Guidelines

==========

When contributing to the Mono project, please follow the [Mono Coding

Guidelines][1]. We have been using a coding style for many years,

please make your patches conform to these guidelines.

[1]: http://www.mono-project.com/community/contributing/coding-guidelines/

Etiquette

=========

In general, we do not accept patches that merely shuffle code around,

split classes in multiple files, reindent the code or are the result

of running a refactoring tool on the source code. This is done for

three reasons: (a) we have our own coding guidelines; (b) Some modules

are imported from upstream sources and we want to respect their coding

guidelines and (c) it destroys valuable history that is often used to

investigate bugs, regressions and problems.

License

=======

The Mono runtime, compilers, and tools and most of the class libraries

are licensed under the MIT license. But include some bits of code

licensed under different licenses. The exact list is [available here](https://github.com/mono/mono/blob/master/LICENSE).

Different parts of Mono use different licenses. The actual details of

which licenses are used for which parts are detailed on the LICENSE

file in this directory.

CLA

=======

Contributions are now taken under the [.NET Foundation CLA](https://cla.dotnetfoundation.org/).

Code Review

=======

We are requiring code reviews before merging a pull request.

GitHub won't let you merge a PR until someone from the Mono team approves it.

Testing

=======

Pull requests go through testing on our [Jenkins server][2]. We will

usually only merge a pull request if it causes no regressions in a

test run there.

When you submit a pull request, one of two things happens:

\* If you are a new contributor, Jenkins will ask for permissions (on

the pull request) to test it. A maintainer will reply to approve

the test run if they find the patch appropriate. After you have

submitted a few patches, a maintainer will whitelist you so that

all of your future pull requests are tested automatically.

\* If you are a well-known, whitelisted contributor, Jenkins will go

ahead and test your pull request as soon as a test machine is

available.

When your pull request has been built, Jenkins will update the build

status of your pull request. If it succeeded and we like the changes,

a maintainer will likely merge it. Otherwise, you can amend your pull

request to fix build breakage and Jenkins will test it again.

[2]: http://jenkins.mono-project.com/

# Inactivity

Occasionally, a pull request sits for several months without any

response from the author. This isn't necessarily an issue, but we may

sometimes decide to close pull requests that have not seen any

progress for a long time. This is in interest of keeping the pull

request list clean so that other pull requests don't get lost in the

clutter.

If we do close your pull request due to inactivity, you're more than

welcome to submit it anew after you address any comments or issues that

were brought up on the original pull request.