**解决Git Revert操作后再次Merge代码被冲掉的问题：**

2017年08月22日 11:21:25 [大林-Java](https://me.csdn.net/paul_wei2008) 阅读数：4787 标签： [git](http://so.csdn.net/so/search/s.do?q=git&t=blog)[合并](http://so.csdn.net/so/search/s.do?q=%E5%90%88%E5%B9%B6&t=blog) 更多 个人分类： [Git](https://blog.csdn.net/paul_wei2008/article/category/2720709)

git reset & git revert 区别：

1. git revert是用一次新的commit来回滚之前的commit，git reset是直接删除指定的commit。   
2. 在回滚这一操作上看，效果差不多。但是在日后继续merge以前的老版本时有区别。因为git revert是用一次逆向的commit“中和”之前的提交，因此日后合并老的branch时，导致这部分改变不会再次出现，但是git reset是之间把某些commit在某个branch上删除，因而和老的branch再次merge时，这些被回滚的commit应该还会被引入。   
3. git reset 是把HEAD向后移动了一下，而git revert是HEAD继续前进，只是新的commit的内容和要revert的内容正好相反，能够抵消要被revert的内容。

git revert fcd8a10 -m 1 #撤回提交的commit id:fcd8a10 1代表当前分支  2代表在当前分支 merge的分支

[解决Git Revert操作后再次Merge代码被冲掉的问题](http://blog.csdn.net/cxn945/article/details/48372327)：

要解决这个问题，需要把revert产生的提交再revert一次

因为git revert是用新提交覆盖旧提交，因此，被覆盖的提交等于不会被采用了。如果两个分支（假设是master和A分支）先合并再用revert回滚，之后又合并（A合并到master），就会发现在master分支上，A分支第一次合并之前的修改大部分不见了。这是因为从时间的发生顺序来看，A分支第一次合并之前的修改发生在revert之前，revert发生在后，而 revert抛弃了A第一合并之前的修改，那么再合并Git就认为你永远抛弃了A第一次之前的修改。

1） git merge topic（当前head指向master分支）

2） git reset --hard HEAD~ 如下图(如果其它人M有改动就会丢失)

3)git revert -m 1 HEAD（反向提交）

新的提交 ^M 与 C6 有完全一样的内容，所以从这儿开始就像合并从未发生过除了“现在还没合并”的提交依然在 HEAD 的历史中。 如果你尝试再次合并 topic 到 master Git 会感到困惑：

1. $ git merge topic
2. Already up-to-date.

topic 中并没有东西不能从 master 中追踪到达。 更糟的是，如果你在 topic 中增加工作然后再次合并，Git 只会引入被还原的合并 之后 的修改。

Figure 141. 含有坏掉合并的历史

解决这个最好的方式是撤消还原原始的合并，因为现在你想要引入被还原出去的修改，然后 创建一个新的合并提交：

1. $ git revert ^M
2. [master 09f0126] Revert "Revert "Merge branch 'topic'""
3. $ git merge topic