

## Context

Production builds of both the Flutter Framework and the Engine repositories are created on LUCI. The recipes (actual build scripts) live at [flutter.googlesource.com/recipes](https://flutter.googlesource.com/recipes), while the builder config is maintained in <https://github.com/flutter/infra>. At this time, there is no version pinning between either of these repositories and the source trees they build (the Flutter framework & engine). This leads to frequent CI failures when preparing beta and stable releases because of changes in the recipes since the release was originally built on master.

**Please note - dev channel has been retired. Refer to this [blog](#) for more information.**

**Deprecated** - *When a dev release is promoted to beta, copies will be made of the engine and framework recipes (and additional dependencies) from the time the release was branched off of master will be made, namespaced by the name of the targeted stable version (e.g. `engine.py` -> `engine_v1_17_0.py`).*

The LUCI builder config now has separate builders for stable and beta channels, with stable and beta having their own recipes. With each new beta release, builder config should be updated to reflect the new version to look up the recipe by and which refs to trigger builds by.

When promoting a beta release to stable, we can delete any recipe copies older than the new stable. We then update constants in the builder config relating to stable version.

## Prerequisites

You must have read/write access to and local clones of the following repositories:

- [Flutter Framework](#)
- [Flutter Engine](#)
- [Flutter Infra](#)
- [Flutter Recipes](#)
- You should have Chromium's depot\_tools [installed](#) (necessary for making LUCI updates)

You should fetch upstream of all of these before proceeding.

## Beta Release Procedure

1. Find the Flutter framework **master commit** that your release has branched off of, and set it as

```
$FRAMEWORK_REVISION :
```

```
cd $FRAMEWORK_REPO
FRAMEWORK_REVISION=$(git merge-base master $RELEASE_BRANCH_NAME)
```

2. Identify the name of the stable version that this is a release candidate for, normalize dots with underscores, set as `$VERSION` :

```
# for 1.18.0-12.0.pre
VERSION='1_18_0'
```

3. Get the date/time this commit landed in the tree (since we use GitHub squash and merge, this will presumably be the commit date). The following command will retrieve the commit date of a given revision:

```
cd $FRAMEWORK_REPO
FRAMEWORK_DATE=$(git show --no-patch --format=%ci $FRAMEWORK_REVISION)
```

4. Get the last LUCI recipe commit before the framework date:

```
cd $RECIPES_REPO
RECIPE_FRAMEWORK_REVISION=$(git log --before="$FRAMEWORK_DATE" -n 1 --format=%H)
```

5. Copy the framework recipe at the time of `$FRAMEWORK_DATE` as `flutter_$VERSION.py` (for the engine, `engine_$VERSION.py`):

```
cd $RECIPES_REPO
./branch_recipes.py --flutter-version="$VERSION" --recipe-
revision="$RECIPE_FRAMEWORK_REVISION"
```

6. Update tests:

```
cd $RECIPES_REPO
./recipes.py test train
```

7. Commit the two new recipes and all updated test expectations to git. Create a new CL with `git cl upload` and get a reviewer from `build/scripts/slave/recipes/flutter/OWNERS`. Upon approval, merge the CL.

8. In flutter/infra, update BRANCHES dictionary in [main.star](#):

- `BRANCHES['stable']['testing-ref']` , a regex to the branch name of the current stable
- `BRANCHES['stable']['version']` , the version element of the recipe filename, e.g. `v1_17_0`
- `BRANCHES['beta']['testing-ref']` , a regex to the branch name of the current beta candidate
- `BRANCHES['beta']['version']`

9. Execute the main.star file to generate the rest of the config files (and validate your changes for mistakes): `$ ./main.star`

10. Commit your changes, push to github and get it reviewed. This PR should be landed after any LUCI recipe changes.

11. After your PR has landed, wait for it to be mirrored to [the chromium tree](#). LUCI post-submit builds should now work for your candidate branch.

## Stable Release Procedure

Updating recipes for a stable release is much simpler than that for a beta release, as the requisite recipe should have already been forked when the release was promoted to beta.

1. In flutter/infra, update BRANCHES dictionary in [main.star](#):

- `BRANCHES['stable']['testing-ref']` , a regex to the branch name of the current stable development branch
- `BRANCHES['stable']['version']` , the version element of the recipe filename, e.g. `1_20_0`

2. Execute the main.star file to generate the rest of the config files (and validate your changes for mistakes): `$ ./main.star`

3. Commit your changes, push to github and get it reviewed. Merge it. Note, this updated configuration won't take effect until it has propagated to LUCI infra. The current version of the config can be seen [here](#).
4. In the recipes repo, any recipe forks older than the current stable can be safely deleted. `git rm /path/to/recipe` will both delete the file and stage the change with git.
5. Update tests:

```
cd $RECIPES_REPO
./recipes.py test train
```

6. Commit the file deletions and updated test expectations to git. Create a new CL with `git cl upload` and get a reviewer from `build/scripts/slave/recipes/flutter/OWNERS` . Upon approval, merge the CL.