**1、本地Repository的创建**

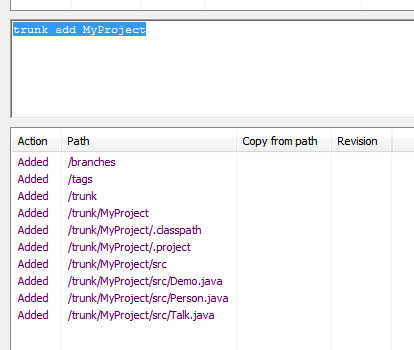
repository的创建很简单，假设我要在D:\TortoiseSVN\TestRepository目录中创建repository，只需右键TestRepository目录，依次选择"TortoiseSVN" -> "Create repository here"便完成了repository的创建。

**2、Check out**

假设要check out到D:\TortoiseSVN\TestSVN，同样很简单，在D:\TortoiseSVN目录下创建TestSVN目录，然后在该目录上右键，选择"SVN Check out..."，在弹出的窗口中的"URL of repository"中填入"file:///D:/TortoiseSVN/TestRepository"，其他默认即可，最后点击ok。

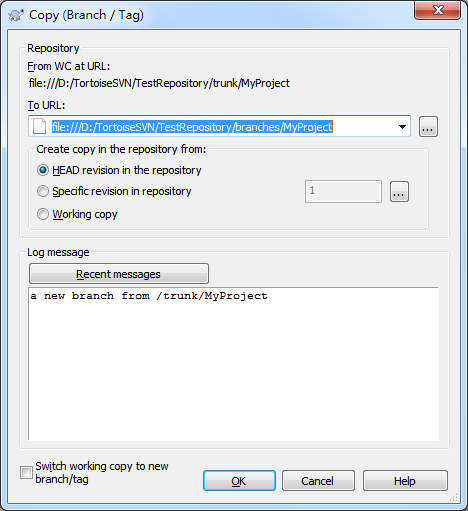
**3、trunk创建新项目MyProject**

相当简单就不赘述了，只列出本次操作所作出的修改：



**4、创建branch**

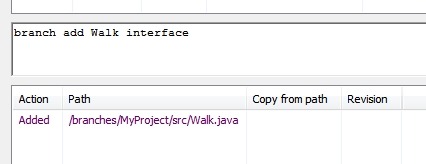
在/trunk/MyProject目录上右键，依次选择"TortoiseSVN" -> "Branch/tag..."，在弹出窗口的"To URL"中填入分支的地址，在这里目标revision选择HEAD revision，如下图所示，添加log后点击ok分支便建立了。这个操作速度非常快，新建的branch在repository中其实只是一个指向trunk某个revision的软连接而已，并没有真的复制文件。



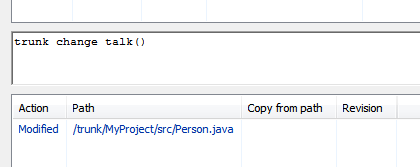
**5、Check out分支**

右键TestSVN目录选择"TortoiseSVN Update"即可将刚刚建立的分支下载回本地。进入/branches/MyProject目录下你会发现其文件结构和/trunk/MyProject一模一样。

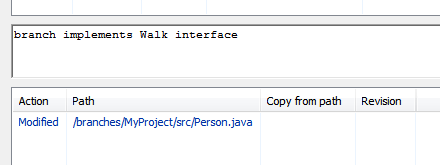
**6、branch提交一个新文件**



**7、trunk紧接着提交一个修改**



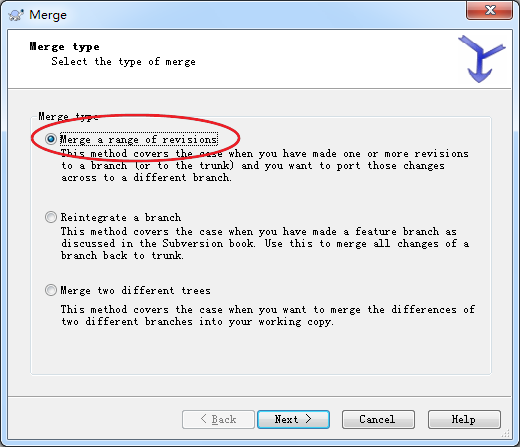
**8、branch再次提交一个修改**



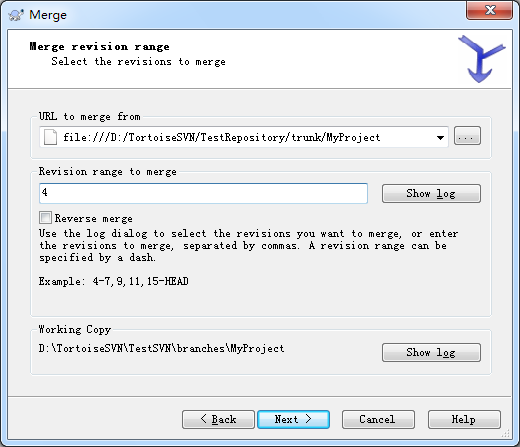
**9、将trunk中的修改同步到branch**

6-8演示的是branch和trunk在独立、并行地开发。为了防止在“错误”的道路上越走越远，现在branch意识到是时候和trunk来一次同步了（将trunk合并到branch）。

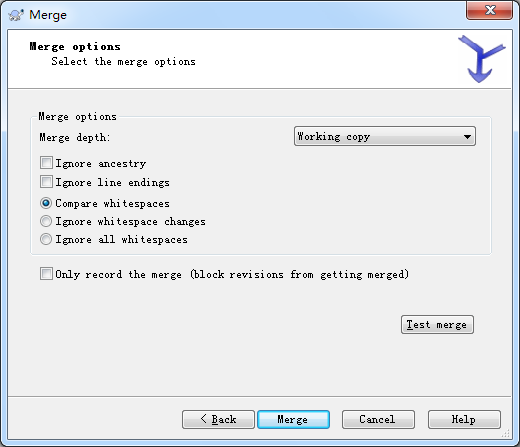
首先，在本地trunk中先update一下，有冲突的解决冲突，保证trunk和repository已经完全同步，然后在/branches/MyProject上右键，依次选择"TortoiseSVN" -> “Merge...”，在弹出的窗口中选择第一项"Merge a range of revision"，这个类型的Merge已经介绍得很清楚，适用于将某个分支或主线上提交的多个revision间的变化合并到另外一个分支上。



点击next后，出现如下窗口：

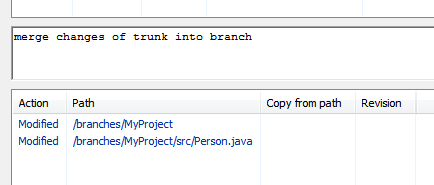


由于是要从trunk合并到branch，理所当然这里的"URL to merge from"应该填trunk的路径，"Revision range to merge"很好理解，就是你要将trunk的哪些revision所对应的变化合并到branch中，可以是某一连串的revision，比如4-7，15-HEAD，也可以是某个单独的revision号。由于在r4中，trunk修改了Person.[**Java**](http://lib.csdn.net/base/javase)中的talk()方法，所以这里的revision只需填4即可。点击next后出现下图：



在这里只需保留默认设置即可。在点击Merge按钮前你可以先Test merge一把，看成功与否，以及merge的详细信息。点击Merge按钮后trunk所做的修改将同步到branch中。

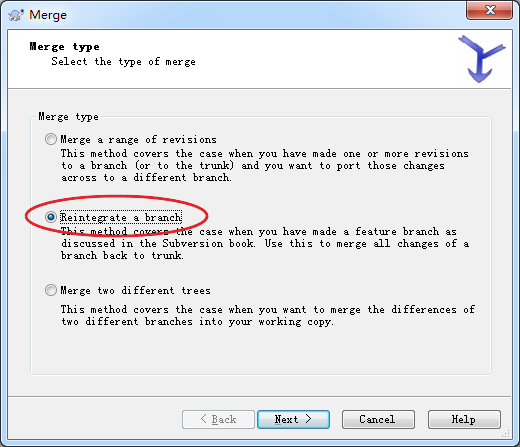
**10、提交合并后的branch**



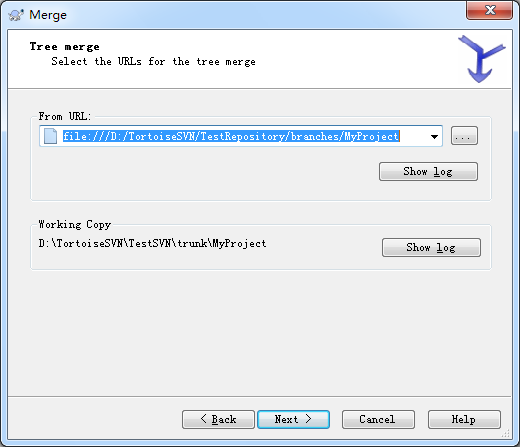
至此，branch已经完全和trunk同步，branch和trunk的代码相处很融洽，没有任何冲突，如果branch已经开发结束，那是时候将branch合并回trunk了，当然，如果branch还要继续开发，那你将不断地重复6-10这几个步骤。

**11、将branch合并回trunk**

在/trunk/MyProject上右键（注意是在主线的目录上右键），依次选择"TortoiseSVN" -> "Merge..."，在弹出的窗口中，Merge type选择第二项"Reintegrate a branch"，这种类型的合并适合在分支开发结束后将所有的改动合并回主线。



点击next后出现如下窗口：



在这里，"From URL"选择/branches/MyProject，无需选择revision号，Reintegrate会将branch上所有修改合并到trunk。后面的步骤和上文第9步中的一样，不再啰嗦了。如无意外，branch将成功合并到trunk，你需要做的只是将合并后的trunk赶紧commit！

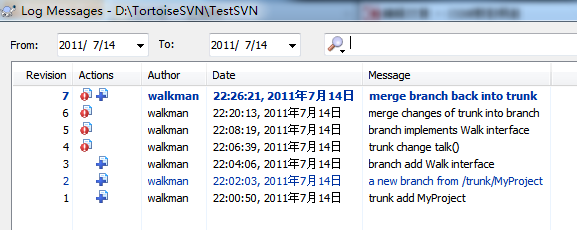
**12、提交合并后的trunk**

so easy...

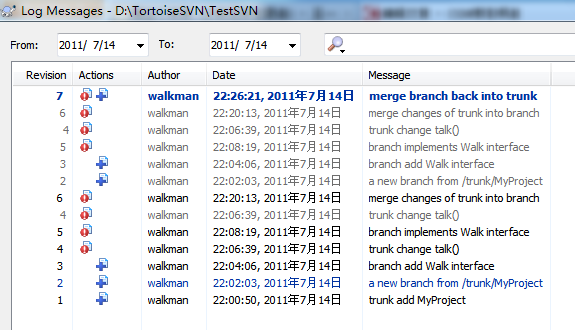
**13、删除branch**

如果你认为你新加的功能已经开发完成了，你可以删除你的分支

到这里，我已经给你演示完了整个过程，我一身的汗也下来了，我想罢工了，不过最后我们还是看看所有的log信息吧，通过log能发现我们干的所有事情：



r1-r7正是我上文在干的事情，从Message中你能发现我对trunk和branch都干了什么，另外，在Log Messages窗口的左下角勾选了"Include merged revisions"你还能看到额外的Merge information：



图中灰色的是和merge相关的log，共发生了两次merge，第一次是在r6，在r6中，branch合并了trunk在r4时提交的变化；第二次是在r7，在r7中，trunk合并了branch从r2到r6的所有变化。