# Contributing to OpenTTD

Looking to contribute something to OpenTTD? \*\*Here's how you can help.\*\*

Please take a moment to review this document in order to make the contribution process easy and effective for everyone involved.

Following these guidelines helps to communicate that you respect the time of the developers managing and developing this open source project.

In return, they should reciprocate that respect in addressing your issue or assessing patches and features.

## Using the issue tracker

The [issue tracker](https://github.com/OpenTTD/OpenTTD/issues) is the preferred channel for [bug reports](#bug-reports), but please respect the following restrictions:

\* Please \*\*do not\*\* use the issue tracker for help playing or using OpenTTD.

Please try [irc](https://wiki.openttd.org/IRC\_channel), or the [forums](https://www.tt-forums.net/)

\* Please \*\*do not\*\* derail or troll issues. Keep the discussion on topic and respect the opinions of others.

\* Please \*\*do not\*\* post comments consisting solely of "+1" or ":thumbsup:".

Use [GitHub's "reactions" feature](https://github.com/blog/2119-add-reactions-to-pull-requests-issues-and-comments) instead.

We reserve the right to delete comments which violate this rule.

\* Please \*\*do not\*\* open issues or pull requests regarding add-on content in NewGRF, GameScripts, AIs, etc.

These are created by third-parties. Please try [irc](https://wiki.openttd.org/IRC\_channel) or the [forums](https://www.tt-forums.net/) to discuss these.

\* Please use [the web translator](https://translator.openttd.org/) to submit corrections and improvements to translations of the game.

## Bug reports

A bug is a \_demonstrable problem\_ that is caused by the code in the repository.

Good bug reports are extremely helpful, so thanks!

Guidelines for bug reports:

0. Please don't report issues with games where you changed NewGRFs mid-game. (This can be verified with the `gamelog` console command in-game.)

1. Please don't report issues with modified versions of OpenTTD (patchpacks, unofficial ports, and similar).

2. \*\*Use the GitHub issue search\*\* — check if the issue has already been

reported.

3. \*\*Check if the issue has been fixed\*\* — try to reproduce it using the latest `nightly` build of OpenTTD, available from https://www.openttd.org

4. \*\*Isolate the problem\*\* — ideally create reproducible steps with an attached savegame and screenshots. Try to use few or no NewGRFs, AIs etc if possible.

A good bug report shouldn't leave others needing to chase you up for more information.

Please try to be as detailed as possible in your report.

\* What is your environment?

\* What steps will reproduce the issue?

\* Which operating system(s) experience the problem?

\* What would you expect to be the outcome?

All these details will help people to fix any potential bugs.

Example:

> Short and descriptive example bug report title

>

> A summary of the issue and the OS environment in which it occurs. If

> suitable, include the steps required to reproduce the bug.

>

> 1. This is the first step

> 2. This is the second step

> 3. Further steps, etc.

>

> Attached savegame

> Attached screenshots showing the issue

> Crashlogs if the bug causes a crash

>

> Any other information you want to share that is relevant to the issue being

> reported. This might include the lines of code that you have identified as

> causing the bug, and potential solutions (and your opinions on their

> merits).

## Feature requests

Before opening a feature request, please take a moment to find out whether your idea fits with the [scope and goals](./CONTRIBUTING.md#project-goals) of the project.

It's up to \*you\* to make a strong case to convince the project's developers of the merits of this feature.

Please provide as much detail and context as possible.

This means don't request for a solution, but describe the problem you see and how/why you think it should be fixed.

For feature request we have a strict policy.

Keeping issues around with "a good idea" or "not really a bug but we should maybe fix it" turns out to have the reversed effect: nobody looks at it anymore.

Although we really appreciate feedback and ideas, we will close feature requests that we don't expect to fulfill in the next year.

Many of those ideas etc do have a place on the [forums](https://www.tt-forums.net); and if enough people like it, someone will stand up and make it.

It's usually best discuss in [irc](https://wiki.openttd.org/IRC\_channel) before opening a feature request or working on a large feature in a fork.

Discussion in irc can take time, but it can be productive and avoid disappointment :)

## Pull requests

Good pull requests—patches, improvements, new features—are a fantastic help.

Pull requests should fit with the [goals of the project](./CONTRIBUTING.md#project-goals).

\*\*Please do ask first\*\* before embarking on any significant pull request (e.g. implementing features, refactoring code, porting to a different language), otherwise you risk spending a lot of time working on something that the project's developers might not want to merge into the project.

Every pull request should have a clear scope, with no unrelated commits.

[Code style](https://wiki.openttd.org/Coding\_style) must be complied with for pull requests to be accepted; this also includes [commit message format](https://wiki.openttd.org/Coding\_style#Commit\_message).

Adhering to the following process is the best way to get your work included in the project:

1. [Fork](https://help.github.com/fork-a-repo/) the project, clone your fork, and configure the remotes:

```bash

git clone https://github.com/<your-username>/OpenTTD.git openttd

git clone https://github.com/OpenTTD/OpenTTD-git-hooks.git openttd\_hooks

cd openttd

git remote add upstream https://github.com/OpenTTD/OpenTTD.git

cd .git/hooks

ln -s ../../../openttd\_hooks/hooks/\* .

```

2. If you cloned a while ago, get the latest changes from upstream:

```bash

git fetch upstream

```

3. Create a new topic branch (off the main project development branch) to

contain your feature, change, or fix:

```bash

git checkout upstream/master -b <topic-branch-name>

```

4. Commit your changes in logical chunks. Please adhere to these [git commit message guidelines](https://wiki.openttd.org/Commit\_style#Commit\_message) or your code is unlikely to be merged into the main project.

Use Git's [interactive rebase](https://help.github.com/articles/interactive-rebase) feature to tidy up your commits before making them public.

5. Locally rebase the upstream development branch into your topic branch:

```bash

git fetch upstream

git rebase upstream/master

```

6. Push your topic branch up to your fork the first time:

```bash

git push --set-upstream origin <topic-branch-name>

```

And any time after that:

```bash

git push

```

7. [Open a Pull Request](https://help.github.com/articles/using-pull-requests/) with a clear title and description against the `master` branch.

\*\*IMPORTANT\*\*: By submitting a pull request or patch, you agree to the [License](#license) and the [Privacy Notice](CONTRIBUTING.md#privacy-notice).

### Pull request validation

Continuous integration (CI) tools monitor pull requests, and help us identify build and code quality issues.

The results of the CI tests will show on your pull request.

By clicking on Details you can further zoom in; in case of a failure it will show you why it failed.

In case of success it will report how awesome you were.

Tip: [commit message format](https://wiki.openttd.org/Coding\_style#Commit\_message) is a common reason for pull requests to fail validation.

### Are there any development docs?

There is no single source for OpenTTD development docs. It's a complex project with a long history, and multiple APIs.

A good entry point is [Development](https://wiki.openttd.org/Development) on the OpenTTD wiki; this provides links to wiki documentation and other sources.

The GitHub repo also includes some non-comprehensive documentation in [/docs](./docs).

You may also want the guide to [compiling OpenTTD](./COMPILING.md).

## Project goals

### What are the goals of the official branch?

The main goals of the official branch are:

- Stay faithful to the original gameplay from Transport Tycoon Deluxe

- Improve the user interface

- Allow extending the gameplay with add-ons / mods via supported content APIs

- Provide a (relatively) stable core for both players of the official branch, and for authors of add-ons and maintainers of patchpacks

In contrast, extending or altering the gameplay of the base game is not encouraged.

The rationale behind these goals is that people have different opinions about what OpenTTD is and what it should be.

When it comes to gameplay, there are at least these groups of people:

- \*Model railway (mostly singleplayer)\*: build "realistic" landscapes, roleplay a world, or even replicate historical scenarios

- \*Economical challenge (mostly singleplayer)\*: run a business with economical challenges

- \*Transport challenge (singleplayer or cooperative multiplayer)\*: build efficient track layouts with high cargo throughput and tons of vehicles

- \*Competitive speed run (competitive multiplayer)\*: maximize some goal in some limited amount of time

When it comes to gameplay features there are at least these groups of interests:

- \*Control freak:\* micromanagement like conditional orders, refitting and loading etc.

- \*Casual:\* automatisation like cargodist, path based signalling etc.

To please everyone, the official branch tries to stay close to the original gameplay; after all, that is what everyone brought here.

The preferred method to alter and extent the gameplay is via add-ons like NewGRF and GameScripts.

For a long time, the official branch was also open to features which could be enabled/disabled, but the corner-cases that came with some configurations have rendered some parts of the code very complicated.

Today, new features have to work with all the already existing features, which is not only challenging in corner cases, but also requires spending considerable more work than just "making it work in the game mode that I play".

The preferred method to introduce new gameplay features is to extend the content APIs, supporting ever more add-on content / mods.

This moves conflict-solving away from the codebase to content authors / players.

It is more accepted for add-ons not working together than the base game not working with certain setting combinations.

In general the game should allow anything that doesn't violate basic rules, but it should warn players if they take potentially dangerous or "stupid" actions.

For example, players are not prevented from starting vehicles without orders, but will receive a warning about vehicles having too few orders.

This lack of limitation has led to players challenging themselves to create networks where all vehicles have no orders, increasing gameplay possibilities.

### I do not agree with the goals of the official branch, what can I do instead?

Fork! There is a rich history of experimental patches for OpenTTD.

Many of these will never be accepted for core, but are creative and interesting ways to modify OpenTTD.

Sometimes patches are combined into long-running patchpacks, modified OpenTTD versions which can be downloaded by anyone, or modified OpenTTD clients for dedicated multiplayer servers.

One of the reasons to keep core relatively stable is to make life easier for patch authors and patchpack maintainers where possible.

Patchpack discussions and related topics may be found in community sites such as [TT-Forums development section](https://www.tt-forums.net/viewforum.php?f=33).

## Legal stuff

### License

By contributing your code, you agree to license your contribution under the [GPL v2](https://github.com/OpenTTD/OpenTTD/blob/master/COPYING.md).

### Privacy Notice

We would like to make you aware that contributing to OpenTTD via git will permanently store the name and email address you provide as well as the actual changes and the time and date you made it inside git's version history.

This is inevitable, because it is a main feature of git.

If you are concerned about your privacy, we strongly recommend to use "Anonymous &lt;anonymous@openttd.org&gt;" as the git commit author. We might refuse anonymous contributions if malicious intent is suspected.

Please note that the contributor identity, once given, is used for copyright verification and to provide proof should a malicious commit be made.

As such, the [EU GDPR](https://www.eugdpr.org/key-changes.html) "right to be forgotten" does not apply, as this is an overriding legitimate interest.

Please also note that your commit is public and as such will potentially be processed by many third-parties.

Git's distributed nature makes it impossible to track where exactly your commit, and thus your personal data, will be stored and be processed.

If you would not like to accept this risk, please do either commit anonymously or refrain from contributing to the OpenTTD project.

### Attribution of this Contributing Guide

This contributing guide is adapted from [Bootstrap](https://github.com/twbs/bootstrap/blob/master/CONTRIBUTING.md) under the [Creative Commons Attribution 3.0 Unported License](https://github.com/twbs/bootstrap/blob/master/docs/LICENSE) terms for Bootstrap documentation.

The GDPR notice is adapted from [rsyslog](https://github.com/rsyslog/rsyslog/blob/master/CONTRIBUTING.md) under the [GNU General Public License](https://github.com/rsyslog/rsyslog/blob/master/COPYING).