

# ssm练习第一天

## 第一章：SVN相关的概念

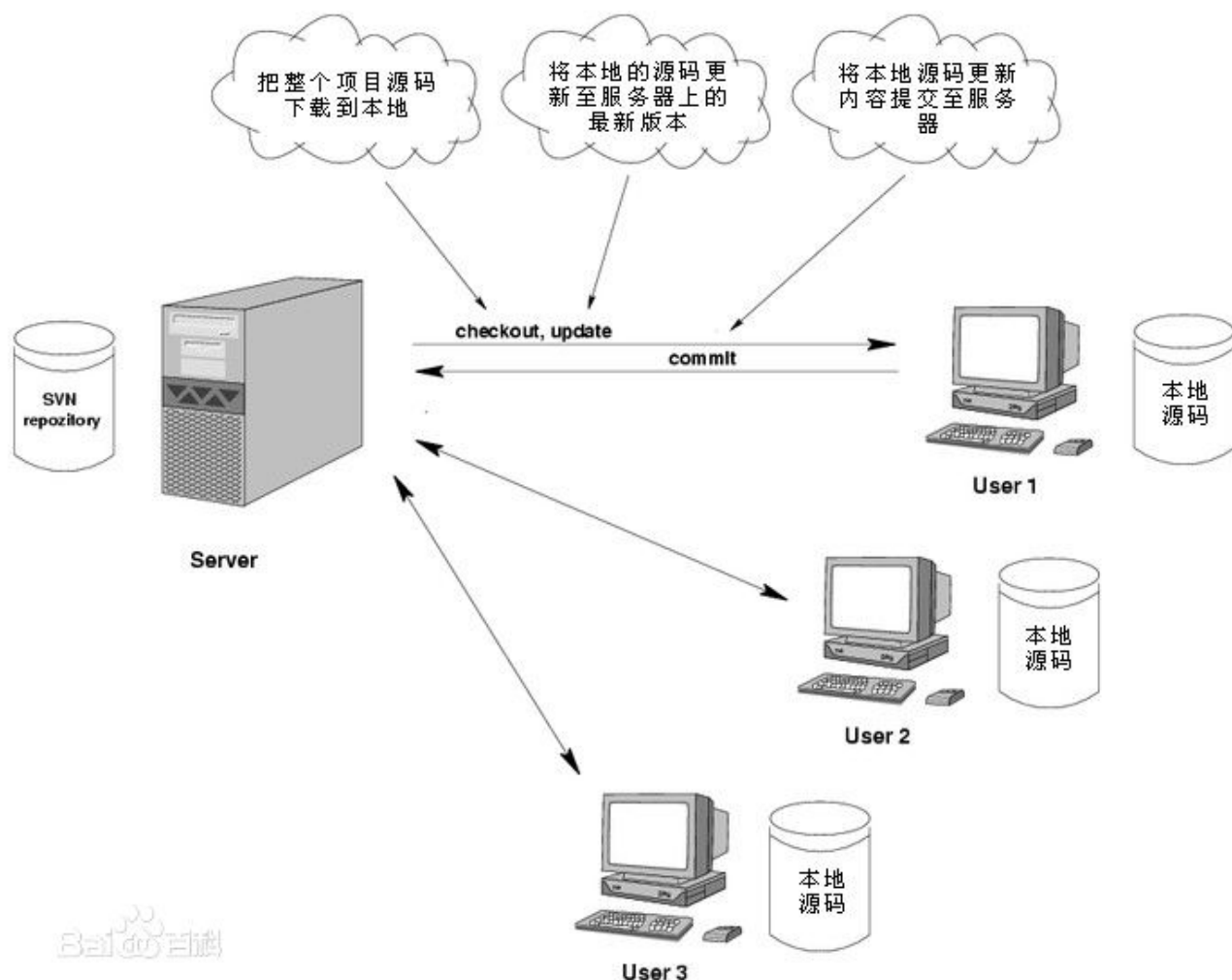
### 第一节 SVN 介绍

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](#)，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](#)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。

### 第二节 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？

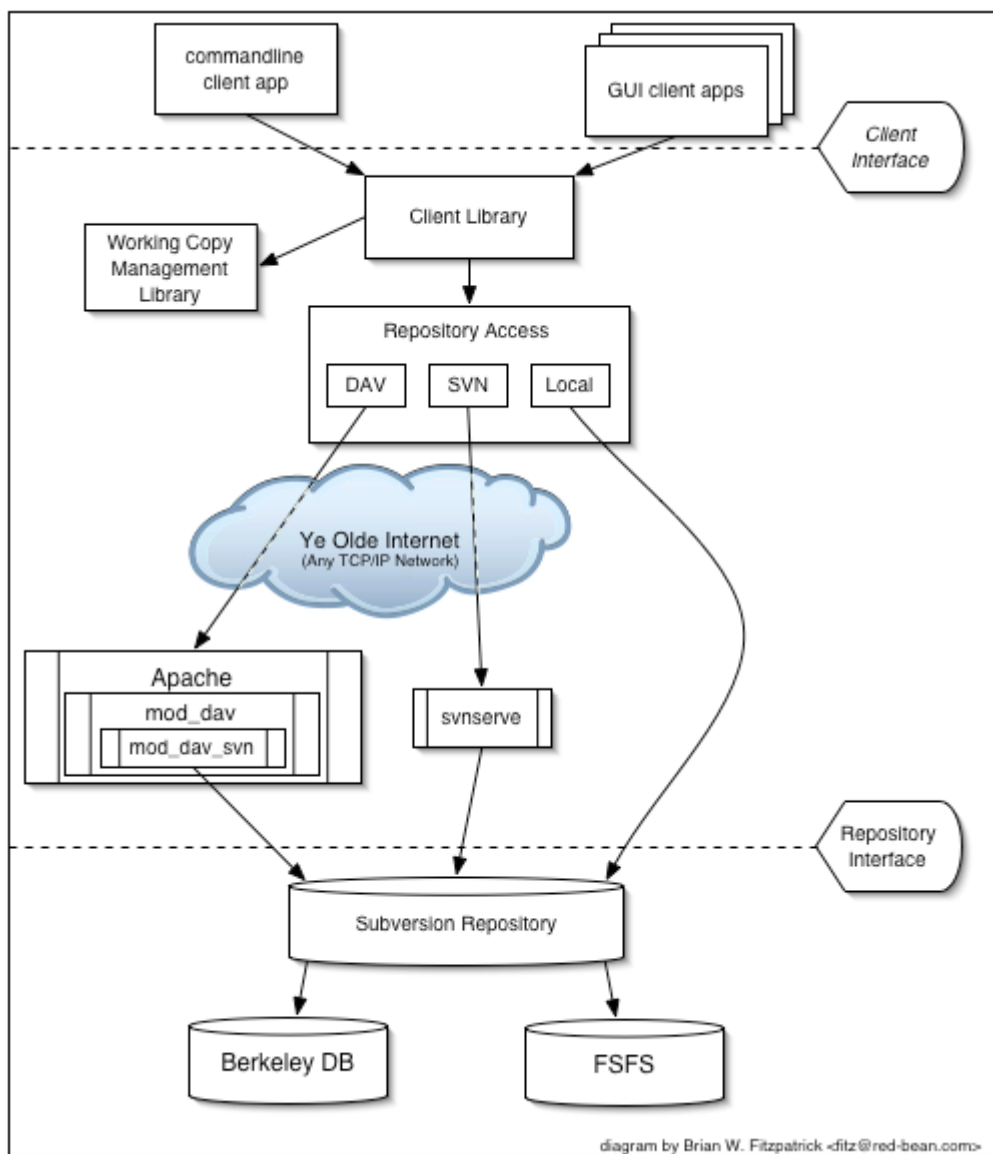
复制-修改-合并方案(Subversion默认的模式)

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。

## 第三节 SVN架构



Subversion支持Linux和Windows，更多是安装在Linux下。

svn服务器有2种运行方式：[独立服务器](#)和借助[apache](#)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](#)(一种[事务安全型表类型](#))和[FSFS](#)(一种不需要数据库的[存储系统](#))。

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。

## 第二章：SVN的安装和使用

### 第一节 服务器端VisualSVN安装与配置

#### 1 VisualSVN下载

##### Windows



- [CollabNet](#) (supported and certified by [CollabNet](#); requires registration)
- [SlikSVN](#) (32- and 64-bit client MSI; maintained by [Bert Huijben](#), [SharpSvn project](#))
- [TortoiseSVN](#) (optionally installs 32- and 64-bit command line tools and svnserve; supported and maintained by the [TortoiseSVN project](#))
- [VisualSVN](#) (32- and 64-bit client and server; supported and maintained by [VisualSVN](#))
- [VANDisco](#) (32- and 64-bit client and server; supported and certified by [VANDisco](#); requires registration)

#### 2 VisualSVN安装与创建仓库

Please select which edition of VisualSVN Server you'd like to install.

There are two editions of VisualSVN Server available and depending on your needs you can choose which one suits you best.

**Standard Edition**

A fully functional server that is great for individuals and small groups. Truly free of charge and permitted for commercial use.

**Enterprise Edition**

The best option for SMB and enterprises. Provides additional features such as Active Directory Single Sign-On and Remote Server Administration.

[Compare Editions...](#)

Location:  [Browse...](#) 安装路径

Repositories:  [Browse...](#) 仓库位置

Server Port:  ☒ Use secure connection (https://)

**一定要选择8443端口号**

在开始运行中录入 services.msc

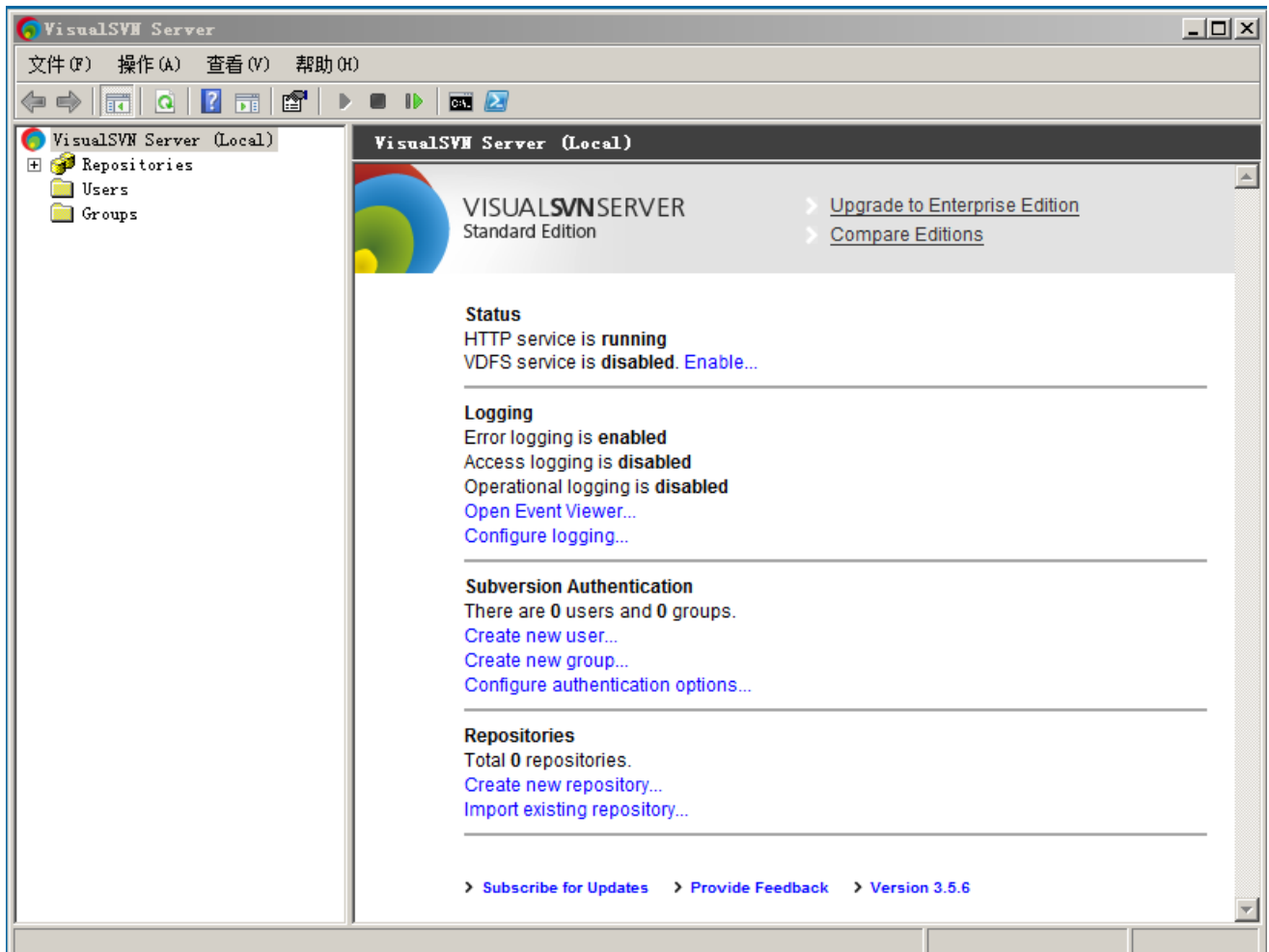


VisualSVN Server

Vis... 已启动 自动

网络服务

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

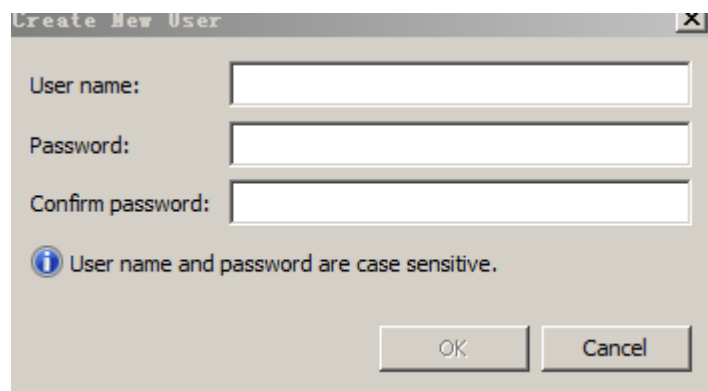
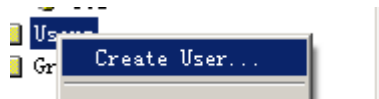


### 3 VisualSVN创建工程目录

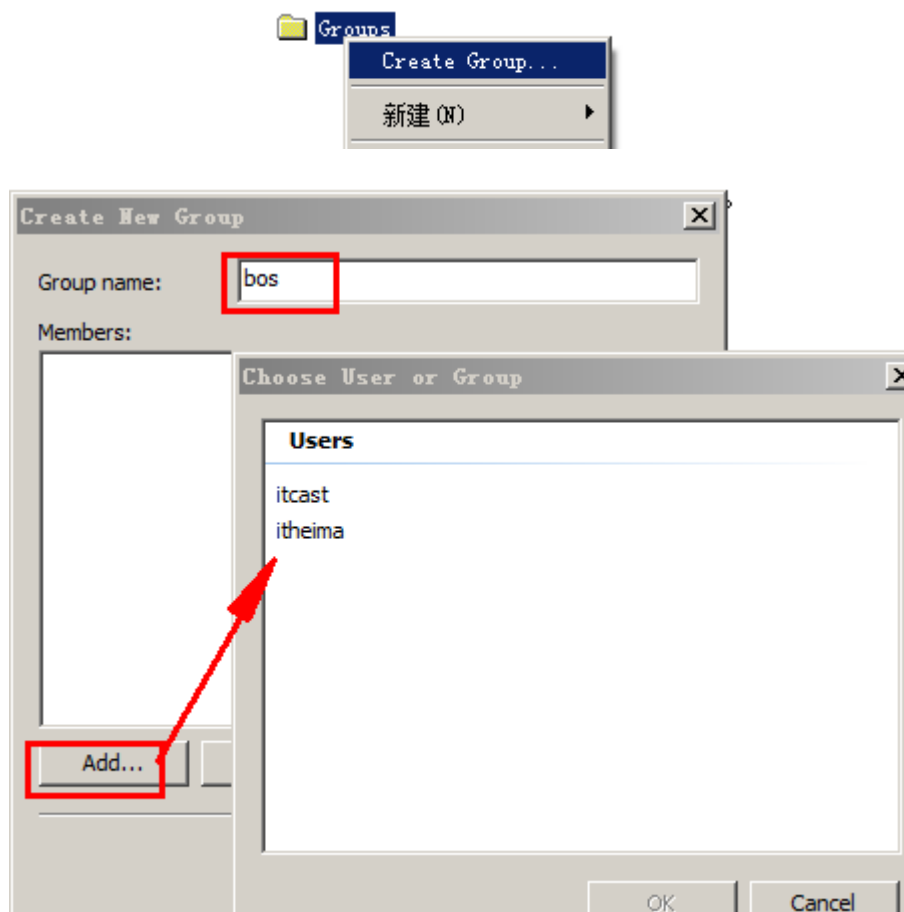


### 4 VisualSVN 权限控制

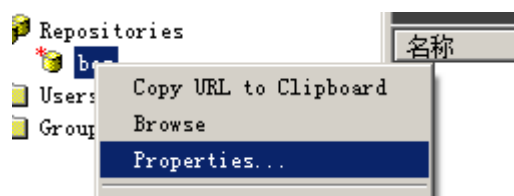
创建用户

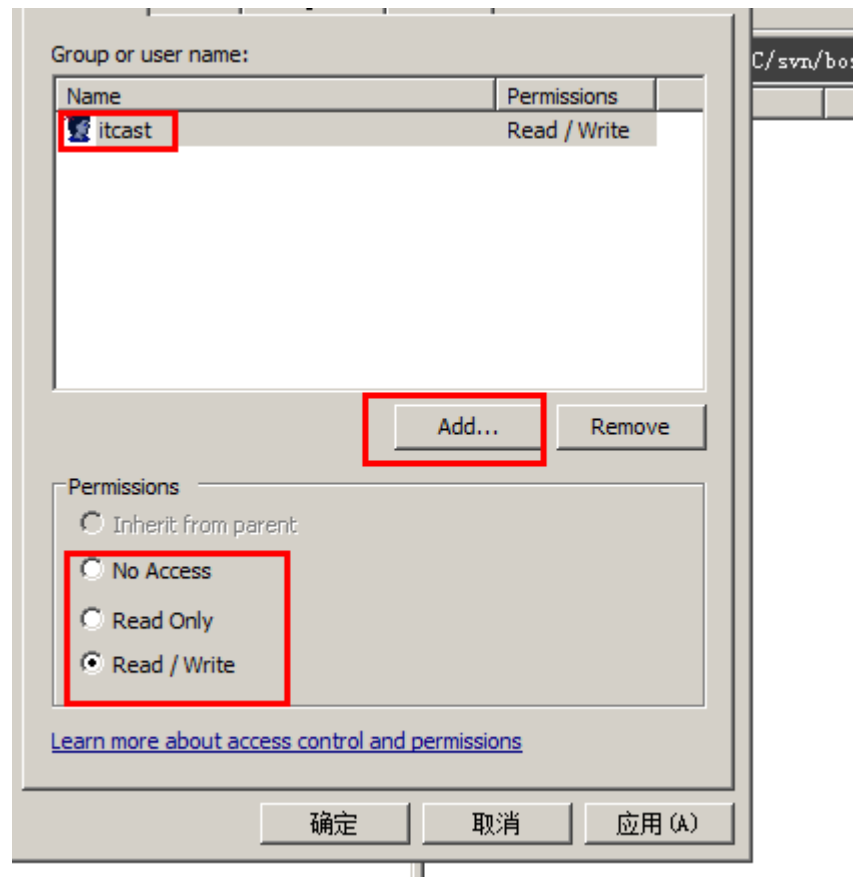


## 创建组



## 分配权限





## 第二节 客户端TortoiseSVN安装与使用

TortoiseSVN是一个基于windows系统的svn客户端图形化界面.

### 1 Tortoise SVN下载

for 32-bit OS

TortoiseSVN 1.9.5 - 32-bit

for 64-bit OS

TortoiseSVN 1.9.5 - 64-bit

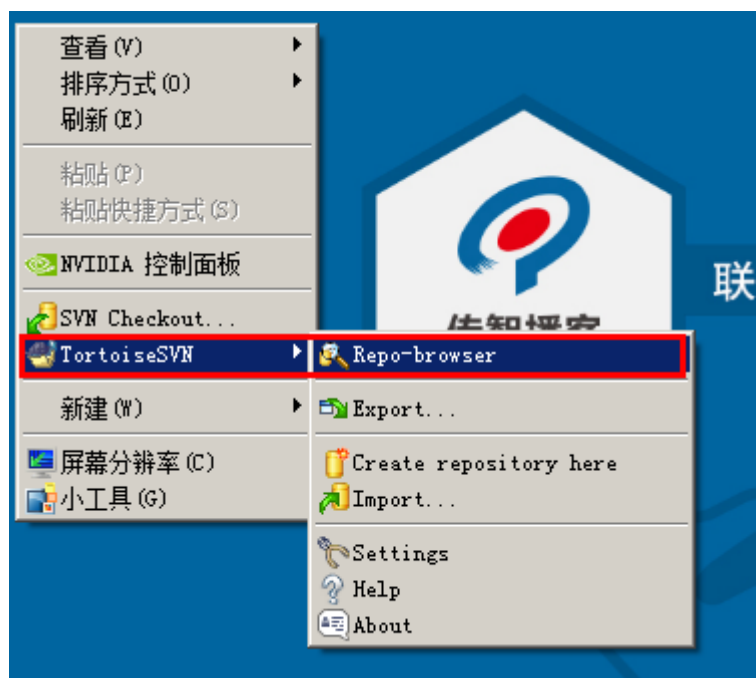
### 2 Tortoise SVN安装

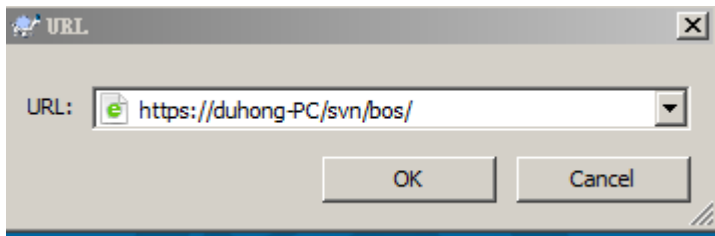
安装后需要重启电脑



### 3 Tortoise SVN 常用操作

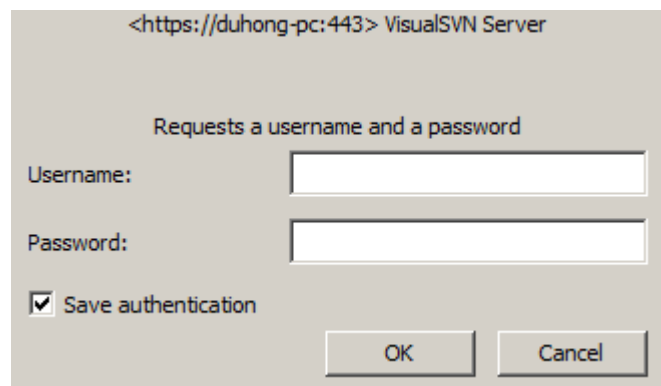
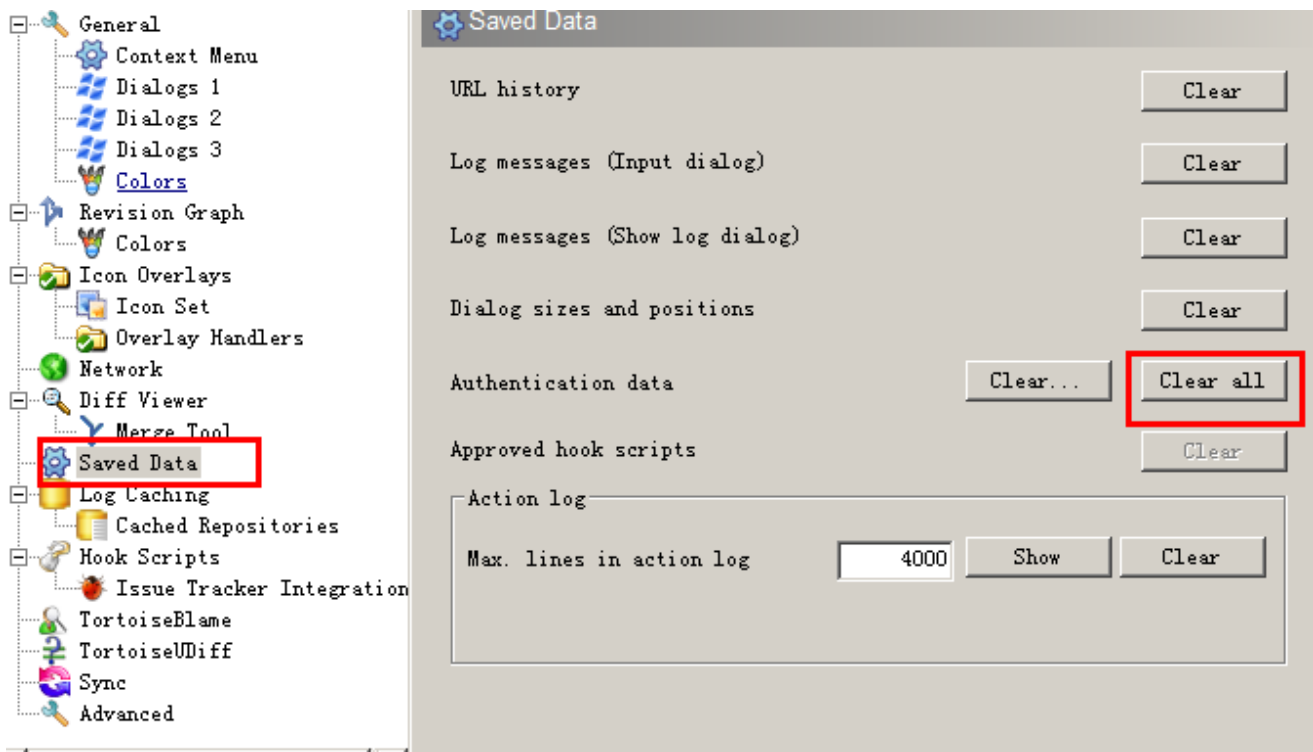
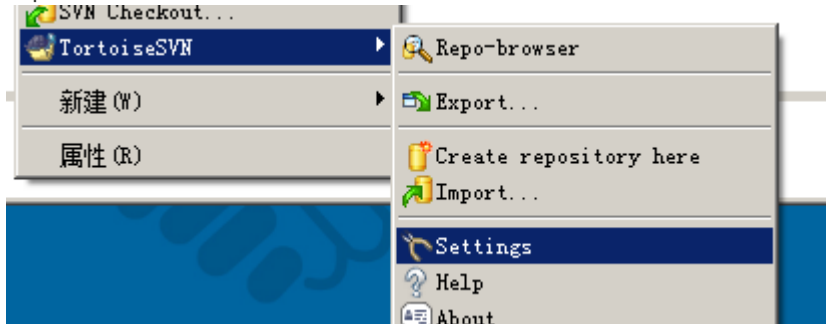
#### 3.1 浏览仓库





正常情况下，在浏览仓库时会要求录入username

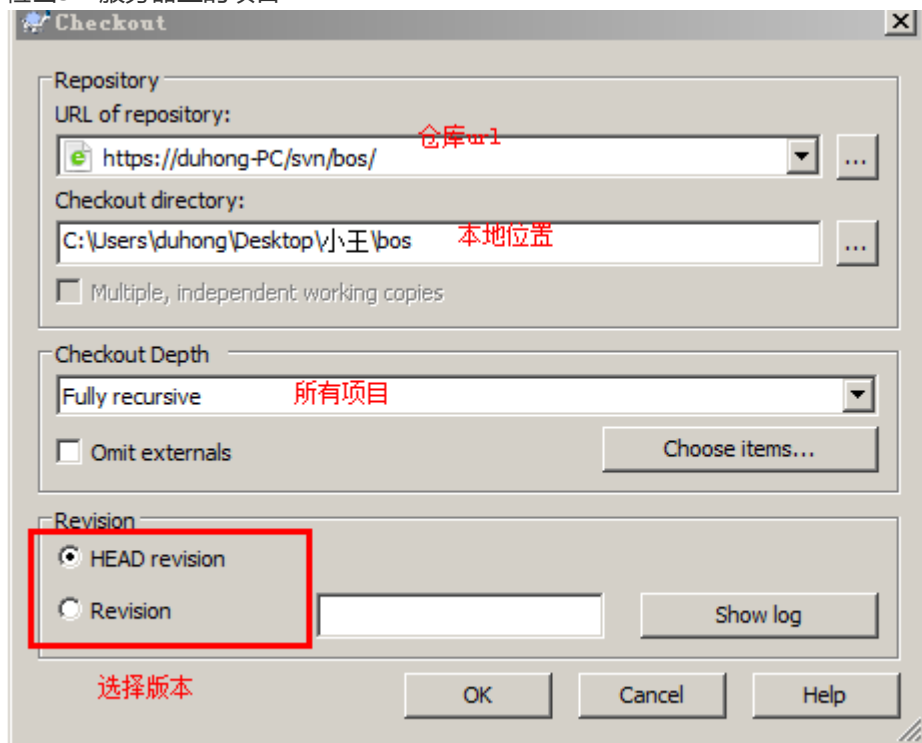
与password 我们可以清空svn客户端的缓存




### 3.2 checkout



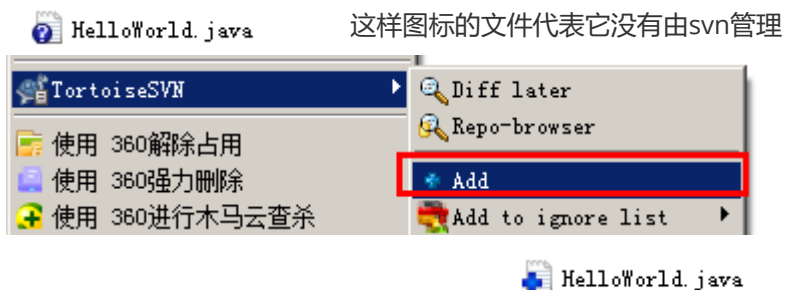
检出svn服务器上的项目



注意：检出后会有

一个.svn隐藏目录  .svn

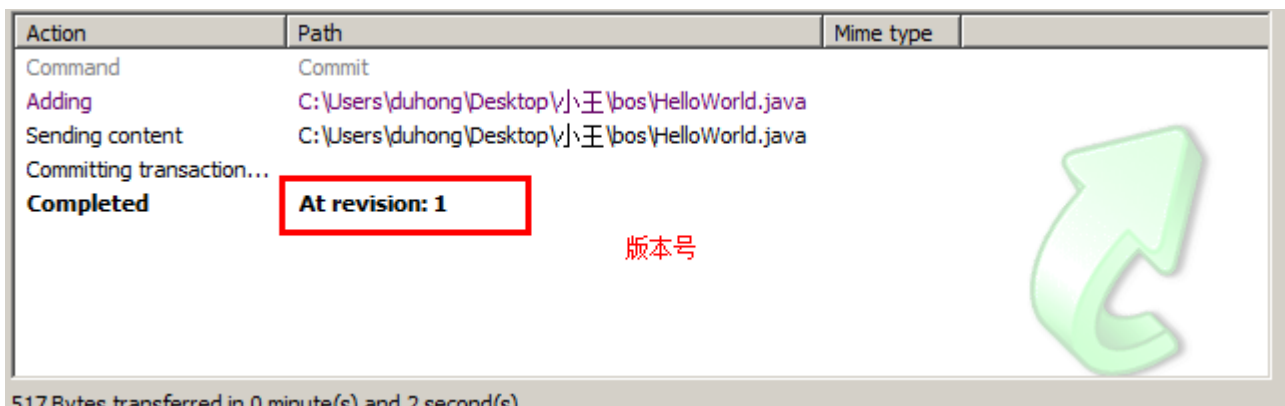
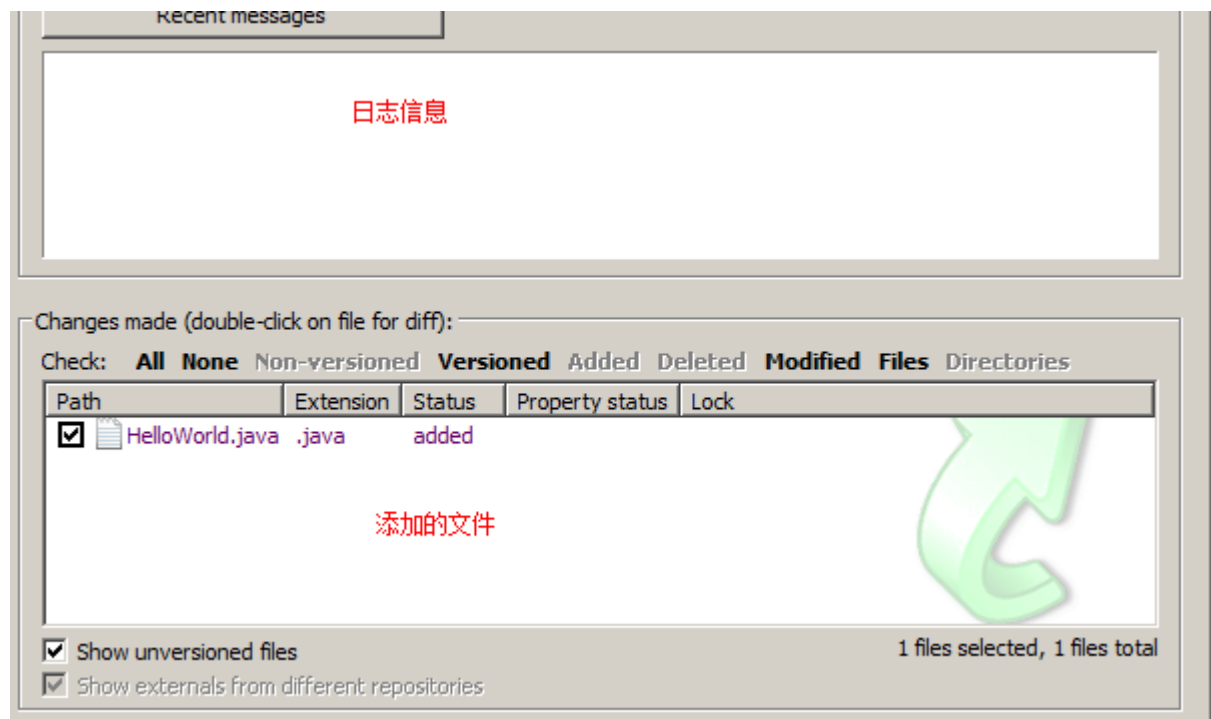
### 3.3 add




这个代表的是它要添加到svn服务器

### 3.4 commit





 HelloWorld.java

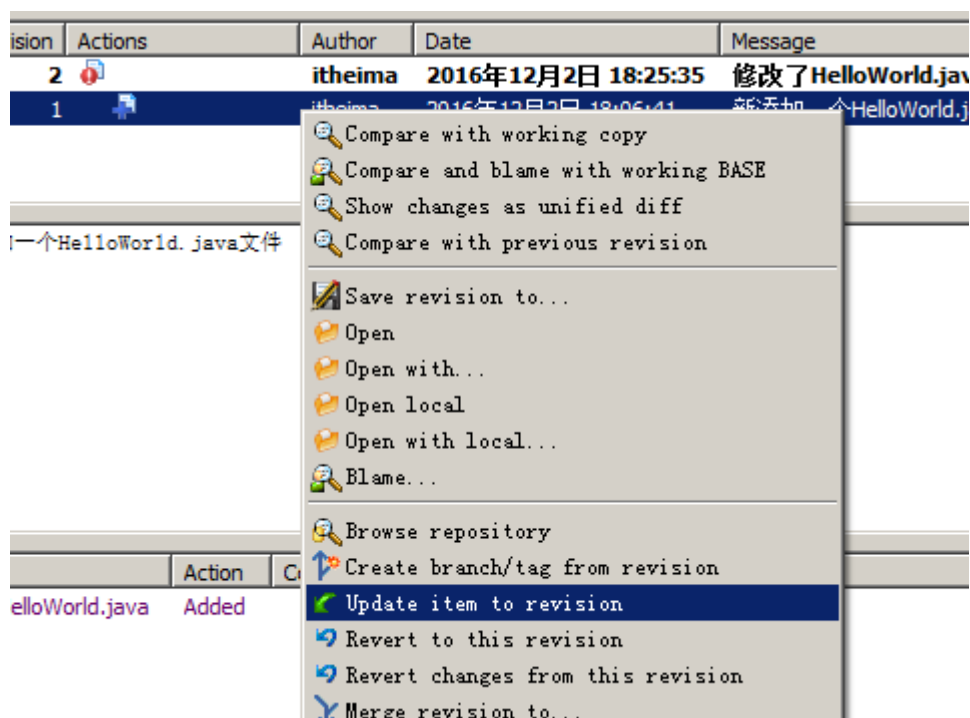
### 3.5 update

Update, 它是更新操作,可以将svn服务器上的内容更新到本地

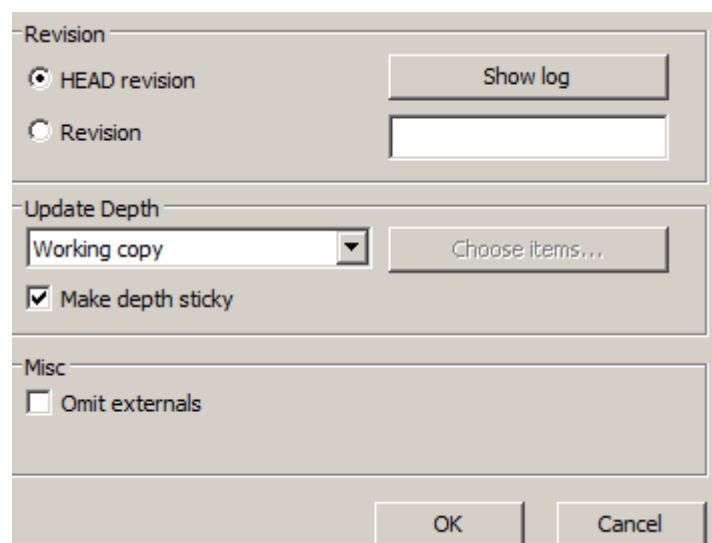


### 3.6 更新到历史版本

查看当前日志

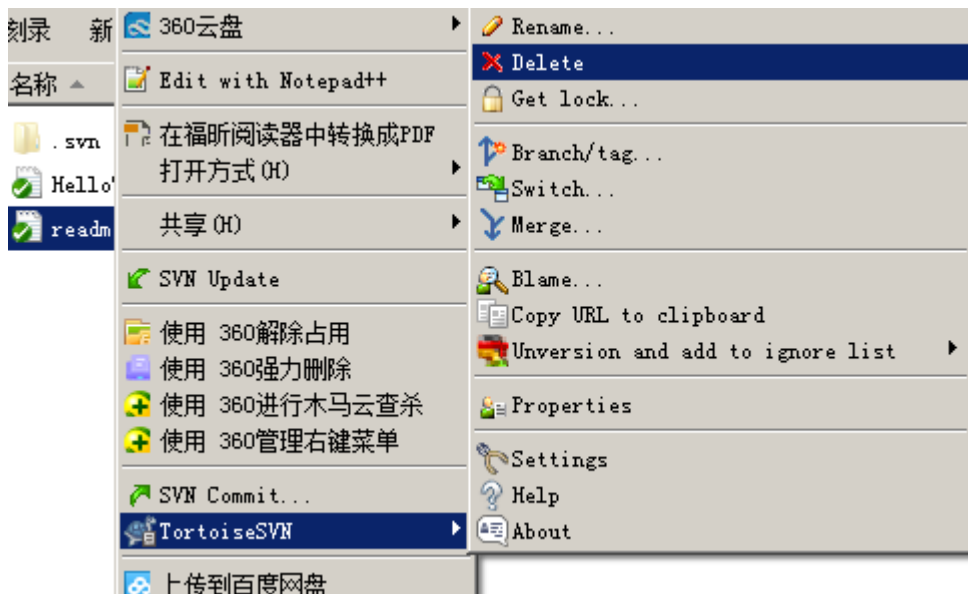


也可以通过下面的方式来更新历史版本

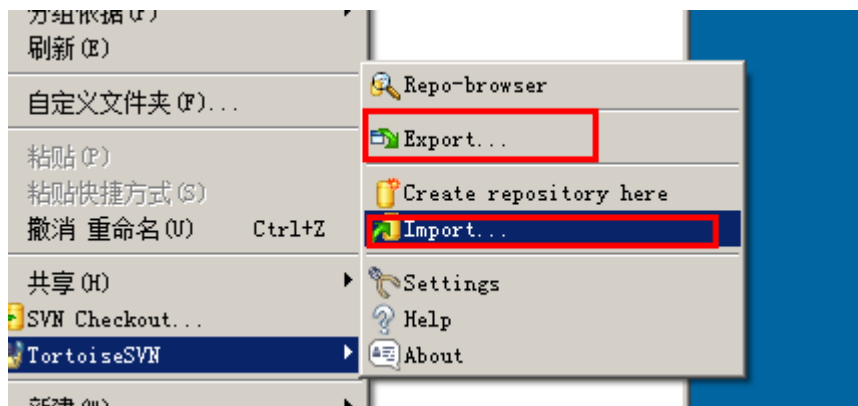


### 3.7 delete

标记删除操作，本地文件会删除，我们需要进行提交，才能真正从服务器删除



### 3.8 导入与导出



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

### 3.9 冲突问题与解决

多个用户同时操作同一个文件时，就可能产生冲突情况。\\

```

Command      Commit
Modified     C:\Users\duhong\Desktop\小刘\bos\HelloWorld.java
Sending content C:\Users\duhong\Desktop\小刘\bos\HelloWorld.java
Error        Commit failed (details follow):
Error        File 'C:\Users\duhong\Desktop\小刘\bos\HelloWorld.java' is out of date
Error        File '/HelloWorld.java' is out of date
Error        You have to update your working copy first.
Completed!
  
```

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	2016/12/2 18:42	JAVA File
HelloWorld.java.mine	2016/12/2 18:42	MINE 文件
HelloWorld.java.r4	2016/12/2 18:42	R4 文件
HelloWorld.java.r5	2016/12/2 18:42	R5 文件

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

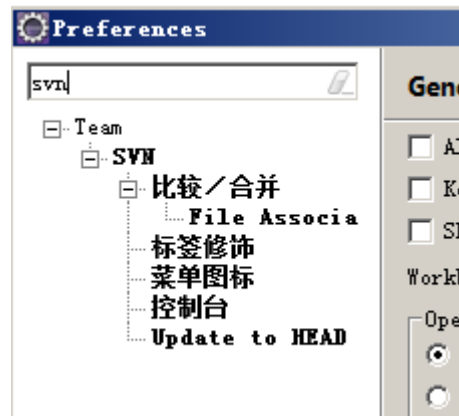


## 第三节 客户端Eclipse SVN 插件安装与使用

### 1 下载与安装svn插件

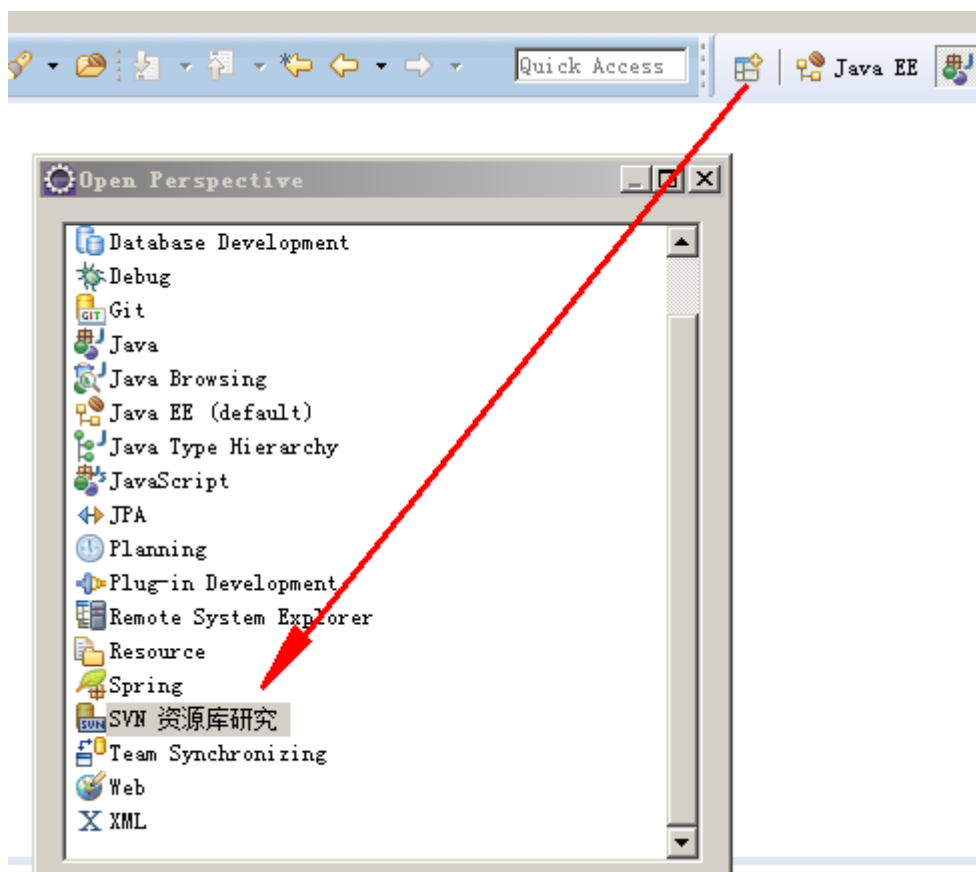
在我们资料中有一个svn插件,交它直接复制到eclipse安装目录下的dropins中

我们怎样可以知道svn插件安装成功



## 2 Eclipse SVN插件使用

### 2.1 浏览仓库

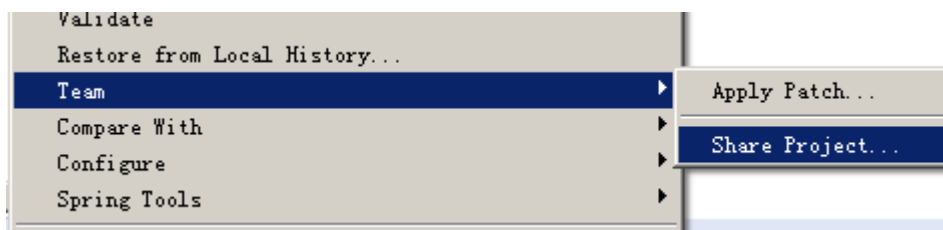


在svn资源库研究

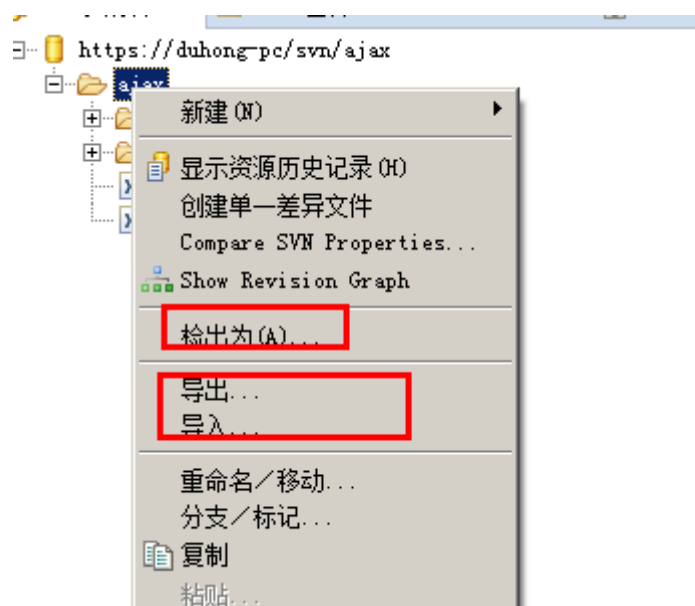


怎样将eclipse中的工程导入到svn仓库, 可以

在工程上右键

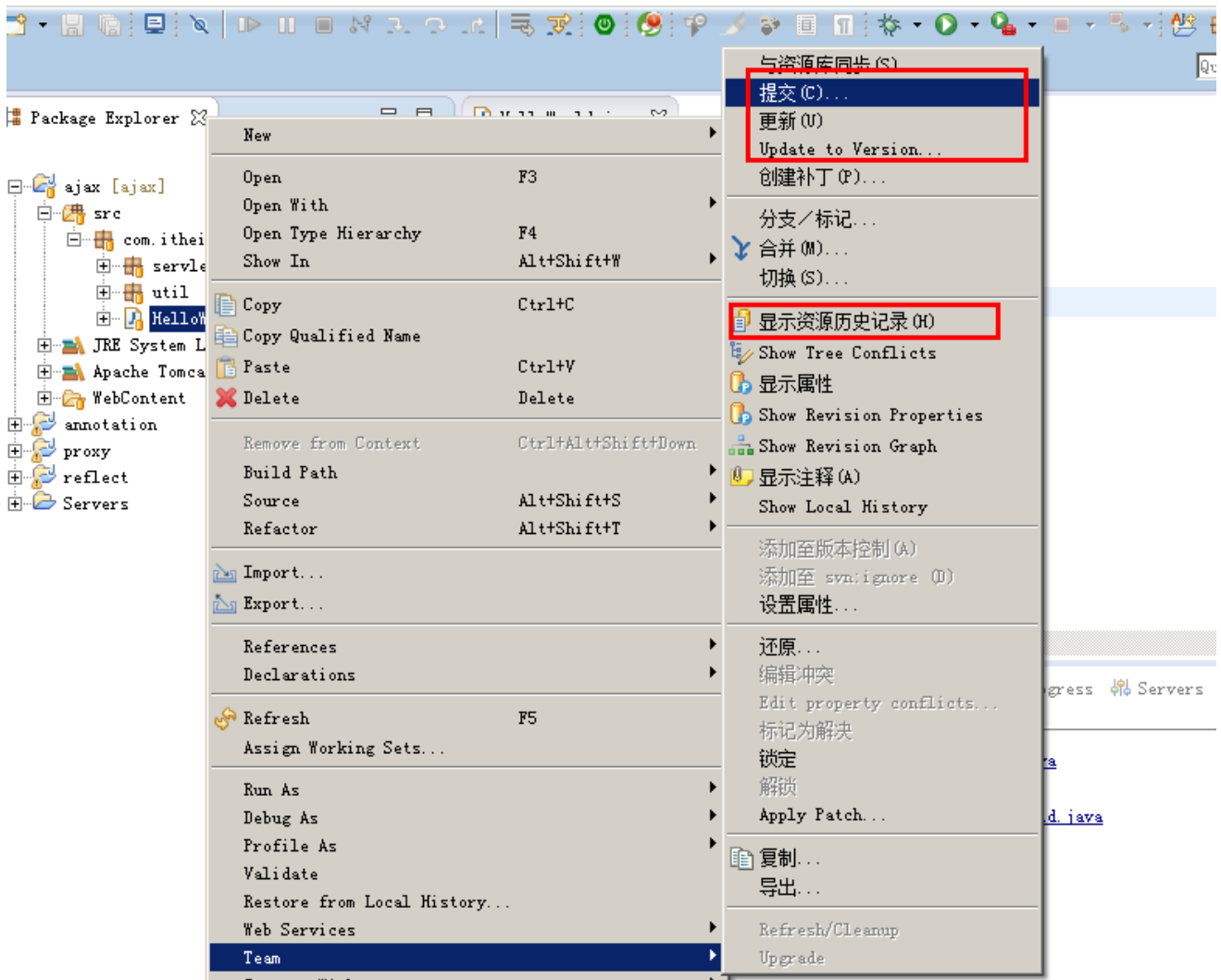


## 2.2 checkout 导入 导出

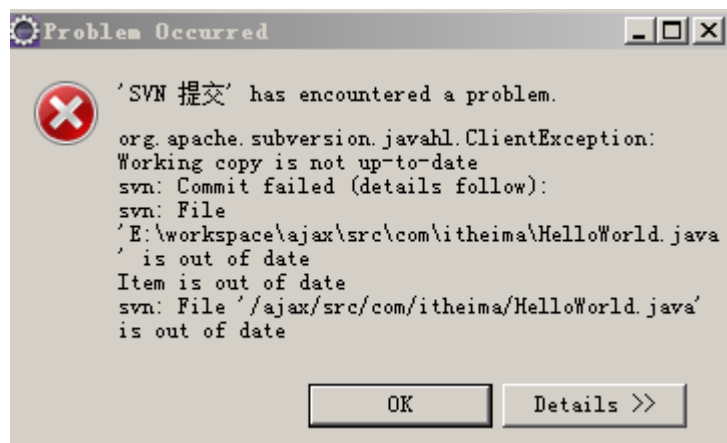


## 2.3 add commit update

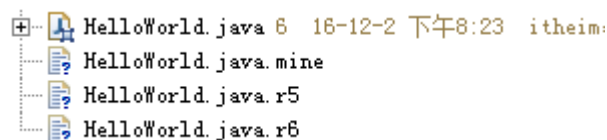




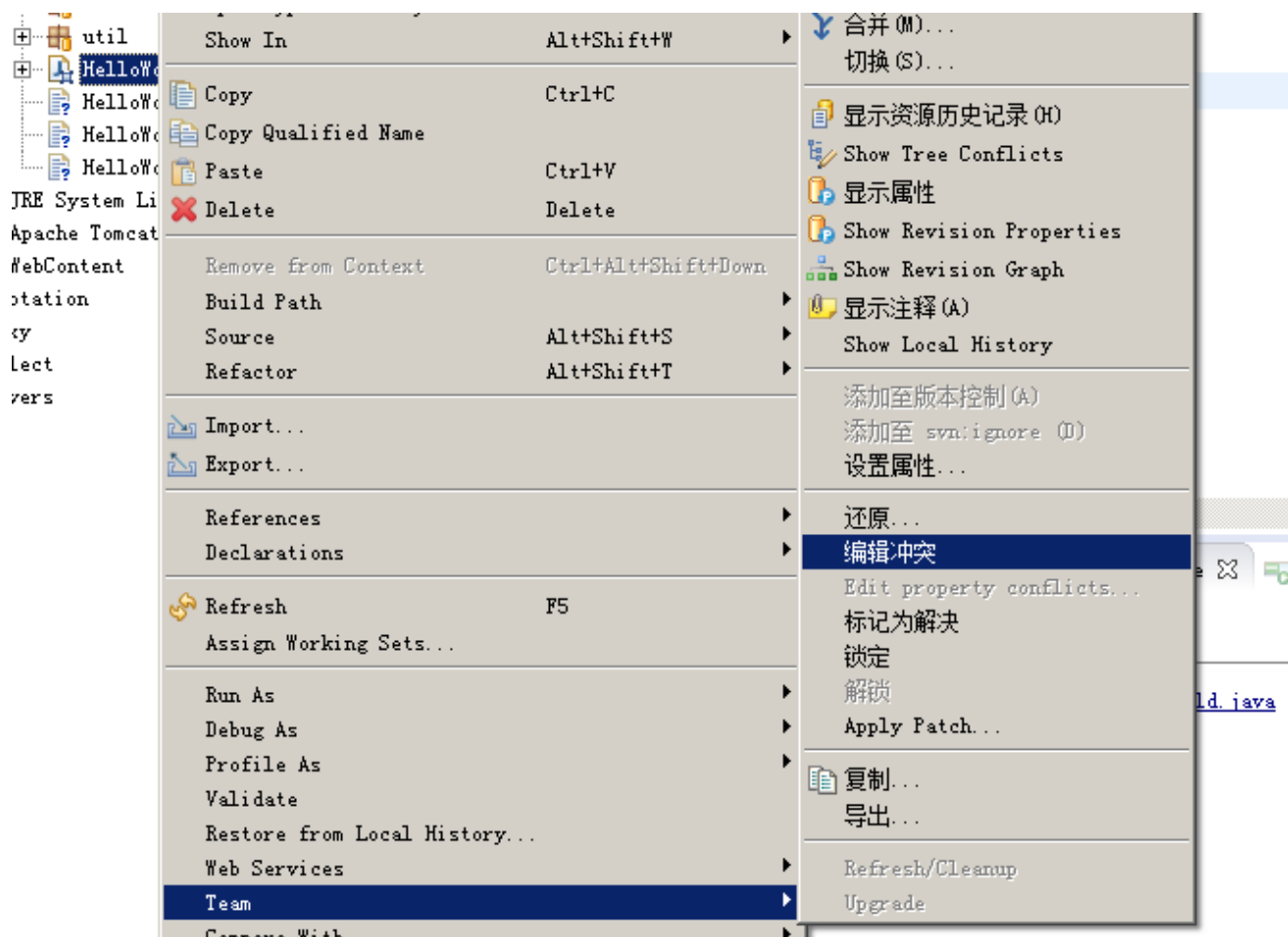
## 2.4 解决冲突



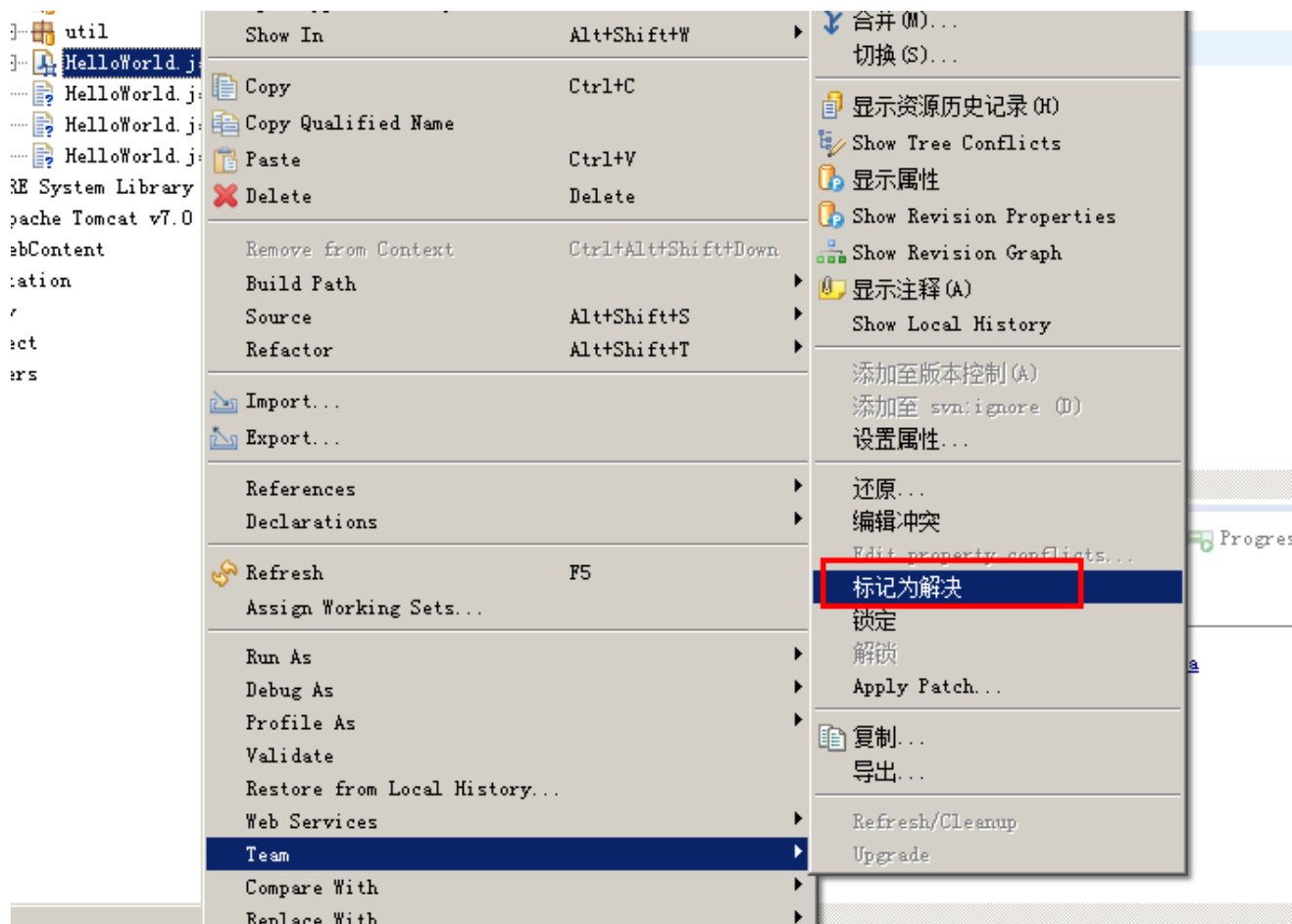
产生了冲突 我们在工程上执行更新操作，就可以查看到冲突的文件



解决冲突



选择编辑冲突

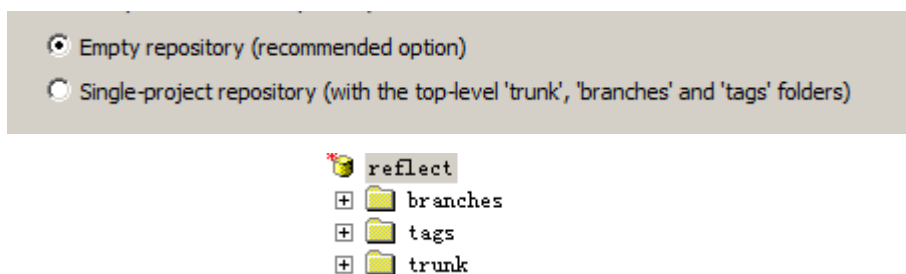


## 第三章 SVN高级

本章节主要介绍我们在真实开中svn使用，例如 合并、分支、标记等。

### 第一节 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

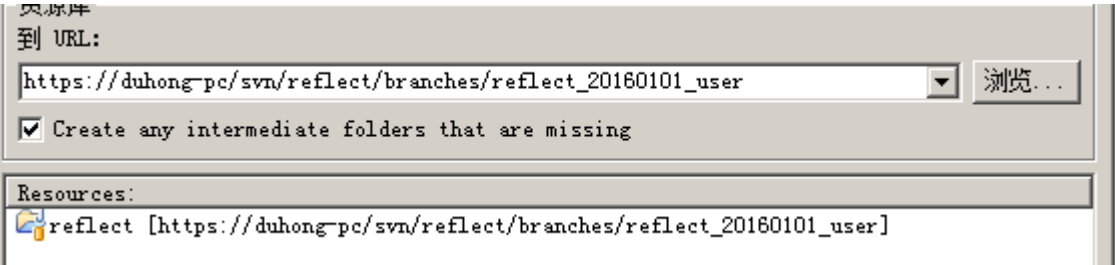
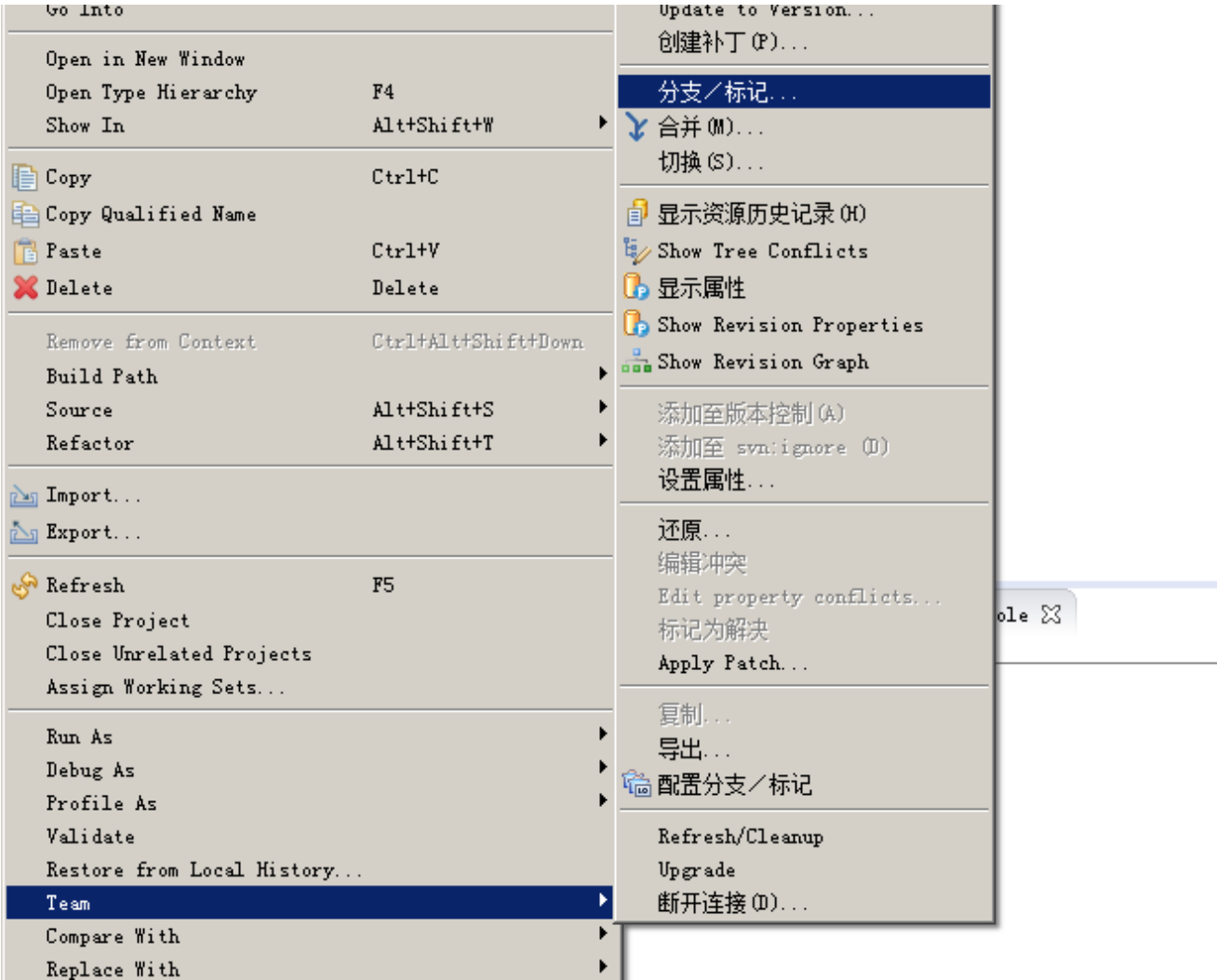
假设有一个项目OA,我们完成了1.0版本，这时就可以打一个tags

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

## 第二节 分支与标记

### 1.打一个分支或标记



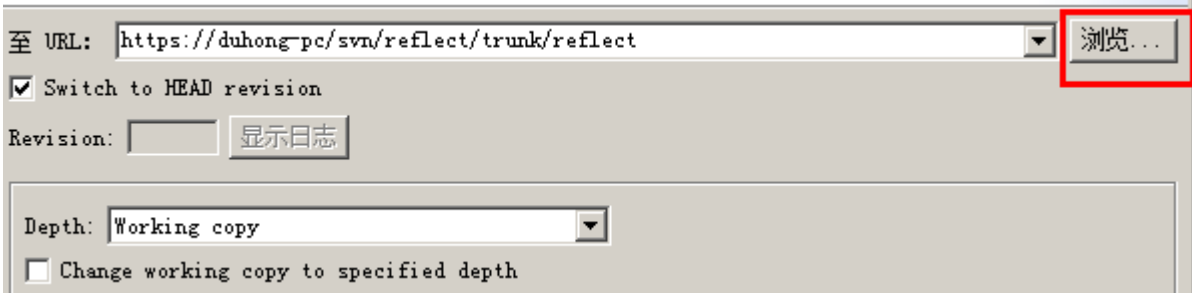
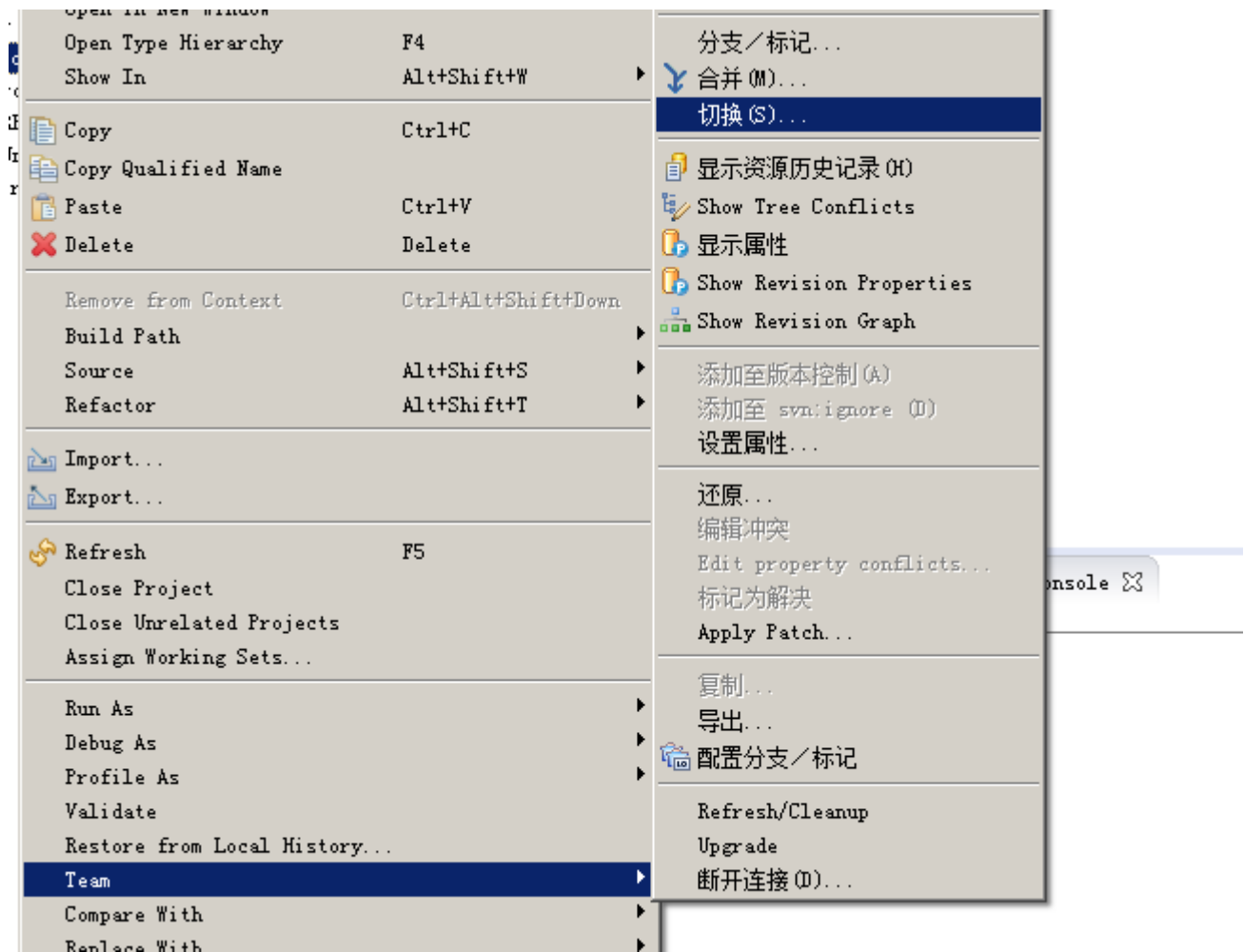
分支的定义规则:

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

2.切换主干，分支与标记



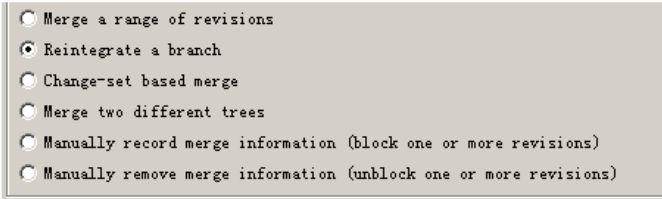
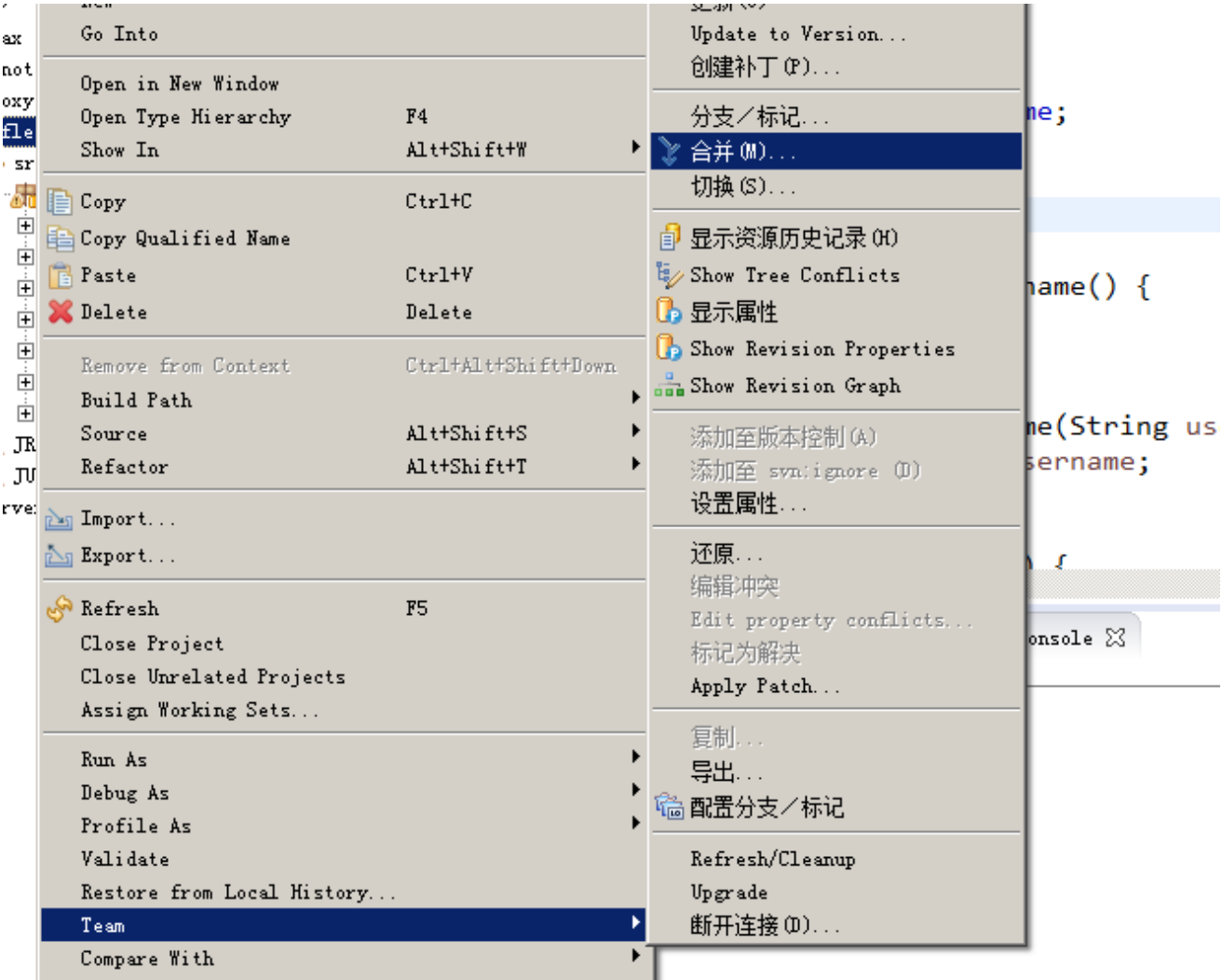
Tags一般是只读，它代表的是发布的版本，所以我们不要进行改变。

第三节 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：



- 1.从主干合并到分支
- 2.从分支合并到主干
- 3.将主干上的修改合并到分支
- 4.合并两个分支到主干
- 5.从主干到分支，手工指定不需要合并的内容
- 6.从主干到分支，手工指定要合并的修改

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

## 第四章：AdminLTE模板使用

### 第一节 概述与基本使用

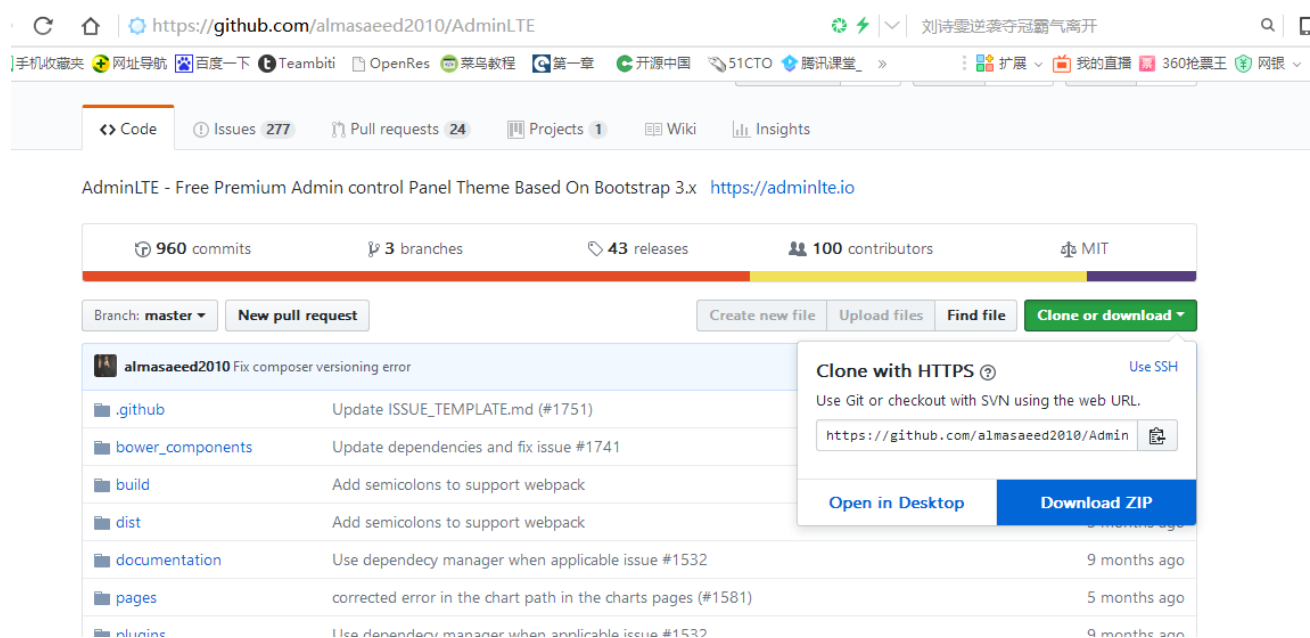
#### 1.AdminLTE介绍

AdminLTE是一款建立在bootstrap和jquery之上的开源的模板主题工具，它提供了一系列响应的、可重复使用的组件，并内置了多个模板页面；同时自适应多种屏幕分辨率，兼容PC和移动端。通过AdminLTE，我们可以快速的创建一个响应式的Html5网站。AdminLTE框架在网页架构与设计上，有很大的辅助作用，尤其是前端架构设计师，用好AdminLTE 不但美观，而且可以免去写很大CSS与JS的工作量。

## 2.GitHub获取AdminLTE






















<https://github.com/almasaeed2010/AdminLTE>

我们可以从上面网址获取AdminLTE源代码



## 3.AdminLTE结构介绍

```
AdminLTE/
├── dist/
│   ├── CSS/
│   ├── JS
│   └── img
├── build/
│   ├── less/
│   │   ├── AdminLTE's Less files
│   │   └── Bootstrap-less/ (Only for reference. No modifications have been made)
│   ├── mixins/
│   ├── variables.less
│   └── mixins.less
└── plugins/
    └── All the customized plugins CSS and JS files
```

	.github	2018/3/14 14:34	文件夹	
	bower_components	2018/3/14 14:34	文件夹	
	build	2018/3/14 14:34	文件夹	
	dist	2018/3/14 14:34	文件夹	
	documentation	2018/3/14 14:34	文件夹	
	pages	2018/3/14 14:34	文件夹	
	plugins	2018/3/14 14:34	文件夹	
	.gitignore	2018/2/4 21:53	文本文档	1 KB
	.npmignore	2018/2/4 21:53	NPMIGNORE 文件	1 KB
	bower.json	2018/2/4 21:53	JSON 文件	2 KB
	changelog.md	2018/2/4 21:53	Typora	10 KB
	composer.json	2018/2/4 21:53	JSON 文件	1 KB
	Gruntfile.js	2018/2/4 21:53	JavaScript 文件	11 KB
	index.html	2018/2/4 21:53	360 se HTML Do...	52 KB
	index2.html	2018/2/4 21:53	360 se HTML Do...	67 KB
	LICENSE	2018/2/4 21:53	文件	2 KB
	package.json	2018/2/4 21:53	JSON 文件	2 KB
	package-lock.json	2018/2/4 21:53	JSON 文件	250 KB
	README.md	2018/2/4 21:53	Typora	4 KB
	starter.html	2018/2/4 21:53	360 se HTML Do...	16 KB
	yarn.lock	2018/2/4 21:53	LOCK 文件	148 KB

AdminLTE依赖于两个框架Bootstrap3与jQuery1.11+

## 4.AdminLTE布局与皮肤

- 布局

.wrapper包住了body下的所有代码

.main-header里是网站的logo和导航栏的代码

.main-sidebar里是用户面板和侧边栏菜单的代码

.content-wrapper里是页面的页面和内容区域的代码

.main-footer里是页脚的代码

.control-sidebar里是页面右侧侧边栏区域的代码

- 布局选项

fixed: 固定

layout-boxed: 盒子布局

layout-top-nav: 顶部隐藏

sidebar-collapse: 侧边栏隐藏

sidebar-mini: 侧边栏隐藏时有小图标

- 皮肤

skin-blue: 蓝色

skin-black: 黑色

skin-purple: 紫色



skin-yellow: 黄色

skin-red: 红色

skin-green: 绿色

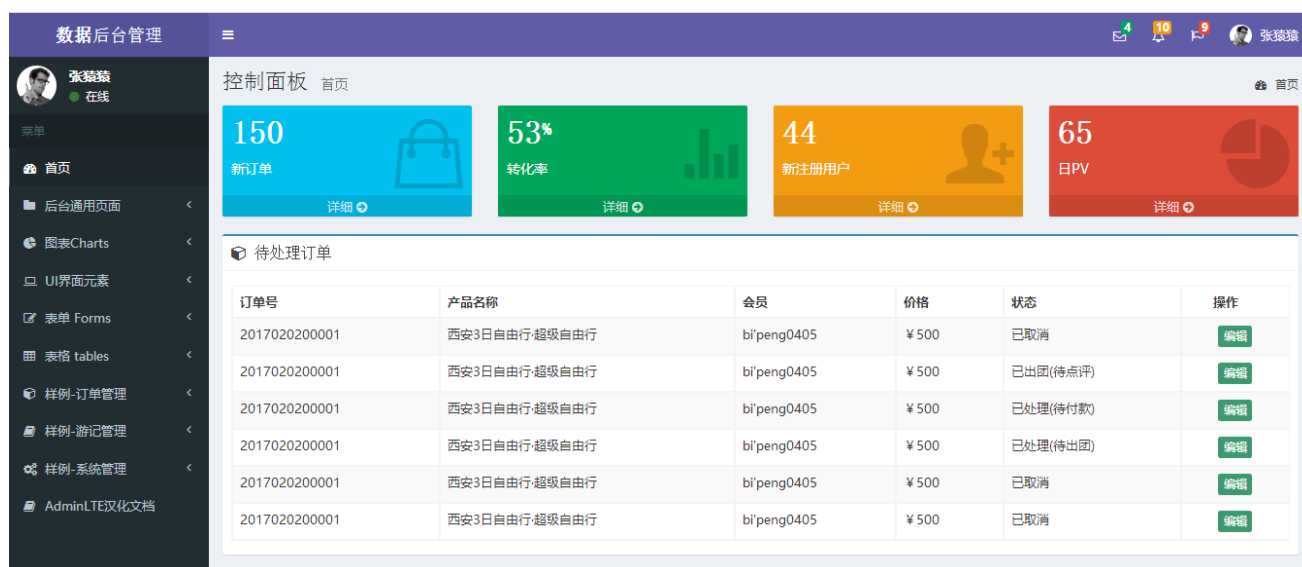
以上项我们可以查看start.html页面中查看。

## 5.AdminLTE2-IT黑马-定制版

传智播客研究院针对英文版本AdminLTE进行了汉化，并优化与定制了部分页面，方便我们的学习与使用。后续SSM综合练习课程中使用的界面就是基于AdminLTE2-IT黑马-定制版。

大家可以去github上下载<https://github.com/itheima2017/adminlte2-itheima>

也可以在 <http://research.itcast.cn/adminlte2-itcast/release/dist/pages/all-admin-index.html> 在线进行浏览



我们将AdminLTE2-IT黑马-定制版本下载解压的目录结构如下：

assets	2018/3/15 10:27	文件夹	
modules	2018/3/15 10:27	文件夹	
pages	2018/3/15 10:27	文件夹	
plugins	2018/3/15 10:27	文件夹	
release	2018/3/15 10:27	文件夹	
fis-conf.js	2017/5/24 16:45	JavaScript 文件	4 KB
fis-plus.js	2017/5/24 16:45	JavaScript 文件	1 KB
gulpfile.js	2017/5/24 16:45	JavaScript 文件	3 KB
package.json	2017/5/24 16:45	JSON 文件	1 KB
README.md	2017/5/24 16:45	Typora	4 KB

由于AdminLTE2-IT黑马-定制版是基于FIS3进行开发，在目录结构中assets、modules、pages、plugins都是前端开发时所使用到的，最终发布的就是release。所以对于我们使用AdminLTE2-IT黑马-定制版来说，我们只需要关注release目录下的结构就可以。

 css	2018/3/15 10:27	文件夹
 img	2018/3/15 10:27	文件夹
 pages	2018/3/15 10:27	文件夹
 plugins	2018/3/15 10:27	文件夹

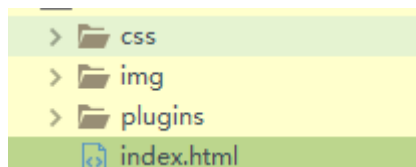
在release目录下有css、img、pages、plugins目录。前两者就不在解决pages是产生的一些定制化的页面，而plugins中是相关的插件，例如jquery、bootstrap等相关的css与js文件。

## 第二节 快速入门案例

本次入门案例我们基于AdminLTE2-IT黑马-定制版来完成

### 1.搭建环境

将AdminLTE2-IT黑马-定制版下的release目录下的css、img、plugins目录导入到我们自己的工程中



### 2.案例代码

编译index.html页面

```
<!DOCTYPE html>
<html>

  <head>

    <meta charset="utf-8">

    <link rel="stylesheet" href="./plugins/bootstrap/css/bootstrap.min.css">
    <link rel="stylesheet" href="./plugins/adminLTE/css/AdminLTE.css">
    <link rel="stylesheet" type="text/css" href="plugins/adminLTE/css/skins/_all-
skins.css"/>
    <title>AdminLTE入门</title>
  </head>

  <body class="skin-blue sidebar-mini">
    <!--
      wrapper包住body下所有代码
    -->
    <div class="wrapper">
      <!--
        描述: main-header是网站的logo和导航栏的代码
      -->
      <header class="main-header">
        <a class="logo">
          网站的logo
```

```
        </a>
        <nav class="navbar navbar-static-top" role="navigation">
            导航栏的代码
        </nav>
    </header>
<!--

    描述：是用户面板和侧边栏菜单的代码
-->
<aside class="main-sidebar">
    <ul>
        <li>菜单1</li>
        <li>菜单2</li>
    </ul>
</aside>
<!--

    描述：页面的页面和内容区域的代码
-->
<div class="content-wrapper">
    页面的页面和内容区域的代码
</div>
<!--

    描述：页脚的代码
-->
<footer class="main-footer">
    页脚的代码
</footer>
</div>
</body>
</html>
```