Svn使用教程

1. **概述**

下载地址：



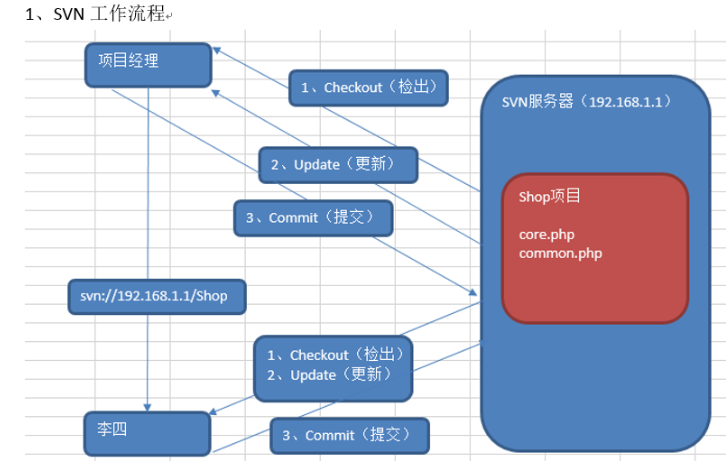
作用：

1.共同协作

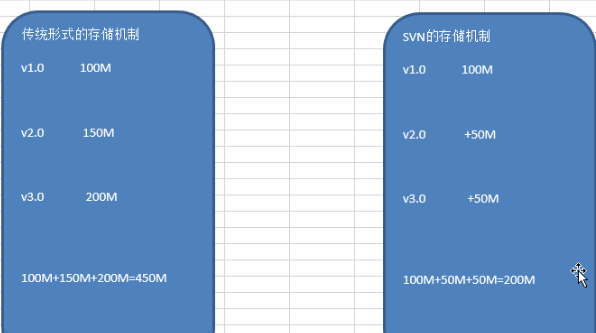
2.远程开发

3.版本控制

工作流程：



版本容量：



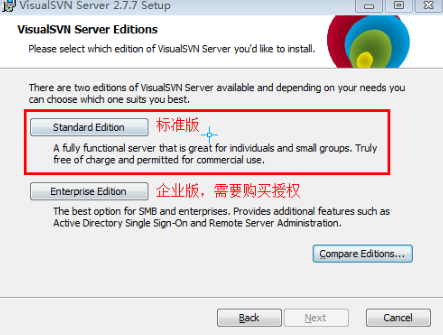
1. **安装**

服务器端安装（Windows版本）：











**Ubuntu安装SVN服务器**

环境：系统版本Ubuntu 16.04.4

1）执行sudo apt-get update命令

2）输入命令svn查看是否已经安装，如下就表示已经安装，则先删除已安装的



删除命令：sudo apt-get remove --purge subversion

3）安装

sudo apt-get install subversion

4）创建仓库文件夹

mkdir -p /usr/local/svn/repository

5）更改repository的权限

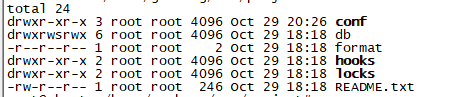
chmod –R 777 /usr/local/svn/repository

6)创建版本库

svnadmin create /usr/local/svn/repository

创建成功后会在repository目录下生成如下文件

ls -l



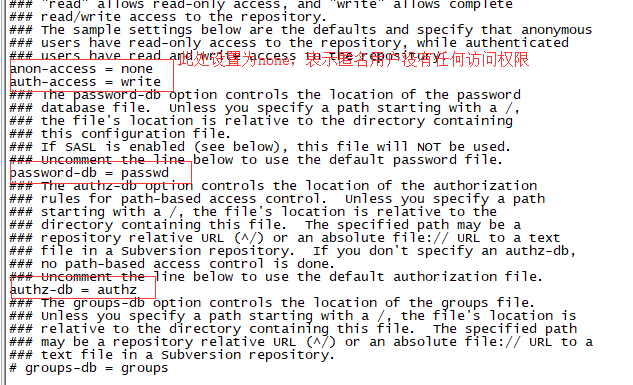
7)对db文件夹修改权限

chmod -R 777 db

8）设置访问权限：需要对conf文件夹下的文件进行修改

vim /usr/local/svn/repository/conf/svnserve.conf

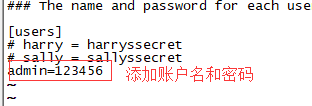
修改点如下图红圈所示



注：去掉前面的#，并且顶格，不然可能会报错

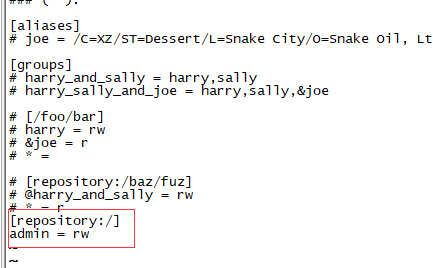
修改passwd文件，添加访问用户

vim /usr/local/svn/repository/conf/passwd



修改authz文件，设置用户权限

vim /usr/local/svn/repository/conf/authz



[repository:/]也可以换成[/]表示/根目录下的所有文件都可以访问

9）启动服务器

svnserve -d -r /usr/local/svn

注：此处注意，之前我由于使用命令

svnserve -d -r /usr/local/svn/repository，检出代码时就会报权限错误，这里一定是svn目录，不能是repository目录。

用如下命令查看svnserve是否已经运行

ps aux | grep svnserve



10)测试是否配置正确

svn co svn://127.0.0.1/repository --username admin --password 123456

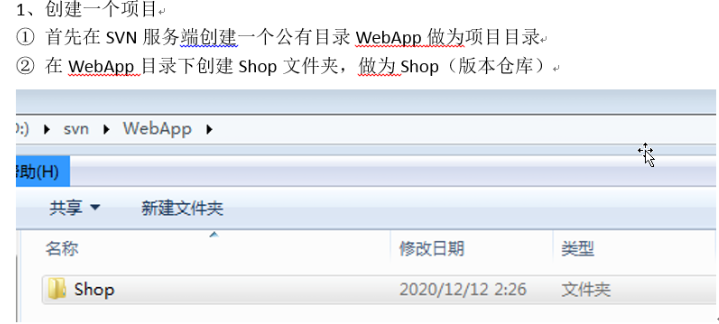
其他机器远程访问

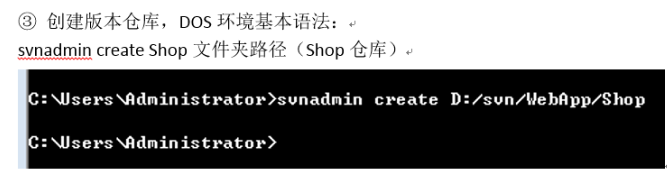
需要我们开放svn端口：ufw allow 3690

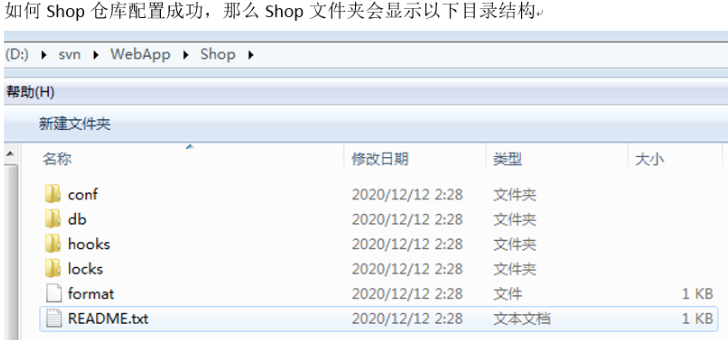
在其他机器就可以通过这个地址下载同步代码

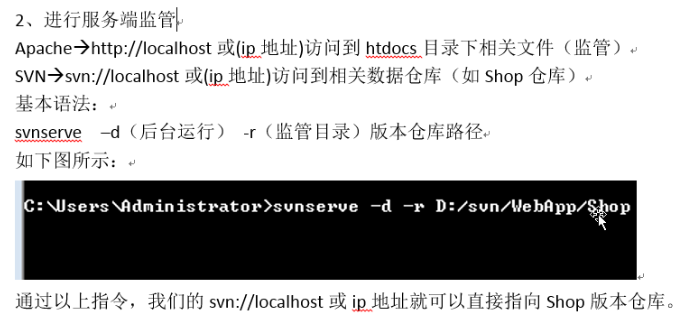
svn://192.168.229.130/repository

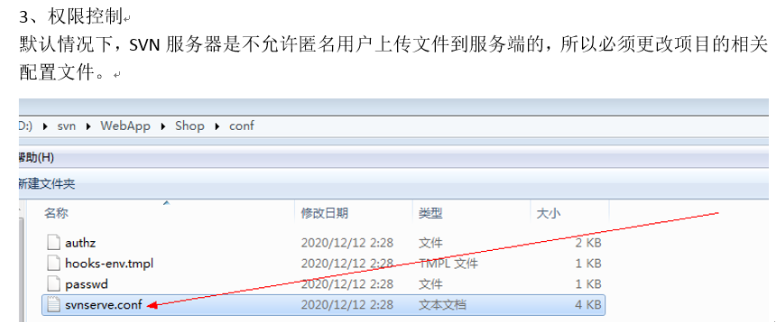
**Svn服务端配置：**

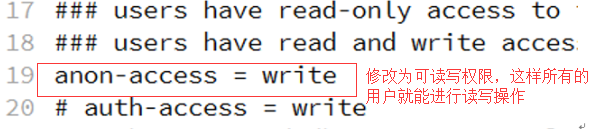


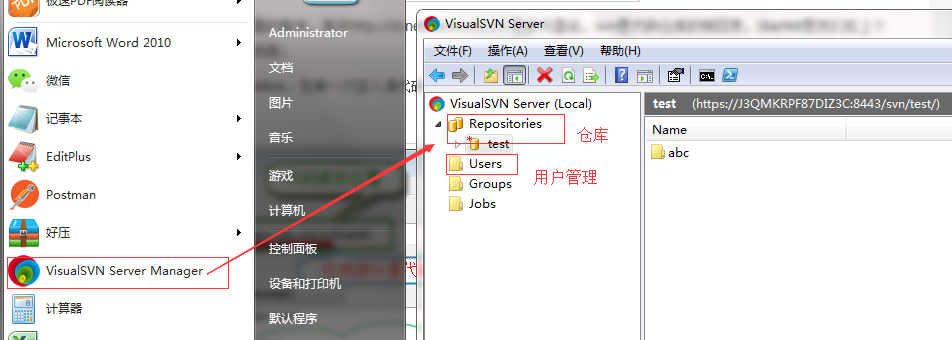


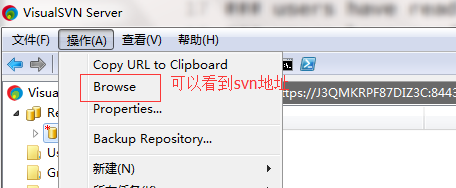


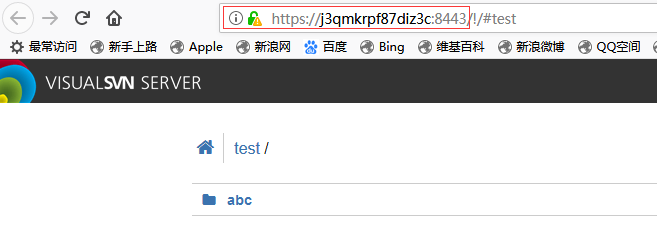


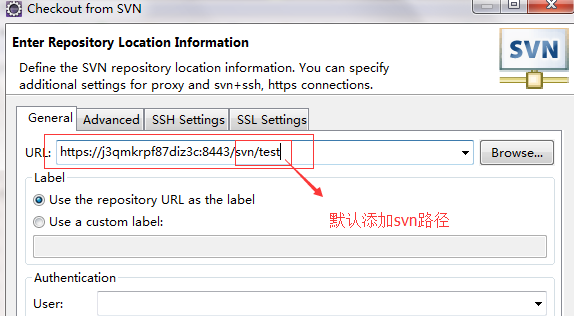












这样就可以把svn代码download到eclipse

客户端安装：

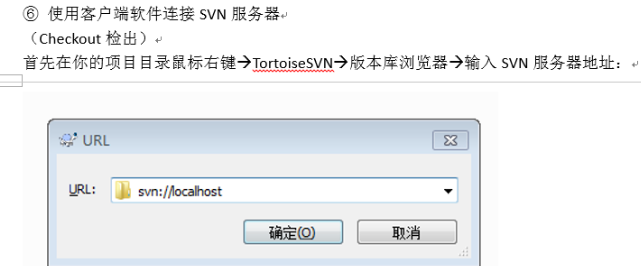


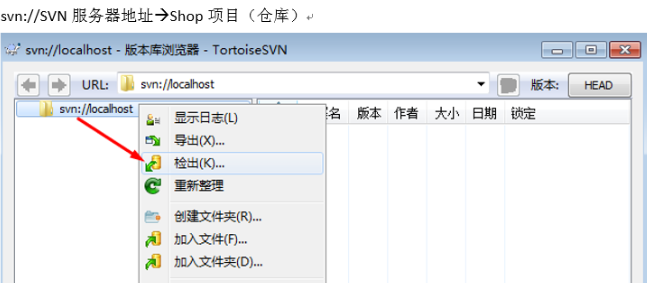


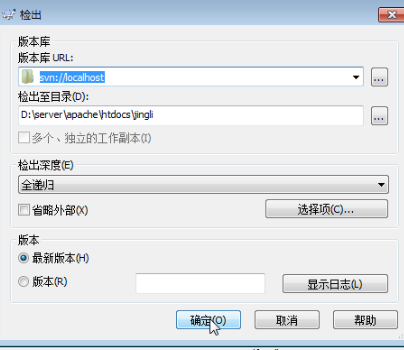
选择默认就行，完成安装

注：在完成客户端软件安装完成后，要重启电脑，要不然是无法使用的

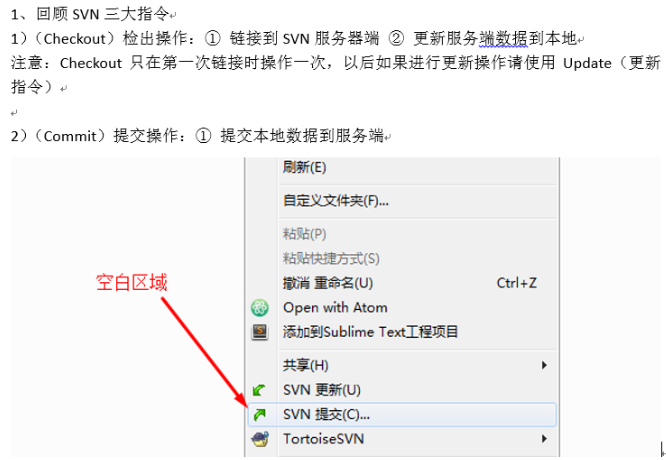




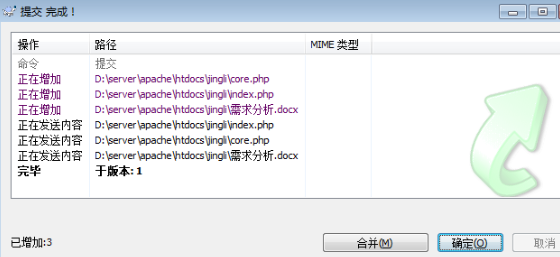




1. **Svn命令**

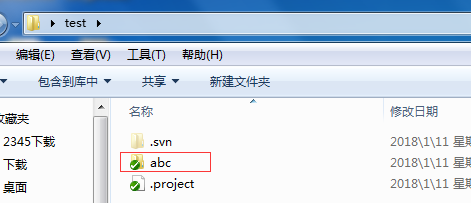






方式2：进入我们checkout后代码的文件位置，比如我们的代码放在test文件夹下，则进入test文件夹下，用命令行进行操作

我们的目录结构如下：



svn diff –r 9:8

查看版本9和版本8的区别

svn log

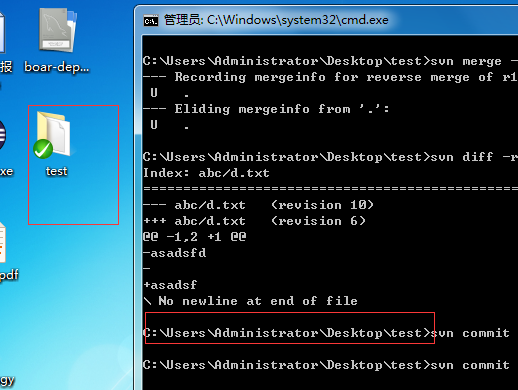
查看svn提交日志信息

Svn merge –r 9:8 abc

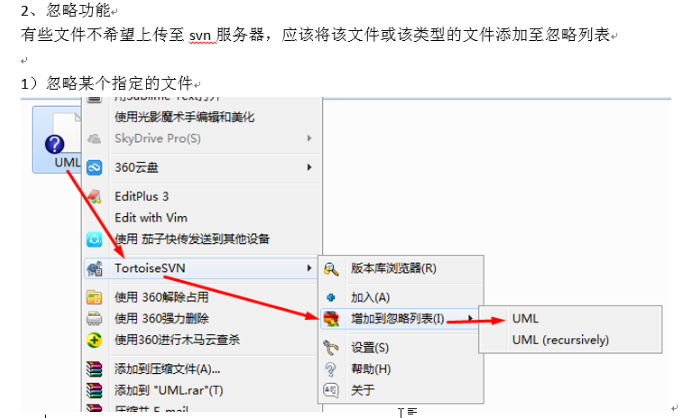
反向合并版本9和版本8,abc指需要合并的文件或者文件夹

Svn up 或者svn update 更新到最新版本

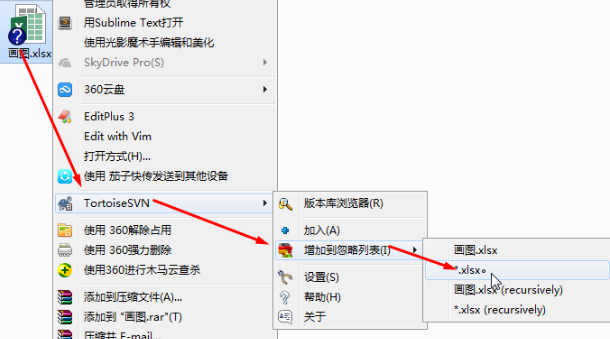




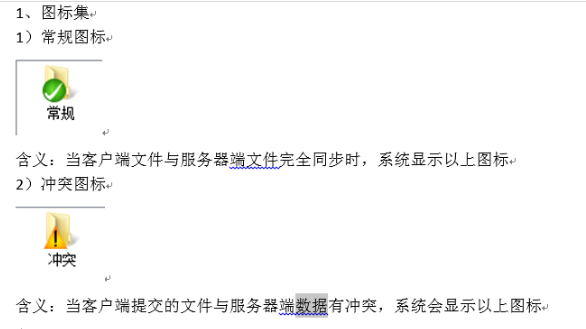
1. **Svn忽略功能**

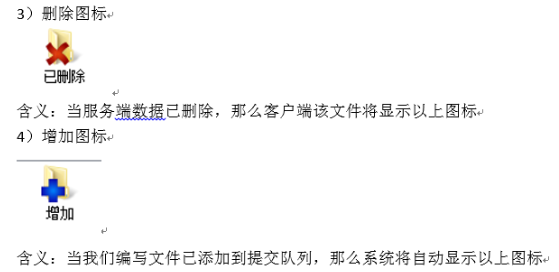


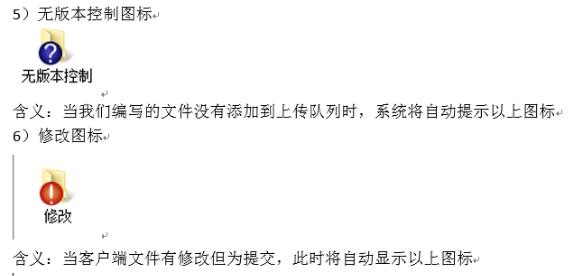
忽略某个类型的文件

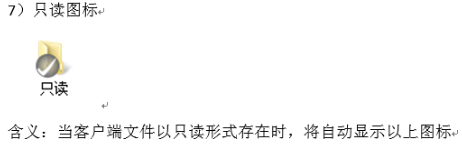


1. **svn图标含义**









锁定图标

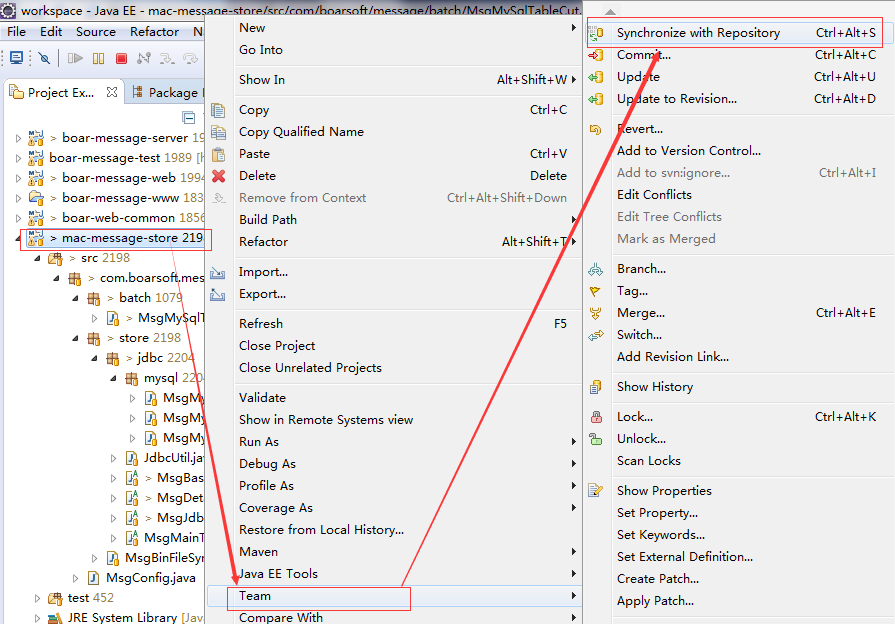


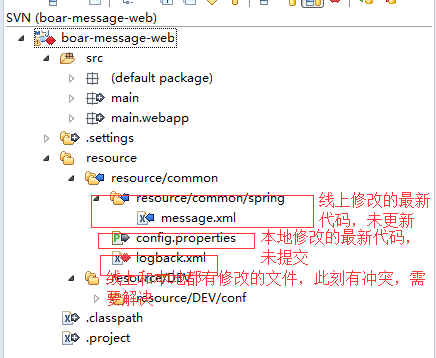
1. **SVN提交代码（eclipse）**

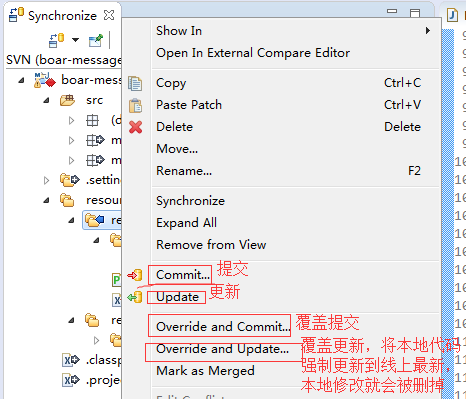
Svn在提交代码之前，最好进行同步，可以看到线上最新代码和本地的区别，特别是有冲突的时候，这样才知道自己修改的地方和线上的区别，易解决冲突。

同步解决冲突，无冲突则先更新，再提交代码

**注：提交代码时，最好不要提交更代码无关的东西，比如编译后的文件，.project文件等等，这些都不要上传**





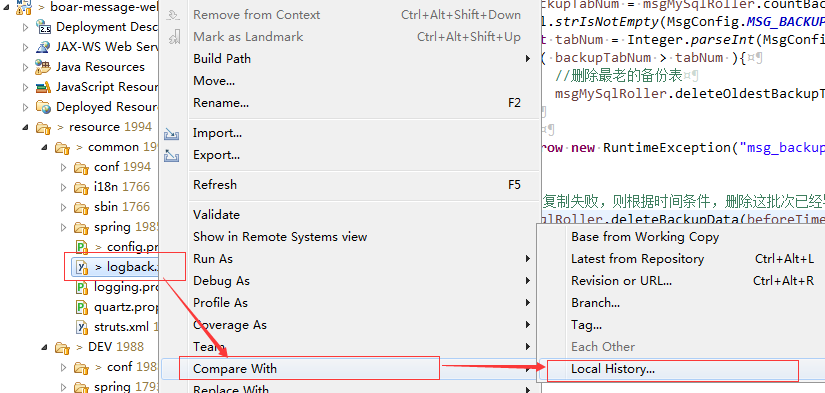


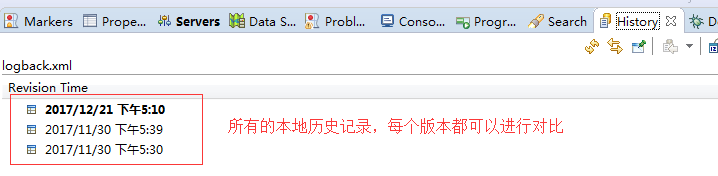
注：Mark as Merged 这里是在冲突解决后（也就是将代码合并后），将此文件标记为合并，则该文件就可以进行提交，要不然不能直接commit

1. **比较代码**

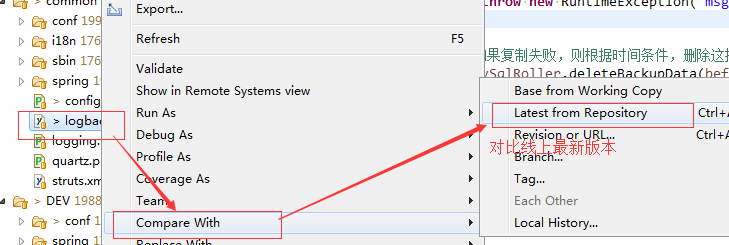
分三种情况，对比本地历史记录，对比线上版本，对比不同SVN地址的代码

对比本地历史记录：

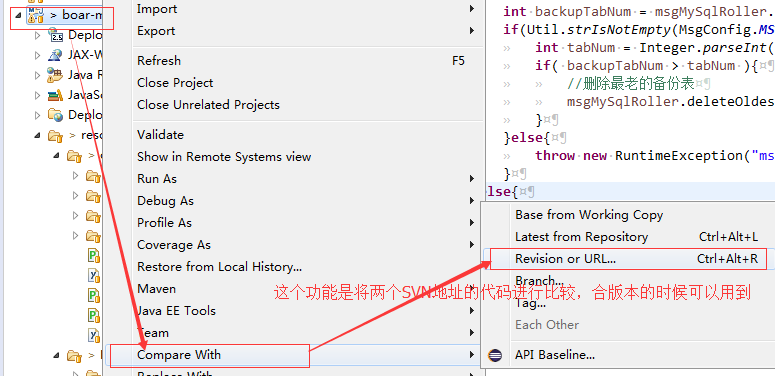


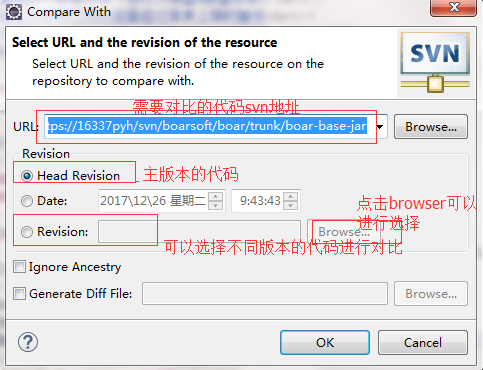


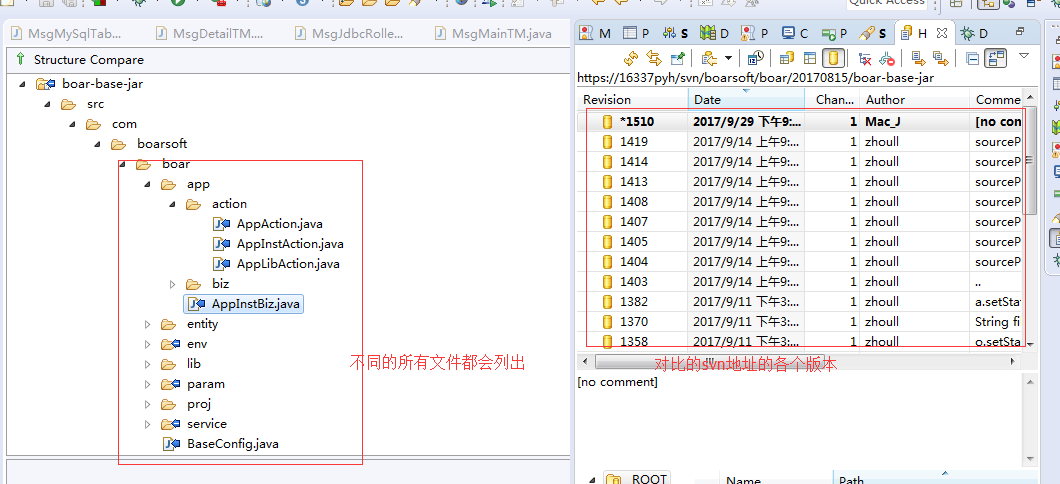
对比线上版本：



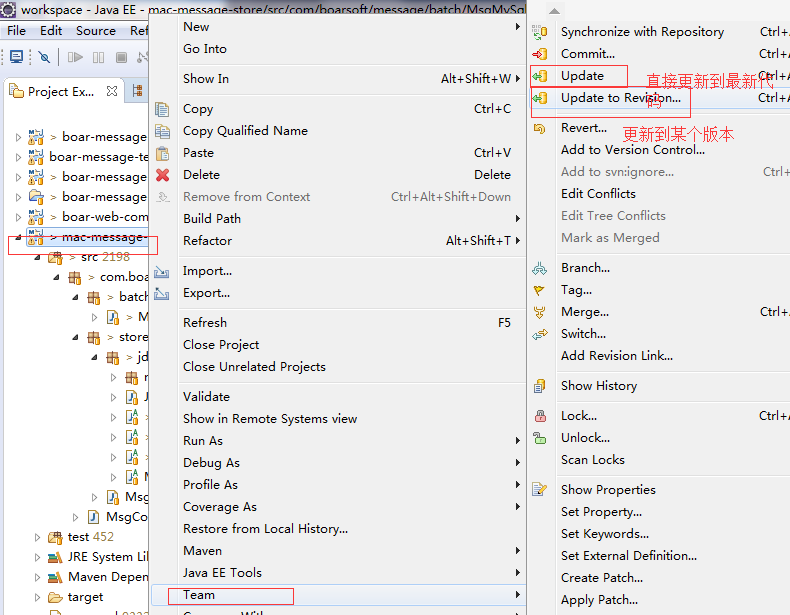
对比不同地址：





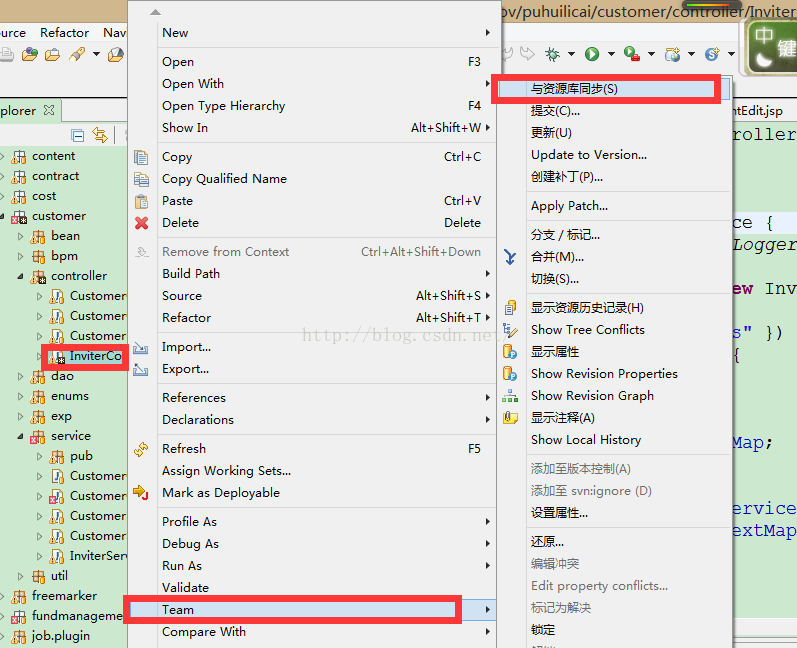


1. **更新代码**

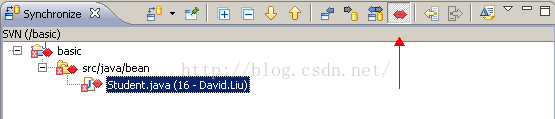


1. **Eclipse中解决svn冲突**

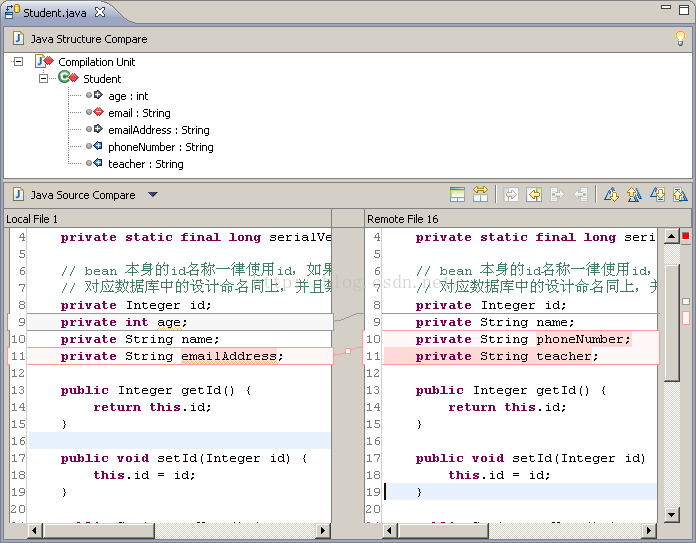
右键项目 -> Team -> 与资源库同步



 在同步视图中选择Conflicts Mode，以便首先查看解决冲突

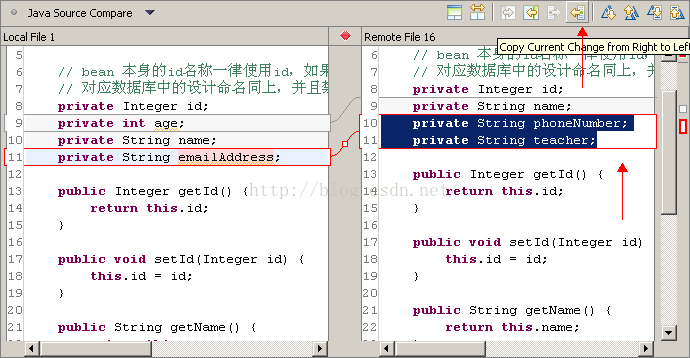


双击需要解决的冲突文件，这里为Student.java，右侧主显示区会出现比较视图。左侧为本地程序，右侧为svn上的最新代码



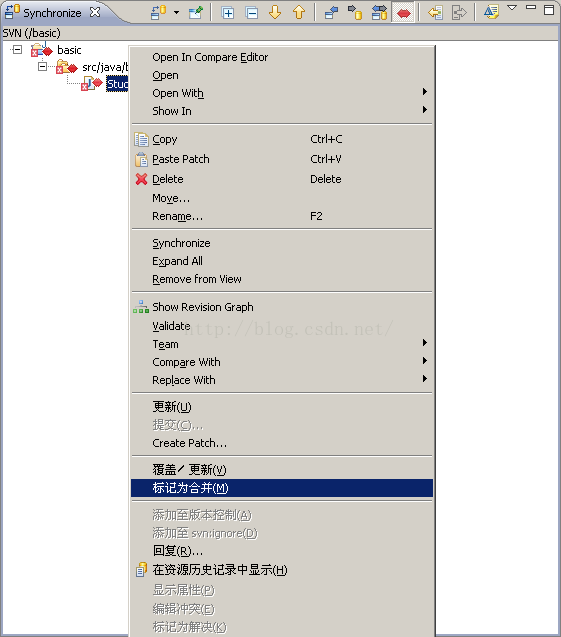
原来字段名为email,本地将其改为emailAddress,线上代码显示有人将原来的email改为了phoneNumber并增加了属性teacher

选中右侧的两行，点击图标拷贝到本地，保存

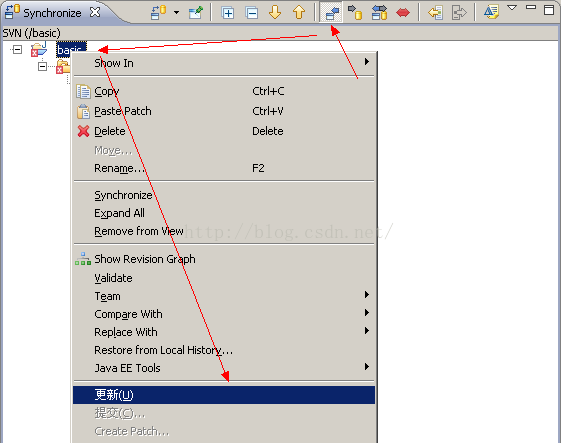


此时线上的代码都已经拿下来了，可以继续执行下一步。

右键冲突文件，选择“标记为合并” Mark as merged

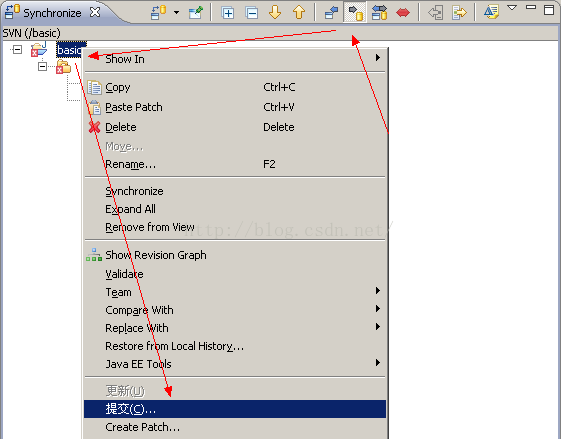


点击Incoming Mode -> 右键项目 -> 更新（更新线上的其他无冲突的代码）

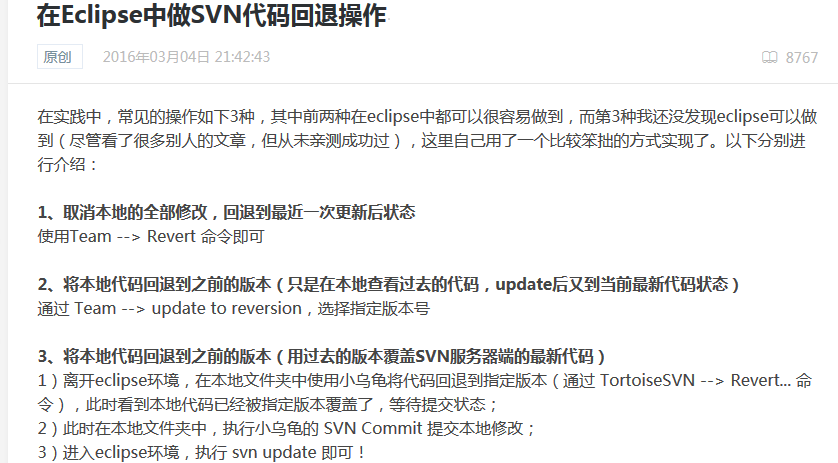


点击Outgoing Mode -> 右键项目 -> 提交。输入注释确定提交。

 经验：在提交前最好将视图中列出的（所有修改过的）文件双击查看审核一边，确保没有错误或者临时的代码。这一步常常会检查出问题。



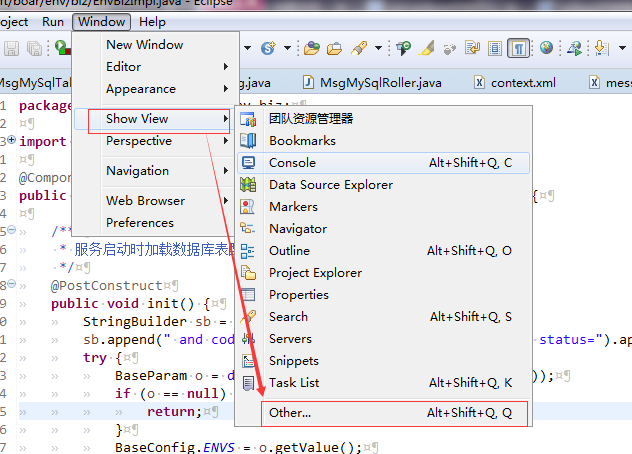
1. **Svn代码版本回退**

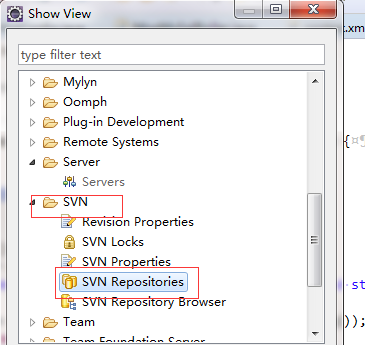


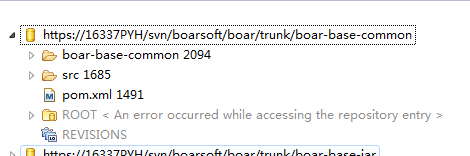
**Eclipse可以实现：**

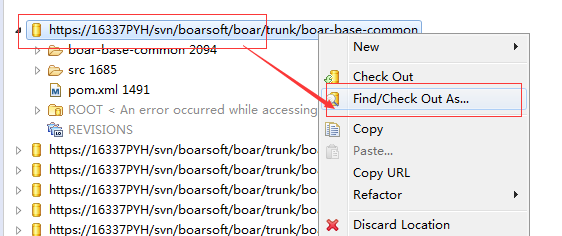
有时候，eclipse中提交代码了，但是发生错误或者想修改，回到某一个历史的版本代码

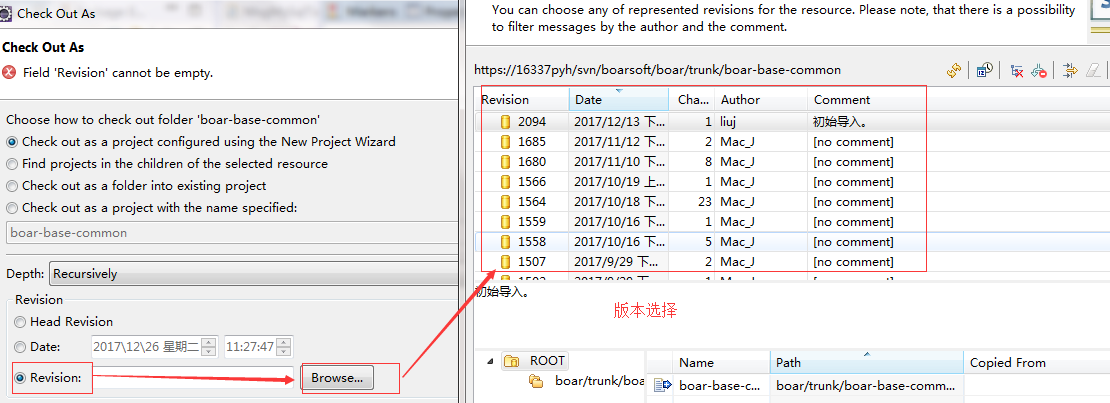
方法1：这种方法是直接把本地代码还原到某个版本，版本号也跟着变，这种方法不能提交代码



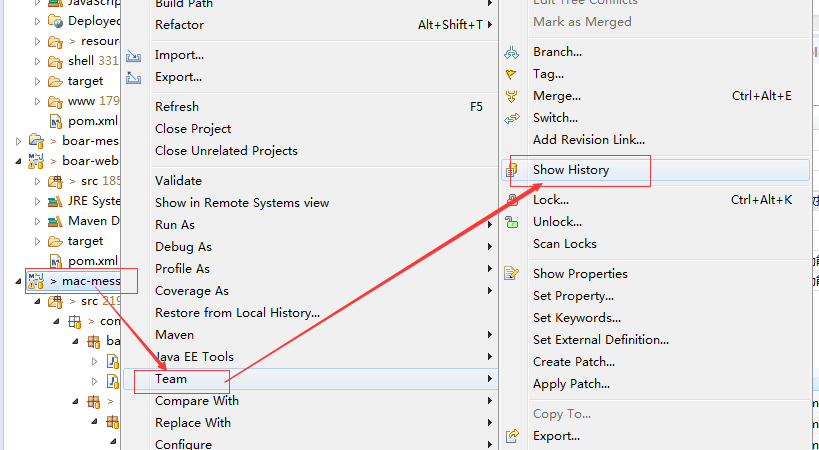


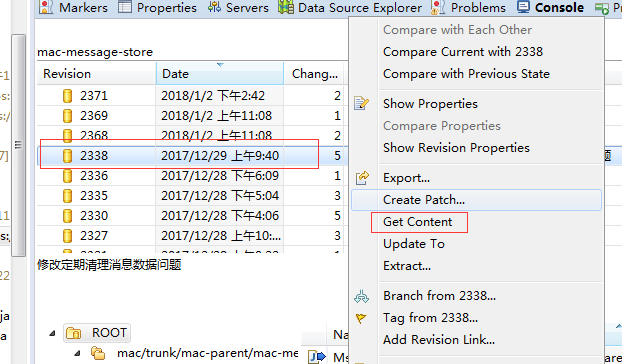






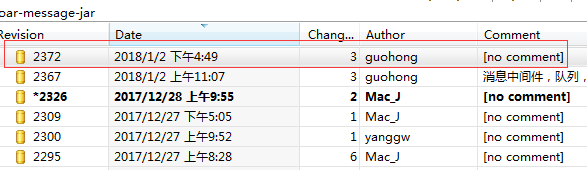
方法2：（可行）





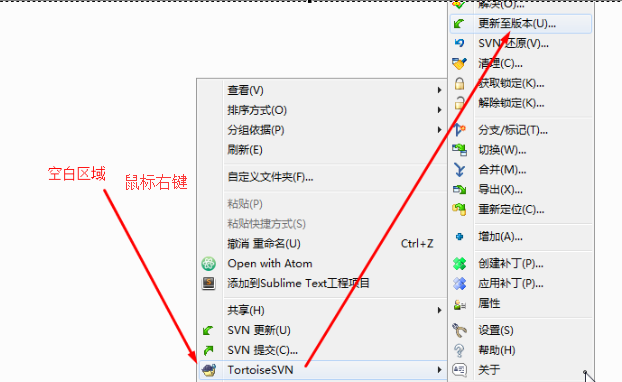
然后再提交，commit。

此时会多一个版本号，该方法对于上次新增的文件修改完全不起作用。也就是比如你这次提交了新增的文件，用该方法get content到上一个版本，但是新增的文件并不会被删掉，需要手动删除。不推荐使用该方法

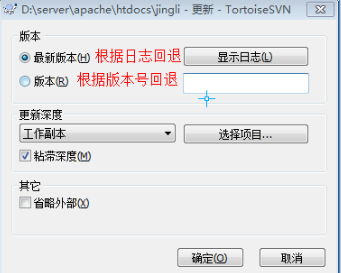


**Windows客户端回退：**

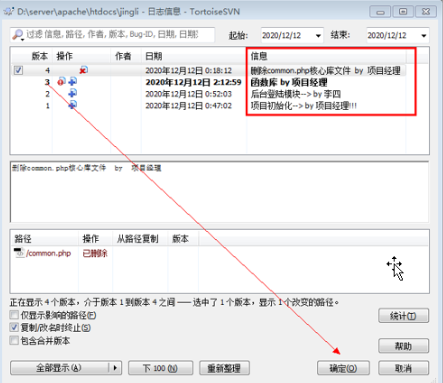
**注：只能回退本地代码到某版本，并不能修改服务器代码版本**



**版本号有时不全，但是日志是全的，因此日志回退能回退的版本更多**

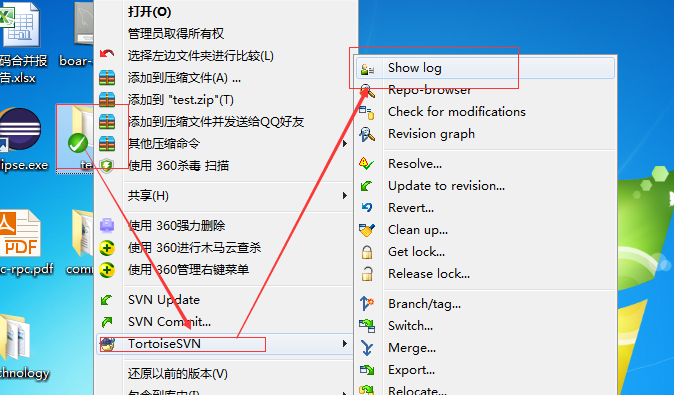


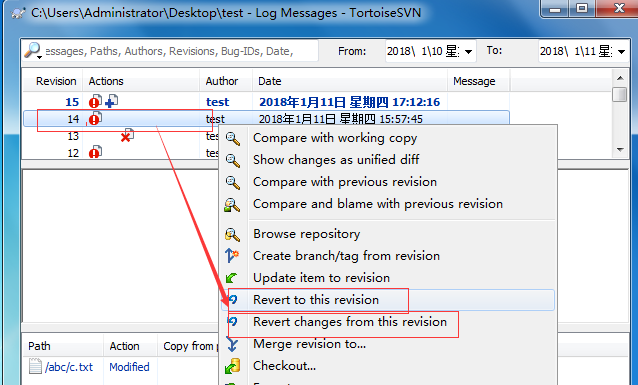
根据日志信息选择要回退的状态，效果如下：



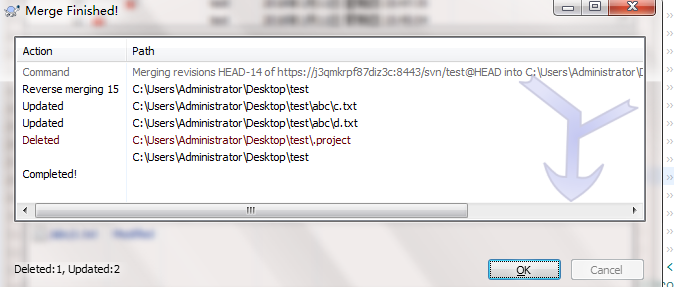
注：客户端直接回退服务器的代码理论上是做不到的，只能通过如下步骤实现版本回退

第一步：

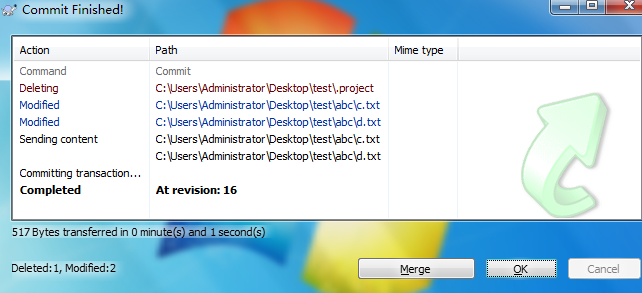




选择红圈中的两个都行，我一般选择第一个



然后直接commit，可以看到svn库多了一个版本号

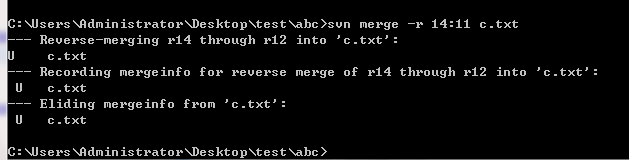


这种方法可以勉强实现版本回滚

命令行回退：

svn merge -r 14:11 c.txt

将c.txt文件从版本14回退到版本11

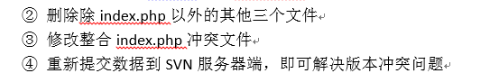


然后进行提交就可以了

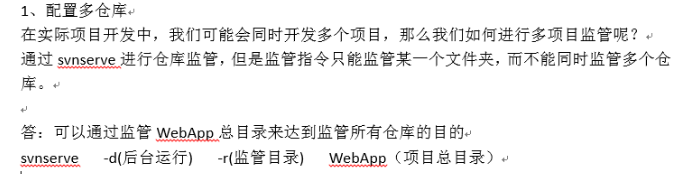
注：这个命令只对文件有效，对文件夹无效

1. **Svn客户端解决冲突**



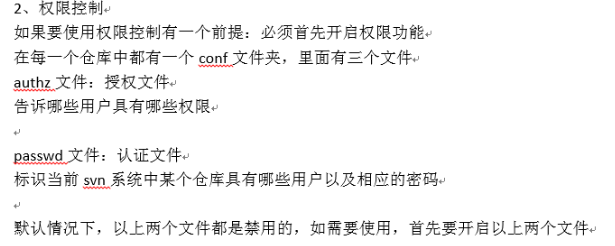


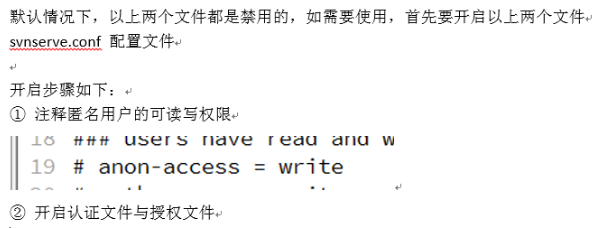
1. **配置多仓库**

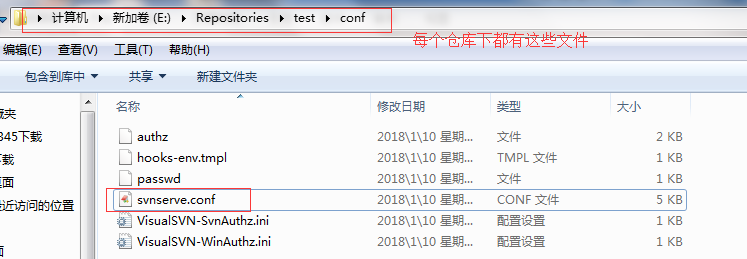


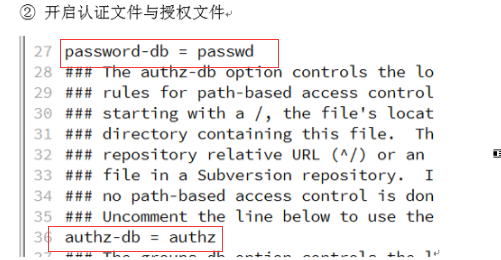


1. **权限控制**

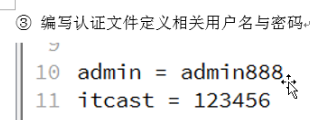




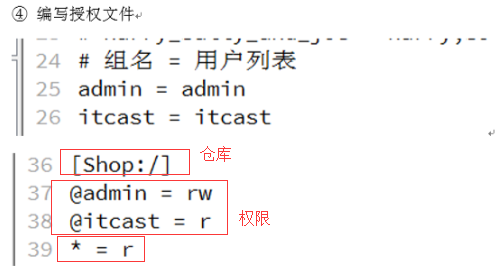




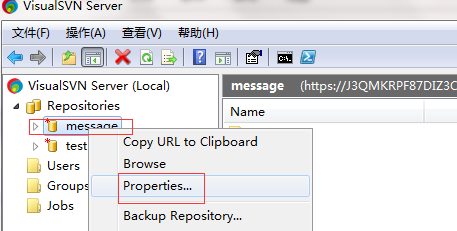
修改文件passwd新增如下信息

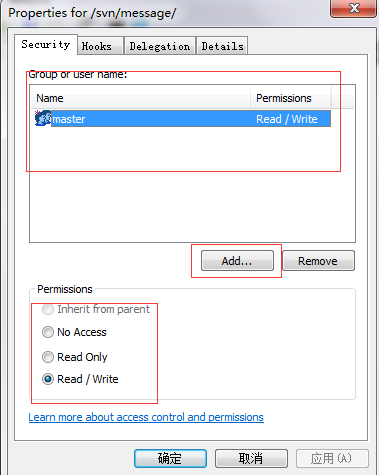


修改authz文件，新增如下内容



还可以通过svn管理器进行设置

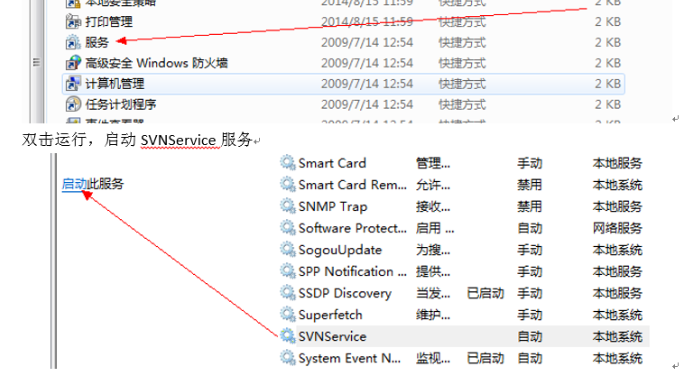




1. **Svn服务的配置与管理**







指令

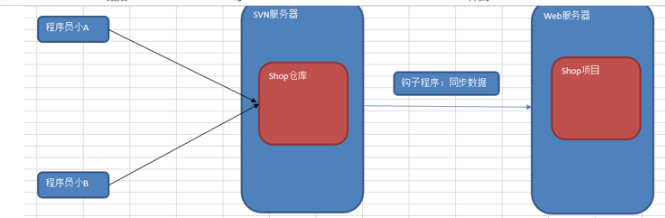
停止服务

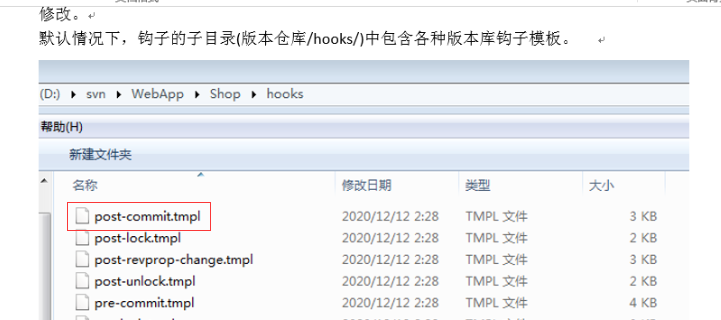


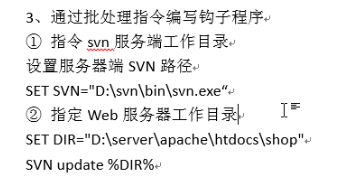


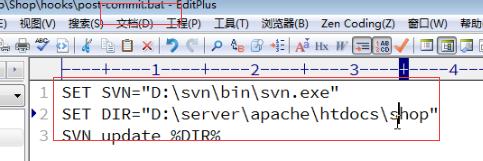
1. **钩子程序，实时更新**

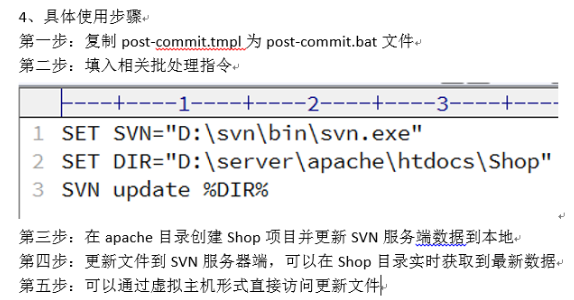
问题描述：我们提交代码到svn上，然后需要把代码部署到web服务器，svn代码经常更新，需要实时将修改的svn代码同步到web服务器上，这个时候就需要钩子程序





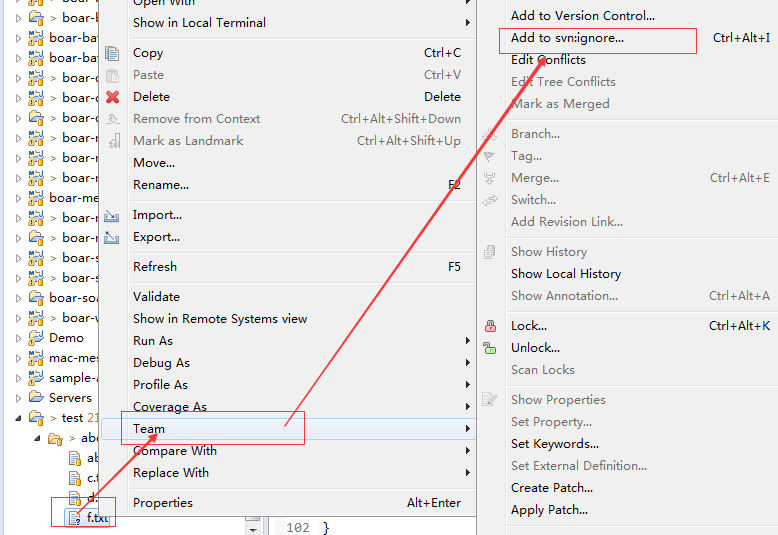


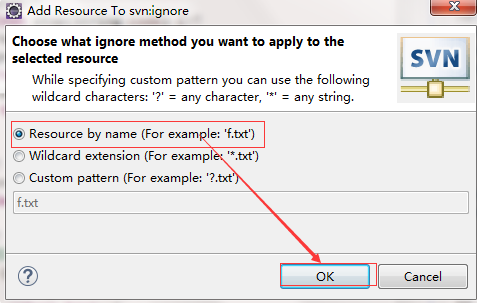




1. **Svn忽略某种文件（不加入到版本库控制）**

有些时候我们编写的某些文件并不想上传到svn库，比如自己的设计文件，或者工程的编译文件，这些对实际项目代码并没有用，因此这个时候需要用到svn的忽略功能。





这样f.txt文件就不会被svn库控制，提交和更新以及svn的其他状态都不会有，svn将忽略该文件的修改

与之对应的功能是加入版本控制，也就是将某文件纳入svn的版本控制之下。

加入版本控制

