

版本控制器 SVN

一、版本控制

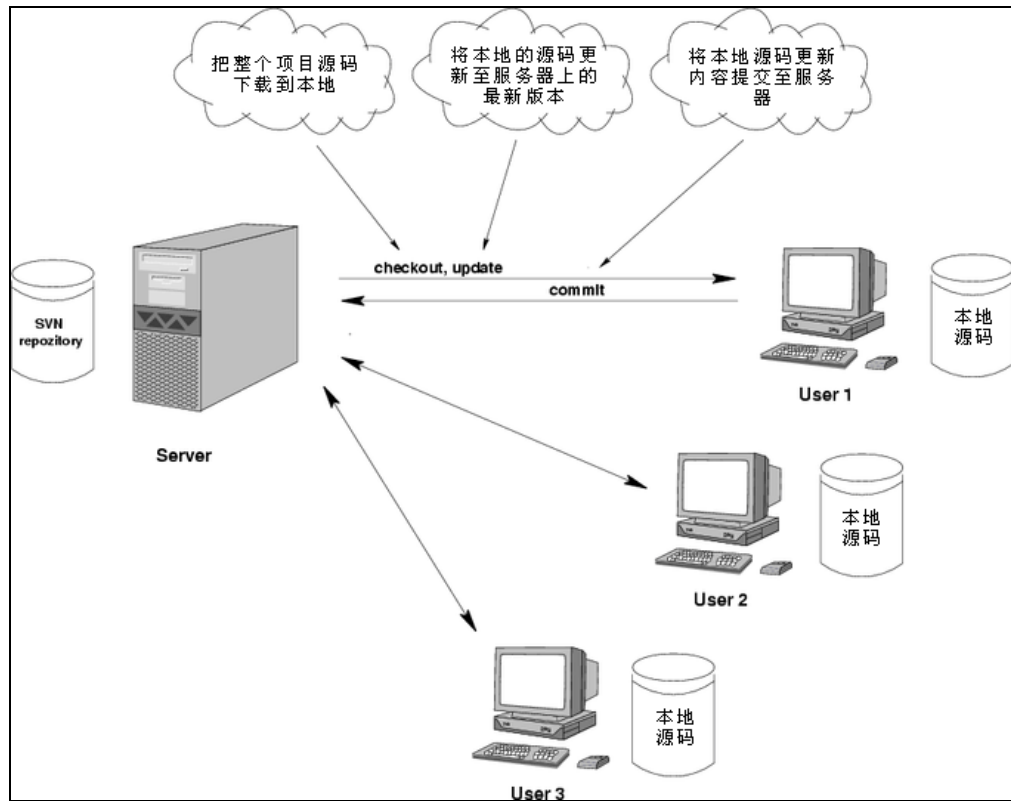
1 为什么需要版本控制？

- 1.1 小明负责的模块就要完成了，就在即将 **Release** 之前的一瞬间，电脑突然蓝屏，硬盘光荣牺牲！几个月来的努力付之东流——需求之一：备份！
 - 1.2 这个项目中需要一个很复杂的功能，老王摸索了一个星期终于有眉目了，可是这被改得面目全非的代码已经回不到从前了。什么地方能买到哆啦 A 梦的时光机啊？需求之二：代码还原！
 - 1.3 小刚在 `Analysis.java` 文件中的第 30 行声明了一个方法，叫 `count()`，保存到了文件服务器上；小强在 `Analysis.java` 文件中的第 50 行声明了一个方法，叫 `sum()`，也保存到了文件服务器上，于是，`count()` 方法就只存在于小刚的记忆中了——需求之三：协同修改！
 - 1.4 老许是一位项目经理，我会告诉你他亲自为工程创建版本号吗？我会告诉你他把每一个版本都保存一份吗？我会告诉你这些工程里其实有很多文件都是重复的吗？我会告诉你老许为这事删了很多电影吗？——需求之四：版本管理和版本发布！
 - 1.5 老王是另一位项目经理，每次因为项目进度挨骂之后，他都不知道该扣哪个程序员的工资！就拿这次来说吧，有个该死的 **Bug** 调试了 30 多个小时才知道是因为相关属性没有应用初始化时赋值！可是二胖、王东、刘流和正经牛都不承认是自己干的！——需求之五：追溯问题代码的编写人和编写时间！
 - 1.6 小温这两天幸福的如同掉进了蜜罐里，因为他成功的得到了前台 MM 丽丽的芳心，可他郁闷的是这几天总是收到 QA 小组的邮件，要求他修正程序中存在的 **Bug**，可他自己本地电脑上是没有这些 **Bug** 的，“难道我的代码被哪个孙子给改了？”。是的，小温没来的时候，丽丽是 QA 小组小郑的女朋友啊！——需求之六：权限控制！
- 使用版本控制能够解决上面的问题吗？是的！

2 什么是版本控制？

- 2.1 版本控制：Revision control，最初来源于工程设计领域，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。
- 2.2 Subversion 就是一款实现版本控制的工具软件，通常也称为版本控制器，简称 SVN。

2.3 SVN 的工作原理：采取客户端/服务器模式，多个协同开发的程序员把各自的源代码提交到版本控制器的服务器端，由 SVN 服务器对各个版本进行编号，并负责管理冲突，SVN 能够将冲突的代码以比较友好的格式呈现出来，便于开发人员比较，进行取舍。



2.4 SVN 基本操作

2.4.1 检出 (checkout)：将一个服务器端创建好的项目整个下载到本地，这是到项目组后参与开发的第一步，只需执行一次。

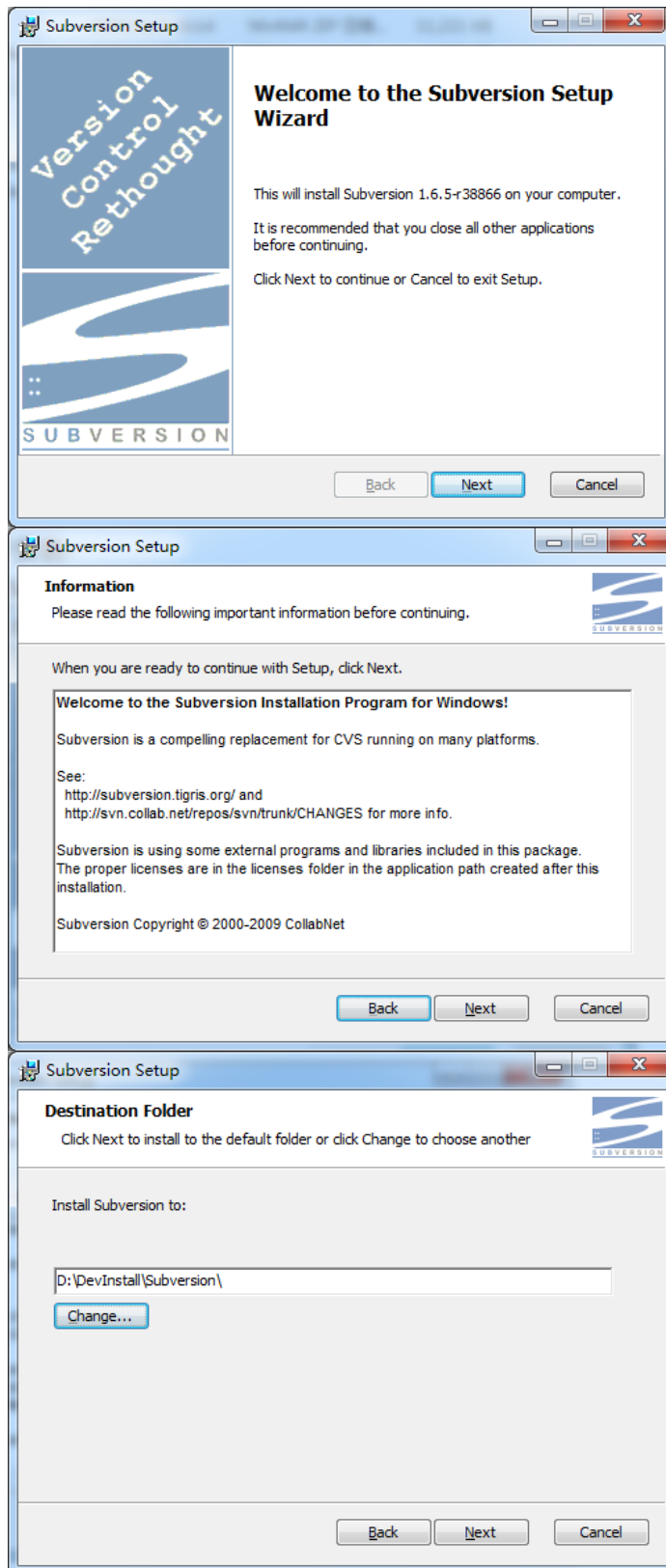
2.4.2 更新 (update)：将本地文件更新为服务器端的最新版本，通常为每天上班时或修改公共文件之前执行一次。

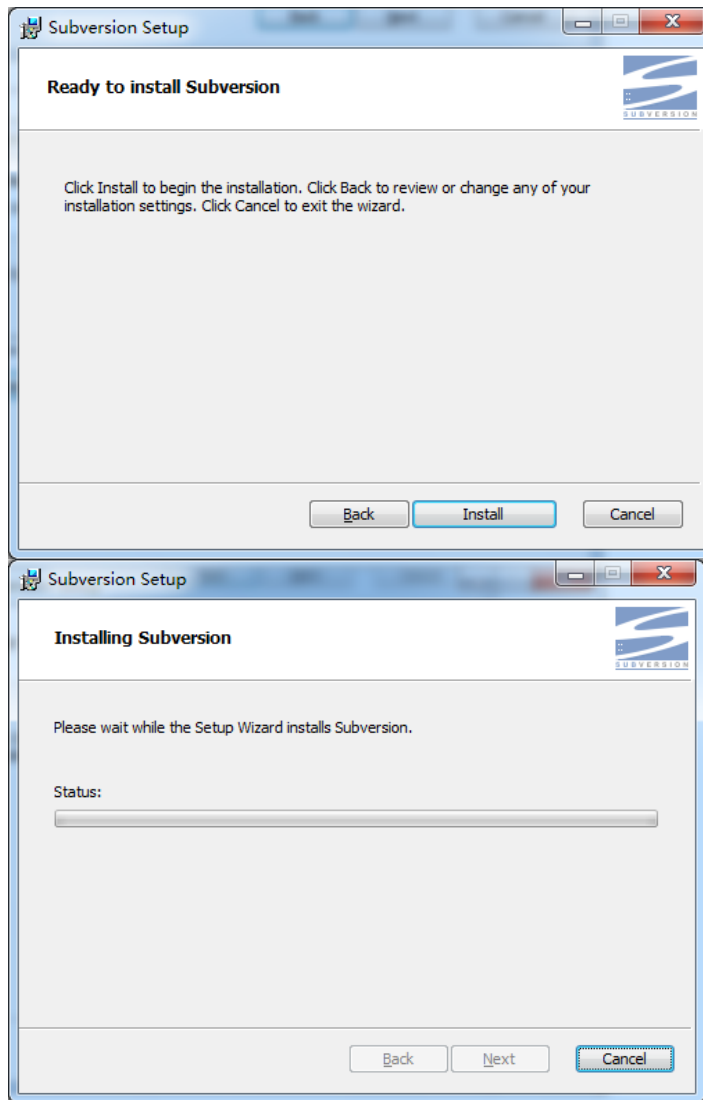
2.4.3 提交 (commit)：将本地修改提交到服务器端。通常每天下班前或每实现一个功能、完成一个模块时执行一次。

二、SVN 服务器端安装与配置

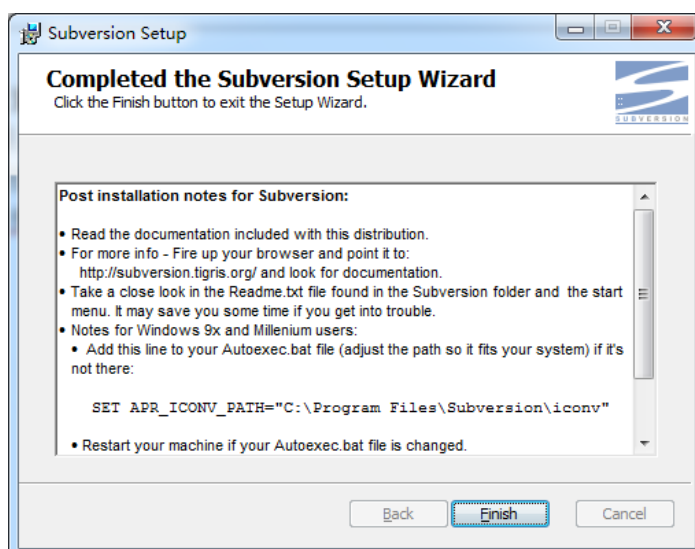
1 服务端程序安装

1.1 安装 Setup-Subversion-1.6.5.msi





Win7 及以上系统此处需要确认权限



1.2 安装程序会自动配置 Path 环境变量
D:\DEVINSTALL\SUBVERSION\BIN;

1.3 验证是否安装成功

在命令行输入：svnserve --version

看到如下信息就表示服务器端程序安装成功

```
C:\Users\Phenix>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。
```

2 SVN 仓库配置

2.1 为什么要配置 SVN 仓库？答：为了给版本信息找一个家，保存客户端上传的各个版本的文件。

2.2 步骤：

2.2.1 创建一个目录作为版本库根目录：要求全英文无空格。

例如：D:\DevRepository\svnRep

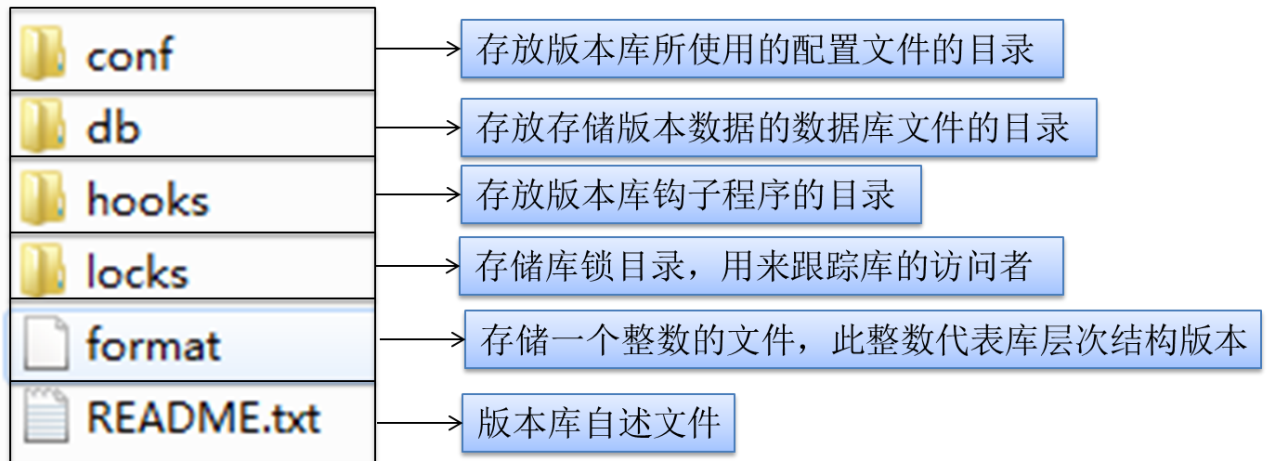
2.2.2 创建版本库

使用命令：

svnadmin create D:\DevRepository\svnRep\AtguiguERP

注意：具体仓库应与具体项目有关，例如仓库 AtguiguERP 将对应一个名为 AtguiguERP 的项目，OA 对应一个名为 OA 的项目，ERP 将对应一个名为 ERP 的项目。项目名和仓库目录名不需要一致，但存在上述对应关系。换句话说只有在版本库中进一步指定与项目对应的一个目录，才能让 SVN 服务器可以为不同项目服务，否则如果让 SVN 服务器只能支持一个项目就太浪费资源了。

2.2.3 如果操作成功[D:\DevRepository\svnRep\AtguiguERP] 目录下会出现四个文件夹两个文件



3 启动 SVN 服务

SVN 服务器仅仅安装和配置好是不够的，还需要让它运行起来，才能响应客户端请求。启动 SVN 服务器有两种方式，命令行方式和注册服务的方式

3.1 命令行方式

① `svnserve -d -r` 文档仓库路径[-d 表示后台执行 -r 表示版本库的根目录]

如：`svnserve -d -r D:\DevRepository\svnRep\AtguiguERP`

②如何验证 svn 服务器已经启动？SVN 服务监听 3690 端口，打开一个新的 cmd 窗口，使用 `netstat -an` 命令查看 3690 端口是否被监听

```
C:\Users\Phenix>netstat -an
```

活动连接

协议	本地地址	外部地址	状态
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:623	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1158	0.0.0.0:0	LISTENING
TCP	0.0.0.0:2401	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3306	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3690	0.0.0.0:0	LISTENING
TCP	0.0.0.0:5520	0.0.0.0:0	LISTENING
TCP	0.0.0.0:10108	0.0.0.0:0	LISTENING

③运行 `svnserve` 的 cmd 窗口关闭后，对应的 svn 服务就会被停止

3.2 注册 Windows 服务

①利用 XP、2000 以上系统自带工具 Service Control，执行文件是 `sc.exe`

②注册服务命令：

i.命令格式：

sc create 要创建的服务名 binpath= "SVN 安装目录\bin\svnserve.exe
--service -r 版本库目录" displayname= "要创建的服务名" start= auto
depend= Tcpip

例如：

[单仓库方式]

```
sc create SVN-Service binpath= "
D:\DevInstall\Subversion\bin\svnserve.exe --service -r
D:\DevRepository\svnRep\AtguiguERP" displayname= " SVN-Service"
start= auto depend= Tcpip
```

[多仓库方式]

```
sc create SVN-Service binpath= "
D:\DevInstall\Subversion\bin\svnserve.exe --service -r
D:\DevRepository\svnRep " displayname= " SVN-Service" start= auto
depend= Tcpip
```

ii.说明：binpath 指定 svnserve 的路径和命令 -r 版本库的路径。start= auto 表示服务自动启动。

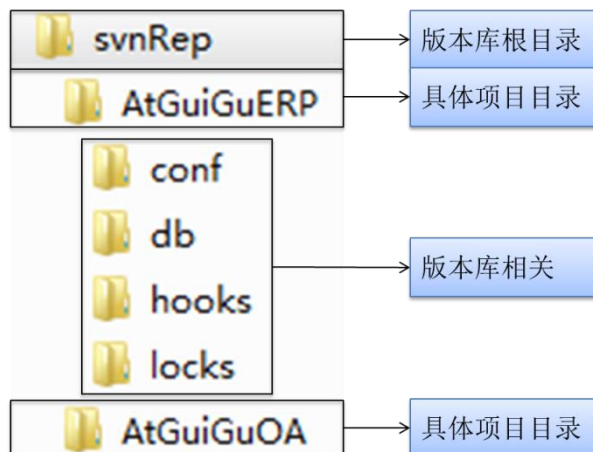
iii.切记：“=” 左边没有空格，“=” 右边一定要有一个空格！

iv.对于 Windows7 及以上系统，在命令行中直接运行上述命令会返回错误提示[SC] OpenSCManager 失败 5:原因是没有权限。

有两种方法可以解决：一是以管理员身份运行 cmd 程序，二是将命令保存到批处理文件中，再以管理员身份运行。

v.关于“单仓库”和“多仓库”

版本库目录结构是“根目录”下设置具体的“项目目录”，而我们使用 svnadmin create 命令创建版本库时使用的是具体的“项目目录”。这样一来，与版本库相关的文件就会放到“项目目录”下。



在注册 SVN 服务的命令中，如果版本库路径指向的是具体的“项目目录”，例如：D:\DevRepository\svnRep\AtguiguERP，则注册的服务将只能为 AtguiguERP 这个目录对应的项目服务，这种方

式我们称之为：“单仓库”；而如果指定版本库目录时仅指定根目录，例如 `D:\DevRepository\svnRep`，则注册的服务可以根据访问地址判断客户端请求的是哪个项目目录，这种方式我们称之为：“多仓库”。

指定为根目录则配置为多仓库[推荐]

指定方式：-r `D:\DevRepository\svnRep`

访问方式 `svn://localhost/AtguiguERP`

这样 SVN 服务器就可以服务于很多个项目

指定为一个具体的仓库则为单仓库

指定方式：-r `D:\DevRepository\svnRep \AtguiguERP`

访问方式 `svn://localhost`

这样虽然访问方式简单，但整个 SVN 服务器只能为一个项目提供服务，浪费资源

vi.启动服务 `net start SVN-Service`[Win7 以上系统同样需要管理员权限]

vii.其它相关命令：

`sc query SVN-Service` 查询当前服务状态

`sc stop SVN-Service` 停止服务

`sc delete SVN-service` 删除服务，注意：删除服务之前先停止服务

4 Subversion 权限配置

4.1 与权限配置信息相关的文件都保存在[版本库根目录 \AtguiguERP\conf]目录下

4.2 `svnserve.conf` 文件配置

①配置格式

i.#表示当前行被注释

# auth-access = write	注释状态
-----------------------	------

ii.去掉#后，配置项要顶格写，不能留空格

12 anon-access = write	有效状态
------------------------	------

②anon-access 匿名访问权限

read: 可读

write: 可写

none: 没有任何权限

②auth-access 授权用户通过认证后的访问权限

③`password-db = passwd` 指定保存用户名密码的文件：默认为同目录下的 `passwd` 文件

④authz-db = authz 指定配置授权信息的文件:默认为同目录下的 authz 文件

4.3 passwd 文件: 创建用户名密码

①格式说明

[users] --> 固定格式, 不要删除改动

harry = harryssecret --> 示例, 表示设置格式为: 用户名 = 密码

sally = sallyssecret --> 示例

②设置方式: 注意要顶格写, 前面不要有空格, 用户名密码区分大小写

user1 = user1pwd

user2 = user2pwd

③示例:

```
[users]
# harry = harryssecret
# sally = sallyssecret
user1 = 123456
user2 = 123456
user_read = 123456
user_none = 123456
```

4.4 authz 文件: 为用户和用户组设定访问权限

①创建用户组

[groups] --> 固定格式, 表示群组

harry_and_sally = harry,sally --> 示例, 表示设置格式为:
组名 = 成员 1,成员 2,...,成员 n

```
21 [groups]固定格式
22 # harry_and_sally = harry,sally 注释里面是示例
23 # harry_sally_and_joe = harry,sally,&joe
24 user_group = user1,user2 组名=用户1, 用户2
```

设置方式 usergroup = user1,user2

②权限设定格式说明

[repository:/baz/fuz] --> 固定格式, 表示为版本库授权, 格式为“版本库名称:/版本库根目录/子目录”此时设置的是版本库根目录下 baz/fuz 子目录的权限

@harry_and_sally = rw --> 示例, 表示为群组授权, 注意为群组授权前面要加@符号, rw 表示可读可写

* = r --> 示例, *表示未设置的所有用户, r 表示可读

③设置方式

[AtguiguERP:/] --> 设置版本库 AtguiguERP 的根目录的权限信息

@usergroup = rw --> 表示 usergroup 下的所有用户都对版本库 atguigu 的根目录及其以下目录有读写权限

user3 = rw --> 表示用户 user3 有读写权限

* = --> 表示除已设置的用户外，其他用户没有任何权限，通过这种方式可以屏蔽其他用户的访问

如果想设置根目录下 doc 目录，只允许 user2 有读写权限，其他用户没有任何权限，则应设置为：

[atguigu:/doc] --> SVN 权限设置具有继承性，对根目录的设置，对根目录以下的目录同样有效

@usergroup = rw

user3 = rw

* =

[atguigu:/doc]

user2 = rw

* = --> 因为 SVN 权限设置具有继承性，所以如果不指定 * = ，则除 user2 外的其他用户的权限也将作用到 doc 目录上，所以设置 * = 的作用就是屏蔽其他用户

④在实际工作中，SVN 服务器上保存的通常不仅仅是程序代码，还可能保存项目文档，这时有些目录就可能是中文目录。

而如果在 authz 文件中出现中文，则权限设置就无法生效。此时需要借助 UltraEdit 工具，将 authz 文件另存为 UTF-8 无 BOM 格式