Stop cherry-picking, start merging, Part 8: How to merge a partial cherry-pick

devblogs.microsoft.com/oldnewthing/20180321-00

March 21, 2018

Raymond Chen

Continuing our exploration of using merges as a replacement for cherry-picking, here's another scenario you can now solve:

What if I want to take only part of a commit into another branch?

Well, if you haven't committed the change yet, then you can follow the usual workflow: Create a patch branch, commit only the part that you want to go into both branches, and then merge that patch branch into the master and feature branches. Once that's done, you can make additional commits in the feature branch for the parts of the change you don't want to go into master immediately.

What if I already committed a change to my feature branch, and I want to take only part of it to the master branch?

You can follow the retroactive merge pattern described earlier under *What if I already made the fix in my feature branch by committing directly to it, rather than creating a patch branch?* Put into the patch branch the piece of the commit that you want to share with the master branch.

	apple apricot		berry apricot	
	M1		M2	master
apple apricot		berry apricot		
A		P		patch
	F1	F1a	F2	feature

apple apricot	berry banana	berry banana

From a starting commit A where the lines are `apple` and `apricot`, we create a feature branch. On the master and feature branches, we make unrelated commits M1 and F1, respectively, that don't change either of the two lines. We then make a commit F1a on the feature branch that changes both lines to `berry` and `banana`. We want to propagate the `berry` part to the master branch, but not the `banana` part.

To do this, we create a patch branch starting at the common commit A. On the patch branch, we create a commit P that changes the first line from `apple` to `berry`, but leaves the second line unchanged; it remains `apricot`. We merge this patch branch into the master branch as M2, resulting in `berry` and `apricot` in the master branch. We also merge this patch branch into the feature branch as F2, resulting in no change in the feature branch because the first line is already `berry`; the lines in the feature branch are still `berry` and `banana`.

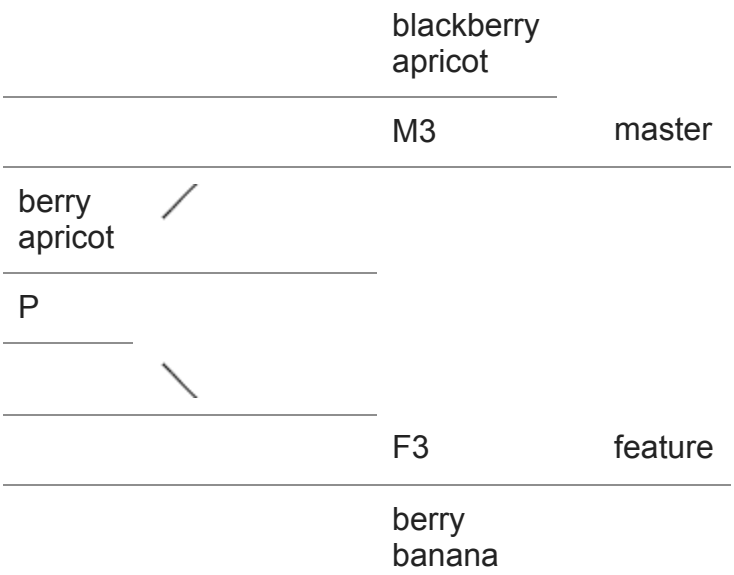
When this merges, the merge base will be `berry apricot`, which is identical to what's in the master branch, which means that the change from the feature branch will be taken, resulting in `berry banana`.

But let's not merge yet. Suppose that the master branch makes a commit M3 which changes `berry` to `blackberry` but leaves `apricot` unchanged.

	apple apricot		berry apricot	blackberry apricot	
	M1		M2	M3	master
apple apricot		berry apricot			
A		P	patch		

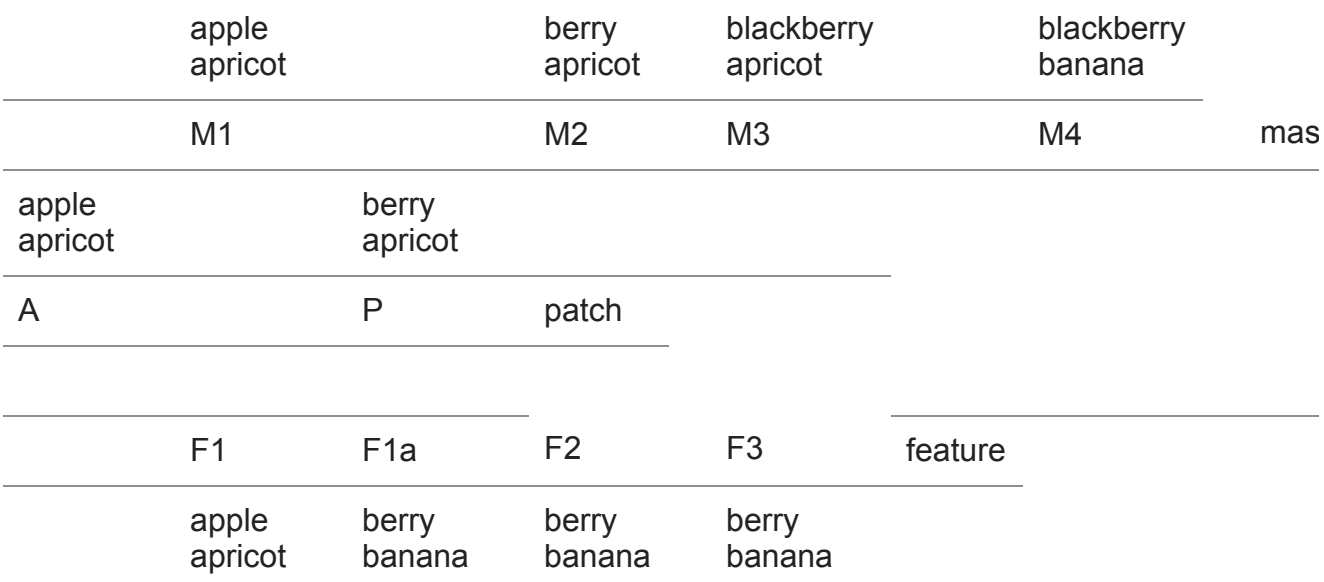
	F1	F1a	F2	F3	feature
	apple apricot	berry banana	berry banana	berry banana	

What happens when we merge? Let's look at the three-way merge:



The three-way merge chooses commit P as the merge base, and in that commit, the lines are `berry` and `apricot`. In the master branch, the lines are `blackberry` and `apricot`: The net change is that the first line changed from `berry` to `blackberry`. In the feature branch, the lines are `berry` and `banana`: The net change is that the second line changed from `apricot` to `banana`.

Therefore, the merge of the two branches is to accept the change of the first line from the master branch and the change of the second line from the feature branch, resulting in `blackberry` and `banana`, as desired.



Raymond Chen

Follow