Flutter's channels

Flutter has the following channels, in increasing order of stability.

This page used to give examples of version numbers for each channel. Our current policy for versioning isn't well established. We intend to redefine our policies on this front in 2022.

master

The current tip-of-tree, absolute latest cutting edge build. Usually functional, though sometimes we accidentally break things.

beta

We will branch from master for a new beta release at the beginning of the month, usually the first Monday. This will include a branch for Dart, the Engine and the Framework. This branch will then be "stabilized" for the next couple of weeks, meaning we will accept cherrypick requests for high impact issues. As we near the end of the month and the next beta branch, we will likely reduce the number of cherrypicks we are willing to do. Once a quarter, the beta branch will live on to become the next stable branch, as detailed below.

stable

Roughly once a quarter, a branch that has been stabilized on beta will become our next stable branch and we will create a stable release from that branch. We recommend that you use this channel for all production app releases.
In case of high severity, high impact or security issues, we may do a hotfix release for the stable channel. This will follow the same cherrypick process.

How to change channels

You can see which channel you're on with the following command:

```
$ flutter channel
Flutter channels:
* stable
  beta
  master
```

To switch channels, run `flutter channel [<channel-name>]`, and then run `flutter upgrade` to ensure you're on the latest.

Which channel should I use?

We recommend using the `stable` branch. This version of Flutter is the one documented by our API documentation. The team's own Flutter plugins are always tested against the latest `stable` branch.

If you want to be using the latest and greatest, the `beta` branch is the right choice. That's the most recent version of Flutter that we have heavily tested. On average it takes about two weeks for a fix to end up in the beta branch after they land in our repository. The beta branch has passed all our public testing as well as having been verified against contributed private test suites (currently only Google is contributing such testing, but if you are interested, please reach out to Hixie on [[Chat]]).

If you want to be developing Flutter, then the `master` branch is the right choice. However, this branch may be broken since we do not run the entirety of our testing before allowing patches to land on this branch.

There used to be a `dev` branch as well (and you may still see it in our tooling). We no longer update this branch and our tooling will be updated in due course to no longer list it (if it hasn't already).

Will a particular bug fix be provided in a hotfix release?

Depending on the severity of the issue, it is possible. Refer to the cherrypick process for details.

If you really need a particular patch and it's a fix to the flutter/flutter repository, you should feel free to create a Flutter branch yourself on your development machine and cherry-pick the fix you want onto that branch. Flutter is distributed as a `git` repository and all of `git`'s tools are available to you. If you need a particular patch that's from the flutter/engine repository or one of our dependencies (e.g. Dart or Skia), you could build your own engine but it's probably easier to just wait until the next release. On average, the next `beta` release is about two weeks away.

See also

- [[Release process]], which describes the details for how we push builds from channel to channel.
- Cherrypick process, where we cover how to request an issue for cherrypicking.
- Release notes, where we document changes to each version of the stable channel.