

出自: crazylinux 工作室--侯在存  
 E-mail:houzaicunsky@gmail.com  
 时间: 2010.12.17 转载请保留此信息



# SVN+APACHE 整合使用手册

## 一、SVN 介绍

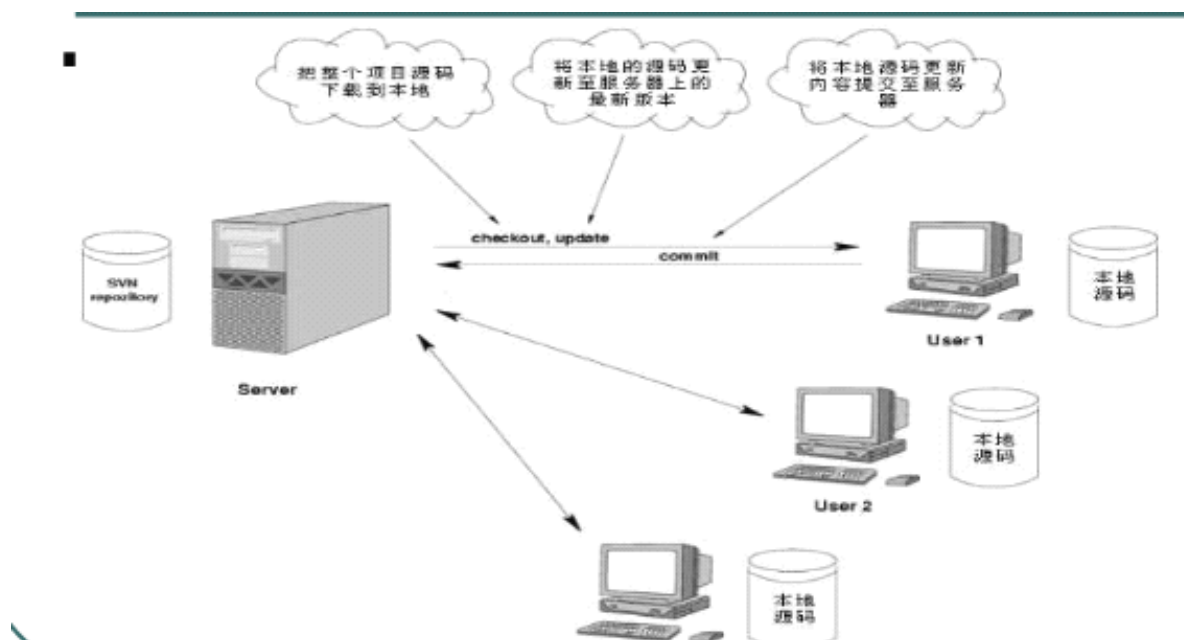
SVN 是一个版本控制工具，Subversion 的版本库 (repository)，就是位于服务器端，统一管理和储存数据的地方。

要创建一个版本库，首先要确定采用哪种数据存储方式。在 Subversion 中，版本库的数据存储有两种方式，一种是在 Berkeley DB 数据库中存放数据；另一种是使用普通文件，采用自定义的格式来储存，称为 FSFS。

表 1 两种版本库数据存储对照表

特性	Berkeley DB	FSFS
对操作中中断的敏感	很敏感；系统崩溃或者权限问题会导致数据库“塞住”，需要定期进行恢复。	不敏感
可只读加载	不能	可以
存储平台无关	不能	可以
可从网络文件系统访问	不能	可以
版本库大小	稍大	稍小
扩展性: 修订版本树数量	无限制	某些本地文件系统在处理单一目录包含上千个条目时会出现问题。
扩展性: 文件较多的目录	较慢	较慢
检出最新代码的速度	较快	可以
大量提交的速度	较慢，但时间被分配在整个提交操作中	较快，但最后较长的延时可能会导致客户端操作超时
组访问权处理	对于用户的 umask 设置十分敏感，最好只由一个用户访问。	对 umask 设置不敏感
功能成熟时间	2001 年	2004 年

Svn 工作原理图示：



## 二、SVN 的搭建

### 1、所需软件包

httpd-2.2.4

apr-1.3.8.tar.gz、apr-util-1.3.9.tar.gz

sqlite-amalgamation-3.6.17.tar.gz

subversion-1.6.5.tar.bz2

expat-1.95.7-4、expat-devel-1.95.7-4

客户端：TortoiseSVN-1.6.5.16974-win32-svn-1.6.5.msi、

### 2、安装 SVN

安装 apache

```
# ./configure --prefix=/usr/local/apache2
```

```
--enable-dav
```

```
--enable-so
```

```
--enable-modules=most
```

//注：编译 apache 时要加上--enable-dav 选项

```
# make
# make install

安装 apr、apr-util
# tar -zxvf apr-1.3.8.tar.gz -C /usr/src
# cd /usr/src/apr-1.3.8
# ./buildconf //验证系统是否有 python、autoconf、libtool
# ./configure --prefix=/usr/local/apr
# make
# make install
```

```
# tar -zxvf apr-util-1.3.9.tar.gz -C /usr/src
# cd /usr/src/apr-util
# ./configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr
# make
# make install
```

```
安装 sqlite
#./configure --prefix=/usr/local/sqlite
# make
# make install
```

```
安装 svn
# tar -jxvf subversion-1.6.5.tar.bz2 -C /usr/src
# cd /usr/src/subversion-1.6.5/
# ./configure --prefix=/usr/local/svn --with-apxs2=/usr/local/apache2/bin/apxs
--with-apr=/usr/local/apr
--with-apr-util=/usr/local/apr-util
--with-sqlite=/usr/local/sqlite/ --with-ssl --with-zlib 打开 zlib 库支持
--enable-maintainer-mode 打开调试和编译时警告，仅供开发人员使用
# make
# make install
```

### 3、apache 的整合

修改 APache 配置文件

```
#vi /usr/local/apache/conf/httpd.conf
```

```
User www 修改 apache 默认启用用户为 www
```

```
Group www
```

```
找到 DocumentRoot “/usr/local/apache/htdocs” 和 </Directory
“/usr/local/apache/htdocs”> 修改 apache 的主页到/WWW 下
```

```
DocumentRoot “/www”
```

```
<Directory “/www”>
```

配置 apache 中 svn 认证

```
<Location /svn>                                //设置 url, 代表要访问的仓库的父目录
  DAV svn
  SVNParentPath /data/svn                      //svn 根目录
  AuthType Basic
  AuthName "SVN Test"
  AuthzSVNAccessFile /data/svn/authz.conf //svn 认证文件, 创建一个 authz.conf
  文件
  AuthUserFile /data/svn/authfile             //apache 认证文件
  Require valid-user
</Location>
```

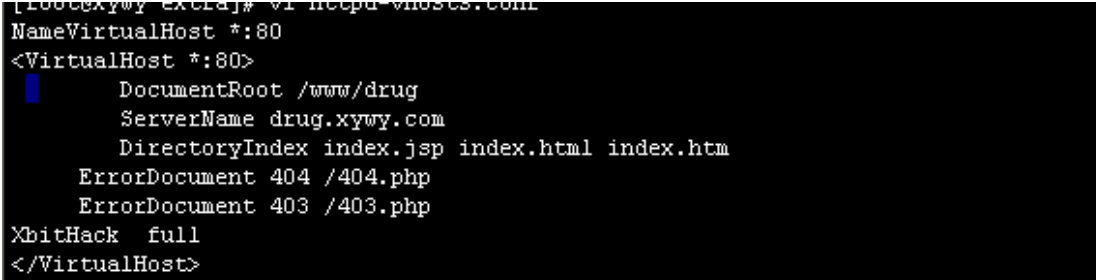
</Location>

启动 apache

```
/usr/local/apache/bin/apachectl start
```

绑定域名

```
#cat /usr/local/apache/conf/extra/httpd-vhosts.conf
```



```
[root@xywy ~]# vi /etc/httpd/vhosts.conf
NameVirtualHost *:80
<VirtualHost *:80>
  DocumentRoot /www/drug
  ServerName drug.xywy.com
  DirectoryIndex index.jsp index.html index.htm
  ErrorDocument 404 /404.php
  ErrorDocument 403 /403.php
  XbitHack full
</VirtualHost>
```

测试域名绑定成功

## 4、SVN 使用

创建 svn 项目仓库

```
# mkdir /data/svn                                创建 svn 根目录
# svnadmin create /data/svn/drug //创建项目仓库 drug
```

```
# /usr/local/svn/bin/svn import /www/drug file:///data/svn/drug -m "import "
将/www/drug 导入到 svn 库下
```

注: 在导入仓库后, 原来的目录树并没有转化成一个工作副本, 需要使用 checkout 手动导出一个工作副本。

```
#chown -R www:www /data/svn/drug
```

```
#svn checkout file:///data/svn/drug /www/drug_new
修改目录的属主为 www
```

```
#chown -R www:www /www/drug_new
```

```
#chmod -R 700 /data/svn/
```

编写 svn 与 apache 同步脚本

```
#vi /data/svn/drug/hooks/post_commit
```

```
#!/bin/sh
```

```
export LC_CTYPE=zh_CN.GB18030 语言支持
```

```
SVN=/usr/local/svn/bin/svn          指定 svn 路径
WEB=/www/drug                       指定 apache 虚拟主机路径
$SVN update $WEB -username www -password xywyxywy
赋予脚本执行权限
#chmod 700 post-commit
#chown www: www post-commit
创建 apache 认证用户
# /usr/local/apache2/bin/htpasswd -(c)m /data/svn/authfile sum
输入密码: sum123
# /usr/local/apache/bin/htpasswd /data/svn/authfile wyj
//再添加一个用户
```

### Svn 权限配置文件

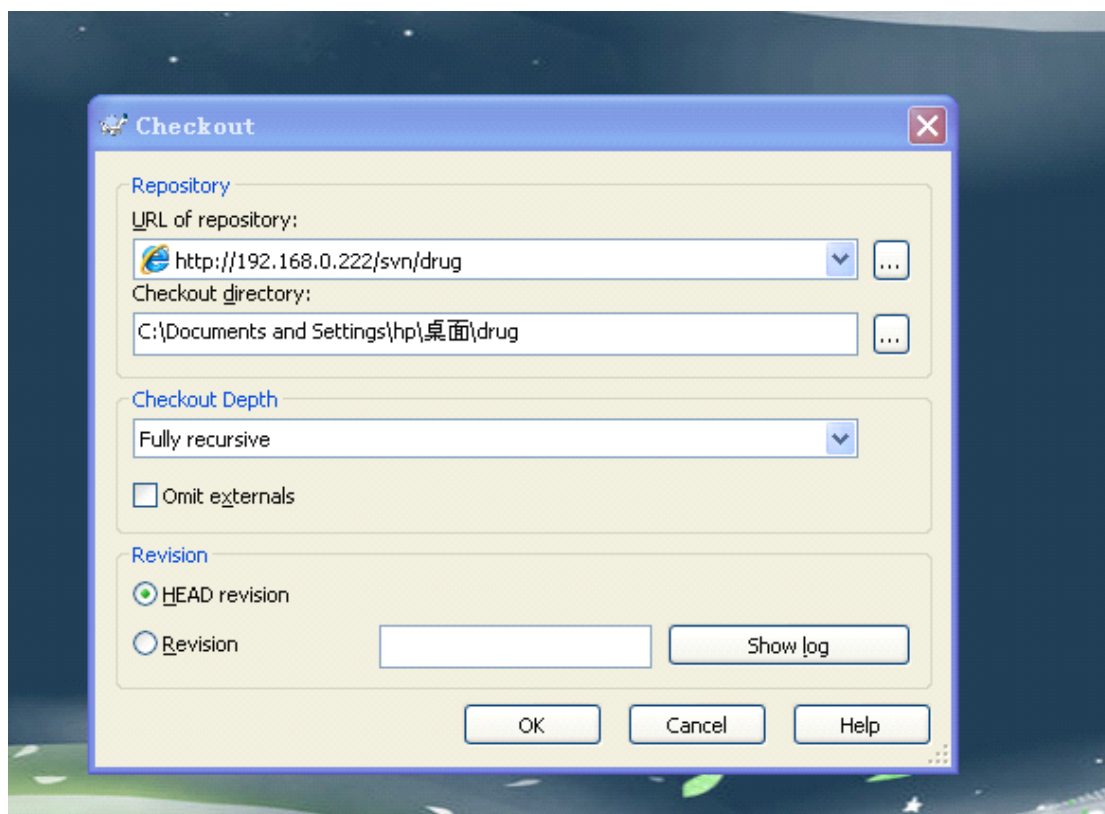
```
# vi /data/svndata/authz.conf
[drug:/]          //表示仓库 drug 根目录
sum = rw          //用户 sum 对 drug 库有读写权限
hou = r           //用户 wyj 对 drug 库有读写权限
[/]
* = r             //这个表示对所有的用户都具有读权限
[groups]         //这个表示群组设置
developers = sum, hou //developers 组中的用户
[pro1:/]
@svn1-developers = rw //对这个组有读写权限
```

### 启动 svn 服务

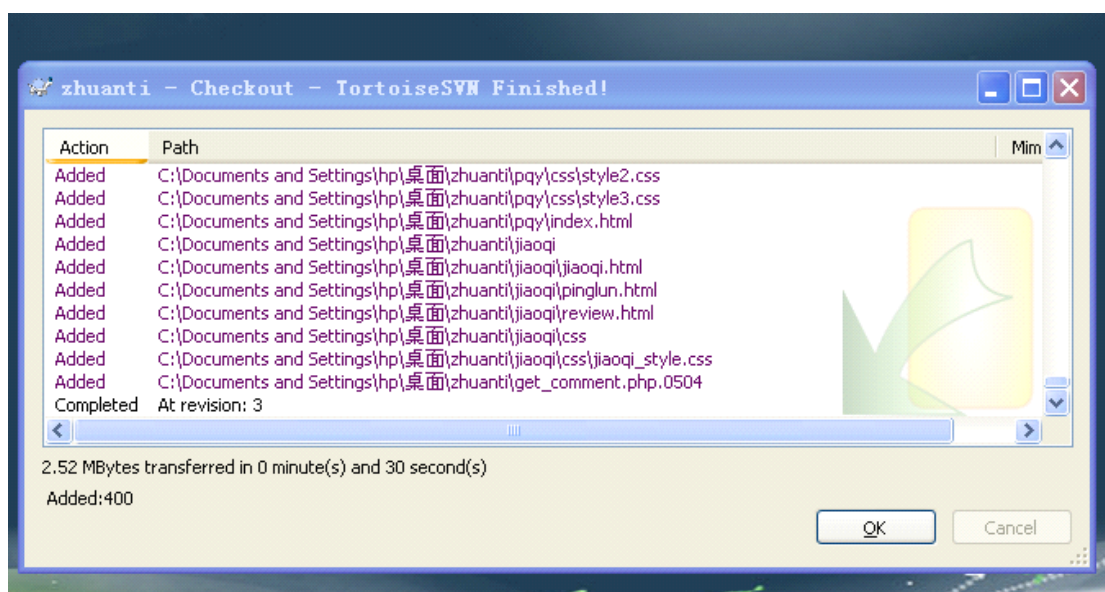
```
svnserver -d -r /data/svndata
设置 svn 和 apache 服务开机自动启用
#vi /etc/rc.local
/usr/local/apache/bin/apachectl start
svnserv -d -r /data/svn
```

## 5.客户端使用

在客户机上安装 TortoiseSVN-1.6.5.16974-win32-svn-1.6.5.msi  
在客户机上从服务器上 checkout 工作副本



在 URL of repository 一栏中输入 svn 服务器的 checkout 地址



在新的项目里面添加一个文件然后提交，查看 `www/drug/` 里面有没有添加的文件，来证实 svn 库能够同步更新到 linux 下面那个虚拟主机的没。

### 三、SVN 管理命令

## 1、svn 命令

导入数据到版本库

```
# svnadmin create /data/svndata/pro1
# svn import /www/shop file:///data/svndata/pro1 -m "import"
```

查看版本库下的内容

```
# svn list file:///data/svndata/pro1
```

注：在导入仓库后，原来的目录树并没有转化成一个工作副本，需要使用 checkout 手动导出一个工作副本。

从版本库导出数据

```
# svn checkout file:///data/svndata/pro1
```

取出任意深度的数据

```
# svn checkout file:///data/svndata/pro1/ad/html
```

将数据放到新目录

```
# svn checkout file:///data/svndata/pro1 testweb
```

//将 pro1 工作副本放到 testweb 中，而不是默认生成的 pro1 目录

```
# svn co file:///data/svn/my_xywy_com/ /www/other/my.xywy.com
```

// 导出 my\_xywy\_com 全部文件，到/www/other/my.xywy.com 下

它可以显示工作副本中的所有项目

```
# svn status -u -v
```

18	16 sum	fff.txt
18	1 root	index2.php
18	10 sum	index20090416.html
18	17 sum	111/fff.txt
18	17 sum	111/index2.php
18	17 sum	111/index20090416.html
18	17 sum	111/index.jsp

## 2、svnlook 命令

查看当前的版本仓库内容

```
# svnlook info 版本仓库
```

发展开源精神，普及 Linux 教育

<http://blog.sina.com.cn/houzaicun>

```
Sum                                     //作者
2009-09-10 11:51:17 +0800 (Thu, 10 Sep 2009) //提交时间
0                                       //最后一次提交参数信息
```

查看最近一次更新的修订号

```
# svnlook youngest 版本仓库
```

显示一个版本仓库中文件和目录的树形结构图

```
# svnlook tree /data/svndata/pro2/ --show-ids
```

```
jytk.html <0-1.0.r1/232049>
gallery.php <3-1.0.r1/232223>
gywm.html <5-1.0.r1/232402>
database.php <7-1.0.r1/232579>
lxyp.html <b-1.0.r1/232979>
```

查看仓库中某一个文件内容

```
# svnlook cat /data/svndata/pro2/ index.html //查看 pro2 项目的 index.html 文件中的内容
```

显示被修改的文件

```
# svnlook diff /data/svndata/pro2/
```

```
Added: 11.txt //显示增加了 11.txt 文件
```

### 3、导出 svn 版本库中的文件夹

若要彻底删除 SVN 版本库某文件夹，可以使用下面的方法。

查看项目 sumitest 中的列表

```
# svn list file:///data/svn/sumitest
```



```
fxwy/  
fxwy/src  
jxywy/  
jxywy/src  
src/
```

要把 fxywy/src 文件夹删除，重新导入到新的项目 sumitest2 中，方法如下：

```
# svnadmin dump /data/svn/sumitest > sumi1.dump  
// 导出 sumitest 整个项目到 sumi1.dump  
# cat sumi1.dump | /usr/local/svn/bin/svndumpfilter exclude /fxywy/src > sumi2.dump  
// 从 sumi1.dump 过滤出/fxywy/src 目录。Windows 中 cat 换成 type  
# svnadmin create /data/svn/sumitest2  
// 创建新的项目库  
# svnadmin load /data/svn/sumitest2 < sumi2.dump  
//导出/fxywy/src 到 sumitest2 库
```

## 4、删除 svn 版本库中的文件夹

svn delete 可以从工作拷贝或版本库删除一个项目。

```
# svn delete -m "Deleting file" file:///data/svn/sumitest2/fxywy/  
// 删除 sumitest2 下的/fxywy 目录
```

## 四、SVN 管理命令

### 1、CentOS5 安装 svn 时，make 时出现下面的错误

```
/usr/bin/ld: cannot find -lexpat  
collect2: ld returned 1 exit status  
make: *** [subversion/svn/svn] Error 1
```

解决方法

缺少-lexpat 库文件，找安装盘中的

```
# rpm -ivh expat-x.x.x.x.rpm  
# rpm -ivh expat-devel-x.x.x.x.rpm
```

3、在 MyEclipse 进行 Java 开发，用 SVN 进行版本控制。Update 时提示如下错误

```
update -r HEAD D:/xywyworkspace/fxywy
Working copy not locked; this is probably a bug, please report
svn: Directory 'D:\xywyworkspace\fxywy\WebRoot\WEB-INF\classes\com\
svn' containing working copy admin area is missing
```

原因是 eclipse 把 src 文件夹中的.svn 文件夹也“编译”到 WEB-INF/classes 中去了，而复制过来的.svn 中存储的是 src 文件夹中的版本信息，从而导致在更新时出错。

解决方法：

打开 Eclipse 中的 Project->Properties->Java Build Path 菜单，在右侧面板中的“Source”选项卡，在 Excluded 中加入“\*\*/.svn/\*\*”。也就是把.svn 文件夹从编译路径中排除，这样就不会出现上面的问题了。

4、当提交时显示如下错误：

```
attempt to write a readonly database
```

则是因为/data/svn/yishengquan/db 目录中，权限不是 apache 可写的权限。修改 db 目录权限为 apache 可写即可。

```
# chmod www:www /data/svn/yishengquan/db
```

## 一、SVN 介绍

SVN 是一个版本控制工具，Subversion 的版本库 (repository)，就是位于服务器端，统一管理和储存数据的地方。

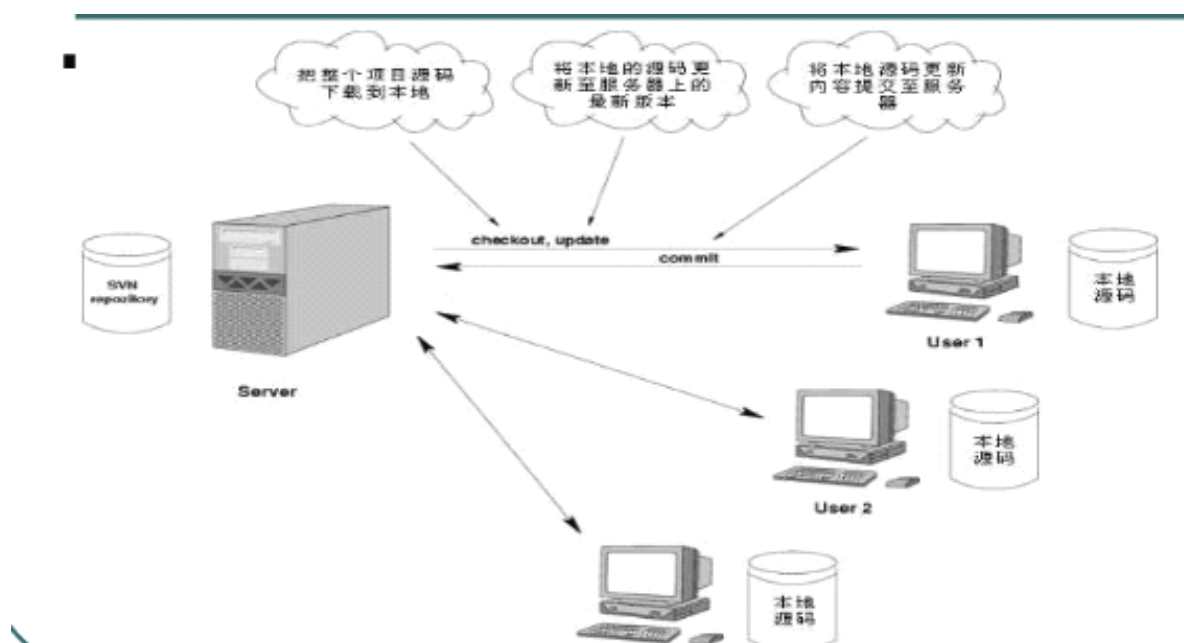
要创建一个版本库，首先要确定采用哪种数据存储方式。在 Subversion 中，版本库的数据存储有两种方式，一种是在 Berkeley DB 数据库中存放数据；另一种是使用普通文件，采用自定义的格式来储存，称为 FSFS。

表 1 两种版本库数据存储对照表

特性	Berkeley DB	FSFS
对操作中断的敏感	很敏感；系统崩溃或者权限问题会导致数据库“塞住”，需要定期进行恢复。	不敏感
可只读加载	不能	可以
存储平台无关	不能	可以

可从网络文件系统访问	不能	可以
版本库大小	稍大	稍小
扩展性: 修订版本树数量	无限制	某些本地文件系统在处理单一目录包含上千个条目时会出现问题。
扩展性: 文件较多的目录	较慢	较慢
检出最新代码的速度	较快	可以
大量提交的速度	较慢, 但时间被分配在整个提交操作中	较快, 但最后较长的延时可能会导致客户端操作超时
组访问权处理	对于用户的 <code>umask</code> 设置十分敏感, 最好只由一个用户访问。	对 <code>umask</code> 设置不敏感
功能成熟时间	2001 年	2004 年

### Svn 工作原理图示:



## 二、SVN 的搭建

## 1、所需软件包

httpd-2.2.4

apr-1.3.8.tar.gz、apr-util-1.3.9.tar.gz

sqlite-amalgamation-3.6.17.tar.gz

subversion-1.6.5.tar.bz2

expat-1.95.7-4、expat-devel-1.95.7-4

客户端：TortoiseSVN-1.6.5.16974-win32-svn-1.6.5.msi、

## 3、安装 SVN

安装 apache

```
# ./configure --prefix=/usr/local/apache2
```

```
--enable-dav
```

```
--enable-so
```

```
--enable-modules=most
```

//注：编译 apache 时要加上--enable-dav 选项

```
# make
```

```
# make install
```

安装 apr、apr-util

```
# tar -zxvf apr-1.3.8.tar.gz -C /usr/src
```

```
# cd /usr/src/apr-1.3.8
```

```
# ./buildconf //验证系统是否有 python、autoconf、libtool
```

```
# ./configure --prefix=/usr/local/apr
```

```
# make
```

```
# make install
```

```
# tar -zxvf apr-util-1.3.9.tar.gz -C /usr/src
```

```
# cd /usr/src/apr-util
```

```
# ./configure --prefix=/usr/local/apr-util --with-apr=/usr/local/apr
```

```
# make
```

```
# make install
```

安装 sqlite

```
#./configure --prefix=/usr/local/sqlite
```

```
# make
```

```
# make install
```

安装 svn

```
# tar -jxvf subversion-1.6.5.tar.bz2 -C /usr/src
```

```
# cd /usr/src/subversion-1.6.5/
```

```
# ./configure --prefix=/usr/local/svn --with-apxs2=/usr/local/apache2/bin/apxs
```

```
--with-apr=/usr/local/apr
```

```

--with-apr-util=/usr/local/apr-util
--with-sqlite=/usr/local/sqlite/ --with-ssl --with-zlib 打开 zlib 库支持
--enable-maintainer-mode  打开调试和编译时警告， 仅供开发人员使用
# make
# make install

```

### 3、apache 的整合

修改 APache 配置文件

```
#vi /usr/local/apache/conf/httpd.conf
```

```
User www          修改 apache 默认启用用户为 www
```

```
Group www
```

```
找到 DocumentRoot “/usr/local/apache/htdocs” 和 </Directory
“/usr/local/apache/htdocs”> 修改 apache 的主页到/WWW 下
```

```
DocumentRoot “/www”
```

```
<Directory “/www”>
```

配置 apache 中 svn 认证

```

<Location /svn>          //设置 url， 代表要访问的仓库的父目录
    DAV svn
    SVNParentPath /data/svn //svn 根目录
    AuthType Basic
    AuthName "SVN Test"
    AuthzSVNAccessFile /data/svn/authz.conf //svn 认证文件， 创建一个 authz.conf
    文件
    AuthUserFile /data/svn/authfile //apache 认证文件
    Require valid-user
</Location>

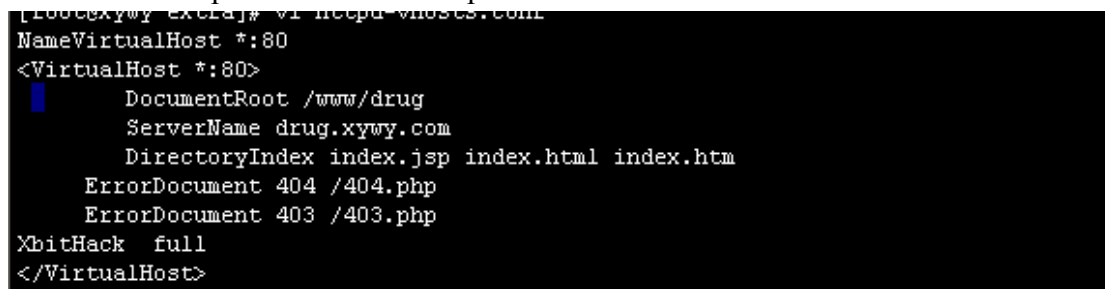
```

启动 apache

```
/usr/local/apache/bin/apachectl start
```

绑定域名

```
#cat /usr/local/apache/conf/extra/httpd-vhosts.conf
```



```

[root@xywy ~]# vi /etc/httpd/conf/extra/httpd-vhosts.conf
NameVirtualHost *:80
<VirtualHost *:80>
    DocumentRoot /www/drug
    ServerName drug.xywy.com
    DirectoryIndex index.jsp index.html index.htm
    ErrorDocument 404 /404.php
    ErrorDocument 403 /403.php
    XbitHack full
</VirtualHost>

```

测试域名绑定成功

### 4、SVN 使用

创建 svn 项目仓库

```
# mkdir /data/svn          创建 svn 根目录
# svnadmin create /data/svn/drug //创建项目仓库 drug
# /usr/local/svn/bin/svn import /www/drug file:///data/svn/drug -m "import" “
将/www/drug 导入到 svn 库下
注：在导入仓库后，原来的目录树并没有转化成一个工作副本，需要使用 checkout
手动导出一个工作副本。
```

```
#chown -R www:www /data/svn/drug
```

```
#svn checkout file:///data/svn/drug /www/drug_new
修改目录的属主为 www
```

```
#chown -R www:www /www/drug_new
```

```
#chmod -R 700 /data/svn/
```

```
编写 svn 与 apache 同步脚本
```

```
#vi /data/svn/drug/hooks/post_commit
```

```
#!/bin/sh
```

```
export LC_CTYPE=zh_CN.GB18030 语言支持
```

```
SVN=/usr/local/svn/bin/svn      指定 svn 路径
```

```
WEB=/www/drug                  指定 apache 虚拟主机路径
```

```
$$SVN update $WEB -username www -password xywyxywy
```

```
赋予脚本执行权限
```

```
#chmod 700 post-commit
```

```
#chown www: www post-commit
```

```
创建 apache 认证用户
```

```
# /usr/local/apache2/bin/htpasswd -(c)m /data/svn/authfile sum
```

```
输入密码：sum123
```

```
# /usr/local/apache/bin/htpasswd /data/svn/authfile wyj
```

```
//再添加一个用户
```

### Svn 权限配置文件

```
# vi /data/svndata/authz.conf
```

```
[drug:]          //表示仓库 drug 根目录
```

```
sum = rw        //用户 sum 对 drug 库有读写权限
```

```
hou = r         //用户 wyj 对 drug 库有读写权限
```

```
[/]
```

```
* = r           //这个表示对所有的用户都具有读权限
```

```
[groups]        //这个表示群组设置
```

```
developers = sum, hou //developers 组中的用户
```

```
[pro1:/]
```

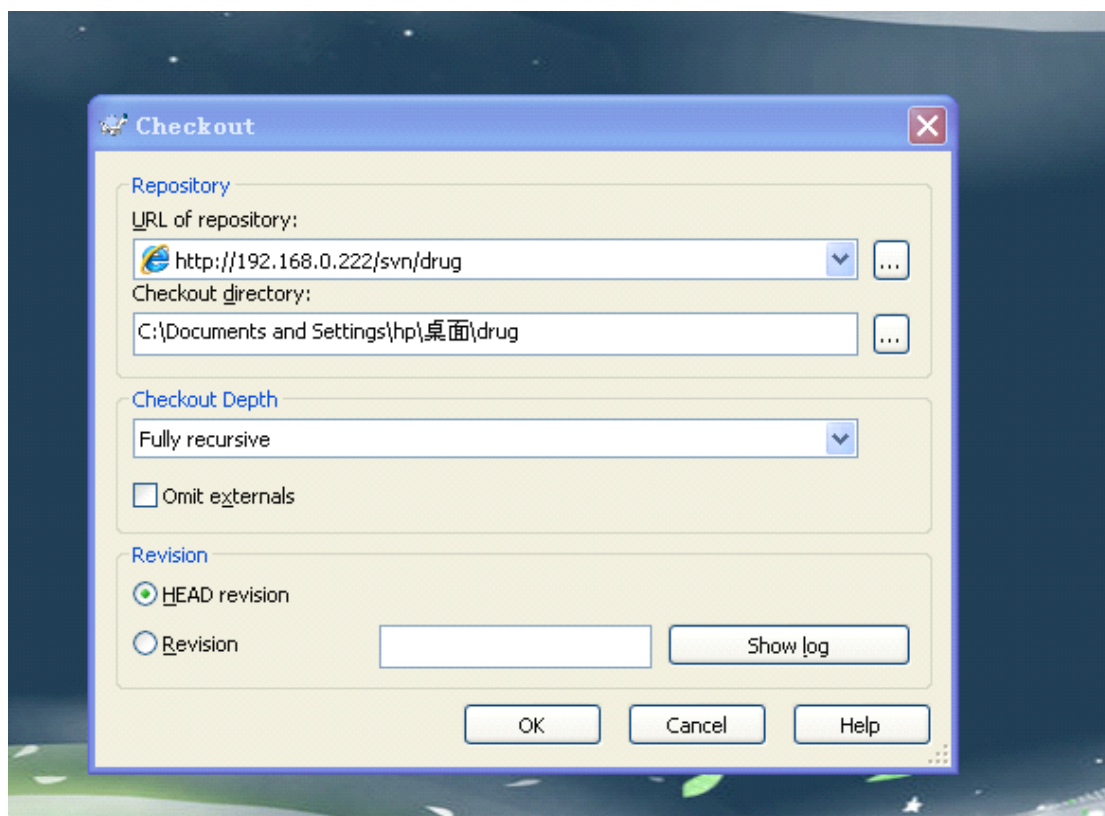
```
@svn1-developers = rw //对这个组有读写权限
```

### 启动 svn 服务

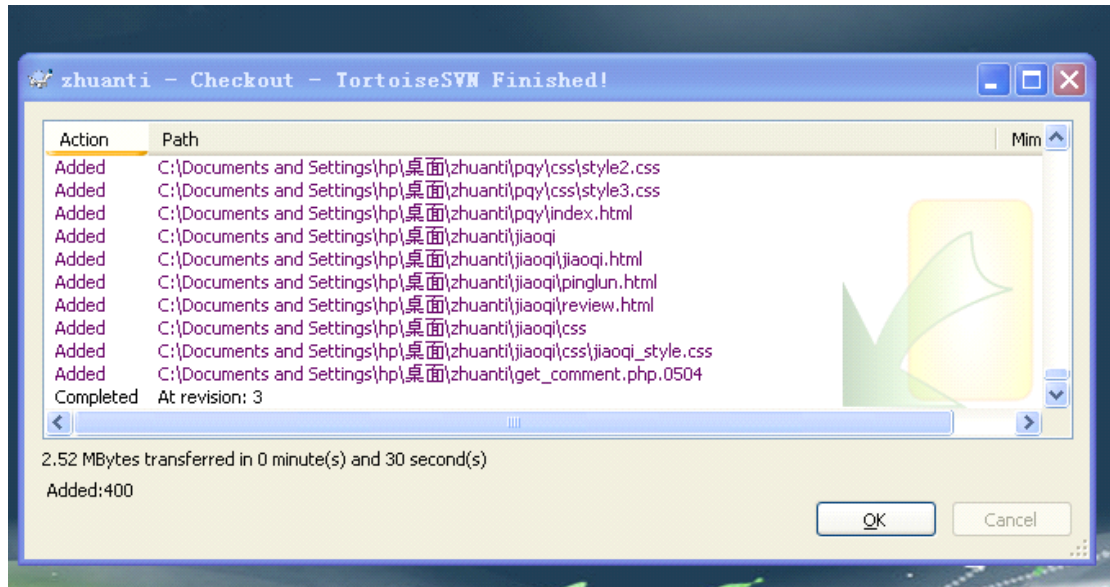
```
svnserver -d -r /data/svndata  
设置 svn 和 apache 服务开机自动启用  
#vi /etc/rc.local  
/usr/local/apache/bin/apachectl start  
svnserve -d -r /data/svn
```

## 6. 客户端使用

在客户机上安装 TortoiseSVN-1.6.5.16974-win32-svn-1.6.5.msi  
在客户机上从服务器上 checkout 工作副本



在 URL of repository 一栏中输入 svn 服务器的 checkout 地址



在新的项目里面添加一个文件然后提交，查看/[www/drug/](http://www.drug/) 里面有没有添加的文件，来证实 svn 库能够同步更新到 linux 下面那个虚拟主机的没。

## 三、SVN 管理命令

### 1、svn 命令

导入数据到版本库

```
# svnadmin create /data/svndata/pro1
```

```
# svn import /www/shop file:///data/svndata/pro1 -m "import"
```

查看版本库下的内容

```
# svn list file:///data/svndata/pro1
```

注：在导入仓库后，原来的目录树并没有转化成一个工作副本，需要使用 checkout 手动导出一个工作副本。

从版本库导出数据

```
# svn checkout file:///data/svndata/pro1
```

取出任意深度的数据

```
# svn checkout file:///data/svndata/pro1/ad/html
```

将数据放到新目录

```
# svn checkout file:///data/svndata/pro1 testweb
```

//将 pro1 工作副本放到 testweb 中，而不是默认生成的 pro1 目录



```
# svn co file:///data/svn/my_xywy_com/ /www/other/my.xywy.com
```

```
// 导出 my_xywy_com 全部文件，到/www/other/my.xywy.com 下
```

它可以显示工作副本中的所有项目

```
# svn status -u -v
```

```
18      16 sum      fff.txt
18       1 root     index2.php
18      10 sum     index20090416.html
18      17 sum     111/fff.txt
18      17 sum     111/index2.php
18      17 sum     111/index20090416.html
18      17 sum     111/index.jsp
```

## 2、svnlook 命令

查看当前的版本仓库内容

```
# svnlook info 版本仓库
```

```
Sum                               //作者
2009-09-10 11:51:17 +0800 (Thu, 10 Sep 2009) //提交时间
0                                   //最后一次提交参数信息
```

查看最近一次更新的修订号

```
# svnlook youngest 版本仓库
```

显示一个版本仓库中文件和目录的树形结构图

```
# svnlook tree /data/svndata/pro2/ --show-ids
```

```
jytk.html <0-1.0.r1/232049>
gallery.php <3-1.0.r1/232223>
gywm.html <5-1.0.r1/232402>
database.php <7-1.0.r1/232579>
lxyp.html <b-1.0.r1/232979>
```

查看仓库中某一个文件内容

```
# svnlook cat /data/svndata/pro2/ index.html //查看 pro2 项目的 index.html 文件中的内容
```

显示被修改的文件

```
# svnlook diff /data/svndata/pro2/
```

```
Added: 11.txt //显示增加了 11.txt 文件
```

### 3、导出 svn 版本库中的文件夹

若要彻底删除 SVN 版本库某文件夹，可以使用下面的方法。

查看项目 sumitest 中的列表

```
# svn list file:///data/svn/sumitest
```

```
fxwy/  
fxwy/src  
jxywy/  
jxywy/src  
src/
```

要把 fxwy/src 文件夹删除，重新导入到新的项目 sumitest2 中，方法如下：

```
# svnadmin dump /data/svn/sumitest > sumi1.dump
```

```
// 导出 sumitest 整个项目到 sumi1.dump
```

```
# cat sumi1.dump | /usr/local/svn/bin/svndumpfilter exclude /fxwy/src > sumi2.dump
```

```
// 从 sumi1.dump 过滤出 /fxwy/src 目录。Windows 中 cat 换成 type
```

```
# svnadmin create /data/svn/sumitest2
```

```
// 创建新的项目库
```

```
# svnadmin load /data/svn/sumitest2 < sumi2.dump
```

```
//导出 /fxwy/src 到 sumitest2 库
```

### 4、删除 svn 版本库中的文件夹

svn delete 可以从工作拷贝或版本库删除一个项目。

```
# svn delete -m "Deleting file" file:///data/svn/sumitest2/fxywy/
```

// 删除 sumitest2 下的/fxywy 目录

## 四、SVN 管理命令

### 1、CentOS5 安装 svn 时，make 时出现下面的错误

```
/usr/bin/ld: cannot find -lexpat
collect2: ld returned 1 exit status
make: *** [subversion/svn/svn] Error 1
```

解决方法

缺少-lexpat 库文件，找安装盘中的

```
# rpm -ivh expat-x.x.x.x.rpm
# rpm -ivh expat-devel-x.x.x.x.rpm
```

### 3、在 MyEclipse 进行 Java 开发，用 SVN 进行版本控制。Update 时提示如下错误

```
update -r HEAD D:/xywyworkspace/fxywy
Working copy not locked; this is probably a bug, please report
svn: Directory 'D:\xywyworkspace\fywy\WebRoot\WEB-INF\classes\com\fywy\
svn' containing working copy admin area is missing
```

原因是 eclipse 把 src 文件夹中的.svn 文件夹也“编译”到 WEB-INF/classes 中去了，而复制过来的.svn 中存储的是 src 文件夹中的版本信息，从而导致在更新时出错。

解决方法：

打开 Eclipse 中的 Project->Properties->Java Build Path 菜单，在右侧面板中的“Source”选项卡，在 Excluded 中加入“\*\*/.svn/\*\*”。也就是把.svn 文件夹从编译路径中排除，这样就不会出现上面的问题了。

### 4、当提交时显示如下错误：

```
attempt to write a readonly database
```

则是因为/data/svn/yishengquan/db 目录中，权限不是 apache 可写的权限。修改 db 目录权限为 apache 可写即可。

```
# chmod www:www /data/svn/yishengquan/db
```

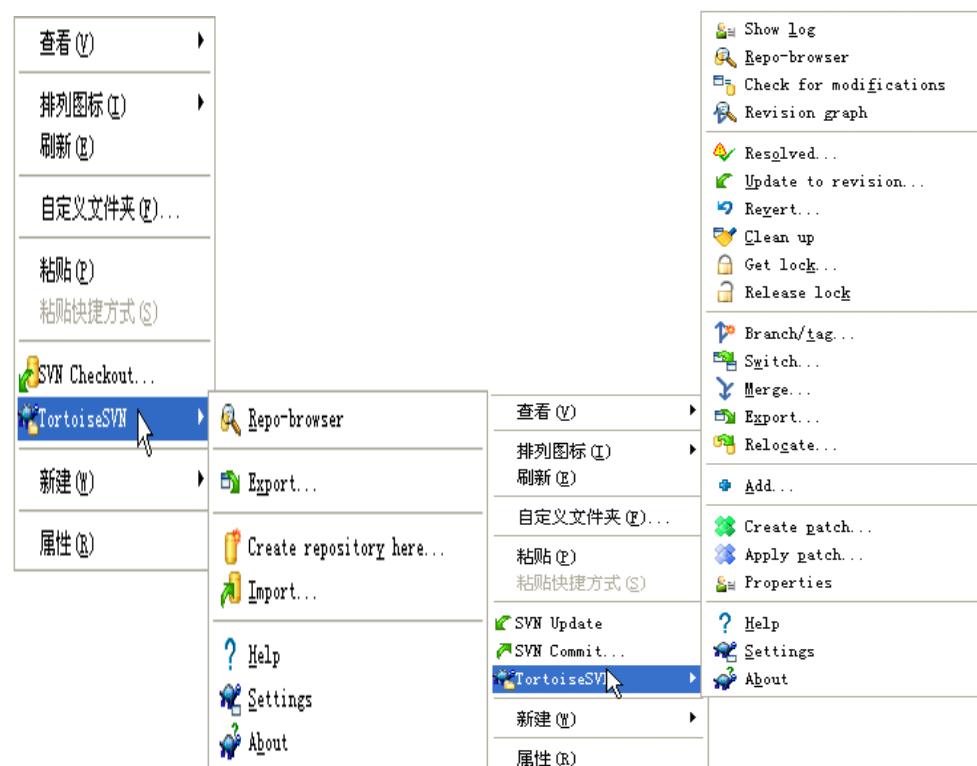
## SVN 客户端使用手册（Windows 平台）

## 安装 TortoiseSVN

点安装程序，按正常步骤下去，安装 TortoiseSVN 完后，需要重启计算机，然后在操作系统的目录浏览器上，点右键，弹出菜单会多出一些 SVN 内容。如图：

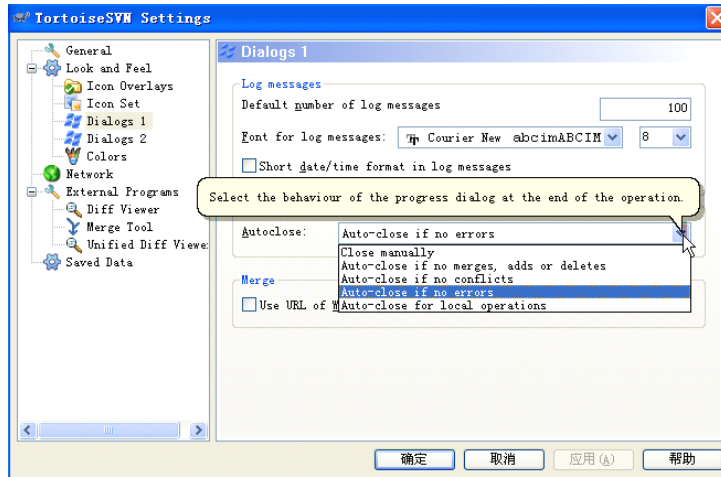


SVN 功能菜单，主要有。如下：



安装后，可设置一些参数，以方便使用，主要如下。

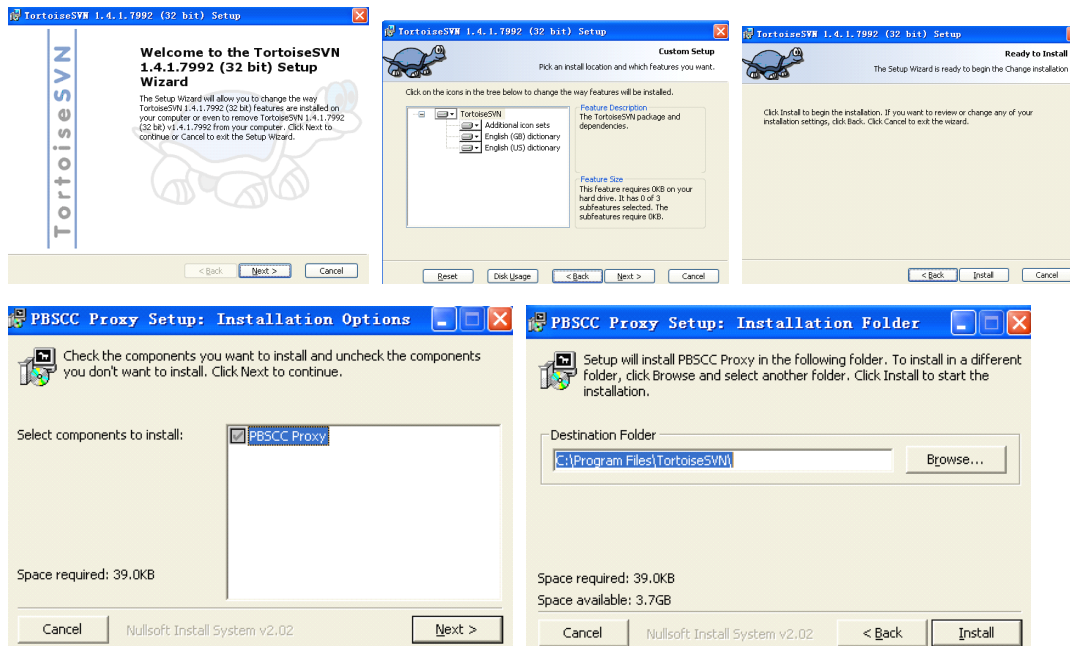
1. 设置提示窗口的参数，在什么情况下，可自动关闭



### PB 使用 SVN 说明

安装方法:

1. 先安装 svn ,再安装 TortoiseSVN, 最后安装 PBsc, 顺序不可反。
2. 在装 PBsc 时, 需要选择原来的“SVN 安装目录”。
3. PBsc 的错误日志, 存放在“TortoiseSVN 目录”下的 PBErr.txt



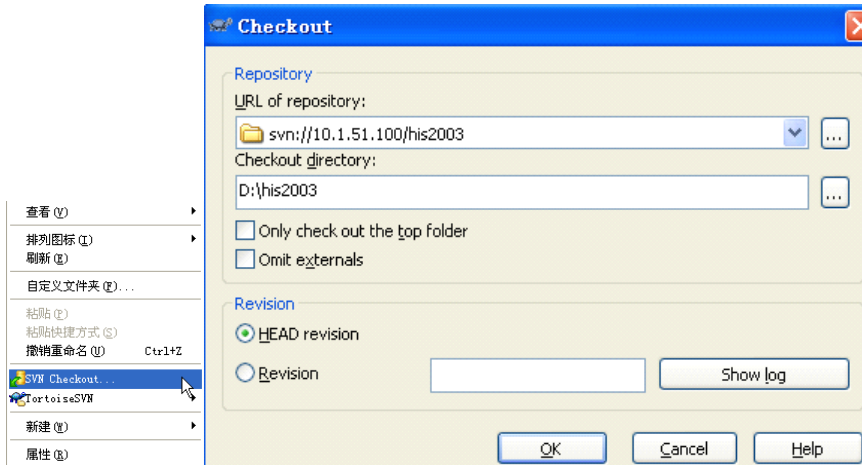
### PB 使用 SVN 总步骤

1. 先要在“工作根目录”(例如: d:\his2003) 建立一个“源代码管理目录”(例如: source)
2. 然后从 SVN 服务器上, 取出代码, SVN 中叫 Check Out  
(注: 跟原来 VSS 的称呼有所不同, 此处签出表示获取之意)
3. 获取后, 再到 PB 中, 设置 PB 环境。

4. 把 PB 程序库加入到“SVN 管理目录”（即此例的 source 目录）。如原来已加入过，就不用此步骤，直接“刷新”PBL 库或其中的对象即可。
5. 使用 PB 的“签出”、“签入”、“取消签出”、“刷新状态”、“查看历史”功能

### 获取 SVN 代码库

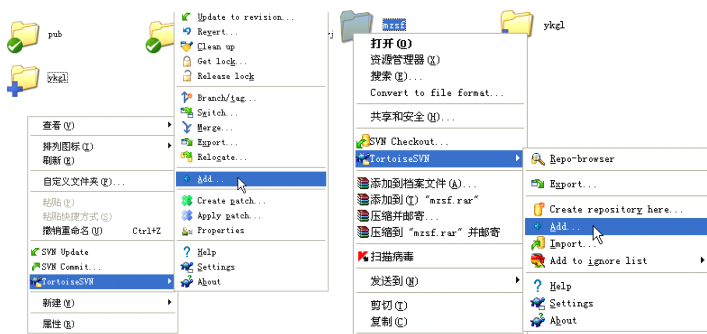
1. 在“工作根目录”建立“源代码管理目录”后，获取 SVN 的“项目库”，称“repository”



2. 取好后，目录图标会改变
3. 在 source 目录下，建立多个目录，一个 PBL 库对应一个目录，文件名同与 PBL 库名。



4. 把建立的多个子目录，加入到（add）到 SVN，图标变成一个“加号+”

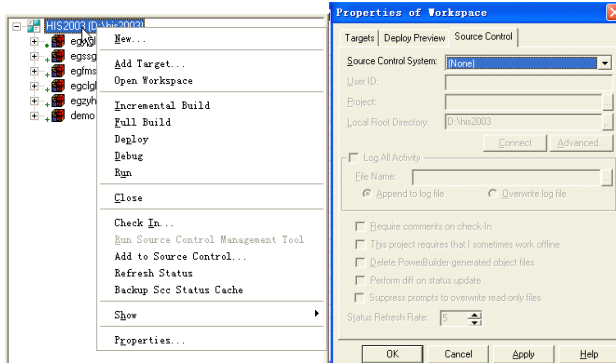


5. 加入到 SVN 后，最后还需要提交到 SVN（commit）

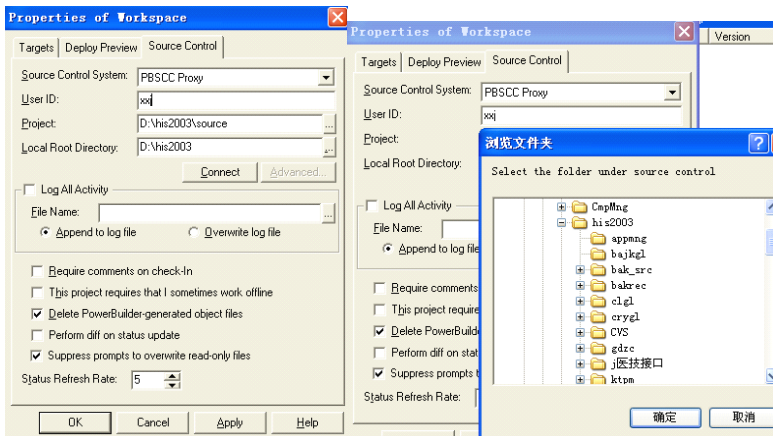


### PB 环境设置:

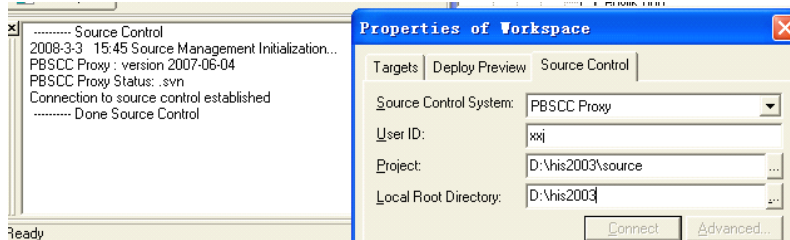
1. 与原来 VSS 的设置大致相同
2. 设置“Workspace”的属性的 Source Control



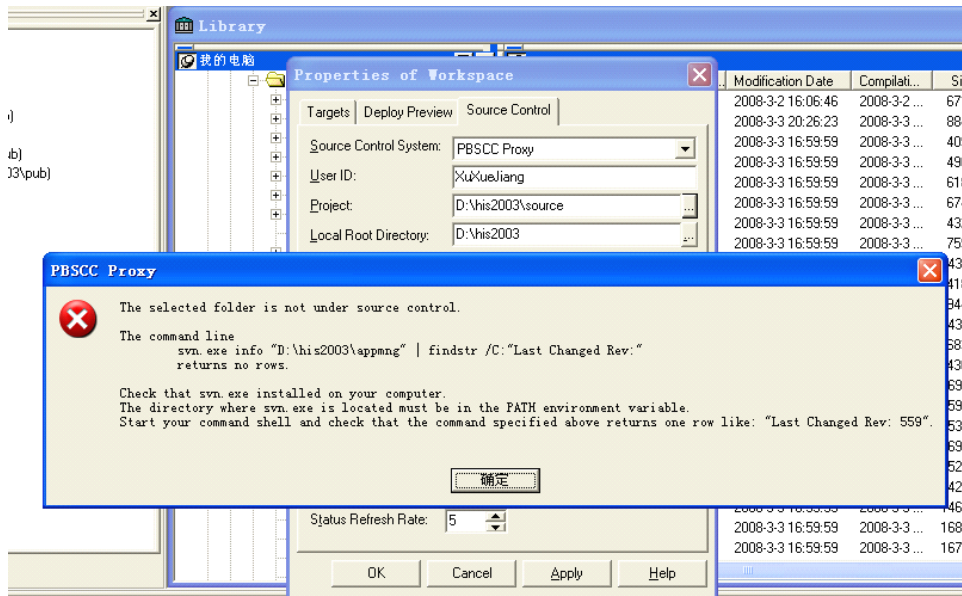
3. 设置“源代码管理工具”，指定用户(User Id)，源代码管理目录(Project)，本地目录(Local Root Directory)，设置完后才能导入代码到 SVN。  
 (注：用户名 UserId，一定要使用 SVN 帐号，而千万不要用默认的本地系统用户)  
 (注：选择 Project 目录时，需先用 SVN 获取到本地)



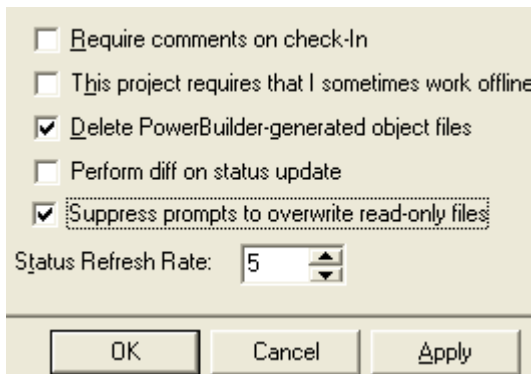
选好“源代码目录”后的图:



如果选中未属于“源码管辖的目录”，选择中时会提示出错:



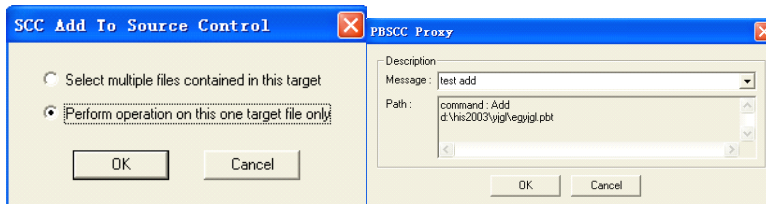
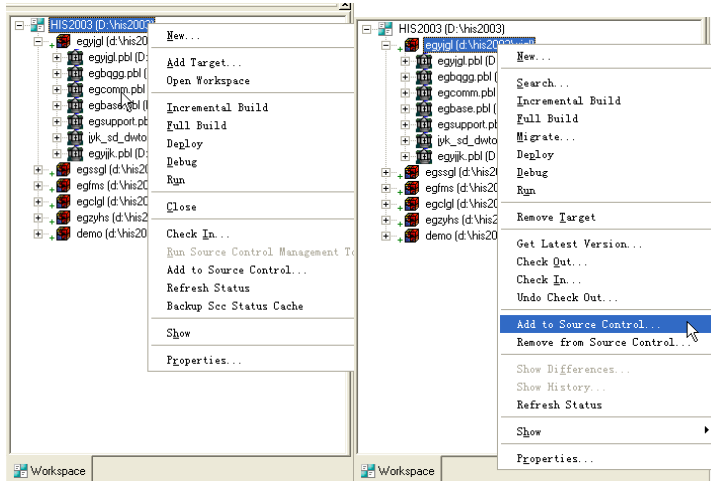
4. 点击《Connect》，登录 SVN 系统，有可能会提示，输入用户密码  
(注：可以记住用户名和密码，免去每次输入)
5. 一定要选上“删除 PB 自动生成的临时对象文件”和“允许覆盖只读文件”，如下：



### PB 程序加入到 SVN:

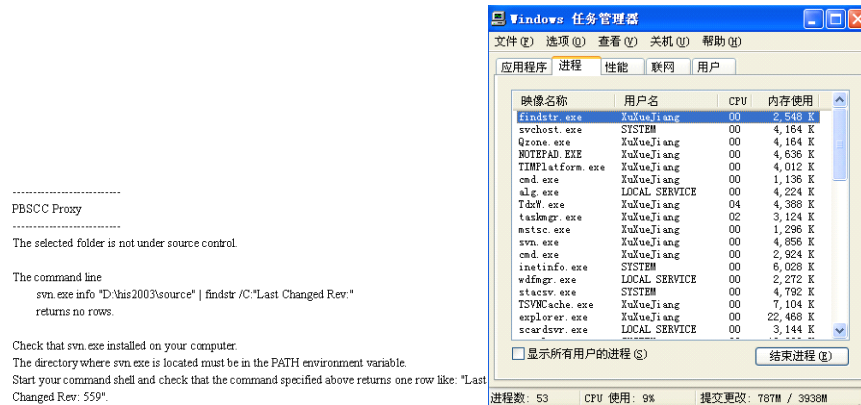
1. 可对整个 workspace,或一个应用，或一个 PBL，或一个对象。  
建议先把单个的应用(\*.pbt)文件加入，然后再按 PBL 库，一个一个的加入。



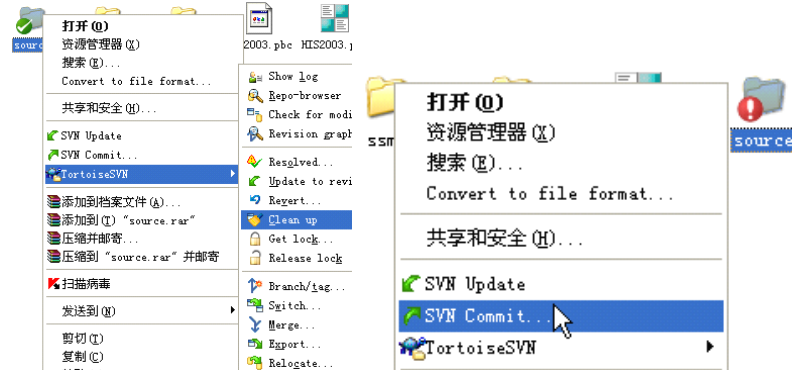


2. 如要加入时，出现错误，或死机，需要特殊处理。

对于 PB 的应用项目文件 (\*.pbt)，或整个 PBL 的对象，加入时可能会死机，出现此情况，要强行中断 PB，再调出“任务管理器”，中断 findstr.exe 和 svn.exe 两个进程，然后在操作系统的目录中，点中“代码根目录”（如 source），点《SVN \ Clean Up》，再《SVN 的 Commit》，用手工提交，而不用 PBscProxy，当再打开 PB 时，查看 Workspace 中的应用项目，其状态会变成“源码托管”（即一个绿点）。

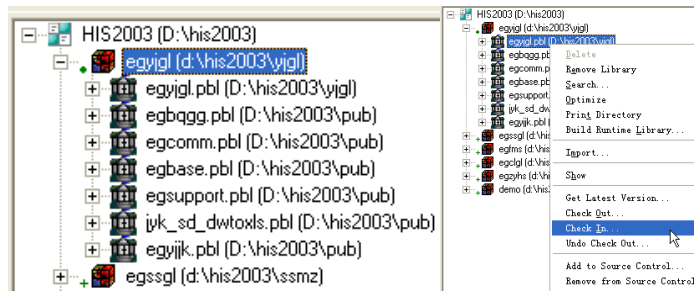


找到：findstr.exe 和 svn.exe 进程，删除之。



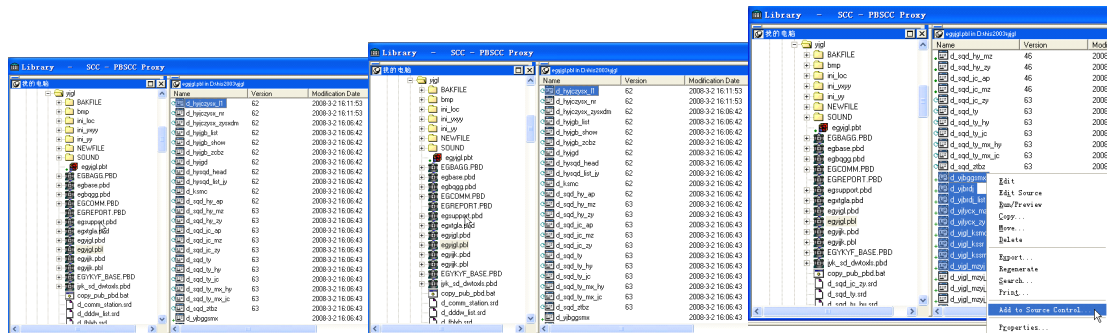
清除 SVN 的非法状态

### 3. 导入 PBL 库中的程序对象



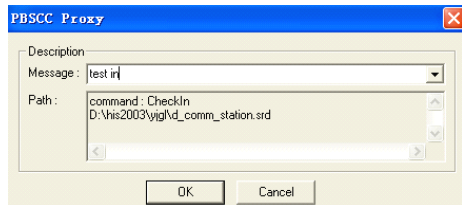
4. 由于在 Workspace 中加入可能会出错，所以最好是在 PB 的 Library 中进行操作，选中所有对象，批量操作。为保险起见，按单个 PBL，只选中一小部分，分批操作。

**注意：在进行大量的对象“add to source control”时，PB 会出错！**

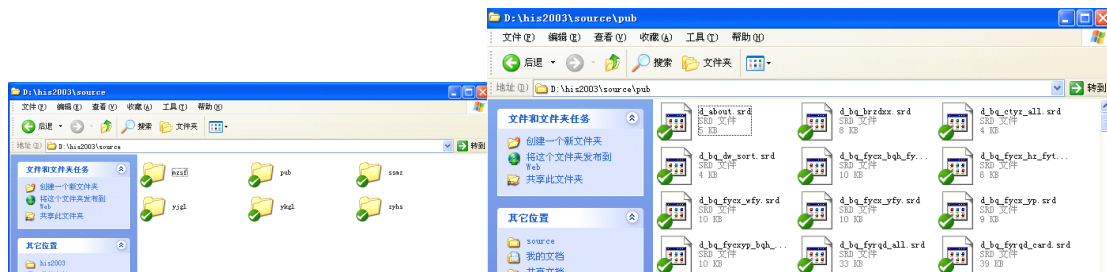


5. 在此操作时，会反复出现“PBSCC Proxy”对话框，此问题还未解决！

可从下拉框中选中随便选中上次输入的内容。但在以后的正式工作中，在“签入”时，需要完全填写此“修改日志”信息。

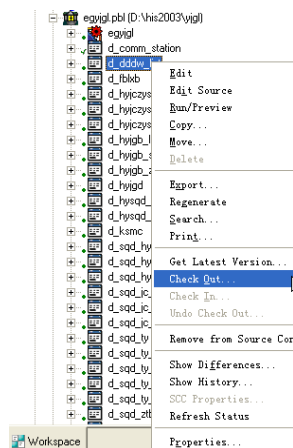


6. 导入后，会在 \his2003\source\ 生成很多 PB 程序文件，一个文件对应 PB 中的一个程序对象。每个文件的图标都是“源代码托管”状态（绿色的勾）。

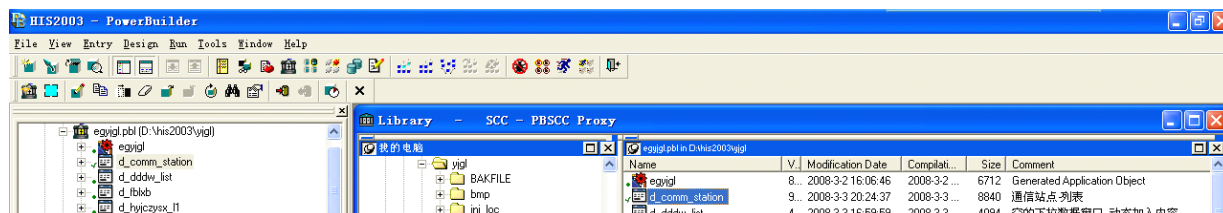


## PB 中 SVN 工作：

1. 操作与原来的 VSS 相近。PB 通过 PBSCC 代理，调用 SVN 的功能，自动进行相应处理。
2. 签出操作

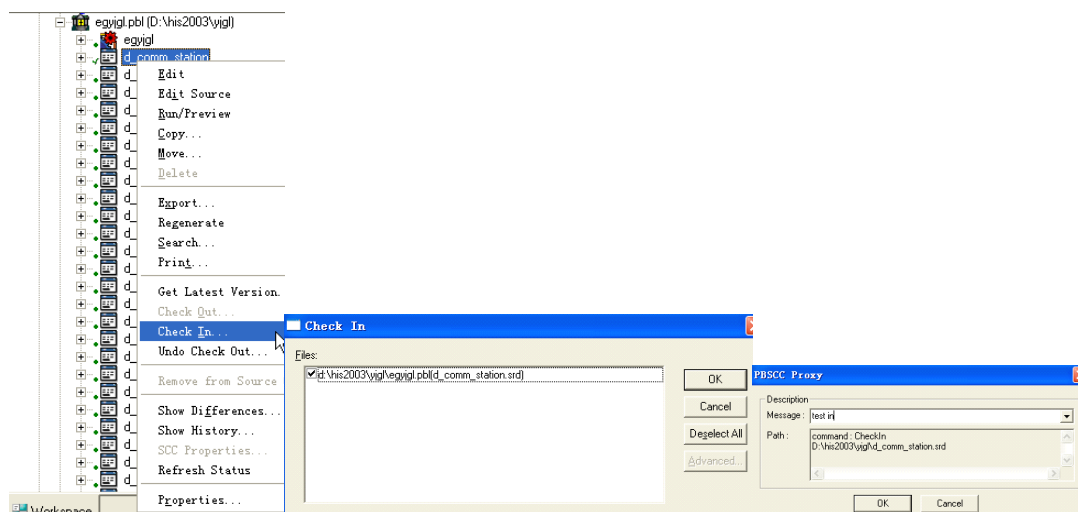


签出后，PB 中对象状态，与，浏览器中文件状态，比较：

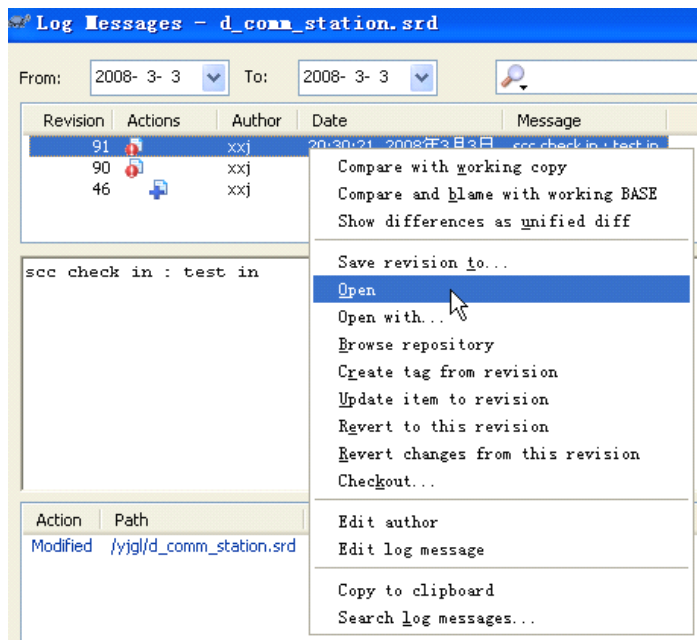
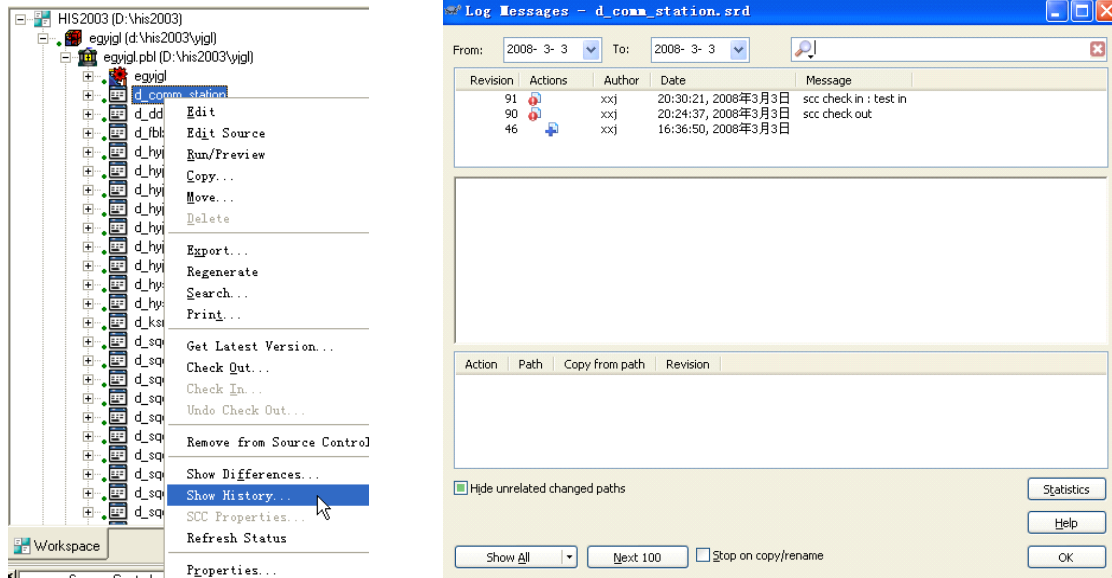


其实，文件的源码状态还是不变，为“在库”状态/

3. 签入操作

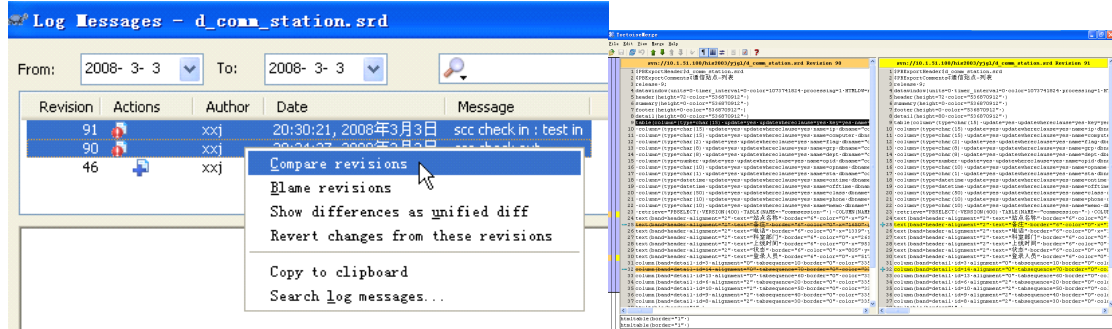


4. 查看历史变更“Show History”



查看文件内容，点《Open》

5. 文件版本比较，也在 PB 的“Show History”，而 PB 中的“Show Differences”将不再使用！

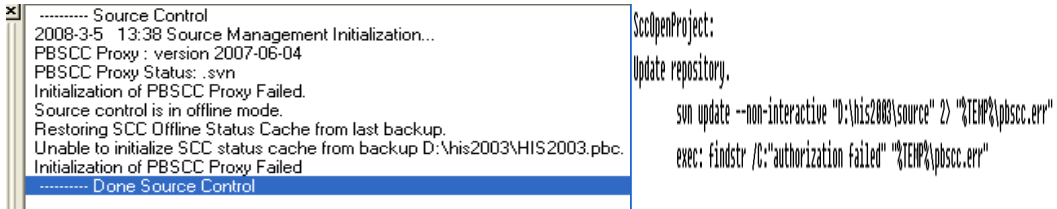


6. 第一次用 PB 连接 SVN，如果 PBL 库的对象很多（即 SVN 中文件很多），速度会很慢！请耐心等待。

## PB 问题处理

1. 如果密码改变，有可能打开 PB，连接 SVN 时，源码管理会没有反映（即菜单出不来）

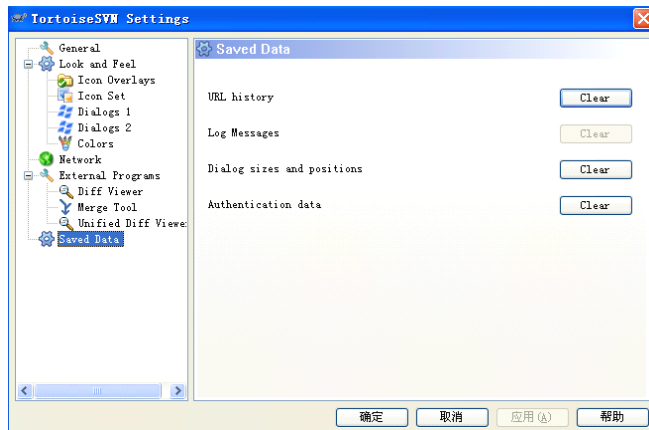
查看 PB 的提示信息，及 Pberr.txt 错误信息，发现如下：



在 DOS 下执行 Pberr 的出错的代理命令：

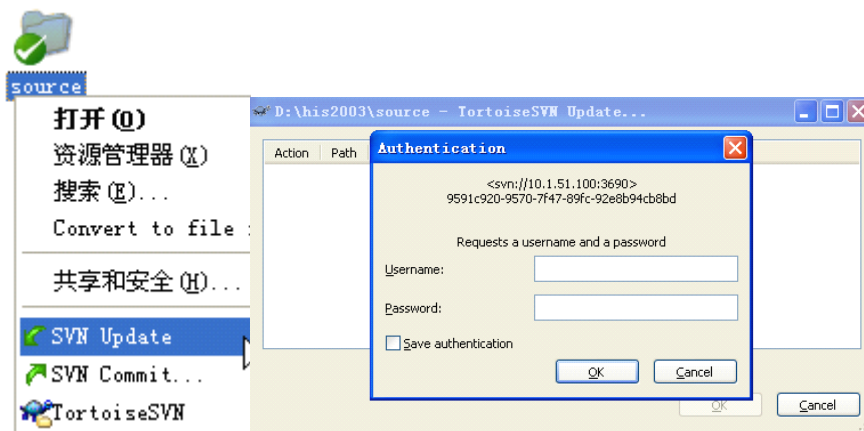
```
D:\>svn update --non-interactive "D:\his2003\source"
svn: 无法取得密码
```

```
D:\>svn update "D:\his2003\source"
认证领域: <svn://10.1.51.100:3690> 9591c920-9570-7f47-89fc-92e8b94cb8bd
"XuXueJiang" 的密码: ^C
```



先要去掉：Authentication data 《Clear》

再到“d:\his2003\source”目录下，点出《SVN \ Update》，重新输入用户密码（可保存）



2. 关于各目录的权限问题。如对某些文件只有读，没有写的权限；或；对某些目录下没有读写权限。都会出现提示“Access denied”。请与管理员联系。

```

----- Source Control
Checking out selected items . . .
PBSCC Proxy : svn: Commit failed (details follow):
PBSCC Proxy : svn: Access denied
Access to SCC Provider Failed during SccCheckout(). Try again.
SccCheckout for D:\his2003\yjig\nd_comm_station.srd appears to have failed.
Operation Completed
----- Done Source Control

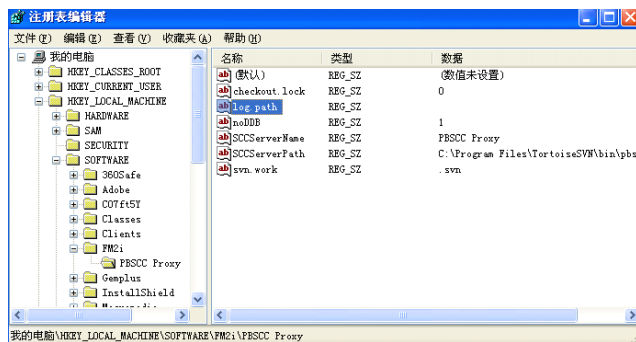
```

### 3. 有关错误日志文件的设置和查看

#### a) 如何设置 PB 错误日志

运行 Regedit.exe 注册表管理程序,找到:

[HKEY\_LOCAL\_MACHINE\SOFTWARE\FM2i\PBSCC Proxy] "log.path"



例如:设置 log.path = D:\source\PBerr.txt, 即可把错误日志保存在此目录。

此值默认为空,表示不保存错误日志。

#### b) 保存错误日志,会影响速度和性能,在找出问题后,请关闭日志。

### 4. End second

## SVN 主要功能介绍:

#### 1. 目录归档

在“源代码管理目录中”(注:非 PB 工作根目录),建立目录,再加入 SVN,再提交。

#### 2. 文件归档

新建文件后(如:word、excel、文本、图片...),或从其他地方拷贝到“源代码管理目录”或其下的子目录(注:其下的子目录也必有归档),文档也要先加入 SVN,再提交。

#### 3. 刷新目录或文件。点《SVN Update》从 SVN 取得最新版本。

#### 4. 删除文件或目录。在目标上,点出删除菜单后《SVN \ Delete》,再提交《SVN \ Commit》

#### 5. 查看文件或目录的日志。在目标上,点出《SVN \ Show log》

#### 6. 重命名文件。可直接修改。或在文件上点出《SVN \ Rename...》

#### 7. 远程访问 SVN。使用 <http://主机地址:端口号/svn/项目名称>,输入用户密码之后方可访问。

(注:公司内网访问 <http://10.1.51.100:90/svn/项目名称/>

公司外网访问 <http://60.190.25.222:90/svn/项目名称/>

用户密码 使用原来服务器上,操作系统的帐户)

#### 8. 更新出错的处理。

如果你更新到指定版本,再提交是将出现 **out of date** 错误!如果你想撤销修改并从以前的版本开始工作,你必须使用从“显示记录”对话框中选择复原自此版本以来的



变更 **Revert changes from this revision** 命令或使用快捷菜单中的 **合并 Merge...** 菜单命令。

**更新至版本 Update to Revision...** 命令允许复原至低版本。比如你的工作目录版本是 100, 你想复原至版本 50-直接填写在复原对话框中填写复原至版本 50。通常, 不复原单个文件, 确实需要, 从记录对话框 (Log) 选择储存版本至 **Save revision to...** 命令。

## 9. 解决冲突 Resolving Conflicts

有时你从档案库更新文件会有冲突。冲突产生于两人都修改文件的某一部分。解决冲突只能靠人而不是机器。当产生冲突时, 你应该打开冲突的文件, 查找以<<<<<<<<开始的行。冲突部分被标记:

```
<<<<<<<< filename
    your changes
=====
    code merged from repository
>>>>>>>> revision
```

Subversion 为每个冲突文件产生三个附加文件:

filename.ext.mine

更新前的本地文件。

filename.ext.rOLDREV

你作改动的基础版本。

filename.ext.rNEWREV

更新时从档案库得到的最新版本。

使用外部合并工具或使用快捷菜单的编辑冲突 **Edit Conflict** 命令来解决冲突。然后从快捷菜单中执行已解决 **Resolved** 命令, 将改动送交到档案库。请注意, 解决命令并不解决冲突, 而仅仅是删除 filename.ext.mine and filename.ext.r\* 文件并允许你送交。