Subversion 系統

多年來，並發版本系統（CVS）一直是在 Linux 上管理代碼或者文本的標準。作？基於 RCS 上建立但卻允許多用戶協作的系統而言，CVS 記錄所有文件的修改資訊。這對於程式開發者、網路設計者和系統管理員而言，是非常有用的。然而，CVS 逐漸顯示出它的衰老，出現了相似的源代碼管理軟體。然而大多這種東西都是以牟利？主要目的的。

Subversion 就是一種相對新鮮的源代碼管理系統。雖然事實上它還在不斷的反展之中，但是 Subversion 已經是一個非常穩定而且成熟的？品。它是一個全新的系統，其功能可以和 CVS 媲美，同時，它要比 CVS 更直觀，更容易操作。本文就 Subversion 的安裝和一些特殊功能作一個介紹。

安裝伺服器端

下載 Apache 和 SVN 源碼包

從官方網站臺下載 httpd-2.0.52.tar.gz,subversion-1.1.1.tar.gz
(因？ redhat 9 默認安裝的 Apache 沒有並包含--enable-so 選項，所以無法？生 mod_dav_svn.沒有這個模組，SVN 就無法採用 http 方式運行，所以必須重新編譯新的 Apache)

以 root 身份執行：

```
#tar zxvf httpd-2.0.52.tar.gz
#cd httpd-2.0.52
#./configure --enable-dav --enable-so --enable-maintainer-mode
#make
#make install
```

此時會？生/usr/local/apache2 目錄,接著執行:

```
#tar zxvf subversion-1.1.1.tar.gz
#./configure --with-apxs=/usr/local/apache2/bin/apxs
# rm /usr/local/lib/libsvn*
# make clean && make && make install
```

此時會自動在/usr/local/apache2/conf/httpd.conf 添加

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

安裝完成後,運行 svnserver --version 確認版本？ 1.1.1。
SVN 伺服器安裝結束.

安裝客戶機端

window 客戶機：

直接安裝 TortoiseSVN-1.1.1-UNICODE_svn-1.1.1.msi，方法同一般軟體安裝相

同。

Linux 客戶機：

方法與安裝伺服器相同。
(注意 redhat 9 默認安裝的 SVN 版本? 0.17.1,它的用戶端命令 svn 無法與新的 SVN 伺服器通訊,必須重新安裝)

建立倉庫 Repository

Subversion 的檔案庫是個中央倉儲，用來存放任意數量專案的受版本控管資料，建立方法很簡單
#svnadmin create path/to/repos
舉個例子：
#svnadmin create /home/mysvn
#chown –R nobody /home/mysvn

運行伺服器

Subversion 伺服器有兩種運行方式，一是可以作? Apache 2.0 的一個模組，以 WebDAV/DeltaV 協定與外界連通；另外，也可使用 Subversion 自帶的小型伺服器程式 svnserve。該程式使用的是自帶的通訊協定，可以很容易地透過 SSH 以

以 http 方式運行

在/usr/local/apache2/conf/httpd.conf 中加入:

```
<Location /svn/repository>
  DAV svn
  SVNPath /home/mysvn
</Location>
```

在伺服器的瀏覽器中輸入網址:
http://localhost/svn/repository/
這時候,你會看到這樣的顯示：

這表明伺服器已經以 http 方式正常運行了.

以 svnserve 方式運行

這種方式的運行又可以分? 以下兩種(這和 vsftp 有些相似)

1) **standalone mode**

直接運行 #svnserve –d
運行 lsof -i :3690 可以看到 SVN 伺服器已經在運行

2) xinetd mode

在/etc/xinetd.d/下生成 svnserve 文件，內容如下

```
service svnserve
{
        disable = no
        socket_type             = stream
        protocol                 = tcp
        wait                    = no
        user                    =  apache
        server                  = /usr/local/bin/svnserve
        server_args             = -i
}
```

編輯 /etc/services 檔,加入底下兩行:

```
svnserve        3690/tcp                        # Subversion svnserve
svnserve        3690/udp                        # Subversion svnserve
```

重? xinetd 服務，運行 lsof -i :3690 可以看到 SVN 伺服器已經在運行

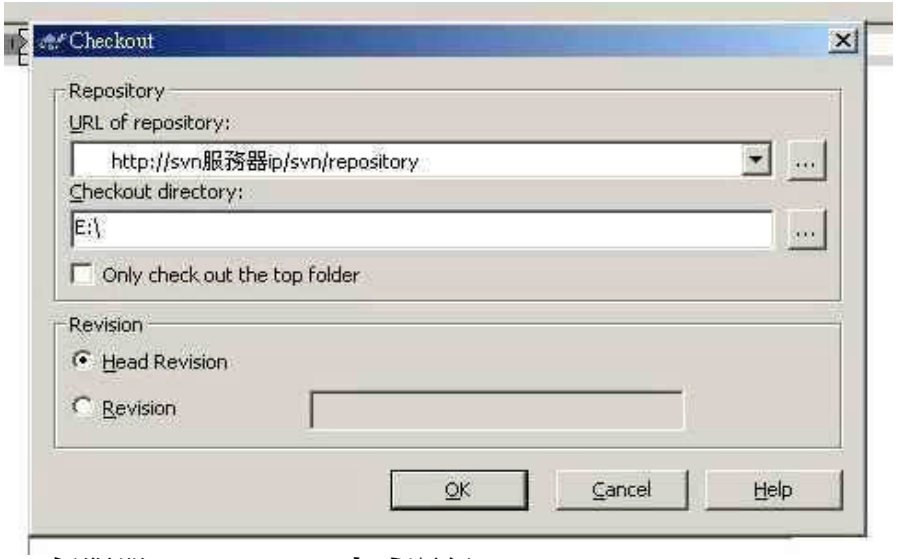
客戶機訪問

客戶機的訪問方法與伺服器的運行方式有直接關係

window 客戶機：

1) 伺服器以 http 方式運行

安裝完 TortoiseSVN-1.1.1-UNICODE_svn-1.1.1.msi 後,在你想工作的目錄下點擊右鍵，執行 checkout，按上圖輸入即可。



2) 伺服器以 svnserve 方式運行

同上的區別只是 URL of repository 變? svn://svn 伺服器 ip/home/mysvn
或者 svn+ssh://svn 伺服器 ip/home/mysvn
(注意不是//svn 伺服器 ip//svn/repository)

linux 客戶機：

1) 伺服器以 http 方式運行

執行 #svn checkout http: //svn 伺服器 ip/svn/repository
2) 伺服器以 svnserve 方式運行
執行 #svn checkout svn://svn 伺服器 ip/home/mysvn
或者 #svn checkout svn+ssh://svn 伺服器 ip/home/mysvn

客戶認證機制

這與伺服器的運行方式有關

伺服器以 http 方式運行

比如我們想給 Sally 與 Harry 送交存取檔案庫的許可權. 首先, 我們必須把它們加入到密碼檔案.

第一次: 以 -c 建立檔案
htpasswd -c /etc/svn-auth-file harry
New password: *****
Re-type new password: *****
Adding password for user harry
htpasswd /etc/svn-auth-file sally
New password: *******
Re-type new password: *******
Adding password for user sally
#
接著,在/usr/local/apache2/conf/httpd.conf 的加入:

```
<Location /svn/repository >
   DAV svn
   SVNPath  /home/mycvs
   AuthType  Basic
   AuthName "Subversion  repository"
   AuthUserFile  /etc/svn-auth-file
   Require valid-user
</Location>
```

重新啟動 Apache 後, 如果有人要訪問 SVN 伺服器, 系統會要求他輸入用戶名和密碼。 只有輸入 Sally 或 Harry 的用戶名和相應的密碼, 才可以對檔案庫進行修改和訪問

伺服器以 svnserve 方式運行

默認下客戶可以以匿名方式通過 svn://方式任意訪問檔案庫, ? 了限制其許可權, 比如只允許讀操作, 可以通過修改檔案庫 conf 子目錄中的 svnseve.conf 文件來實現。
#vi /home/mysvn/conf/svnseve.conf
修改[general]欄位下內容? :

```
anon-access = read
```

如果設? anon-access = none, 則匿名用戶不可以通過 svn://方式訪問檔案庫? 了實現用戶認證,我們一般採用svn+ssh://訪問機制。
首先在svnseve.conf文件設置anon-access = none禁止匿名用戶通過svn://方式

訪問檔案庫，然後在其後加入
auth-access = write
auth-access 是限制有援權的使用者(使用svn+ssh:// 來登入) 的存取許可權 我們設? 是可以讀寫。
當用戶通過svn+ssh://訪問時，伺服器會自動啟動ssh認證機制，要求用戶輸入密碼，對於window用戶來說還需要安裝第三方軟體openssh，才可以採用這種機制

Hook scripts

挂勾 (hook) 是改動檔案庫時所觸發的程式，比如當你提交更動前，會先觸發 pre-commit，提交更動後，則會觸發 post-commit，我們可以利用 hook 來實現一些自動控制。檔案庫的 hook 子目錄中，預設是放置各個檔案庫挂勾的模板:

post-commit.tmpl
pre-revprop-change.tmpl
post-revprop-change.tmpl
start-commit.tmpl
pre-commit.tmpl
如果要使用這些 hook，就必須把它的尾碼名.tmpl 去掉，拷貝?
post-commit
pre-revprop-change
post-revprop-change
start-commit
pre-commit
這裏主要介紹 pre-commit 和 post-commit(事實上它們就是在特定的情況下被觸發的普通的 shell 程式,至於 shell 的內容由用戶自己隨意編寫，但是要保證名稱不能改動)

pre-commit

> 本挂勾執行的時間? 異動完成之後，送交之前.檔案庫會傳遞兩個引數給這個程式: 檔案庫的路徑，以及準備送交的異動名稱. 如果程式傳回一個非零的結束值，送交會被中止，而異動會被刪除.

如何應用 pre-commit 我們不妨舉個例子：
假如有一個專案由 Mail Team,Login Team 和 PHP Team 三個 Team 共同通過 SVN 系統開發完成。當專案準備發佈的時候，PM 人員發現 Mail 功能方面存在一些 bug，需要 Mail Team 去修改，? 了防止其他 Team 的人員修改系統，我們可以在任何改動檔案庫的企圖之前用 pre-commit 去檢查 log message 資訊,(因? 任何更動檔案庫的操作都必須提供 log message 資訊,PM 可以事先與 Mail Team 約定好一個 log message),如果與 pre-commit 中設定的 log message 不相符，則不能提交更動。
pre-commit 根源程式如下:

```
#!/bin/sh
REPOS="$1"
TXN="$2"
SVNLOOK=/usr/local/bin/svnlook
```

```
$SVNLOOK log -t "$TXN" "$REPOS" | \
   grep –w "bug1234" > /dev/null || exit 1
exit 0
```

本例中的 log message？ ”bug1234”,任何人想要提交更動就必須用 –m “bug1234”參數,採用-m “bug123”, -m “bug12345”都會提交失敗。

post-commit

本挂勾執行的時間是在異動送交, 新修訂版被建立之後. 大多數的人用這個挂勾來寄出關於本次送交的電子郵件, 或是建立檔案庫的備份. 檔案庫會傳遞兩個引數給這個程式: 檔案庫的路徑, 以及新建立的修訂版號. 本程式的結束碼會被忽略.

Subversion 源碼樹的 tools/hook-script 目錄中包含了一個 commit-email.pl 命令, 可以用來寄送包含描述指定送交的電子郵件. 這個郵件包含了更動路徑列表, 該送交所對應的記錄訊息, 使用者, 送交的日期,以及一個以 GNU diff 樣式表示的本次更動差異. 我們可以將這個程式與 post-commit 這個 hook 搭配起來使用來實現檔案庫更動後自動 mail 給相關人員的功能。

post-commit 根源程式如下:

```
#!/bin/sh
REPOS="$1"
REV="$2"
commit-email.pl "$REPOS" "$REV" PM@yourdomain.com
##需要指明commit-email.pl的絕對路徑
```

特殊性質

除了對你的目錄與檔案進行版本控制之外, Subversion 還提供了一個介面, 可用來新增, 修改, 以及移除已納入版本控制的目錄與檔案的版本控制描述資料. 我們稱這個描述資料? 性質,在這裏我主要介紹以下幾個比較重要的特殊性質

svn:mime-type

svn:mime-type 性質在 Subversion 中有很多作用. 除了作? 儲存檔案的多用途網際網路郵件延伸語法 (MIME) 分類之外, 這個性質的內容還會決定幾項 Subversion 的行? 特徵.

舉個例子, 如果 svn:mime-type 性質設? 文字的 MIME 類別 , Subversion 會假設該檔的內容是二進位(也就是人類看不懂的資料). Subversion 提供的功能中, 其中一項是在從伺服器收到工作檔的更新中, 依文字內容與文字列進行合併. 但是對含有二進位資料的檔案, 根本就沒有 “ 文字列 ” 的概念. 因此, Subversion 對這些檔案在更新時, 不會試著進行內文合併. 它改用另一種方式。

一般來說 Subversion 在執行 **svn import** 與 **svn add** 子命令時, 會使用二進位元偵測運演算法的方式來協助使用者.但是如果 Subversion 猜錯了, 或是你希望將 svn:mime-type 設定成更? 明確的值(可能是 image/png)你都可以移除或是手動編輯這個性質.

svn:ignore

svn:ignore 性質包含了檔案樣式的列表, Subversion 處理時會忽略. 它可以與執行時期設定的 global-ignores 選項一起工作, 以便在類似 **svn status** 的命令中過濾掉未納入版本控制的目錄與檔案.

我們知道新增的文件和目錄必須透過 **svn add** 命令, 才會被納入 Subversion 的管理. **svn status** 命令會將工作複本中未納入版控制目錄與檔案顯示出來.

```
$ svn status calc
  M       calc/button.c
?         calc/calculator
?         calc/data.c
?         calc/debug_log
?         calc/debug_log.1
```

在這個範例中, 用?標注出來的文件就是未納入版控制的檔案.如果你不想每次執行 **svn status** 時, 都看到這些檔案, 那? svn:ignore 性質就是解決方案。你可以透過 **svn propedit svn:ignore calc** 對 calc 目錄加上一些忽略樣式. 舉個例子,將以下的值作? svn:ignore 性質的新內容:

```
calculator

debug_log*
```

加上這個性質後再執行你的 **svn status** 輸出便會不同：

```
$ svn status
  M       calc
  M       calc/button.c
?         calc/data.c
```

現在, 所有不想看到的東西都從輸出中消失了!

svn:keywords

Subversion 具有取代關鍵字(有關納入版本控制檔案的有用資訊)進入檔案內容的功能.

舉個例子, 假設你有個文件, 想要在裏面顯示最近一次修改的日期. 你可以把這個負擔加諸文件的作者身上, 讓他們每一次送交更動之前, 順便添加最近一次修改日期的部份. 但是遲早有人會忘記這件事. 換個方式, 只要叫 Subversion 對 LastChangedDate 關鍵字進行關鍵字取代即可.

Subversion 定義了可用來進行取代的關鍵字列表. 這個列表包含了以下五個關鍵字:

LastChangedDate
LastChangedRevision
LastChangedBy
HeadURL
Id
如果只把關鍵字定位錨加進檔案裏的話, 什? 事也不會發生.要告訴 Subversion 是否該對某一個檔案進行關鍵字取代,得使用 svn:keywords 這個性質。當它被設定時, 它會控制該檔案哪個關鍵字應該被取代.

舉個例子, 假設你有一個納入版本控制的檔案, 名? weather.txt, 看起來像這樣:

Here is the latest report from the front lines.
$LastChangedDate$
Rev
Cumulus clouds are appearing more frequently as summer approaches.

如果沒有設定該檔案的 svn:keywords 性質, Subversion 什? 事也不會作. 讓我們開? 關鍵字 LastChangedDate 的內容取代.

$ svn propset svn:keywords "LastChangedDate Author" weather.txt
property `svn:keywords' set on 'weather.txt'
$

在你送交了這個性質更動之後, Subversion 會顯示? :

Here is the latest report from the front lines.
$LastChangedDate: 2002-07-22 21:42:37 -0700 (Mon, 22 Jul 2002) $
Rev
Cumulus clouds are appearing more frequently as summer approaches.

這樣不管誰提交這個文件, 都會在裏面顯示最近一次修改的日期。

svn:eol-style

除非另外指定版本控制檔案的 svn:mime-type 性質, Subversion 會假設檔案包含人類可讀的資料.這對於列尾符號 (EOL) 是很不幸地, 因? 不同的作業系統會使用不同的符號來表示一列的結尾. 舉個例子, 一般用在 Windows 平臺上的列尾符號是兩個 ASCII 控制字元 :返回字元 (CR) 與換行字元 (LF). 但是 Unix 軟體就只使用 LF 字元來表示一列的結尾.這樣以來 window 客戶提交的檔案中的 CR 字元在 linux 用戶端會顯示成 ^M, 而 linux 客戶提交的檔案中 CR 字元在 Windows 用戶端會被忽略。結果將檔案裏的所有文字列合併成一個超長的文字列, 這是因? 沒有返回 CRLF 字元組合的存在來表示一個換行。 解決的

方法是 svn:eol-style 性質. 當這個性質設定? native 時, Subversion 會根據系統的類型來決定是否對該檔案的結尾進行自動處理。.

svn:externals

有的時候, 一個工作複本可能包含了數個不同來源的工作複本. 舉個例子, 你可能想要有數個不同的目錄, 各來自不同的檔案庫.我們可以通過 svn:externals 性質來宣告這一對對應關係。內容是子目錄對應至 Subversion 檔案庫 URL 的多行表格.

```
$ svn propget svn:externals calc

third-party/sounds                http://sounds.red-bean.com/repos
third-party/skins                 http://skins.red-bean.com/repositories/skinproj
third-party/skins/toolkit     http://svn.red-bean.com/repos/skin-maker
```

當有人取出 calc 目錄的工作複本, Subversion 還會繼續取出在外部定義裏的專案.

```
$ svn checkout http://svn.example.com/repos/calc
A    calc
A    calc/Makefile
A    calc/integer.c
A    calc/button.c
Checked out revision 148.

Fetching external item into calc/third-party/sounds
A    calc/third-party/sounds/ding.ogg
A    calc/third-party/sounds/dong.ogg
A    calc/third-party/sounds/clang.ogg
Checked out revision 14.

Fetching external item into calc/third-party/skins
…
```

小結

Subversion 有一份很好的文檔——《Version Control with Subversion》（http://svnbook.red-bean.com/）。它提供了有關 Subversion 的各方面內容, 如使用、管理和開發等。
經過數年的開發, 以替代 CVS？目標的 Subversion, 相信以其強大的功能, 對 CVS 良好的繼承性, 一定會有很好的發展。

作者簡介

姓名：雷凱
工作單位：升技主板(蘇州)研發中心
聯繫地址：蘇州市新區馬運路羅禮科技有限公司研發中心 郵編 215000
E-mail: tigerleihm@yahoo.com.cn

參考資料：Version Control with Subversion （http://svnbook.red-bean.com/）

“本文作者是雷凱 升技主板(蘇州)研發中心工程師。他目前在中國蘇州 升技主板(蘇州)研發中心工作。可以通過 tigerleihm@yahoo.com.cn 與他聯繫。”