# Mundane git commit-tree tricks, Part 2: Building a merge commit manually out of a tree

devblogs.microsoft.com/oldnewthing/20190507-00

May 7, 2019

Raymond Chen

When it comes time to make a public update to the UWP samples repo, I usually have to update multiple branches. Each branch encompasses the changes from the previous-version branch, and then adds new changes specific to that branch.

For the oldest branch, I create a plain commit from a tree and a parent commit, like we saw last time.

```
git checkout --detach

git commit-tree public/win10-1507^{tree} -p win10-1507 -m "comment"
(this prints a hash)

git fetch . ⟨hash⟩:win10-1507
```

First I detach the head from whatever branch it happens to be on. This allows me to update branches without getting any complaints of the form "You can't do that to the current branch." Since the head is detached, there is no current branch. Detaching the head also means that any branch updates do not alter the working directory.

The second command is the one we saw last time. It creates a commit from a tree and a parent. (If your command shell is `CMD.EXE`, don't forget to double the `^` character because `^` is `CMD.EXE`'s escape character.)

The third command updates the win10-1507 branch to the specified hash. This is equivalent to

```
git branch -f win10-1507 ⟨hash⟩
```

except that it verifies that the new branch head is a descendant of the current branch head. (In other words, it's like a `--ff-only`.) I use this alternate version as a safety check.

For the second oldest branch win10-1511, I want the commit to be a merge of win10-1507 and the changes specific to that branch. To do this, I use the `commit-tree` command, but provide multiple parents. The first parent is the previous commit for the branch, and the second parent is the incoming changes from its ancestor branch.

```
git commit-tree private/win10-1511^{tree} -p win10-1511 -p win10-1507 -m "comment"
(this prints a hash)

git fetch . ⟨hash⟩:win10-1511
```

The next branches follow the same pattern as win10-1511.

```
git commit-tree private/win10-1607^{tree} -p win10-1607 -p win10-1511 -m "comment"
(this prints a hash)

git fetch . ⟨hash⟩:win10-1607

git commit-tree private/master^{tree} -p master -p win10-1607 -m "comment"
(this prints a hash)

git fetch . ⟨hash⟩:master

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

git fetch . ⟨hash⟩:dev
```

At this point the branches are ready to be pushed to the public repo.

Note that at no point did we update the working directory. All we are doing is creating commits from existing trees and updating branches. These are fast operations since they manipulate relatively little data, and no files on disk need to be updated.

Raymond Chen

**Follow**