link back to the discussion.

\* Once you get a nod from one of the [Cake Team](https://github.com/cake-build?tab=members), you can start on the feature.

\* Alternatively, if a feature is on the issues list with the

[Up For Grabs](https://github.com/cake-build/cake/labels/up-for-grabs) label,

it is open for a community member (contributor) to patch. You should comment that you are signing up for it on

the issue so someone else doesn't also sign up for the work.

### Set up your environment

\* You create, or update, a fork of cake-build/cake under your GitHub account.

\* From there you create a branch named specific to the feature.

\* In the branch you do work specific to the feature.

\* Please also observe the following:

\* No reformatting

\* No changing files that are not specific to the feature.

\* More covered below in the \*\*Prepare commits\*\* section.

\* Test your changes and please help us out by updating and implementing some automated tests.

It is recommended that all contributors spend some time looking over the tests in the source code.

You can't go wrong emulating one of the existing tests and then changing it specific to the behavior you are testing.

\* Please do not update your branch from the develop unless we ask you to. See the responding to feedback section below.

### Prepare commits

This section serves to help you understand what makes a good commit.

A commit should observe the following:

\* A commit is a small logical unit that represents a change.

\* Should include new or changed tests relevant to the changes you are making.

\* No unnecessary whitespace. Check for whitespace with `git diff --check` and `git diff --cached --check` before commit.

\* You can stage parts of a file for commit.

### Submit pull request

Prerequisites:

\* You are making commits in a feature branch.

\* All code should compile without errors or warnings.

\* All tests should be passing.

Submitting PR:

\* Once you feel it is ready, submit the pull request to the `cake-build/cake` repository against the `develop` branch

unless specifically requested to submit it against another branch.

\* In the case of a larger change that is going to require more discussion,

please submit a PR sooner. Waiting until you are ready may mean more changes than you are

interested in if the changes are taking things in a direction the maintainers do not want to go.

\* In the pull request, outline what you did and point to specific conversations (as in URLs)

and issues that you are resolving. This is a tremendous help for us in evaluation and acceptance.

\* Once the pull request is in, please do not delete the branch or close the pull request

(unless something is wrong with it).

\* One of the Cake team members, or one of the maintainers, will evaluate it within a

reasonable time period (which is to say usually within 1-3 weeks). Some things get evaluated

faster or fast tracked. We are human and we have active lives outside of open source so don't

fret if you haven't seen any activity on your pull request within a month or two.

We don't have a Service Level Agreement (SLA) for pull requests.

Just know that we will evaluate your pull request.

### Respond to feedback on pull request

We may have feedback for you to fix or change some things. We generally like to see that pushed against

the same topic branch (it will automatically update the Pull Request). You can also fix/squash/rebase

commits and push the same topic branch with `--force` (it's generally acceptable to do this on topic

branches not in the main repository, it is generally unacceptable and should be avoided at all costs

against the main repository).

If we have comments or questions when we do evaluate it and receive no response, it will probably

lessen the chance of getting accepted. Eventually, this means it will be closed if it is not accepted.

Please know this doesn't mean we don't value your contribution, just that things go stale. If in the

future you want to pick it back up, feel free to address our concerns/questions/feedback and reopen

the issue/open a new PR (referencing old one).

Sometimes we may need you to rebase your commit against the latest code before we can review it further.

If this happens, you can do the following:

\* `git fetch upstream` (upstream would be the mainstream repo or `cake-build/cake` in this case)

\* `git checkout develop`

\* `git rebase upstream/develop`

\* `git checkout your-branch`

\* `git rebase develop`

\* Fix any merge conflicts

\* `git push origin your-branch` (origin would be your GitHub repo or `your-github-username/cake` in this case).

You may need to `git push origin your-branch --force` to get the commits pushed.

This is generally acceptable with topic branches not in the mainstream repository.

The only reasons a pull request should be closed and resubmitted are as follows:

\* When the pull request is targeting the wrong branch (this doesn't happen as often).

\* When there are updates made to the original by someone other than the original contributor.

Then the old branch is closed with a note on the newer branch this supersedes #github\_number.

## Other general information

If you reformat code or hit core functionality without an approval from a person on the Cake Team,

it's likely that no matter how awesome it looks afterwards, it will probably not get accepted.

Reformatting code makes it harder for us to evaluate exactly what was changed.

If you do these things, it will be make evaluation and acceptance easy.

Now if you stray outside of the guidelines we have above, it doesn't mean we are going to ignore

your pull request. It will just make things harder for us.

Harder for us roughly translates to a longer SLA for your pull request.

## Acknowledgement

This contribution guide was taken from the [Chocolatey project](https://chocolatey.org/)

with permission and was edited to follow Cake's conventions and processes.