**Git 撤销 merge**

转载 2015年10月20日 17:53:28

* 标签：
* [git](http://so.csdn.net/so/search/s.do?q=git&t=blog)

在使用Git开发过程中偶尔会遇到合并（merge）错代码的情形。此时需要撤销已经合并的分支（branch）。

虽然对git有了一定的了解和使用，但是这种撤销合并分支的情况还是不太清楚改如何处理，这里有一个比较好的资料。

[Git 撤销合并](http://blog.psjay.com/posts/git-revert-merge-commit/)

这里通过本人亲身试验对连接中文章的revert a merge commit 部分

做一定程度的解释：

先原因文章内容：

**Merge Commit**

在描述 merge commit 之前，先来简短地描述一下常规的 commit。每当你做了一批操作（增加、修改、或删除）之后，你执行 git commit 便会得到一个常规的 Commit。执行 git show <commit> 将会输出详细的增删情况。

*Merge commit 则不是这样。每当你使用 git merge 合并两个分支，你将会得到一个新的 merge commit。执行 git show <commit> 之后，会有类似的输出：*

commit 83281a8e9aa1ede58d51a6dd78d5414dd9bc8548 //本人实际git信息，这里对应git演进图中的 g

Merge: 312a518 fa87415 //312a518和fa87415 可以在git log中找到对应的提交信息（只是commit一长串字符的头部分）

Author:XXX

Date:   Wed May 28 10:02:10 2014 +0800

    Merge branch 'dev' into master

    Merge branch 'dev' into master

    Conflicts:

        gitRevert.rtf

*其中，Merge 这一行代表的是这个合并 parents，它可以用来表明 merge 操作的线索。*

*举个例子，通常，我们的稳定代码都在 master 分支，而开发过程使用 dev 分支，当开发完成后，再把 dev 分支 merge 进 master 分支：*

a -> b -> c -> f -- g -> h (master) \ / d -> e (dev)

*上图中，g 是 merge commit，其他的都是常规 commit。g 的两个 parent 分别是 f 和 e。*

*Revert a Merge Commit*

*当你使用 git revert 撤销一个 merge commit 时，如果除了 commit 号而不加任何其他参数，git 将会提示错误：*

$ git revert 83281a8e9aa1ede58d51a6dd78d5414dd9bc8548 //本人实际git信息，这里对应git演进图中的 g  
error: Commit g is a merge but no -m option was given.  
fatal: revert failed

*在你合并两个分支并试图撤销时，Git 并不知道你到底需要保留哪一个分支上所做的修改。从 Git 的角度来看，master 分支和 dev 在地位上是完全平等的，只是在 workflow 中，master 被人为约定成了「主分支」。*

*于是 Git 需要你通过 m 或 mainline 参数来指定「主线」。merge commit 的 parents 一定是在两个不同的线索上，因此可以通过 parent 来表示「主线」。m 参数的值可以是 1 或者 2，对应着 parent 在 merge commit 信息中的顺序。*

*以上面那张图为例，我们查看 commit g 的内容*：

$ git show 83281a8e9aa1ede58d51a6dd78d5414dd9bc8548 //本人实际git信息，这里对应git演进图中的 g

commit 83281a8e9aa1ede58d51a6dd78d5414dd9bc8548

Merge: 312a518 fa87415 //312a518和fa87415 可以在git log中找到对应的提交信息（只是commit一长串字符的头部分）

......

*那么，$ git revert -m 1 g 将会保留 master 分支上的修改，撤销 dev 分支上的修改。//(1就是1，表示*312a518对应的父来源，2表示fa87415对应的父来源)撤销成功之后，Git 将会生成一个新的 Commit，提交历史就成了这样：

a -> b -> c -> f -- g -> h -> G (master)  
\ /  
d -> e (dev)

*其中 G 是撤销 g 生成的 commit。通过 $ git show G 之后，我们会发现 G 是一个常规提交，内容就是撤销 merge 时被丢弃的那条线索的所有 commit 的「反操作」的合集。*

请注意红色注释部分。

转自：http://hanhan2611.blog.163.com/blog/static/2075771392014428102714312/