

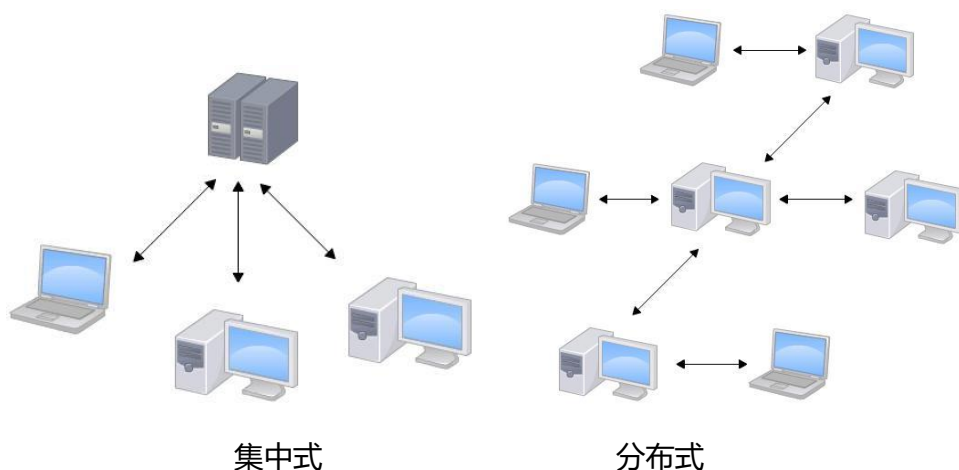
# 一、版本控制和 SVN 入门

## 1. 什么是版本控制

- a) 只要进行团队软件开发，就会涉及源代码的合并共享以及带来的相关问题
  - i. 代码如何合并（手工合并几乎不可能）
  - ii. 历史版本的保存（各个版本间有大量重复，多大变化算一个版本）
  - iii. 跟踪哪些代码是谁修改（每个代码都写上注释说明作者？故意写成别人呢）
  - iv. 以上问题不解决，就会影响团队开发效率
- b) 如何有效解决以上问题呢？这就需要版本控制
- c) 目前计算机开发领域流行的是采用专门的版本管理软件来进行管理，大大提高了效率

## 2. 常用版本管理软件

- a) VSS：Microsoft 提供，Window 平台，小型开发团队所适合的工具。
- b) CVS：开源工具，跨平台。绝大多数 CVS 服务已改用 SVN。CVS 已停止维护
- c) SVN（Subversion）：开源，跨平台，在 CVS 的基础上开发。使用企业众多，稳定安全，操作简单。集中式版本控制系统。
- d) GIT：开源的分布式版本控制系统，用以有效高速的处理从很小到非常大的项目版本管理。（GITHUB 是一个用 GIT 做版本控制的项目托管平台，B/S 架构。放到 GITHUB 上的代码需要开源，并且是互联网开发）



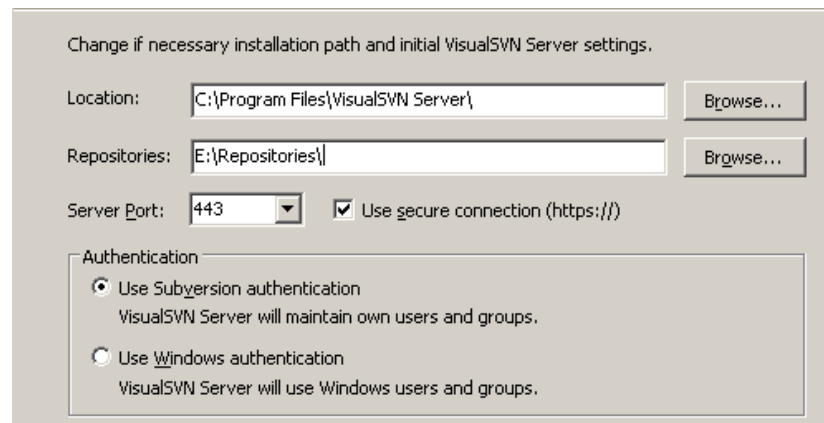
### 3. SVN

- a) 传统的 SVN 是通过命令进行操作的，现在一般通过 GUI 形式使用 SVN
- b) 服务器端 Visual SVN (GUI)
- c) 客户端:TortoiseSVN 第三方客户端(GUI)
- d) 客户端：subclipse Eclipse 的插件 第三方客户端(GUI)
- e) TortoiseSVN 功能更强、使用范围更广，但是 eclipse 下使用 subclipse 更方便

## 二、服务器端 Visual SVN

### 1. 安装软件

- a) 指定软件安装位置和版本库位置
- b) 默认端口 443
- c) 使用 SVN authentication



### 2. 创建版本库

- a) 创建默认结构 use default struction
- b) 默认结构的含义
  - i. trunk: 主干 开发时代码存储的位置, 永远是最新的代码
  - ii. brank: 分支 在不影响 Trunk 其它用户情况下进行一些关于新功能的探索性或实验性的开发, 待新功能完善后它也可以合并到 Trunk 中
  - iii. tags: 标签 历史版本 阶段性里程碑版本 比如 1.0 2.0 3.0

### 3. 创建用户

### 4. 创建组

### 5. 为组指定用户

### 6. 为组指定对版本库的操作权限

- a) 默认 everyone 具有 read/write 权限, 无法删除, 要修改为 no access

### 7. 通过浏览器访问服务器端

## 三、客户端 subclipse

### 1. 简介

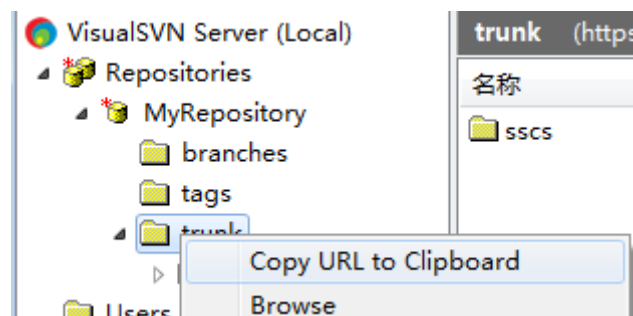
- a) subclipse=subversion+eclipse
- b) 一个为 Eclipse 添加 Subversion 支持的项目。支持几乎所有版本的 Eclipse

### 2. 安装

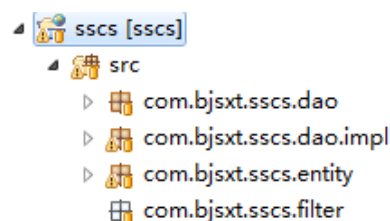
- a) 解压后将其中的 features 和 plugin 文件夹放入 MyEclipse 的 dropins 目录下
- b) 重新启动 MyEclipse, 即可自动发现并安装
- c) 安装后在 window-----preferences-----team 中可以看到 SVN

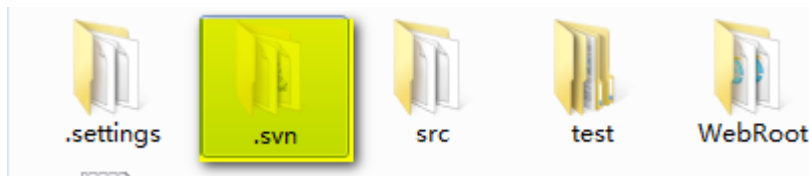
### 3. 提交项目到服务器端

- a) 选中项目右键-----team----share project----选择 SVN
- b) 提交版本位置、用户名、密码都会由相关服务器管理人员提供。



- c) 可以选择记住密码，避免多次的重复输入。
- d) 提交后会跳到 team synchronizing view, 还需要真正同步代码到服务器, **同步之前可以选择哪些代码不需要使用 svn 进行管理**
- e) 提交成功后可以观察版本库容量的变化
- f) 提交成功后客户端项目会显示专门的标记, 并且增加了相应的.svn 项目, 来存储操作记录





#### 4. 断开和服务端连接

- a) team—断开连接-----从项目中删除 SVN 元信息

#### 5. 客户端从 SVN 检出项目

- a) new--- Project...-----SVN----从 SVN 检出项目

#### 6. 更新和提交操作

- a) 更新将服务器最新代码更新到本地；提交是将本地最新代码提交到服务器
- b) **提交之前要先更新**，因为可能其他程序员期间以及提交了最新代码到服务器
- c) **提交一定要给出注释，对提交内容进行说明，作为以后辨别版本的主要依据。**

#### 7. 解决冲突

- a) 模拟两个用户对一个类进行修改，分别修改相同的方法和增加一个新方法
- b) 更新后出现冲突，和冲突方沟通后，对冲突文件给出最终解决方案
- c) 还需要 team----标记为解决，相应冲突文件为自动删除，当前文件由冲突状态转换为已修改状态，待提交

#### 8. 还原没有提交的代码

- a) team----还原

#### 9. 还原已经提交的代码

- a) 如果最新的代码出现问题，可以直接修改，也可以回滚到之前的某个历史版本，直接使用或者进行修改
- b) 查看资源历史记录
- c) 根据注释确定要还原的版本，无法确定可以查看或者比较代码
- d) 确定版本后，右键选择“从修订班 x 回复更改”
- e) 可能出现冲突，解决后提交代码到服务器