

Mundane git commit-tree tricks, Part 7: Combining more than two files into one while preserving line history, manual octopus merging

 devblogs.microsoft.com/oldnewthing/20190515-00

May 15, 2019



Raymond Chen

Last time, we saw how to combine two files to form a third file, while preserving line history. But what if you need to combine more than two files? For example, maybe you want to take a whole bunch of `csv` files and merge them into one big file, but still track the origin of each line.

Let's set up a scratch repo to try it out.

```
git init

>fruits echo apple
>>fruits echo grape
>>fruits echo orange
git add fruits
git commit --author="Alice <alice>" -m "create fruits"

>veggies echo celery
>>veggies echo lettuce
>>veggies echo peas
git add veggies
git commit --author="Bob <bob>" -m "create veggies"

>dairy echo cheese
>>dairy echo eggs
>>dairy echo milk
git add dairy
git commit --author="Carol <carol>" -m "create dairy"

git tag ready
```

We can use the trick from last time to merge two files, and extend it to three files by performing an octopus merge.

```
git checkout -b d2f
git mv dairy food
git commit -m "dairy to food"
git checkout -

git checkout -b f2f
git mv fruits food
git commit -m "fruits to food"
git checkout -

git checkout -b v2f
git mv veggies food
git commit -m "veggies to food"
git checkout -

git merge d2f f2f v2f
```

Except that it doesn't work. In fact, it explodes quite spectacularly.

```
Fast-forwarding to: d2f
Trying simple merge with f2f
Simple merge did not work, trying automatic merge.
Added food in both, but differently.
fatal: unable to read blob object e69de29bb2d1d6434b8b29ae775ad8c2e48c5391
error: Could not stat : No such file or directory
ERROR: content conflict in food
fatal: merge program failed
Automated merge did not work.
Should not be doing an octopus.
Merge with strategy octopus failed.
```

I like that second-to-last line that scolds you for attempting this sort of thing in the first place.

Let's clean up the work that merge had left in progress. You normally would do this with a `git merge --abort`, but octopus merges are not abortable because they don't record enough information to permit an abort. (This is arguably a bug in git, but it's merely an annoyance, and not something normal people are going to encounter.)

```
git reset --hard
```

The problem is that octopus merges work only if there are no conflicts. We're going to have to build our own octopus merge.

```
cat dairy fruits veggies | sort >food
git rm dairy fruits veggies
git add food
git write-tree
```

The `git write-tree` creates a tree from the index. It's the tree that a `git commit` would create, but we don't want to do a normal commit. This is the tree we want to commit, but we need to set custom parents, so we'll ask `git write-tree` for the tree that *would be committed*, so we can build our custom commit.

```
git commit-tree <tree-hash> -p HEAD -p d2f -p f2f -p v2f -m "combine dairy, fruits, and veggies"
```

The `commit-tree` will print another hash. This is the hash of the manually-constructed octopus merge.

```
git merge --ff-only <commit-hash>
```

I like to use `--ff-only` to make sure that I really am just moving forward.

```
git blame food
```

```
^7c5ae53 fruits (Alice 2019-05-15 07:00:00 -0700 1) apple
03c4572c veggies (Bob 2019-05-15 07:00:01 -0700 2) celery
65430aff dairy (Carol 2019-05-15 07:00:02 -0700 3) cheese
65430aff dairy (Carol 2019-05-15 07:00:02 -0700 4) eggs
^7c5ae53 fruits (Alice 2019-05-15 07:00:00 -0700 5) grape
03c4572c veggies (Bob 2019-05-15 07:00:01 -0700 6) lettuce
65430aff dairy (Carol 2019-05-15 07:00:02 -0700 7) milk
^7c5ae53 fruits (Alice 2019-05-15 07:00:00 -0700 8) orange
03c4572c veggies (Bob 2019-05-15 07:00:01 -0700 9) peas
```

There are other ways we could have produced the same result. For example, we could have performed a series of two-files-into-one merges, but this way gives us a single commit on the trunk that captures the “combine multiple files into one”.

[Raymond Chen](#)

Follow

