

唐巧的技术博客

记录下自己学习的点滴

- [RSS](#)

<input type="text" value="Search"/>
<input type="text" value="Navigate..."/> 

- [首页](#)
- [所有文章](#)
- [iOS开发专题](#)
- [我的微博](#)

Git Submodule的坑

May 8th, 2013

前言

随着近几年的发展，Git 已经成为开源界的标准版本控制工具。开源界的重量级项目，如 Linux, Android, Eclipse, Gnome, KDE, Qt, ROR, Debian，无一例外的都是使用 git 来进行版本控制。如果你还不会 Git，那么恕我直言，你已经 out 了，赶紧抽空充充电吧。本文并不打算做 Git 入门级介绍，想学习 git 的同学，推荐国内作者蒋鑫写的 [《Git 权威指南》](#)。

对于一些比较大的工程，为了便于复用，常常需要抽取子项目。例如我开发的猿题库客户端现在包括 3 门考试，客户端涉及的公共 UI、公共底层逻辑、公共的第三方库、以及公共的答题卡扫描算法就被我分别抽取成了子项目。这些子项目都以 git submodule 的形式，增加到工程中。

在使用了 git submodule 一段时间后，我发现了一些 submodule 的问题，在此分享给大家。

更新 submodule 的坑

submodule 项目和它的父项目本质上是 2 个独立的 git 仓库。只是父项目存储了它依赖的 submodule 项目的版本号信息而已。如果你的同事更新了 submodule，然后更新了父项目中依赖的版本号。你需要在 git pull 之后，调用 git submodule update 来更新 submodule 信息。

这儿的坑在于，如果你 git pull 之后，忘记了调用 git submodule update，那么你极有可能再次把旧的 submodule 依赖信息提交上去。对于那些习惯使用 git commit -a 的人来说，这种危险会更大一些。所以建议大家：

1. git pull 之后，立即执行 git status, 如果发现 submodule 有修改，立即执行 git submodule update
2. 尽量不要使用 git commit -a， git add 命令存在的意义就是让你对加入暂存区的文件做二次确认，而 git commit -a 相当于跳过了这个确认过程。

更复杂一些，如果你的 submodule 又依赖了 submodule，那么很可能你需要在 git pull 和 git submodule update 之后，再分别到每个 submodule 中再执行一次 git submodule update，这里可以使用 git submodule foreach 命令来实现： `git submodule foreach git submodule update`

修改 submodule 的坑

有些时候你需要对 submodule 做一些修改，很常见的做法就是切到 submodule 的目录，然后做修改，然后 commit 和 push。

这里的坑在于，默认 git submodule update 并不会将 submodule 切到任何 branch，所以，默认下 submodule 的 HEAD 是处于游离状态的（‘detached HEAD’ state）。所以在修改前，记得一定要用 git checkout master 将当前的 submodule 分支切换到 master，然后才能做修改和提交。

如果你不慎忘记切换到 master 分支，又做了提交，可以用 cherry-pick 命令挽救。具体做法如下：

1. 用 git checkout master 将 HEAD 从游离状态切换到 master 分支，这时候，git 会报 Warning 说有一个提交没有在 branch 上，记住这个提交的 change-id（假如 change-id 为 aaaa）
2. 用 git cherry-pick aaaa 来将刚刚的提交作用在 master 分支上
3. 用 git push 将更新提交到远程版本库中

以下是相关命令的操作示范和命令行输出结果：

```
1 ➔ ui_common git:(df697f9) git checkout master
2 Warning: you are leaving 1 commit behind, not connected to
3 any of your branches:
4
5     df697f9 forget to check out master
6
7 If you want to keep them by creating a new branch, this may be a good time
8 to do so with:
9
10 git branch new_branch_name df697f911e5a0f09d883f8f360977e470c53d81e
11
12 Switched to branch 'master'
13 ➔ ui_common git:(master) git cherry-pick df697f9
```

使用第三方工具

对于 submodule 的重度使用者，有几个工具可作推荐：

1. [Repo](#) Google 用于管理 Android 项目的工具。
2. [Gitslave](#)

3. [Git Subtree](#)

以上工具，我都没有实际用过，所以无法提供更多信息。

Tips

由于常常使用 submodule 的相关命令，可以在 ~/.gitconfig 文件中将其设置别名，方便操作，我设置的所有相关别名如下：

```
1 [alias]
2   st = status -s
3   ci = commit
4   l = log --oneline --decorate -12 --color
5   ll = log --oneline --decorate --color
6   lc = log --graph --color
7   co = checkout
8   br = branch
9   rb = rebase
10  dci = dcommit
11  sbi = submodule init
12  sbu = submodule update
13  sbp = submodule foreach git pull
14  sbc = submodule foreach git co master
```

参考链接

1. [《why-your-company-shouldnt-use-git-submodules》](#) (需翻墙)
2. [《Git submodule tutorial》](#)

Posted by 唐巧 May 8th, 2013 [shell](#)

关注我的「iOS开发」微信公众号，每天获得精选的 iOS 开发文章和创业心得：



原创文章，版权声明：自由转载-非商用-非衍生-保持署名 | [Creative Commons BY-NC-ND 3.0](#)

分享到： [新浪微博](#) [网易微博](#) [腾讯微博](#) [人人网](#) [豆瓣](#) [更多](#) 9