

Mercurial Patch Queue

Sep 18, 2011 by Tao

Mercurial是一个使用Python编写的分布式版本管理系统，其具有良好的跨平台性，并且带有一个简单易用的Web界面。在平时我经常使用Mercurial来管理源代码。

Mercurial Queue是它的一个非常强大的插件，其可以用来解决源代码管理中经常碰到的一些问题，例如：

- 有时候我们需要在代码中试验一些新的想法，但是这些想法不大成熟，甚至可能是完全行不通的，我们不想这些不成熟的修改“污染”版本库的历史。由于版本库的历史是不可变更的，一旦commit之后就无法把它们从历史记录中删除。
- 使用patch系统是一个解决这个问题方法，但是手动管理patch非常低效。

MQ (Mercurial Queue) 克服了这些缺点，它是Mercurial的一个插件，是与Mercurial集成的自动化patch管理系统。

启用MQ插件

在最近版本的Mercurial中已经自带了MQ插件，通过修改配置文件来启用，在.hgrc中添加以下配置：

```
[extensions]
mq =
```

使用MQ插件

在使用Patch Queue之前，首先要在一个版本库上初始化MQ：

```
hg init --mq
```

下面假设我们要对版本库中的文件做一些修改，但又不想马上把修改commit成为永久性的changeset，因此新建一个patch：

```
hg qnew -m "some changes" patch1
```

接下来对文件做一些修改，然后更新patch：

```
hg qrefresh
```

此时用hg diff已经看不到被修改的文件了。

如果还有其他修改，可以再新建一个patch：

```
hg qnew -m "more changes" patch2
```

MQ使用一个stack来管理patch，可以通过hg qpush和hg qpop来在这个stack上添加一个或者移除一个patch。要查看patch的状态，可以使用hg qseries命令。

当我们已经完成了修改，希望将这系列的patch转换为永久性的changeset时，可以使用hg qfinish命令：

```
hg qfinish -a
```

上面的命令将所有已经应用的patch提交到版本库，成为changeset，如果我们对所做的修改不满意，只需要用hg qdelete把patch删除即可。

Tags: [DVCS](#), [Mercurial](#)

0 Comments IMCT 

 1 Login 

 Recommend  Tweet  Share

Sort by Best 

Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Name

Be the first to comment.



Copyright © 2017

Powered by [Hugo](#) & [Pixyll](#)