<http://www.cnblogs.com/khler/archive/2008/10/16/1312249.html>

[Subversion for Windows 安装指南](http://www.cnblogs.com/khler/archive/2008/10/16/1312249.html)

*=========================================================*

*本文为转载资料，转载必须确保本文完整并完整保留原作者信息和本文链接*

*E-mail:* [*khler@163.com*](mailto:khler@163.com)

*QQ:     23381103*

*MSN:*[*pragmac@hotmail.com*](mailto:pragmac@hotmail.com)

*=========================================================*

最近因为工作需要需要配置一个svn服务器，经过几番试验并阅读了大量资料，终于最终配置完成。

网上资料实在良莠不齐，很容易浪费宝贵时间，所以我在此搜集整理了一些资料，并经过自己实践证明确实可用，也希望大家能快速而顺利的完成配置工作。

本人基于Apache 2.2.9 + Subversion1.5.3配置完成，需要注意的是：

1、一般可以通过浏览器敲入IP地址来验证Apache是否工作，如我的<http://192.168.1.102/>，则出现"It works!"页面则表示Apache已经能正常工作。但是你会发现同一个局域网的另一台机器则不能访问，这是因为防火墙阻止了访问，你可以把卡巴斯基（我自己用的卡巴，因人而异）以及Windows的防火墙关闭；

2、每次改变Apache的httpd.conf文件后必须重启Apache；

3、如果你的Subversion不是安装版，则必须在Windows的系统环境变量中添加Subversion的bin目录，如我的 Subversion放在了C:\Program Files\Subversion11.5.3\下，则我在系统变量的Path项的后面添加如下路径：C:\Program Files\Subversion11.5.3\bin。添加完了必须重启才能有效。

                                                        ---khler

                                                        2008.10.15

本文是基于蔡煥麟的文章整理的，整理者：airun、khler  
摘 要  
本文提供一个简易的安装指南，说明在 Windows 环境下安装 Subversion 服务器的步骤，以及 TortoiseSVN 客户端工具的安装步骤。文中提到的所有操作（2.6.2除外）本人都做过实践验证。  
目 录  
1 简介

1.1 阅读基本概念

1.2 环境与版本

2 安装和建立Subversion服务器

2.1 安装 Apache HTTP Server

2.2 安装 Subversion

2.3 设定 Subversion 文件库路径

2.3.1 SVNPath

2.3.2 SVNParentPath

2.4 建立文件库

2.5 引入项目

2.5.1 文件库与项目的配置方式

2.5.2 项目的目录结构

2.6 访问控制

2.6.1 基本(Basic)访问控制

2.6.2 SSPI认证访问控制（使用 Windows 网域帐户验证

2.7 提交控制

2.7.1 限制提交特定的文件类型

2.7.2 强制要求输入Log-Message

2.7.3 一个修改Log-Message的方法

3 安装客户端TortoiseSVN

3.1 安装 TortoiseSVN

4 结束语

5 参考资料

1 简介  
Subversion 是一个版本控制系统，它是根据 CVS（Concurrent Versions System）的功能为基础来设计，但是改进了一些 CVS 的缺点，例如：在 CVS 中移动文件目录很不方便；此外，它也增加了其他的功能，例如：不可分割的送交（如同资料库交易的概念，送交多个文件时，若有任何一个文件失败，则这次送交 的所有文件都不会进入文件库中），支援多种网络协议，一致的文件差异比对（不管什么文件类型，均使用二进位差异比对方式），等等。

由于目前手边查到的 Subversion 文件主要都是针对 Linux 用户来撰写的，所以这份文件特地针对 Windows 环境下安装 Subversion 的步骤来说明，希望透过这份文件，能够帮助你很快的把 Subversion 安装起来。如果熟悉了的话，一两个小时就可以搞定。

在安装过程中，会需要输入一些命令列的指令，我不会详细解释某些指令的用途和意义，因此你除了要熟悉 DOS 的基本指令，还应该随时查阅 Subversion 的电子书（有中文版，比如“svn宝典.pdf”等），以了解 Subverion 命令工具的使用方法。图形化介面虽然方便，但是熟悉命令工具的使用，才能让你得到完全的自由。

1.1 阅读基本概念  
如果你缺乏版本控制系统的基本概念，就算能够顺利安装好 Subversion，可能安装完成后就不知道下一步怎么做了。这里只简单的提一点必要的基础概念，你最终还是得阅读 Subversion 的官方文件。

1.2 环境与版本  
以下是我安装 Subversion 的环境与版本：  
• Windows 2000 Server with SP4  
• Apache HTTP Server: Apache\_2.0.63-win32-x86-no\_ssl.msi  
• SSPI：mod\_auth\_sspi-1.0.2-2.0.54.zip  
• Subversion: Setup-Subversion-1.5.1.en-us.msi  
• TortoiseSVN: TortoiseSVN-1.4.0.7501-win32-svn-1.4.0.msi

2 安装和建立Subversion服务器  
本节介绍安装 Subversion 所需的步骤，请准备一台稳定的机器，作为 Subversion 的服务器，硬盘空间要尽量大一些，以避免因磁盘空间不够而频繁更换服务器。

2.1 安装 Apache HTTP Server  
到 <http://httpd.apache.org/> 下载 Apache HTTP Server 2.0 版 for Windows 的安装程序，我下载的文件是 Apache\_2.0.63-win32-x86-no\_ssl.msi。  
下载之后直接安装，安装过程很简单，就不赘述了，但安装之前请先检查你的电脑是否有安装 IIS，由于 Apache 预设使用 80 port，会跟 IIS 的网站冲突，你必须把 IIS 的 Web 站台关闭，再安装 Apache HTTP Server。  
安装完成以后，启动Apache，开启浏览器，浏览网址 <http://127.0.0.1/> 看看有没有安装成功。如果页面显示“您能看见这个页面，说明Apache web server 已经安装成功。您可以在这个目录中增加内容，或者把这个页面替换掉。”等文字，则表示已经安装成功了。  
为方便描述，下文我们假设Apache HTTP Server 安装到了<Apache\_root>目录下。

2.2 安装 Subversion  
1. 到 <http://subversion.tigris.org/> 下载最新版的Subversion，可以下载.zip 或者打包好的自动安装程序，我下载的是 Setup-Subversion-1.5.1.en-us.msi。

2. 下载后直接安装，安装过程都是下一步，没什么特别的。安装完成后，连必要的环境变量都帮你设定好了（如果你下载的是 .zip 文件，就要自行设定）。为方便描述，下文我们假设Subversion 安装到了<Subversion\_root>目录下。

3. 把 <Subversion\_root>/bin/ 目录下的 mod\_dav\_svn.so 和 mod\_authz\_svn.so 复制到 <Apache\_root>/modules/ 目录下。

4. 把 <Subversion\_root>/bin/ 目录下的 libdb44.dll、libeay32.dll、以及 ssleay32.dll 复制到 <Apache\_root> 底下的 bin 或 modules 目录下。

5. 接着用文本编辑器打开 Apache HTTP Server 的<Apache\_root>/conf/httpd.conf，寻找一堆 LoadModule 指令，先找到以下两行：

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

把这两行最前面的 '#' 字元刪除，然后把下面这行加到这群 LoadModule 指令的后面：

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

6. 重新启动 Apache HTTP Server。

问题排除如果 Apache HTTP Server 无法启动，请依下列步骤检查：  
1. 检查 Subversion 的路径是否有在系统的 %PATH% 环境变量里面。  
2. 检查你加入 httpd.conf 里的项目是否正确，记住 mod\_dav\_svn.so 和 mod\_authz\_svn.so  
必须在其他 mod\_dav\*.so 模组之后载入。

2.3 设定 Subversion 文件库路径

现在要设定 URL 路径与文件库实际路径的对应关系。对应的方式有两种，分别是 SVNPath 与 SVNParentPath。

2.3.1 SVNPath

SVNPath 适合用来个别指定文件库的路径，语法是：

<Location /svn/repos\_name>  
DAV svn  
SVNPath /absolute/path/to/repository  
</Location>

其中 "/svn/repos\_name" 就是用户端存取特定文件库的 URI（Uniform Resource Indentifier），SVNPath 后面指定的路径则是文件库的绝对路径，假设我们的文件库实际存放的路径是 d:/svn/MyRepository，并且希望用户端使用 "<http://myserver/svn/myrepos>" 的 URL 来存取文件库，那么要加入 httpd.conf 的內容就是：

<Location /svn/myrepos>  
DAV svn  
SVNPath d:/svn/MyRepository  
</Location>

注意 Location 标签后面的 /svn/myrepos 的第一个斜线不可少！

2.3.2 SVNParentPath

如果你的文件库都集中放在某个目录之下，例如：d:/svn，那你就可以使用 SVNParentPath 的方式指定文件库的根路径，语法是：

<Location /svn>  
DAV svn  
SVNParentPath /absolute/path/to/repository  
</Location>

其中的 SVNParentPath 后面指定的路径是文件库的绝对路径，假设我们的文件库实际存放的路径是 d:/svn，那么要加入 httpd.conf 的内容就是：

<Location /svn>  
DAV svn  
SVNParentPath d:/svn  
</Location>

这表示可以让任何人都可以透过 <http://myserver/svn/><文件库名称> 的方式，存取位于 d:/svn 这个目录以下的所有文件库。也就是说，这个设定动作只需要一次，如果使用 SVNPath，那么你必须为各个文件库分别指定对应的路径。

为方便起见，我们这里用 SVNParentPath 来统一指定所有文件库的父层 URL 路径。将上述的设定加到 Apache HTTP Server 的 httpd.conf 文件尾部就行了。  
接下来就可以在 d:/svn 底下建立文件库了。

2.4 建立文件库

假设我们要把所有的文件库都放在 d:/svn 目录下，现在要建立一个测试用的文件库，名称叫做 repository，指令为：

md d:/svn  
svnadmin create d:/svn/repository

命令执行完后，检查看看 d:/svn/repository 目录底下产生了哪些目录和文件。

警告文件库绝对不可以在建立在任何远端的储存媒体上，例如：网路硬盘机。

这时候你已经建立了一个文件库，你可以先在本机用浏览器测试一下，网址输入 <http://127.0.0.1/svn/repository>或<http://localhost/svn/repository>，看看能不能看到文件库的内容，正常的话应该像下图一样。



如果以上的测试可以通过，应该就行了，假如你的局域网中安装svn的服务器IP地址是192.168.1.8，则在局域网中另外一台机器（客户端）上输入<http://192.168.1.8/svn/repository>，应该是可以看到同样画面的。如果不行，则要找一下你们网管了。

如果你想要测试一下能否从文件库取出整个工作副本（此操作并非必需），则可以进入目标目录后（假设为d:/svn\_temp）执行以下指令从文件库取出整个工作副本：

svn co <http://localhost/svn/repository> WholeRepos

该指令是要 svn.exe 执行 check out 动作（缩写 co），如果正确的话，应该会显示 "Checked out revision 0." 的信息，此时 d:/svn/repository 这个文件库底下的所有文件目录都已经取出，并且复制了一份到d:/svn\_temp/WholeRepos 目录下了。

问题排除如果显示的错误信息是：  
svn: PROPFIND request failed on '/svn/repository'  
svn: PROPFIND of '/svn/repository': 405 Method Not Allowed (<http://localhost/>)  
请检查 Apache HTTP Server 的 httpd.conf 文件里面的 <Location> 标记所定义的位置是否跟你指定的 URL 样式相同，注意一定要完全相同，以上面的例子而言，你的 httpd.conf 的最后面应该会有以下文字：  
<Location /svn>  
DAV svn  
SVNPath 指向文件库的绝对路径  
</Location>  
如果显示的错误信息是：  
svn: PROPFIND request failed on '/svn/repository'  
svn:  
Could not open the requested SVN filesystem  
那表示在 /svn 对应的实验目录（即 d:/svn）下找不到 reposiroty 这个目录。  
注：PROPFIND 是给 WebDAV 用的 HTTP method，用来从资源中取得属性。

到目前为止，可以确定文件库已经建立完成，接下来就可以引入项目了。

2.5 引入项目

不用急着把你现有的正式项目引入到文件库，假设我们现在有一个测试项目，项目目录位于d:/svn\_temp/projectA，底下有下列子目录（当然也可以拷贝些文件到这些子目录中）：

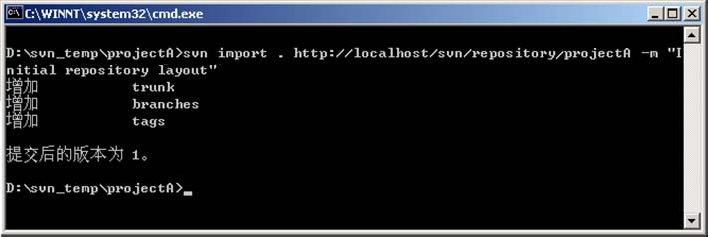
trunk  
branches  
tags

则进入d:/svn\_temp/projectA目录后执行以下指令就可以将测试项目projectA引入到文件库

d:/svn/repository/projectA中。  
svn import . <http://localhost/svn/repository/projectA> -m "Initial repository layout"

提示本文在执行 svn 命令时，都是使用 http 协议的方式，这样我们可以确知 Subversion 与 Apache HTTP Server 的设定无誤，其他人就可以透过 Internet 存取文件库。当然你也可以用其他的协议，例如：[http://www.cnblogs.com/khler/admin/file:///](http://www.cnblogs.com/khler/admin/file:/)，如果使用 file 协议，则该指令要变成：  
svn import . [http://www.cnblogs.com/khler/admin/file:///d:/svn/repository/projectA](http://www.cnblogs.com/khler/admin/file:/d:/svn/repository/projectA) -m "Initial repository layout"

命令执行无误的话，应会看到如下的画面：



这时候 projectA 这个项目已经引入文件库了（此时你会发现在d:/svn/repository/db/revs目录下多了一个叫1的文件），也就是说，其他使用者可以开 始存取这个文件库的项目取出文件和程序码了。你可以参考 Subversion 的官方手册中关于 svn.exe 这个客户端命令列工具的使用方法，多练习一下取出文件、加入文件、以及存入文件等指令。万一练习的过程中发生错误，或者文件库弄乱了，你可以把整个文件库 的目录砍掉，回到 2.4 节重新做一遍。

以下会进一步讨论文件库和项目目录结构的安排方式，如果你急着想试试看客户端如何存取 Subversion 文件库，可以先跳到2.6 节或第 3 节。

2.5.1 文件库与项目的配置方式

如果我们要在repository文件库中引入多个项目，比如再将d:/svn\_temp/projectB项目放到repository/projectB目录下，则进入projectB目录，用下面的命令引入projectB：

svn import . <http://localhost/svn/repository/projectB> -m "Initial repository layout"

引入projectB后，repository文件库的目录结构就会变成这样：

/svn/repository/  
+-- projectA/  
+-- projectB/

也就是说 repository 这个文件库里面包含了两个项目。

如果你希望为每个项目建立一个文件库，那么在 2.4 节中建立文件库的指令就变成：

md d:/svn  
svnadmin create d:/svn/projectA  
svnadmin create d:/svn/projectB

这样就变成有两个文件库了，文件库名称分别是 ProjectA 与 ProjectB。

提示如果项目之间有共享的文件，建议把这些相关的项目放进同一个文件库；如果项目之间彼此毫无关系，那就采用一个文件库放一个项目的方式，这种方式等于项目就是文件库。  
第一种方式有个比较奇怪的「功能」你应该要知道，就是一个项目的 check in 动作，也会令其他项目的文件的修订版号增加 ，如果这不是你想要的，请选择第二种方式，即一个文件库只存放一个项目。

2.5.2 项目的目录结构

这里补充说明一下 ProjectA 的目录结构。在 ProjectA 项目的根目录下建立的 trunk、branches、和 tags 这三个目录是有特别意义的，它们的作用分别是：

• trunk 目录用来存放目前项目正在进行开发的程序文件和文档（又称为主线，即 mainline）；  
• branches 用来存放主线的各个仍在发展中的分支；  
• tags 则用来存放已经不再变动的分支，也就是其中的文件不会再修改了。

这是 Subverion 官方手册建议的目录结构安排方式，你可以自己决定要不要用这种配置方式，详细说明请参考官方手册的第五章，子标题为 "Choosing a Repository Layout"。

提示  
目录名称建议尽量不要用中文名称，这样在使用命令列时比较方便，也比较不会有问题。

2.6 访问控制

2.6.1 基本(Basic)访问控制

照着前面的步骤做，你会发现存取文件库时都不用输入帐号密码，这是因为我们之前的设定没有启用身分验证的功能。但是我们通常不希望所有人都能任意存取你的文件库，免得重要资料外泄，或者资料被破坏，因此了解如何加入身分验证也是必要的。

Sebversion 提供了多种验证使用者身份的方式，首先介绍Basic访问控制，按下列步骤进行：

1. 修改 httpd.conf 的 <Location> ，设定如下：

<Location /svn>  
DAV svn  
SVNParentPath d:/svn  
AuthName "Subversion repository"  
AuthzSVNAccessFile "d:/svn/accessfile.txt"  
Require valid-user  
Satisfy Any  
# The following is Basic authentication setting  
AuthType Basic  
AuthUserFile "d:/svn/passwd.txt"  
</Location>

其中AuthzSVNAccessFile指定了文件库访问政策文件，Require valid-use表示只有经过认证的用户才可以访问， Satisfy Any表示允许匿名访问（要与后面提到的\*参数配套使用），AuthUserFile用来指示使用htpasswd创建的密码文件的位置。

如果你想用不同的访问政策文件和密码文件控制不同的文件库，那么可以在httpd.conf中写多个<Location></Location>模块，并用SVNPath指定各文件库的绝对路径（参考2.3.2）。

2. 用Apache的htpasswd指令生成用户名和密码：

htpasswd -cm d:/svn/passwd.txt username1  
htpasswd -m d:/svn/passwd.txt username2  
htpasswd的参数-c是创建文件的意思，故只在首次使用，仅使用-m参数时也可用来修改密码。

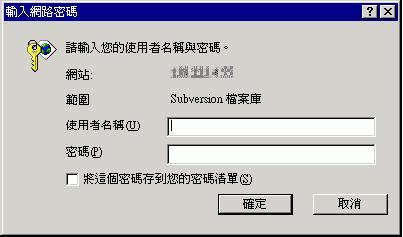
3. 配置文件库访问政策文件（假设repository文件库下已经建立了projectA和projectB），如：

[repository:/]  
username1 = rw  
[repository:/projectA]  
username2 = r  
[repository:/projectA/trunk]  
username1 =  
username2 = rw  
[repository:/projectB]  
\* = r  
[repository:/projectB/trunk]  
username1 =

小节名称中冒号前面的文件库名称不要轻易省略，省略就代表是针对所有文件库，这样是很为危险的，\*号代表所有用户，等于号后面为空表示没有权 限，mod\_authz\_svn模块首先会找到匹配自己的目录，然后是父目录，父目录的父目录，依次下去，并会以最近找到的描述为准。另外，你还可以定义 一组用户，然后按组进行权限配置，比如：

[groups]  
department\_1 = username1, username2  
department\_2 = username3, username4, username5  
[repository:/projectA]  
@department\_1 = r  
@department\_2 = rw

4. 重新启动 Apache（注意，对httpd.conf的修改必须重启Apache后才会生效），开启浏览器，输入网址 <http://127.0.0.1/svn/repository> 看看，你预期应该会看到如下的验证画面：



5. 如果要让用户能远程修改自己的Subversion密码，也很简单，有人已经把你做了相关工作了，可访问<http://bbs.iusesvn.com/thread-20-1-1.html>站 点下载Apache20Passwd.rar，将其中的passwd.cgi及passwd.cgi.ini复制 到<Apache\_root>的cgi-bin子目录中，然后修改配置文件passwd.cgi.ini中的auth\_user\_file路 径，使得其指向实际的AuthUserFile的全路径（比如“d:/svn/passwd.txt”），这时用户就可以访问<http://localhost/cgi-bin/passwd.cgi>修改subversion密码了。

2.6.2 SSPI认证访问控制（使用 Windows 网域帐户验证）

这种方式很适合用在开发团队成员都在区域网路内的情況，按下列步骤进行：

1. 取得 SSPI 模组，下载网址为（Apache与SSPI之间是有版本对应关系的，要特别注意）： <http://www.gknw.net/development/apache/httpd-2.0/win32/modules/mod_auth_sspi-1.0.2-2.0.54.zip>

2. 把 mod\_auth\_sspi.zip 里面bin目录下的 mod\_auth\_sspi.so 解压縮到 <Apache\_root>\modules 目录下。

3. 把下面这行加入到 Apache 的 httpd.conf 里面：

LoadModule sspi\_auth\_module modules/mod\_auth\_sspi.so

注意上面加入的这行一定要放在下面这行的前面：

LoadModule auth\_module modules/mod\_auth.so

4. 修改 httpd.conf 的 <Location> ，设定如下：

<Location /svn>  
DAV svn  
SVNParentPath d:/svn  
AuthName "Subversion repository"  
AuthzSVNAccessFile "d:/svn/accessfile.txt"  
Require valid-user  
Satisfy Any  
# The following is SSPI authentication setting  
AuthType SSPI  
SSPIAuth On  
SSPIAuthoritative On  
SSPIDomain <domaincontroller>  
SSPIOfferBasic On  
</Location>

其中<domaincontroller> 就是你的 Windows 网域控制器的电脑名称（例如：WIN2KDC），注意两边的括号 <> 不用保留。如果你的环境没有网域控制器，就维持原来的 <domaincontroller> 就行了。在我的环境下，我发现即使有网域控制器，但是这里不去设定它，还是能够正常的验证使用者身份。另外注意，这种方式不再需要用 AuthUserFile来指示密码文件了。文件库访问政策文件也要做相应修改：

[repository:/]  
domain\username1 = rw  
[repository:/projectA]  
domain\username2 = r  
[repository:/projectA/trunk]  
domain\username1 =  
domain\username2 = rw  
[repository:/projectB]  
\* = r  
[repository:/projectB/trunk]  
domain\username1 =

5. 重新启动 Apache，开启浏览器，输入网址 <http://127.0.0.1/svn/repository> ，这时也应该看到验证画面。但是如果没有出现这个画面，而是直接显示文件库内容，怎么回事？

因为我们现在是使用 Windows 帐户验证，你目前已经登入这台机器了，而你要存取的也是本机的资源，换句话说，你的身分已经被验证过了，所以就不会再要求你输入帐号跟密码，这是采用 SSPI 网域验证的好处。

那么，如果你的同事 John 的电脑有加入网域，但是他平时都是登入本机，而非登入网域，在存取文件库时会不会要求输入帐号密码？答案是如果 John 登入他本机的帐号和密码跟他在网域使用者的帐号密码完全一样的话，就无需再输入密码；相反的，如果登入本机的使用者帐号和密码与网域使用者帐号密码不同， 第一次存取时就必须输入密码。

你可以在别台机器上，用一个网域里没有的使用者帐号去存取 Subverion 文件库，如果正确的话，应该就会出现要求输入帐号密码的窗口。

以上是些基本的设定，如果你希望做进一步的设定，或者要加入 SSL 加密机制，建议您参考 [2] 的第三章。

提示启用身分验证之后，你会发现用命令列工具 svn.exe 存取文件库时，如果是用 http:// 协议，有些子命令（subcommand）执行时会出现 "authorization failed" 的错误，这时候你可以在 svn 命令中加入 --username 和 --password 来提供使用者名称和密码，例如：  
svn co <http://myserver/svn/> --username michael --password guesswhat  
或者你也可以改用 [http://www.cnblogs.com/khler/admin/file:///](http://www.cnblogs.com/khler/admin/file:/) 协议。

2.7 提交控制

2.7.1 限制提交特定的文件类型

客户端提交修改点时，若要限制提交某些特定类型的文件，则可以在相应的版本库目录的hooks目录下创建pre-commit.bat文件，并将下面的指令拷贝到pre-commit.bat中。

@echo off  
set SVN\_BINDIR="c:\Program Files\Subversion\bin"  
setlocal  
set REPOS=%1  
set TXN=%2  
rem check that the file type is permitted or not  
%SVN\_BINDIR%\svnlook changed "%REPOS%" -t "%TXN%" | findstr "\.obj$ \.o$ \.bak$ \.scc$ \.ncb \/复件." > nul  
if %errorlevel% equ 0 goto FILETYPEERROR  
exit 0  
:FILETYPEERROR  
echo 提交失败，不允许提交.obj、.o、.bak、.scc、.ncb文件或以"复件"开头的文件。>&2  
exit 1

其中SVN\_BINDIR后面的路径要修改为你实际的Subversion安装路径；REPOS代表svn调用pre-commit时的第一个参数 REPOS-PATH (the path to this repository)；TXN代表第二个参数TXN-NAME (the name of the transaction about to be committed)；findstr "\.obj$ \.o$ \.bak$ \.scc$ \.ncb \/复件."表示查找以.obj、.o、.bak、.scc、.ncb为结尾或含有"/复件"的行。当然，你也可以通过类似的方法让用户只能提交某些特定 类型的文件。

2.7.2 强制要求输入Log-Message

客户端提交修改点时，若需强制要求提交者在Message栏中输入信息，则可以使用svnlook log命令实现，在pre-commit.bat文件的对应位置加入下面两段指令即可：

rem check that log-message contains at least 10 characters  
%SVN\_BINDIR%\svnlook log "%REPOS%" -t "%TXN%" | findstr ".........." > nul  
if %errorlevel% gtr 0 goto NOMESSAGE  
:NOMESSAGE  
echo 提交失败，请在Message输入框中输入提交说明，至少10个字符。>&2  
exit 1

其中要求至少输入的字符个数由findstr后面的.的个数决定。

2.7.3 一个修改Log-Message的方法

当客户端提交修改点后，其Log-Message信息通常也就不能再被修改了，但是如果Log-Message信息中出现关键错误时，可用下面的方法挽救：

1. 在服务器上建立一个临时.txt文件，比如newlog.txt，然后将新的Log-Message写入其中。

2. 打开命令行模式并进入到newlog.txt所在的目录，执行如下命令：

svnadmin setlog d:/svn/repository –r N --bypass-hooks newlog.txt

其中N代表需要修改Log-Message的Revision number，--bypass-hooks代表不通过 pre-revprop-change hook的控制。

修改Log-Message是一个不好的做法，若不是迫不得已，建议不要用。

3 安装客户端TortoiseSVN

现在你已经有一个可以运作的 Subversion 服务器，可以试着在其他电脑上存取文件库了。如果你习惯使用命令列工具，那就只要在客户端电脑上安装 Subversion 就行了，存取文件库都是透过命令列工具（主要是 svn.exe）。这里要介绍的是一个专门为 Windows 操作系统设计的 Subversion 客户端：TortoiseSVN（以下简称 TSVN）。

3.1 安装 TortoiseSVN

1. 到 <http://tortoisesvn.tigris.org/> 下载最新的安装程序，比如TortoiseSVN-1.4.0.7501-win32-svn-1.4.0.msi。安装完成后会要求你重新开机。

2. 到 <http://tortoisesvn.tigris.org/> 下载简体中文的语言包（language pack），请注意语言包的版本应该要跟你安装的 TSVN 版本相同，否则最好不要安装。

语言包装完之后，用资源管理器在 Windows 桌面上或任何一个文件夹点一下滑鼠右键，选择 TortoiseSVN -> Settings，在 "Main" 页夹中更改 Language 设定为「中文(简体)」，再按「确定」钮即可。

3. 如果你是透过 proxy server 存取 Internet，请在 TSVN 的设定窗口中，切到「网络」页面，然后输入你的 proxy server 相关信息，否则你将无法存取位于 Internet 上的文件库。

安装完成之后，在任何目录名称上点一下滑鼠右键都可以看到 TSVN 的功能选项，这也是 TSVN 方便的地方，它不用跟开发工具整合，而是跟操作系统整合在一起，这样不管你用什么开发工具，都可以轻松的使用 TSVN 来存取文件库。如果你希望在TSVN中使用Beyond Compare工具比较文件，则在客户端安装Beyond Compare后，在任意目录点鼠标右键，进入Tortoise-->Settings-->External Programs-->Diff Viewer/Merge Tool/Unified Diff Viewer，在External Programs中指定BC2.exe的路径即可。

接下来你可以用 TSVN 练习一下存取之前建立好的文件库，试着把你现有的项目引入文件库中，并且在客户端使用 TSVN 执行取出、存入、更新等动作。

TSVN 虽然是客户端工具，不过它也提供了建立文件库、以及引入、引出等功能，因此安装在服务器端也挺方便的。

这里顺便提几个客户端提交修改点时的注意事项：

􀁺 要保证本地编译通过后方可提交，否则会导致其他人取下新版本后编译不过。  
􀁺 要一次性全部提交，即在项目的根目录下按右键选“SVN Commit”，若必须部分提交，则需要保证部分提交后其他人取下新版本后能编译通过。  
􀁺 提交时一定要在Message栏写上有效的说明信息，以便后续进行Patch提取，版本比较等。如果是修改了一个BUG然后提交，则需要将BUG的ID号和简单描述写到Message栏中。  
􀁺 不要提交无需进行版本控制的文件，比如编译产生的目标文件，备份文件，可执行文件，以及临时文件等。  
􀁺 如果提交失败，则很可能是文件冲突导致，即提交者同其他人同时修改了某文件，且其他人的修改点已经提交到了服务器，这时需要在项目的根目录下按右键选“SVN Update”，然后Merge修改点。

4 结束语

按照本文说明的安装步骤，希望能让你顺利在 Windows 环境下把 Subversion 安装起来。但是安装成功以后，真正的工作要才开始，如果你没有花点时间阅读 Subversion 的相关文件，在使用版本控制系统的过程中，一定会碰到许多问题。

在正式将你的项目加入 Subversion 文件库之前，建议您多考虑一下：

1. 文件库的配置方式。究竟要为每一个项目建立一个文件库，还是把多个项目放进同一个文件库里？  
2. 项目目录的结构。你要依照官方手册的方式在项目的根目录下建立 trunk、branches和 tags 吗？  
3. 哪些东西要放进文件库里？

前两个问题你可以参考“参考资料”中 [1] 的建议，再衡量自己的需求来決定。你不见得要依照官方的建议，第一次也许采用最单纯的配置方式会比较好，例如：一个文件库就只放一个项目，而且只把程序的 原始码放进文件库，也不去分主线支线了，因此项目的目录结构可以很单纯，程序原始码的根目录就是项目的根目录。自己动手做过几次以后， 再去观察文件库的内容，就会比较有感觉了，然后再来考虑自己团队的需求，自然就能找到最适合自己团队的配置方式了。