

第 3 章 配置服务器

目录

[基于 Apache 的服务器](#)

- [简介](#)
- [安装 Apache](#)
- [安装 Subversion](#)
- [配置](#)
- [多版本库](#)
- [路径为基础的授权](#)
- [使用 Windows 域认证](#)
- [多重认证源](#)
- [用 SSL 使服务器更安全](#)
- [Using client certificates with virtual SSL hosts](#)

[基于 svnserve 的服务器](#)

- [简介](#)
- [安装 svnserve](#)
- [运行 svnserve](#)
- [以服务形式运行 svnserve](#)
- [svnserve 的认证](#)
- [使用 svn+ssh 认证](#)
- [svnserve 以路径为基础的授权](#)
- [Cyrus SASL Support](#)

为了使用TortoiseSVN(或任何其他Subversion客户端), 你需要放置你的版本库, 你可以将版本库存于本地并使用`file://`协议访问, 也可以放置于服务器, 使用`http://`或`svn://`访问, 这两种服务器协议也可以被加密, 使用`http://`或`svn+ssh://`。本章介绍了在Windows主机上设置一个服务器的步骤。

More detailed information on the Subversion server options, and how to choose the best architecture for your situation, can be found in the Subversion book under [Server Configuration](#).

If you don't have a server and you work alone then local repositories are probably your best choice. You can skip this chapter and go directly to [第 4 章 版本库](#).

If you were thinking about setting up a multi-user repository on a network share, think again. Read [“Accessing a Repository on a Network Share”一节](#) to find out why we think this is a bad idea.

基于 Apache 的服务器

简介

所有可能的服务器当中, Apache为基础的服务器是最灵活的, 尽管配置有一点复杂, 但是提供了其他服务器没有的便利:

WebDAV

The Apache based Subversion server uses the WebDAV protocol which is supported by many other programs as well. You could e.g. mount such a

repository as a "Web folder" in the Windows explorer and then access it like any other folder in the file system.

浏览版本库

你可以将浏览器指向版本库的URL，无需安装Subversion客户端就可以浏览内容，这样可以扩大访问你数据的用户圈。

认证

你可以使用所有Apache支持的认证机制，包括SSPI和LDAP。

安全

因为Apache非常稳定和安全，你的版本库可以自动获得同样的安全性，包括SSL加密。

安装 Apache

The first thing you need before installing Apache is a computer with Windows 2000, Windows XP+SP1, Windows 2003, Vista or Server 2008.

警告

Please note that Windows XP without the service pack 1 will lead to bogus network data and could therefore corrupt your repository!

1. Download the latest version of the Apache web server from <http://httpd.apache.org/download.cgi>. Make sure that you download the version 2.2.x - the version 1.3.xx won't work!

The msi installer for Apache can be found by clicking on `other files`, then browse to `binaries/win32`. You may want to choose the msi file `apache-2.2.x-win32-x86-openssl-0.9.x.msi` (the one that includes OpenSSL).

2. Once you have the Apache2 installer you can double click on it and it will guide you through the installation process. Make sure that you enter the server-URL correctly (if you don't have a DNS name for your server just enter the IP-address). I recommend to install Apache *for All Users, on Port 80, as a Service*. Note: if you already have IIS or any other program running which listens on port 80 the installation might fail. If that happens, go to the programs directory, `\Apache Group\Apache2\conf` and locate the file `httpd.conf`. Edit that file so that `Listen 80` is changed to a free port, e.g. `Listen 81`. Then restart the installation - this time it should finish without problems.
3. Now test if the Apache web server is running correctly by pointing your web browser to `http://localhost/` - a preconfigured Website should show up.

小心

如果你决定将Apache安装为服务，缺省情况以本地系统帐户运行会发出警告，更安全的方法是为Apache创建一个单独的运行帐户。

请确认Apache运行的帐户是版本库目录的访问控制列表(右键目录|属性|安全)中一个明确的条目，对目录有完全的控制能力，否则，用户不能提交他们的修改。

即使Apache运行于本地系统，你仍然需要这个条目(这种情况下将是SYSTEM帐户)。

If Apache does not have this permission set up, your users will get "Access denied" error messages, which show up in the Apache error log as error 500.

安装 Subversion

1. Download the latest version of the Subversion Win32 binaries for Apache. Be sure to get the right version to integrate with your version of Apache, otherwise you will get an obscure error message when you try to restart. If you have Apache 2.2.x go to <http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=8100>.
2. 运行Subversion安装程序, 并根据指导安装, 如果Subversion认识到你安装了Apache, 你就几乎完成了工作, 如果它没有找到Apache服务器, 你还有额外的步骤。
3. 使用Windows资源管理器, 来到Subversion的安装目录(通常是c:\program files\Subversion), 找到文件/httpd/mod_dav_svn.so和mod_authz_svn.so, 复制这些文件到Apache的模块目录(通常是c:\program files\apache group\apache2\modules)。
4. 从 Subversion 安装目录将 /bin/libdb*.dll 和 /bin/intl3_svn.dll 复制到 Apache 的 bin 目录。
5. 使用记事本之类的文本编辑器修改Apache的配置文件(通常是 C:\Program Files\Apache Group\Apache2\conf\httpd.conf), 做出如下修改:

去掉以下几行的注释(删除 '#' 标记):

```
#LoadModule dav_fs_module modules/mod_dav_fs.so
#LoadModule dav_module modules/mod_dav.so
```

将以下两行到 LoadModule 节的末尾。

```
LoadModule dav_svn_module modules/mod_dav_svn.so
LoadModule authz_svn_module modules/mod_authz_svn.so
```

配置

Now you have set up Apache and Subversion, but Apache doesn't know how to handle Subversion clients like TortoiseSVN yet. To get Apache to know which URL will be used for Subversion repositories you have to edit the Apache configuration file (usually located in c:\program files\apache group\apache2\conf\httpd.conf) with any text editor you like (e.g. Notepad):

1. At the end of the config file add the following lines:

```
<Location /svn>
  DAV svn
  SVNListParentPath on
  SVNParentPath D:\SVN
  #SVNIndexXSLT "/svnindex.xsl"
  AuthType Basic
  AuthName "Subversion repositories"
  AuthUserFile passwd
  #AuthzSVNAccessFile svnaccessfile
  Require valid-user
</Location>
```

This configures Apache so that all your Subversion repositories are physically located below D:\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 passwd file.

2. To create the passwd file, open the command prompt (DOS-Box) again, change to the apache2 folder (usually c:\program files\apache group\apache2) and create the file by entering

```
bin\htpasswd -c passwd <username>
```

This will create a file with the name passwd which is used for authentication. Additional users can be added with

```
bin\htpasswd passwd <username>
```

3. 再次重启Apache服务。
4. 将浏览器指向`http://MyServer/svn/MyNewRepository`(*MyNewRepository*是你此前创建的版本库名), 如果一切正常, 你会被提示输入用户名和密码, 然后你会看到版本库的内容。

A short explanation of what you just entered:

表 3.1. Apache *httpd.conf* Settings

设置	解释
<Location /svn>	意思是Subversion版本库的URL是 <code>http://MyServer/svn/</code>
DAV svn	告诉Apache是哪个模块响应那个URL的请求 - 此刻是Subversion模块。
SVNListParentPath on	For Subversion version 1.3 and higher, this directive enables listing all the available repositories under <code>SVNParentPath</code> .
SVNParentPath D:\SVN	告诉Subversion需要查看的版本库位于D:\SVN之下
SVNIndexXSLT "/svnindex.xsl"	Used to make the browsing with a web browser prettier.
AuthType Basic	激活基本认证, 就是用户名/密码
AuthName "Subversion repositories"	用来说明何时弹出要求用户输入认证信息的认证对话框
AuthUserFile passwd	指定使用的认证密码文件
AuthzSVNAccessFile	位置Subversion版本库的访问控制文件的路径
Require valid-user	指定只有输入了正确的用户/密码的用户可以访问URL

But that's just an example. There are many, many more possibilities of what you can do with the Apache web server.

- 如果你希望所有人可以读你的版本库, 但是只有特定用户可以写, 你可以修改下面几行

```
Require valid-user
```

to

```
<LimitExcept GET PROPFIND OPTIONS REPORT>
Require valid-user
</LimitExcept>
```

- Using a *passwd* file limits and grants access to all of your repositories as a unit. If you want more control over which users have access to each folder inside a repository you can uncomment the line

```
#AuthzSVNAccessFile svnaccessfile
```

and create a Subversion access file. Apache will make sure that only valid users are able to access your `/svn` location, and will then pass the username to Subversion's `AuthzSVNAccessFile` module so that it can enforce more granular access based upon rules listed in the Subversion access file. Note that paths are specified either as `repos:path` or simply `path`. If you don't specify a particular repository, that access rule will apply to all repositories under `SVNParentPath`. The format of the authorization-policy file used by `mod_authz_svn` is described in [“路径为基础的授权”一节](#)

- To make browsing the repository with a web browser 'prettier', uncomment the line

```
#SVNIndexXSLT "/svnindex.xsl"
```

and put the files *svnindex.xsl*, *svnindex.css* and *menucheckout.ico* in your document root directory (usually *C:/Program Files/Apache Group/Apache2/htdocs*). The directory is set with the `DocumentRoot` directive in your Apache config file.

你可以直接在我们的代码库

<http://tortoisetsvn.tigris.org/svn/tortoisetsvn/trunk/contrib/other/svnindex>中拿到这三个文件。如果访问这个链接需要认证, 输入用户名称 `guest`, 无需密码。

The XSL file from the TortoiseSVN repository has a nice gimmick: if you browse the repository with your web browser, then every folder in your repository has an icon on the right shown. If you click on that icon, the TortoiseSVN checkout dialog is started for this URL.

多版本库

If you used the `SVNParentPath` directive then you don't have to change the Apache config file every time you add a new Subversion repository. Simply create the new repository under the same location as the first repository and you're done! In my company I have direct access to that specific folder on the server via SMB (normal windows file access). So I just create a new folder there, run the TortoiseSVN command TortoiseSVN → Create repository here... and a new project has a home...

如果你使用Subversion 1.3或更高版本, 可以使用`SVNListParentPath on`指示, 这样当你使用浏览器访问父路径而不是具体某个版本库时就会显示所有版本库列表。

路径为基础的授权

The `mod_authz_svn` module permits fine-grained control of access permissions based on user names and repository paths. This is available with the Apache server, and as of Subversion 1.3 it is available with `svnserve` as well.

一个可能的例子:

```
[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
```

请注意, 检查每一条路径是一件消耗极大的操作, 特别是修订版本日志, 服务器会检查在每一个修订版本的每一条路径是否可读, 对于影响很多文件的修订将会花费很多时间。

Authentication and authorization are separate processes. If a user wants to gain access to a repository path, she has to meet *both*, the usual authentication requirements and the authorization requirements of the access file.

使用 Windows 域认证

你已经注意到了，你需要为每个用户在`passwd`文件中创建用户名/密码条目，如果(因为安全原因)他们希望周期性的修改他们的密码，你需要手动的做出修改。

But there's a solution for that problem - at least if you're accessing the repository from inside a LAN with a windows domain controller: `mod_auth_sspi`!

The original SSPI module was offered by Syneapps including source code. But the development for it has been stopped. But don't despair, the community has picked it up and improved it. It has a new home on [SourceForge](#).

- Download the module which matches your apache version, then copy the file `mod_auth_sspi.so` into the Apache modules folder.
- Edit the Apache config file: add the line

```
LoadModule sspi_auth_module modules/mod_auth_sspi.so
```

to the `LoadModule` section. Make sure you insert this line *before* the line

```
LoadModule auth_module modules/mod_auth.so
```

- To make the Subversion location use this type of authentication you have to change the line

```
AuthType Basic
```

to

```
AuthType SSPI
```

also you need to add

```
SSPIAuth On
SSPIAuthoritative On
SSPIDomain <domaincontroller>
SSPIOmitDomain on
SSPIUsernameCase lower
SSPIPerRequestAuth on
SSPIOfferBasic On
```

within the `<Location /svn>` block. If you don't have a domain controller, leave the name of the domain control as `<domaincontroller>`.

Note that if you are authenticating using SSPI, then you don't need the `AuthUserFile` line to define a password file any more. Apache authenticates your username and password against your windows domain instead. You will need to update the users list in your `svnaccessfile` to reference `DOMAIN\username` as well.

重要

The SSPI authentication is only enabled for SSL secured connections (https). If you're only using normal http connections to your server, it won't work.

To enable SSL on your server, see the chapter: [“用 SSL 使服务器更安全”一节](#)

提示

Subversion *AuthzSVNAccessFile* files are case sensitive in regard to user names (*JUser* is different from *juser*).

In Microsoft's world, Windows domains and user names are not case sensitive. Even so, some network administrators like to create user accounts in CamelCase (e.g. *JUser*).

使用SSPI的一个问题是用户名和密码是用户在提示输入时发送到Subversion的，而IE经常会不管你的帐户是如何建立的都会自动发送你的用户名。

The end result is that you may need at least two entries in your *AuthzSVNAccessFile* for each user -- a lowercase entry and an entry in the same case that Internet Explorer passes to Apache. You will also need to train your users to also type in their credentials using lower case when accessing repositories via TortoiseSVN.

Apache's Error and Access logs are your best friend in deciphering problems such as these as they will help you determine the username string passed onto Subversion's *AuthzSVNAccessFile* module. You may need to experiment with the exact format of the user string in the *svnaccessfile* (e.g. *DOMAIN\user* vs. *DOMAIN//user*) in order to get everything working.

多重认证源

也可以为Subversion使用不止一个的认证源，为此，你需要将每一种认证设置为non-authoritative，这样Apache会在多个源检查用户名/密码。

一个常见的场景就是同时使用Windows域和*passwd*文件认证，这样你可以为没有Windows域帐户的用户提供访问SVN的权限。

- To enable both Windows domain and *passwd* file authentication, add the following entries within the *<Location>* block of your Apache config file:

```
AuthAuthoritative Off
SSPIAuthoritative Off
```

Here is an example of the full Apache configuration for combined Windows domain and *passwd* file authentication:

```
<Location /svn>
  DAV svn
  SVNListParentPath on
  SVNParentPath D:\SVN

  AuthName "Subversion repositories"
  AuthzSVNAccessFile svnaccessfile.txt

  # NT Domain Logins.
  AuthType SSPI
  SSPIAuth On
  SSPIAuthoritative Off
  SSPIDomain <domaincontroller>
  SSPIOfferBasic On

  # Htpasswd Logins.
  AuthType Basic
  AuthAuthoritative Off
  AuthUserFile passwd

  Require valid-user
</Location>
```

用 SSL 使服务器更安全

Even though Apache 2.2.x has OpenSSL support, it is not activated by default. You need to activate this manually.

1. In the apache config file, uncomment the lines:

```
#LoadModule ssl_module modules/mod_ssl.so
```

and at the bottom

```
#Include conf/extra/httpd-ssl.conf
```

then change the line (on one line)

```
SSLMutex "file:C:/Program Files/Apache Software Foundation/\
Apache2.2/logs/ssl_mutex"
```

to

```
SSLMutex default
```

2. Next you need to create an SSL certificate. To do that open a command prompt (DOS-Box) and change to the Apache folder (e.g. *C:\program files\apache group\apache2*) and type the following command:

```
bin\openssl req -config bin\openssl.cnf -new -out my-server.csr
```

You will be asked for a passphrase. Please don't use simple words but whole sentences, e.g. a part of a poem. The longer the phrase the better. Also you have to enter the URL of your server. All other questions are optional but we recommend you fill those in too.

Normally the *privkey.pem* file is created automatically, but if it isn't you need to type this command to generate it:

```
bin\openssl genrsa -out conf\privkey.pem 2048
```

Next type the commands

```
bin\openssl rsa -in conf\privkey.pem -out conf\server.key
```

and (on one line)

```
bin\openssl req -new -key conf\server.key -out conf\server.csr \
-config conf\openssl.cnf
```

and then (on one line)

```
bin\openssl x509 -in conf\server.csr -out conf\server.crt
               -req -signkey conf\server.key -days 4000
```

This will create a certificate which will expire in 4000 days. And finally enter:

```
bin\openssl x509 -in conf\server.crt -out conf\server.der.crt -outform DER
```

These commands created some files in the Apache *conf* folder (*server.der.crt*, *server.csr*, *server.key*, *.rnd*, *privkey.pem*, *server.cert*).

3. Restart the Apache service.
4. 将你的浏览器指向 [http://servername/svn/project ...](http://servername/svn/project...)

SSL and Internet Explorer

如果你使用SSL保护你的服务器，并使用windows域来进行认证，你会发现不能使用IE浏览版本库了，不需要担心 - 那只是因为IE没有经过认证，其他浏览器没有这个问题，TortoiseSVN和其他Subversion客户端仍然可以得到认证。

如果你一直希望使用IE浏览你的版本库，你可以选择:

- define a separate `<Location /path>` directive in the Apache config file, and add the `SSPIBasicPreferred On`. This will allow IE to authenticate again, but other browsers and Subversion won't be able to authenticate against that location.
- 也提供未加密(没有SSL)认证的浏览，奇怪的IE在没有使用SSL的认证时没有任何问题。
- In the SSL "standard" setup there's often the following statement in Apache's virtual SSL host:

```
SetEnvIf User-Agent ".*MSIE.*" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
```

There are (were?) good reasons for this configuration, see http://www.modssl.org/docs/2.8/ssl_faq.html#ToC49 But if you want NTLM authentication you have to use `keepalive`. If You uncomment the whole `SetEnvIf` you should be able to authenticate IE with windows authentication over SSL against the Apache on Win32 with included `mod_auth_sspi`.

强制 SSL 访问

When you've set up SSL to make your repository more secure, you might want to disable the normal access via non-SSL (http) and only allow https access. To do this, you have to add another directive to the Subversion `<Location>` block: `SSLRequireSSL`.

An example `<Location>` block would look like this:

```
<Location /svn>
  DAV svn
  SVNParentPath D:\SVN
  SSLRequireSSL
  AuthType Basic
  AuthName "Subversion repositories"
  AuthUserFile passwd
  #AuthzSVNAccessFile svnaccessfile
  Require valid-user
</Location>
```

Using client certificates with virtual SSL hosts

Sent to the TortoiseSVN mailing list by Nigel Green. Thanks!

In some server configurations you may need to setup a single server containing 2 virtual SSL hosts: The first one for public web access, with no requirement for a client certificate. The second one to be secure with a required client certificate, running a Subversion server.

Adding an `SSLVerifyClient Optional` directive to the *per-server* section of the Apache configuration (i.e. outside of any `VirtualHost` and `Directory` blocks) forces Apache to request a client Certificate in the initial SSL handshake. Due to a bug in `mod_ssl` it is essential that the certificate is requested at this point as it does not work if the SSL connection is re-negotiated.

The solution is to add the following directive to the virtual host directory that you want to lock down for Subversion:

```
SSLRequire %{SSL_CLIENT_VERIFY} eq "SUCCESS"
```

This directive grants access to the directory only if a client certificate was received and verified successfully.

To summarise, the relevant lines of the Apache configuration are:

```
SSLVerifyClient Optional

### Virtual host configuration for the PUBLIC host
### (not requiring a certificate)

<VirtualHost 127.0.0.1:443>
  <Directory "pathtopublicfileroot">
    </Directory>
  </VirtualHost>

### Virtual host configuration for SUBVERSION
### (requiring a client certificate)
<VirtualHost 127.0.0.1:443>
  <Directory "subversion host root path">
    SSLRequire %{SSL_CLIENT_VERIFY} eq "SUCCESS"
  </Directory>

  <Location /svn>
    DAV svn
    SVNParentPath /pathtorepository
  </Location>
</VirtualHost>
```

[上一页](#)

摘要

[起始页](#)

[下一页](#)

基于 svnserve 的服务器