# Contributing to Dynamo

Dynamo is an open source project.

It is the work of many contributors. We appreciate your help!

Please Note:

1. Before submitting the PR, please review this page for guidelines

2. Dynamo Team will meet 1x a month to review PRs found on Github (Issues will be handled separately)

3. PRs will be reviewed from oldest to newest

4. If a reviewed PR requires changes by the owner, the owner of the PR has 30 days to respond. If the PR has seen no activity by the next session, it will be either closed by the team or depending on its utility will be taken over by someone on the team

5. PRs should use either Dynamo's default PR template or [one of these other template options](https://github.com/DynamoDS/Dynamo/wiki/Choosing-a-Pull-Request-Template) in order to be considered for review.

6. PRs that do not have one of the Dynamo PR templates completely filled out with all declarations satisfied will not be reviewed by the Dynamo team.

7. PRs made to the `DynamoRevit` repo will need to be cherry-picked into all the DynamoRevit Release branches that Dynamo supports. Contributors will be responsible for cherry-picking their reviewed commits to the other branches after a `LGTM` label is added to the PR.

## Filing issues

When filing an issue, make sure to answer these five questions:

1. Which version of Dynamo are you using (check the About box)?

2. Which operating system are you using?

3. What did you do?

4. What did you expect to see?

5. What did you see instead?

General questions about using Dynamo should be submitted to [the forum at dynamobim.org](http://dynamobim.org/forums/forum/dyn/)

## Contributing code

Please see the [Pull request template guide](https://github.com/DynamoDS/Dynamo/wiki/Choosing-a-Pull-Request-Template)

before submitting a pull request.

Unless otherwise noted, the Dynamo source files are distributed under

the Apache 2.0 License.

Contribution "Bar"

------------------

The Dynamo team will merge changes that make it easier for customers to use Dynamo.

The Dynamo team will not merge changes that have narrowly-defined benefits. Contributions must also satisfy the other published guidelines defined in this document.

DOs and DON'Ts

--------------

Please do:

\* \*\*DO\*\* follow our [coding standards](https://github.com/DynamoDS/Dynamo/wiki/Coding-Standards) and [naming standards](https://github.com/DynamoDS/Dynamo/wiki/Naming-Standards)

\* \*\*DO\*\* include unit tests when adding new features. When fixing bugs, start with

adding a test that highlights how the current behavior is broken.

\* \*\*DO\*\* keep the discussions focused. When a new or related topic comes up

it's often better to create new issue than to side track the discussion.

\* \*\*DO\*\* blog and tweet (or whatever) about your contributions, frequently!

Please do not:

\* \*\*DON'T\*\* surprise us with big pull requests. Instead, file an issue and start

a discussion so we can agree on a direction before you invest a large amount

of time.

\* \*\*DON'T\*\* commit code that you didn't write. If you find code that you think is a good fit to add to Dynamo, file an issue and start a discussion before proceeding.

\* \*\*DON'T\*\* submit PRs that alter licensing related files or headers. If you believe there's a problem with them, file an issue and we'll be happy to discuss it.

\* \*\*DON'T\*\* add API additions without filing an issue and discussing with us first.

Managed Code Compatibility

--------------------------

Contributions must maintain API backwards compatibility following semantic versioning. Contributions that include breaking changes will be rejected. Please file an issue to discuss your idea or change if you believe that it may affect managed code compatibility.

Commit Messages

---------------

Please format commit messages as follows (based on [A Note About Git Commit Messages](http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html)):

```

Summarize change in 50 characters or less

Provide more detail after the first line. Leave one blank line below the

summary and wrap all lines at 72 characters or less.

If the change fixes an issue, leave another blank line after the final

paragraph and indicate which issue is fixed in the specific format

below.

Fix #42

```

Also do your best to factor commits appropriately, not too large with unrelated things in the same commit, and not too small with the same small change applied N times in N different commits.

Notes

---------------

This guide was based off of [DotNet Core Contributing Guide](https://github.com/dotnet/coreclr/blob/master/Documentation/project-docs/contributing.md)