How to backport a pull request to a release line

Staging branches

Each release line has a staging branch that the releaser will use as a scratch pad while preparing a release. The branch name is formatted as follows: `vN.x-staging` where `N` is the major release number.

For the active staging branches see the Release Schedule.

What needs to be backported?

If a cherry-pick from master does not land cleanly on a staging branch, the releaser will mark the pull request with a particular label for that release line (e.g. `backport-requested-vN.x`), specifying to our tooling that this pull request should not be included. The releaser will then add a comment requesting that a backport pull request be made.

What can be backported?

The "Current" release line is much more lenient than the LTS release lines in what can be landed. Our LTS release lines (see the Release Plan) require that commits mature in the Current release for at least 2 weeks before they can be landed in an LTS staging branch. Only after "maturation" will those commits be cherry-picked or backported.

How to label backport issues and PRs?

For the following labels, the `N` in `vN.x` refers to the major release number.

Label	Description
backport-blocked-vN.x	PRs that should land on the vN.x-staging branch but are blocked by another PR's pending backport.
backport-open-vN.x	Indicate that the PR has an open backport.
backport-requested-vN.x	PRs awaiting manual backport to the vN.x-staging branch.
backported-to-vN.x	PRs backported to the vN.x-staging branch.
baking-for-lts	PRs that need to wait before landing in a LTS release.
lts-watch-vN.x	PRs that may need to be released in vN.x.
vN.x	Issues that can be reproduced on vN.x or PRs targeting the vN.x-staging branch.

How to submit a backport pull request

For the following steps, let's assume that a backport is needed for the v10.x release line. All commands will use the `v10.x-staging` branch as the target branch. In order to submit a backport pull request to another branch, simply replace that with the staging branch for the targeted release line.

1. Checkout the staging branch for the targeted release line.
2. Make sure that the local staging branch is up to date with the remote.
3. Create a new branch off of the staging branch, as shown below.

```
# Assuming your fork of Node.js is checked out in $NODE_DIR,
# the origin remote points to your fork, and the upstream remote points
# to git://github.com/nodejs/node
cd $NODE_DIR
# If v10.x-staging is checked out `pull` should be used instead of `fetch`
git fetch upstream v10.x-staging:v10.x-staging -f
# Assume we want to backport PR #10157
git checkout -b backport-10157-to-v10.x v10.x-staging
# Ensure there are no test artifacts from previous builds
# Note that this command deletes all files and directories
# not under revision control below the ./test directory.
# It is optional and should be used with caution.
git clean -xfd ./test/
```

4. After creating the branch, apply the changes to the branch. The cherry-pick will likely fail due to conflicts. In that case, you will see something like this:

```
# Say the $SHA is 773cdc31ef
$ git cherry-pick $SHA # Use your commit hash
error: could not apply 773cdc3... <commit title>
hint: after resolving the conflicts, mark the corrected paths
hint: with 'git add <paths>' or 'git rm <paths>'
hint: and commit the result with 'git commit'
```

5. Make the required changes to remove the conflicts, add the files to the index using `git add`, and then commit the changes. That can be done with `git cherry-pick --continue`.
6. Leave the commit message as is. If you think it should be modified, comment in the pull request. The `Backport-PR-URL` metadata does need to be added to the commit, but this will be done later.
7. Make sure `make -j4 test` passes.
8. Push the changes to your fork.
9. Open a pull request:

1. Be sure to target the `v10.x-staging` branch in the pull request.
2. Include the backport target in the pull request title in the following format: `[v10.x backport] <commit title>`. Example: `[v10.x backport] process: improve performance of nextTick`
3. Check the checkbox labeled "Allow edits and access to secrets by maintainers".
4. In the description add a reference to the original pull request.
5. Amend the commit message and include a `Backport-PR-URL:` metadata and re-push the change to your fork.
6. Run a `node-test-pull-request` CI job (with `REBASE_ONTO` set to the default `<pr base branch>`)

10. If during the review process conflicts arise, use the following to rebase: `git pull --rebase upstream v10.x-staging`

After the pull request lands, replace the `backport-requested-v10.x` label on the original pull request with `backported-to-v10.x`.