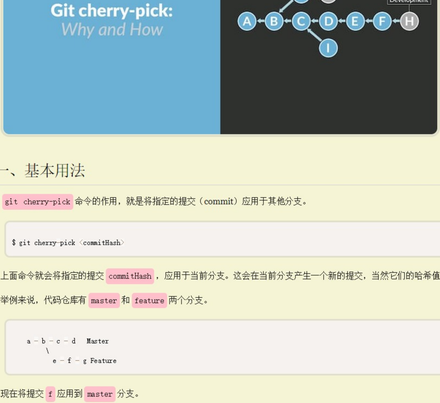
# git cherry-pick 教程

作者： 阮一峰

日期： 2020年4月27日

对于多分支的代码库，将代码从一个分支转移到另一个分支是常见需求。

这时分两种情况。一种情况是，你需要另一个分支的所有代码变动，那么就采用合并（`git merge`）。另一种情况是，你只需要部分代码变动（某几个提交），这时可以采用 Cherry pick。




## 一、基本用法

`git cherry-pick` 命令的作用，就是将指定的提交（commit）应用于其他分支。

```
$ git cherry-pick <commitHash>
```

上面命令就会将指定的提交 `commitHash`，应用于当前分支。这会在当前分支产生一个新的提交，当然它们的哈希值会不一样。

举例来说，代码仓库有 `master` 和 `feature` 两个分支。

```
a - b - c - d   Master
     \
      e - f - g Feature
```

现在将提交 `f` 应用到 `master` 分支。

```
# 切换到 master 分支
$ git checkout master

# Cherry pick 操作
$ git cherry-pick f
```

上面的操作完成以后，代码库就变成了下面的样子。

```
a - b - c - d - f   Master
     \
      e - f - g Feature
```

从上面可以看到，`master` 分支的末尾增加了一个提交 `f`。

`git cherry-pick` 命令的参数，不一定是提交的哈希值，分支名也是可以的，表示转移该分支的最新提交。

```
$ git cherry-pick feature
```

上面代码表示将 `feature` 分支的最近一次提交，转移到当前分支。

## 二、转移多个提交

Cherry pick 支持一次转移多个提交。

```
$ git cherry-pick <HashA> <HashB>
```

上面的命令将 A 和 B 两个提交应用到当前分支。这会在当前分支生成两个对应的新提交。

如果想要转移一系列的连续提交，可以使用下面的简便语法。

```
$ git cherry-pick A..B
```

上面的命令可以转移从 A 到 B 的所有提交。它们必须按照正确的顺序放置，提交 A 必须早于提交 B，否则命令将失败，但不会报错。

注意，使用上面的命令，提交 A 将不会包含在 Cherry pick 中。如果要包含提交 A，可以使用下面的语法。

```
$ git cherry-pick A^..B
```

## 三、配置项

`git cherry-pick` 命令的常用配置项如下。

**（1）`-e`，`--edit`**

打开外部编辑器，编辑提交信息。

**（2）`-n`，`--no-commit`**

只更新工作区和暂存区，不产生新的提交。

**（3）`-x`**

在提交信息的末尾追加一行（`(cherry picked from commit ...)`），方便以后查到这个提交是如何产生的。

**（4）`-s`，`--signoff`**

在提交信息的末尾追加一行操作者的签名，表示是谁进行了这个操作。

**（5）`-m parent-number`，`--mainline parent-number`**

如果原始提交是一个合并节点，来自于两个分支的合并，那么 Cherry pick 默认将失败，因为它不知道应该采用哪个分支的代码变动。

`-m` 配置项告诉 Git，应该采用哪个分支的变动。它的参数 `parent-number` 是一个从 `1` 开始的整数，代表原始提交的父分支编号。

```
$ git cherry-pick -m 1 <commitHash>
```

上面命令表示，Cherry pick 采用提交 `commitHash` 来自编号1的父分支的变动。

一般来说，1号父分支是接受变动的分支（the branch being merged into），2号父分支是作为变动来源的分支（the branch being merged from）。

## 四、代码冲突

如果操作过程中发生代码冲突，Cherry pick 会停下来，让用户决定如何继续操作。

**（1）`--continue`**

用户解决代码冲突后，第一步将修改的文件重新加入暂存区（`git add .`），第二步使用下面的命令，让 Cherry pick 过程继续执行。

```
$ git cherry-pick --continue
```

**（2）`--abort`**

发生代码冲突后，放弃合并，回到操作前的样子。

**（3）`--quit`**

发生代码冲突后，退出 Cherry pick，但是不回到操作前的样子。

## 五、转移到另一个代码库

Cherry pick 也支持转移另一个代码库的提交，方法是先将该库加为远程仓库。

```
$ git remote add target git://gitUrl
```

上面命令添加了一个远程仓库 `target`。

然后，将远程代码抓取到本地。

```
$ git fetch target
```

上面命令将远程代码仓库抓取到本地。

接着，检查一下要从远程仓库转移的提交，获取它的哈希值。

```
$ git log target/master
```

最后，使用 `git cherry-pick` 命令转移提交。

```
$ git cherry-pick <commitHash>
```

（完）