# How To Contribute

There are several ways to contribute to the development of ReactOS. The most often encountered problem is not knowing where to begin or what to do. If you are able to program or understand the technical information that is pertinent to this project, helping the development can be easy.

- [What To Do?](#what-to-do?)

- [How To Contribute?](#how-to-contribute?)

- [Where To Start?](#where-to-start?)

\*\*Legal notice:\*\* If you have seen Microsoft Windows source code, your contribution won't be accepted because of potential copyright violation. Before contributing, you must affirm that the following is true:

>I hereby swear that I have not used nor seen the source code to any version of the Windows operating system

>nor any Microsoft product that may be related to the proposed project that is under a license incompatible

>with contribution to ReactOS, including but not limited to the leaked Windows 2000 source code and the Windows Research Kernel.

## What To Do?

### Fix bugs

You can try to fix a few bugs that are already listed in [JIRA]. Squashing bugs is not a simple task. It requires a lot more skill than simply searching for them, and can be time consuming; however, by doing that you greatly help ReactOS become a stable system.

\_NOTE: patches related to 3rd party code such as Wine or BtrFS should be sent to upstream of the said projects. See [3rd Party Files.txt], [README.WINE] and [README.FSD] in [media/doc](media/doc) for details.\_

### Fix tests

Tests are used to check the functionality and correctness of APIs on ReactOS compared to Windows implementations. There are some unit tests that you could help ReactOS pass, which can be found [in the Web Test Manager][testman] and some that are broken or yet to be written.

### Fix Coverity scans

[Coverity] is enhanced static analysis that uncovers leaks, buffer overflows, security issues and other. We do such scans on ReactOS codebase pretty often. You can [request][request-coverity] to see Coverity 'defects' and help to fix them.

### Test ReactOS

By localizing bugs, developers can identify what causes the bug and which part it affects. There are a variety of methods to [debug] ReactOS while testing it. After identifying a bug, check if it is already known about by searching on JIRA and adding any additional information to the report. If you think that it is an unidentified bug, consider [filing a bug report].

### Implement new things

Considering ReactOS is alpha quality software, there is a lot of [missing functionality] that Windows operating systems have. Before starting a project to implement something, find out whether another person is working on the same thing. If you find that someone is already working on it, ask if any assistance is needed for what specifically is being worked on or a related project. More often than not, someone will start to implement something and move onto something else before it's complete. Make sure you stay committed to what you are going to implement, and do not be afraid to ask for assistance if you need help with something.

### Write documentation

There are some important points if you would like to help document ReactOS:

1. Make sure the documentation does not exist yet (if it does, help improve it).

2. Respect [clean room reverse engineering] practices.

3. Add your knowledge to a place where the other developers can find it.

## How To Contribute?

Your contribution can be of numerous forms. We currently accept two ways to contribute - Pull Requests and Patches.

### Pull Requests

Since our [migration to GitHub] we gladly accept \_\_[Pull Requests]\_\_. Pull requests let you tell others about changes you have pushed to a repository on GitHub. Once a pull request is opened, you can discuss and review the potential changes with collaborators and add follow-up commits before the changes are merged into the repository. \_\_Pull request is a preferred way to submit your work\_\_ - it makes reviewing and merging your contribution much easier.

### Patches

A \_\_[patch]\_\_ is a set of changes to existing source code. The changes in a patch can be merged into existing source code. This process is referred to as applying a patch (to source code). Which changes a patch contains and the way the patch is structured can have significant impact on the consequences that can happen from applying the patch.

See [Submitting Patches] for details.

### Commit style

Our commit style is defined in a \_\_[commit template]\_\_. Use it as a reference or turn it on using `git config commit.template .gitmessage`. This will set this template as an initial commit message for the new commits in your local repository.

### Rules and Recommendations

- \*Use your \_\_real name\_\_ and \_\_real email\_\_.\* We do not accept anonymous contributions!

- \*Ensure your contribution is properly described.\* Include the relevant issue number if applicable.

- \*Put only related changes.\* It will make reviewing easier as the reviewer needs to recall less information about the existing source code that is changed.

- \*Search for similar pull requests/patches before submitting.\* It may be that a similar pull request or issue was opened previously. Comment and review on that one instead.

- \*Keep your contribution small and focused on the topic.\* It can be tempting to fix existing issues as you come across them while reading the source code. Resist the temptation and put in a note in the source code instead, or (even better) put the issue in the issue tracking system.

- \*Respect our \_\_[Coding Style]\_\_ and \_\_[Programming Guidelines]\_\_.\*

- \*Do not be afraid to ask questions.\* Ask our developers on JIRA or [IRC] channel.

To amend your commit with your name and e-mail (in any case you've forgot to set your name/e-mail) please take a look at this [guide](https://reactos.org/wiki/ReactOS\_Git\_For\_Dummies#Amending\_your\_commit\_with\_name.2FE-mail). To set your name/e-mail globally for future commits that you push, [read this](https://reactos.org/wiki/ReactOS\_Git\_For\_Dummies#Assign\_commits\_with\_your\_name\_.26\_E-mail\_automatically).

## Where To Start?

Finding a good project to start with can be a challenge, because when starting out you are (usually) not aware of all the possibilities. To help you find a project, here are some ideas to try:

- Find a test that fails, and try to make it succeed: <https://reactos.org/testman/>

- Look around in JIRA, and if you have problems finding nice projects to start with, there is a label for this: <https://jira.reactos.org/issues/?jql=labels%20%3D%20starter-project>

- Ask for help on [IRC]

- Additionally, there are some tests that cause crashes/hangs, but these might be slightly harder: <https://jira.reactos.org/browse/ROSTESTS-125>

[clean room reverse engineering]: https://en.wikipedia.org/wiki/Clean\_room\_design

[debug]: https://reactos.org/wiki/Debugging

[JIRA]: https://jira.reactos.org/

[filing a bug report]: https://reactos.org/wiki/File\_Bugs

[testman]: https://reactos.org/testman/

[migration to GitHub]: https://reactos.org/project-news/reactos-repository-migrated-github/

[humans are terrible at tracking large amount of information]: https://www.eurekalert.org/pub\_releases/2005-03/aps-hmc030805.php

[Pull requests]: https://help.github.com/articles/about-pull-requests/

[tips for reviewing patches]: https://www.drupal.org/patch/review

[missing functionality]: https://reactos.org/wiki/Missing\_ReactOS\_Functionality

[patch]: https://git-scm.com/docs/git-format-patch

[Submitting Patches]: https://reactos.org/wiki/Submitting\_Patches

[Coding Style]: https://reactos.org/wiki/Coding\_Style

[IRC]: https://reactos.org/wiki/Connect\_to\_the\_ReactOS\_IRC\_Channels

[Programming Guidelines]: https://reactos.org/wiki/Programming\_Guidelines

[3rd Party Files.txt]: /media/doc/3rd\_Party\_Files.txt

[README.WINE]: /media/doc/README.WINE

[README.FSD]: /media/doc/README.FSD

[Coverity]: https://scan.coverity.com/projects/reactos

[request-coverity]: https://scan.coverity.com/memberships/new?project\_id=reactos

[commit template]: .gitmessage

# See Also

- [Rules for managing Pull Requests](PULL\_REQUEST\_MANAGEMENT.md)