SVN-版本控制

# SVN概述

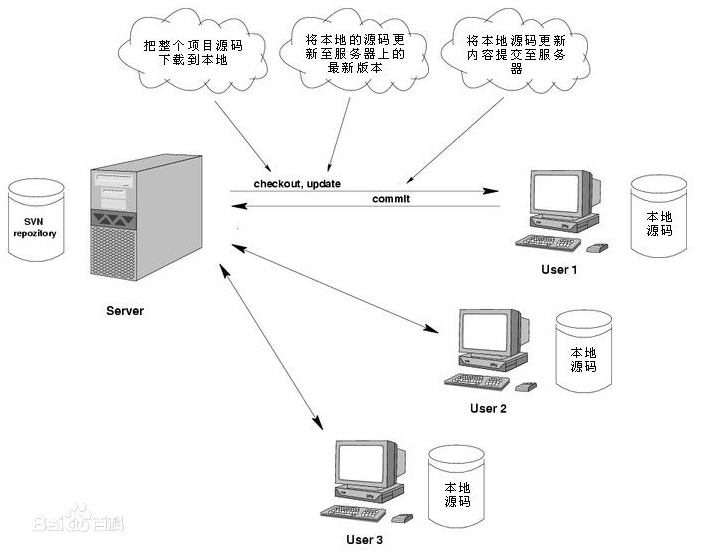
## SVN介绍

SVN是Subversion的简称，是一个自由开源的版本控制系统。

Subversion将文件存放在中心版本库里，这个版本库很像一个普通的[文件服务器](http://baike.baidu.com/view/566090.htm" \t "_blank)，不同的是，它可以记录每一次文件和目录的修改情况，这样就可以借此将[数据恢复](http://baike.baidu.com/view/185060.htm)到以前的版本，并可以查看数据的更改细节

早期版本控制使用的是CVS,后来SVN替代了CVS,随着android兴起，出现Git版本控制工具，后续我们会学到。

## SVN基本概念



问题: 怎样让系统允许用户共享信息，而不会让他们因意外而互相干扰？

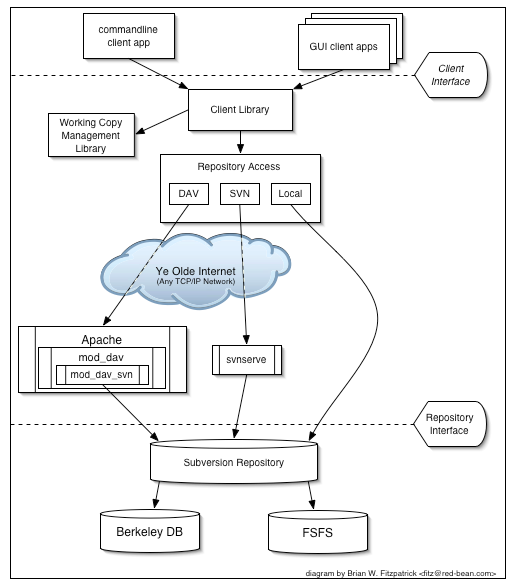
**复制-修改-合并方案(Subversion默认的模式)**

在这种模型里，每一个客户读取项目配置库建立一个私有工作副本——版本库中文件和目录的本地映射。用户并行工作，修改各自的工作副本，最终，各个私有的复制合并在一起，成为最终的版本，这种系统通常可以辅助合并操作，但是最终要靠人工去确定正误。

**锁定-修改-解锁方案**

在这样的模型里，在一个时间段里配置库的一个文件只允许被一个人修改。 此模式不适合软件开发这种工作。

## SVN架构



Subversion支持[Linux](http://baike.baidu.com/view/1634.htm" \t "_blank)和[Windows](http://baike.baidu.com/view/4821.htm" \t "_blank)，更多是安装在Linux下。

svn服务器有2种运行方式：[独立服务器](http://baike.baidu.com/view/2106199.htm" \t "_blank)和借助[apache](http://baike.baidu.com/view/28283.htm)运行。两种方式各有利弊，用户可以自行选择。

svn存储版本数据也有2种方式：[BDB](http://baike.baidu.com/view/1303712.htm" \t "_blank)(一种[事务](http://baike.baidu.com/view/121511.htm" \t "_blank)安全型表类型)和[FSFS](http://baike.baidu.com/view/1633161.htm" \t "_blank)(一种不需要数据库的[存储系统](http://baike.baidu.com/view/51839.htm" \t "_blank))。因为BDB方式在服务器中断时，有可能锁住数据，所以还是FSFS方式更安全一点。

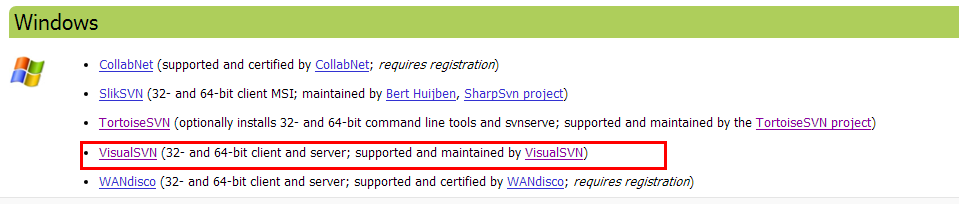
# 使用篇(重点)

SVN是一种集中式管理代码的版本控制系统，原理就是把代码都保存到一个固定的位置（仓库），每次从这个位置拷贝更新代码，进行编辑；再把修改后的代码提交到该目录中。多人协作开发也是如此。因此需要一个类似Oracle或者Mysql的服务器用于保存和管理库文件（要保存的代码等文件）的服务端——VisualSVN Server。还需要一个用户的操作端，用于提交更新检出代码，常用的有Eclipse的Svn插件，以及TortoiseSVN（小乌龟）。

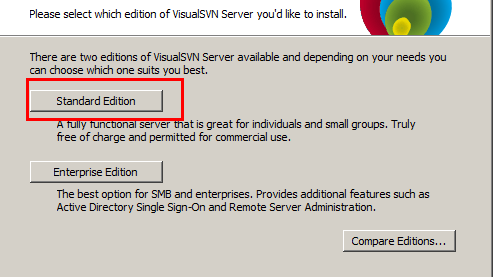
SubVersion 官网：<http://subversion.apache.org/>

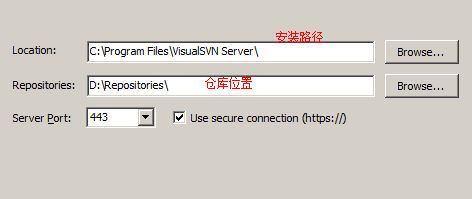
## 服务器端-VisualSVN安装与配置

### VisualSVN下载



### VisualSVN安装与创建仓库



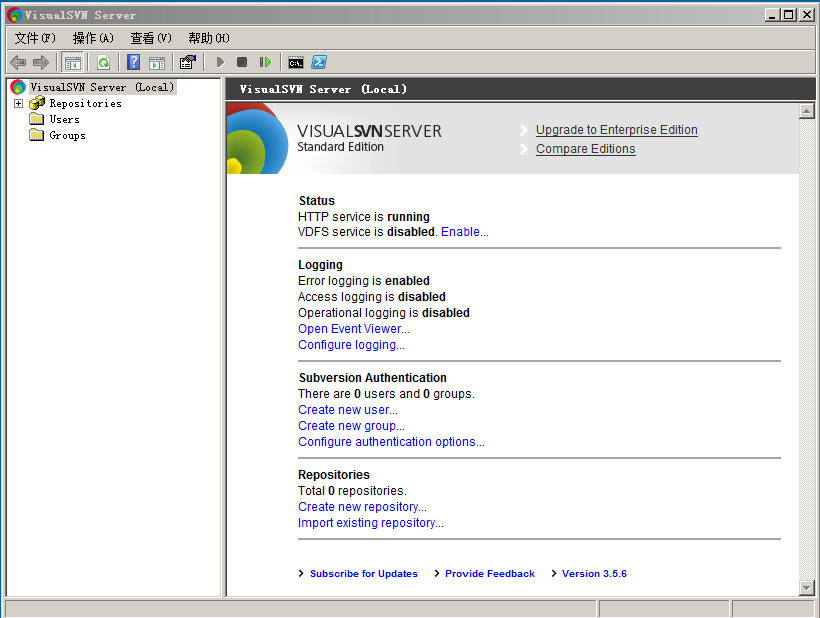


在开始运行中录入 services.msc



这个服务要开启

在开始运行中查找visualSVN Server Manager就可以打开图形化管理界面



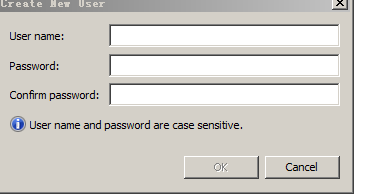
### VisualSVN创建工程目录



### VisualSVN权限控制

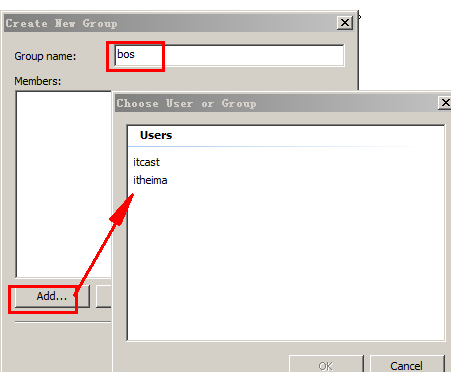
#### 创建用户



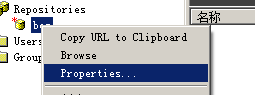


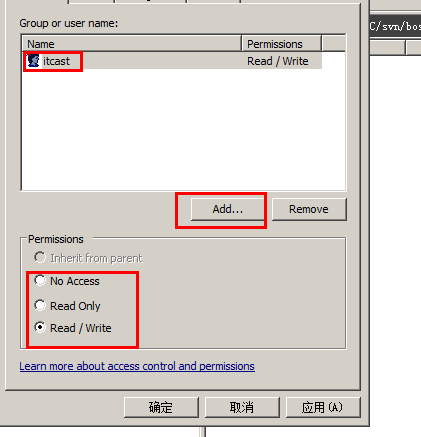
#### 创建组





#### 分配权限





## 客户端-TortoiseSVN安装与使用

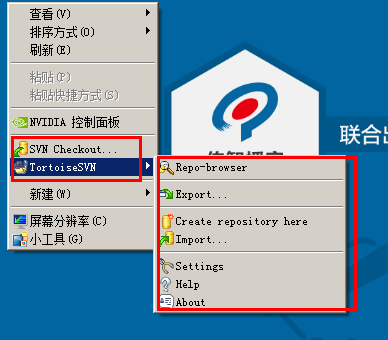
它是一个基于windows系统的svn图形化界面.

### TortoiseSVN下载



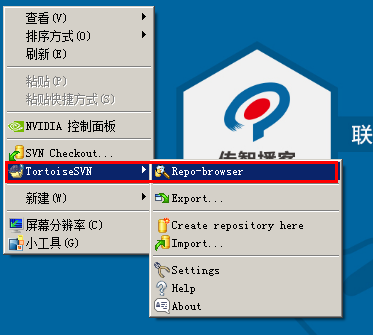
### TortoiseSVN安装

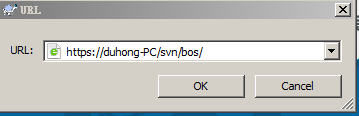
安装后需要重启电脑.



### Tortoise SVN常用操作

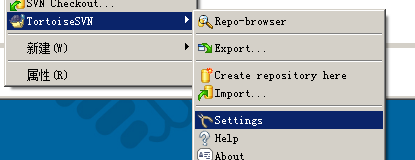
#### 浏览仓库

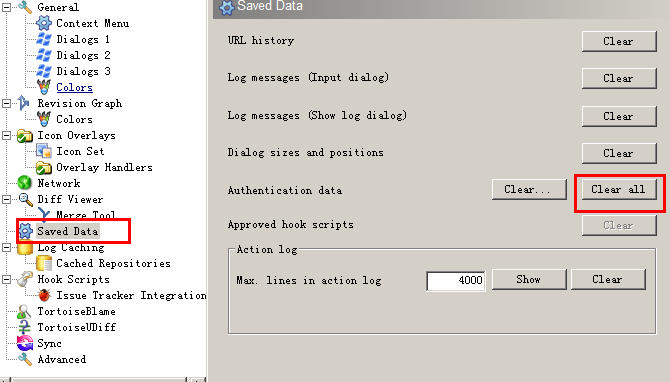


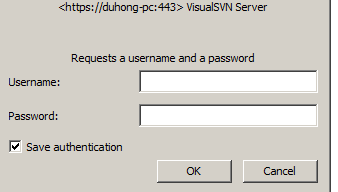


正常情况下，在浏览仓库时会要求录入username与password

我们可以清空svn客户端的缓存

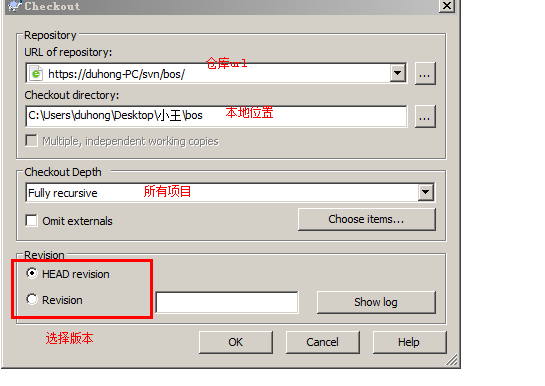






#### Checkout

检出svn服务器上的项目



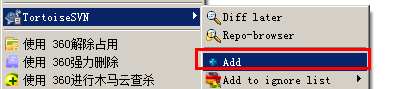
注意：检出后会有一个.svn隐藏目录



#### Add



这样图标的文件代表它没有由svn管理

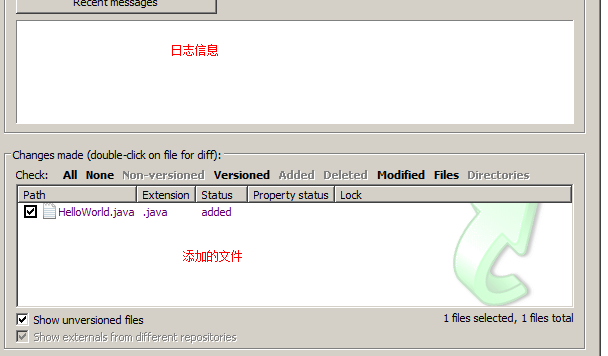


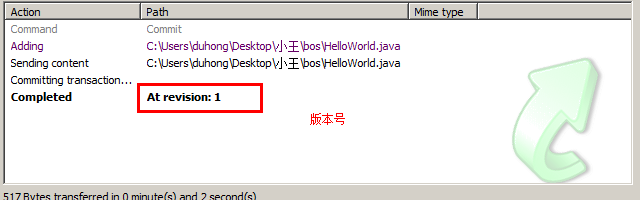


这个代表的是它要添加到svn服务器

#### Commit









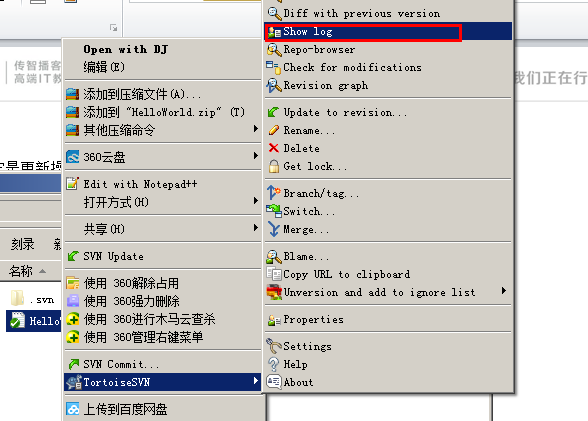
#### Update

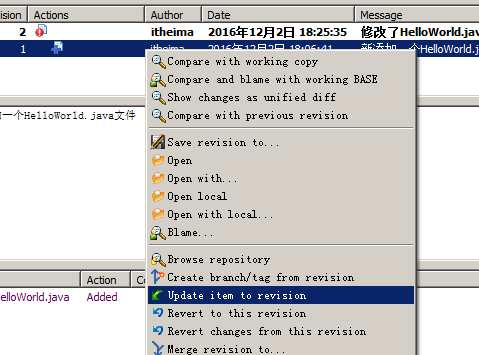
Update，它是更新操作,可以将svn服务器上的内容更新到本地



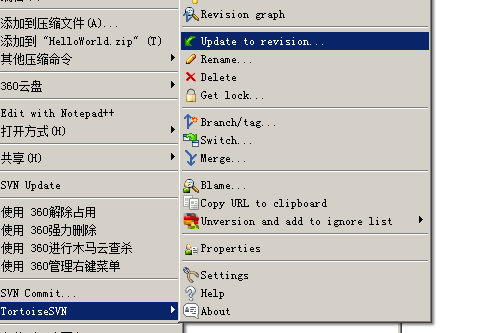
#### 更新到历史版本

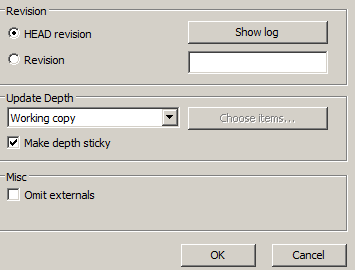
查看当前日志





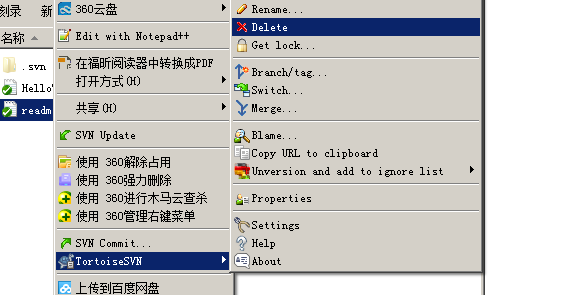
也可以通过下面的方式来更新历史版本



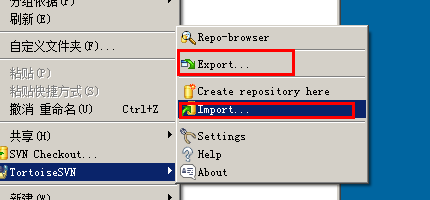


#### Delete

标记删除操作，本地文件会删除，我们需要在它的父目录进行提交，才能真正从服务器删除



#### 导入导出

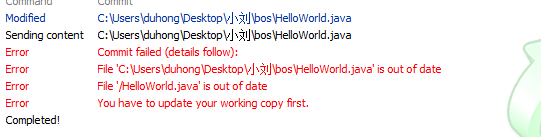


Import将本地资源导入到svn服务器

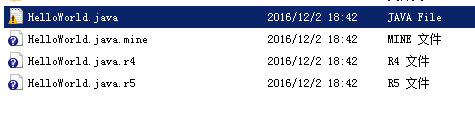
Export导出项目，和checkout的区别,它不存在.svn隐藏文件

#### 冲突问题与解决（重要）

多个用户同时操作同一个文件时，就可能产生冲突情况。



这个冲突产生后如何解决，我们可以采用手动Merge,解决冲突后，重新commit.



HelloWorld.java 它是服务器上的文件与本地文件合并

HelloWorld.java.mine 本地修改后的文件

HelloWorld.java.r4 修改前的文件

HelloWorld.java.r5 对方修改后的文件

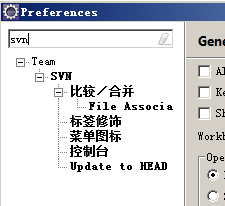


## 客户端-Eclipse SVN插件安装与使用(重要)

### 下载与安装SVN插件

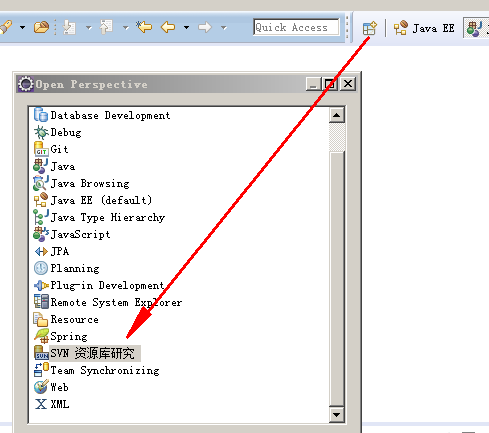
在我们资料中有一个svn插件,交它直接复制到eclipse安装目录下的dropins中

我们怎样可以知道svn插件安装成功



### Eclipse SVN插件使用

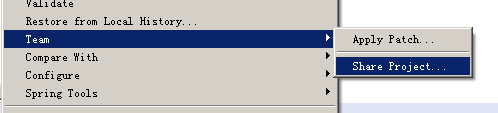
#### 浏览仓库，将eclipse工程导入到svn服务器



在svn资源库研究

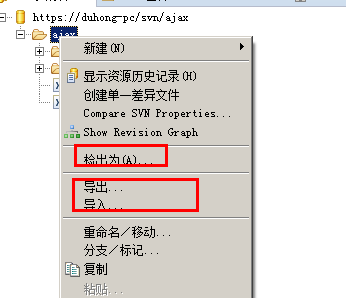


怎样将eclipse中的工程导入到svn仓库，可以在工程上右键

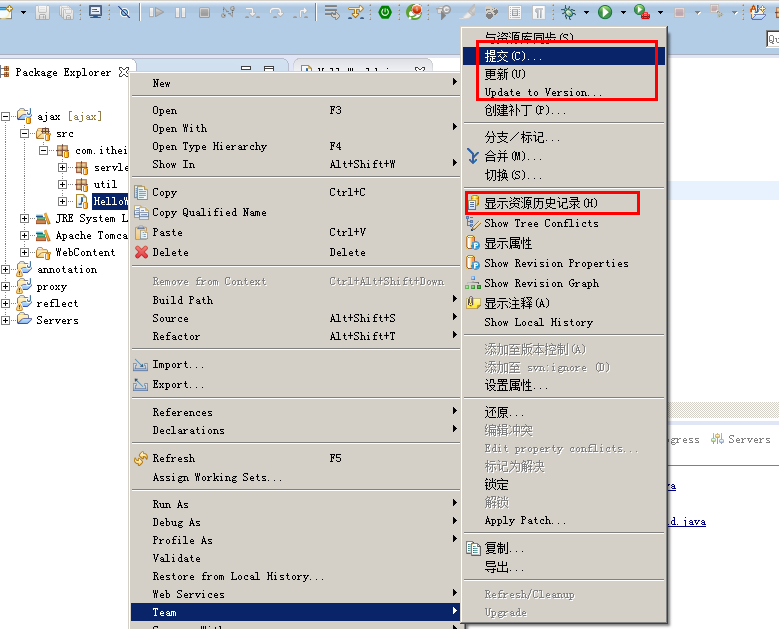


完成后，它其实没有真正的导入，需要我们在进行提交才真正导入到svn仓库

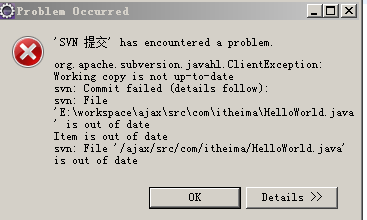
#### Checkout 导出 导入



#### Add commit update 更新到历史版本

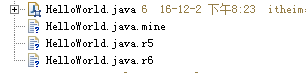


#### Eclipse的svn插件解决冲突



产生了冲突

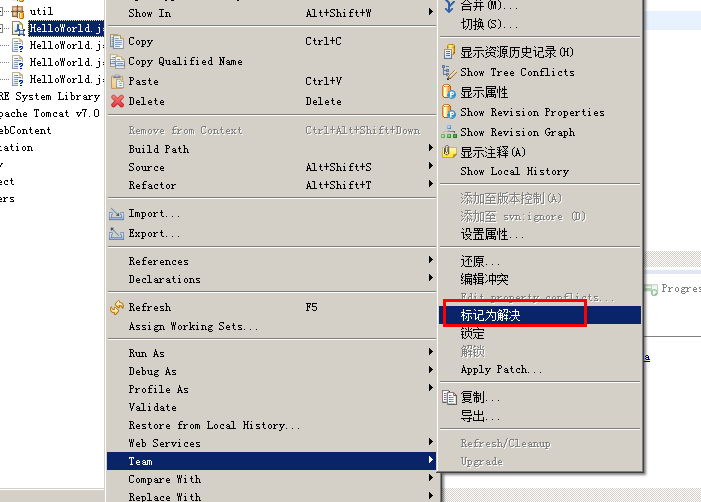
我们在工程上执行更新操作，就可以查看到冲突的文件



解决冲突



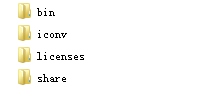
选择编辑冲突



# 理论篇(了解)

## 服务器-SubVersion安装与配置

目录结构



bin目录存放svn相关命令

svnadmin:给服务器管理员使用，创建仓库，帐户等

svnserve:启动svn服务器

svn:客户端命令工具

Iconv目录存放svn依赖插件

Licenses 目录存放版权信息文件

Share存放国际化文件

当安装完成后，默认情况下会将subversion安装目录bin加入到环境变量path中，如果没有，需要手动添加



查看服务器版本号

Svnserve –version

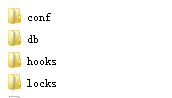


启动svn服务器

1. 在服务器指定目录建立仓库 d:/my-reps

Svnadmin create itcast

查看仓库目录结构



conf 存放当前仓库的配置文件

db 存放仓库版本的控制文件

hooks 仓库脚本

locks 锁文件

1. 启动服务

Svnserve –d –r 路径



1. 测试客户端连接服务器，默认端口3690

Svn checkout svn://localhost/itcast

## 客户端-命令行方式

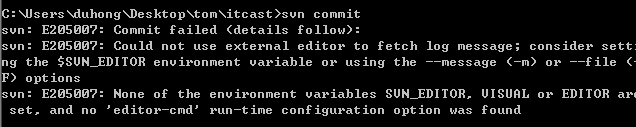
### Checkout

Svn checkout svn://localhost/itcast 默认端口3690

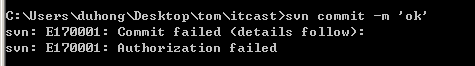
### add



### commit

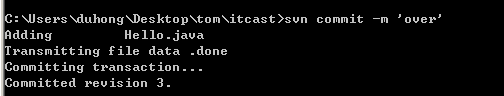


出错的原因:1.没有日志信息

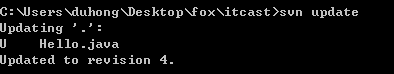


出错的原因:权限问题

我们可以在D:\my-reps\itcast 查找到conf/ svnserve.conf文件



### Update操作

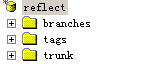


# 扩展篇

## svn目录规范

在visualSVN中创建仓库时，可以选择svn目录结构





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

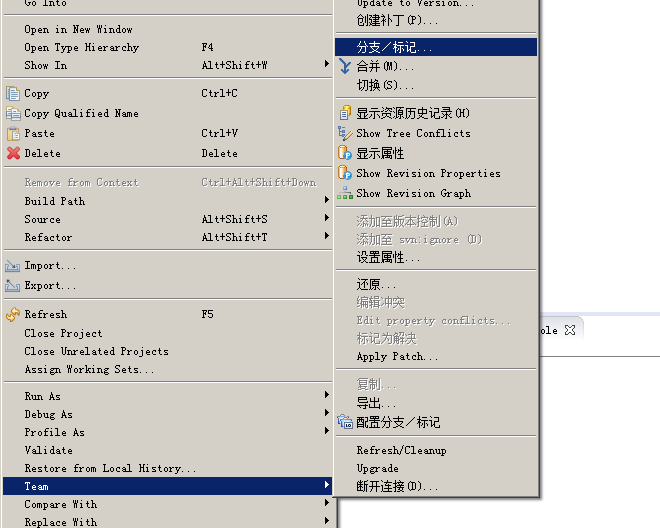
Tags 用于发布的版本目录

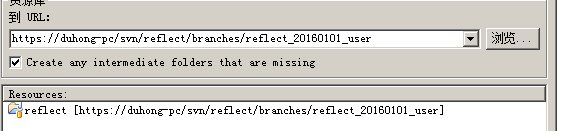
假设有一个项目OA,我们完成了1.0版本，这时就可以打一个tags

后续我们在OA项目上添加一个新的模块(及时通讯),我们就可以开一个分支,又有一个公司需要在我们OA基础上添加财务管理模块，我们又可以打一个分支。

我们后续针对OA的1.0版本在升级，我们不需要原来附加功能,就可以在原来的主干上继续开发，形成OA2.0版本，开发完成后就可以在打一个tags

## 打一个分支或标记





分支的定义规则:

Project name+日期时间+功能点

Tags的定义规则

Project name+版本号

版本号定义为三段数字编号

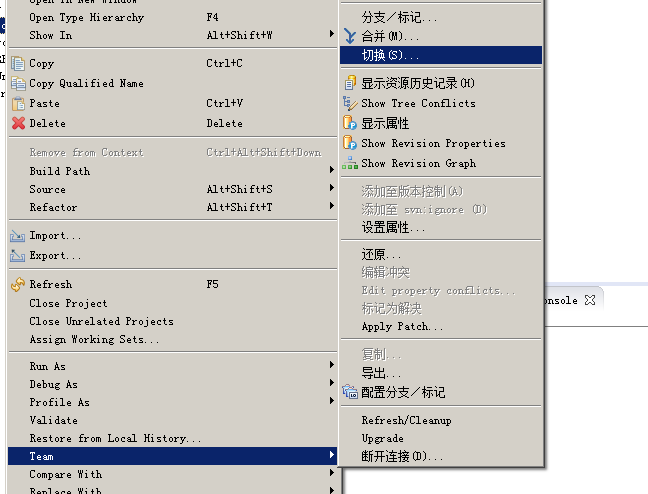
xxx.xxx.xxx

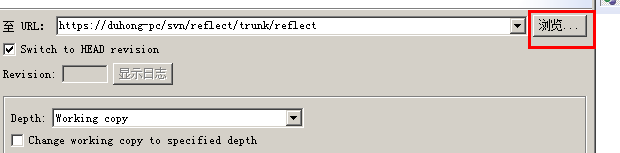
第一个:革命性的产品升级版

第二个:新功能版

第三个:修正bug

切换主干，分支与标记





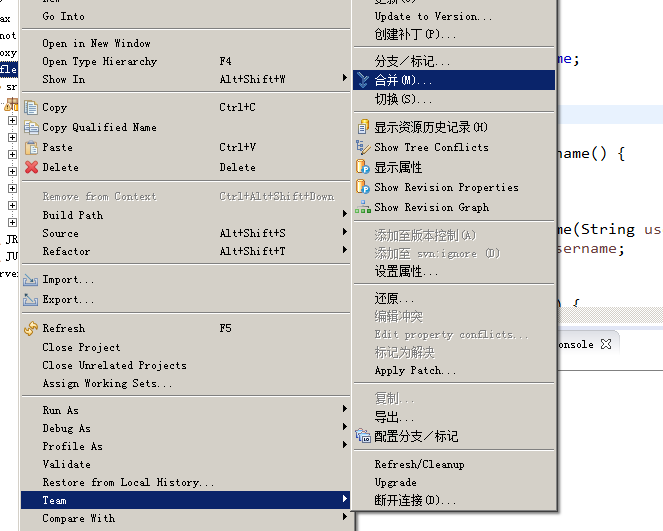
Tags一般是只读，它代表的是发布的版本，所以我们不要进行改变。

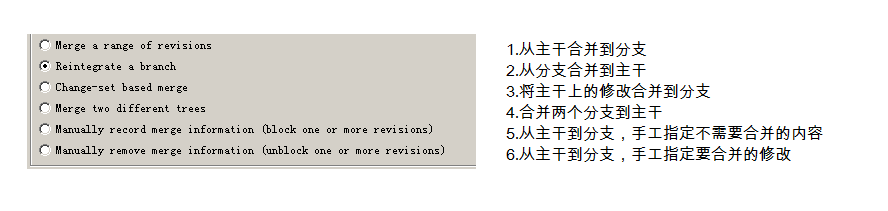
## 主干与分支的合并

如果要将主干内容合并到分支上，我们需要在分支的工作副本上操作。

如果要将分支的改变合并到主干上，我们需要在主干的工作副本上操作。

我们的需要是将分支的改变合并到主干上：





注意:在合并时要选择在相应的版本号，合并后，可能会出现冲突，将冲突解决，commit就可以。