

Developer Workflow

- [Github](#)
 - [Sending Pull Requests to Nordix Repos](#)
 - [Sending Pull Requests to Upstream Repos](#)
- [Gerrit](#)
 - [When automatic rebase fails](#)

Github

The forks created from existing Github projects into Nordix namespace uses nordix-dev as default/development branch.

master branch is locked in order to use it as a reference to upstream repo in order to make it easier to compare changes against and rebase as needed.

Please ensure you always for against nordix-dev branch, create your local branches from there, and frequently rebase against master branch.

Sending Pull Requests to Nordix Repos

- Get the project you want to contribute to forked to [Nordix namespace](#) on Github.
 - This can be done by sending an email to discuss@lists.nordix.org.
 - Please include the Github link to the project you want the fork created for.
- Login to Github using your Github username and password.
- Navigate to [Nordix namespace](#) and click the name of the project fork you will be contributing to.
- Click "[Clone or Download](#)" button on the opening page and grab the URI to clone the repo.
- Clone the repo to your computer.
 - `git clone <URI>`
- Create a local branch to do your work on.
 - `git checkout -b <branch_name>`
- Do your work, commit.
 - `git add -A`
 - `git commit -m "Commit message"`
- Push your branch to origin
 - `git push origin <branch_name>`
- Go to your Web browser and navigate to the project you did your work
 - click Pull Requests tab
 - click New Pull Request button
 - Change base fork to Nordix/<forked project>
 - Change base branch to nordix-base
 - Change compare branch to the branch you pushed
 - Click Create New Pull Request
- Merge Pull Request if you are done with your work.

Sending Pull Requests to Upstream Repos

- Navigate to the project which the fork was created from
- Click New Pull Request
- Click Compare Across Forks
- Change Base Fork/branch to the project/branch you want to get your changes merged into
- Change Head Fork/branch to the Nordix fork you merged your pull requests into and nordix-dev branch
- Add description for your Pull Request
- Click Create Pull Request

You can experiment with this workflow using [Nordix/sandbox](#) repo which is forked from a personal repo [fdegir/sandbox](#) for experimentation.

If you find ways to improve the workflow, please do not hesitate to update this page.

Gerrit

If you are working based on Apache licensed upstream project, the work you are doing first needs to land in to Nordix Infrastructure.

In order to achieve this, you should have got the upstream repo imported to Nordix Gerrit so you can do your work in a repo cloned from Nordix. The repos are automatically kept in synch.

When automatic rebase fails

When you run into the situation that you cannot commit in ONAP because a rebase is needed, but the "Rebase" button in Gerrit doesn't work, then rebase your change manually in your local repository.

Note: Cherry-picking a commit on top of a new base is the same as rebasing a commit on top of a new base. It's just a matter of which commit you have checked out before running the command. Cherry-picking is more flexible though and can be used for more than just rebasing.

1. Fetch the latest changes for all branches from Nordix repo to local repo.
git fetch
2. Make sure that latest commit on Nordix master is the same as the latest one on ONAP master. If not, send a mail to the "discuss" mailing list that there might be something wrong with the synch job.
3. Create a temporary branch
git checkout -B temp_rebase_branch origin/master
4. Get the change from Gerrit by using the **cherry-pick** link on the Gerrit review page in Nordix.
git fetch <https://gerrit.onap.org/r/dcaegen2/collectors/datafile> refs/changes/69/67269/2 && git cherry-pick FETCH_HEAD
5. Resolve the rebase conflicts
git mergetool
git cherry-pick --continue
6. Resubmit to master in Nordix
git push usptream HEAD:refs/for/master
7. Move to another branch so the temporary rebase branch can be deleted (git branch shows you the branches you have which you can move to)
git checkout <some other branch>
8. Remove the temporary local delivery branch
git branch -D temp_rebase_branch