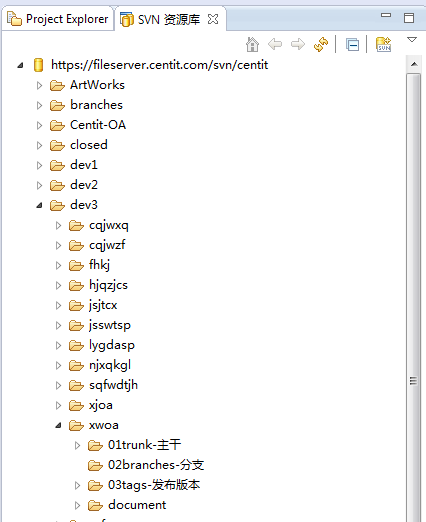
## Eclipse下SVN版本分支创建与分支合并

新入职的单位需要使用SVN对项目代码进行管理，需要频繁的建立分支，以及将分支（branches）合并到主版本上去。以前都是整个小组都在主版本（trunk）上进行开发，所以从来没有研究过关于分支和合并的知识，现在用到了，查阅了好多资料算是把分支创建和分支合并弄清了吧，所以记录下来，防止以后忘了（估计是不会忘了，经常用~）。

首先关于Eclipse下SVN插件的安装这里就不详细介绍了，网上教程有很多。但大体可以分为通过Eclipse中Help中的Install New Software进行安装，或者是通过Link的方式进行安装。前一种方式的缺点是特别慢，而且后续如果需要对插件进行管理时也不好管理。后一种方式特别快，而且时候对插件进行管理时也方便。这里只介绍个Eclipse SVN插件的下载网站：<http://subclipse.tigris.org/servlets/ProjectDocumentList?folderID=2240>。我下载和使用的版本是site-1.0.6。

### SVN目录结构

一般情况下，使用SVN进行版本管理时的目录结构如下所示：



Trunk ： 主干目录，此目录下的文件为基准文件。

Branches ： 用于开发的分支目录。

Tags ： 用于发布的版本目录。

### 分支版本的命名方式

通常情况下branches下的版本名称的定义规则为project name + 日期时间 + 功能点 ， 比如 oa\_xwoa\_2016-12-7\_dashbord，在创建每一个branch时，必须增加标注。根据公司要求的不同，命名方式有可能会有不同。

### 创建一个分支版本

分支版本实际上就是主版本（基准版本，基线版本）的某个版本的一个拷贝，通过命名标识他分支版本的身份。比如，现有如下主版本项目，如图1所示：

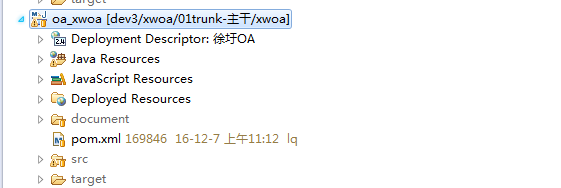
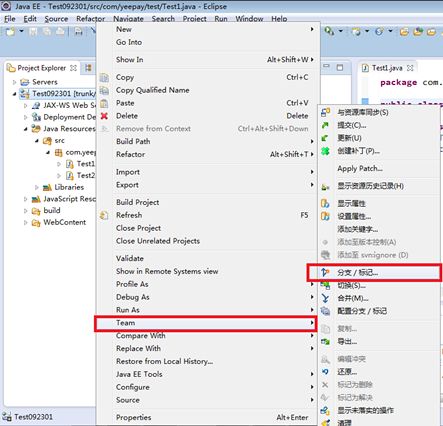


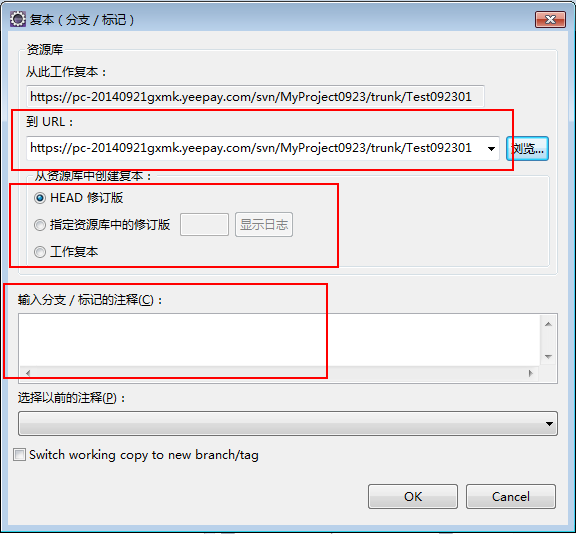
图1

我们创建一个分支。如图2所示：



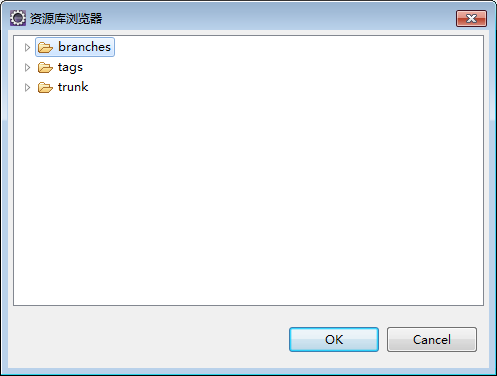
分支/标记在英文版的SVN插件中叫做Branch/Tag。

弹出对话框进行分支信息录入，如下图所示弹出的对话框：

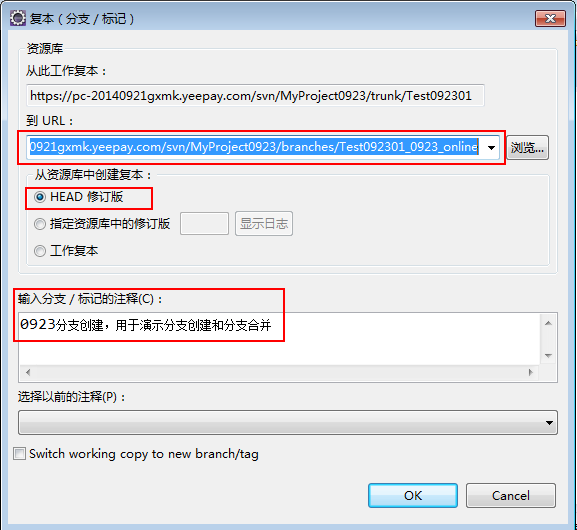


第一个红框要求我们选择分支的存放位置和分支的名称，可以点击浏览按钮进行位置选择，这里一定要将分支存放的位置放在branches目录下，分支的名称也需要按照分支命名规则进行填写。

点击浏览按钮弹出的对话框如下：



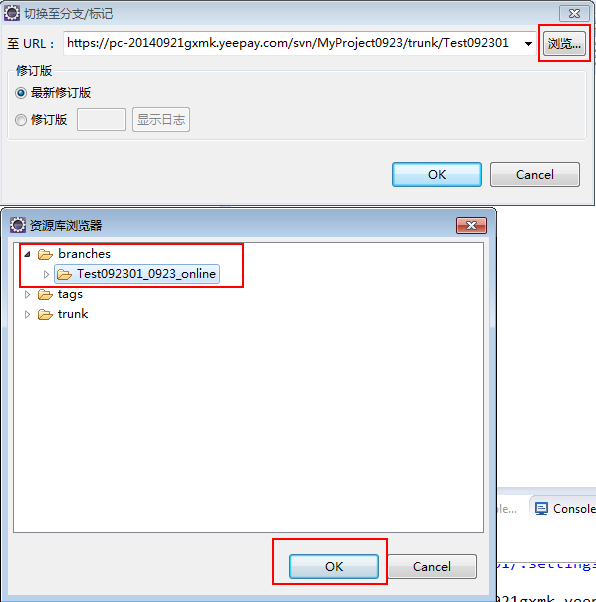
选择branches并对新建的分支进行命名，并选择从哪个主版本进行拷贝，以及填写分支创建注释，推荐填写分支创建注释。



我选择将分支命名为Test092301\_0923\_Online，存放在branches目录下，同时以最新的主版本作为分支拷贝的对象。填写必要的注释完成分支创建。至此一个主版本的分支就已经创建完毕了。

### 切换到分支版本

同样使用右键→Team→切换（Switch）将当前项目切换到最新的分支中去。在弹出的对话框中选择你要切换的分支，点击确认即可。也可以使用检出功能将分支检出成为本地项目。但是这里我们推荐使用第二种方式，也就是检出的方式将分支存放在本地，因为可以同时看到主版本和分支版本的项目代码，后续合并起来会十分方便。使用切换方式示例如下图所示。



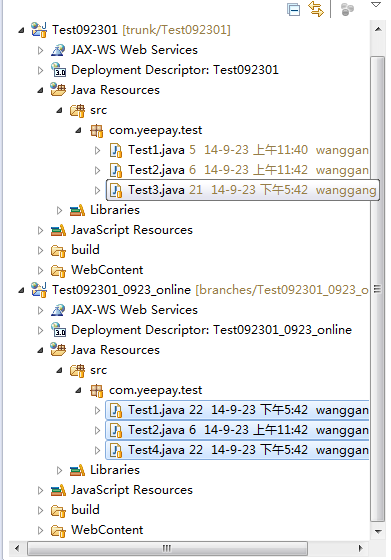
使用检出方式检出分支代码后的项目目录如下图所示，可以清楚的看到主版本和分支版本的目录。



仔细对比主版本和分支版本之间的代码可以发现分支版本和主版本之间一模一样，这也就是为什么说分支版本是主版本的一个拷贝了。

### 修改主版本和分支版本

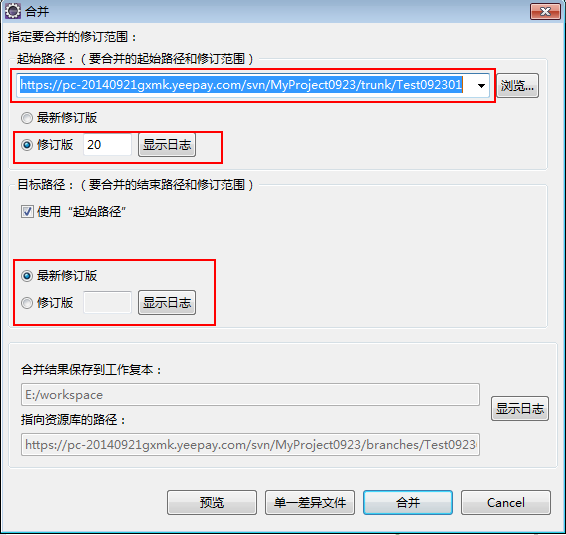
修改分支版本中的Test1.java文件，向主版本中添加一个Test3.java文件，向分支版本中添加Test4.java文件，文件中有一句打印语句即可。将分支版本的修改和主版本的代码修改都进行提交，修改后的项目目录如下所示：



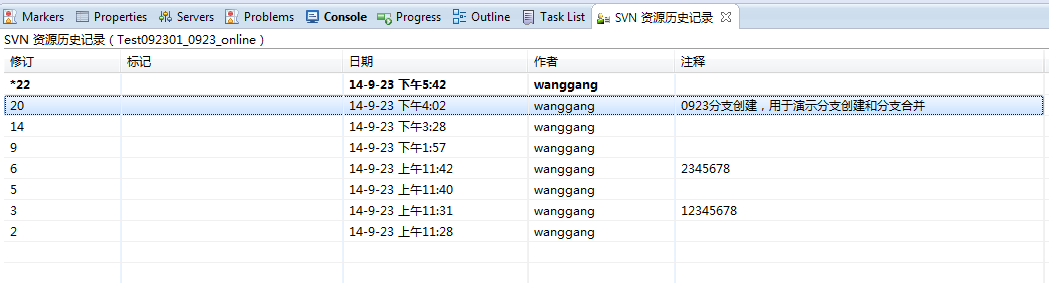
### 将主版本中的修改合并到分支版本中去

开发中常见分支版本开发过程中需要将主版本中的修改合并到分支版本中去，以避免分支版本离主版本太远。以之前的代码为例，主版本相对于分支版本不一样的文件有两个，一个是Test1.java文件，该文件在分支版本中进行了修改。一个是Test3.java文件，该文件分支版本中没有。

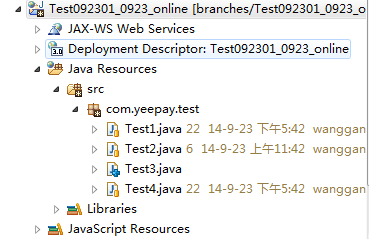
选择项目右键→Team→合并（Merge）在弹出框中选择要进行合并的原文件，这里因为是要将主版本合并到分支版本中，因此目标我们选择trunk下的主版本项目。同时需要指定要进行合并的版本号区间，如下图所示，会合并起始版本号（上面的）到截至版本号（下面的）之间的版本修改记录，注意这里合并的是不包括起始版本在内版本。如下图所示，由于打分支版本的时候，分支版本的版本号是20，所以起始版本可以从20开始，截至版本可以到某一个版本也可以到最新。



通过点击分支版本右键→Team→显示历史资源记录可以查看该分支在生成时的版本号。如下图所示：



点击合并按钮之后，可以看到分支版本中多了主版本项目中新增的Test3.java文件，而其他文件包括分支版本中修改的Test1.java并没有做任何改变，这就实现了主版本项目到分支版本项目代码的合并。合并后的分支版本项目目录如下图所示：

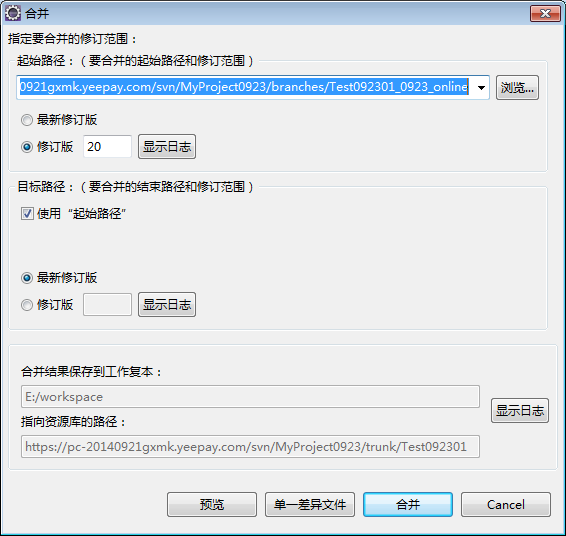


将Test3.java文件进行合并，完成分支版本代码库的更新。至此，将主版本中的代码合并到分支版本代码的过程就介绍完了。

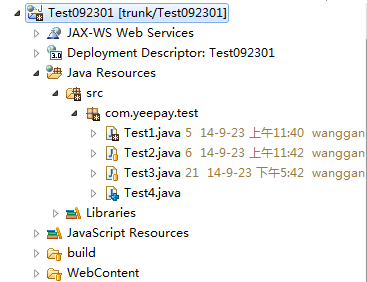
### 将分支版本的修改合并到主版本中

操作类似于之前介绍的将主版本中的修改合并到分支版本中。当我们在分支版本中开发完成后，就需要将分支版本中的修改合并到主版本中。

首先对主版本和分支版本分别进行同步，保证两个版本的代码是最新的。然后主版本进行合并，由于这次代码来源是分支版本，因此我们在对话框选择时选择branches目录下的某个分支版本。版本号的起点可以选择打分支版本时分支版本对应的版本号，截至版本号可以选择分支版本当前最新的版本号。注意其实路径一定要是分支版本。



点击合并按钮后，可以注意到主版本的代码有所改变，如图所示，Test1.java变为了在分支版本中修改后的代码，添加了Test4.java文件。



**合并分支版本至主版本后，主版本中的文件修改还仅仅是在本地，需要再次针对分支版本修改的内容进行详尽的测试后才能将主版本中的修改进行提交，否则可能会对主版本产生一定的影响。**