



我来教你学 Java

专题课程

主讲: Reyco

北京动力节点教育科技有限公司

动力节点课程讲义

DONGLIJIEDIANKECHENGJIANGYI www.bjpowernode.com



专题: SVN 版本控制技术

第1章 SVN 概述

1.1SVN 简介

1.1.1 什么是 SVN

SVN 是 SubVersioN 的简称,是 Apache 研发的一个开源的版本控制系统。

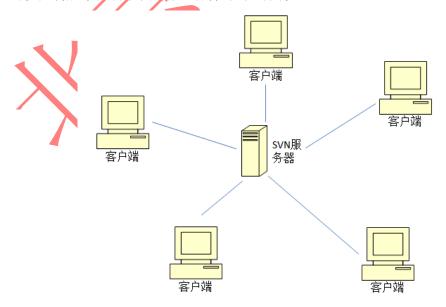
1.1.2 SVN 的作用

使用 SVN 的目的主要有两个:

- 实现项目组中的资源共享
- 实现资源文件的版本控制

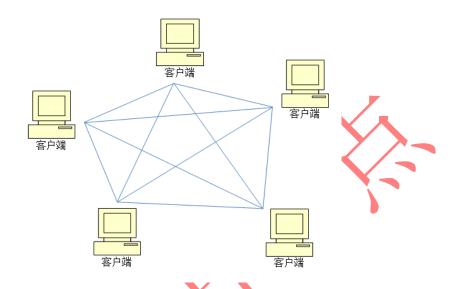
1.1.3 SVN 的工作原理

在 Subversion 管理下,文件和目录可以超越时空。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。正因为如此,许多人将版本控制系统当作一种神奇的"时间机器"。



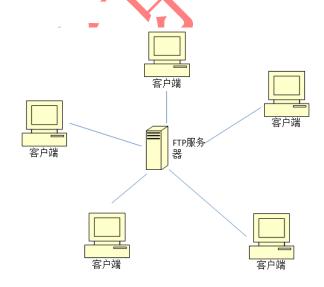
1.2传统文件共享方式的弊端

1.2.1 拷贝方式的弊端



- 发送与接收文件的工作量巨大
- 无法及时获取到更新
- 对于之前版本的恢复较难,即使可以恢复,也大量浪费空间。
- 若一个文件由多个修改,则会出现内容冲突问题。且难于确定修改的具体时间及修改人。
-

1.2.2 文件服务器方式的弊端

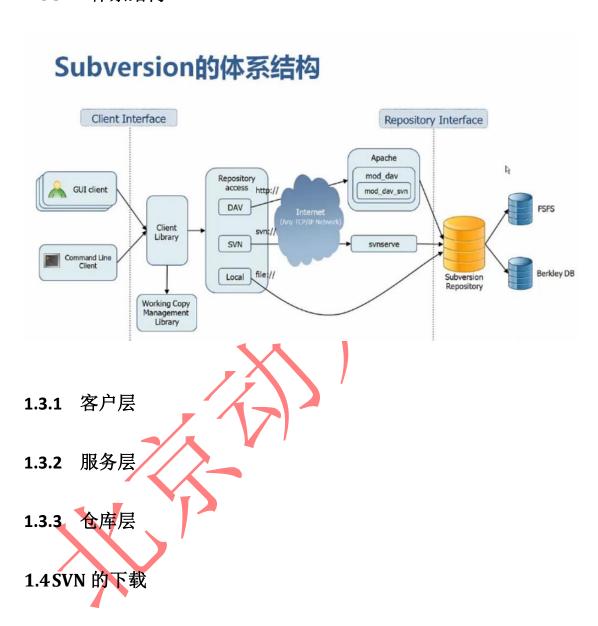


- 空间大量浪费
- 资源无法设定访问权限



- 资源安全性很低
- 发生冲突资源,无法自动合并
-

1.3 SVN 体系结构



第2章 SVN 命令行的使用

- 2.1SVN 的安装与配置
- 2.2 SVN 常用命令
- 2.2.1 SVN 管理员命令
 - (1) svnadmin help
 - (2) svnadmin help create
 - (3) svnadmin -version
 - (4) svnadmin create

SVN 仓库分为两级:顶层仓库与根仓库。

该命令用于创建 SVN 版本仓库,确切说,创建的是根仓库。

注意,在创建根仓库时,项层仓库目录必须是存在的,其不会自动创建。根仓库目录是 否存在,均是可以的。若根仓库目录不存在,命令会自动创建该根仓库目录。

🚾 管理员: 命令提示符

C:\WINDOWS\system32<mark>/</mark>svnadmin create d:\course\svn\repository\sms

C:\\INDO\S\system32\svnadmin create d:\course\svn\repository\ems

2.2.2 SVN 服务端命令

(1) synserve -h

获取帮助



C:\Users\Reycosvnserve -h usage: svnserve [-d | -1 | -t | -X | --service] [options] Subversion repository server. Type 'svnserve --version' to see the program version. Valid options: -d [--daemon] : daemon mode -i [--inetd] : inetd mode -t [--tuppe1] : tuppe1 mode

(2) synserve-d

用于开启 DOS 系统下的 SVN 服务。 SVN 服务默认的端口号为 3690。



netstat –a 命令查看当前网络的连接状态。 net status svn://localhost:3690/d:/course/svn/repository/sms 等价于

svn://localhost/d:/course/svn/repository/sms

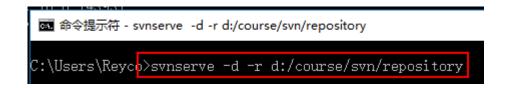
(3) synserve-d-listener-port=8888

指定 SVN 服务占用的端口号为 8888。 svn://localhost:8888/d:/course/svn/repository/sms

```
配命令提示符 - svnserve -d --listen-port=8888
C:\Users\Reyc<mark>o</mark>>svnserve -d --listen-port=8888
```

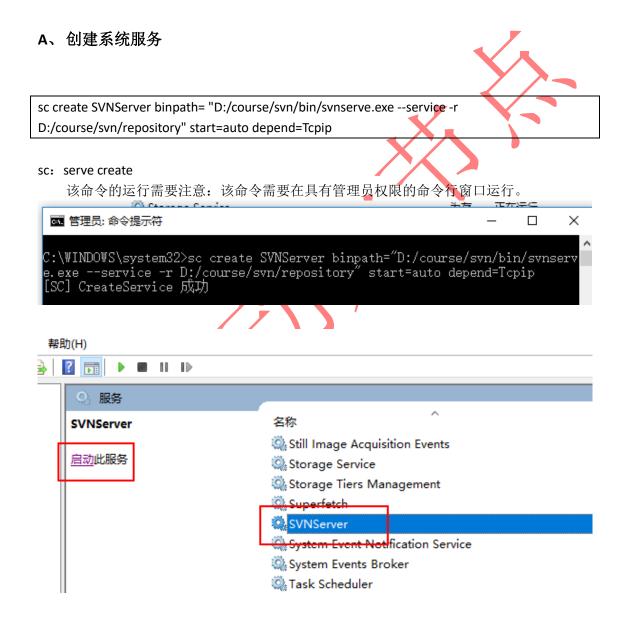
(4) synserve -d -r

指定默认的 SVN 顶层仓库的路径。一旦指定,那么客户端在使用 SVN 时直接给出根仓库名即可。



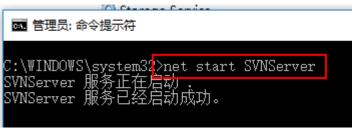
svn://localhost/sms

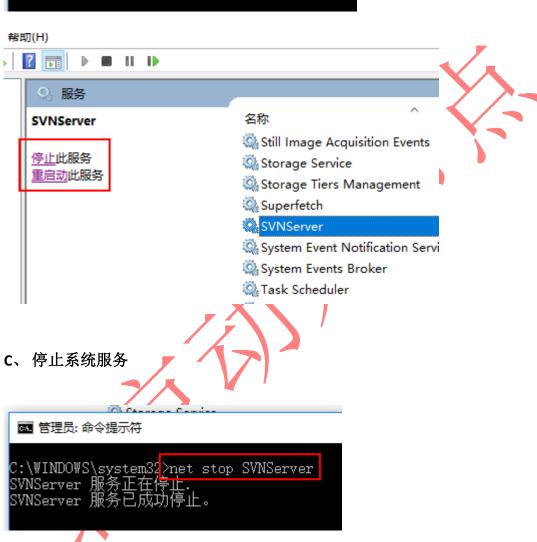
(5) 将 SVN 服务注册为开机自启动的 Win 服务

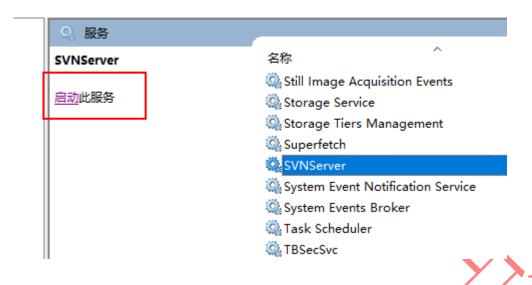




B、启动系统服务

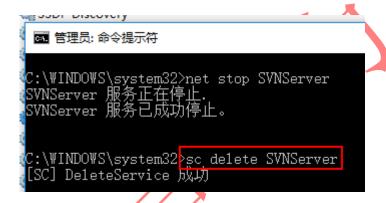






D、删除系统服务

删除之前,最好先停止系统服务。



2.2.3 SVN 客户端命令

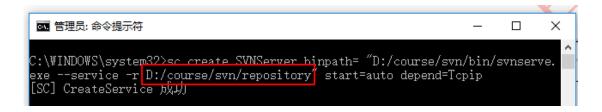
(1) syn checkout

checkout,称为检出。其意义为:创建客户端指定目录与服务端指定根仓库间的连接关系。客户端指定的这个目录,官方名称为 working copy,通常称为客户端连接目录。

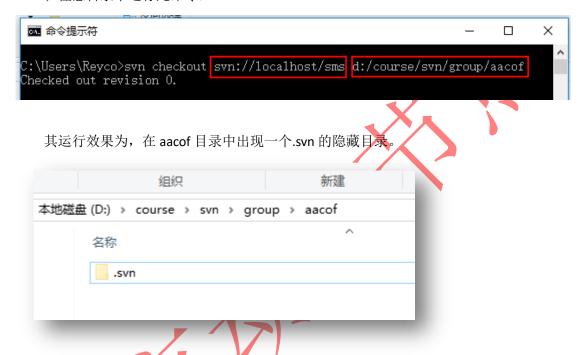
一个客户端一般情况下,只需要检出一次。



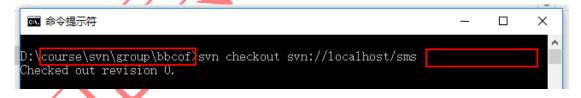
A、基于顶层仓库的 checkout



在任意目录下运行此命令:

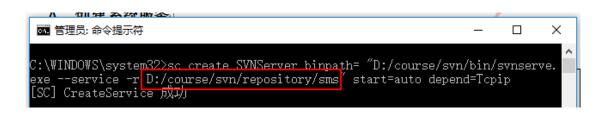


若当前执行命令的目录为 Working Copy 目录,则运行以下命令后的结果为:

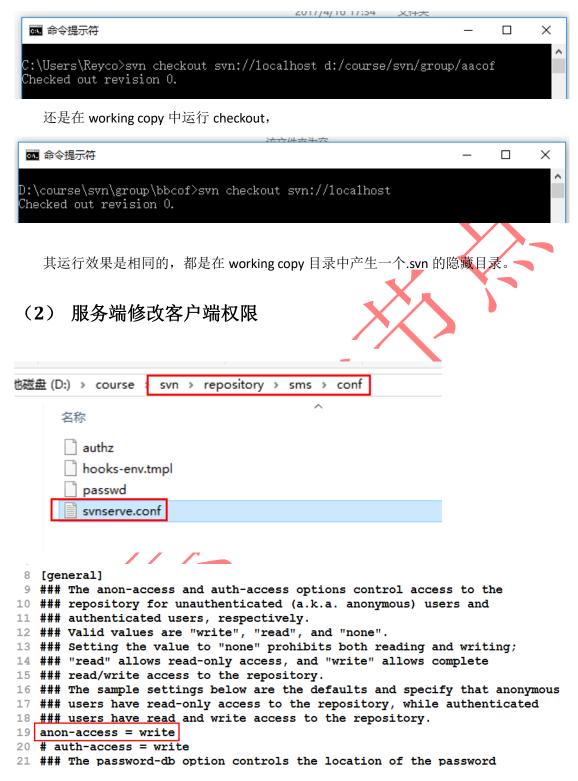


在 working copy 中会出现根仓库目录,而根仓库目录中会存在.svn 隐藏目录。

B、基于根仓库的 checkout



无论是在任意目录运行 checkout,



(3) syn add

当一个文件/目录,被存放到 Working Copy 中时, SVN 并不会感知到它们的存在,即 SVN 并不会对其进行管理。若要使 SVN 对其进行管理,必须将其通知 add 命令,添加到 SVN 管



理中。

需要注意,

- (1) 被 add 的文件/目录,必须存在于 Working Copy 中
- (2) add 命令的作用就是将指定文件/目录交由 SVN 进行管理,所以一个文件/目录一般情况下,就执行一次 add 命令即可。add 命令的执行与文件是否被修改过没有任何关系
- (3) 被 add 的目录,会将当前目录及其所包含的所有文件/目录,一次性都交给 SVN 管理

A、add 文件

区:\WINDOWS\system32>svn add d:/course/svn/group/aacof/Students.txtA D:\course\svn\group\aacof\Students.txt

B、add 目录

(4) svn commit

commit 命令用于将客户端 Working Copy 中所有对文件/目录的操作提交到服务端。注意:

- commit 命令必须携带参数-m,用于完成日志记录
- -m 参数与目标文件在 commit 命令中的顺序是可颠倒的
- 提交后的文件,在服务端是无法直接看到的。
- 对于已经提交过的文件,在没有被修改的情况下再次提交是没有意义的。

(5) syn update

该命令用于将当前客户端 Working Copy 中的文件/目录更新到与 SVN 服务端相同版本。

(6) syn delete

用于删除指定的文件/目录。当然,这个删除仅仅删除的是客户端自己的,并不会影响服务端的文件/目录。只有当前客户端执行了 commit 命令,才会将这个删除操作同步到 SVN 服务端。

需要注意:

- 这个操作在当前客户端本地效果等同于在 Window 中执行的删除操作,但该 windows 下的删除通过 svn commit 操作无法同步到 SVN 服务端。
- 对于目录的 svn delete 操作,会将当前目录及其所包含的所有文件/目录全部删除。

(7) svn revert

该命令用于恢复客户端被删除的文件/目录。但这个文件/目录的删除操作不能是被commit 过的。

(8) syn list

该命令用于获取当前目录中所包含的所有文件/目录

(9) svn info

该命令用于获取当前 SVN 客户端与服务端的相关信息。

(10) svn help

用于获取 SVN 命令的帮助信息。可以查看所有的 SVN 命令,也可以获取到指定命令的帮助信息。



第3章 TortoiseSVN 客户端

3.1下载与安装

3.1.1 下载

需要下载两个安装程序: SVN 客户端安装程序与 SVN 语言包安装程序。均可从 Apache 官网下载。

3.1.2 安装

安装过程很简单,一直下一步即可。

3.2 可视化命令操作

3.2.1 创建服务端根仓库

3.2.2 客户端图标不显示

Windows 支持 Overlay Icon(覆盖图标)最多 15 个。Windows 本身及其它应用程序已经 将这 15 个 Overlay Icon 占用,所以 SVN 安装成功后,图标没有变化。

3.2.3 客户端检出 Checkout

3.2.4 客户端导入 Import

在要导入内容所在目录上右击,或在要导入的内容所在的目录中右击,选择 TortoiseSVN/Import,即可完成导入。需要注意的是,导入时并不会将内容所在目录本身导 入到 SVN 服务端。

3.2.5 客户端更新 Update

3.2.6 客户端导出 Export

需要注意的是,在哪个目录上右击选择的导出,该目录本身是不会被导出的,只会导出 该目录中的内容。

3.2.7 客户端添加 Add

3.2.8 客户端提交 Commit

3.2.9 客户端删除 Delete

使用 SVN 下的删除操作,与使用 Windows 中的删除操作,效果相同。

3.2.10 客户端恢复 Revert

3.2.11 客户端返回指定版本 Update to revision

3.3冲突问题

当一个文件在两个客户端都被修改后,第一个客户端进行提交是没有问题的,会将其修改过的内容提交到服务端。但第二个客户端在进行提交时会出问题,不能完成提交。这就是冲突问题。



3.3.1 冲突分类

根据冲突引发的具体原因的不同,可以将冲突分为两类: 异行修改冲突,与同行修改冲突。

(1) 异行修改冲突

多个客户端对同一版本文件做出的修改,可以是任意的添加、修改、删除操作,只要他们修改的不是同一行数据内容,那么此时引发的冲突称为异行修改冲突。 **▼**

由于不同的客户端的修改行是不相同的,所以 SVN 给出了比较简单的冲突解决方案: 将修改进行合并,即大家的修改都起作用。

(2) 同行修改冲突

只要多个客户端对同一版本文件的同一行数据进行了添加、修改、删除操作,此时引发的冲突称为同行修改冲突。

由于不同客户端对于同一版本文件的修改内容是不同的, SVN 无法对各个客户端的修改进行取舍,即无法自动给出冲突解决方案。此时只能由人工进行冲突内容的选择,即由人工完成取舍。

3.3.2 同行修改冲突解决

3.3.3 异行修改冲突解决

3.3.4 混合修改冲突解决

与同行修改冲突解决方式相同。

- 3.4为用户添加权限
- 3.4.1 SVN 服务器设置
 - (1) 为认证用户设置权限
 - (2) 注册密码配置文件
 - (3) 注册权限配置文件
 - (4) 指定要应用的根仓库
- 3.4.2 为用户设置密码
- 3.4.3 为用户设置权限
- 3.5 文件锁
- 3.5.1 为文件加锁



3.5.2 获取文件锁

3.5.3 删除文件锁



第4章 Eclipse SVN 客户端

- 4.1 SVN 插件下载与安装
- 4.1.1 SVN 插件下载





4.2常规用法

- 4.2.1 分享项目
- 4.2.2 查看根仓库内容
- 4.2.3 检出
- 4.2.4 添加
- 4.2.5 提交
- 4.2.6 更新
- 4.2.7 删除后的恢复
- 4.2.8 查看历史版本
- 4.2.9 查看本地历史版本
- 4.2.10 返回指定版本
- 4.3冲突问题
- 4.3.1 同行修改冲突
- 4.3.2 异行修改冲突
- 4.4为文件加锁
- 4.4.1 加锁



为文件加锁后,需要注意:

- 加锁文件,只有加锁用户才可修改、提交
- 文件加锁操作会直接自动提交到 SVN 服务端,即文件加锁后,不用再做 Commit 操作来 将加锁信息提交到 SVN 服务端
- 文件加锁后的自动提交没有更改文件的版本号
- 默认情况下对加锁文件修改后的提交操作,会使文件的锁自动解除

4.4.2 解锁





第5章 SVN 与 Apache 服务器整合

使用 SVN 自身的服务协议 svn://可以实现客户端与服务端的通讯,但一般仅限于局域网内。若要从外网访问 SVN 服务器,一般是通过使用 http 协议完成的,且服务端所使用的服务器一般为 Apache 服务器。

5.1 Apache 服务器

Apache 服务器,确切地说,其名称为 Apache Http Server。

5.1.1 Apache 服务器的下载

Apache 官网不提供 Apache 服务器的二进制运行文件,只提供源码。但可以从 Apache 官网查找 Apache 服务器的二进制文件的推荐下载地址。

5.1.2 Apache 服务器 zip 版的安装

- (1) 解压
- (2) 配置
- (3) 修改

5.1.3 常用命令

命令需要在管理员权限的命令行下运行。

(1) httpd

不带任何参数的该命令,可以启动 Apache 服务。但该启动方式的弊端是,当前命令行一直处于等待状态,还不能关闭。一旦关闭,Apache 服务停止。一般不使用该方式。

(2) httpd-h

该命令用于获取 httpd 的帮助信息。

(3) httpd-k install

该命令将 Apache 服务器的服务安装为了一个 Windows 服务。默认该 Windows 服务的名称为 Apache2.4。

(4) httpd -k uninstall

卸载名称为 Apache2.4 的 Windows 服务。

(5) httpd -k install -n "Apache-Server"

将 Apache 服务安装为一个名称为 Apache-Server 的 Windows 服务。

(6) httpd -k start -n "Apache-Server"

该命令用于启动名称为 Apache-Server 的服务。若不指定-n 参数,那么默认启动的是名称为 Apache2.4 的服务。

(7) httpd -k stop -n "Apache-Server"

该命令与 httpd - k shutdown -n "Apache-Server"等价。

(8) httpd -k restart -n "Apache-Server"

该命令用于重启 Apache-Server 服务。

5.1.4 Apache 服务启动成功测试

在浏览器地址栏中输入 http://localhost,则在打开的页面中可以看到"It Works"字样,则表示 Apache 服务启动成功。



5.1.5 Apache 服务器 msi 版的安装

5.2SVN 与 Apache 服务器整合

5.2.1 安装 SVN 服务端 msi 版本

5.2.2 SVN 与 Apache 服务器的整合步骤

(1) 修改 httpd.conf 文件

打开第83、84行注释,用于加载两个so文件。

so 文件,即 Shared Object,共享对象文件,是 Linux 系统下的动态链接库文件,类似于 Windows 下的.dll 文件。

(2) 复制两个 so 文件

将 SVN 服务器安装目录的 bin 目录下的两个 so 文件复制到 Apache 服务器安装目录的 modules 目录中。



(3) 再修改 httpd.conf 文件

在第 85、86 行,插入如下内容,用于加载刚刚复制来的两个 so 文件。 LoadModule dav_svn_module modules/mod_dav_svn.so LoadModule authz_svn_module modules/mod_authz_svn.so

(4) 复制 dll 文件

从 SVN 服务端安装目录的 bin 目录中,复制 8 个 dll 动态链接库文件到 Apache 服务器安装目录的 bin 目录下。因为刚刚复制的两个 so 文件的运行还依赖于这 8 个 dll 文件。











libsvn_client-1.dll libsvn_delta-1.dll libsvn_diff-1.dll libsvn_fs-1.dll libsvn_ra-1.dll







libsvn_repos-1.dll libsvn_subr-1.dll libsvn_wc-1.dll

(5) 运行 htpasswd 命令

这是 Apache 服务器的命令,用于为 SVN 用户分配访问 Apache 服务器的用户名与密码。 这个命令需要在管理员权限的命令行运行。

htpasswd -cb apachepwd.conf aa 222

htpasswd -b apachepwd.conf bb 222

htpasswd -b apachepwd.conf cc 222

(6) 复制密码文件

将刚刚生成的 apachepwd.conf 文件复制到 SVN 根仓库的 conf 目录中。

(7) 第三次修改 httpd.conf 文件

在 httpd.conf 文件最后添加如下内容:

#指定访问使用的 URI

<location /svn/ems>

#指定远程要访问的模块为 SVN

DAV svn

#指定 SVN 根仓库路径

SVNPath D:/course/svn/repository/ems

#指定 SVN 授权文件路径

AuthzSVNAccessFile D:/course/svn/repository/ems/conf/authz

#指定所有用户都需要进行身份验证

Satisfy Any

Require valid-user

#指定验证方式为基础验证,即密码验证

AuthType Basic

#指定项目名称

AuthName "ems"

#指定用户密码文件路径

AuthUserFile D:/course/svn/repository/ems/conf/apachepwd.conf

</location>



(8) 整合成功标志

配置完毕后,重启 Apache 服务器,然后在地址栏中输入: http://localhost/svn/ems, 会弹出输入用户名与密码的对话框,输入 httpsswd 命令中生成的用户名与密码后,在页面可以看到 ems - Revision 0: /字样,表示整合成功。

5.2.3 整合后的使用

- (1) 在 TortoiseSVN 客户端使用
- (2) 在 Eclipse 客户端使用
- 5.3 Visual SVN Server

5.3.1 安装

安装前需要将之前安装好的 SVNServer、Apache 服务器删除和卸载。

5.3.2 SVN 服务器设置

(1) 创建根仓库

https://DESKTOP-PQLK8M0/svn/abc/

- (2) 创建用户
- (3) 创建用户组
- (4) 权限分配

5.3.3 在 TortoiseSVN 客户端中使用

5.3.4 在 Eclipse 客户端中使用

5.4根仓库的标准目录结构

5.4.1 目录用法说明

一般的用法是: trunk: 作为主开发目录 branches: 作为分支开发目录

tags: 作为存档目录

5.4.2 目录用法示例

5.4.3 删除保存过的身份验证数据

打开"开始菜单->TortoiseSVN 目录->Settings",找到 Saved Data,单击 Authentication Data 后的 Clear 按钮,即可将保存过的身份验证数据删除。

