## 2.1SVN的安装与配置

安装，配置环境变量。

## 2.2SVN常用命令

### 2.2.1管理员命令

svnadmin help

svnadmin help create

svnadmin --version

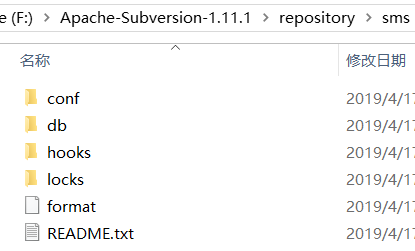
svnadmin create 根仓库路径

创建SVN版本仓库，确切说创建的是根仓库。

SVN仓库分为两级：顶层仓库与根仓库。

注意：在创建根仓库时，顶层仓库目录必须是存在的。根仓库目录是否存在均是可以的。

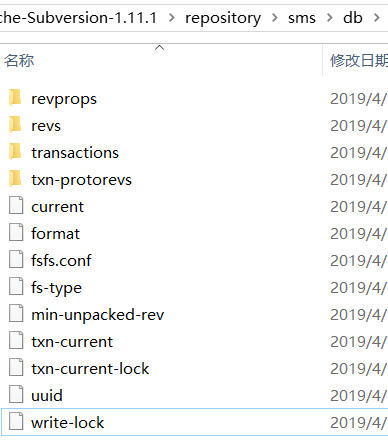




存放配置文件：conf



存放具体的版本数据内容：db



### 2.2.2服务端命令

1. svnserve -h

获取帮助

1. svnserve -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。

svn://localhost:3690/ f:/Apache-Subversion-1.11.1/repository/sms

等价于

svn://localhost / f:/Apache-Subversion-1.11.1/repository/sms

1. svnserve -d –listen-port=8888

指定SVN服务占用的端口号为8888。

svn://localhost:8888/ f:/Apache-Subversion-1.11.1/repository/sms

1. svnserve -d -r 路径

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:8888/sms

### 2.2.3将SVN服务注册为开机自启动的Win服务

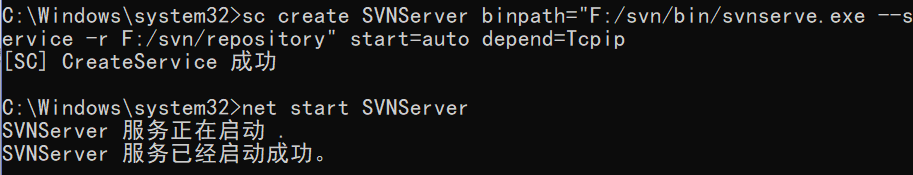
1. 创建系统服务

sc create SVNServer binpath="F:/svn/bin/svnserve.exe --service -r F:/svn /repository" start=auto depend=Tcpip

注意：该命令的运行需要在具有管理员权限的命令行窗口运行。

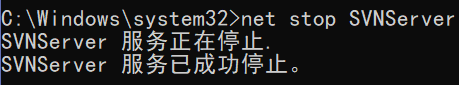
1. 启动SVNServer服务

net start SVNServer



1. 停止SVNServer服务

net stop SVNServer



1. 删除SVNServer服务

删除之前，最好先停止SVNServer服务。

sc delete SVNServer

### 2.2.4客户端命令（重点）

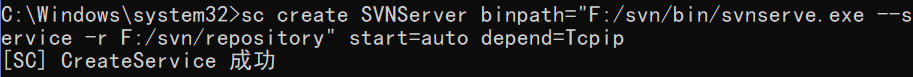
1. svn checkout（检出）

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，学名working copy

一个客户端一般情况下，只需要检出一次。

checkout命令用来从版本库检出一个工作副本。

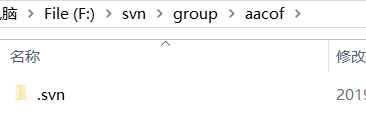
1. 基于顶层仓库的checkout



在任意目录下运行此命令：



其效果为，在aacof目录中出现一个.svn的隐藏文件。

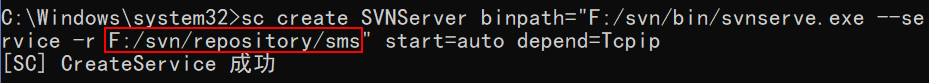


若当前执行的目录为working copy目录，则运行以下命令后的结果为：



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。

1. 基于根仓库的checkout（使用较多）

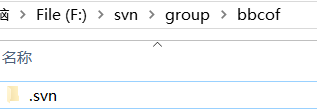


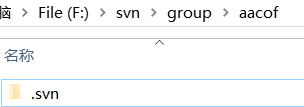
无论当前执行的目录为working copy目录或者任意目录运行checkout：





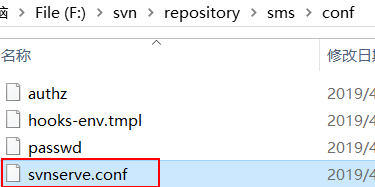
其运行效果是相同的，都是在working copy目录中产生一个.svn的隐藏项目：

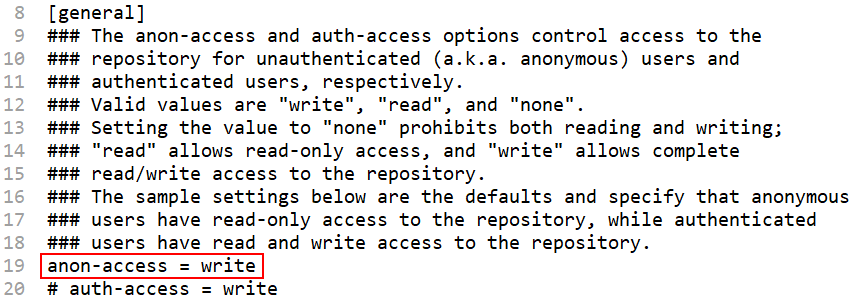




1. 服务端修改客户端权限

修改根仓库下项目里的配置文件svnserve.conf。





1. svn 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，必须将其通过add命令，添加到svn管理中。

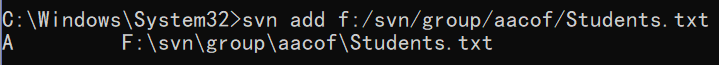
需要注意：

被add的文件/目录，必须存在于working copy中；

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，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；

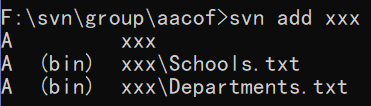
被add的目录，会将当前目录及其包含的所有文件/目录，一次性都交给svn管理。

1. add文件





1. add 目录



1. 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。

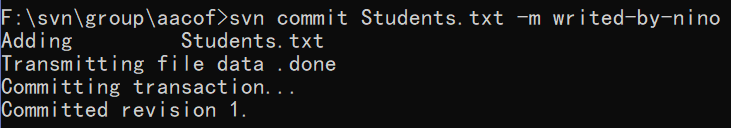
注意：

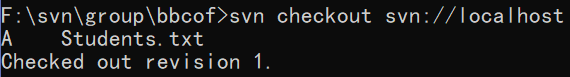
commit命令必须携带参数-m用于完成日志记录；

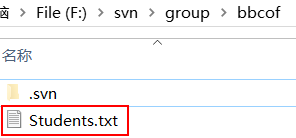
-m参数与目标文件在commit命令中的顺序是可颠倒的；

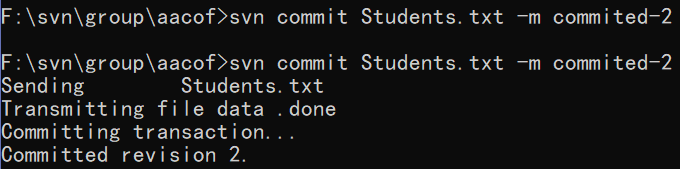
提交后的文件在服务端是无法直接看到的；

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。



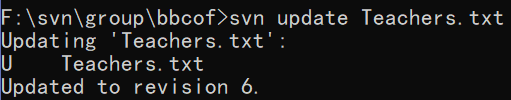


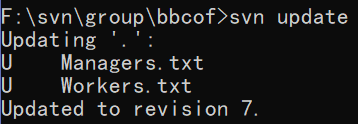




1. svn update

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。





1. svn delete

用于删除指定的文件/目录，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。

注意：

这个操作在当前客户端在当前客户端本地效果等同于在Windows中执行删除操作，但该删除通过svn commit操作无法同步到svn服务端；

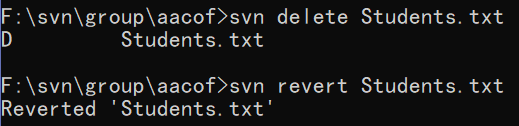
对于目录的svn delete操作，会将当前目录及其所包含的所有文件/目录全部删除。



1. svn revert

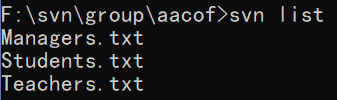
该命令用于恢复被客户端删除的文件/目录。

注意：一旦被commit就不能被恢复了。



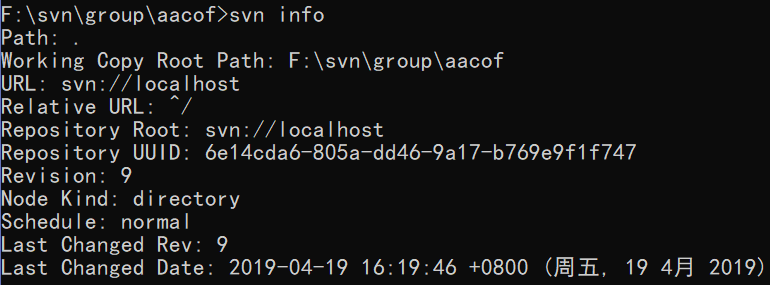
1. svn list

该命令用于获取当前目录中所包含的所有文件/目录



1. svn info

该命令用于获取客户端与服务端的相关信息。



1. svn help

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

