以svn为例，git的master相当于trunk，dev分支相当于branches

--------------------------------------------------------------------------------------------------------------------------------------------

trunk：是用来做主方向开发的一直向前进行，一个新模块的开发，这个时候就放在trunk，当模块开发完成后，需要修改，就用branch。

branch：是用来做并行开发的，这里的并行是指和trunk进行比较。

tag：是用来做一个milestone（里程碑）的，不管是不是发布版本，但都是一个可用的版本，不需要修改如果要修改就要选一个tag进行branche开发。

--------------------------------------------------------------------------------------------------------------------------------------------

为什么要分3个目录？且都是重复的代码重新Copy？来看下下面这个例子就明白这样做的好处了。

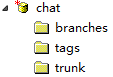
例如：开发一个聊天软件QQ，第一期只要求软件具有基本的聊天功能，后期会加入视屏聊天、魔法表情、换皮肤等功能。

当第一期开发完，经测试人员测试后没问题。此时应该给予当前代码库打tag命名为QQ\_tag\_release\_v1.0.0，同时trunk如火如荼进入v2.0.0开发，但是此时经广大用户反馈v1.0.0的QQ登录后经常闪退，由于v2.0.0正在开发即使bug改了也发布不了。那么就要基于QQ\_tag\_release\_v1.0.0的标记（tag）做一个分支（branch）命名为QQ\_branch\_bugfix\_v1.0.0并基于这个分支快速进行BUG修改，等到BUG修改结束后做一个tag命名为QQ\_tag\_release\_v1.0.1，根据需要决定QQ\_branch\_bugfix\_v1.0.0是否合并（Merge）并入主干（trunk）。

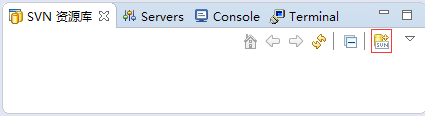
--------------------------------------------------------------------------------------------------------------------------------------------

步骤：

1、新建svn仓库单项目/多项目仓库都可以，结构如下图这样



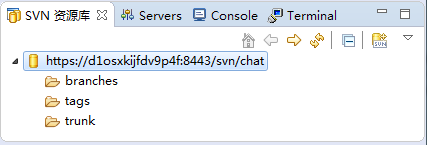
2、新建资源库



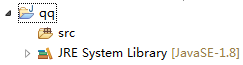
填写URL仓库地址点击“Finish”



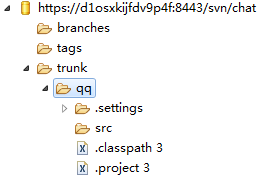
如下



新建qq项目导入svn的trun目录（选中项目右键Team-->Share Projects...）

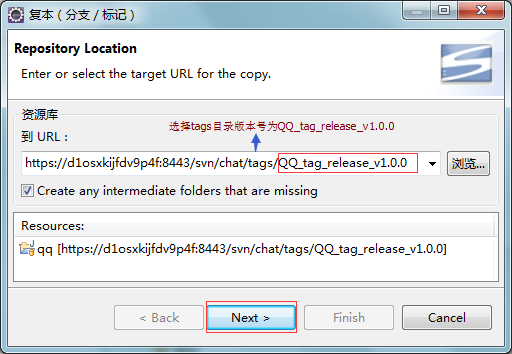


导入后项目后的svn仓库如下

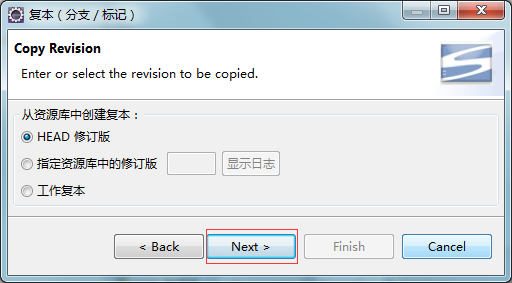


由于紧锣密鼓的工作v1.0.0终于开发完（v1.0.0只有基本的聊天功能），经测试没有发现问题可以提供给用户使用，同时在此代码库上打一个tag版本命名为QQ\_tag\_release\_v1.0.0

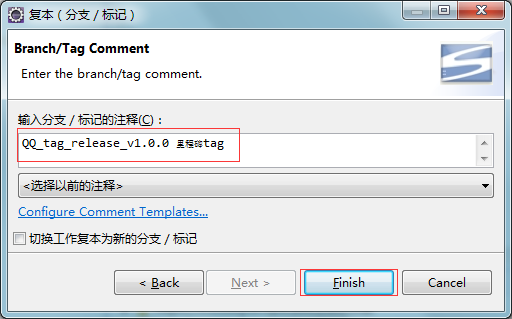
选中项目右键 “Team--> 分支/标记... ” 填写要打tag的版本号后点击”Next“



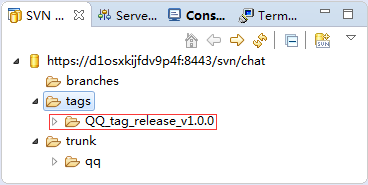
下一步”Next“



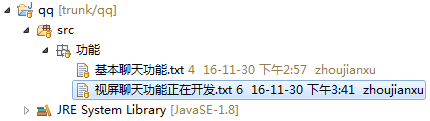
填写注释后点击”Finish“



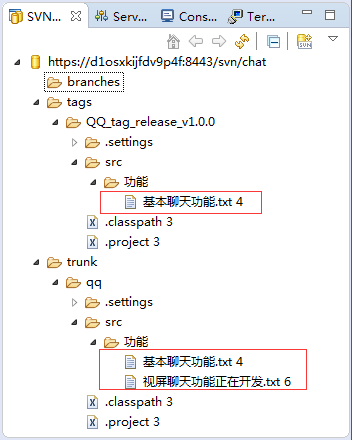
这时svn资源库tag目录下会多了个QQ\_tag\_release\_v1.0.0



同时trunk如火如荼进入v2.0.0开发，视屏聊天功能正在开发

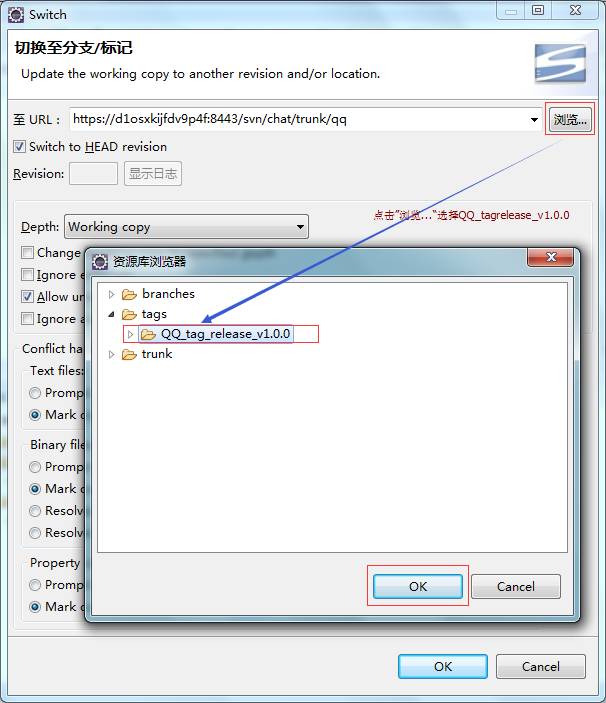


看一下这时的svn资源库

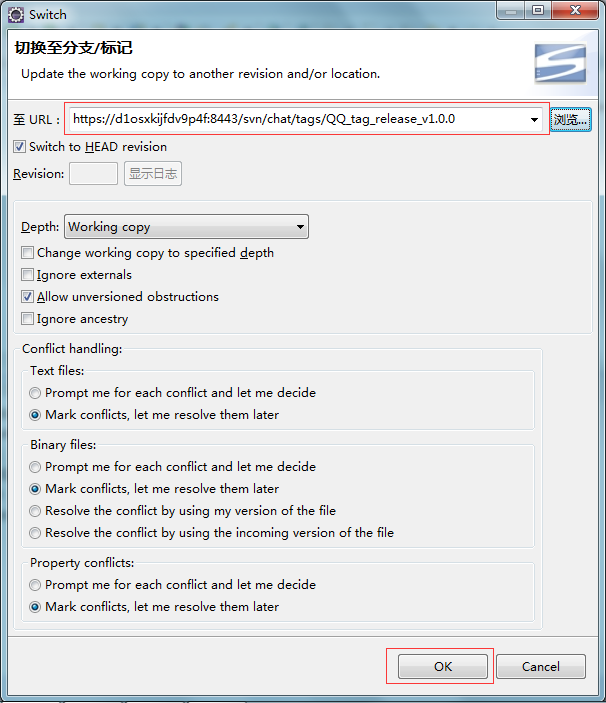


此时经广大用户反馈v1.0.0的QQ登录后经常闪退，由于v2.0.0正在开发即使bug改了也发布不了。那么就要基于QQ\_tag\_release\_v1.0.0的标记（tag）做一个分支（branch）命名为QQ\_branch\_bugfix\_v1.0.0

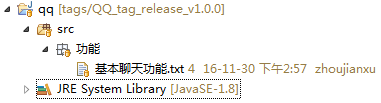
选择项目右键 ”Team“-->”切换(S...)“ 选择tags的QQ\_tag\_release\_v1.0.0



更改路径后点击”OK“

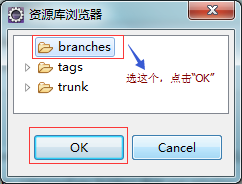


已经切换到标记的QQ\_tag\_release\_v1.0.0

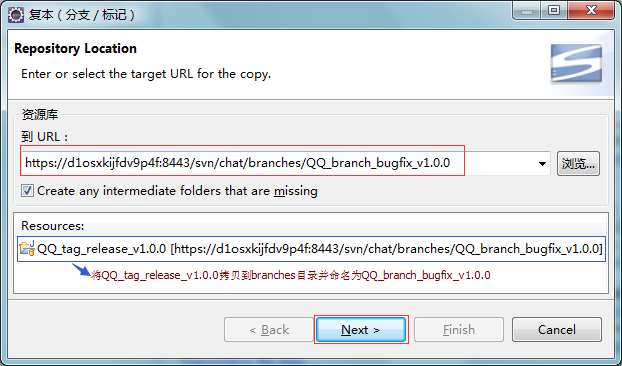


接下来要以QQ\_tag\_release\_v1.0.0 为基础创建一个branch名为QQ\_branch\_bugfix\_v1.0.0并基于这个分支快速进行BUG修改

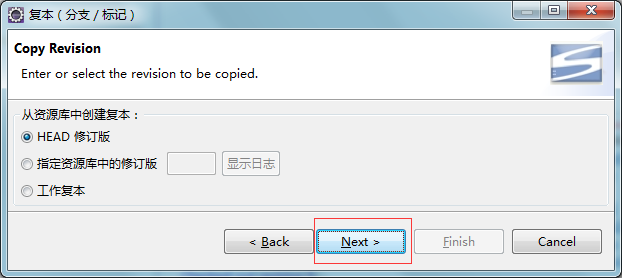
选中项目右键 “Team--> 分支/标记... ” 点击 ”浏览...“ 选择branches目录



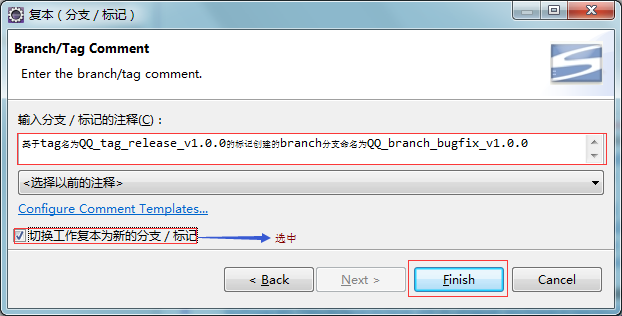
填写要创建分支（branch）名为 ”QQ\_branch\_bugfix\_v1.0.0“ 后点击 ”Next“



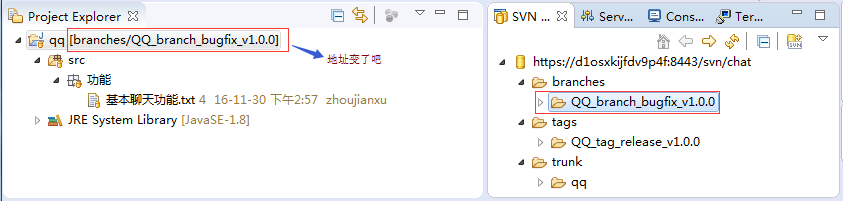
点击”Next“



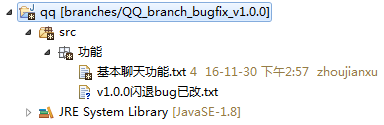
填写注释后选中 ”切换工作复本为新的分支/标记“ 后 点击 ”Finish“



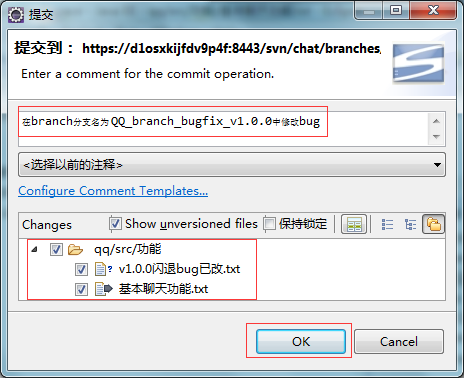
看下工作空间的和svn资源库的变化



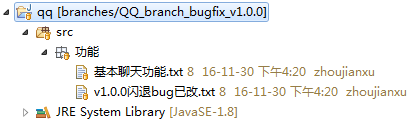
在branch分支名为 ”QQ\_branch\_bugfix\_v1.0.0“ 中修改bug



提交修改bug后的代码

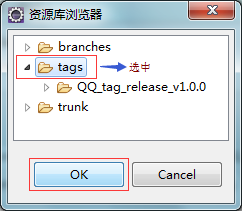


提交后

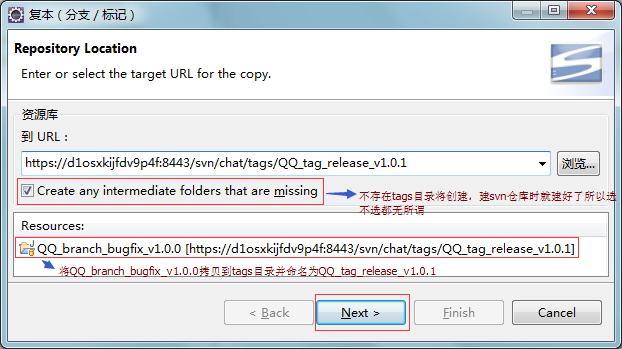


闪退bug已经修改可以提供给用户使用了，接下来要再打一个tag命名为 ”QQ\_tag\_release\_v1.0.1“

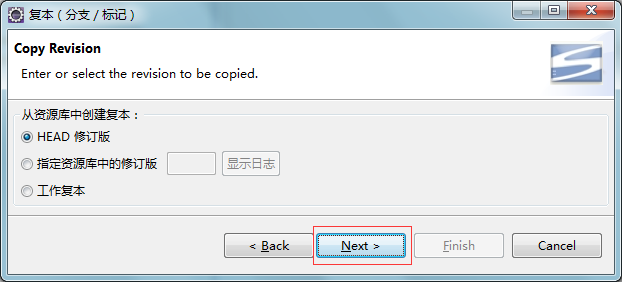
选中名为”QQ\_branch\_bugfix\_v1.0.0“ 的分支项目右键 “Team--> 分支/标记... ” 点击 ”浏览...“ 选择tags目录



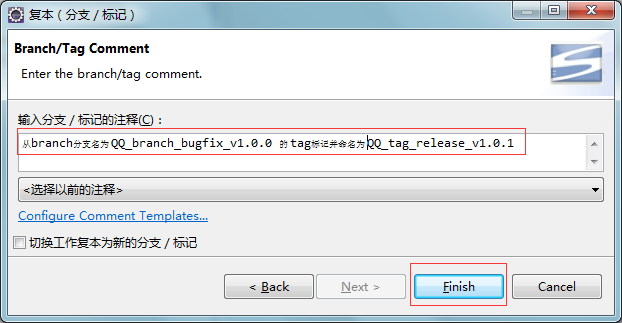
填写要创建标记（tag）名为 ”QQ\_tag\_release\_v1.0.1“ 后点击 ”Next“



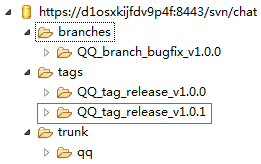
点击”Next“



填写注释信息后点击”Finish“



来看看资源库中有什么变化，tags目录下多了一个名为”QQ\_tag\_release\_v1.0.1“ 的标记



--------------------------------------------------------------------------------------------------------------------------------------------

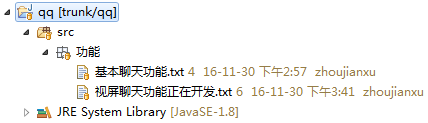
**下面演示 将 分支（branches） 合并到 主干（trunk）**

根据需要决定将分支名为 ”QQ\_branch\_bugfix\_v1.0.0“ 的branch 合并（Merge）并入主干（trunk）。

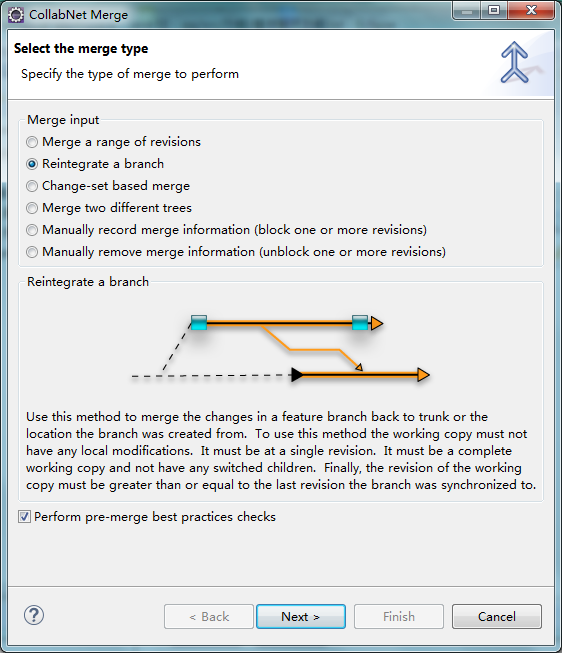
步 骤：

1. 要想将分支合并到主干，首先需要切换到主干。  
在项目上面点右键，选择Team — 切换，URL地址为主干的访问路径

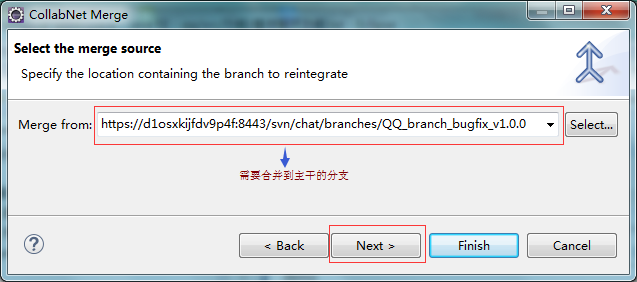
（略过。。。 上面有）



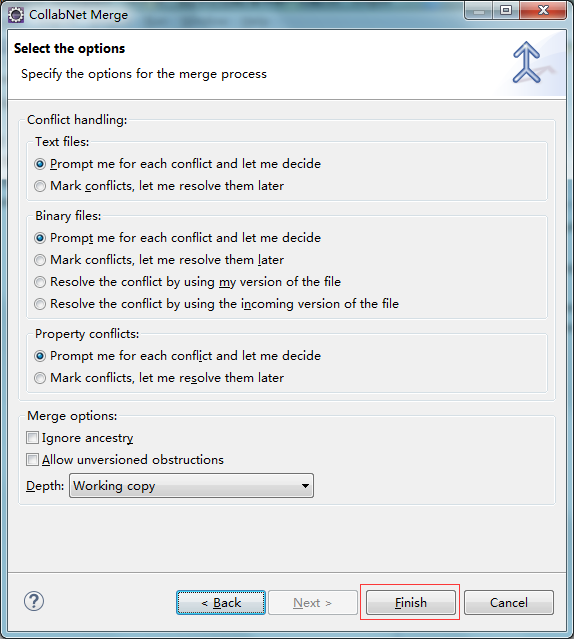
2. 在项目上面点右键，选择 ”Team“--> ”合并(M)...“ ，选中“Reinitegrate a branch”。



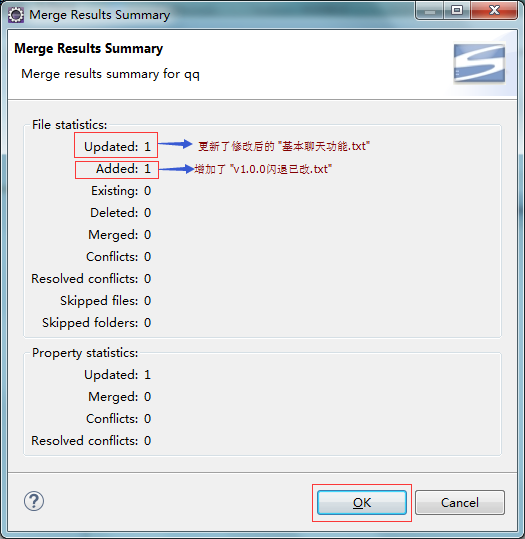
3. 选择要合并到主干的分支”QQ\_branch\_bugfix\_v1.0.0“



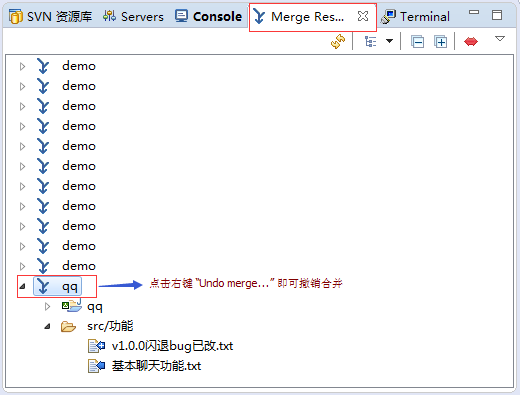
4. 点击”Finish“ 结束合并



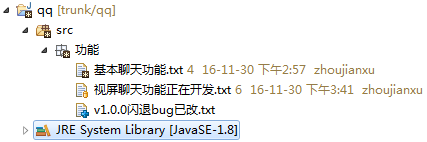
5. 点击”OK“



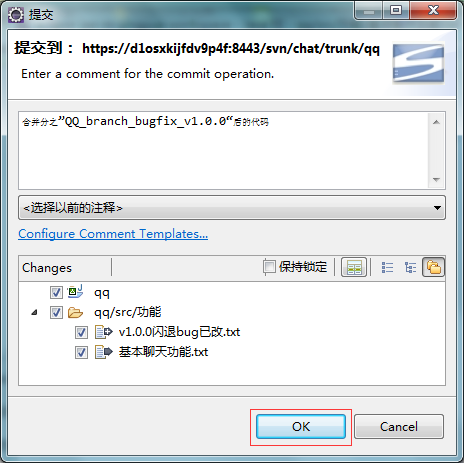
6. 如果想要撤销合并选中项目右键”Undo merge...“



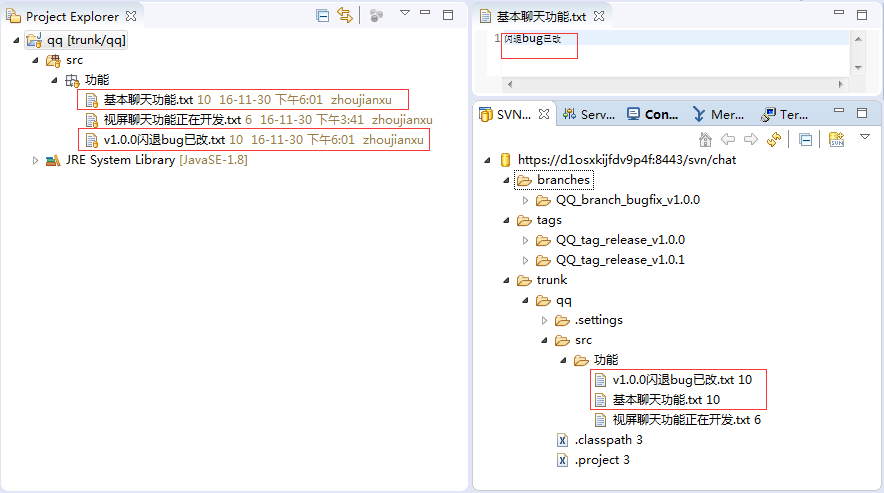
 7. trunk中的qq项目下就会多了从分支”QQ\_branch\_bugfix\_v1.0.0“里Merge合并的代码



8. 提交合并后的代码



9. 提交后如下



--------------------------------------------------------------------------------------------------------------------------------------------

一般的，我们的所有的开发都是基于trunk进行开发，当一个版本/release开发告一段落（开发、测试、文档、制作安装程序、打包等）结束后，代码处于冻结状态（人为规定，可以通过hook来进行管理）。此时应该基于当前冻结的代码库，打tag。当下一个版本/阶段的开发任务开始，继续在trunk进行开发。  
此时，如果发现了上一个已发行版本（ReleasedVersion）有一些bug，或者一些很急迫的功能要求，而正在开发的版本（DevelopingVersion）无法满足时间要求，这时候就需要在上一个版本上进行修改了。应该基于发行版对应的tag，做相应的分支（branch）进行开发。