# How to contribute to Open MPI

First off, thank you for taking the time to prepare a contribution to

Open MPI!

![You're awesome!](https://www.open-mpi.org/images/youre-awesome.jpg)

General information about contributing to the Open MPI project can be found at the [Contributing to Open MPI webpage](https://www.open-mpi.org/community/contribute/).

The instructions below are specifically for opening issues and pull requests against Open MPI.

## Content

We love getting contributions from anyone. But keep in mind that Open

MPI is used in production environments all around the world.

If you're contributing a small bug fix, awesome!

If you're contributing a large new piece of functionality, that will

be best viewed if you — or someone, anyone — is also stepping up to

help maintain that functionality over time. We love new ideas and new

features, but we do need to be realistic in what we can reliably test

and deliver to our users.

## Contributor's Declaration

In order to ensure that we can keep distributing Open MPI under our

[open source license](/LICENSE), we need to ensure that all

contributions are compatible with that license.

To that end, we require that all Git commits contributed to Open MPI

have a "Signed-off-by" token indicating that the commit author agrees

with [Open MPI's Contributor's

Declaration](https://github.com/open-mpi/ompi/wiki/Administrative-rules#contributors-declaration).

If you have not already done so, please ensure that:

1. Every commit contains exactly the "Signed-off-by" token. You can

add this token via `git commit -s`.

1. The email address after "Signed-off-by" must match the Git commit

email address.

## \*\*Did you find a bug?\*\*

\* \*\*Ensure the bug was not already reported\*\* by searching on GitHub under [Issues](https://github.com/open-mpi/ompi/issues).

\* If you're unable to find an open issue addressing the problem, [open a new one](https://github.com/open-mpi/ompi/issues/new).

\* For more detailed information on submitting a bug report and creating an issue, visit our [Bug Tracking webpage](https://www.open-mpi.org/community/help/bugs.php).

## \*\*Did you write a patch that fixes a bug?\*\*

\* Open a new GitHub pull request with the patch.

\* Ensure the PR description clearly describes the problem and solution. If there is an existing GitHub issue open describing this bug, please include it in the description so we can close it.

\* Before submitting, please read the [Contributing to the Open MPI Project FAQ](https://www.open-mpi.org/faq/?category=contributing) web page, and the [SubmittingPullRequests](https://github.com/open-mpi/ompi/wiki/SubmittingPullRequests) wiki page. In particular, note that all Git commits contributed to Open MPI require a "Signed-off-by" line.

## \*\*Do you intend to add a new feature or change an existing one?\*\*

\* Suggest your change on the [devel mail list](https://www.open-mpi.org/community/lists/ompi.php) and start writing code. The [developer level technical information on the internals of Open MPI](https://www.open-mpi.org/faq/?category=developers) may also be useful for large scale features.

\* Do not open an issue on GitHub until you have collected positive feedback about the change. GitHub issues are primarily intended for bug reports and fixes.

## \*\*Do you have questions about the source code?\*\*

\* First checkout the [developer level technical information on the internals of Open MPI](https://www.open-mpi.org/faq/?category=developers). A paper describing the [multi-component architecture](https://www.open-mpi.org/papers/ics-2004/ics-2004.pdf) of Open MPI may also be helpful. The [devel mail list](https://www.open-mpi.org/community/lists/ompi.php) is a good place to post questions about the source code as well.

## Style

There are a small number of style rules for Open MPI:

1. For all code:

\* 4 space tabs. No more, no less.

\* No tab characters \*at all\*. 2 indentations are 8 spaces — not a tab.

\* m4 code is a bit weird in terms of indentation: we don't have a

good, consistent indentation style in our existing code. But

still: no tab characters at all.

1. For C code:

\* We prefer if all blocks are enclosed in `{}` (even 1-line

blocks).

\* We prefer that if you are testing equality with a constant, put

the constant on the \*left\* of the `==`. E.g., `if (NULL ==

ptr)`.

\* If there are no parameters to a C function, declare it with

`(void)` (vs. `()`).

That's about it. Thank you!

— The Open MPI Team