# Contributing to osquery

We want to make contributing to osquery as simple and transparent as

possible. These guidelines explain the basics of the osquery

development process and how you can contribute. Please read these

guidelines before submitting your code as they are designed to save

you time later on when your code is under review.

## Contributing 101

All contributions are submitted via pull requests (PRs) open against

the osquery's [master](https://github.com/osquery/osquery/tree/master)

branch on GitHub. After being reviewed by the \_core team\_ and tested

by CI, if all is well, they will be pushed to master and the

corresponding PR closed.

You can see who the \_core team\_ is by viewing the [team

page](https://github.com/orgs/osquery/teams) on the osquery GitHub

organization.

If you need help, both the core team and community members are on the

osquery [Slack](https://osquery.slack.com). Feel free to register

using the following [link](https://slack.osquery.io/) if you haven't

done so yet and get in touch with us. The `#code-review` Slack

channel has been set up to handle urgent review needs as well as

questions about your PR. Note: prefer to keep discussion about code

changes in the GitHub pull request thread.

The osquery team also hosts regular office hours where the community

is invited to discuss osquery development with the core team. You are

welcome to join. Office hours are announced on our Slack on the

`#officehours` channel.

## Development Process Guidelines

For documentation on building, testing, and formatting code, please

review the ReadTheDocs article on [building

osquery](https://osquery.readthedocs.io/en/latest/development/building/).

This CONTRIBUTING guide focuses more on concepts and high level

workflow.

### Blueprints

If you plan to submit a change to the osquery core, a new big feature,

or in general a change that merits discussion, start by opening a

[Blueprint](https://github.com/osquery/osquery/issues/new?template=Blueprint.md)

issue.

A blueprint issue is a standard GitHub issue, tagged with the label

[#blueprint](https://github.com/osquery/osquery/labels/blueprint),

which describes your idea, the problem you are solving and how you

plan to implement your solution. The goal of the blueprint is to allow

both the core team and the community to discuss whether a certain

change is desirable and will be accepted, and identify possible

problems with the implementation before it even starts.

There aren't strict guidelines on when a blueprint is needed or not,

so you should use your best judgement or just ping the osquery team on

our `#core` channel on Slack. Here are some examples of changes which

\*\*would\*\* benefit from a blueprint:

\* Change the basic functioning of the query scheduler

\* Alter the thrift interfaces

\* Reimplement the logger interface

\* Add a new plugin type

There isn't either a strict format for the blueprints, but make sure

to include what problem you are trying to solve and how you plan to

solve it. We can go from that and ask more information if

necessary. If you have code already, even if it is only a

proof-of-concept that will be dropped later, please submit it as a PR

and associate it with the blueprint by mentioning the blueprint issue

on the pull request.

Please remember that blueprints are mostly designed to save \*\*you\*\*

time by preventing you from implementing code which won't be accepted

or will need to be extensively modified later on. Please use the right

[template](https://github.com/osquery/osquery/issues/new?template=Blueprint.md)

for the issue. Feel free to advertise your blueprint and ask for

feedback on Slack.

### Pull requests

\*\*Do not submit multiple unrelated changes on the same PR.\*\* A pull

request must represent a single body of work. If your work requires a

bug-fix, submit that first on a separate PR, the same goes for

refactors. If you can split your work into multiple smaller PRs please

also do so. This is of utmost importance to allow fast reviews and to

simplify regression tracking, reverts and references.

Start by developing your feature on a [feature

branch](https://guides.github.com/introduction/flow/), possibly

formatting your code before each commit, and when ready submit a pull

request against the osquery master branch. The initial PR should

preferably \*\*contain a single commit\*\*. If you are unfamiliar with

GitHub or how pull requests work, GitHub has a very easy to follow

guide that teaches you [how to fork the project and submit your first

PR] (https://guides.github.com/activities/forking/).

It is helpful if you tag the GitHub issues you are addressing on the

body of your PR description. If your PR is intended to close an issue

keywords (like `fixes` or `closes`) as defined on [GitHub

Help](https://help.github.com/articles/closing-issues-using-keywords/).

Once you submit your PR, a formatting check and continuous integration

tests will be triggered on the CI systems for the multiple platforms

we support. If all the required checks and tests are successful the

core team will review your PR. If the tests fail or the reviewer

requests changes, please submit those changes by \*\*appending new

commits\*\* to your feature branch. \*\*Avoid amending old commits\*\* as

that makes it harder for the reviewer to track your updates. If you

need to keep your PR up-to-date with master the preferred way is to

[rebase your

branch](https://help.github.com/en/articles/about-git-rebase) on

`master` and `git push` with the `--force` option. Finally, the core

team might help you with getting your PR accepted by pushing directly

to your branch when that makes sense.

Once both the core team and CI are happy with the PR (remember tests

need to pass for all of the supported platforms) the PR will be

squashed into a single commit and pushed to the master branch. Only

the core team can merge pull requests and therefore at least one core

team member will always review your PR, however reviews from the

community are highly encouraged and desirable.

Finally we try to keep only active PRs open. If your PR is stale we

will close it, however if you want to get back to it at a certain

point feel free to re-open, or comment on it.

### A note about labels

The core team uses labels to tag each and every pull request. If you

care about their meaning take a look at

[labels](https://github.com/osquery/osquery/labels) on

GitHub. However, only the core team can label issues and PRs, so you

don't need to care too much about this.

### Milestones and release versions

We currently do not use any strict versioning scheme and we cut new

versions as we feel it makes sense according to the new features

implemented, whether critical bug-fixes where merged, the size of the

release (i.e. how many commits since last version), etc. We will

however keep some near future milestones open and tag each PR with the

milestone we think it is going to be merged for.

Milestones are used for larger releases and we might cut patch

releases as we go. If your PR is tagged with the next milestone you

can expect it to be merged as soon as it is ready. If your PR is

tagged with a later milestone we'll only merge it after the previous

milestones are closed.

### Branches and tags

The osquery repo contains only the

[master](https://github.com/osquery/osquery/tree/master) branch which

we do our best to keep stable. We don't keep feature or release

branches. The master branch will always keep a linear history and no

merge commits are allowed. All our releases are tagged.

## Bug reports and feature requests

Developing code is not the only way to contribute to

osquery. Submitting bug reports and new ideas is also valuable and

appreciated.

We use GitHub issues to track bugs and feature requests. To submit a

bug report follow the [Bug

Report](https://github.com/osquery/osquery/issues/new?template=Bug\_Report.md)

template, to submit a feature request use the [Feature

Request](https://github.com/osquery/osquery/issues/new?template=Feature\_Request.md)

template.

\*\*Please only use issues for bug reports or feature requests\*\*. If you

have deployment questions or issues or a general question about

osquery use our Slack instead as you will have better support

there. For the fastest result, you should search the available

channels and choose the most appropriate one for your question. You

should post in the general channel as a last resort.

\*\*If you are using a vendor product please use the appropriate channel

as we won't be able to support vendor deployments on the non-vendor

channels.\*\*

## Guidelines for contributing features to osquery core

The software housed in this repo is known as osquery core. While there

are occasional exceptions, contributions to core should abide by the

following osquery guiding principles in order to be accepted:

1. osquery does not change the state of the system

2. osquery does not create network traffic to third parties

3. osquery binaries have a light memory footprint

4. osquery minimizes system overhead & maximizes performance

5. osquery does not 'shell out' to other binaries for data collection

6. The query schema for osquery seeks uniformity between operating systems

For new features that do not align with the mission principles of

core, you may build outside of osquery core in separate integrated

processes called extensions:

https://osquery.readthedocs.io/en/stable/development/osquery-sdk/.

### Does my contribution belong in Core or in an Extension?

Belongs in Core:

\* Observes guiding principles

\* Has been shared with and approved by osquery project maintainers

\* Meets osquery's testing and quality standards

Belongs in an extension:

\* Might not observe the osquery core guiding principles

\* Expands the scope of use for osquery beyond endpoint monitoring

\* Integrates with a proprietary or esoteric tool that is not widely applicable

## Contributor License Agreement

You must submit a Contributor License Agreement (CLA) before we can

accept any of your pull requests. You only need to submit one CLA for

any of osquery's open source projects.

This is managed through the Linux Foundations's EasyCLA. It will

comment appropriately on your PR.

## Technical Steering Committee

As defined by the [the osquery

charter](https://github.com/osquery/foundation/blob/master/CHARTER.md),

the Technical Steering Committee (or TSC for short) is responsible for

oversight of the osquery project.

The [GitHub

Team](https://github.com/orgs/osquery/teams/technical-steering-committee)

is the authoritative source, though we maintain the list here as well.

Current Members (in alphabetical order):

\* Alessandro -- [@alessandrogario](https://github.com/alessandrogario)

\* Nick -- [@muffins](https://github.com/muffins)

\* seph -- [@directionless](https://github.com/directionless)

\* Teddy -- [@theopolis](https://github.com/theopolis)

\* Victor -- [@groob](https://github.com/groob)

\* Zach -- [@zwass](https://github.com/zwass)

## License

By contributing to osquery you agree that your contributions will be licensed

in accordance with the terms specified in the [LICENSE](LICENSE) file.