

# git cherry-pick合并某个commit

## 1、使用方法及其作用

**git cherry-pick**可以**选择某一个分支中的一个或几个commit(s)来进行操作**（操作的对象是**commit**）。例如，假设我们有个稳定版本的分支，叫v2.0，另外还有个开发版本的分支v3.0，我们不能直接把两个分支合并，这样会导致稳定版本混乱，但是又想增加一个v3.0中的功能到v2.0中，这里就可以使用cherry-pick了。

就是对已经存在的commit 进行 再次提交；

使用方法如下：

```
git cherry-pick <commit id>
```

查询commit id 的查询可以使用git log查询（查询版本的历史），最简单的语法如下：

```
git log
```

详细的git log 语法如下：

```
git log [<options>] [<since>..<until>] [--] <path>...
```

主要参数选项如下：

- p：按补丁显示每个更新间的差异
- stat：显示每次更新的修改文件的统计信息
- shortstat：只显示--stat中最后的行数添加修改删除统计
- name-only：尽在已修改的提交信息后显示文件清单
- name-status：显示新增、修改和删除的文件清单
- abbrev-commit：仅显示SHA-1的前几个字符，而非所有的40个字符
- relative-date：使用较短的相对时间显示（例如："two weeks ago"）
- graph：显示ASCII图形表示的分支合并历史
- pretty：使用其他格式显示历史提交信息

结果大概如下：

```
commit 0771a0c107dbf4c96806d22bbc6ef4c58dfe7075  
Author: zhengcanrui <zhengcanrui@cvte.com>  
Date: Mon Aug 8 14:41:54 2016 +0800
```

```
[modify] [what] commit的备注信息
```

其中**0771a0c107dbf4c96806d22bbc6ef4c58dfe7075**就是我们的commit id

**注意：当执行完 cherry-pick 以后，将会 生成一个新的提交；这个新的提交的哈希值和原来的不同，但标识名一样；(commit id会变)**

## 2、实践

首先切换到你要添加commit的分支，如：你要将A分支上面的commit添加到B分支上面，我们可以要先切换到B分支上面。（注意：cherry-pick是一个本地的操作，假如你pull代码之后有人在A分支上有了新的commit，需要你先pull代码在进行cherry-pick，原因及其错误提示请见最后）。

```
git checkout B
```

将0771a0c107dbf4c96806d22bbc6ef4c58dfe7075这个commit（提交）合并到B分支上面。正常情况下，可以给出全部的commit id，也可以只给出前面的一段，只要你提交中没有这一段重复的就好，剩下的部分git会帮你填充。

```
git cherry-pick 0771a0c107dbf4c #将上面的commit id为0771a0c107dbf4c96806d22bbc6ef4c58dfe7075的提交添加到B分支上面
```

## 1.成功的情况

顺利的话，出现下面的情况证明你已经成功了

```
Finished one cherry-pick.  
# On branch B  
# Your branch is ahead of 'origin/B' by 1 commits.
```

## 2.有冲突的情况

下面是有文件冲突，和15a2b6c61927e5aed6718de89ad9dafba939a90b这个提交冲突

```
Automatic cherry-pick failed. After resolving the conflicts,  
mark the corrected paths with 'git add <paths>' or 'git rm <paths>'  
and commit the result with:
```

```
git commit -c 15a2b6c61927e5aed6718de89ad9dafba939a90b
```

解决的冲突的方法也和普通的一样，手工检查。

### 1) 查看冲突的文件

使用git status

```
both modified:      app/models/user.rb
```

### 2) 打开上面的那个文件，解决冲突。执行add命令、执行commit命令，最后在提交即可了

## 3、遇到的一些错误

使用下面cherry-pick命令执行某个commit（编号为：77c6905dcf7f946cff594a69a33d12e22bedfae4）

```
git cherry-pick 77c6905dcf7f946cff594a69a33d12e22bedfae4
```

出现了如下的错误：

```
fatal: bad object 77c6905dcf7f946cff594a69a33d12e22bedfae4
```

场景及出现错误的原因：

我的情况是在B分支的同步A分支的一个commit，出现了如标题的错误。我是直接在web上看到A分支新提交的这个commit，然后我直接在本地的B分支中进行git cherry-pick xxx。就出现了这个问题。经过尝试**问题是出在了我没有切到B分支pull一下。总结过来就是git cherry-pick是本地特性，本地要有这个commit才可以被git cherry-pick。**

**致谢：**感谢您的阅读！有错误请加以纠正，有其他欠缺的东西请不吝赐教，谢谢！