CVS-RCS- (原始程式碼版本控制系統) HOW-TO 文件

作者:Al Dev (Alavoor Vasudevan) *alavoor@yahoo.com*
譯者:Cyril Huang *cyril_huang@yahoo.com*

v7.0, 20 Feb 2000 翻譯日期: 5 April 2000

這份文件是一份 "實際操作的說明"，以便於能使您很快的設定 CVS/RCS 原始程式碼控制系統。這份文件裡也有一些將 CVS 上常用的混合命令包成可設定的 shell scripts。這些 scripts 爲 CVS 提供了一個簡單的使用者介面。這份文件的內容不僅能適用於 Linux 系統也適用於其他像 Unix 的系統，例如Solaris, HPUX, AIX, SCO, Sinix, BSD, SCO 等等。

目錄

1 簡介

原始碼控制系統是一個必須能管理那些在軟體計劃發展時原始碼所做的改變。軟體開發者需要一個完整的原始碼改變歷史紀錄， 以便於在發生問題時，能夠追溯到以前穩定的版本。 既然原始程式碼對於任何的軟體計劃與開發，都是花時間與金錢中最關鍵的部分，所以花時間藉由使用原始碼控制系統像 CVS 和 RCS 來安全的保護***(safe-guarding)***原始程式碼是非常重要的。

CVS (Concurrent Version Control System) 是一個能讓很多程式開發者同時做軟體開發的非常強大工具。它使用了RCS 的檔案規定格式但多了一層像應用程式介面的包裝，架在 RCS 的上層。

(譯註: RCS 是較老的版本控制，一個受 RCS 管制的檔案看起來是這樣子的 proj1.c,v，CVS 沿用了一些 RCS 的規定。)

CVS 能夠紀錄你的檔案的歷史紀錄(通常是原始程式碼，但是其他型態的檔案則不一定)。 CVS 只存了不同版本中檔案的差異，而不是你所建立的每個版本中的每個檔案。 CVS 也保持了一個何時，何人更改檔案，爲什麼更改檔案等等不同觀點的歷史紀錄。

CVS 對於軟體的發行和多人同時更改目前原始碼的管理是非常的有幫助。 他並不只是要對單一目錄下的檔案提供版本控制， 相反的，CVS 更提供了多層有組織的目錄檔案的版本控制。 在這個目錄下除了你的原始程式碼外，還包含有一個 CVS 所建立的改版控制目錄與檔案。

這些目錄與檔案最後被合併在一起形成一個軟體的發行。

CVS 能被使用在 "C", "C++", Java, Perl, HTML 和其他檔案。

2 那一種版本控制系統適合我? CVS 或 RCS

CVS 實際上是架在 RCS 之上的， CVS 只是一堆更強大能控制一個有完整原始程式碼階層目錄的工具。 我們非常強烈的推薦您使用 CVS，因爲您能夠很有彈性的用 perl , korn bash shell 等 scripts 語言設定您自己的 CVS 系統。 請看一些 korn shell scripts 的範例 4 ()。

CVS 的優點

- CVS 是非集中式的管理，使用者從儲存櫃 (repository) 登出一個檔案目錄， 並且有他自己的獨立的穩定目錄樹。
- CVS 能夠在發行整個計劃的原始目錄樹中"蓋上印記" ("STAMP")。
- CVS 能夠使大家同時修改檔案。
- CVS 能夠用 shell scripts 或 perl 設定成檔案鎖住成單一使用或同時修改檔案模式。

CVS 的缺點

- 需要比 RCS 多一點的管理。

- 非常成熟複雜的系統，是目前應用上已有的最高技術。(感謝網友tsaipaw@mars.seed.net.tw來信指證我的錯誤)
- 有豐富的命令還有命令選項，因此對於初學者來說有很陡的學習曲線。 簡單使用的 shell scripts 可在這裡找到 4 ()。

RCS 的優點

- RCS 非常容易設定。較少一些管理上的工作。
- RCS 用在一個每個人在一起工作的集中區域。
- RCS 對於簡單的系統很有用。
- 非常嚴謹的單一檔案修改模式 - 同步與同時是不允許的。

RCS 的缺失

- 由於使用單一目錄控制與檔案鎖住，不可能由很多的程式設計者做同時的開發。因爲單一目錄下很多人對檔案的改變，會造成 make 的使用錯誤。
- 不能對整個軟體計劃戳上發行(releases)的印記。

這份文件也包含一些 shell scripts 以提供簡單的命令來作登出 (check-out), 登錄 (check-in), 送交(commit) 檔案的動作。 請看一些 shell scripts 的範例 4 ()。

對於 RCS 而言，請看 Linux CD-ROM 裡面的RCS mini-howto。

```
cd /mnt/cdrom/Redhat/RPMS
ls -l howto-6.0-*.noarch.rpm
rpm -qpl howto-6* | grep -i rcs
```

或者看 <http://sunsite.unc.edu/LDP/HOWTO/mini/RCS-HOWTO.html>

3 設定 CVS

首先，你需要安裝 CVS 套件，在Redhat Linux 上請用

```
cd /mnt/cdrom/Redhat/RPMS
rpm -i rcs*.rpm
rpm -i cvs*.rpm
To see the list of files installed do -
rpm -qpl cvs*.rpm | less
```

然後用 j, k, CTRL+f, CTRL+D, CTRL+B, CTRL+U 或上下左右鍵, page up/down 瀏覽一下結果。 請用 'man less' 查看 less 的用法

在其他的 unix 機器上，你可能需要下載 RCS CVS 的 tar.gz 檔案， 然後根據 README, INSTALL 檔的指示來安裝 CVS。 請到 <http://www.cyclic.com> 和 <http://www.loria.fr/~molli/cvs-index.html>

3.1 CVS 的專有環境變數

下列的環境變數需要在 /etc/profile 檔中設定，/etc/profile 是對所有使用者都有效的內定值設定檔， 如果沒有設定 /etc/profile，那麼你應該加這些設定到你自己的設定檔 /.bash_profile 內。

```
export EDITOR=/bin/vi
export CVSROOT=/home/cvsroot
export CVSREAD=yes
```

建造一個目錄來存你原始程式碼的儲藏櫃 (repository) 並且給予 unix group 與 user 讀寫的權力。 (譯註 :這個目錄下將會有很多你將來的原始碼。)

```
export CVSROOT=/home/cvsroot
mkdir $CVSROOT
chmod o-rwx $CVSROOT
chmod ug+rwx $CVSROOT
```

要初始化你的 CVS，並且從現在開始把你的原始程式碼交給 CVS 管理。請做 -

```
cvs init

(譯註；這個初始化的動作在於建造一個儲藏櫃，是一個目錄$CVSROOT/。
同時 $CVSROOT/CVSROOT 也在此時被建造，這個模組目錄下面是控制你CVS的administration files，
裡面的檔案做一些修改後，可以使CVS更強大好用。
$CVSROOT 下的目錄每個都是 module 的意思，一個 module 可以就是一個專案計劃。
但也可能是你把一個計劃拆成很多 module，不同 module 交給不同的 team 去發展。)

# 一定要換到想要 CVS 控制的計劃目錄下喔
cd $HOME/my_source_code_dir

# 把整個目錄納入管理用 import 命令
cvs import my_source_code_dir V1_0 R1_0

(譯註:其實是 cd 到你的project下後，cvs import 模組 vendor_tag release_tag，
不一定要是目錄名稱 my_source_code_dir，vendor_tag, release_tag 只是識別用的東西，
將來你可以用 tag 來存取你要的特定版本
這個動作會在 $CVSROOT/ 下開個" 模組 "的目錄，然後把 my_source_code_dir 整個放到 CVS 下管理，
$HOME/my_source_code_dir 就沒用了。import 的動作是把已經寫好的一堆 code 擺進來，
如果將來想新增檔案xxxx.c，必須先寫好xxxx.c，再用 cvs add xxxx.c)
```

3.2 從 RCS 轉換到 CVS 系統

要轉換已經存在的 RCS 檔案到 CVS，請使用下面的 script。並確定你從你的 Linux CD-ROM 安裝了 korn shell 套件 pdksh*.rpm。

注意：***Korn shell /bin/ksh*** 在你從 ***Linux CD-ROM*** 安裝 ***pdksh*.rpm*** 時就會產生

```
#!/bin/ksh

#############################################################
# Program to Migrate the existing source code in RCS to CVS
#
# Needs the korn shell RPM package  pdksh*.rpm from Linux
# contrib cdrom
#############################################################

#
# rcs2cvs - convert source tree from RCS to CVS
#

# project to convert
PROJECT='project'

# current RCS root
RCSROOT="$HOME/rcs"

if cd "$RCSROOT/$PROJECT"
then
        cd "$RCSROOT"
else
        echo >&2 "`basename "$0"`: can't change to RCS directory '$RCSROOT/$PROJECT'."
        exit 1
fi

# current CVS root
CVSROOT="$HOME/cvs"

# create new CVS directory for project 'project'
if mkdir "$CVSROOT/$PROJECT"
then
        :
else
        echo >&2 "`basename "$0"`: can't create CVS directory '$CVSROOT/$PROJECT'."
        exit 2
fi

# create CVS project tree from RCS tree
find "$PROJECT" -type d -name RCS -print |
while read RCS
do
        CVS="`dirname "$RCS"`"
        (if cd "$RCS"
        then
#               if find . -type f -name '*,v' -print | cpio -pdmv "$CVSROOT/$CVS"
```

```
                if find . -type f -print | cpio -pdmv "$CVSROOT/$CVS"
                then
                        :
                else
                        echo >&2 "`basename "$0"`: can't convert RCS subdirectory '$RCSROOT/$RCS' to CVS subdirectory '$CVSROOT/$CVS'."
                fi
        else
                echo >&2 "`basename "$0"`: can't change to RCS subdirectory '$RCSROOT/$RCS'."
        fi)
done
```

現在 RCS 已經被改成 CVS 系統下的 'project'. 你可以開始用CVS命令來存取 'project' 這個模組了.

4 Shell Scripts

下面的 scripts 是基本 CVS 命令的集合，而且是 Korn shell 的 scripts。你可以把他轉成 perl 或者 bash 。你可以自己修改成你想要的樣子。 這些只是運用基本 CVS 命令但有些特殊的花樣加在裡面。例如， sedit 這個 script 提供了檔案鎖住的功能使得其他人知道有某人正在修改這個檔案， 當然你也可以直接使用 CVS 命令而不用這些 scripts，這些 scripts 只是在展示 CVS 是多麼的有彈性。

把這些 scripts 複製到 /usr/local/bin 下，並且此目錄應該在你的 PATH 環境變數中。

1. **sget** [-r revision_number] <file/directory name> 要從 CVS 獲得一個唯讀檔案或整個唯讀目錄，請按 4.1 ()
2. **sedit** [-r revision_number] <filename> 要修改一個一個程式碼時，這個 scripts 會做檔案鎖住的動作，因此沒有別人可以登出這個檔案了。當然你可以改變這個 script 成你想要的功能 - 例如不鎖住，只出現警告訊息，或者相反的，非常嚴謹的鎖檔案。 請按 4.2 ()
3. **scommit** [-r revision_number] <filename> 要交出某個你修改的檔案或整個目錄。 把你的改變交給 CVS。 請按 4.3 ()
4. **supdate** <filename/directory> 要藉由從 CVS 得到最新的檔案來update 一個檔案或整個目錄。 請按 4.4 ()
5. **sunlock** [-r revision_number] <filename> 要把因為用 sedit 後的檔案鎖關掉。這會釋放檔案鎖 (Release File Lock)。 請按 4.5 ()
6. **slist** 要看目前正被你修改的檔案列表。 做 'ls -l | grep | ...' 命令， 請按 4.6 ()
7. **sinfo** <filename/directory> 要得到一個檔案的改版資訊。 請按 4.7 ()
8. **slog** <filename> 要得到一個 CVS 檔案改版的歷史紀錄， 請按 4.8 ()
9. **sdif** <filename> **sdif** -r rev1 -r rev2 <filename> 要得到你的檔案與 CVS 櫃子裡的檔案不同的地方在哪裡。 請按 4.9 ()

   注意: sdif 只有一個 'f'，因為這裡已經有一個 unix 命令叫 'sdiff'。
10. **sadd** <filename> 要新增一個檔案到 CVS 櫃子裡。 請按 4.10 ()

11. **sdelete** <filename> 要從 CVS 櫃子裡清掉一個檔案。 請按 4.11 ()

12. **sfreeze** <revision name> <directory name> 要凍結原始碼 (freeze codes)，這是將要發行 (release) 整個原始碼目錄樹。 請按 4.12 ()

13. **saddtree** <revision name> <directory name> 要新增一個目錄樹到 CVS。 請按 4.13 () 例如：

```
        cd $HOME;
        sfreeze REVISION_1_0  srctree
```

這將會凍結原始碼，並貼上一個標籤 REVISION_1_0，如此一來你就可以稍後用版本名字登出整個目錄樹。

****

4.1 sget

注意：***Korn shell /bin/ksh*** 在你從 ***Linux CD-ROM*** 安裝 ***pdksh*.rpm*** 時就會產生

請把他存成一般文字檔並改變存取權限 chmod a+rx

```
#!/bin/ksh

# CVS program sget
# Program to check out the file from CVS read-only

cmdname=`basename $0`

Usage()
{
        print "\nUsage: $cmdname [-r revision_number/symbolic_tag_name] <file/directory name> "
        print "The options -r are optional "
        print "For example - "
        print " $cmdname -r 1.1 foo.cpp"
        print " $cmdname foo.cpp "
        print " $cmdname some_directory "
        print "Extract by symbolic revision tag like - "
        print " $cmdname -r REVISION_1 some_directory "
        print " "
        exit
}

# Command getopt will not supported in next major release.
# Use getopts instead.
while getopts r: ii
do
        case $ii in
        r) FLAG1=$ii; OARG1="$OPTARG";;
```

```
        ?) Usage; exit 2;;
        esac
done
shift ` expr $OPTIND - 1 `

#echo FLAG1 = $FLAG1 , OARG1 = $OARG1

if [ $# -lt 1 ]; then
        Usage
fi

bkextn=sget_bak

hme=` echo $HOME | cut -f1 -d' '  `
if [ "$hme" = "" ]; then
        print "\nError: \$HOME is not set!!\n"
        exit
fi

# Check if file already exists....
if [ -f $1 ]; then
        user_perms=" "
        group_perms=" "
        other_perms=" "
        user_perms=`ls -l $1 | awk '{print $1 }' | cut -b3-3 `
        group_perms=`ls -l $1 | awk '{print $1 }' | cut -b6-6 `
        other_perms=`ls -l $1 | awk '{print $1 }' | cut -b9-9 `
        if [ "$user_perms" = "w" -o "$group_perms" = "w"  \
                -o "$other_perms" = "w" ]; then
                print "\nError: The file is writable. Aborting $cmdname ......"
                print "       You should either backup, scommit or delete the file and"
                print "       try $cmdname again\n"
                exit
        fi
fi

cur_dir=`pwd`
#echo $cur_dir

len=${#hme}
len=$(($len + 2))
#echo $len

subdir=` echo $cur_dir | cut -b $len-2000 `
#echo $subdir

if [ "$subdir" = "" ]; then
```

```
        fdname=$1
else
        fdname=$subdir"/"$1
fi

# Move the file
touch $1 2>/dev/null
\mv -f $1 $1.$bkextn

# Create subshell
(
cd $hme
#echo $fdname

# Use -A option to clear all sticky flags
if [ "$FLAG1" = "" ]; then
        cvs -r checkout -A $fdname
else
        cvs -r checkout -A -$FLAG1 $OARG1 $fdname
fi
)
#pwd

if [ -f $1 ]; then
        print "\nREAD-ONLY copy of the file $fdname obtained."
        print "Done $cmdname"
        #print "\nTip (Usage): $cmdname <file/directory name> \n"
fi
```

4.2 sedit

注意：***Korn shell /bin/ksh*** 在你從 ***Linux CD-ROM*** 安裝 ***pdksh*.rpm*** 時就會產生
請把他存成一般文字檔並改變存取權限 chmod a+rx

```
#!/bin/ksh
# CVS program sedit
# Program to check out the file from CVS read/write mode with locking

cmdname=`basename $0`

Usage()
{
#       print "\nUsage: $cmdname [-r revision_number] [-F] <filename>"
#       print "The options -r, -F are optional "
#       print "The option -F is FORCE edit even if file is "
#       print "locked by another developer"
```

```
        print "\nUsage: $cmdname [-r revision_number] <filename>"
        print "The options -r are optional "

        print "For example - "
        print " $cmdname -r 1.1 foo.cpp"
        print " $cmdname foo.cpp "
#       print " $cmdname -F foo.cpp "
        print " "
}

# Command getopt will not supported in next major release.
# Use getopts instead.
#while getopts r:F ii
while getopts r: ii
do
        case $ii in
        r) FLAG1=$ii; OARG1="$OPTARG";;
#       F) FLAG2=$ii; OARG2="$OPTARG";;
        ?) Usage; exit 2;;
        esac
done
shift ` expr $OPTIND - 1 `

#echo FLAG1 = $FLAG1 , OARG1 = $OARG1

if [ $# -lt 1 ]; then
        Usage
        exit
fi

hme=` echo $HOME | cut -f1 -d' '  `
if [ "$hme" = "" ]; then
        print "\nError: \$HOME is not set!!\n"
        exit
fi

bkextn=sedit_bak

cur_dir=`pwd`
#echo $cur_dir

len=${#hme}
len=$(($len + 2))
#echo $len

subdir=` echo $cur_dir | cut -b $len-2000 `
```

```
#echo $subdir

if [ "$subdir" = "" ]; then
        fdname=$1
else
        fdname=$subdir"/"$1
fi

# If file is already checked out by another developer....
cvs_root=` echo $CVSROOT | cut -f1 -d' ' `
if [ "$cvs_root" = "" ]; then
        print "\nError: \$CVSROOT is not set!!\n"
        exit
fi
cldir=$CVSROOT/$subdir/Locks
mkdir $cldir 2>/dev/null
rcsfile=$CVSROOT/$subdir/$1,v
#echo $rcsfile

if [ ! -e $rcsfile ]; then
        print "\nError: File $1 does not exist in CVS repository!!\n"
        exit
fi

# Get the tip revision number of the file....
# Use tmpfile as the arg cannot be set inside the sub-shell
tmpfile=$hme/sedit-lock.tmp
\rm -f $tmpfile 2>/dev/null
if [ "$FLAG1" = "" ]; then
        (
        cd $hme
        cvs log $fdname | head -6 | grep head: | awk '{print $2}' > $tmpfile
        )
        OARG1=`cat $tmpfile`
        \rm -f $tmpfile 2>/dev/null
fi

lockfile=$cldir/$1-$OARG1
#if [ -e $lockfile -a "$FLAG2" = "" ]; then
if [ -e $lockfile ]; then
        print "\nError: File $1 Revision $OARG1 already locked by another developer !!"
        aa=` ls -l $lockfile | awk '{print "Locking developers unix login name is = " $3}' `
        print $aa
        print "That developer should do scommit OR sunlock to release the lock"
        print " "
#       print "You can also use -F option to force edit the file even if"
#       print "the file is locked by another developer. But you must talk to"
```

```
#       print "other developer to work concurrently on this file."
#       print "For example - this option is useful if you work on a seperate"
#       print "C++ function in the file which does not interfere with other"
#       print "developer."
#       print " "
        exit
fi

# Get read-only copy now....
if [ ! -e $1 ]; then
        (
        cd $hme
        cvs -r checkout $fdname 1>/dev/null
        )
fi

# Check if file already exists....
if [ -f $1 ]; then
        user_perms=" "
        group_perms=" "
        other_perms=" "
        user_perms=`ls -l $1 | awk '{print $1 }' | cut -b3-3 `
        group_perms=`ls -l $1 | awk '{print $1 }' | cut -b6-6 `
        other_perms=`ls -l $1 | awk '{print $1 }' | cut -b9-9 `
        if [ "$user_perms" = "w" -o "$group_perms" = "w"  \
                -o "$other_perms" = "w" ]; then
                print "\nError: The file is writable. Aborting $cmdname ......"
                print "       You must backup, scommit or delete file and"
                print "       try $cmdname again\n"
                exit
        fi
        #print "\nNote: The file $1 is read-only."
        #print "Hence I am moving it to $1.$bkextn ....\n"
        \mv -f $1 $1.$bkextn
        chmod 444 $1.$bkextn
elif [ -d $1 ]; then
        print "\nError: $1 is a directory and NOT a file. Aborting $cmdname ....\n"
        exit
fi

# Create subshell
print "\nNow getting the file $1 from CVS repository ...\n"
(
cd $hme
#echo $fdname
# Use -A option to clear the sticky tag and to get
# the HEAD revision version
```

```
if [ "$FLAG1" = "" ]; then
        cvs -w checkout -A $fdname
else
        cvs -w checkout -A -$FLAG1 $OARG1 $fdname
fi
)

if [ -e $1 ]; then
        touch $lockfile
fi

#pwd

print "\nDone $cmdname"
#print "\nTip (Usage): $cmdname <filename> \n"
```

4.3 scommit

注意：***Korn shell /bin/ksh*** 在你從 ***Linux CD-ROM*** 安裝 ***pdksh*.rpm*** 時就會產生
請把他存成一般文字檔並改變存取權限 chmod a+rx

```
#!/bin/ksh
# CVS program scommit
# Program to commit the changes and check in the file into CVS

cmdname=`basename $0`

Usage()
{
        print "\nUsage: $cmdname [-r revision_number] <filename>"
        print "The options -r are optional "
        print "For example - "
        print " $cmdname -r 1.1 foo.cpp"
        print " $cmdname foo.cpp "
        print " "
}

# Command getopt will not supported in next major release.
# Use getopts instead.
while getopts r: ii
do
        case $ii in
        r) FLAG1=$ii; OARG1="$OPTARG";;
        ?) Usage; exit 2;;
        esac
done
```

```
shift ` expr $OPTIND - 1 `

#echo FLAG1 = $FLAG1 , OARG1 = $OARG1

if [ $# -lt 1 ]; then
        Usage
        exit 2
fi

if [ -d $1 ]; then
        Usage
        exit 2
fi

hme=` echo $HOME | cut -f1 -d' '  `
if [ "$hme" = "" ]; then
        print "\nError: \$HOME is not set!!\n"
        exit
fi

# Find sub-directory
cur_dir=`pwd`
#echo $cur_dir
len=${#hme}
len=$(($len + 2))
#echo $len
subdir=` echo $cur_dir | cut -b $len-2000 `
#echo $subdir
if [ "$subdir" = "" ]; then
        fdname=$1
else
        fdname=$subdir"/"$1
fi

# If file is already checked out by another user....
cvs_root=` echo $CVSROOT | cut -f1 -d' '  `
if [ "$cvs_root" = "" ]; then
        print "\nError: \$CVSROOT is not set!!\n"
        exit
fi
cldir=$CVSROOT/$subdir/Locks
mkdir $cldir 2>/dev/null

# Get the working revision number of the file....
# Use tmpfile as the arg cannot be set inside the sub-shell
tmpfile=$hme/sedit-lock.tmp
\rm -f $tmpfile 2>/dev/null
```

```
if [ "$FLAG1" = "" ]; then
        (
        cd $hme
        cvs status $fdname 2>/dev/null | grep "Working revision:" | awk '{print $3}' >$tmpfile
        )
        OARG1=`cat $tmpfile`
        \rm -f $tmpfile 2>/dev/null
fi

if [ "$OARG1" = "" ]; then
        print "The file $fdname is NEW, it is not in the CVS repository"
else
        lockfile=$cldir/$1-$OARG1
        if [ -e $lockfile ]; then
                # Check if this revision is owned by you...
                aa=` ls -l $lockfile | awk '{print $3}' `
                userid=`id | cut -d'(' -f2 | cut -d')' -f1 `
                if [ "$aa" != "$userid" ]; then
                        print " "
                        print "The file $fdname is NOT locked by you!!"
                        print "It is locked by unix user name $aa and your login name is $userid"
#                       print "If you are working concurrently with other developer"
#                       print "and you used -F option with sedit."
                        print "You need to wait untill other developer does scommit"
                        print "or sunlock"
                        print "Aborting the $cmdname ...."
                        print " "
                        exit 2
                fi
        else
                if [ -f $CVSROOT/$subdir/$1,v ]; then
                        print "You did not lock the file $fdname with sedit!!"
                        print "Aborting the $cmdname ...."
                        exit 2
                else
                        print "\nThe file $fdname does not exist in CVS repository yet!!"
                        print "You should have done sadd on $fdname ...."
                fi
        fi
fi

if [ -d $1 ]; then
        Usage
        exit 2
        # Do not allow directory commits for now ...
        #cvs commit
else
```

```
        cvs commit $1
        exit_status=$?
fi

if [ $exit_status -eq 0 ]; then
        print "\nDone $cmdname. $cmdname successful"
        #print "\nTip (Usage): $cmdname <filename/directory name>\n"
fi
```

4.4 supdate

注意：***Korn shell /bin/ksh*** 在你從 ***Linux CD-ROM*** 安裝 ***pdksh*.rpm*** 時就會產生
請把他存成一般文字檔並改變存取權限 chmod a+rx

```
#!/bin/ksh

# CVS program supdate
# Program to update the file from CVS read/write mode

cmdname=`basename $0`

if [ $# -lt 1 ]; then
        print "\nUsage: $cmdname <filename>"
        exit
fi

# Check if file already exists....
if [ $# -gt 0 -a  -f $1 ]; then
        user_perms=" "
        group_perms=" "
        other_perms=" "
        user_perms=`ls -l $1 | awk '{print $1 }' | cut -b3-3 `
        group_perms=`ls -l $1 | awk '{print $1 }' | cut -b6-6 `
        other_perms=`ls -l $1 | awk '{print $1 }' | cut -b9-9 `
        if [ "$user_perms" = "w" -o "$group_perms" = "w"  \
                -o "$other_perms" = "w" ]; then
                while :
                do
                        print "\n$cmdname will backup your working file "
                        print "$1 to $1.supdate_bak before doing any merges."
                        print "Are you sure you want the merge the changes from"
                        print -n "CVS repository to your working file ? <y/n> [n]: "
                        read ans
                        if [ "$ans" = "y" -o "$ans" = "Y" ]; then
                                if [ -f $1.supdate_bak ]; then
                                        print "\nWarning : File $1.supdate_bak already exists!!"
```

```
                                        print "Please examine the file $1.supdate_bak and delete it"
                                        print "and than re-try this $cmdname "
                                        print "Aborting $cmdname ...."
                                        exit
                                else
                                        cp $1 $1.supdate_bak
                                        break
                                fi
                        elif [ "$ans" = "n" -o "$ans" = "N" -o "$ans" = "" -o "$ans" = " " ]; then
                                exit
                        fi
                done
        fi
fi

if [ -d $1 ]; then
        print "\nDirectory update is disabled as cvs update"
        print "merges the changes from repository to your working directory"
        print "So give the filename to update - as shown below: "
        print " Usage: $cmdname <filename>"
        exit
#       cvs update
else
        cvs update $1
fi

print "\nDone $cmdname. $cmdname successful"
#print "\nTip (Usage): $cmdname <filename/directory name>\n"
```

4.5 sunlock

注意 : ***Korn shell /bin/ksh*** 在你從 ***Linux CD-ROM*** 安裝 ***pdksh*.rpm*** 時就會產生
請把他存成一般文字檔並改變存取權限 chmod a+rx.

```
#!/bin/ksh
# CVS program sunlock
# Program to unlock the file to release the lock done by sedit

cmdname=`basename $0`

Usage()
{
        print "\nUsage: $cmdname [-r revision_number] <filename>"
        print " The options -r is optional "
        print "For example - "
        print " $cmdname -r 1.1 foo.cpp"
```

```
        print " $cmdname foo.cpp "
        print " "
}

# Command getopt will not supported in next major release.
# Use getopts instead.
while getopts r: ii
do
        case $ii in
        r) FLAG1=$ii; OARG1="$OPTARG";;
        ?) Usage; exit 2;;
        esac
done
shift ‘ expr $OPTIND - 1 ‘

if [ $# -lt 1 ]; then
        Usage
        exit
fi

hme=‘ echo $HOME | cut -f1 -d’ ’  ‘
if [ "$hme" = "" ]; then
        print "\nError: \$HOME is not set!!\n"
        exit
fi

cur_dir=‘pwd‘
#echo $cur_dir

len=${#hme}
len=$(($len + 2))
#echo $len

subdir=‘ echo $cur_dir | cut -b $len-2000 ‘
#echo $subdir

if [ "$subdir" = "" ]; then
        fdname=$1
else
        fdname=$subdir"/"$1
fi

# If file is already checked out by another user....
cvs_root=‘ echo $CVSROOT | cut -f1 -d’ ’  ‘
if [ "$cvs_root" = "" ]; then
        print "\nError: \$CVSROOT is not set!!\n"
        exit
```

```
fi
cldir=$CVSROOT/$subdir/Locks
rcsfile=$CVSROOT/$subdir/$1,v
#echo $rcsfile

if [ ! -e $rcsfile ]; then
        print "\nError: File $1 does not exist in CVS repository!!\n"
        exit
fi

# Get the tip revision number of the file....
# Use tmpfile as the arg cannot be set inside the sub-shell
tmpfile=$hme/sedit-lock.tmp
\rm -f $tmpfile 2>/dev/null
if [ "$FLAG1" = "" ]; then
        (
        cd $hme
        cvs log $fdname | head -6 | grep head: | awk '{print $2}' > $tmpfile
        )
        OARG1=`cat $tmpfile`
        \rm -f $tmpfile 2>/dev/null
fi

lockfile=$cldir/$1-$OARG1
#echo lockfile is : $lockfile
if [ ! -e $lockfile ]; then
        print "\nFile $1 revision $OARG1 is NOT locked by anyone"
        print " "
        exit
fi

ans=""
while :
do
        print "\n\n***************************************************"
        print "WARNING: $cmdname will release lock and enable other"
        print "         developers to edit the file. It is advisable"
        print "         to save your changes with scommit command"
        print "***************************************************"
        print -n "\nAre you sure you want to unlock the file <y/n>? [n]: "
        read ans
        if [ "$ans" = "" -o "$ans" = " " -o "$ans" = "n" -o "$ans" = "N" ]; then
                print "\nAborting $cmdname ...."
                exit
        fi
        if [ "$ans" = "y" -o "$ans" = "Y" ]; then
                print "\n\n\n\n\n "
```

```
                print "CAUTION: You may lose all the changes made to file!!"
                print -n "Do you really want to unlock the file <y/n>? [n]: "
                read ans
                if [ "$ans" = "y" -o "$ans" = "Y" ]; then
                        break
                else
                        exit
                fi
        else
                print "\n\nWrong entry. Try again..."
                sleep 1
        fi
done

if [ -e $lockfile ]; then
        \rm -f $lockfile
        print "\nDone $cmdname"
else
        print "\nFile $1 is NOT locked by anyone"
        print " "
fi
```

4.6 slist

注意：***Korn shell /bin/ksh*** 在你從 ***Linux CD-ROM*** 安裝 ***pdksh*.rpm*** 時就會產生

請把他存成一般文字檔並改變存取權限 chmod a+rx

```
#!/bin/ksh

# CVS program slist
# Program to list all edited source files from CVS

#cmdname=`basename $0`

#echo "no of params : " $#
#echo "all args : " $@

recurse_flag=""

if [ "$1" = "" ]; then
        dir=.
        recurse_flag=""
else
        dir=$@
        recurse_flag=" -prune "
fi
```

```
FOUT=slist_temporary_file.out

\rm -f $FOUT

find $dir  $recurse_flag -type f -exec ls -ltr {} \; \
| grep -v "/CVS/" \
| grep ^\-rw \
| grep -v \\.o \
| grep -v \\.log \
| grep -v \\.out \
| grep -v \\.pid \
| awk '{ if ($NF != "tags") print $0 }' \
| awk '{ if ($NF != "a.out") print $0 }' \
| awk '{ if ($NF != "core") print $0 }' \
| awk '{ print $NF }' > $FOUT

aa=`cat $FOUT`
\rm -f $FOUT

for ii in $aa ; do
        ftype=" "
        ftype=`file $ii | awk '{print $2 }' `

        # find . -type f -exec file {} \;
        # 1)ELF 2)commands 3)[nt]roff, 4)c 5)English  6)executable
        # 7)ascii 8)current 9)empty
        # Binaries are ELF, lib.a are current
        #
        if [ "$ftype" = "ascii" -o "$ftype" = "commands" \
                -o "$ftype" = "[nt]roff," -o "$ftype" = "c" -o "$ftype" = "data" \
                -o "$ftype" = "English" -o "$ftype" = "executable" ]; then
                pcfile=` echo $ii | cut -d'.' -f1`
                pcfile=${pcfile}".pc"
                if [ ! -f $pcfile ]; then
                        ls -l $ii
                else
                        if [ "$ii" = "$pcfile" ]; then
                                ls -l $ii
                        fi
                fi
        fi
done;

#| grep -v ^\-rwx \

#ls -l | grep ^\-rw | grep -v \\.o
```

```
#ls -l | grep ^\-rw | grep -v \\.o | awk '{ if ($NF != "tags") print $0 }'
#ls -l | grep ^\-rw | grep -v ^\-rwx | grep -v \\.o |  awk '{ if ($NF != "tags") print $0 }' | awk '{ if ($NF != "core") print $0 }'

#print "\nDone $cmdname. $cmdname successful"
#print "\nTip (Usage): $cmdname <filename>\n"
```

4.7 sinfo

注意：***Korn shell /bin/ksh*** 在你從***Linux CD-ROM*** 安裝 ***pdksh*.rpm*** 時就會產生
請把他存成一般文字檔並改變存取權限 chmod a+rx

```
#!/bin/ksh

# CVS program sinfo
# Program to get the status of files in working directory

cmdname=`basename $0`

if [ $# -lt 1 ]; then
        print "\nUsage: $cmdname [file/directory name] "
        print "For example - "
        print " $cmdname foo.cpp"
        print " $cmdname some_directory "
        print " "
        exit
fi

hme=` echo $HOME | cut -f1 -d' '  `
if [ "$hme" = "" ]; then
        print "\nError: \$HOME is not set!!\n"
        exit
fi

tmpfile=$hme/cvs_sinfo.tmp
rm -f $tmpfile

cur_dir=`pwd`
#echo $cur_dir

len=${#hme}
len=$(($len + 2))
#echo $len

subdir=` echo $cur_dir | cut -b $len-2000 `
#echo $subdir
```

```
if [ "$subdir" = "" ]; then
        fdname=$1
else
        fdname=$subdir"/"$1
fi

# Create subshell
if [ -f $1 ]; then
        (
        cd $hme
        clear
        cvs status $fdname
        )
elif [ -d $1 ]; then
        (
        cd $hme
        clear
        echo "  " >> $tmpfile
        echo "  ****************************************" >> $tmpfile
        echo "        Overall Status of Directory" >> $tmpfile
        echo "  ****************************************" >> $tmpfile
        cvs release $fdname 1>>$tmpfile 2>>$tmpfile << EOF
Y
EOF
        echo "\n   -------------------------------\n" >> $tmpfile

        aa=`cat $tmpfile | grep ^"M " | awk '{print $2}' `
        for ii in $aa
        do
                jj="(cd $hme; cvs status $subdir/$ii );"
                echo $jj | /bin/sh  \
                        | grep -v Sticky | awk '{if (NF != 0) print $0}' \
                        1>>$tmpfile 2>>$tmpfile
        done

        cat $tmpfile | grep -v ^? | grep -v "Are you sure you want to release" \
        | less
        rm -f $tmpfile
        )
else
        print "\nArgument $1 if not a file or directory"
        exit
fi
```

4.8 slog

注意：***Korn shell /bin/ksh*** 在你從***Linux CD-ROM*** 安裝 ***pdksh*.rpm*** 時就會產生
請把他存成一般文字檔並改變存取權限 chmod a+rx

```
#!/bin/ksh

# CVS program slog
# Program to list history of the file in CVS

cmdname=`basename $0`

if [ $# -lt 1 ]; then
        print "\nUsage: $cmdname <filename> \n"
        exit
fi

# Check if file does not exist....
if [ ! -f $1 ]; then
        print "\nError: $1 is NOT a file. Aborting $cmdname ......"
        exit
fi

cvs log $1 | /usr/local/bin/less

print "\nDone $cmdname. $cmdname successful"
#print "\nTip (Usage): $cmdname <filename>\n"
```

4.9 sdif

注意：***Korn shell /bin/ksh*** 在你從***Linux CD-ROM*** 安裝 ***pdksh*.rpm*** 時就會產生
請把他存成一般文字檔並改變存取權限 chmod a+rx

```
#!/bin/ksh

# CVS program sdif
# Program to see difference of the working file with CVS copy

cmdname=`basename $0`

Usage()
{
        print "\nUsage: $cmdname <filename> "
        print "$cmdname -r<rev1> -r<rev2> <filename> \n"
        exit
```

```
}
FLAG1=""
FLAG2=""
OARG1=""
OARG2=""
# Command getopt will not supported in next major release.
# Use getopts instead.
while getopts r:r: ii
do
        case $ii in
        r)
                if [ "$FLAG1" = "" ]; then
                        FLAG1=$ii;
                        OARG1="$OPTARG"
                else
                        FLAG2=$ii;
                        OARG2="$OPTARG"
                fi
                ;;
        ?) Usage; exit 2;;
        esac
done
shift ` expr $OPTIND - 1 `

if [ "$FLAG2" = "" ]; then
        FLAG2=r
        OARG2=HEAD
fi

if [ "$FLAG1" = "" ]; then
        cvs diff -r HEAD $1 | less
else
        cvs diff -$FLAG1 $OARG1 -$FLAG2 $OARG2 $1 | less
fi
```

4.10 sadd

注意：***Korn shell /bin/ksh*** 在你從 ***Linux CD-ROM*** 安裝 ***pdksh*.rpm*** 時就會產生
請把他存成一般文字檔並改變存取權限 chmod a+rx

```
#!/bin/ksh

# test
# CVS program sadd
# Program to add the file to CVS
```

```
cmdname=`basename $0`
if [ $# -lt 1 ]; then
        print "\nUsage: $cmdname <filename/directory> \n"
        exit
fi

# Check if file exists ....
if [ -f $1 ]; then
        cvs add $1
        exit
fi

if [ ! -d $1 ]; then
        print "\nArgument $1 is not a file and not a directory!"
        print "Usage: $cmdname <filename/directory> \n"
        exit
fi

# Argument is a directory name .....
hme=` echo $HOME | cut -f1 -d' '  `
if [ "$hme" = "" ]; then
        print "\nError: \$HOME is not set!!\n"
        exit
fi

cur_dir=`pwd`
len=${#hme}
len=$(($len + 2))
subdir=` echo $cur_dir | cut -b $len-2000 `

if [ "$subdir" = "" ]; then
        if [ -d $CVSROOT/$1 ]; then
                print "\nDirectory $1 already exists in CVSROOT"
                exit
        else
                # You are adding at root directory $CVSROOT
                if [ "$2" = "" -o "$3" = "" ]; then
                        print "\nUsage: $cmdname <directory> <vendor tag> <release tag>"
                        print "For example - "
                        print " $cmdname foo_directory V_1_0 R_1_0"
                        exit
                else
                        (
                        cd $1;
                        cvs import $1 $2 $3
                        )
                fi
```

```
        fi
else
        # If current directory exists in CVS...
        if [ -d $CVSROOT/$subdir ]; then
                if [ -d $CVSROOT/$subdir/$1 ]; then
                        print "\nDirectory $1 already in CVS repository!"
                else
                        cvs add $1
                fi
        else
                print "\nSub-directory $subdir does not exist in CVS"
                print "You need to first add $subdir to CVS"
                exit
        fi
fi
```

4.11 sdelete

注意：***Korn shell /bin/ksh*** 在你從 ***Linux CD-ROM*** 安裝 ***pdksh*.rpm*** 時就會產生

請把他存成一般文字檔並改變存取權限 chmod a+rx

```
#!/bin/ksh

# CVS program sdelete
# Program to delete the file from CVS

cmdname=`basename $0`

if [ $# -lt 1 ]; then
        print "\nUsage: $cmdname <filename> \n"
        exit
fi

# Check if file does not exist....
if [ ! -f $1 ]; then
        # Try to get the file from CVS
        sget $1
        if [ ! -f $1 ]; then
                print "\nError: $1 does NOT exist in CVS repository. Aborting $cmdname ......"
                exit
        fi
fi

bkextn=cvs_sdelete_safety_backup
\mv -f $1 $1.$bkextn
```

```
cvs remove $1

print "\nsdelete command removes the file from CVS repository"
print "and archives the file in CVS Attic directory. In case"
print "you need this file in future than contact your CVS administrator"
print " "

print "\nDone $cmdname. $cmdname successful"
#print "\nTip (Usage): $cmdname <filename>\n"
\mv -f $1.$bkextn $1
```

4.12 sfreeze

注意：***Korn shell /bin/ksh*** 在你從 ***Linux CD-ROM*** 安裝 ***pdksh*.rpm*** 時就會產生
請把他存成一般文字檔並改變存取權限 chmod a+rx

```
#!/bin/ksh

# CVS program sfreeze
# Program to freeze and cut out the release of source tree from CVS

cmdname=`basename $0`

Usage()
{
        print "\nUsage: $cmdname symbolic_tag <directory name> "

        print "\nFor example :- "
        print "   cd \$HOME"
        print "   $cmdname  REVISION_1   mesa"
        print "To see the list of revisons do -"
        print "slog <filename> and see the symbolic name and do -"
        print "cvs history -T"

        print "\nTo create a branch off-shoot from main trunk, use"
        print "the -b and -r options which makes the tag a branch tag. This is"
        print "useful for creating a patch to previously released software"
        print "For example :- "
        print "   cd \$HOME"
        print "   cvs rtag -b -r REVISION_1   REVISION_1_1   mesa"
        print " "

#       print "\nTag info is located at \$CVSROOT/CVSROOT/taginfo,v"
#       print "You can do - cd $HOME; sget CVSROOT"
#       print "to see this file"
        exit
```

```
}

# Command getopt will not supported in next major release.
# Use getopts instead.
#while getopts r: ii
#do
#       case $ii in
#       r) FLAG1=$ii; OARG1="$OPTARG";;
#       ?) Usage; exit 2;;
#       esac
#done
#shift ‘ expr $OPTIND - 1 ‘

#echo FLAG1 = $FLAG1 , OARG1 = $OARG1

if [ $# -lt 2 ]; then
        Usage
fi

if [ ! -d $2 ]; then
        print "\nError: Second argument $2 is not a directory!"
        print "       Aborting $cmdname...."
        print " "
        exit
fi

# cvs rtag symbolic_tag <directory name>
cvs rtag $1 $2

print "\nDone $cmdname. $cmdname successful"
```

4.13 saddtree

注意：***Korn shell /bin/ksh*** 在你從 ***Linux CD-ROM*** 安裝 ***pdksh*.rpm*** 時就會產生
請把他存成一般文字檔並改變存取權限 chmod a+rx。

```
#!/bin/ksh

#################################################################
# Sample Program to checkin a directory tree (let's say SAMP) into CVS
# Note that if SAMP directory is not in CVS than you would use sadd
# command and -
#       cd SAMP; cvs import SAMP V_1_0 R_1_0
# After running this program do -
#       cd $HOME/foo/SAMP
#       cvs import foo/SAMP V1_0 Rev_1_0
```

```
#################################################################

hme=` echo $HOME | cut -f1 -d' '  `
if [ "$hme" = "" ]; then
        print "\nError: \$HOME is not set!!\n"
        exit
fi
sampdir=$hme/foo/SAMP

check_out_files()
{
        # Now check out the files
        tmp2f=$hme/tmp2.baksamp.sh

        cd $hme
        \rm -rf foo/SAMP
        cvs -w checkout foo/SAMP
        cd $hme/foo
        find SAMP -type f -print > $tmp2f

        cd $hme
        for ii in `cat $tmp2f`
        do
                iidir=`dirname $ii`
                iifile=`basename $ii`
                if [ "$iifile" = "Root" -o "$iifile" = "Repository" -o "$iifile" = "Entries" ]; then
                        continue
                fi
                jjdir=` echo $iidir | cut  -d'/' -f2-1000 `

                cp $hme/foo/SAMP.tobe/$jjdir/$iifile  $hme/foo/$iidir/$iifile
                echo "cp $hme/foo/SAMP.tobe/$jjdir/$iifile  $hme/foo/$iidir/$iifile "

                cvs add  foo/$iidir/$iifile
        done

        print
        print "================================================="
        print " Now run cvs commit foo/SAMP"
        print " After commit. Do - "
        print "         cd foo; rm -rf SAMP and "
        print " get fresh copy, sget SAMP"
        print " Verify with slog filename.samp to see new revision"
        print "================================================="
        print
}
```

check_out_files

5 CVS 的其他文件

在 unix 提示符號下，請打 -

1. cvs –help
2. cvs –help-options
3. cvs –help-commands
4. cvs -H checkout
5. cvs -H commit
6. man cvs
7. man tkcvs
8. 網站 <http://www.cyclic.com>
9. 網站 <http://www.loria.fr/~molli/cvs-index.html>
10. (譯註:或者在 unix 提示符號下打 info cvs 也可得到一樣的訊息)
11. (譯註:這篇也不錯簡單明瞭) <http://www.csc.calpoly.edu/~dbutler/tutorials/winter96/cvs/>

tkcvs <http://www.tkcvs.org> 是 CVS 的 Tcl/Tk GUI 介面。這裡也有線上求助。

- cd $HOME/src/foo.cpp
- tkcvs
- 在 foo.cpp 上點一下
- 在 'spectacle Icon' 旁邊的 'Revision Log Icon' 點一下。
- 這將會顯示一個 Tree 組織的圖在視窗裡。然後在文字 '1.3' 上用滑鼠的右鍵點一下還有 '1.1' 滑鼠的左鍵點一下，然後再點一下 "Diff"。這樣將會顯示兩個視窗出來!!
- 在 "Next" 上點一下將會顯示更多版本'1.3' 與 '1.1' 的 diffs。 請按 "Center" 將文字對齊置中。 (譯註:這邊原文好像有脫誤)

這裡也有 Windows 95 用的 CVS 喔，叫 WinCVS。 <http://www.wincvs.org> WinCVS 可以用在 Samba 系統上喔 - <http://www.samba.org>

基本重要的命令 -

- cvs checkout <filename >
- cvs update <filename>

- cvs add <file, ..>
- cvs remove <file, ..>
- cvs commit <file>
- cvs status <filename>
- cvs log <filename>
- cvs diff -r1.4 -r1.5 <filename> 這行指令將會輸出檔案 filename 1.4 版和 1.5 版的差異在哪裡。

6 Emacs 編輯器

Emacs 是非常強大的編輯器而且還支援了 CVS/RCS - 尤其是對於改版後的合併和比較。 Emacs的主要網站在 <http://www.emacs.org>.

7 問題反應系統 (Problem Reporting System)

伴隨著 CVS 的使用，你可能會想要用計劃追蹤系統(Project Tracking system)或問題反應系統(Problem Reporting System)。每一個軟體計劃需要問題反應系統來作 bugs 的回報與追蹤，並且把相關負責部分，分配給不同的程式設計師。

想知道計劃追蹤系統(Project Tracking system)請看這個網站 <http://www.stonekeep.com>。

8 這份文件的其他檔案格式

這份文件用了將近 10 個不同的檔案格式來發行 - DVI, Postscript, Latex, LyX, GNU-info, HTML, RTF(Rich Text Format), Plain-text, Unix man pages and SGML。

- 你可以在下面的 Web 獲得單一的 HTML, DVI, Postscript or SGML 型式的 tar.gz 檔案， <ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/other-formats/> 或者 <ftp://metalab.unc.edu/pub/Linux/docs/HOWTO/other-formats/>
- Plain text 格式的在: <ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO>，或者 <ftp://metalab.unc.edu/pub/Linux/docs/HOWTO>
- 翻譯成其他國家語言像是法文，德文，西班牙文，中文，日文等等可在下面網址找得到。 <ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO> 或者 <ftp://metalab.unc.edu/pub/Linux/docs/HOWTO> 對於任何想翻譯這份文件到其他語言是都歡迎的。

這份文件是用一種叫 "SGML" 的工具寫成的。你可以從下列網址得到 - <http://www.xs4all.nl/~cg/sgmltools/> 編譯原始碼後，你就會得到下列命令，像是

- sgml2html cvs-rcs-howto.sgml (產生 html 檔案)
- sgml2rtf cvs-rcs-howto.sgml (產生 RTF 檔案)
- sgml2latex cvs-rcs-howto.sgml (產生 latex 檔案)

這份文件是位於 -

- <http://sunsite.unc.edu/LDP/HOWTO/CVS-RCS-HOWTO.html>

你也可以在下面映射網站找到這份文件 -

- <http://www.caldera.com/LDP/HOWTO/CVS-RCS-HOWTO.html>
- <http://www.WGS.com/LDP/HOWTO/CVS-RCS-HOWTO.html>
- <http://www.cc.gatech.edu/linux/LDP/HOWTO/CVS-RCS-HOWTO.html>
- <http://www.redhat.com/linux-info/ldp/HOWTO/CVS-RCS-HOWTO.html>
- 其他比較靠近你的映射網站 <http://sunsite.unc.edu/LDP/hmirrors.html> 選擇其中一個網站並到這個目錄 /LDP/HOWTO/CVS-RCS-HOWTO.html

想看 dvi 格式的文件，請用 xdvi。若原本沒安裝，可在 Redhat Linux 下的 tetex-xdvi*.rpm 套件中找得到。要不然可以經由桌面上的ControlPanel | Applications | Publishing | TeX menu buttons找到他。

```
想看 dvi 格式的文件，請下命令 -
        xdvi -geometry 80x90 howto.dvi'
並且用滑鼠調整視窗大小。請看 xdvi 的線上求助 man xdvi。
使用方向鍵， Page Up, Page Down 來檢視文件。你也可以用
'f', 'd', 'u', 'c', 'l', 'r', 'p', 'n'來移動。
要關掉專家模式 menu 請按 x。
```

你可以用軟體 'gv' (ghostview) 或 'ghostscript' 來讀 postscript 檔案。

在 Redhat Linux 下，ghostscript 在 ghostscript*.rpm 套件裡， gv 在 gv*.rpm 套件裡， 如果你已經裝了這個應用程式，可以透過 ControlPanel | Applications | Graphics menu 裡找到， gv 是比 ghostscript 更友善使用。Ghostscript 和 gv 也可以在其他作業平台如 OS/2, windows 95 和 NT 下使用。

```
要讀 postscript 文件，請下這個命令 -
        gv howto.ps

用 ghostscript 請打 -
        ghostscript howto.ps
```

你可以用 Netscape Navigator, Microsoft Internet explorer, Redhat Baron Web browser 或其他瀏覽器來讀 HTML 格式的文件。

你可以用一個 latex 的 "X-Windows" 介面軟體叫做 LyX 的，來讀 latex 或 LyX 的輸出。

9 版權