

**项目管理工具**

**svn**

# 教学目标

1、掌握svn服务端、svn客户端、svn eclipse插件安装方法

2、掌握svn的基本使用方法

# svn介绍

## 项目管理中的版本控制问题

通常软件开发由多人协作开发，如果对代码文件、配置文件、文档等没有进行版本控制，将会出现很多问题：

* 备份多个版本，占用磁盘空间大
* 解决代码冲突困难
* 容易引发BUG
* 难于追溯问题代码的修改人和修改时间
* 难于恢复至以前正确版本
* 无法进行权限控制
* 项目版本发布困难

## 什么是版本控制

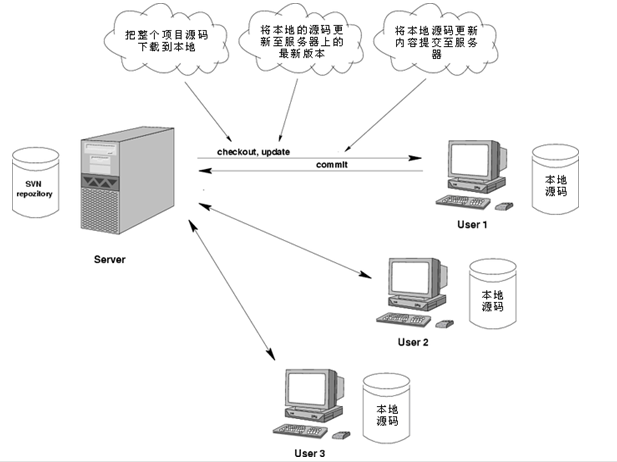
版本控制(Revision control)是维护工程蓝图的标准做法，能追踪工程蓝图从诞生一直到定案的过程。是一种记录若干文件内容变化，以便将来查阅特定版本修订情况的系统。

## svn是什么？

SVN（Subversion）是近年来崛起的版本管理工具，在当前的开源项目里(J2EE)，几乎95%以上的项目都用到了 SVN。Subversion 项目的初衷是为了替换当年开源社区最为流行的版本控制软件 CVS，在 CVS的功能的基础上有很多的提升同时也能较好的解决 CVS 系统的一些不足。

## svn的使用方法

svn是基于客户/服务器模式：



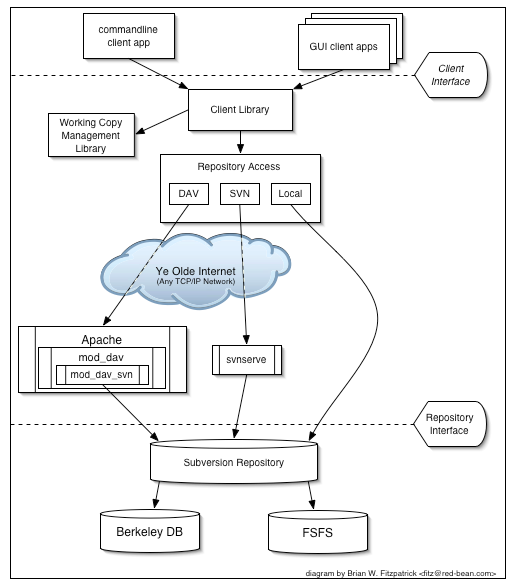
* 复制-修改-合并方案(Subversion默认的模式)

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服务器的工作方式



独立服务器方式：

svnserve

借助Apache方式：

mod\_dav\_svn

SVN版本数据存储方式

BDB （Berkeley DB）

数据库方式

FSFS

文件方式 （推荐）

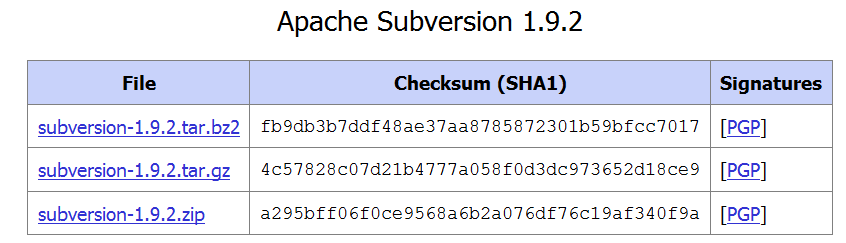
# svn服务端安装配置

## 两种服务端安装包

### 官方安装包

官方网站：http://subversion.apache.org/

下载：http://subversion.apache.org/download.cgi

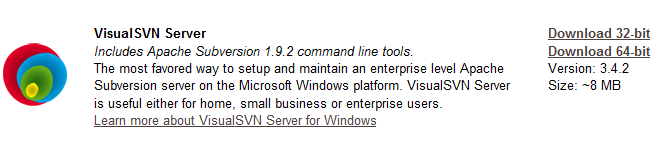


官方提供的服务端安装包，安装后需要通过命令行操作，适用于专业配置管理员使用。

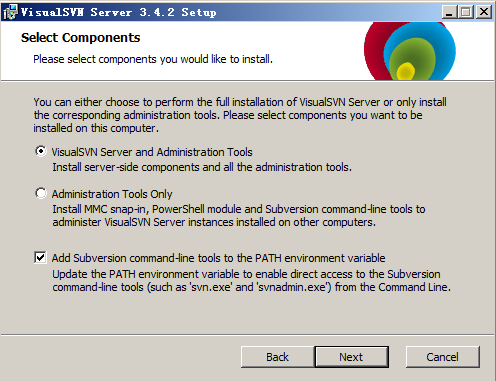
### 图形化服务端

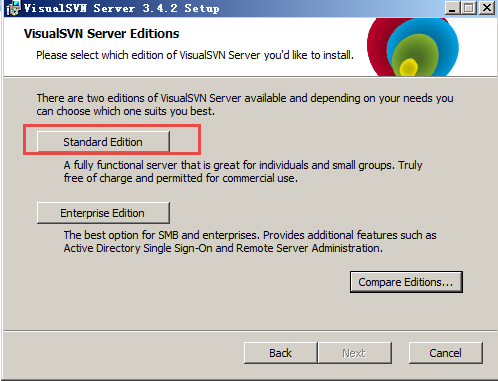
志愿者开发的图形化操作界面的svn服务端，它适用于普通软件开发人员使用。

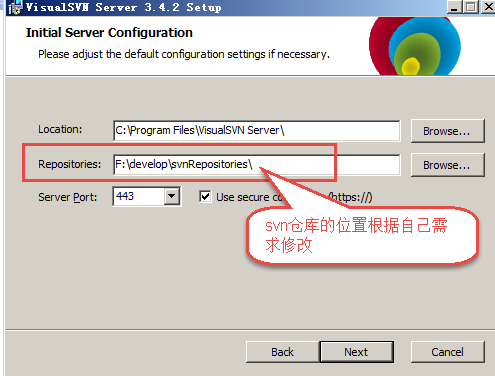
下载地址：https://www.visualsvn.com/downloads/



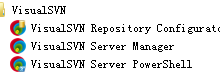
## 安装图形化服务端







查看程序菜单：



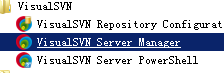
查看服务，VisualSVN成功启动：

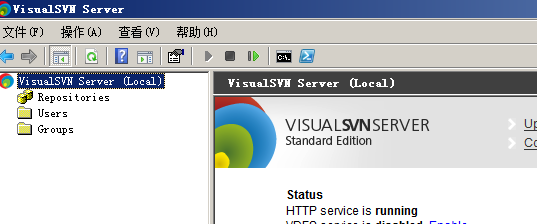


## 创建仓库

svn服务端创建完成需要创建仓库，仓库中存放要版本控制的文件。

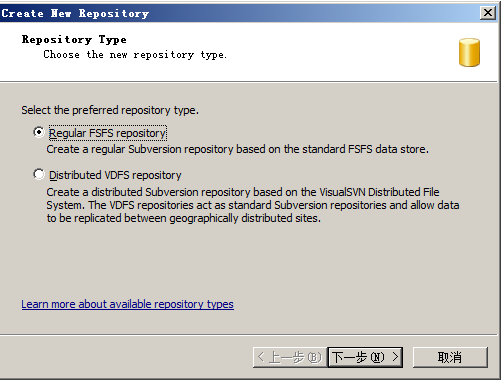
通过开始菜单进入VisualSVN server manager：

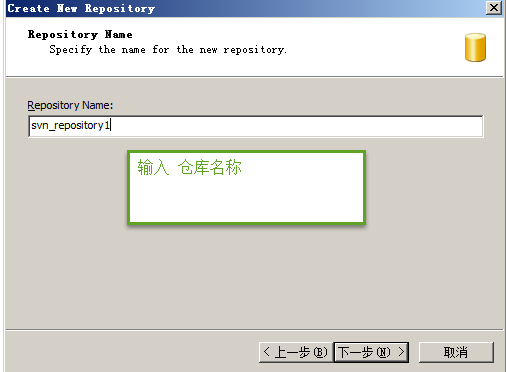




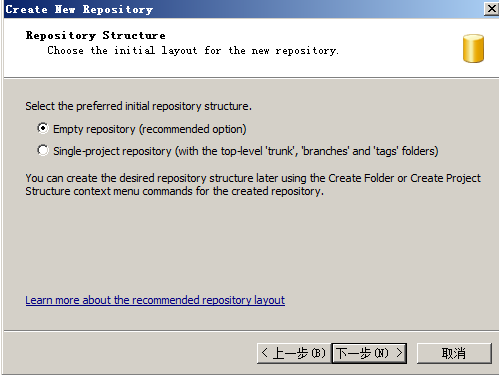


选择文件存储方式：

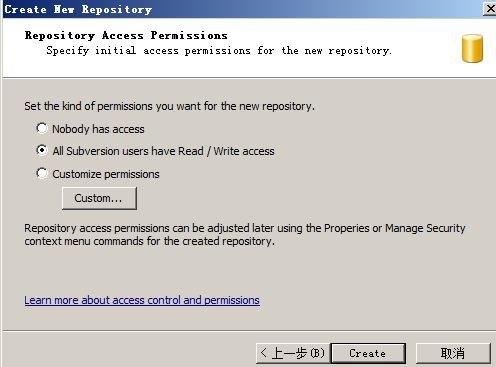




创建一个空的仓库：

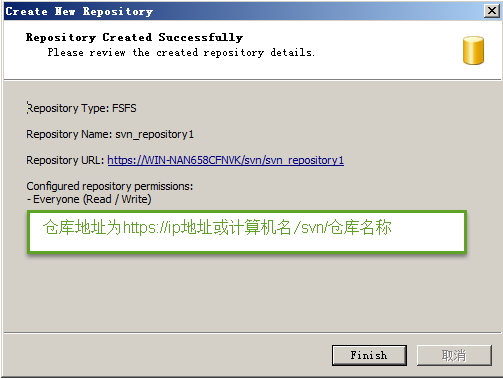


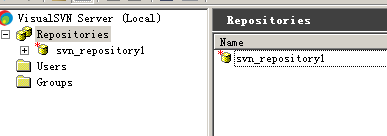
设置用户访问仓库权限：



仓库创建成功：

仓库地址为https://ip地址或计算机名/svn/仓库名称





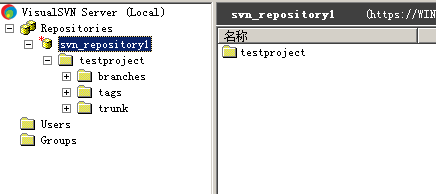
## 创建工程目录

仓库中存放开发项目代码、文档等，需要创建一个工程目录。





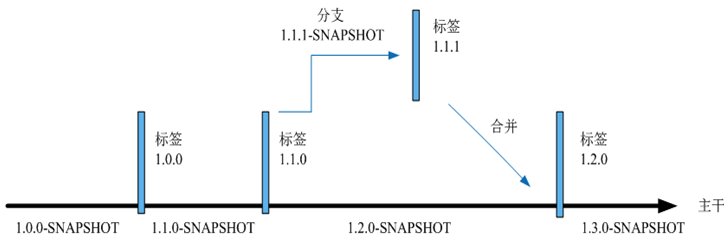
创建成功：



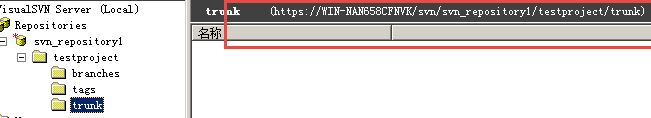
trunk：项目开发代码的主体，是从项目开始直到当前都处于活动的状态，从这里可以获得项目最新的源代码以及几乎所有的变更历史信息。

branch：从trunk的某个点分离出来的代码拷贝，通常可以在不影响主干的前提下在这里进行重大bug的修改，或者做实验性的开发，以及定制功能开发等。如果分支达到了预期的目的，通常可以被合并（Mgerge）到主干中。

tag：用来表示trunk和branch的某个点的状态，以代表项目的某个稳定状态，通常为最终发布状态。



工程目录创建完成，查看它的svn地址：



拷贝svn地址：



# TortoiseSVN客户端

## svn客户端类型

svn客户端需要通过网络访问svn服务端提交文件、查询文件等，可通过以下客户端类型访问svn服务端：

1、使用Subversion提供的客户端命令

使用方式：在命令行下输入命令操作。

2、使用Torotise图形化界面操作（推荐）

3、使用Eclipse等开发工具插件操作（推荐）

## 下载安装

TortoiseSVN是Subversion版本控制系统的一个免费开源客户端，不需要为使用它而付费。

TortoiseSVN是 Subversion 的 Windows 扩展。它使你避免接触 Subversion 枯燥而且不方便的 Command Line。它完全嵌入 Windows Explorer，使用时只需在正常的窗口里右键操作就可以了



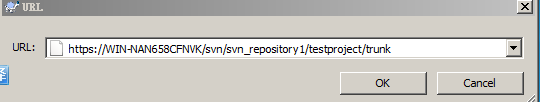
下载：http://tortoisesvn.net/downloads.html 提供 32位和64位不同版本，安装tortoiseSVN 需要修改客户端电脑右键菜单，安装后需要重启电脑。

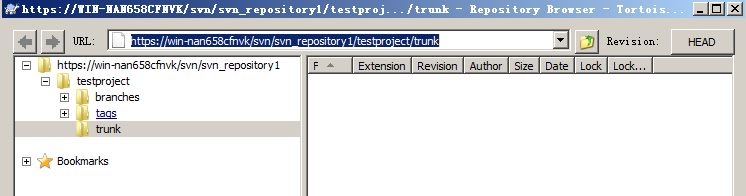


## 浏览仓库

使用Tortoise浏览svn服务端的仓库的内容：







# 权限管理(了解)

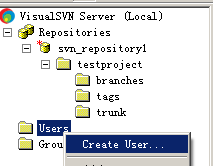
## 认证授权机制

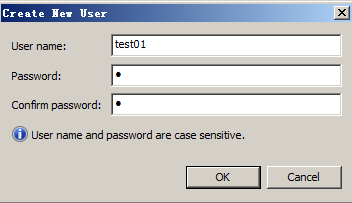
在企业开发中会为每位程序员、测试人员等相关人员分配一个账号，用户通过使用svn客户端连接svn服务时需要输入账号和密码，svn服务对账号和密码进行校验，输入正确可以继续访问，当用户访问仓库下某个目录时，svn服务对用户进行授权，如果用户拥有该目录的访问权限方可访问。

判断账号和密码输入是否正确的过程即认证过程。

判断用户是否拥有目录的读/写权限时即授权过程。

## 创建用户





查看已创建的用户：

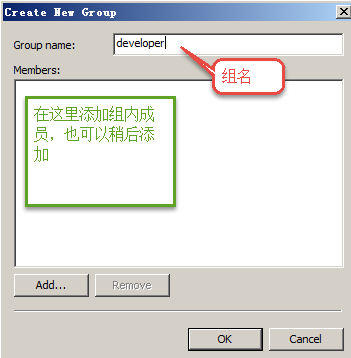


修改用户：

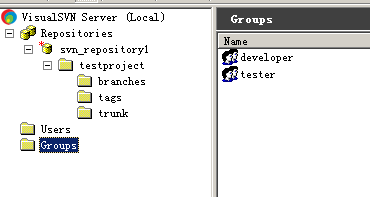


## 创建组

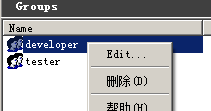




查看创建的组：



修改组：

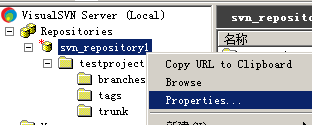


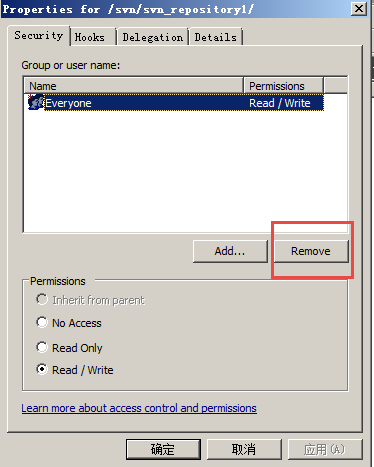
## 分配仅限

给仓库下的每个目录分配权限对访问进行控制。

### 删除默认权限

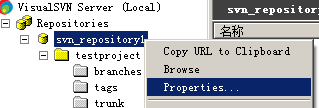
删除系统安装后默认权限：



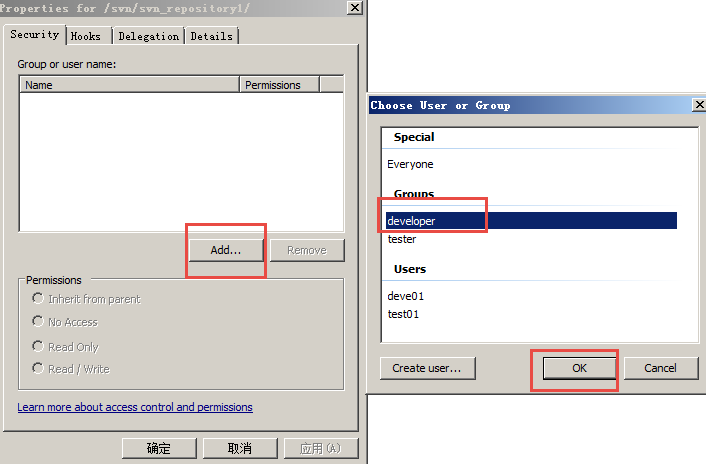


### 示例一：开发人员拥有读写权限

进入权限分配界面：

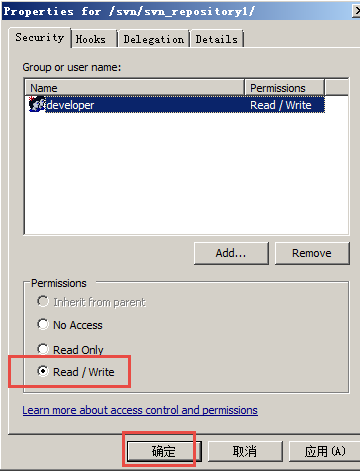


添加组或用户：

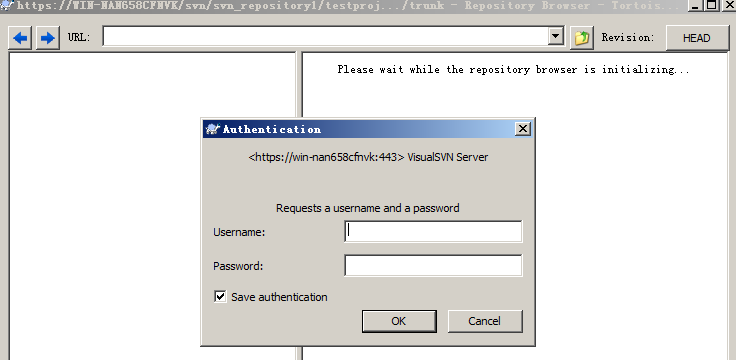


分配权限：

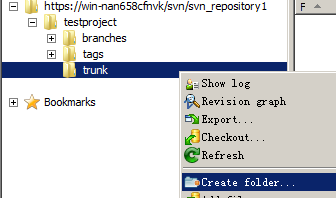
继承父目录权限、不可访问、读权限、读/写权限



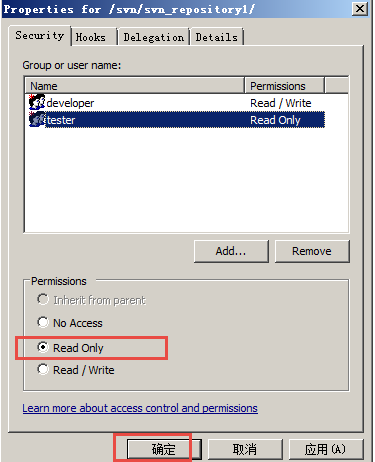
访问时输入账号：



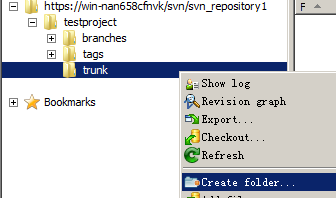
登陆测试是否有读/写权限：



### 示例二：测试人员拥有读权限



登陆测试是否有读/写权限：

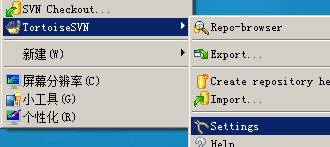


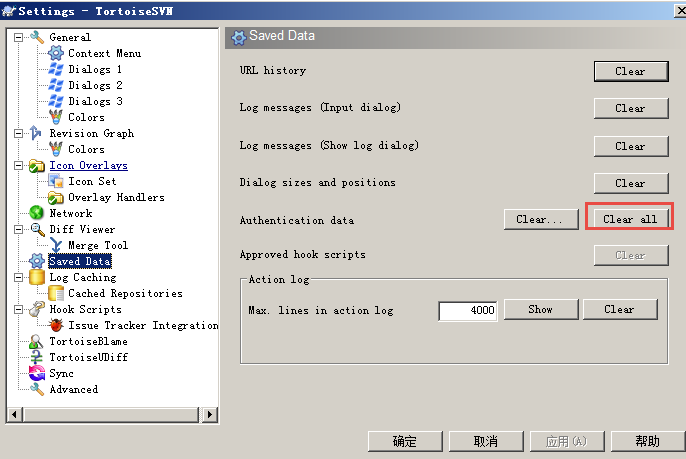
### 清除认证缓存

有几种情况需要清除认证缓存：

1、本地使用多个账号登陆，每次输入的账号和密码都不一样

2、当账号密码修改后（建议清理）

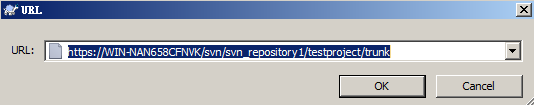




# TortoiseSVN日常使用

## 浏览仓库

Repo-browser ： 浏览仓库中资源信息

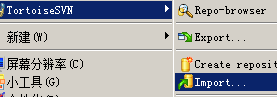


## 导入导出

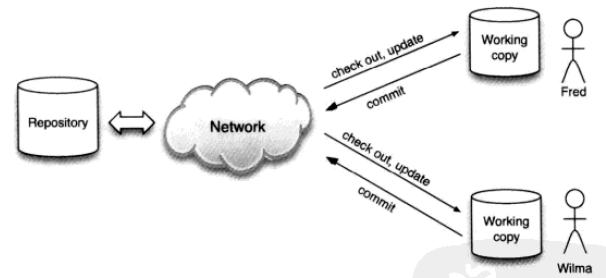
* Export ：导出项目 ，和checkout区别 （checkout检出后文件，含有.svn隐藏文件夹， 会和SVN仓库交互， export导出，没有.svn隐藏文件夹）



* import 将本地资源导入到svn 服务器



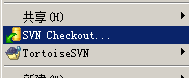
## 修改提交

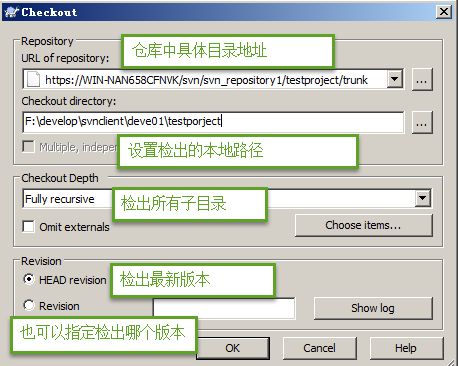


### Checkout

检出项目，复制项目的副本到本地。

在要检出的目录中右键：





### add

在检出的目录中添加文件：

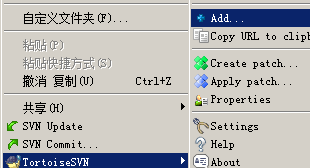


图标： 这是一个新文件

* Add to ignore list ：添加到忽略列表 （标记该文件不需要版本控制 ）



* Add ： 标记这个文件添加到服务器



 已经标记要添加到版本库

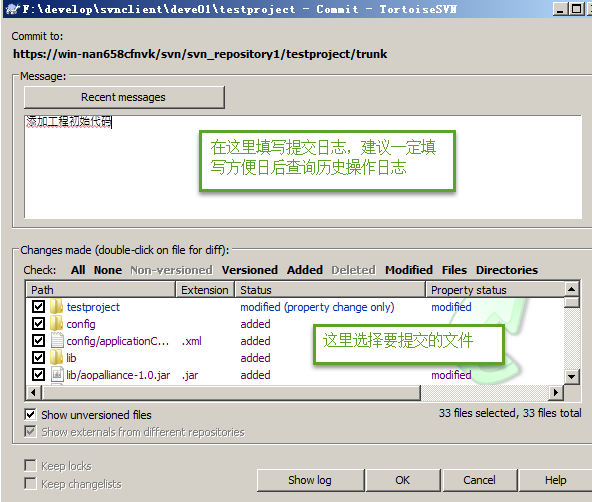
### Commit

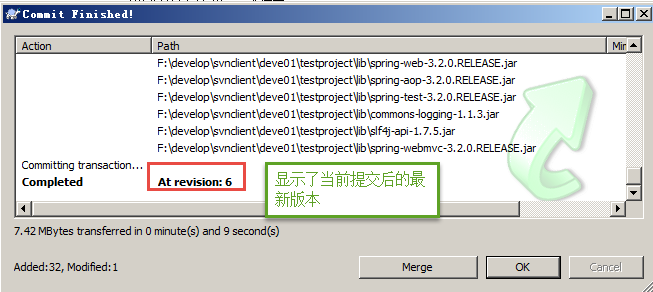
当检出目录或子目录中内容有修改，目录图标变为：

提交Commit 提交本地修改至svn服务器：

在检出目录或要提交修改的目录右键：





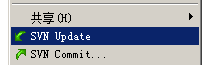


提交后目录中的内容与svn服务同步，目录图标变为：

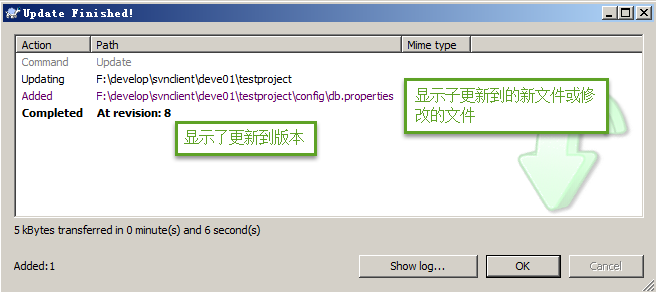
### update

更新仓库的文件到本地

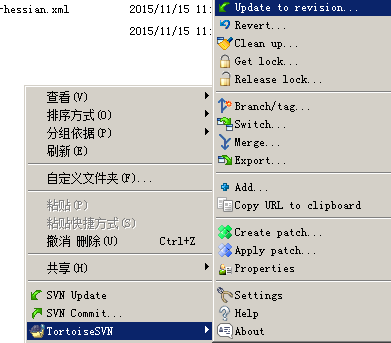
在检出目录或子目标或文件上右键：

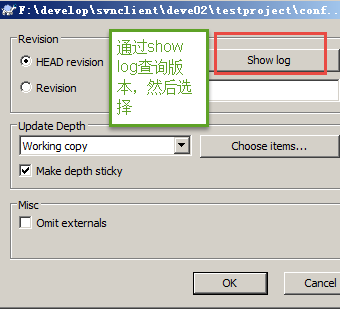


#### 更新到最新版本



#### 更新到指定版本





### Delete

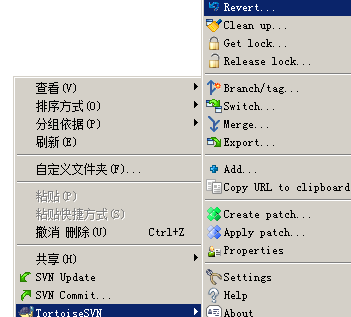
Delete ：删除版本库文件

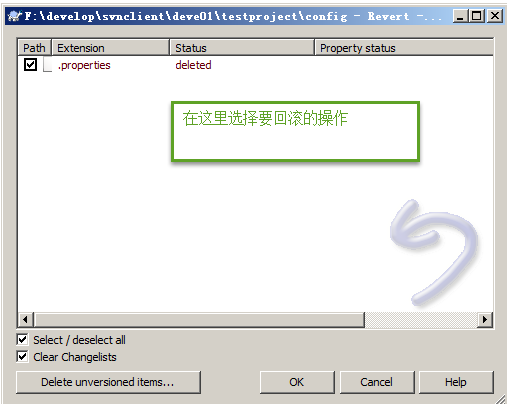
 标记删除后，本地文件删除，标记删除后需要提交。

### 恢复

在检出目录或子目录操作会记录操作日志，提交前可以回滚操作。

在要回滚的检出目录或子目录中右键：

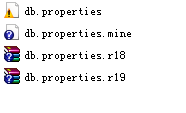




## 冲突处理

两个客户端同时修改同一个文件， 改动同一个位置，发生冲突情况

如果当commit 遇到文件已经过时，说明另一个人可能改动过 ----- update

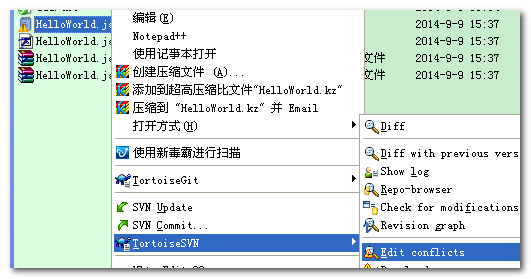


db.properties 将本地和服务器合并到一起的文件 （不要直接看）

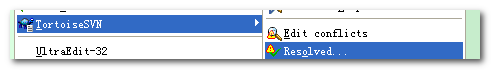
db.properties.mine 我本地自己修改后的文件

db.properties.r16 我修改之前的文件

db.properties.r17 别人修改后的文件



手动Merge 后，需要将编辑后冲突文件，标记为已经解决 ， 再进行commit



# eclipse的SVN插件使用

## svn插件安装

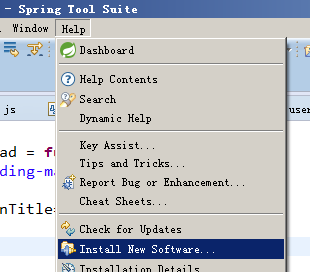
下载Subversion的eclipse插件

<http://subclipse.tigris.org/servlets/ProjectProcess?pageID=p4wYuA>

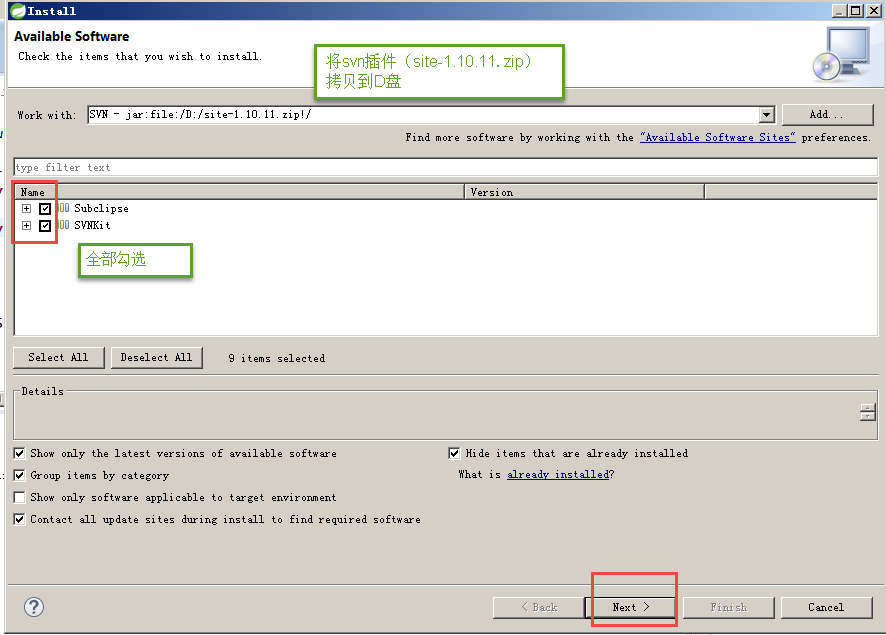
下载 site-1.10.11.zip，本教程将此zip放在D盘。

下边是在STS上安装svn插件步骤：

1、进入STS软件安装界面



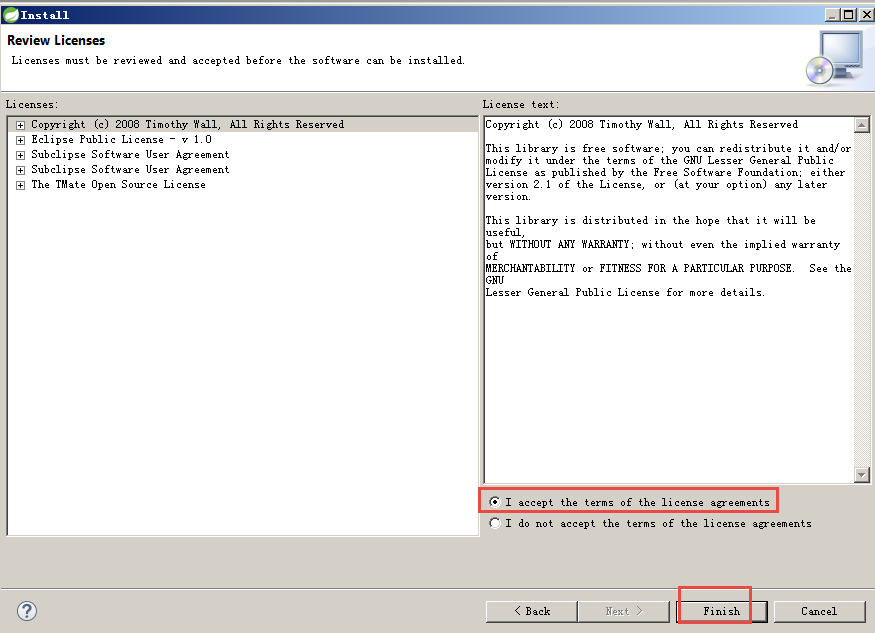
2、安装svn插件



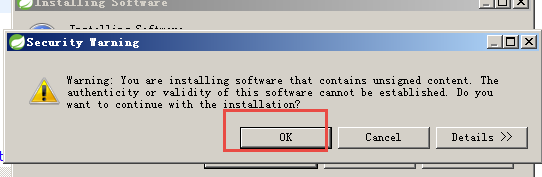
上边命令行输入：SVN - jar:file:/d:/site-1.10.11.zip!/

点击下一步。

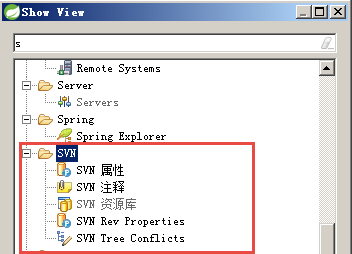
接受协议，完成：



出现提示，选择继续安装：



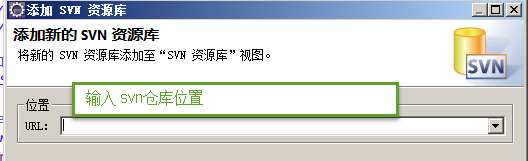
安装完成，查看STS视图有SVN选项说明安装成功：

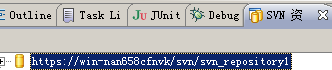


## 将项目共享到SVN

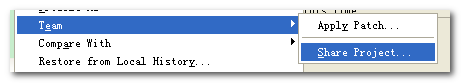
新建SVN仓库连接 ， 切换“SVN资源库” 视图







共享项目到SVN

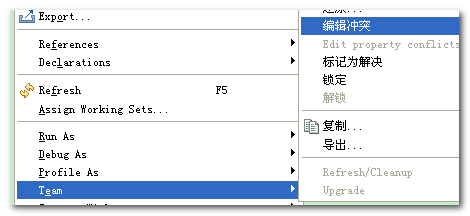


## 从svn检出

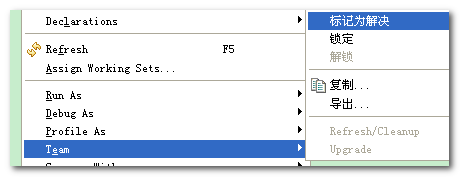
将svn管理项目检出到工作区



## 解决冲突

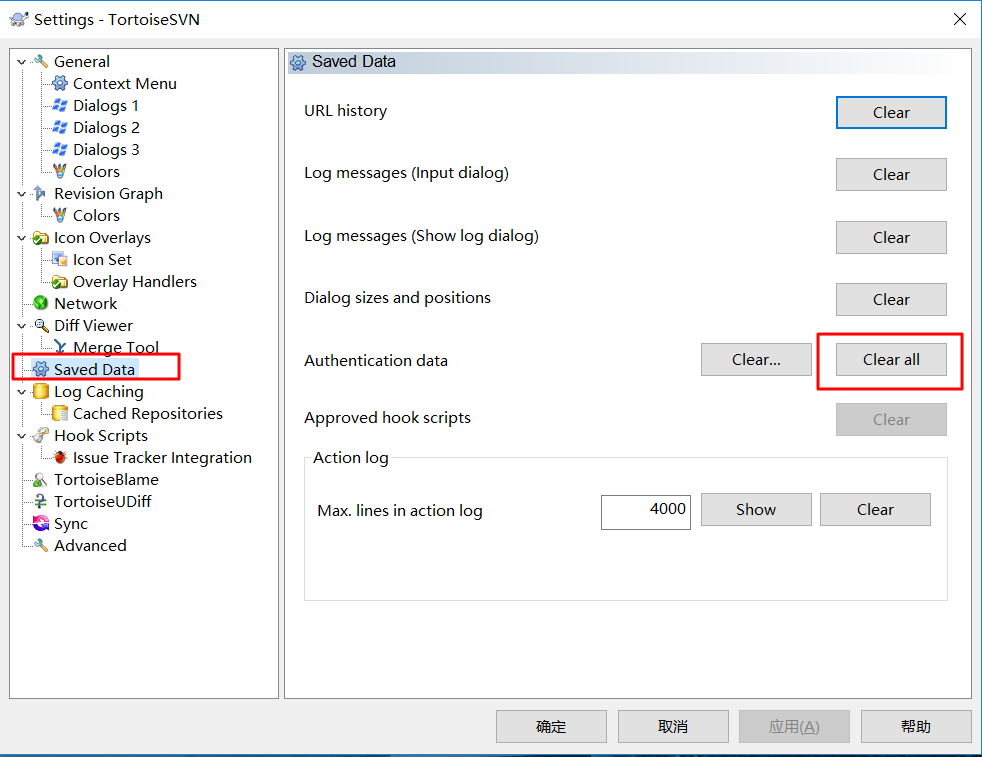
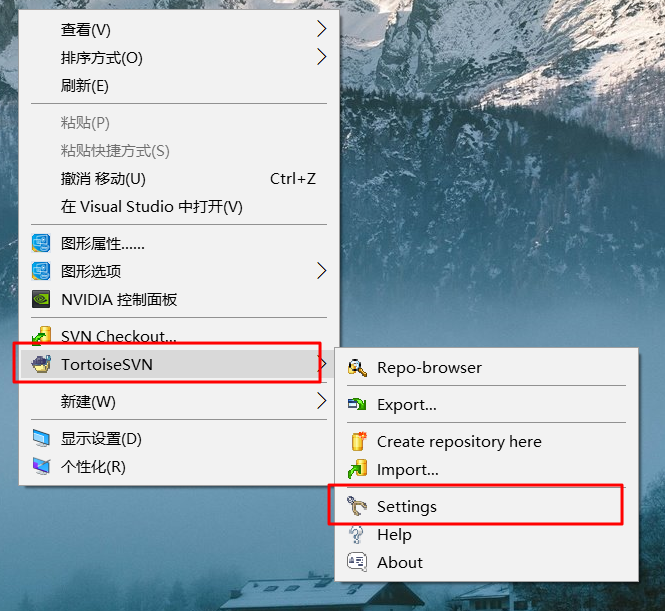


手动merge后，标记为解决



## Eclipse插件，重置用户名密码

只需要将自带的SVN客户端配置清除即可，eclipse中的svn自动断开连接



删除C:\Users(用户)\"你的用户名"\AppData\Roaming\Subversion\auth\下的所有文件

