# Contributing to Electron Packager

Electron Packager is a community-driven project. As such, we welcome and encourage all sorts of

contributions. They include, but are not limited to:

- Constructive feedback

- [Questions about usage](https://github.com/electron/electron-packager/blob/master/SUPPORT.md)

- [Bug reports / technical issues](#before-opening-bug-reportstechnical-issues)

- Documentation changes

- Feature requests

- [Pull requests](#filing-pull-requests)

We strongly suggest that before filing an issue, you search through the existing issues to see

if it has already been filed by someone else.

This project is a part of the Electron ecosystem. As such, all contributions to this project follow

[Electron's code of conduct](https://github.com/electron/electron/blob/master/CODE\_OF\_CONDUCT.md)

where appropriate.

## Before opening bug reports/technical issues

### Debugging

One way to troubleshoot potential problems is to set the `DEBUG` environment variable before

calling electron-packager. This will print debug information from the specified modules. The

value of the environment variable is a comma-separated list of modules which support this logging

feature. Known modules include:

\* `@electron/get:\*`

\* `electron-notarize\*`

\* `electron-osx-sign`

\* `electron-packager` (always use this one before filing an issue)

\* `get-package-info`

We use the [`debug`](https://www.npmjs.com/package/debug#usage) module for this functionality. It

has examples on how to set environment variables if you don't know how.

\*\*If you are using `npm run` to execute `electron-packager`, run the `electron-packager` command

without using `npm run` and make a note of the output, because `npm run` does not print out error

messages when a script errors.\*\*

## Contribution suggestions

We use the label [`help wanted`](https://github.com/electron/electron-packager/issues?q=is%3Aopen+is%3Aissue+label%3A%22help+wanted%22) in the issue tracker to denote fairly-well-scoped-out bugs or feature requests that the community can pick up and work on. If any of those labeled issues do not have enough information, please feel free to ask constructive questions. (This applies to any open issue.)

## Filing Pull Requests

Here are some things to keep in mind as you file pull requests to fix bugs, add new features, etc.:

\* CircleCI and AppVeyor are used to make sure that the project builds packages as expected on the

supported platforms, using supported Node.js versions.

\* Unless it's impractical, please write tests for your changes. This will help us so that we can

spot regressions much easier.

\* If your PR changes the behavior of an existing feature, or adds a new feature, please add/edit

the package's documentation. Files that will likely need to be updated include `README.md`,

`src/index.d.ts`, and `usage.txt`.

\* This project uses the [JavaScript Standard Style](https://standardjs.com/) as a coding convention.

CI will fail if the PR does not conform to this standard.

\* One of the philosophies of the project is to keep the code base as small as possible. If you are

adding a new feature, think about whether it is appropriate to go into a separate Node module,

and then be integrated into this project.

\* If you are contributing a nontrivial change, please add an entry to `NEWS.md`. The format is

similar to the one described at [Keep a Changelog](http://keepachangelog.com/).

\* Please \*\*do not\*\* bump the version number in your pull requests, the maintainers will do that.

Feel free to indicate whether the changes require a major, minor, or patch version bump, as

prescribed by the [semantic versioning specification](http://semver.org/).

\* Once your pull request is approved, please make sure your commits are rebased onto the latest

commit in the master branch, and that you limit/squash the number of commits created to a

"feature"-level. For instance:

bad:

```

commit 1: add foo option

commit 2: standardize code

commit 3: add test

commit 4: add docs

commit 5: add bar option

commit 6: add test + docs

```

good:

```

commit 1: add foo option

commit 2: add bar option

```

Squashing commits during discussion of the pull request is almost always unnecessary, and makes it

more difficult for both the submitters and reviewers to understand what changed in between comments.

However, rebasing is encouraged when practical, particularly when there's a merge conflict.

If you are continuing the work of another person's PR and need to rebase/squash, please retain the

attribution of the original author(s) and continue the work in subsequent commits.

### Running tests

To run the test suite on your local machine, you may first need to do a little setup.

If you're using macOS or a Debian/Ubuntu-derived distribution of Linux with x86\_64 architecture:

```sh

./test/ci/before\_install.sh

```

Then you can install dependencies and run the suite:

```sh

npm install

npm test

```

### Creating test fixtures

For some unit tests, a test fixture Electron project is required. Sometimes it's OK to use an

existing fixture, such as `basic`. If you need to add a new fixture:

1. Create a new subdirectory in `test/fixtures/`.

2. Add a `package.json` with only the minimal configuration necessary for your test(s).

3. If necessary, add supporting files, such as the JS file specified in the `main` key in the

`package.json` file.

4. Use `fixtureSubdir` from `test/util.js` to reference the fixture subdirectory in your test.

## For Collaborators

Make sure to get a `:thumbsup:`, `+1` or `LGTM` from another collaborator before merging a PR.

### Release process

- if you aren't sure if a release should happen, open an issue

- make sure that `NEWS.md` is up to date

- make sure the tests pass

- `npm version <major|minor|patch>`

- `git push && git push --tags` (or `git push` with `git config --global push.followTags true` on latest git)

- create a new GitHub release from the pushed tag with the contents of `NEWS.md` for that version

- close the milestone associated with the version if one is open

- `npm publish`