**SVN概述**

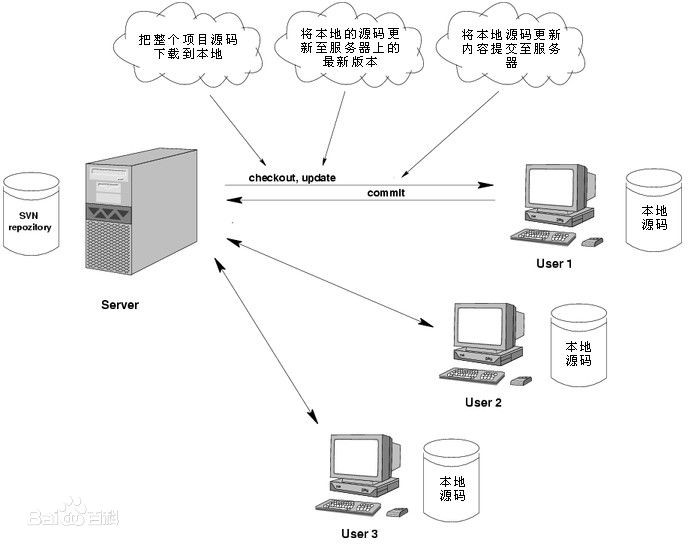
# SVN介绍

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，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。

## SVN基本概念



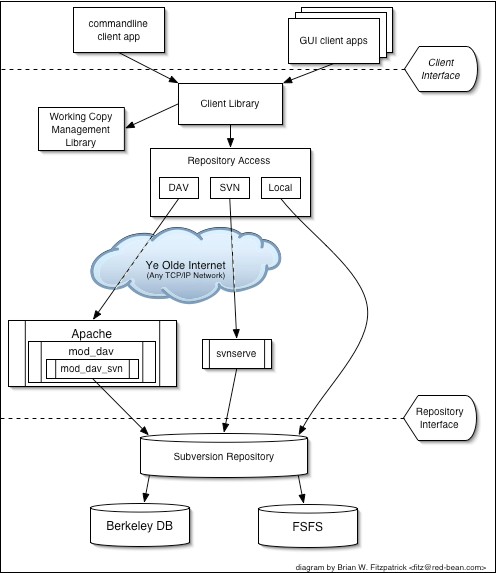
问题: 怎样让系统允许用户共享信息，而不会让他们因意外而互相干扰？ **复制-修改-合并方案(Subversion默认的模式)**

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。

# SVN使用

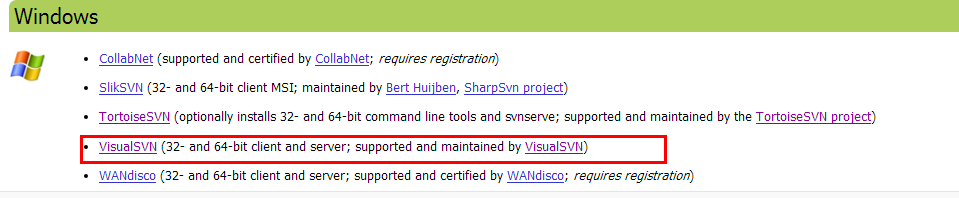
主要介绍常用的svn工具，如window下svn服务器VisualSVN，客户端工具TortoiseSVN以及eclipse svn插件的使用

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）。

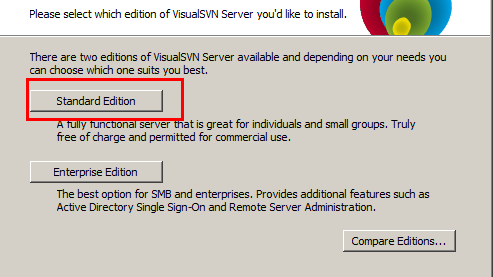
SubVersion 官网：<http://subversion.apache.org/>

## 服务器端VisualSVN安装与配置

### VisualSVN下载

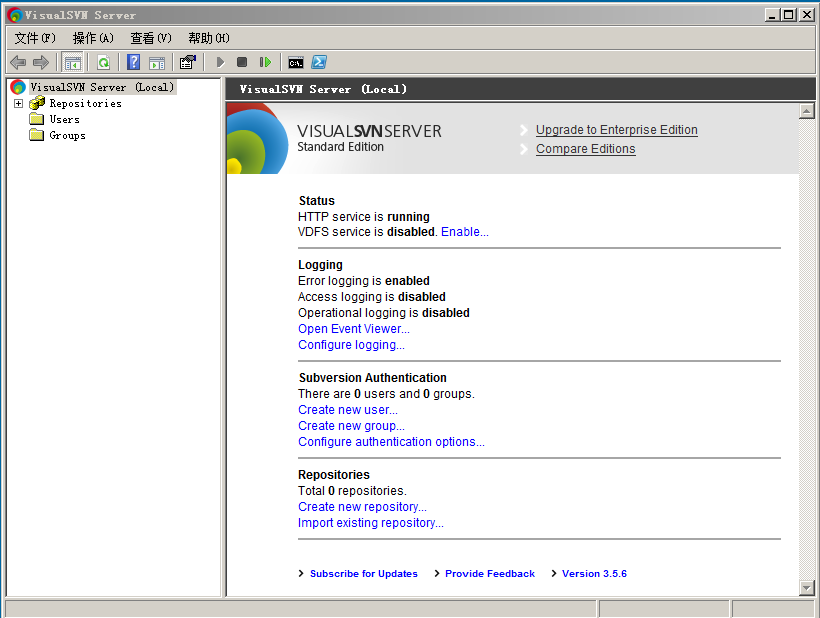


* + 1. **VisualSVN安装与创建仓库**



在开始运行中录入 services.msc



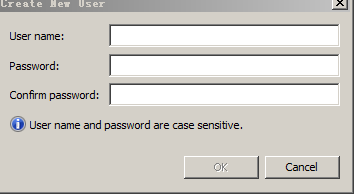
这个服务要开启 在开始运行中查找visualSVN Server Manager就可以打开图形化管理界面

### VisualSVN创建工程目录

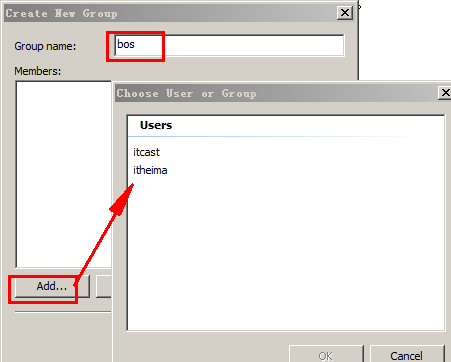


* + 1. **VisualSVN 权限控制**

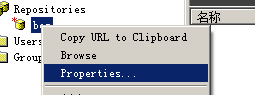
创建用户



创建组



分配权限



【】

## 客户端TortoiseSVN安装与使用

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

### Tortoise SVN下载



* + 1. **Tortoise SVN安装**

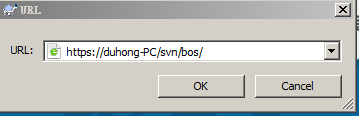
安装后需要重启电脑

【】

### Tortoise SVN 常用操作

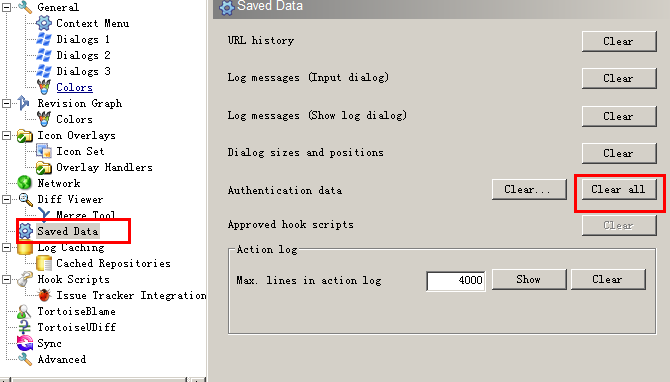
#### 浏览仓库

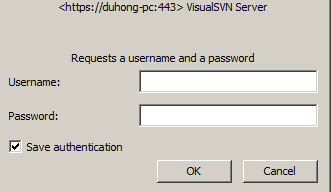
【】

与password 我们可以清空svn客户端的缓存

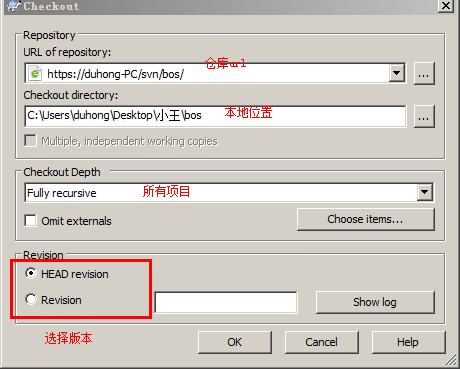
正常情况下，在浏览仓库时会要求录入username

【】





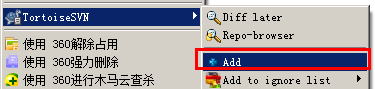
#### checkout

检出svn服务器上的项目

一个.svn隐藏目录 

注意：检出后会有

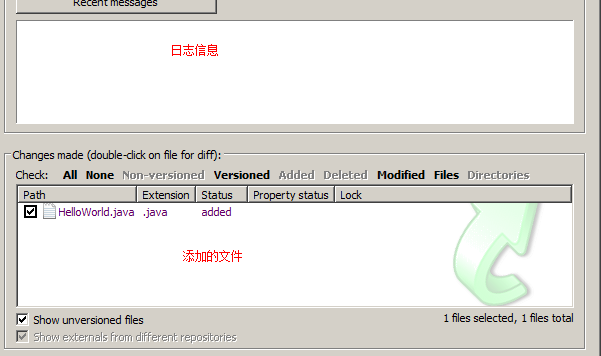
#### add

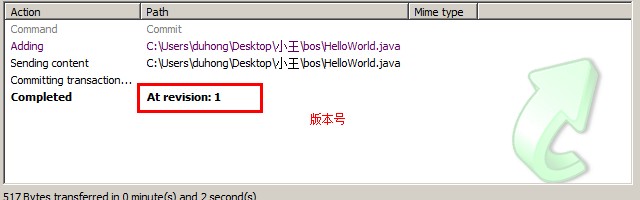
这样图标的文件代表它没有由svn管理

这个代表的是它要添加到svn服务器

#### commit





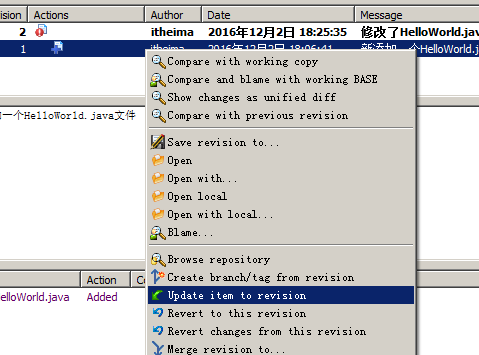


* + - 1. **update**

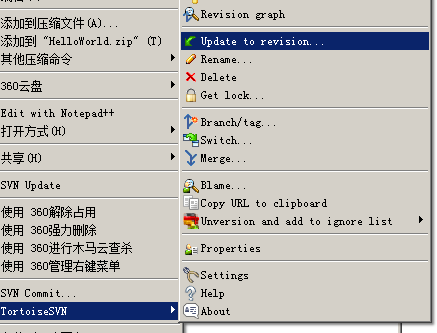
Update，它是更新操作,可以将svn服务器上的内容更新到本地

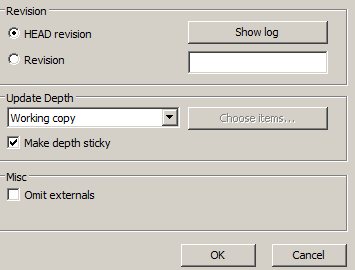
* + - 1. **更新到历史版本**查看当前日志

【】



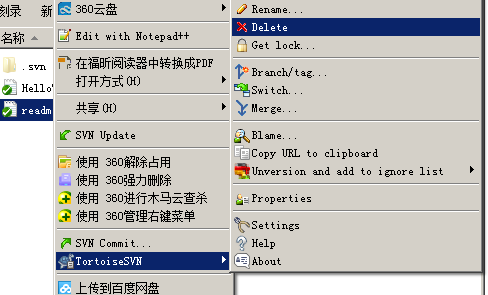
也可以通过下面的方式来更新历史版本



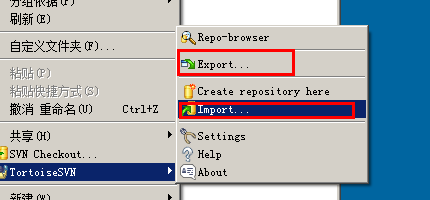


#### delete

标记删除操作，本地文件会删除，我们需要进行提交，才能真正从服务器删除



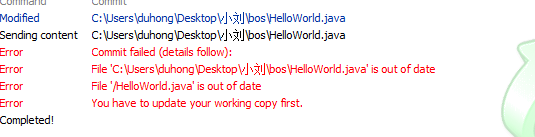
#### 导入与导出



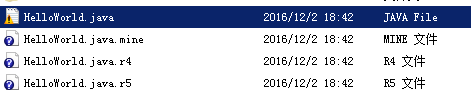
Import将本地资源导入到svn服务器

Export导出项目，和checkout的区别,它不存在.svn隐藏文件

#### 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。\

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

