

# Merging a subdirectory from another repo via git-subtree

Greg Wilson [wanted to know](#) how to move a directory from one git repo to another while preserving history. Ash Wilson posted a [gist](#) demonstrating a merge and delete strategy. A similar result can be achieved using git-subtree. The advantage of the git-subtree approach is that git-subtree creates a new branch containing only the commits pertaining to files located in a specified subdirectory. Like Ash's approach, git-subtree will result in the final state of the target repo is such that changes to the target repo won't get pushed back upstream to the source.

If you want to carry *all* of the commit history from another repo into the target repo, Ash's approach is probably the way to go. If you want only the commits pertaining to files within a subdirectory of the source repo, git-subtree is the way to go.

The workflow with git-subtree is pretty simple: you specify a subdirectory in a source repo which you want to break out, then you merge those commits into a subdirectory in a target repo.

I am modifying my example slightly from Ash's because it looks like the /openstack-swift/ directory has been removed from jclouds/jclouds-labs-openstack as of 2014-10-12. I will use openstack-glance instead. Thus, we're moving the /openstack-glance/ directory from jclouds/jclouds-labs-openstack to /apis/openstack-glance/ in jclouds/jclouds. The two approaches begin in the same way.

```
# Clone the target repo
git clone git@github.com:jclouds/jclouds.git
cd jclouds

# Add the source repository as a remote, and perform the initial fetch.
git remote add -f sourcerepo git@github.com:jclouds/jclouds-labs-openstack.git

# Create a branch based on the source repositories' branch that contains the state you want to copy.
git checkout -b staging-branch sourcerepo/master

# Here's where the two approaches diverge.
# Create a synthetic branch using the commits in `/openstack-glance/` from `sourcerepo`
git subtree split -P openstack-glance -b openstack-glance

# Checkout `master` and add the new `openstack-glance` branch into `/apis/openstack-glance/`. At this
point, the desired result will have been achieved.
git checkout master
git subtree add -P apis/openstack-glance openstack-glance

# Clean up by removing the commits from the `openstack-glance` branch and the `sourcerepo` remote.
git branch -D openstack-glance staging-branch
git remote rm sourcerepo
```

And that's it. No additional (potentially superfluous) commits were brought into the jclouds repo from the jclouds-labs-openstack repo.