



SVN概述

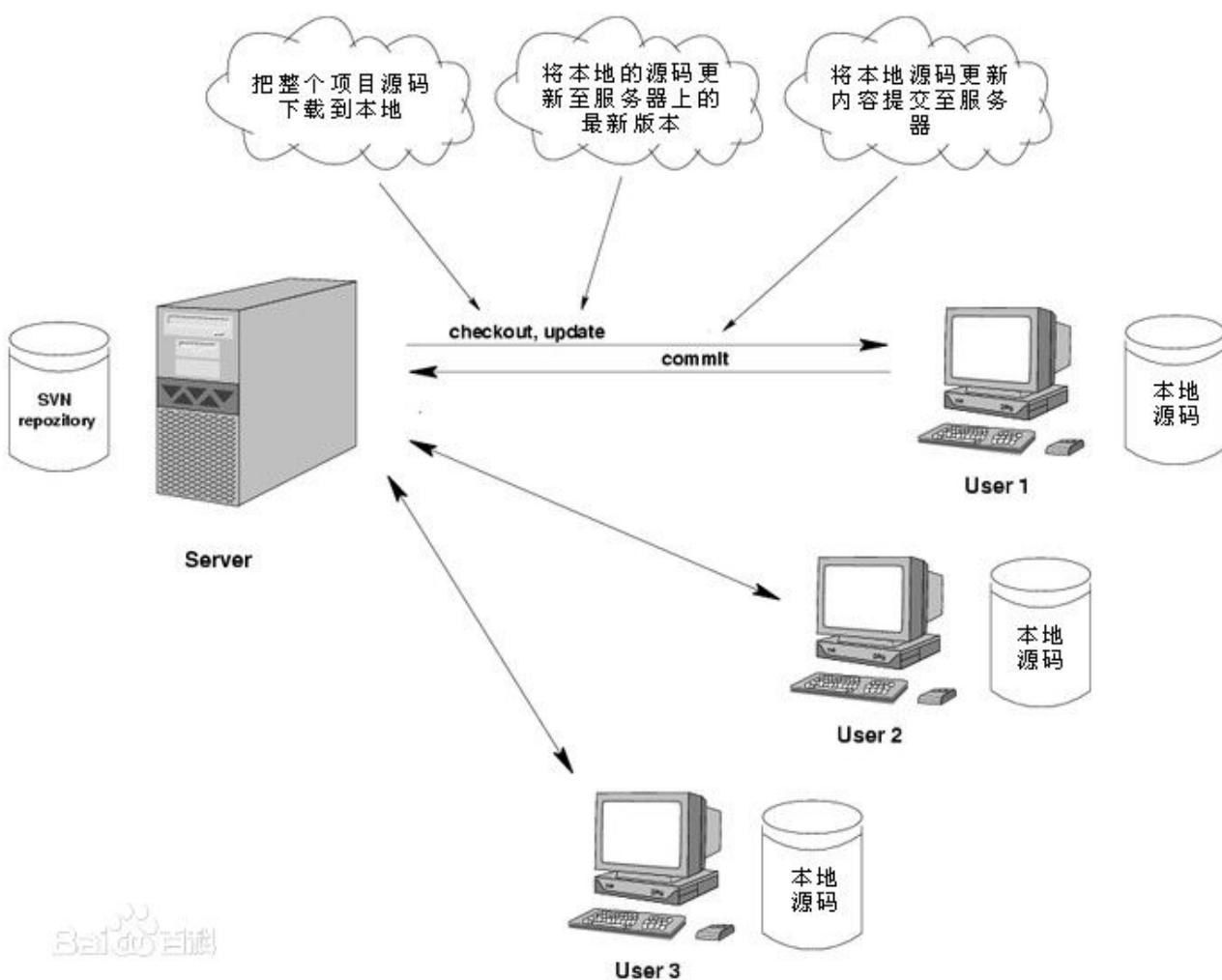
1.SVN介绍

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。

1.1 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)

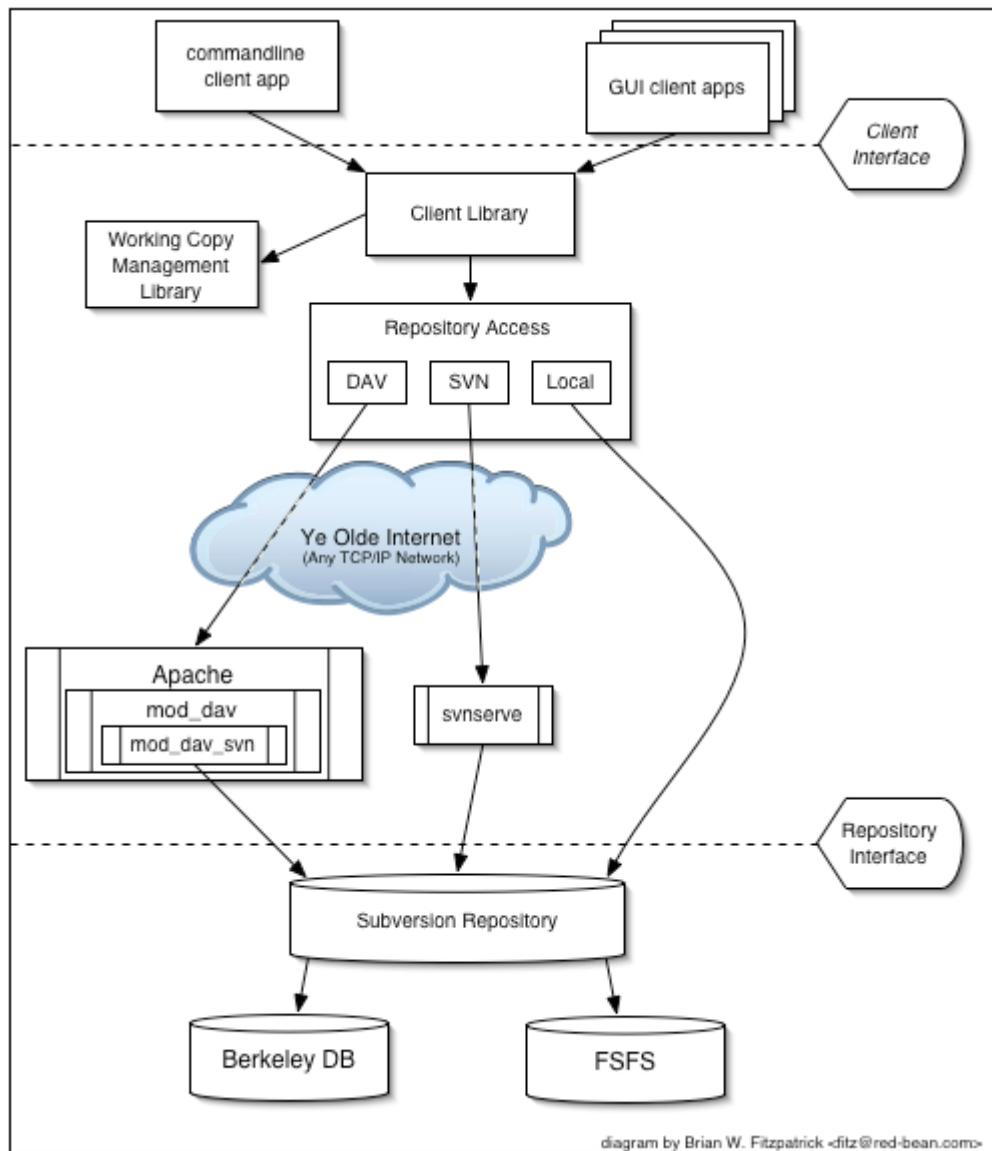
在这种模型里，每一个客户读取项目配置库建立一个私有工作副本——版本库中文件和目录的本地映射。用户并行工作，修改各自的工作副本，最终，各个私有的复制合并在一起，成为最终的版本，这种系统通常可以辅助合并操作，但是最终要靠人工去确定正误。



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。

1.2 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。

2 SVN使用



本章节主要介绍常用的svn工具，如window下svn服务器VisualSVN，客户端工具TortoiseSVN以及idea svn插件的使用

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idea的Svn插件，以及TortoiseSVN（小乌龟）。

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

2.1 服务器端VisualSVN安装与配置

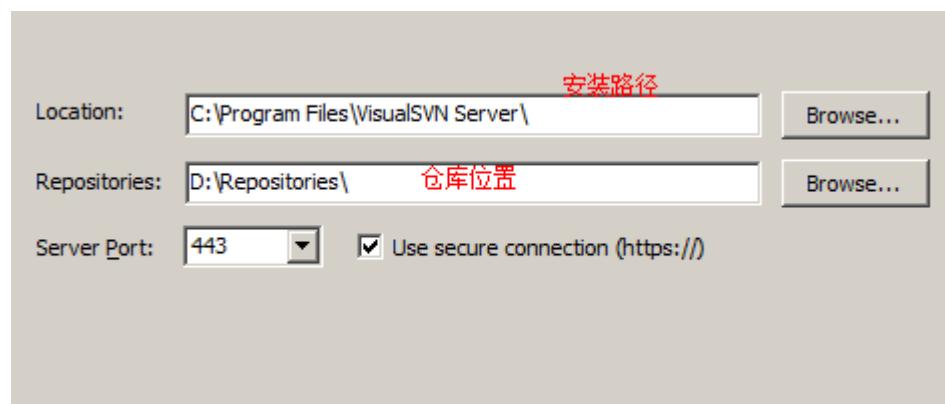
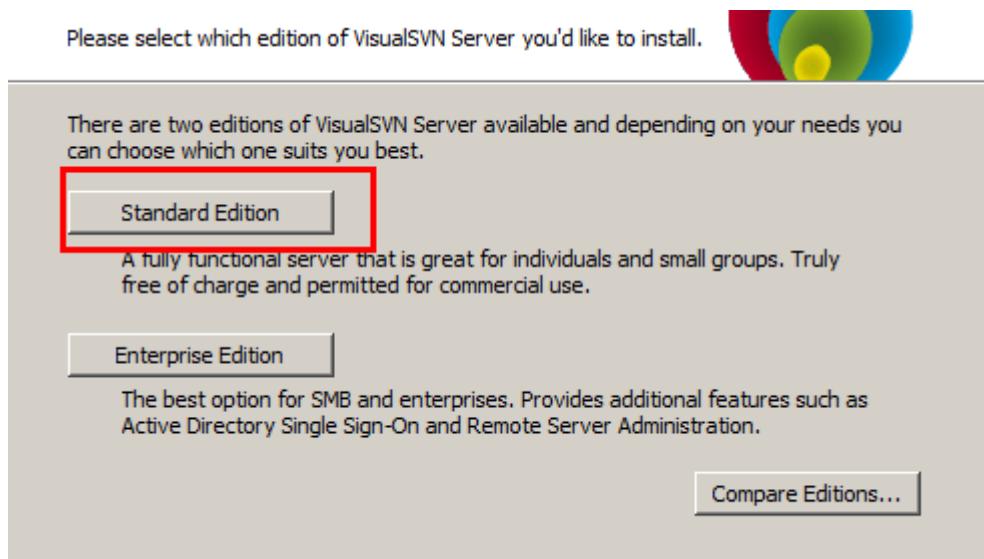
2.1.1 VisualSVN下载

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](#))
- [WANDisco](#) (32- and 64-bit client and server; supported and certified by [WANDisco](#); requires registration)

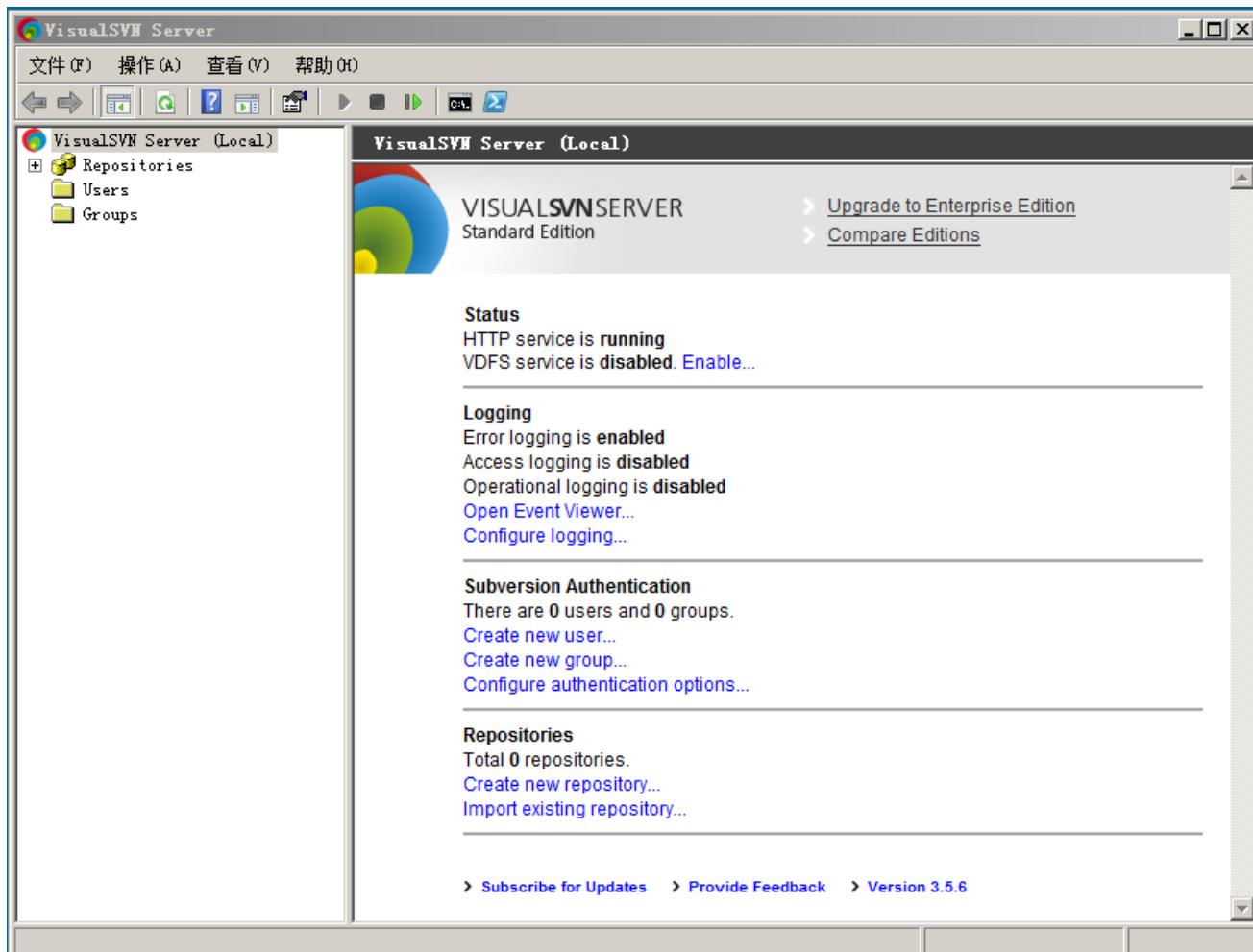
2.1.2 VisualSVN安装与创建仓库



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



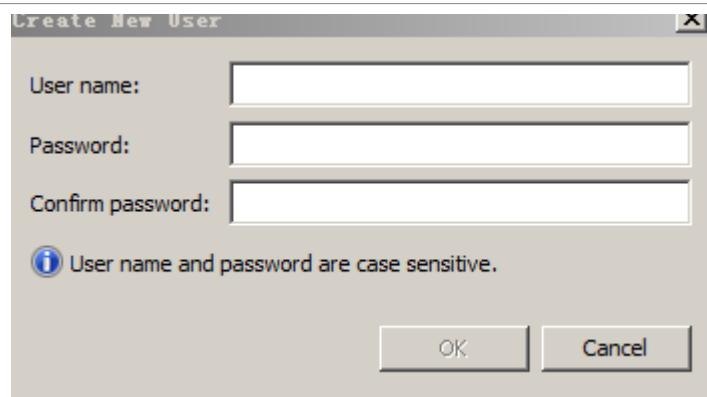
2.1.3 VisualSVN创建工程目录



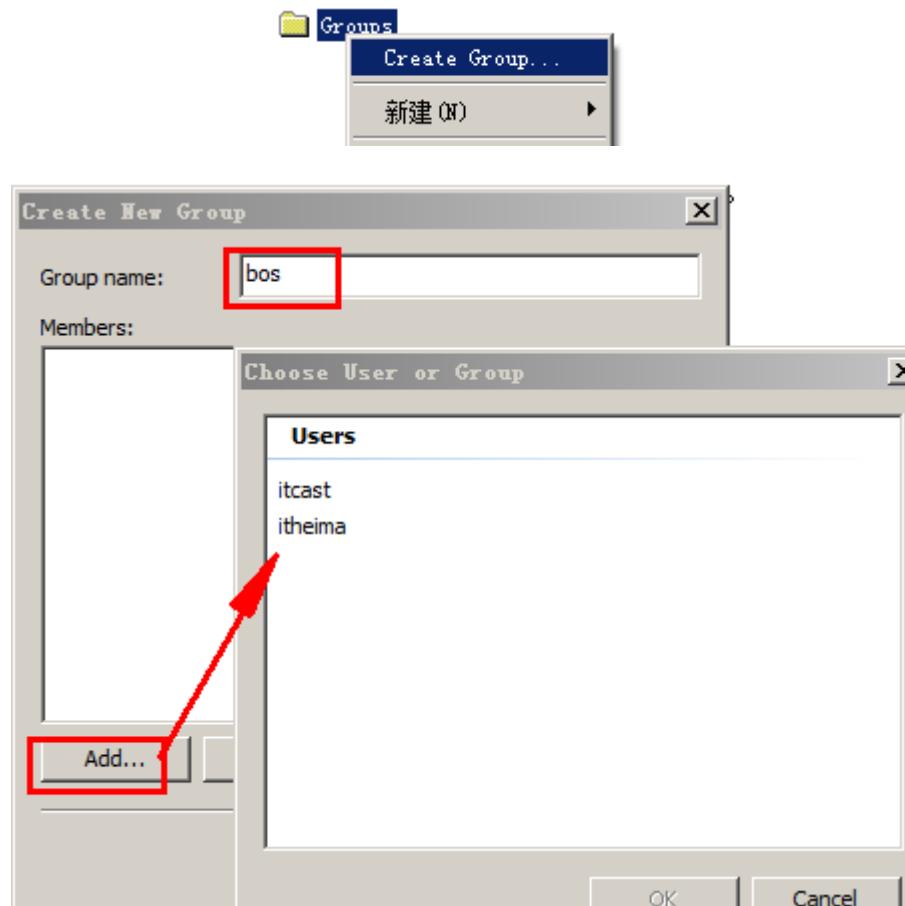
2.1.4 VisualSVN 权限控制

创建用户

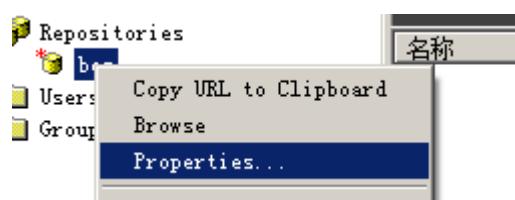


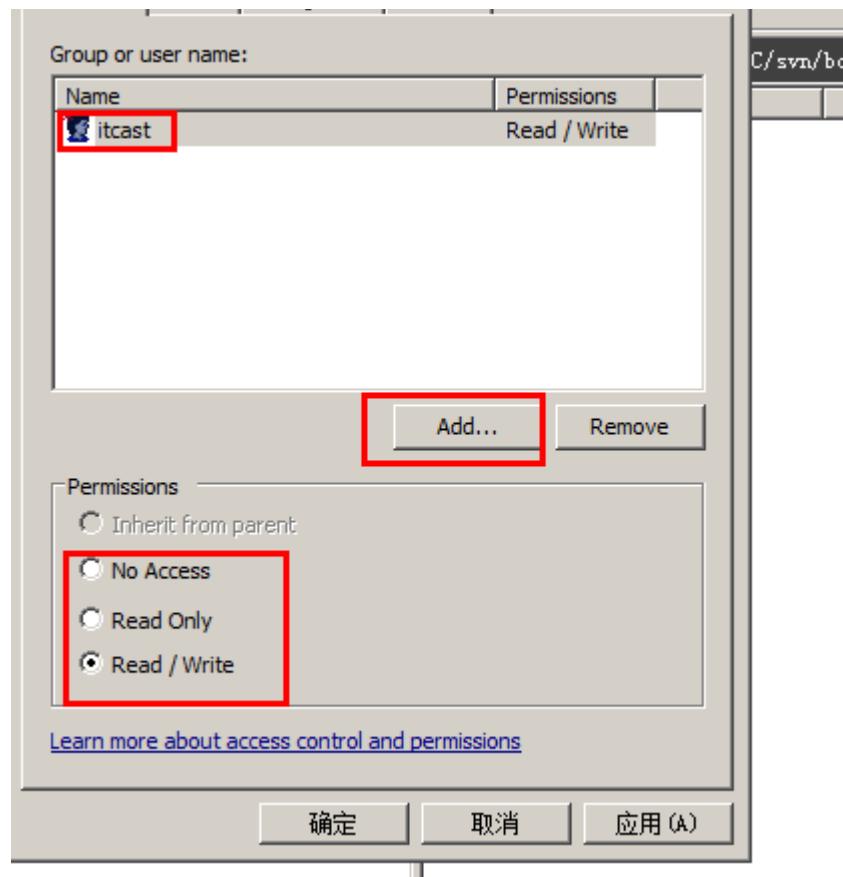


创建组



分配权限





2.2 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。

2.2.1 Tortoise SVN下载

for 32-bit OS

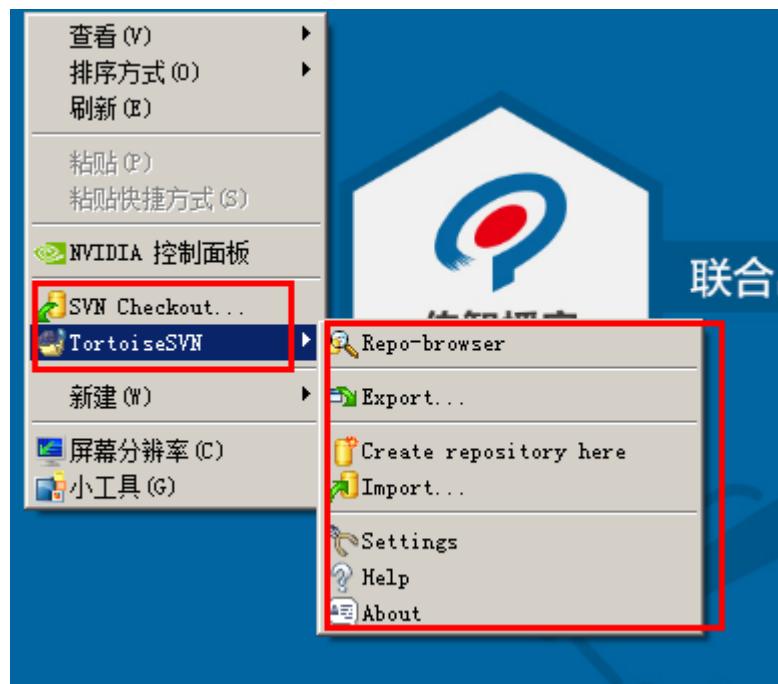
TortoiseSVN 1.9.5 - 32-bit

for 64-bit OS

TortoiseSVN 1.9.5 - 64-bit

2.2.2 Tortoise SVN安装

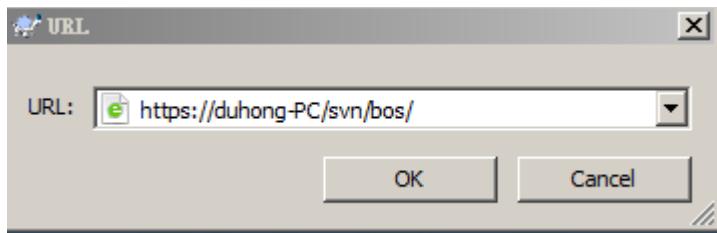
安装后需要重启电脑



2.2.3 Tortoise SVN 常用操作

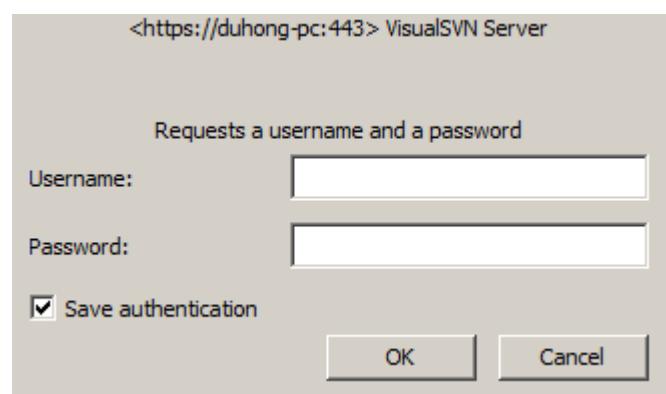
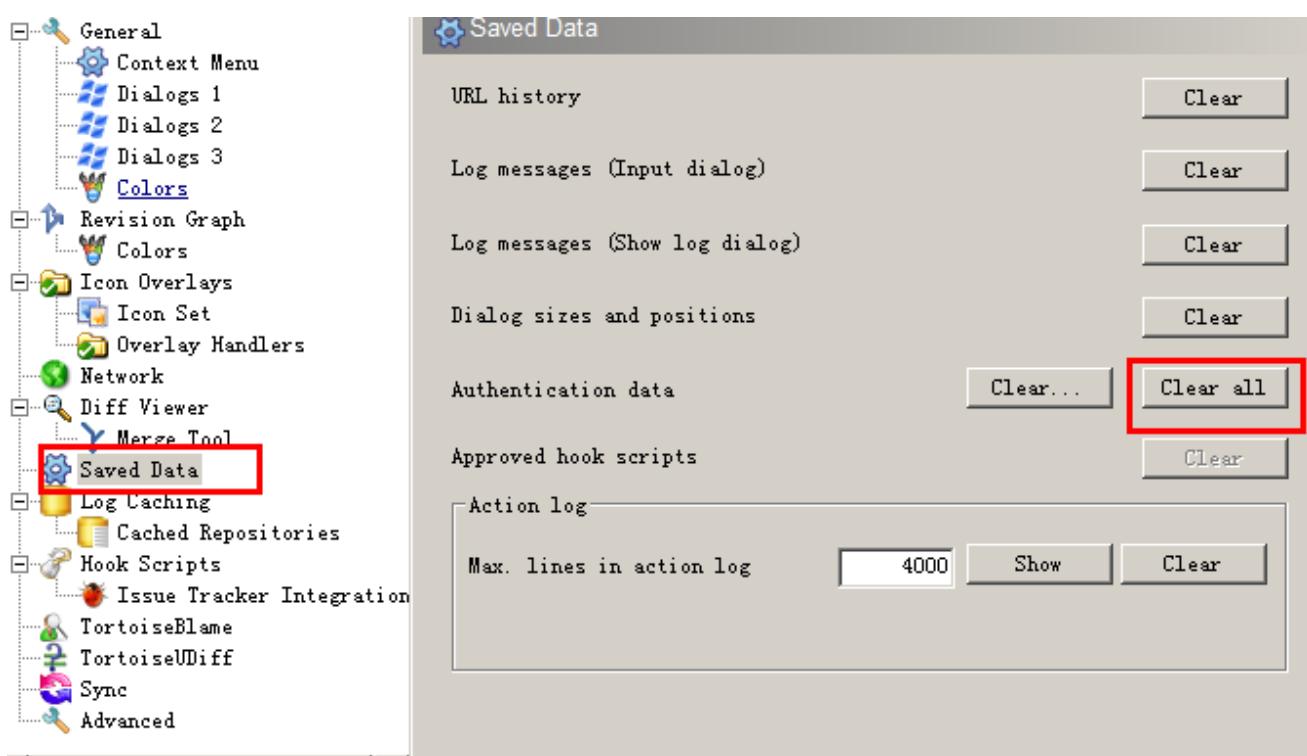
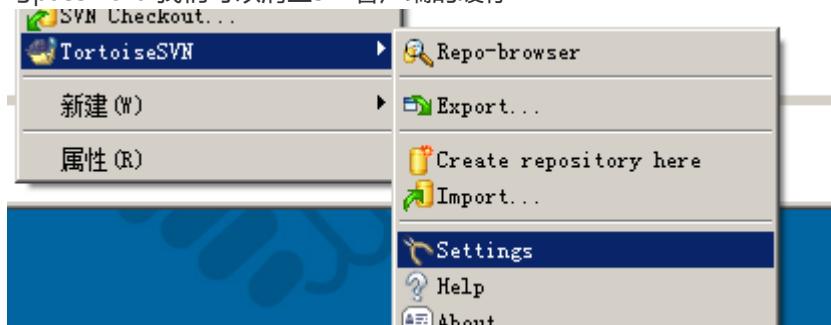
2.2.3.1 浏览仓库





正常情况下，在浏览仓库时会要求录入username

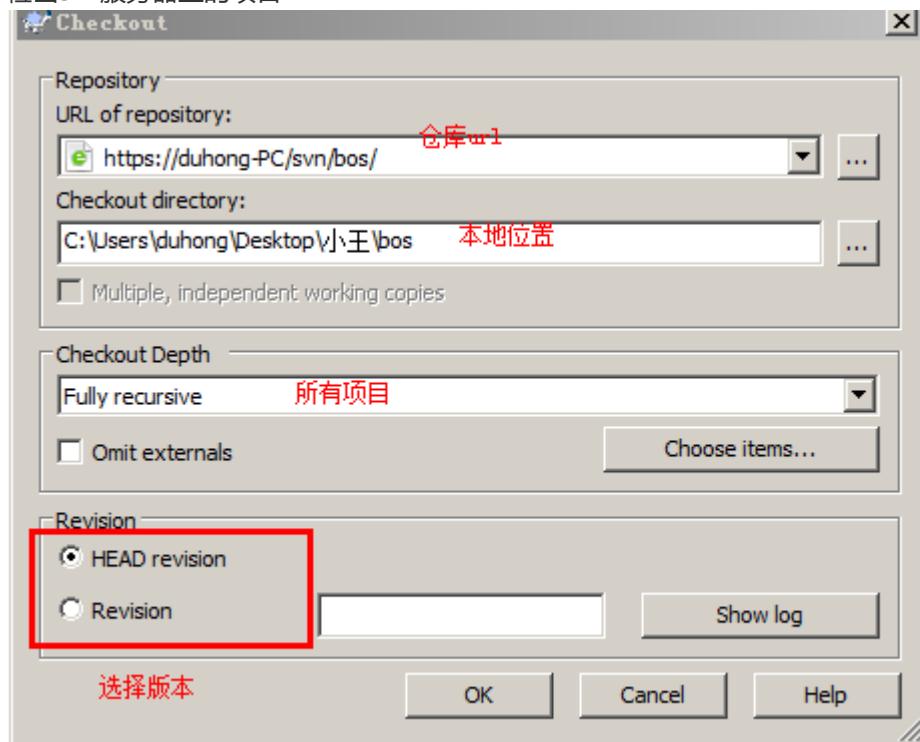
与password 我们可以清空svn客户端的缓存



2.2.3.2 checkout

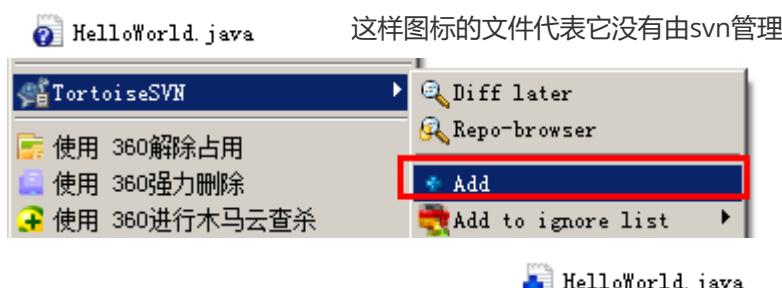


检出svn服务器上的项目



一个.svn隐藏目录 .svn

2.2.3.3 add 添加到本地仓库



这个代表的是它要添加到svn服务器

2.2.3.4 commit





Recent messages

日志信息

Changes made (double-click on file for diff):

Check: All None Non-versioned Versioned Added Deleted Modified Files Directories

Path	Extension	Status	Property status	Lock
<input checked="" type="checkbox"/> HelloWorld.java	.java	added		

添加的文件

Show unversioned files 1 files selected, 1 files total

Show externals from different repositories

Action	Path	Mime type
Command	Commit	
Adding	C:\Users\duhong\Desktop\小王\bos\HelloWorld.java	
Sending content	C:\Users\duhong\Desktop\小王\bos\HelloWorld.java	
Committing transaction...		
Completed	At revision: 1	

版本号

517 Bytes transferred in 0 minute(s) and 2 second(s)



2.2.3.5 update

Update，它是更新操作,可以将svn服务器上的内容更新到本地

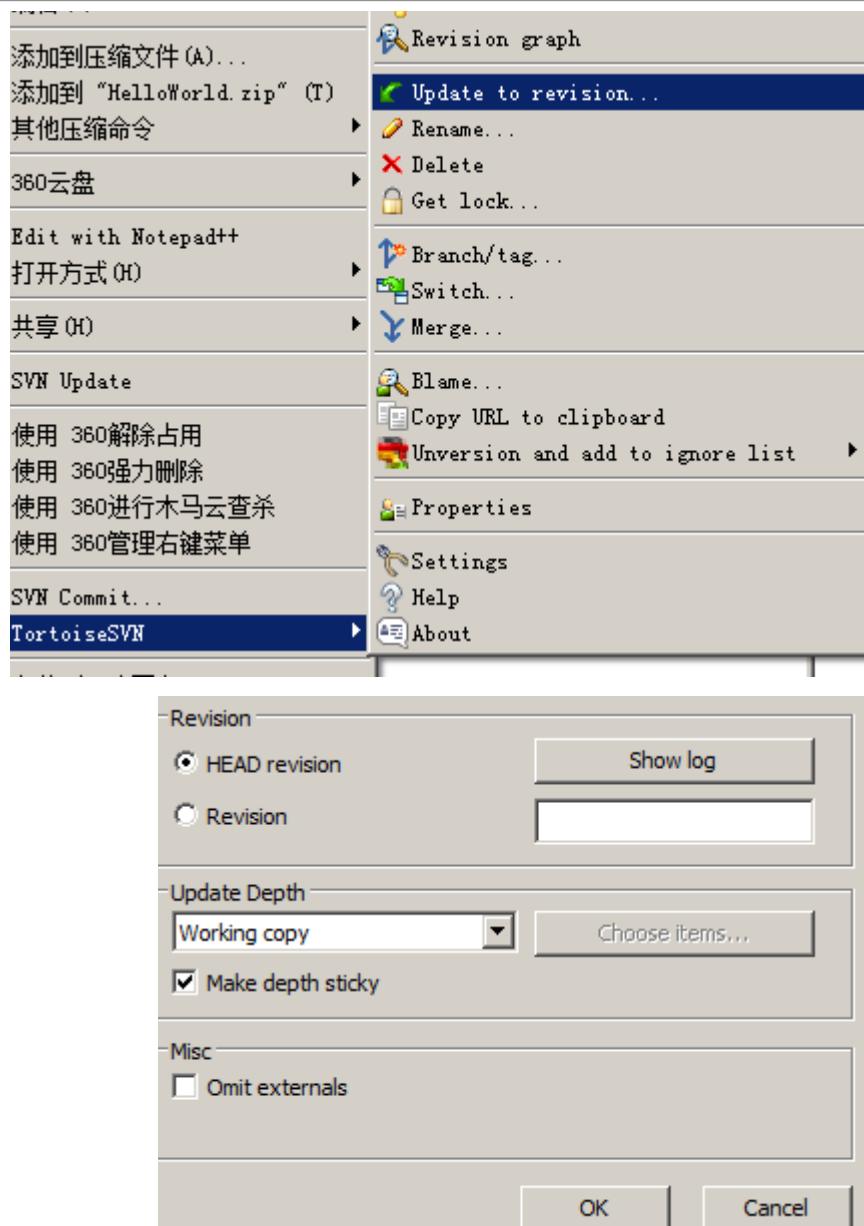


2.2.3.6 更新到历史版本

查看当前日志



也可以通过下面的方式来更新历史版本

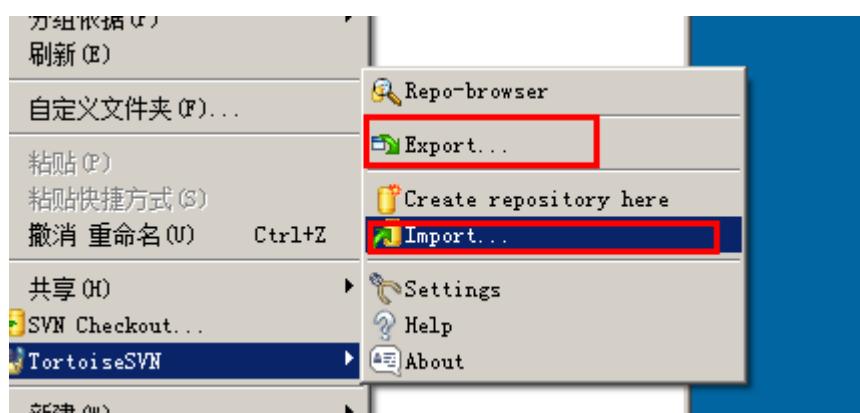


2.2.3.7 delete

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



2.2.3.8 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

2.2.3.9 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	COMMENT
Modified	C:\Users\duhong\Desktop\小��\bos\HelloWorld.java
Sending content	C:\Users\duhong\Desktop\小��\bos\HelloWorld.java
Error	Commit failed (details follow):
Error	File 'C:\Users\duhong\Desktop\小��\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.

edit也行



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

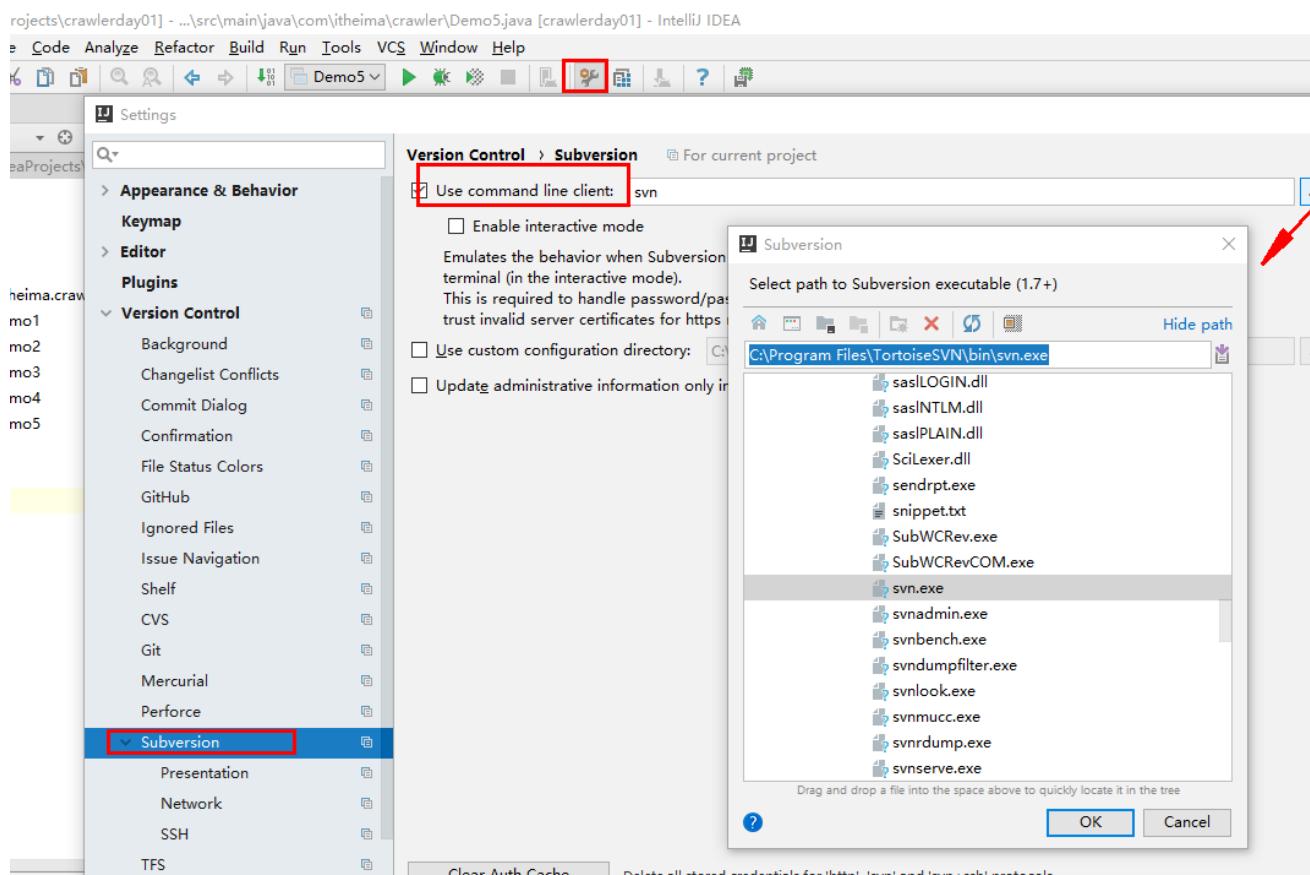
HelloWorld.java.r5 对方修改后的文件



2.3 IDEA下 SVN 配置与使用

2.3.1 IDEA下svn配置

前提是我们的电脑上已经安装了TortoiseSVN.



这样安装就完成了



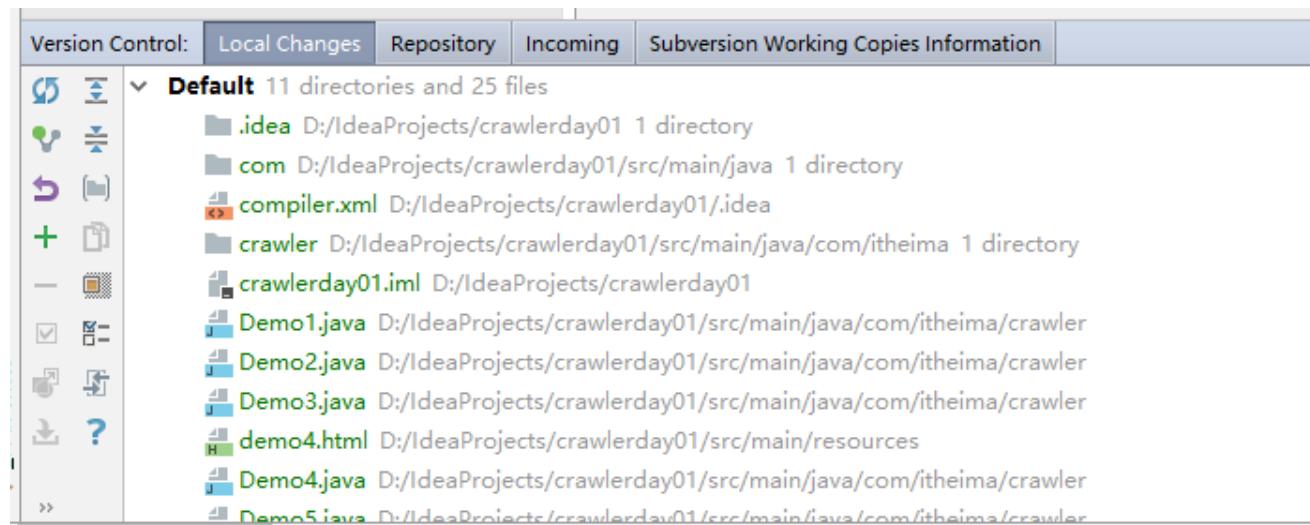
1.Update Project 更新项目

2.Commit changes 提交项目上所有变化文件

3.Compare with the Same Repository Version 当前文件与服务器上该文件同版本的内容进行比较。如果当前编辑的文件没有修改，则是灰色不可点击

4.Show history 显示当前文件的历史记录

5.Revert 还原当前被修改的文件到未被修改的版本状态下





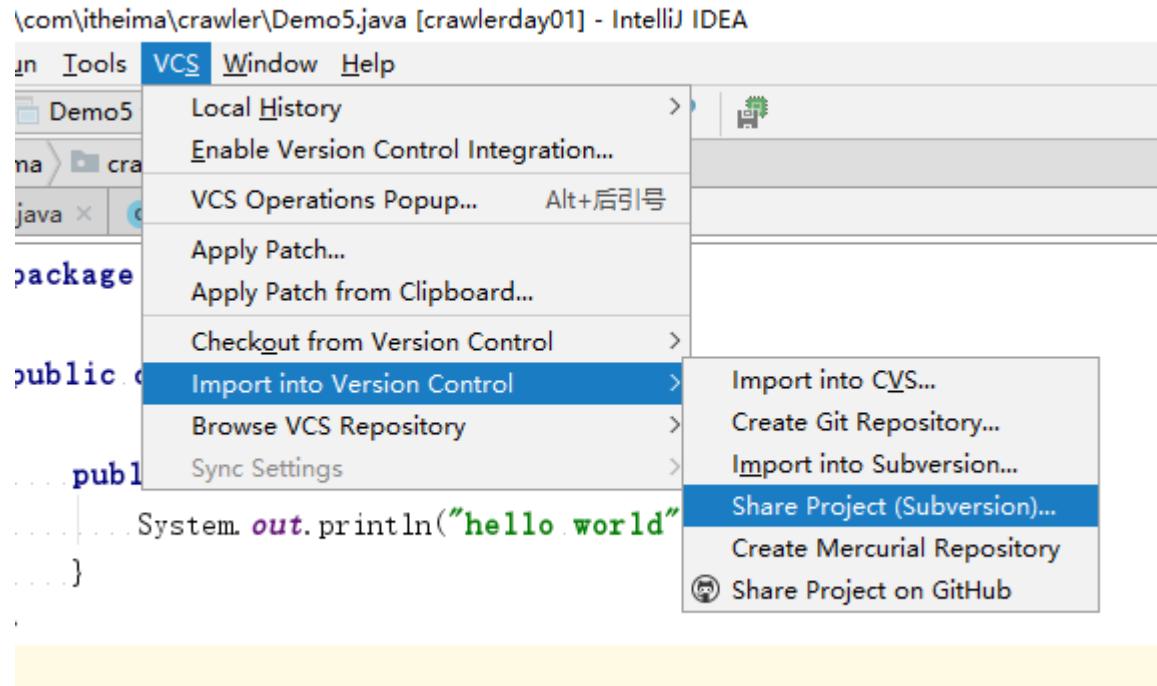
1.Local Changes 本地修改过的文件都会出现在这里

2.Repository svn仓库所有提交记录

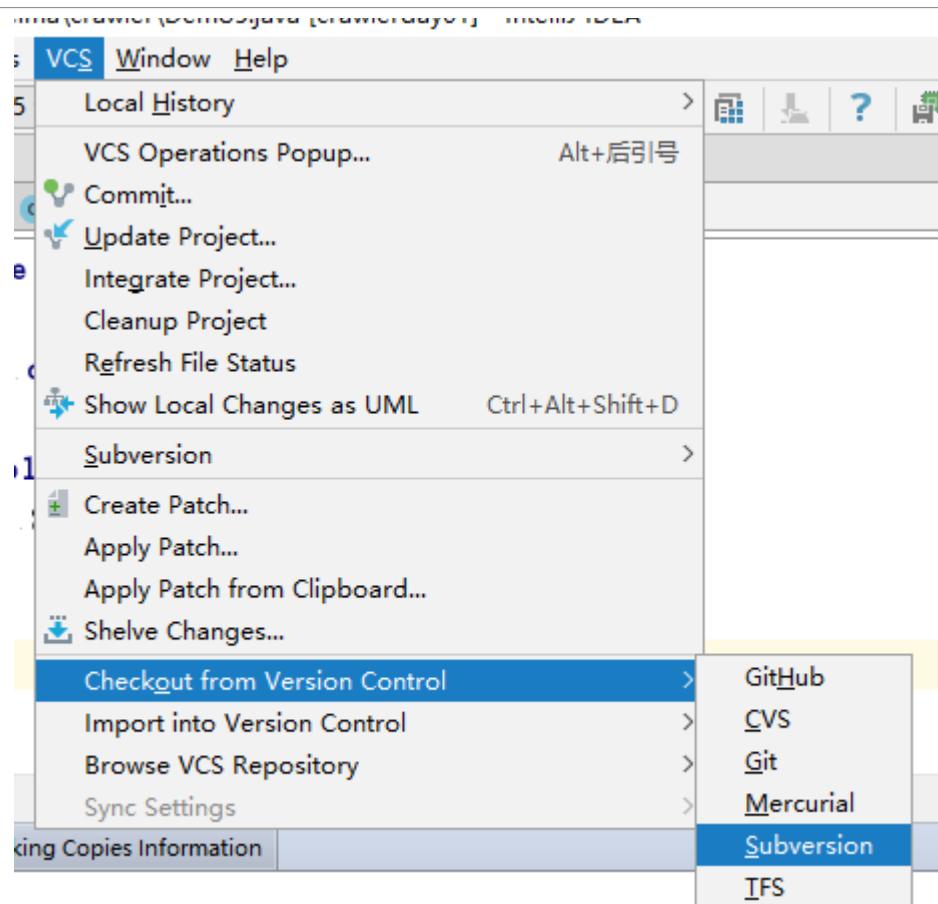
3.Incoming 本地还没有更新的别人提交的代码

2.3.2 IDEA 下SVN使用

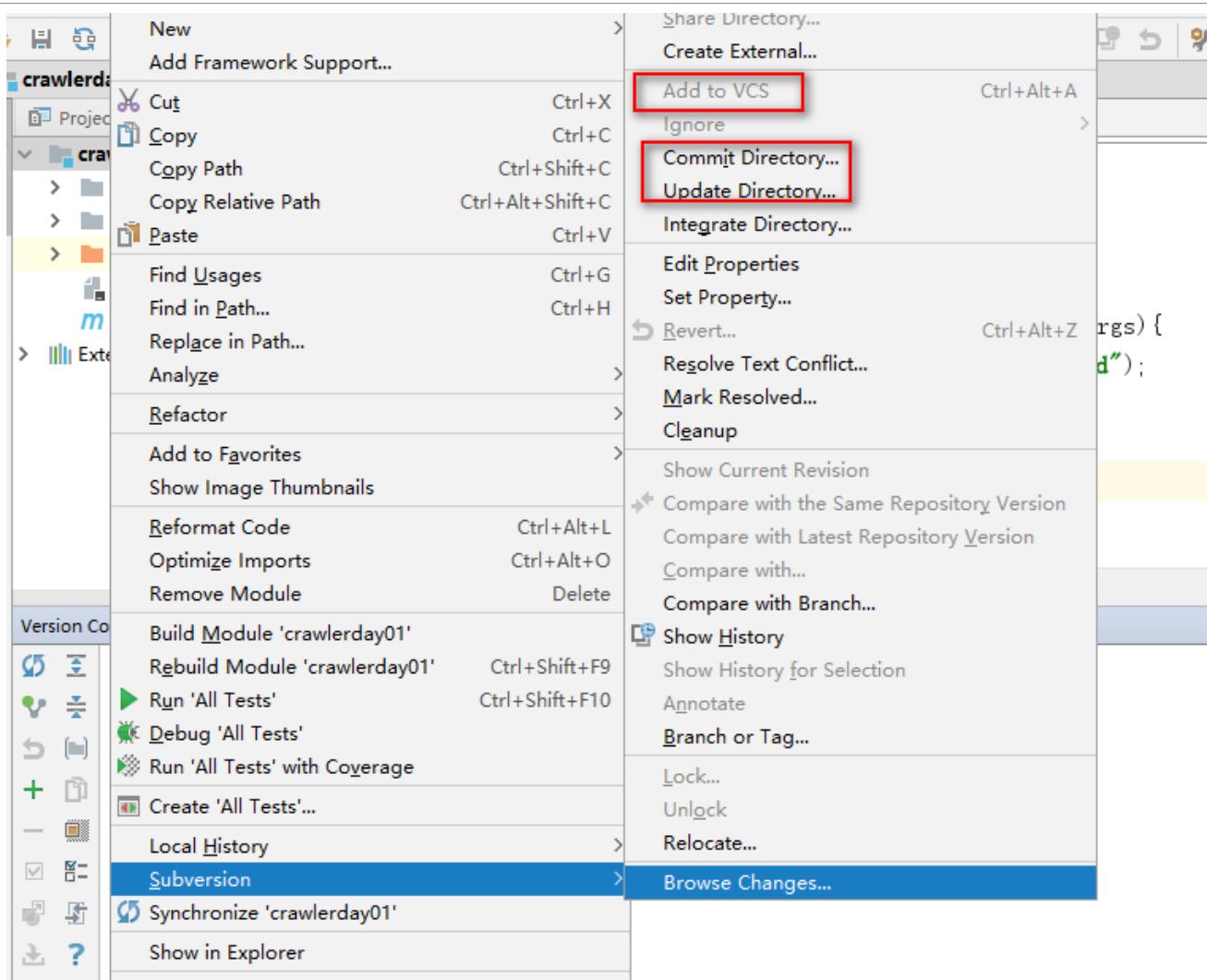
2.3.2.1 share project



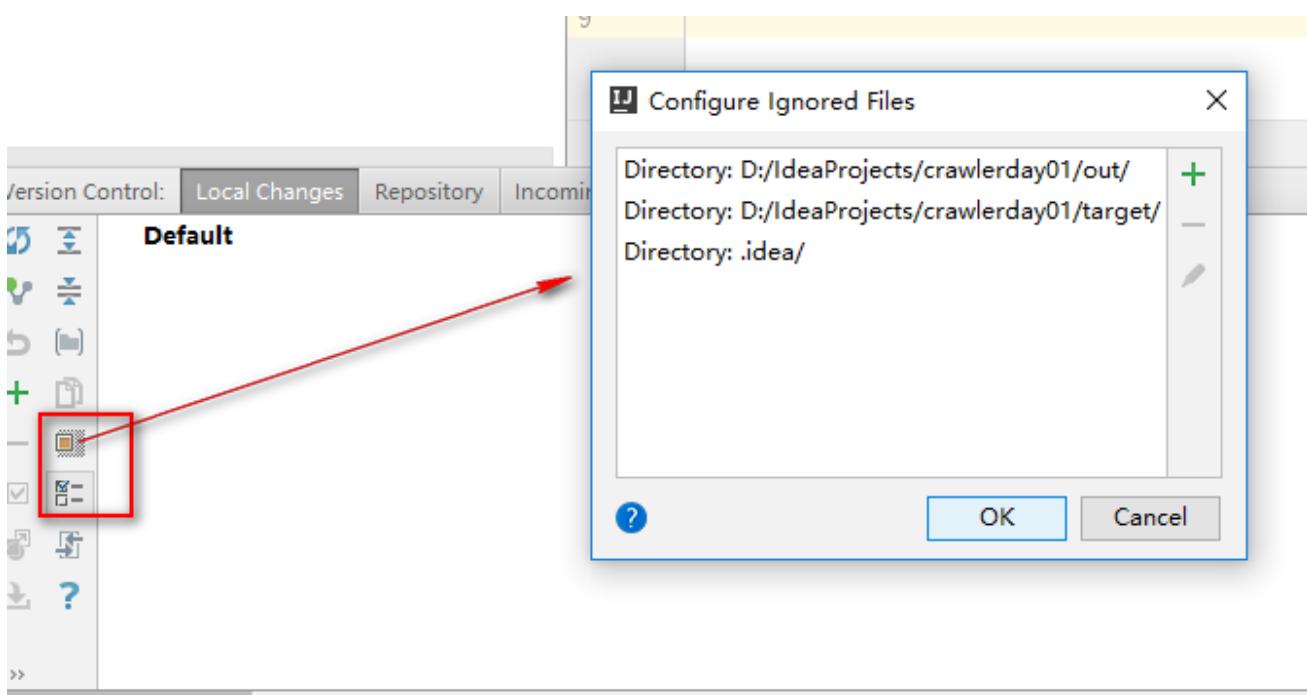
2.3.2.2 checkout

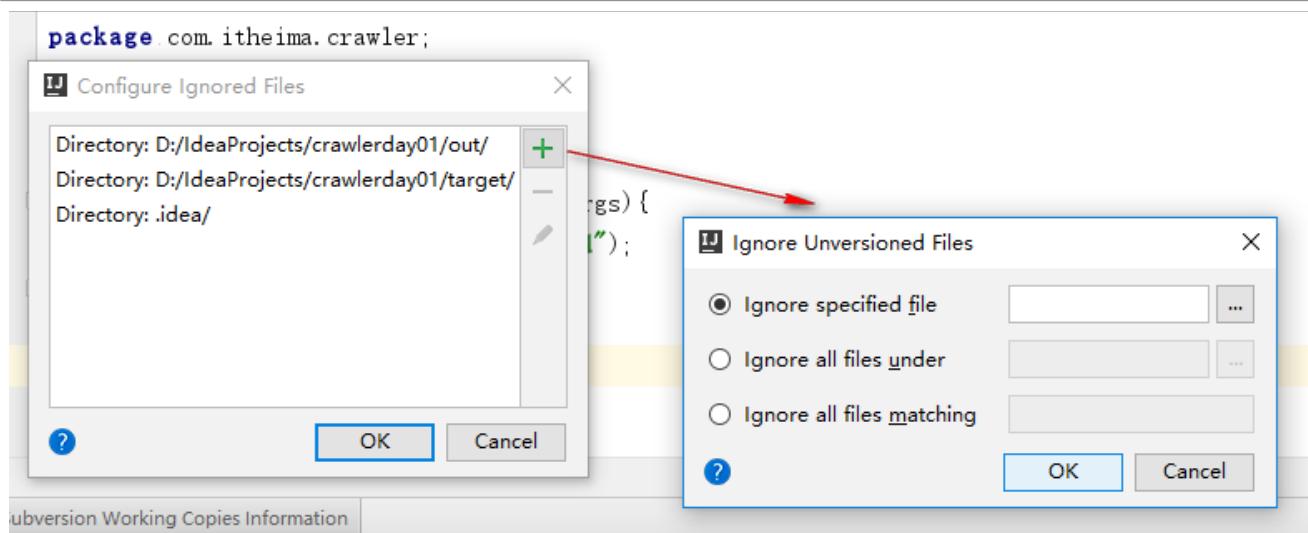


2.3.2.3 add commit update

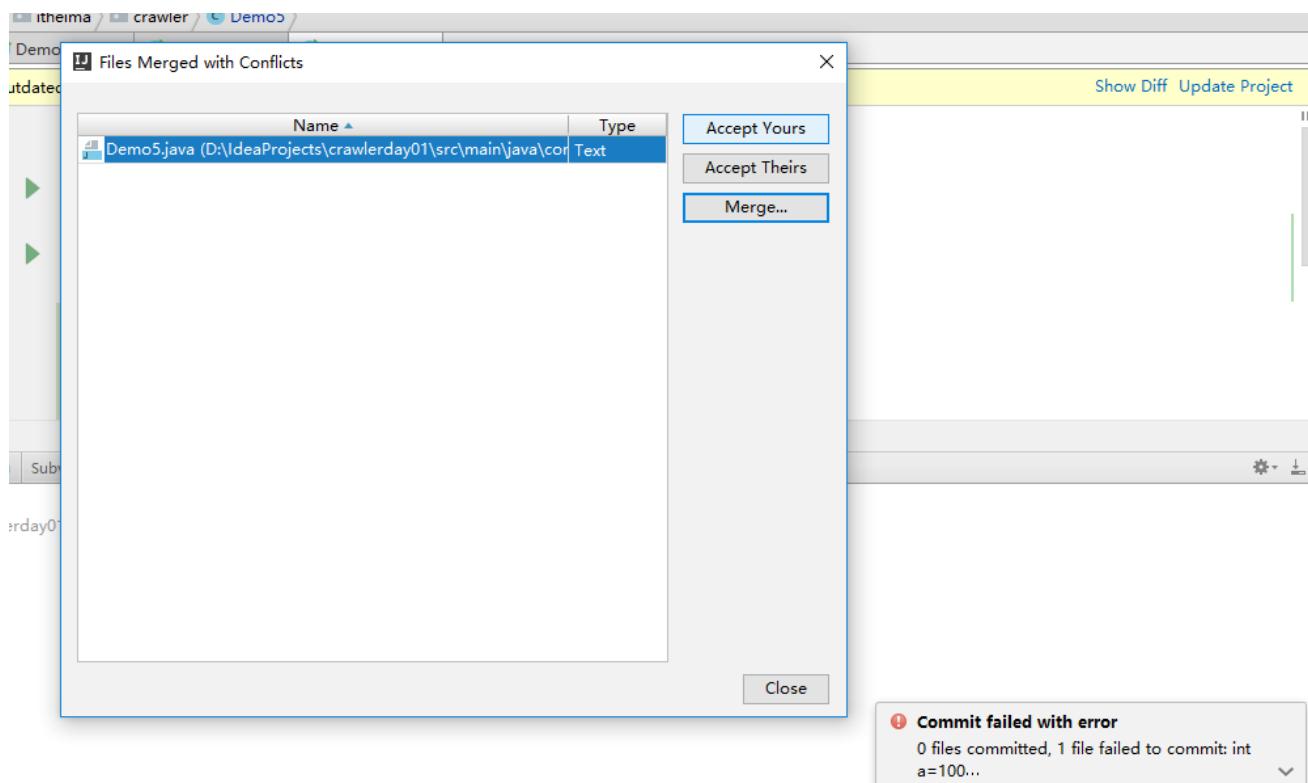


2.3.2.4 ignor 不起作用





2.3.2.5 解决冲突





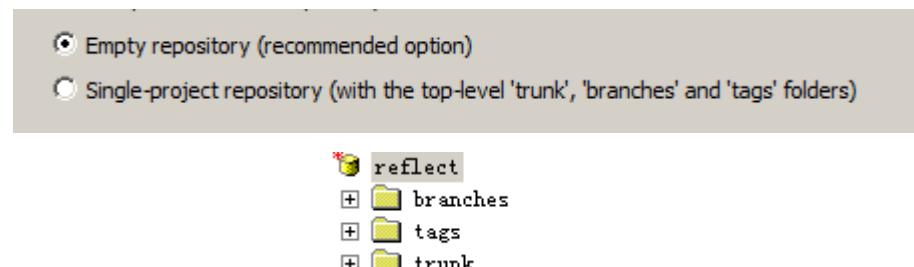
Merge Revisions for D:\deaProjects\crawlerday01\src\main\java\com\itheima\crawler\Demo5.java

The screenshot shows a merge conflict in a Java file named Demo5.java. The left pane shows 'Local Changes (Read-only)' with code that prints 'hello'. The right pane shows 'Changes from Server (revision 7) (Read-only)' with code that prints 'hello' and has a line 'int a=10;'. A conflict marker 'int a=10;' is highlighted in red in both panes. Below the panes are buttons for 'Accept Left', 'Accept Right', 'Apply', and 'Abort'.

3. SVN规范

3.1 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(即时通讯),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，开发完成后就可以打一个tags

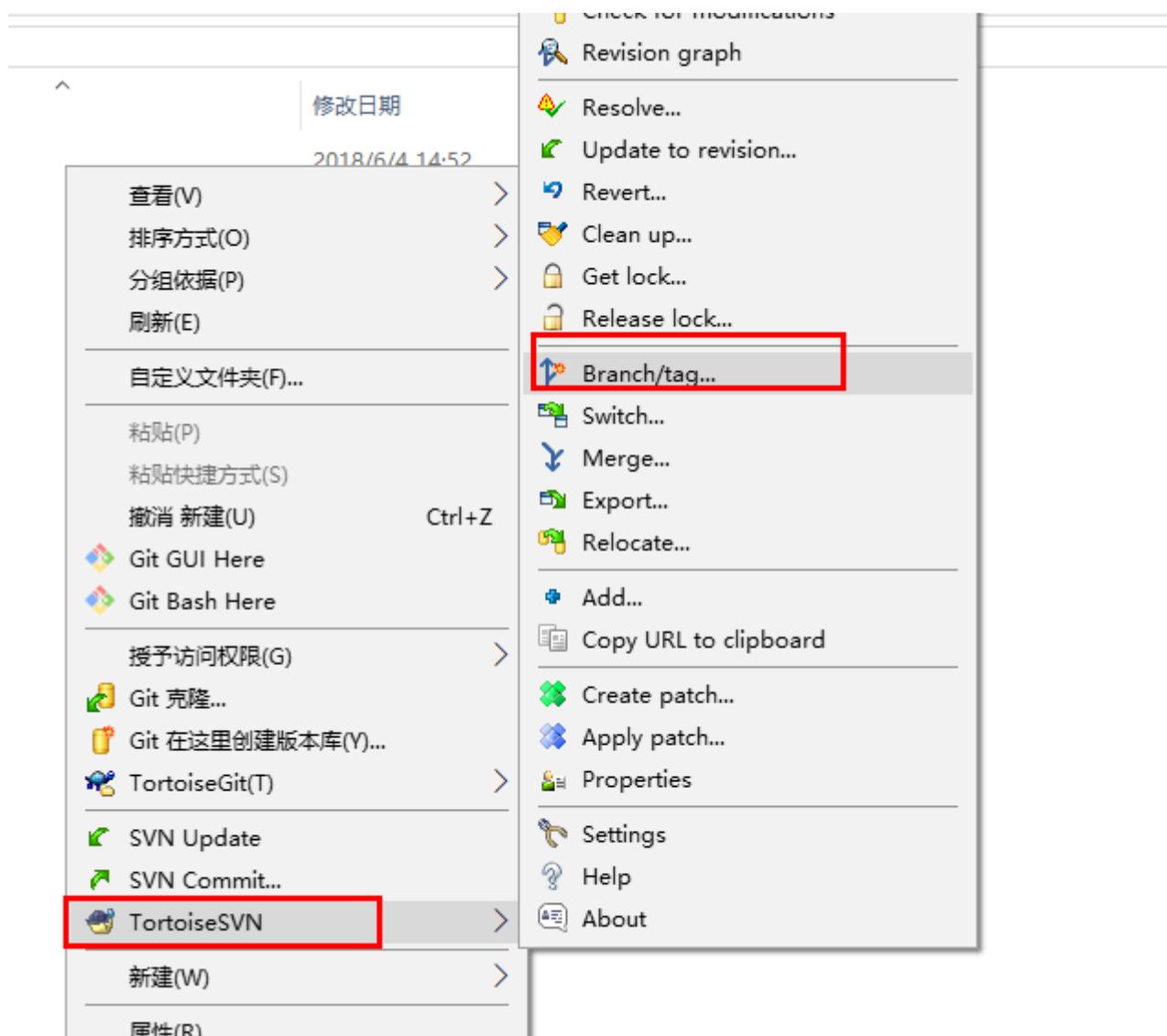
3.2 分支与标记

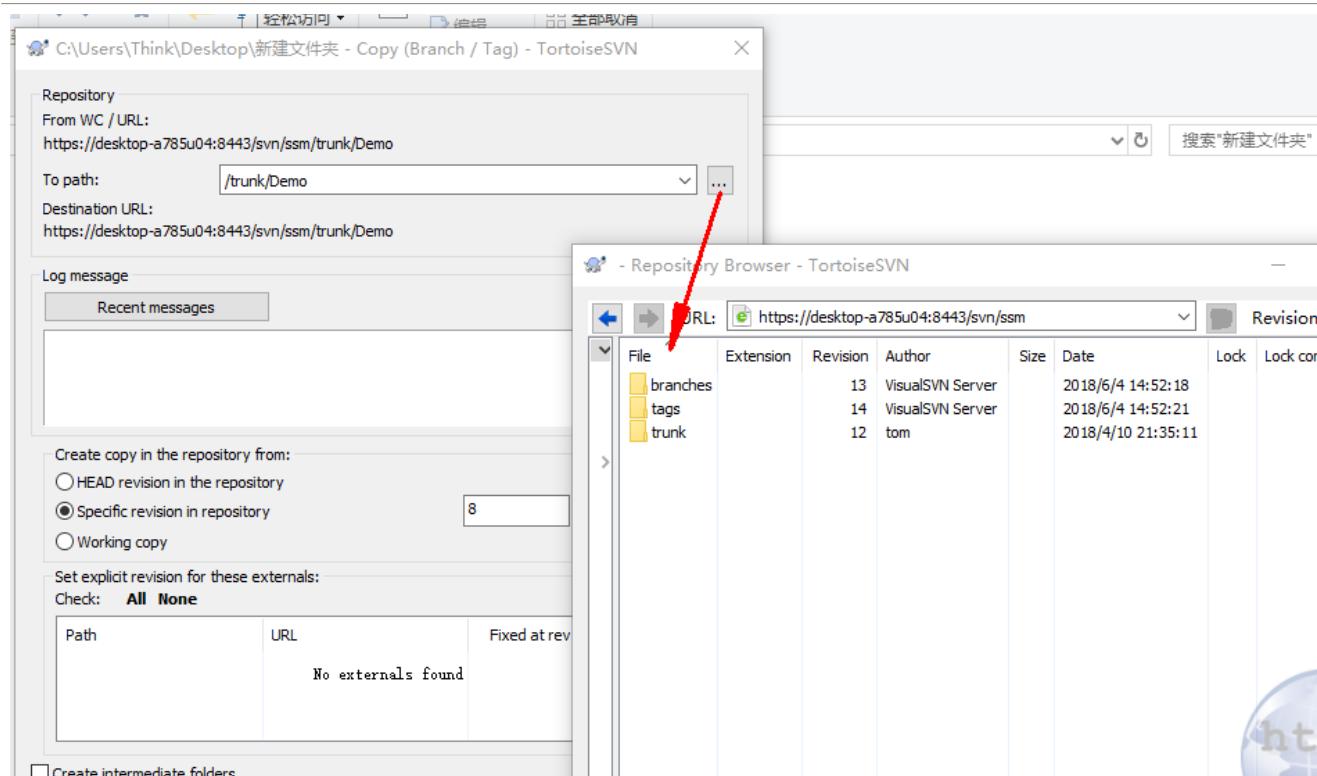
3.2.1. 分支或标记规范



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

3.2.2.打分支或标记



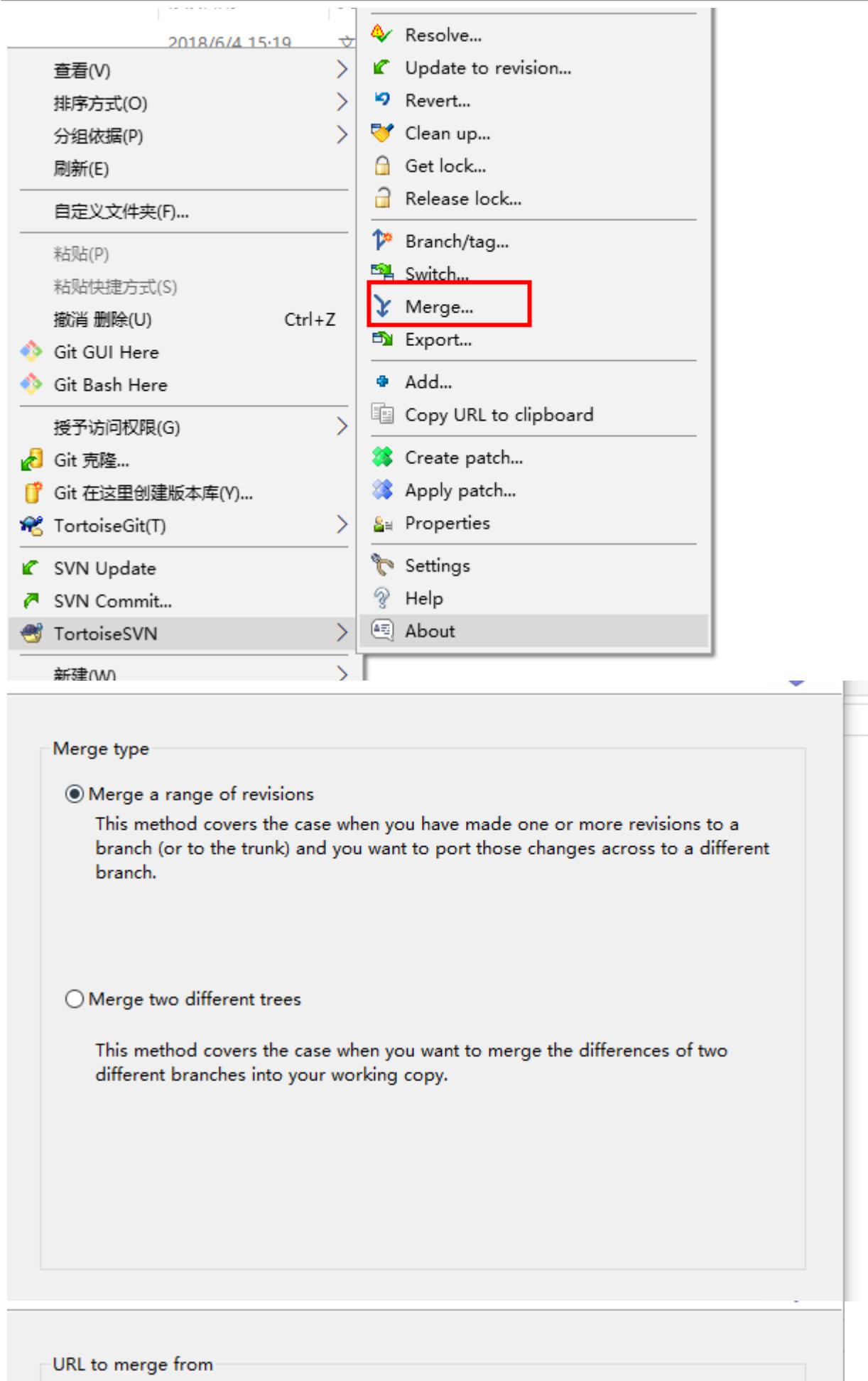


Tags一般是只读，它代表的是发布的版本，所以我们不要进行改变。

3.3 主干与分支或标记的合并

3.3.1 主干合并到分支

首先保证主干是最新的(先update)，在分支这边进行操作



The screenshot shows the TortoiseGit interface with a context menu open over a file. The menu path is: **右键 -> Merge...**. The 'Merge...' option is highlighted with a red box.

Merge type

Merge a range of revisions
This method covers the case when you have made one or more revisions to a branch (or to the trunk) and you want to port those changes across to a different branch.

Merge two different trees
This method covers the case when you want to merge the differences of two different branches into your working copy.

URL to merge from



https://desktop-a785u04:8443/svn/ssm/branches/Demo-2018-12-12_dog ▾ ...

Revision range to merge

all revisions

specific range Show log

Reverse merge

Use the log dialog to select the revisions you want to merge, or enter the revisions to merge, separated by commas. A revision range can be specified by a dash.

Example: 4-7,9,11,15-HEAD@pegrevision

To merge all revisions (reintegrate/automatic merge), leave the box empty.

Working Copy

C:/Users/Think/Desktop/新建文件夹/trunk/Demo Show log

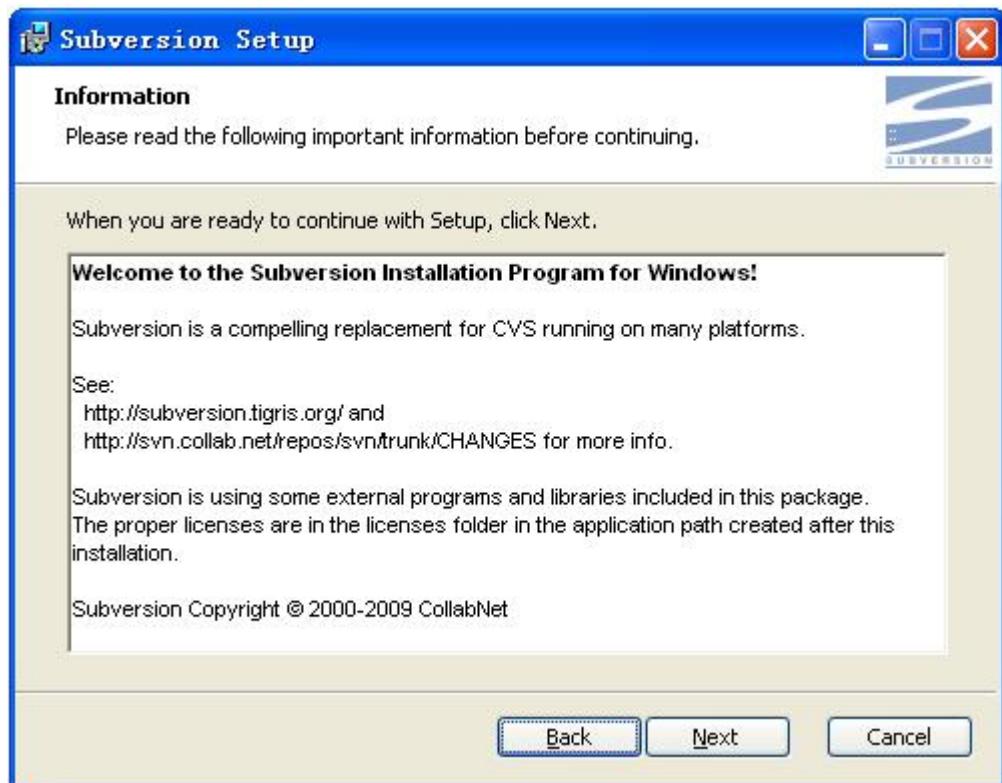
< Back Next > Cancel Help

SVN 安装界面

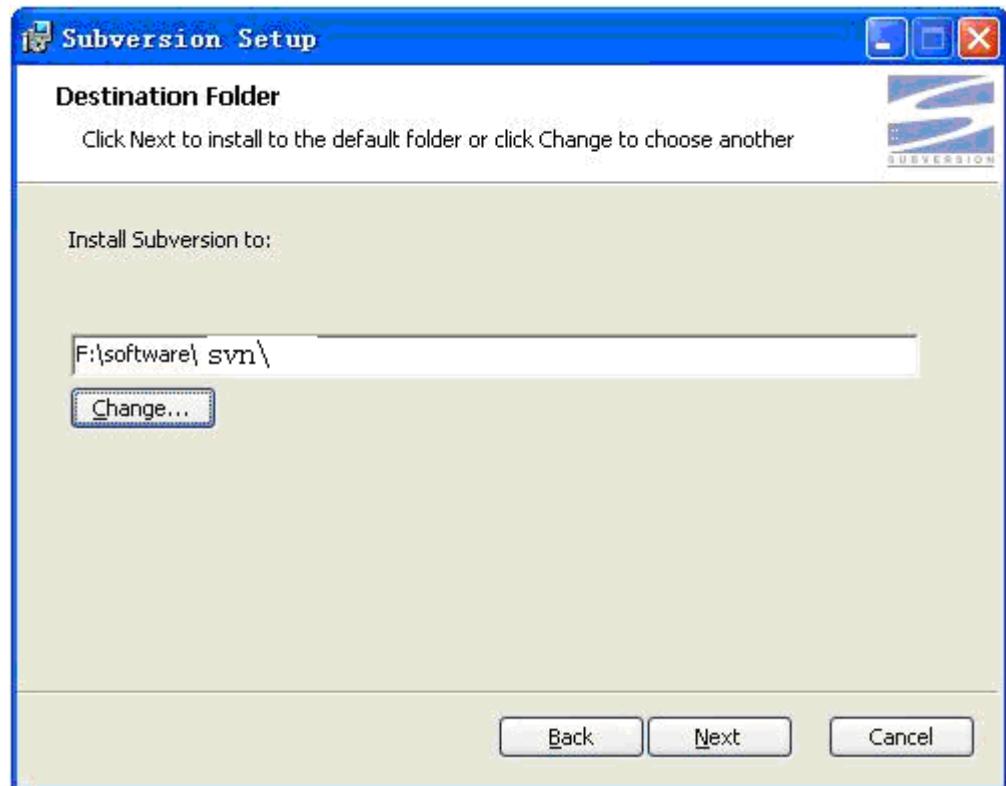
第一步： 点击 Setup-Subversion-1.6.5.msi



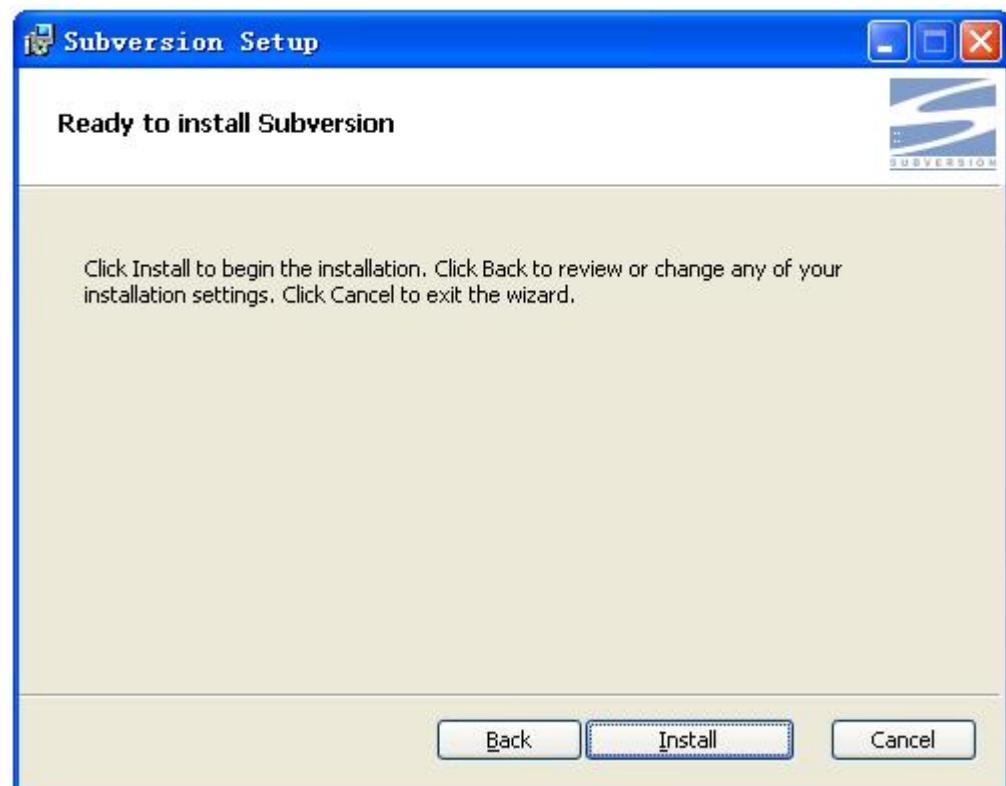
第二步：



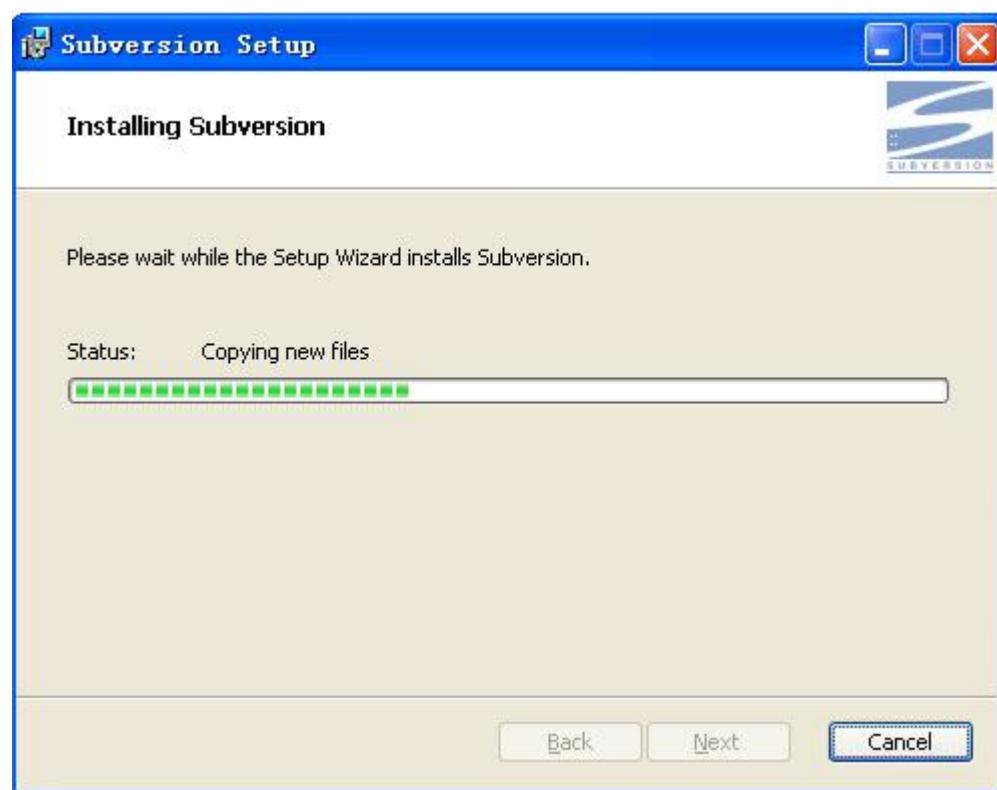
第三步： 安装到指定的 svn 目录下



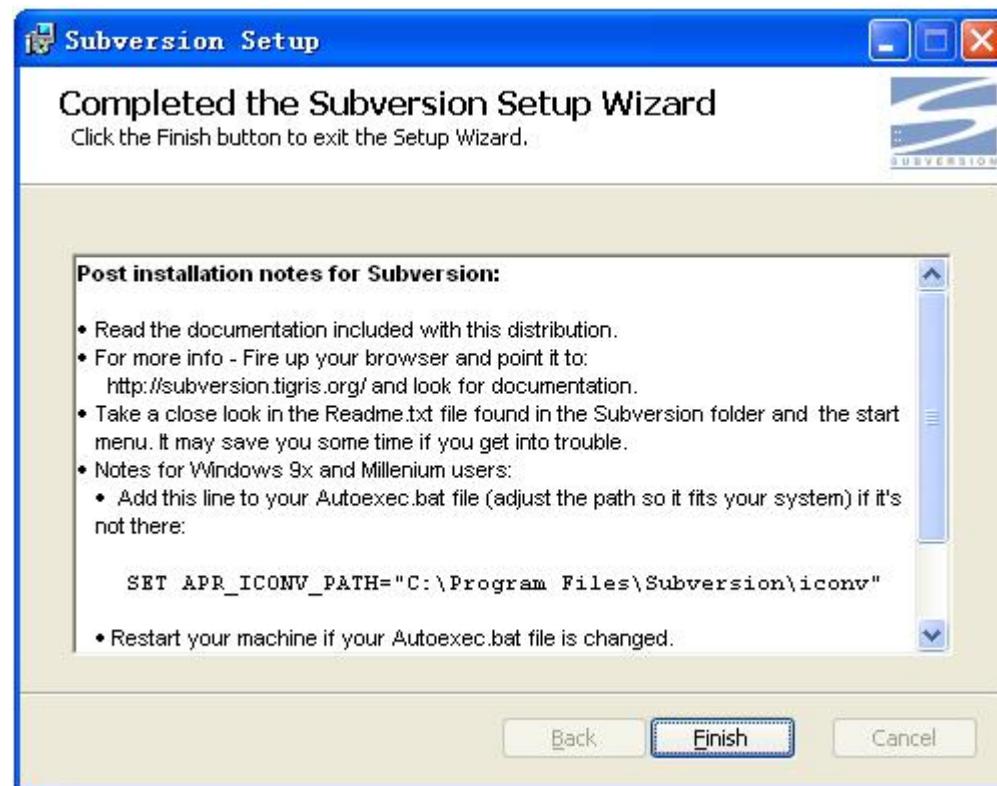
第四步：



第五步



第六步：安装完成



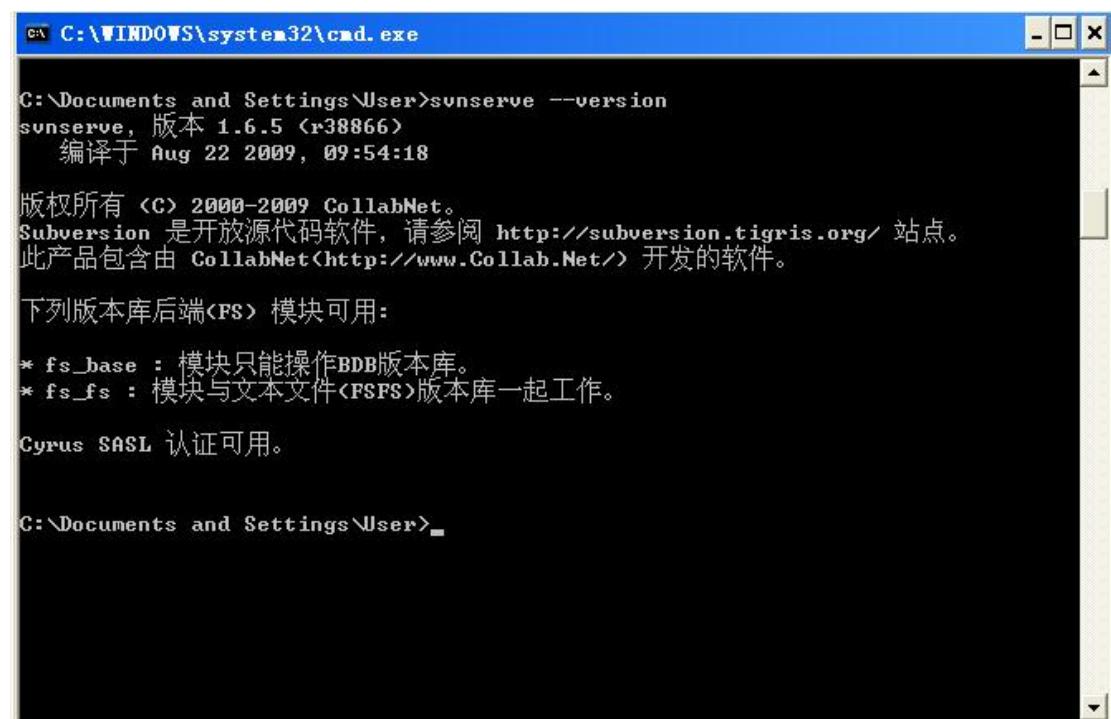
第七步： 最终安装完成后，如何验证在我们的服务器上就成功安装了 Subversion 呢？Subversion 在安装时，同 JAVA 的开发环境(JDK)的安装一样，同样会自动的把安装目录下的 bin 目录设置到系统的系统变量中去。如果你是使用的绿色版，那么设置系统变量这个步骤就得你自己手动的来设置了，如：



第八步：，如果我们在系统的命令行执行命令：

【svnserve --version】

如果能正常显示如下图所类示的信息，则表明我们的 Subversion 的安装是成功的。



```
C:\Documents and Settings\User>svnserve --version
svnserve, 版本 1.6.5 (r38866)
编译于 Aug 22 2009, 09:54:18

版权所有 (C) 2000-2009 CollabNet。
Subversion 是开放源代码软件，请参阅 http://subversion.tigris.org/ 站点。
此产品包含由 CollabNet (http://www.Collab.Net/) 开发的软件。

下列版本库后端(FS) 模块可用:

* fs_base : 模块只能操作BDB版本库。
* fs_fs : 模块与文本文件(FSFS)版本库一起工作。

Cyrus SASL 认证可用。

C:\Documents and Settings\User>
```