

Any PR that changes a package's version (which should be most PRs) should be published to pub.dev.

Automatic release

The packages in flutter/plugins and flutter/packages are automatically released with a Github Action workflow named ["release"](#). If a commit on master branch contains version updates to one or more packages, the "release" CI will publish the new versions to pub.dev and push the release tag to GitHub. The "release" CI passes if

1. the release process is successful, or
2. there are no version updates in the commit, or
3. the new versions have already been published.

If you are a Flutter team member and you want to know more about the publisher account, please see [b/191674407](#).

The "release" CI only runs on post-submit, and waits until all the other CI jobs have passed before starting. Like any other CI job, The "release" CI blocks future PRs if failed.

Note: the "release" CI does not automatically publish the `flutter_plugin_tools` package.

What if the "release" CI failed?

If it is a flake (for example, network issue), a Flutter team member can simply run the CI again. For more complicated cases, a Flutter team member can also manually release the packages, then re-run the CI to pass.

The most common source of failure of the `release` task is that another test failed; if that is due to flake, you will need to first re-run the failing test task, then once it's green re-run `release`.

Note: Loading a flutter/plugins `release` run's output in the GitHub UI currently [hangs the page](#), so to re-run a `release` task use [the Actions UI](#) rather than the commit's task page.

Manual release [Deprecated]

The Flutter team member most involved with the PR should be the person responsible for publishing. In cases where the PR is authored by a Flutter team member, the publisher should probably be the author. In other cases, the reviewing Flutter team member should publish.

Some things to keep in mind before publishing the release:

- Is the post-submit CI for that commit green? Even if CI shows as green on the PR it's still possible for it to fail on merge (e.g., due to a conflict with another recent change). Always check the post-submit CI before publishing.
- [Publishing is forever](#). Hopefully any bugs or breaking in changes in this PR have already been caught in PR review, but now's a second chance to revert before anything goes live.
- "Don't deploy on a Friday." Consider carefully whether or not it's worth immediately publishing an update before a stretch of time where you're going to be unavailable. There may be bugs with the release or questions about it from people that immediately adopt it, and uncovering and resolving those support issues will take more time if you're unavailable.

To release a package:

1. `git checkout <commit_hash_to_publish>`. This should be the commit of the PR you are publishing unless there's a very specific reason you are using a different version.
2. Ensure that `git status` is clean, and that there are no extra files in your local repository (e.g., via `git clean -xkd`).

3. Use the `publish-plugin` command from `flutter_plugin_tools` . This command checks that you've done the step above, publishes the new version to pub.dev, and tags the commit in the format of `<package_name>-v<package_version>` then pushes it to the upstream repository.

Fully manual backup option

If for some reason you can't use `flutter_plugin_tools` in step 3, you can publish manually:

1. Push the package update to pub.dev using `dart pub publish` .
2. Tag the commit with `git tag` in the format of `<package_name>-v<package_version>`
3. Push the tag to the upstream master branch with `git push upstream <tagname>` .