

# 1.1相关知识点

---

## 1.1.1 SVN的概述

### 1.1.1.1 项目管理中版本控制的问题

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：

- 代码管理混乱
- 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

### 1.1.1.2 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到定案的过程。是一种记录多个文件内容变化，以便将来查阅特定版本修订情况的系统。

### 1.1.1.3 主流的版本控制

VSS:Visual Source Safe(Microsoft Visual Studio成员)主要任务是负责项目文件的管理 CVS:march-hare出品的一套用于进行文件版本控制软件

SVN:Apache软件基金会名下的一套用于进行文件版本控制软件

在2000年初，开发人员要写一个CVS的自由软件代替品，它保留CVS的基本思想，但没有它的错误和局限，保留CVS的基本特性但去除CVS的bug和不好的特性。

在2000年2月，他们联系《使用CVS开发开源项目》(Open Source Development with CVS) (Coriolis, 1999)的作者Karl Fogel,并征求了他是否愿意在这个新的项目中担任一个角色。巧合的是，当时Karl已经和他的朋友Jim Blandy讨论了一个关于新的版本控制系统的设计。在1995年，这两人就成立了Cyclic Software, 一个提供CVS的商业支持的软件公司。虽然他们经营商业服务，但是仍然在每天的工作中使用CVS。使用CVS的挫折感使得jim认真思考更好的方法来管理数据，不但确定名字为"Subversion",而且完成了Subversion档案库的基础设计。

当CollabNet的电话到来时，Karl立即答应了加入项目中，而且Jim让他的雇主RedHat Software同意让他在这个项目中不定期工作。CollabNet雇用了Karl和Ben Collins-Sussman,并在5月开始了详细设计工作。在得到了来自CollabNet的BrianBehlendorf、JasonRobbms和Greg Stem (当时是一名活跃在WebDAV/DeltaV规范过程的自由程序员)很多创意的帮助下，Subversion很快地引起了一个活跃开发者社区的注意。它找出并欢迎很多同样在CVS上受到挫折的社员能来为这个项目做些什么。

Subversion最初的设计Team定下了几个简单的目标。它必须在功能上可取代CVS,也就是说，所有CVS可做到的事，它都要能够作到。在修正最明显的瑕疵的同时，还要保留相同的开发模式。还有，Subversion应该要和CVS很相像，任何CVS使用者只要花费少许的力气，就可以很快地上手。

经过十四个月的编码后,Subversion于2001年8月31日开始实现“自行管理”。也就是说，开发人员不再使用CVS来管理Subversion的代码，而以Subversion自己来管理。

2009年11月，Subversion被Apache Incubator 专案所接收。

2010年1月，正式成为Apache软件基金会的一个顶级专案，所以为Apache Subversion。  
目前 Apache Subversion 的主席为 Greg Stein,项目领导者 Release manager 为 Wandisco 公司。

### 1.1.1.4 什么是 SVN

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

编辑

讨论

上传视频

本词条由“科普中国”科学百科词条编写与应用工作项目 审核。

SVN是subversion的缩写，是一个开放源代码的版本控制系统，通过采用分支管理系统的高效管理，简而言之就是用于多人共同开发同一个项目，实现共享资源，实现最终集中式的管理。 [1]

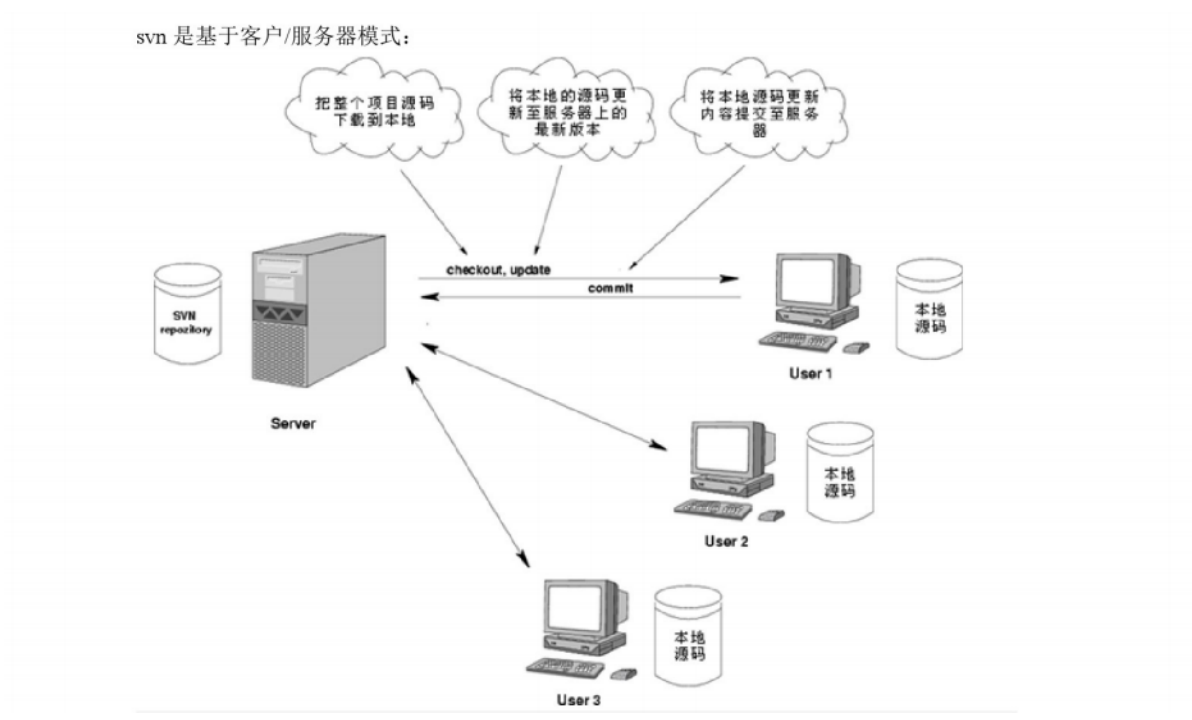
中文名	SVN	优势	速度快、安全性高等
外文名	Subversion	应用	软件程序开发
实质	开放源代码的版本控制系统	学科	软件工程

### 1.1.1.5 SVN的作用

针对软件研发企业的软件生产过程而言，SVN用于管理整个开发过程中的源码，进行版本控制。

## 1.1.2 SVN的使用

### 1.1.2.1 SVN的使用方法



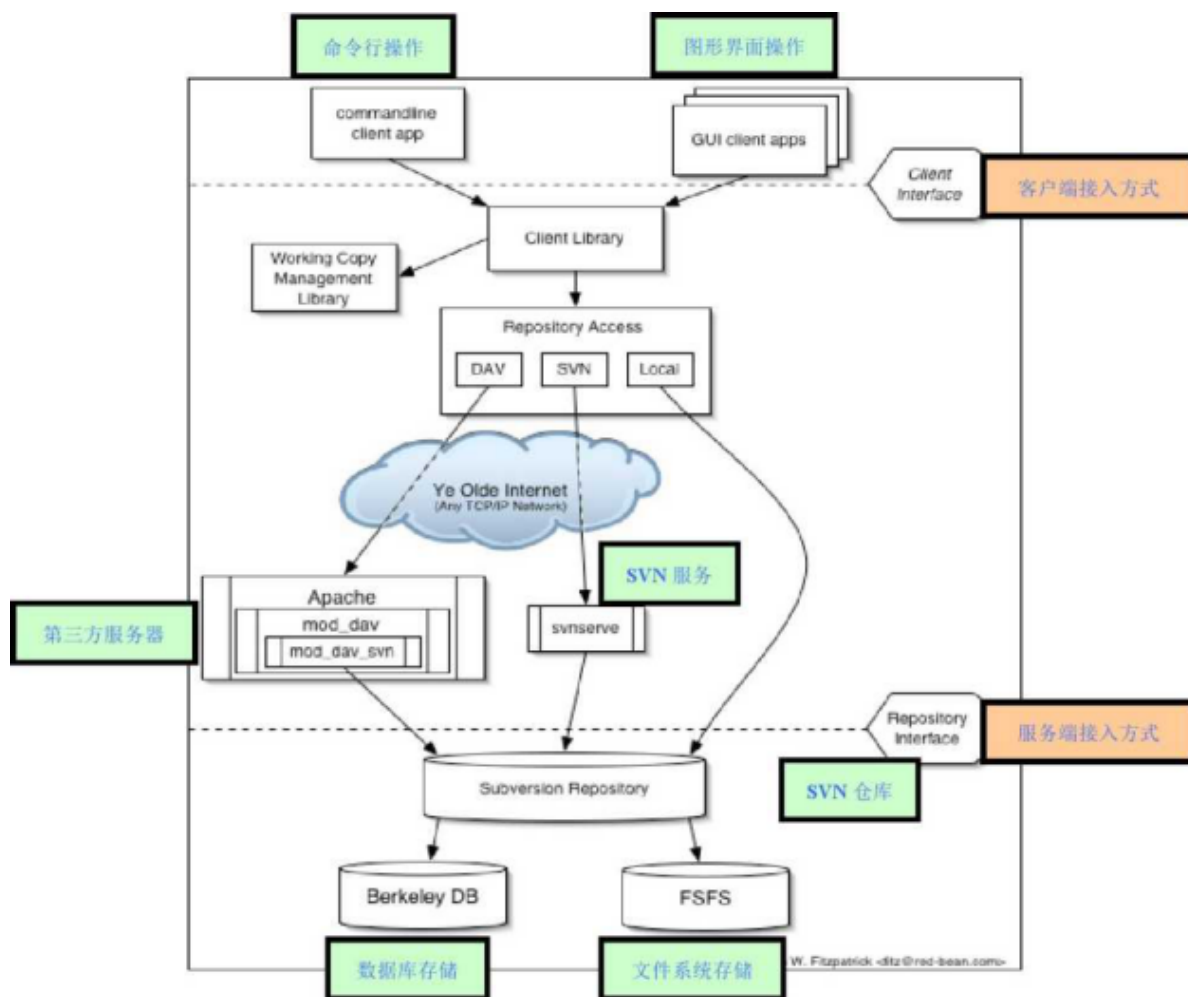
- 复制-修改-合并方案（Subversion默认的模式）

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。

### 1.1.2.2 SVN的工作方式



### 1.1.3 SVN服务的安装

#### 1.1.3.1 官方安装包

官方网站: <http://subversion.apache.org/>

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

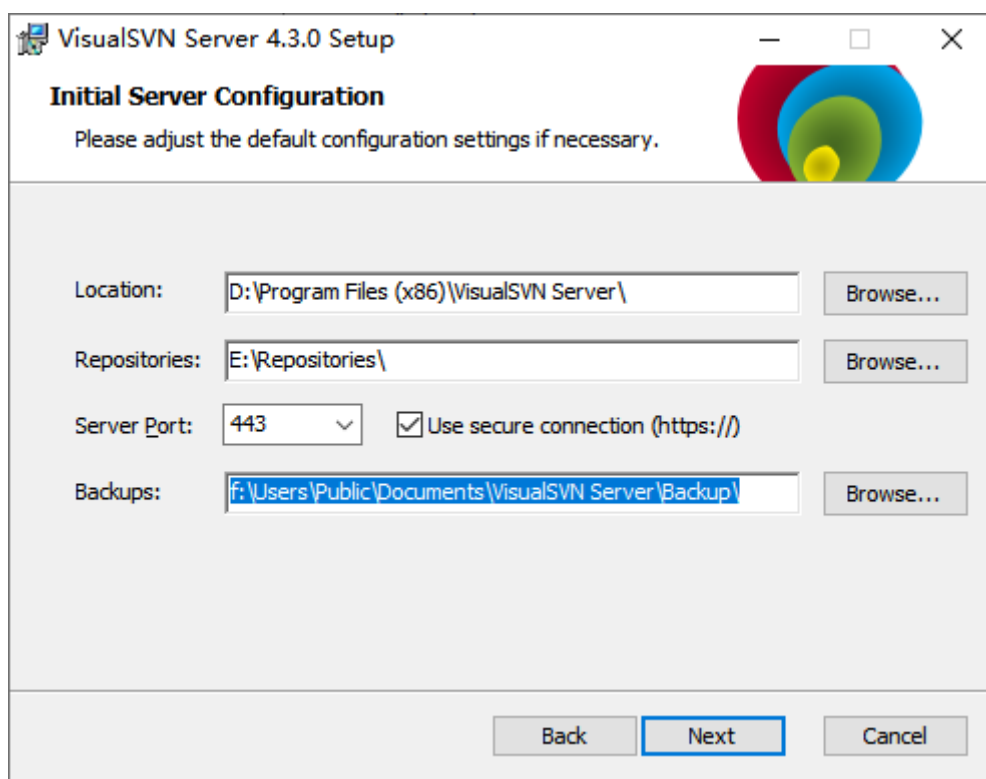
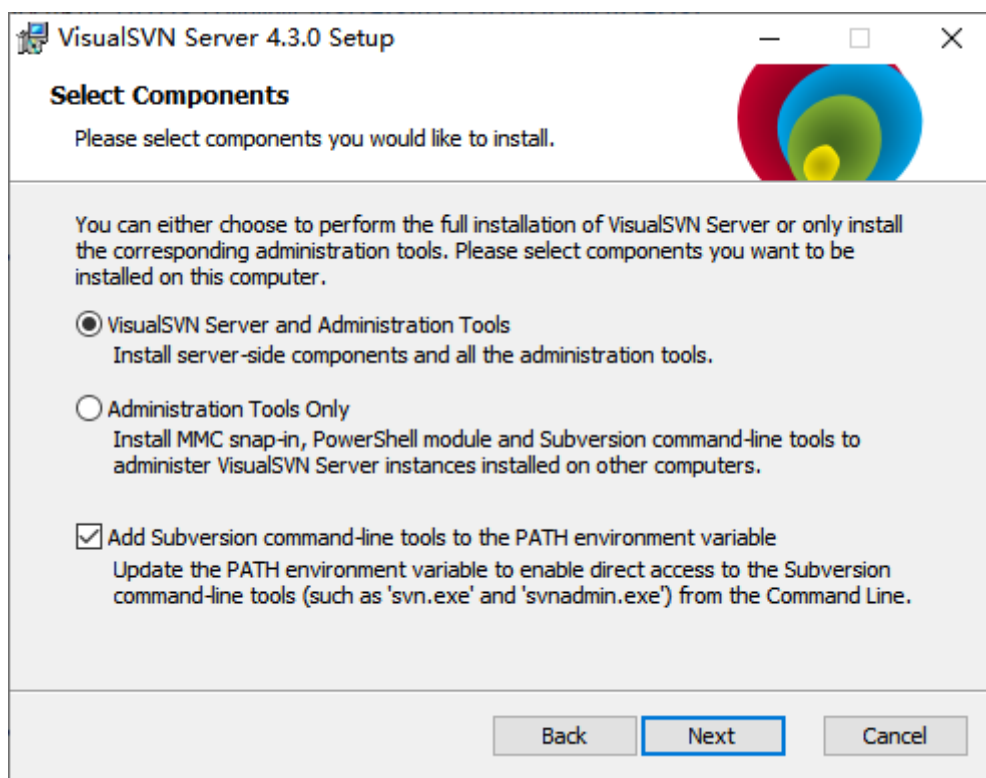
官方提供的服务端安装包，安装后需要通过命令行操作，适用于专业配置管理员使用。

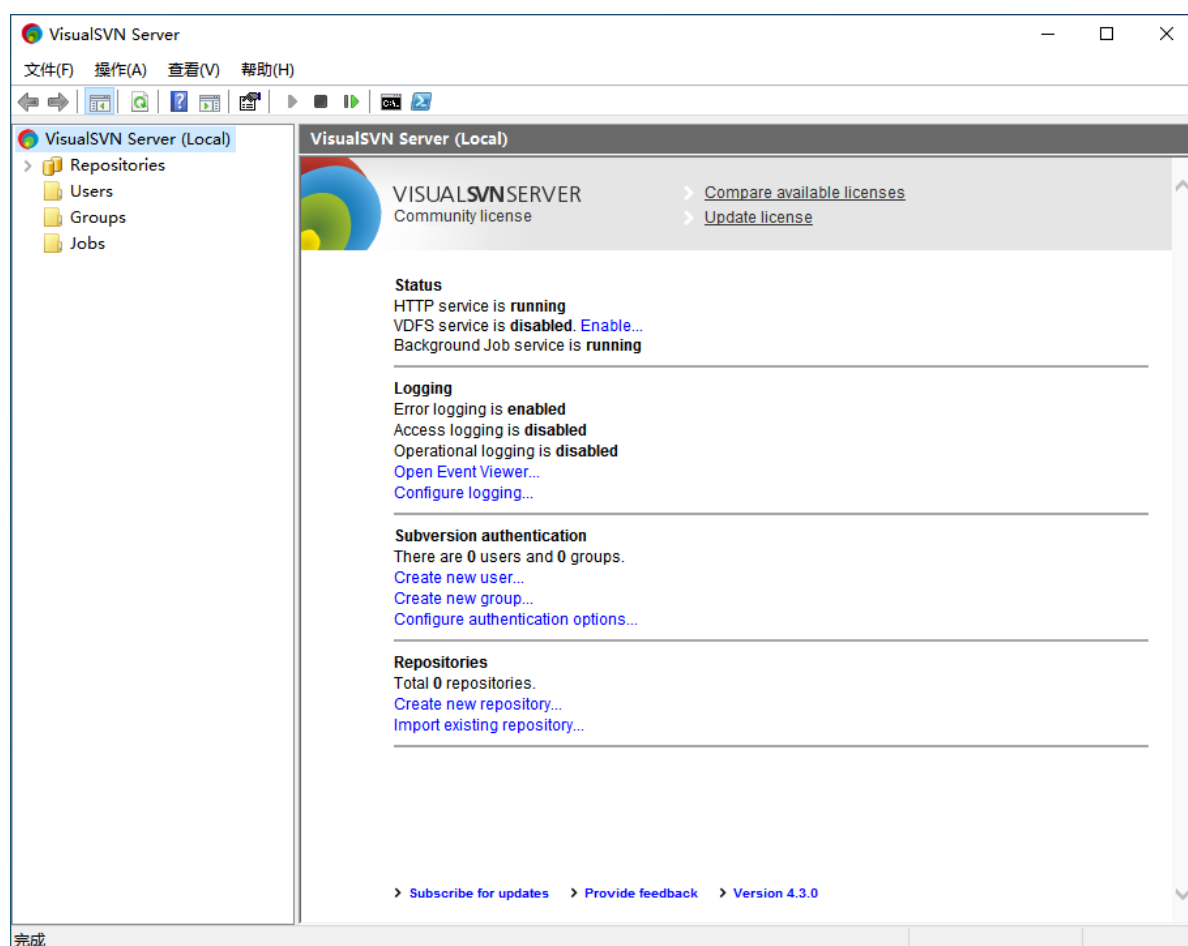
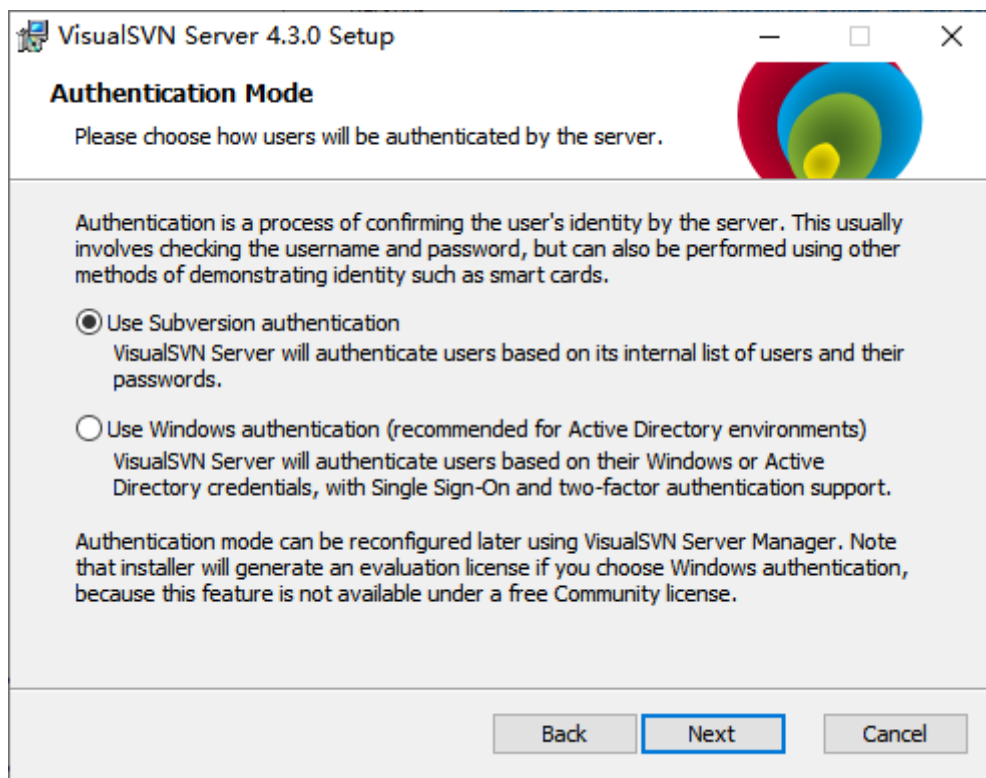
#### 1.1.3.2 图形化服务端

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。

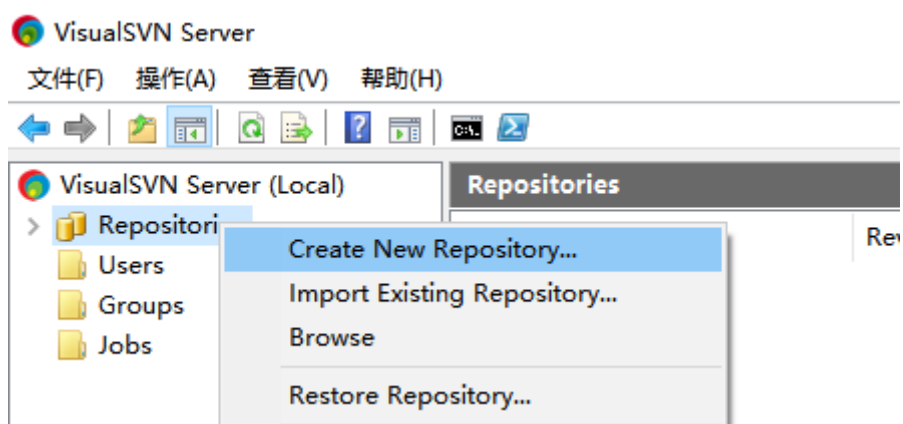
下载地址: <https://www.visualsvn.com/downloads/>

### 1.1.3.3 安装图形化服务端

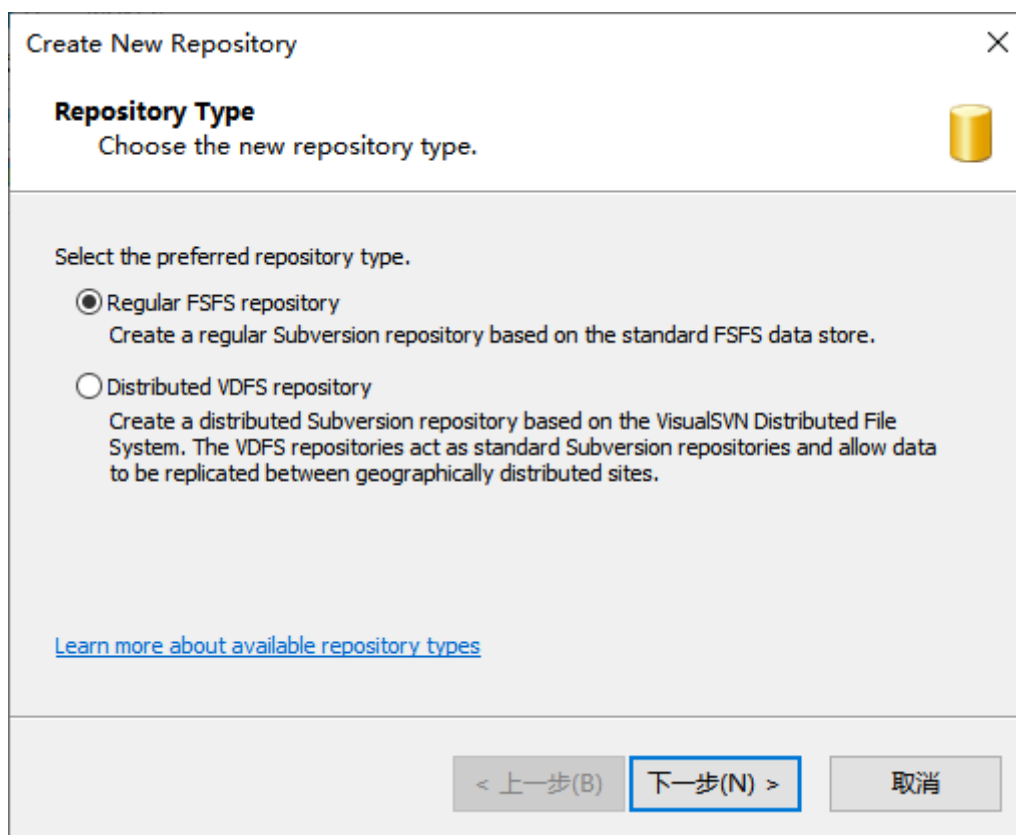




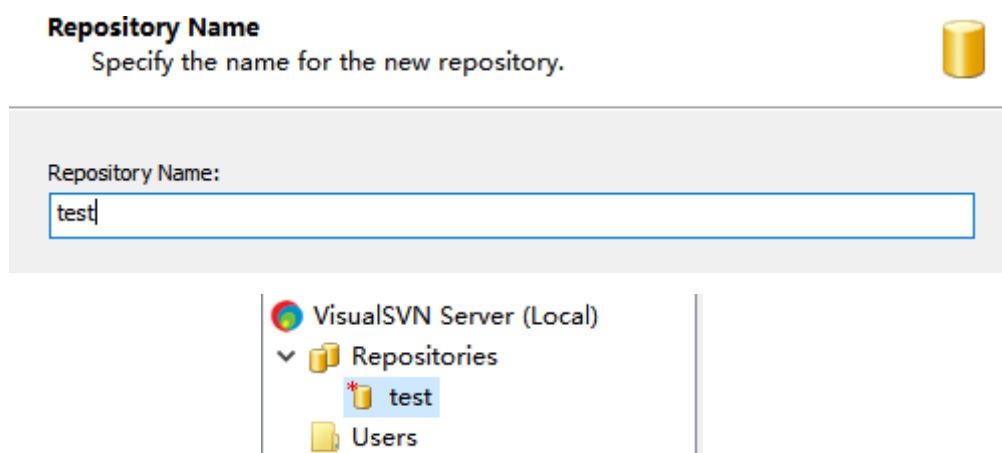
### 1.1.3.4 创建仓库



- 选择仓库存储方式

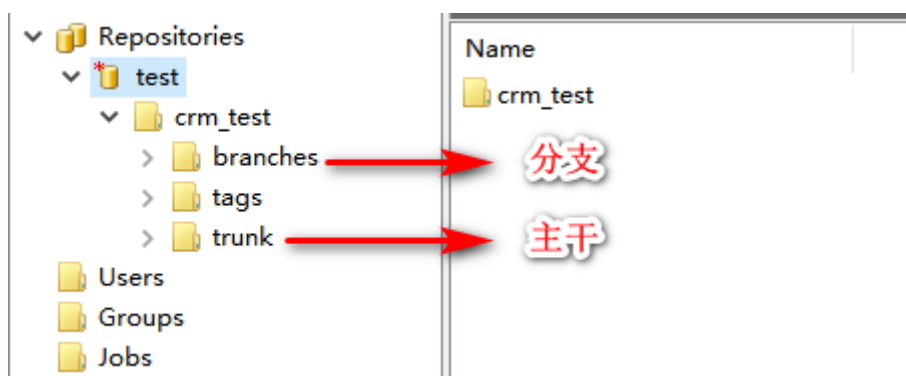
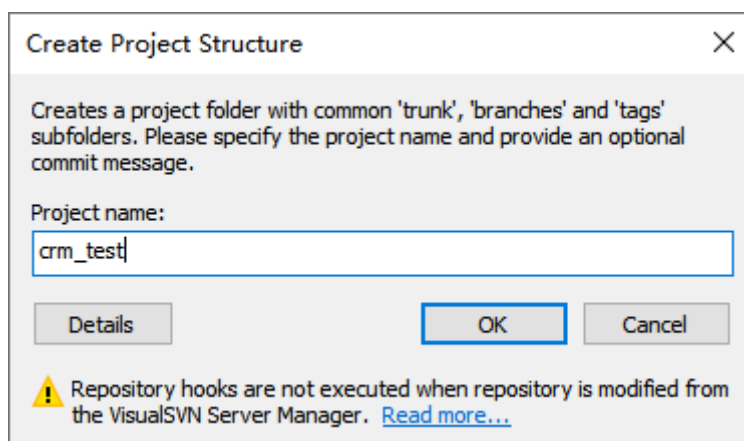
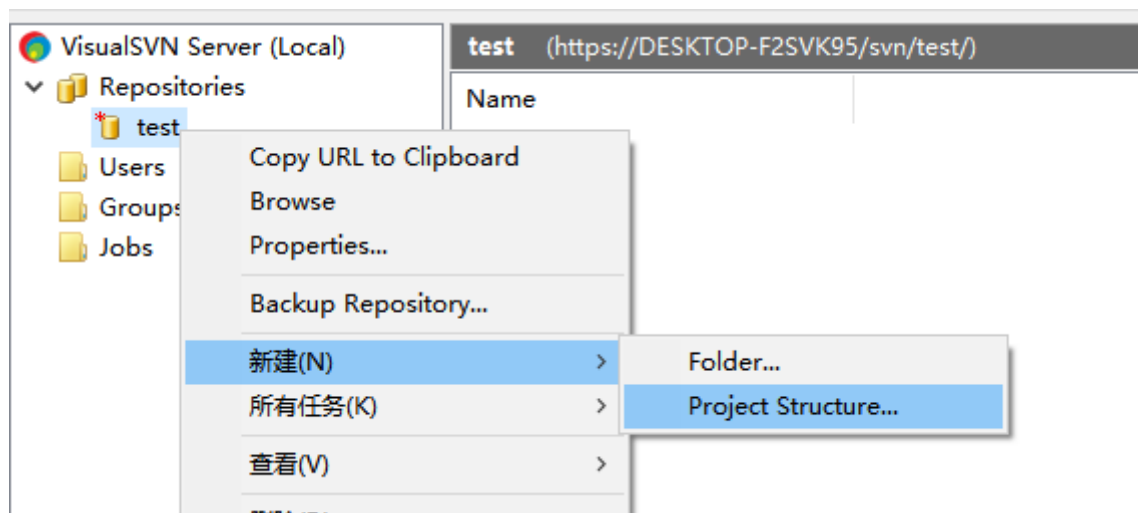


- 输入仓库名称



### 1.1.3.5 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



## 1.1.4 TortoiseSVN 客户端

### 1.1.4.1 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TortoiseSVN图形化界面操作（推荐）
3. 使用Eclipse等开发工具插件操作（推荐）

### 1.1.4.2 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。

### 1.1.4.3 浏览仓库

使用Tortoise浏览svn服务端的仓库的内容：



## 1.1.5 权限管理（了解）

### 1.1.5.1 认证授权机制

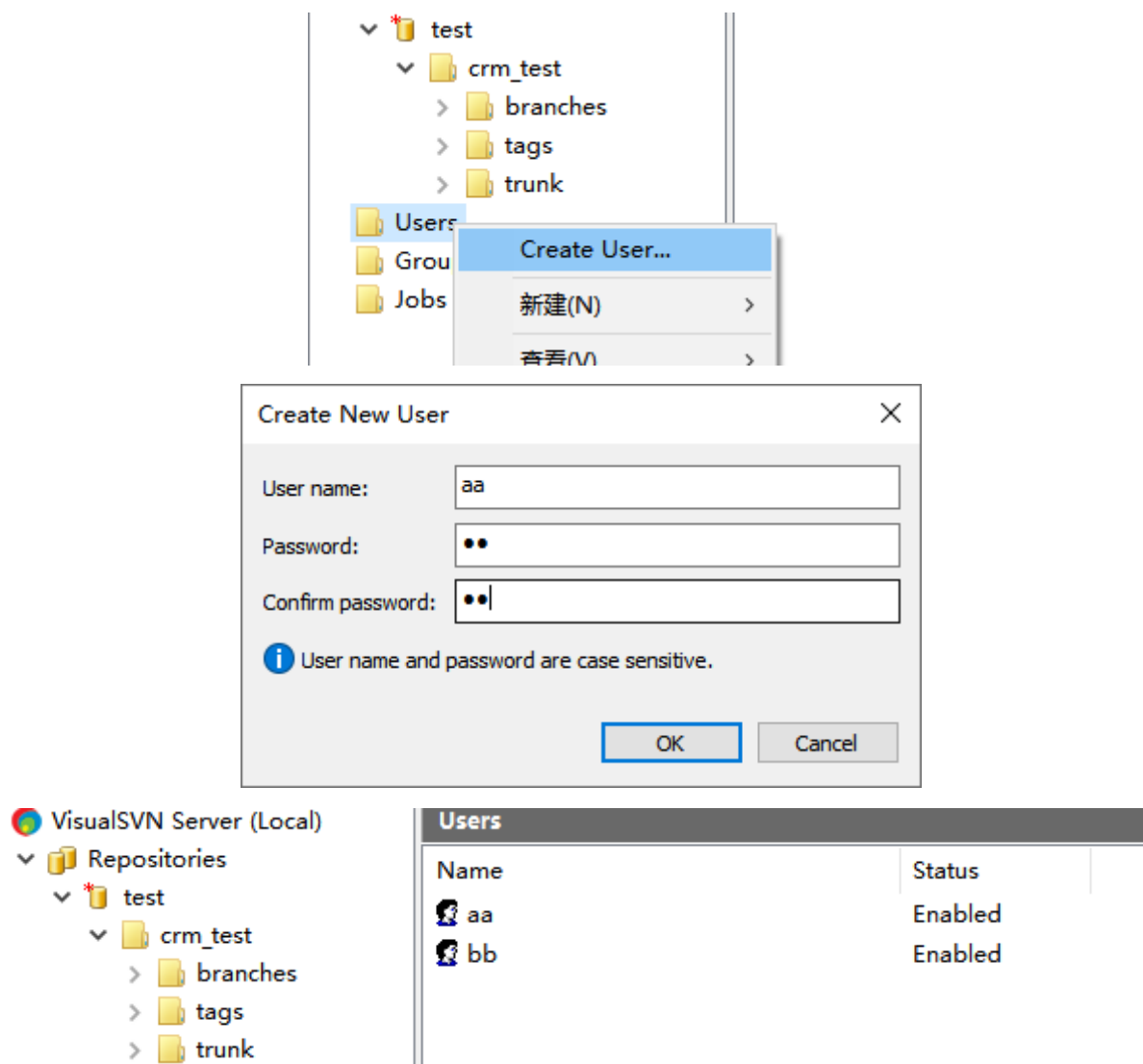
在企业开发中会为每位程序员、测试人员等相关人员分配一个账号，用户通过使用svn客户端 连接svn服务时需要输入账号和密码，svn服务对账号和密码进行校验，输入正确可以继续访问，当 用户访问仓库下某个目录时，svn服务对用户进行授权，如果用户拥有该目录的访问权限方可访问。

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。



### 1.1.5.2 创建用户



### 1.1.5.3 创建组

## 1.1.6 分配权限

### 1.1.6.1 删除默认权限

### 1.1.6.2 示例一：开发人员拥有读写权限

### 1.1.6.3 示例二：测试人员拥有读权限

### 1.1.7 TortoiseSVN 日常使用

### 1.1.7.1 浏览仓库

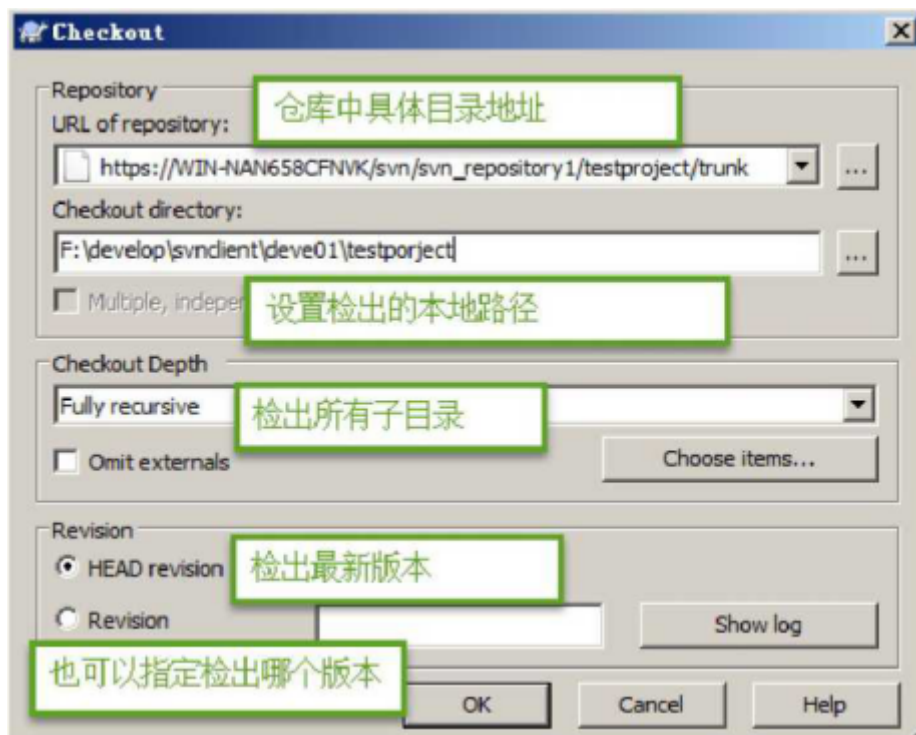
### 1.1.7.2 导入导出

### 1.1.8 修改提交

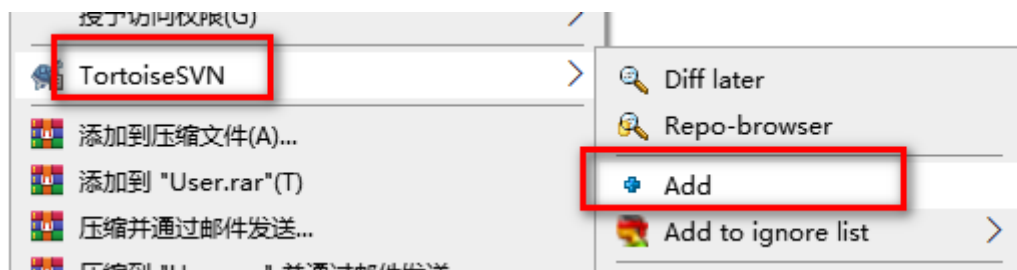
### 1.1.8.1 Checkout

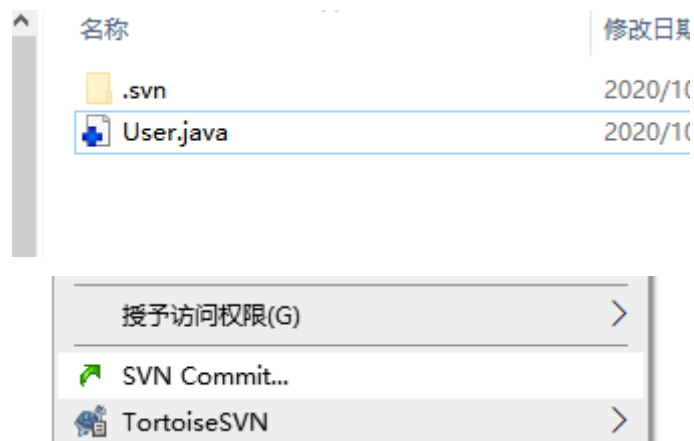
检出项目，复制项目的副本到本地。在要检出的目录中右键：



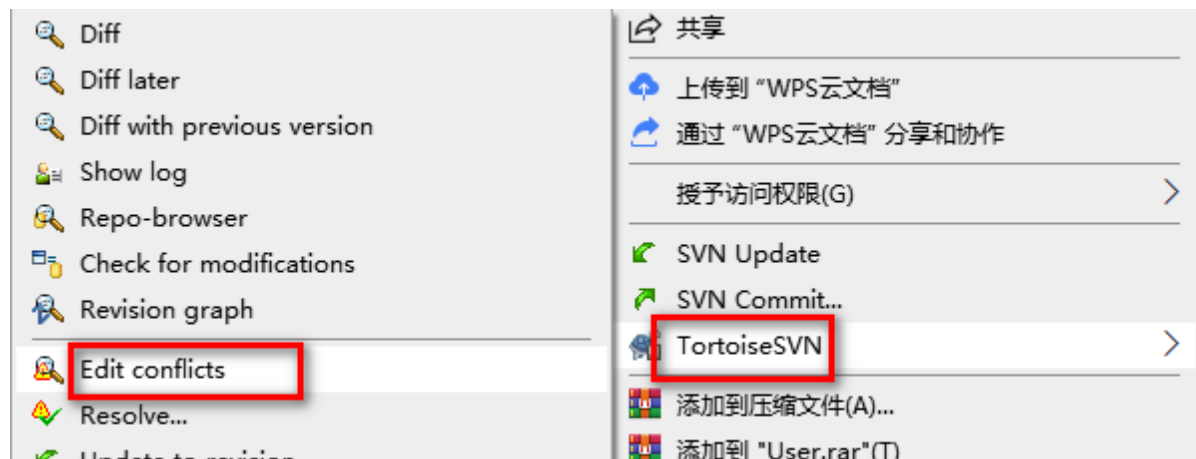
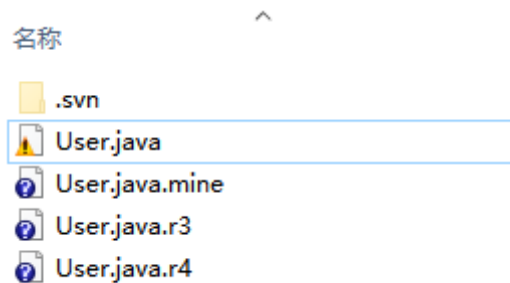


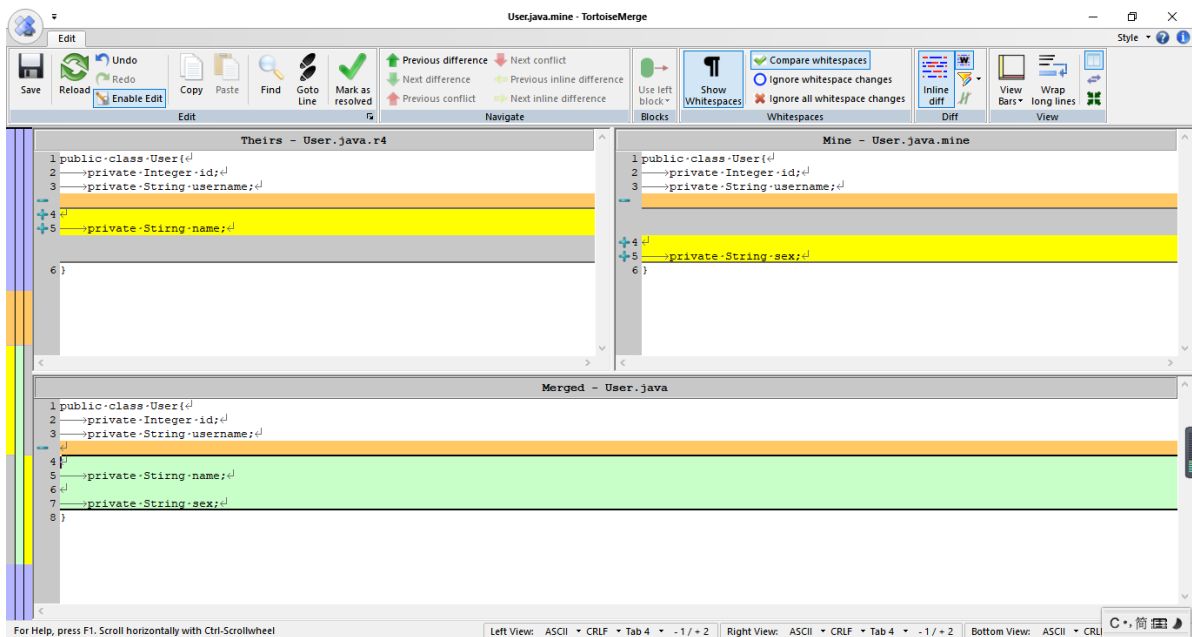
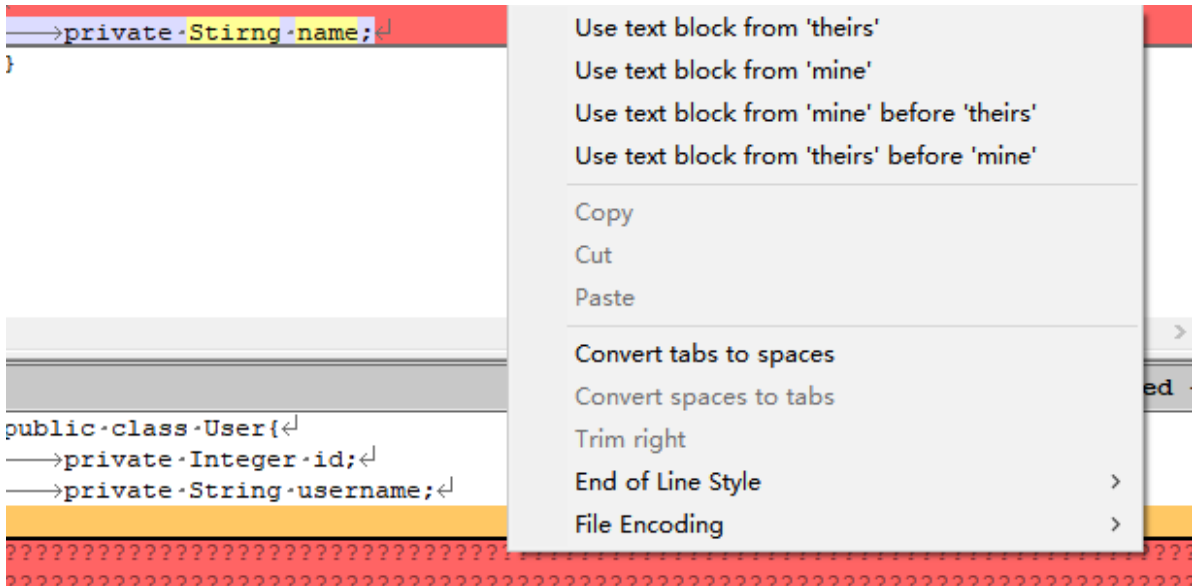
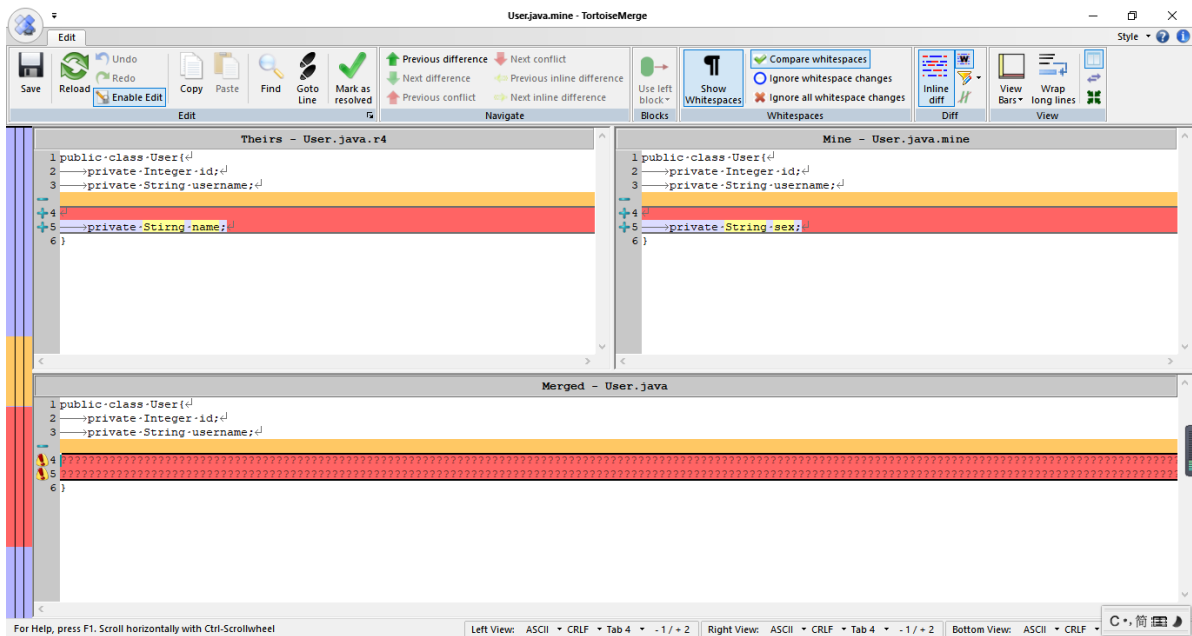
### 1.1.8.2 add





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



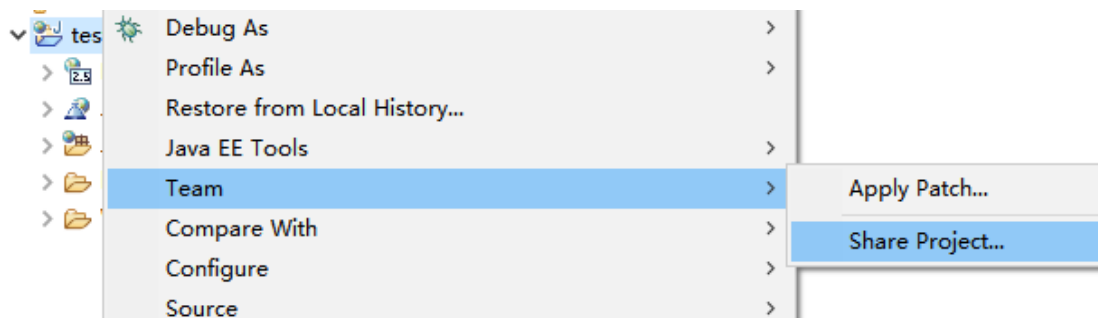


## 1.1.10 eclipse 的 SVN 插件使用

### 1.1.10.1 svn插件安装

下载 Subversion 的 eclipse 插件 <http://subclipse.tigris.org/servlets/ProjectProcess?Dag:eID=D4wYu>  
[A](#)

### 1.1.10.2 将项目共享到SVN



### 1.1.10.3 从 svn 检出



#### 1.1.10.4 解决冲突