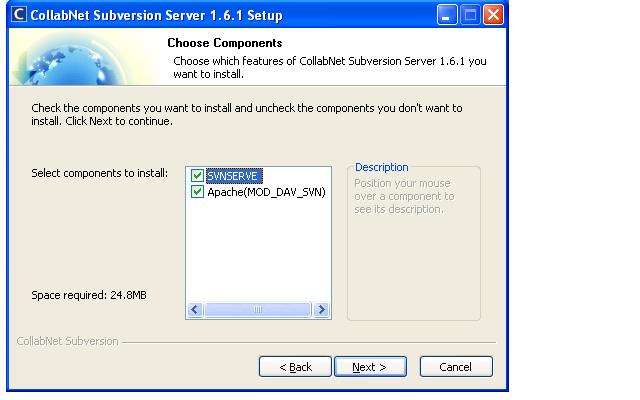
SVN日常操作指南

# 安装

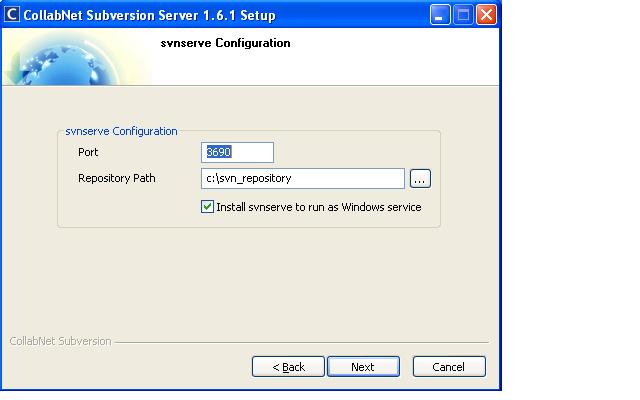
## Server

### Install

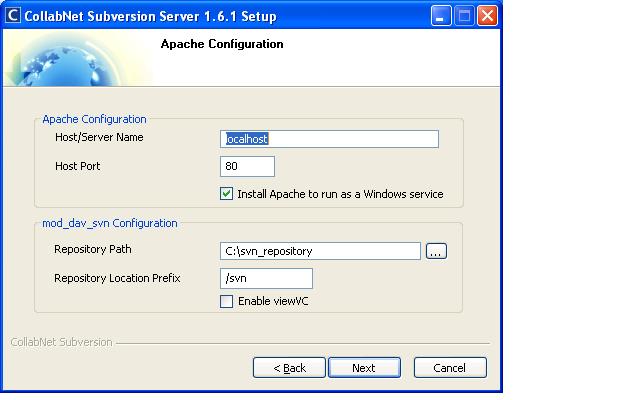
Run CollabNetSubversion-server-1.6.1-2.win32.exe, Next, Next, Select SVNSERVER and Apache(MOD\_DAV\_SVN) as the following picture.



Click Next then set the port 3690 as default ,and set Repository path to whatever you want such as c:\svn\_repository, select “Install SVN to run as windows service” as the following picture



Click Next, set the Apache server name localhost and port to 80 or other, set mod\_dav\_svn configuration as default, as the following picture.



Click Next and install to a path such as “C:\Program Files\CollabNet Subversion Server”, then restart the computer.

### Configuration

Lunch Dos prompt go to c:\SVN, and execute the command “svnadmin create --fs-type fsfs 911 to create a repository named 911.”

Go to httpd\conf where your server installed, open httpd.conf with notepad.Uncomment (remove the '#' mark) the following lines:

#LoadModule dav\_fs\_module modules/mod\_dav\_fs.so

#LoadModule dav\_module modules/mod\_dav.so

Add the following two lines to the end of the LoadModule section.

LoadModule dav\_svn\_module modules/mod\_dav\_svn.so

LoadModule authz\_svn\_module modules/mod\_authz\_svn.so

Go to the hpptd\bin folder and execute command “htpasswd –cm ..\conf\svnusers Sidney” (-c means create a file, so do no need it if the file exist; -m means use md5 to encrypt the user’s password). This command creates a file and adds a user named “Sidney” in the file.

4).Go to httpd\conf where your server installed, open httpd.conf with notepad, then set as following:

<Location /svn>

DAV svn

SVNListParentPath on

SVNParentPath C:\SVN

#SVNIndexXSLT "/svnindex.xsl"

AuthType Basic

AuthName "Subversion repositories"

AuthUserFile conf\svnusers

#AuthzSVNAccessFile conf\svnaccessfile

Require valid-user

</Location>

This configures Apache so that all your Subversion repositories are physically located below C:\SVN. The repositories are served to the outside world from the URL: http://MyServer/svn/. Access is restricted to known users/passwords listed in the svnusers file.

Following is a short explanation of what you just entered:

|  |  |
| --- | --- |
| Setting | Explanation |
| <Location /svn> | means that the Subversion repositories are available from the URL <http://MyServer/svn/> |
| DAV svn | Tells Apache which module will be responsible to serve that URL - in this case the Subversion module. |
| SVNListParentPath on | For Subversion version 1.3 and higher, this directive enables listing all the available repositories under SVNParentPath. |
| SVNParentPath C:\SVN | tells Subversion to look for repositories below C:\SVN |
| SVNIndexXSLT "/svnindex.xsl" | Used to make the browsing with a web browser prettier. |
| AuthType Basic | Is to activate basic authentication, i.e. Username/password |
| AuthName "Subversion repositories" | is used as an information whenever an authentication dialog pops up to tell the user what the authentication is for |
| AuthUserFile conf\svnusers | specifies which password file to use for authentication |
| AuthzSVNAccessFile conf\svnaccessfile | Location of the Access file for paths inside a Subversion repository |
| Require valid-user | specifies that only users who entered a correct username/password are allowed to access the URL |

5) The AuthzSVNAccessFile will be like the following:

[groups]

admin = john, kate

devteam1 = john, rachel, sally

devteam2 = kate, peter, mark

docs = bob, jane, mike

training = zak

# Default access rule for ALL repositories

# Everyone can read, admins can write, Dan German is excluded.

[/]

\* = r

@admin = rw

Dangerman=

# Allow developers complete access to their project repos

[proj1:/]

@devteam1 = rw

[proj2:/]

@devteam2 = rw

[bigproj: /]

@devteam1 = rw

@devteam2 = rw

trevor = rw

# Give the doc people write access to all the docs folders

[/trunk/doc]

@docs = rw

# Give trainees write access in the training repository only

[TrainingRepos: /]

@training = rw

5) Close the hpptd.conf file and restart Apache server, open ie Explorer type <http://localhost/svn/911> to have a test.

## Client

### TortoiseSVN

Run TortoiseSVN-1.6.1.16129-win32-svn-1.6.1.msi to complete installation.

TortoiseSVN integrates seamlessly into the Windows shell (i.e. the explorer). This means you can keep working with the tools you're already familiar with. And you do not have to change into a different application each time you need functions of the version control! All Subversion commands are available from the explorer context menu. TortoiseSVN adds its own submenu there. But his tool use lots of your computer resource even you do not use version control, and the context menu it added is confusing.

### RapidSVN

Run RapidSVN-0.9.8.exe to complete installation.

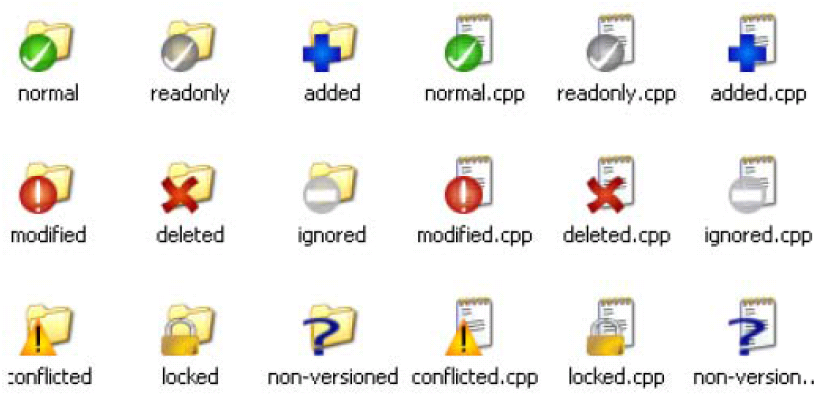
RapidSVN has a good UI and has Linux version, but it do not have some capabilities such as diff, merge, so the best way is to combine other tools with RapidSVN. Run RapidSVN, go to view-> preference, you can add your own editor, diff tool and merge tool with TortoiseSVN’s TortoiseUDiff.exe ,TortoiseMerge.exe, or winMerge.

# TortoiseSVN使用说明

## 基本特点介绍

### 图标重载

TortoiseSVN安装之后，重载了原来的图标。根据文件的Subversion 的状态不同，重载的图标也不同：





一个新check out 的复本使用绿色的对勾做重载。表示Subversion状态正常。



一个文件经过编辑以后，状态变成了已修改。图标变成了红色感叹号。



如果在提交的过程中出现了冲突，图标变成黄色感叹号。



如果给一个文件设置了svn:needs-lock属性,Subversion会让此文件只读，直到你获得了文件锁。只读文件用此重标来表示在你修改这个文件之前要先得到一个锁。



如果你拥有了一个文件的锁，并且Subversion的状态是正常，这个图标就提醒你如果不用此文件的话应该释放锁，允许别人提交对此文件的修改。



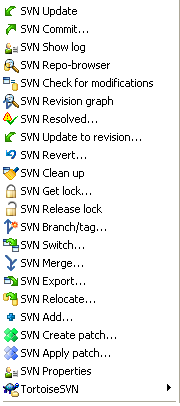
此图标表示当前文件夹下的某些文件或文件夹已经谋划要从版本控制中删除，或是该文件夹下某个受控的文件丢失了。



此图标表示此文件或文件夹已经计划要加入版本控制。

### 右键菜单

TortoiseSVN有着非常强的右键菜单，基本上所有的操作都可以在右键菜单中实现。它的有些菜单项是一直存在的，有的菜单项是要在满足一定的条件下都会出现。



### F5刷新键

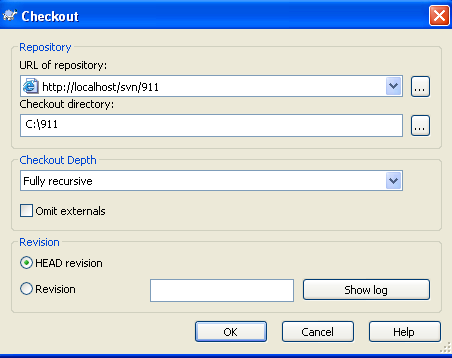
F5刷新当前视图。在资源管理器中可以刷新工作复本中图标的重载、在提交对话框中可以刷新哪些更新是需要提交的、在版本日志对话中可以可是重新联系版本库以检查更多的最近修改情况。

## 创建代码库并添加初始文件

在服务器上建立好Repository (库)以后可以在本地用TortoiseSVN check out， 然后在相应文件夹下添加新文件再用右键－>TortoiseSVN->Add，再Commit就可把代码或文件添加到SVN服务器。

## 创建本地视图

建立空文件夹（比如c:\911），右键TortoiseSVN->check out 这样就创建了本地视图，同时会更新代码到本地视图。



## .SVN文件夹

.svn文件夹保存了Subversion版本控制信息，不要对此文件夹做修改。如果你想得到一个没有subversion信息的工作拷贝请执行TortoiseSVN->Export命令。

## 删除视图

到本地资源管理器中删除该目录视图就可以。

## 修改代码

修改代码完成后用TotoiseSVN->Commit 提交到服务器。另外，在每次打算Commit之前最好先update下，这样可以避免很多不必要的冲突。

## 增加文件或文件夹

增加文件或文件夹后用TortoiseSVN->Add然后TortoiseSVN->Commit提交到服务器。

## 删除文件或文件夹

删除文件可以在资源管理器中删除然后TortoiseSVN->Commit，也可以用TortoiseSVN->delete,然后TortoiseSVN->Commit提交到服务器。

## 获得服务器最新代码

使用TortoiseSVN->Update来获得最新的服务器代码。

## 文件夹改名

在本地对文件夹改名然后F5再TortoiseSVN->Commit提交到服务器或者直接使用TortoiseSVN->Rename, 再TortoiseSVN->Commit提交到服务器。

## 导出文件到工作目录

用TortoiseSVN-export可以将服务器上的文件导出到提定的工作目录。

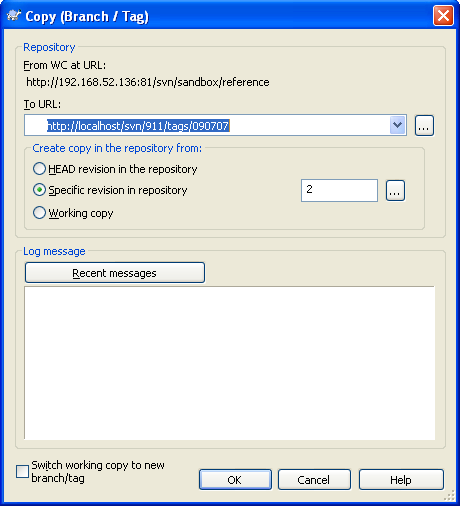
## 更改登陆用户

通常情况下我们会让windows来记住Subversion的用户信息，如果要改变登陆用户请如图位置删除svn.simple文件夹就可以了。



## 打标签

一般情况下每个project都会有trunk（主库）、branches（分支）和tags(标签)三个文件夹。打标签时选择TortoiseSVN->branches/tags,然后会如下对话框弹出：



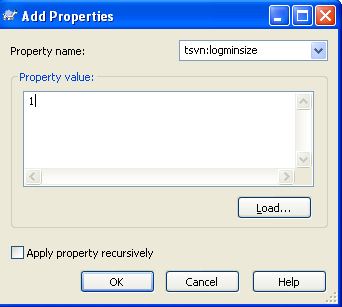
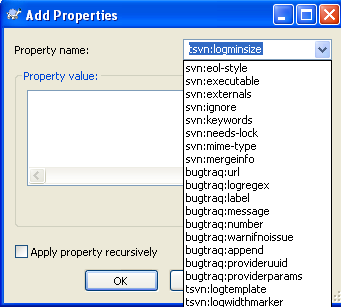
To URL指定标签名和位置，可以选择对最新的版本打标签也可以指定一个特定版本（Specific revision in repository）打标签。

## 建分支

建分支和打标签操作基本一致，不过To URL 要选择branches/xxx, Create copy in the repository from 要选择”Working copy”。分支会将源文件复制到目标文件夹一份。分支是可以修改的而标签不可以被修改。

## 设置文件/文件夹提交时必须填写日志

选择某个文件夹，右键->TortoiseSVN properties。会弹出如下图的对话框，设置大小就可以了。



## 把工作副本还原为基线版本

如果Update之前check in了代码，提交的时候会提示错误，然后Update的时候，TortoiseSVN更新代码的同时会保留刚才已有的修改，右键“diff”可以看到工作副本与服务器副本的差异，右键（revert）可以还原到基线状态。

实际上，修改视图中的代码之后，提交代码之前都可以用“比较差异”查看工作副本与服务器副本（上一次Update之后的视图文件）的差别，也都可以用revert还原

## 解决冲突（Merge）

有时当你从版本库中更新你的文件时，会有冲突。冲突出现的原因是两个开发人员修改了文件中相同的几行。由于Subversion不知道你的项目的具体情况，它把解决冲突的工作留给了开发人员。一旦出现冲突，你就应该打开有问题的文件，查找以字符串<<<<<<<开头的行。有冲突的区域用如下的方式标记:

<<<<<<< filename

你的修改

=======

来自版本库中的代码

>>>>>>> revision

对于每个冲突的文件Subversion在你的目录下放置了三个文件:

filename.ext.mine

这是你的文件，在你更新你的工作副本之前存在于你的的工作副本中——也就是说，没有冲突标志。这个文件除了你的最新修改外没有别的东西。

filename.ext.rOLDREV

这是在你更新你的工作副本之前的基础版本(BASE revision)文件。也就是说，它是在你做最后修改之前所检出的文件。

filename.ext.rNEWREV

这个文件是当你更新你的工作副本时，你的Subversion客户端从服务器接收到的。这个文件对应与版本库中的最新版本。

你可以通过TortoiseSVN->Edit conficts运行Merge工具/冲突编辑器，或者你可以使用任何别的编辑器手动解决冲突。你需要冲定哪些代码是需要的，做一些必要的修改然后保存。

然后，执行命令TortoiseSVN->Resolved并提交人的修改到版本库。需要注意的是resolved命令并不是真正的解决了冲突，它只是删除了filename.ext.mine和filename.ext.r\*两个文件，允许你提交修改。

如果你的二进制文件有冲突，Subversion不会试图合并文件。本地文件保持不变(完全是你最后修改时的样子)，但你会看到filename.ext.r\*文件。如果你要撤消你的修改，保留版本库中的版本，请使用还原(Revert)命令。如果你要保持你的版本覆盖版本库中的版本，使用已解决命令，然后提交你的版本。

你可以右击父文件夹，选择TortoiseSVN->resolved...，使用“resolved”命令来解决多个文件。这个操作会出现一个对话框，列出文件夹下所有有冲突的文件，你可以选择将哪些标记成已解决。

## 比较文件差异

### 本地修改

如果你想看到你的本地副本有哪些更加，只用在资源管理器中右键菜单下选TortoiseSVN → Diff。

### 与另外一个分支/标签之间的差异

如果你想查看主干程序(假如你在分支上开发)有哪些修改或者是某一分支(假如你在主干上开发)有哪些修改，你可以使用右键菜单。在你点击文件的同时按住**Shift**键，然后选择TortoiseSVN →diff with URL比较。在弹出的对话框中，将特别显示将与你本地版本做比较的版本的URL地址。

### 与历史版本的比较差异

如果你想查看某一特定版本与本地副本之间的差异，使用显示日志对话框，选择要比较的版本，然后选择在右键菜单中选与本地副本比较差异

### 两个历史版本的比较

如果你要查看任意已提交的两个历史版本之间的差异，在版本日志对话框中选择你要比较的两个版本(一般使用 Ctrl-更改)，然后在右键菜单中选比较版本差异

如果你在文件夹的版本日志中这样做，就会出现一个比较版本对话框，显示此文件夹的文件修改列表。

### 文件差异

如果你要查看两个不同文件之间的差异，你可以直接在资源管理器中选择这两个文件(一般使用 **Ctrl**-modifier)，然后右键菜单中选TortoiseSVN->diff。

## 撤消更改

如果你想要撤消一个文件自上次更新后的所有的变更，你需要选择该文件, 右击弹出快捷菜单，然后选择TortoiseSVN → Revert命令，将会弹出一个显示这个你已经变更并能恢复的文件。选择那些你想要恢复的然后按OK

## 清理

也许由于服务器问题，一个Subversion指令不能成功地完成，你的工作副本因此被滞留在一个不一致的状态。 那样的话，你需要在该目录上使用TortoiseSVN->clean up。在工作副本的根目录使用它是一个好主意。

## 谁修改了哪一行？

有时你不仅要知道哪一行做了修改，还要精确地知道谁修改了一个文件中的哪一行。这就是TortoiseSVN →Blame...命令，有时候也叫做 评注 命令派上用场的时候了。对一个文件中的每一行，这个命令列出了作者和该行修改时的版本

## 版本分支图

有时候，我们需要知道从哪开始有了分支和标签，同时想知道这条支路是单独的分支还是树型结构。如果需要你可以使用TortoiseSVN->revision graph...。这个版本历史分析图能够显示分支/标签从什么地方开始创建，以及什么时候删除。

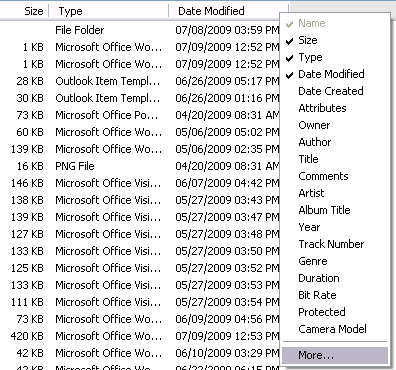
查看哪些文件做了修改

## 查看哪些文件做了修改（增删改）

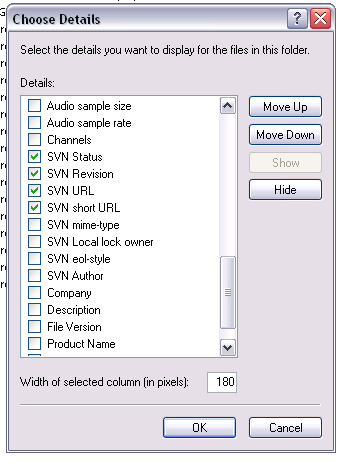
我们经常会去查看哪些文件我们做了修改，哪些文档别人做了修改要查这个状态运行TortoiseSVN->check for modifications此对话框会列出所有的修改。

## 在Winodws explorer中加入svn状态

到Window explorer 选择detail视图，在标题上选择more..



选择你加入的标题



# Subversion 资料站点

Svn的命令和功能很强大，在这里只列出了一些常用的功能，如果大家有兴趣可以查看TortoiseSVN使用手册或者去<http://www.subversion.org.cn/tsvndoc/>查看中文帮助文档。

另外TortoiseSVN还有其它好多版本比如有Eclipse插件版，有visual SVN（svn的visual studio插件）。有还有其它好多Subversion的客户端如RapidSVN等，大家也可以安装来试验一下。