

Mundane git commit-tree tricks, Part 1: Building a commit manually out of a tree

 devblogs.microsoft.com/oldnewthing/20190506-00

May 6, 2019



Raymond Chen

When it comes time to make a public update to the [UWP samples repo](#), I take a snapshot of what's in our internal staging repo and push it to the public repo. All of the intermediate steps are squashed out, so that the public repo isn't cluttered with noisy history.

I accomplish this with a local repo with two remotes. One remote is the public repo and the other repo is our private repo. When it's time to make a public release, I go through these steps.

First: Freshen my local master with the latest public. This technically isn't necessary because nobody should be pushing to public except me, so my local master should always be up to date. But hey, it doesn't hurt to check, because occasionally somebody else will push something to master.

```
git checkout master
git pull --ff-only public
```

Next, I update my private remote so it has a copy of the source code that we want to make public.

```
git fetch private
```

Now the fun part: Commit the latest private tree onto the latest public tree.

```
git commit-tree private/master^{tree} -p master -m "comment"
(this prints a hash)
```

Note: If using the Windows `CMD` command prompt, you need to type

```
git commit-tree private/master^^{tree} -p master -m "comment"
```

The doubling of the `^` character is necessary because `^` is the escape character for `CMD.EXE`. The point is that you want the first parameter to `commit-tree` to be `private/master^{tree}` after the command line has had its say of escaping.

(It's too bad that `commit-tree` requires a tree. Would be nice if they could relax this requirement so it can accept any tree-ish.)

I can then fast-forward to the newly-created commit and bingo, it's as if I did a super-squash of the private repo onto the public repo.

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

In reality, my workflow is a little funkier than this, and I have to update multiple branches and make each one merge into the next. We'll look some more at the workflow next time.

Raymond Chen

Follow

