APT HOWTO (Obsolete Documentation)

Gustavo Noronha Silva <kov@debian.org>

劉浩 <iamlyoo@163.com>

1.8.10.4 - 2005年3月

摘要

這篇文件試圖讓使用者對於Debian包管理工具APT的工作方式有一個很好的理解。 它的目標是讓新的Debian使用者更容易上手，也讓那些想幫助那些想更深入理解 如何管理他們的系統的人。它爲Debian項目而編寫，目的是提高此發行版對它 的使用者的支援水平。

版權聲明

Contents

Chapter 1

導言

最初只有.tar.gz的打包檔案，使用者必須編譯每個他想在GNU/Linux上執行的軟體。使用者們普遍認爲系統很有必要提供一種方法來管理這些安裝在機器上的套裝軟體，當Debian誕生時，這樣一個管理工具也就應運而生，它被命名爲dpkg。從而著名的“package”概念第一次出現在GNU/Linux系統中，稍後Red Hat才決定開發自己的“rpm”包管理系統。

很快一個新的問題難倒了GNU/Linux製作者，他們需要一個快速、實用、高效的方法來安裝套裝軟體，當套裝軟體更新時，這個工具應該能自動管理關聯檔案和維護已有組態檔案。Debian在次率先解決了這個問題，APT(Advanced Packaging Tool)誕生了。APT後來還被Conectiva改造用來管理rpm，並被其它Linux發行版本采用爲它們的套裝軟體管理工具。

本文件不打算講解apt-rpm相關知識，因爲Conectiva移植的APT已很有名了，不過提供有關這部分的補充文件還是歡迎的。

本文件是基於Debian下一個發行版Sarge的。

Chapter 2

基礎設定

2.1 /etc/apt/sources.list檔案

作爲操作的一部分，APT使用一個檔案列出可獲得套裝軟體的鏡像站台位址，這個文 件就是/etc/apt/sources.list。

檔案中的各項資訊通常按如下格式列出：

```
deb http://host/debian distribution section1 section2 section3
deb-src http://host/debian distribution section1 section2 section3
```

當然，上面所列的位址項都是假設的且不應該使用它們。每行的第一個單字deb或deb-src描述了檔案類型：目錄中包含的是二進位套裝軟體(deb)， 即我們通常使用的已編譯好的套裝軟體；或包含的是來源碼包(deb-src)，來源碼 包包含來源程序編碼、Debian控制檔案(.dsc)和“Debian化”這個程式所做變更的 記錄檔案diff.gz。

在Debian預設的sources.list中通常是如下內容：

```
# See sources.list(5) for more information, especially
# Remember that you can only use http, ftp or file URIs
# CDROMs are managed through the apt-cdrom tool.
deb http://http.us.debian.org/debian stable main contrib non-free
deb http://non-us.debian.org/debian-non-US stable/non-US main contrib non-fr
deb http://security.debian.org stable/updates main contrib non-free

# Uncomment if you want the apt-get source function to work
#deb-src http://http.us.debian.org/debian stable main contrib non-free
#deb-src http://non-us.debian.org/debian-non-US stable/non-US main contrib n
```

這些是Debian基本安裝所需的套裝軟體來來源位址。第一個deb行指向官方正 式套裝軟體來來源，第二個行指向non-US套裝軟體來來源，第三行指向Debian安全補綴更新 包來來源。

最後兩行被註解掉了(在句首加“#”)，所以apt-get將忽略它們。這些是deb-src行指向Debian來源碼包來來源，如果你常下載程序來源碼來測試或重編譯，可取消對 它們的註解。

/etc/apt/sources.list檔案可包含多種類型的位址，APT知道如何處理這 些不同的位址類型：http、ftp、file(本地檔案，例 如：一個加載了ISO9600檔案系統的目錄)和ssh。

別忘了在修改完/etc/apt/sources.list檔案後執行apt-get使 變更生效。你必須完成這個步驟，以便讓APT從你指定的地方獲得新的套裝軟體列表。

2.2 如何在本地使用APT

有時你硬碟上有許多.deb套裝軟體，你會希望通過APT來安裝它們，以便讓它去處理軟 件包間複雜的依賴關係。

想這麼做，就建一個目錄，將所有你想要安裝的.deb檔案放入其中。例如：

```
# mkdir /root/debs
```

你可以使用一個override檔案直接去修改套裝軟體中控制檔案中的定義，使 之符合你的軟體儲藏庫管理規則。在這個覆寫檔案中，你可能希望定義一些選項來覆 蓋那些套裝軟體的定義，如下所示：

```
package priority section
```

package是套裝軟體的名稱，priority有三個層級low、medium或high，section是軟體 包所屬的section，多載檔案可任意命名，檔案名稱將在接下來的步驟中做爲參數傳遞給dpkg-scanpackages。如果你不想寫多載檔案，只需在調用dpkg-scanpackages時使用/dev/null就行了。

仍是在/root目錄下執行：

```
# dpkg-scanpackages debs file | gzip > debs/Packages.gz
```

在上述的命令行中，*file*爲override檔案，命令產生一個Packages.gz檔案，它包含了APT所需的各種套裝軟體資訊。最後，如果要使用 這些套裝軟體，加上：

```
deb file:/root debs/
```

完成了上面的工作，就可以通常那樣使用APT命令操作這些套裝軟體了。你可以使用同 樣的方法產生一個來源碼庫，但請記住你需要將.orig.tar.gz檔案、.dsc檔案和.diff.gz檔案包含在目錄中，同時必須產生Source.gz檔案而不是Packages.gz檔案。所使用的命令也不相同， 要使用dpkg-scansources，命令如下所示：

```
# dpkg-scansources debs | gzip > debs/Sources.gz
```

Notice that `dpkg-scansources` doesn't need an `override` file. The sources.list's line is:

```
deb-src file:/root debs/
```

2.3 選擇最佳鏡像發佈站台加入source.list檔案：netselect，netselect-apt

一個新使用者經常問到的問題："該將哪個Debian鏡像發佈站台加入`source.list`檔案？"。有很多方法來選擇鏡像發佈站台，專家們可能會寫一個腳本去測試不同站台的ping時間。不過其實有一個程序可以幫你：**netselect**。

要安裝netselect，通常使用：

```
# apt-get install netselect
```

不帶參數執行它時會顯示它的幫助資訊。執行它時加上以空格分隔的鏡像主機列表，它會傳回一個分值和列表中的一個主電腦名稱。這個分值通過評估ping時間和hops數(一個網路請求報文到達目標主機所經過的轉發主機的個數)得出，它與鏡像站台預計下載速度成反比(數值越小越好)。傳回的主電腦名稱是主機列表中得分最低的那個(檢視列表中所以主機的得分情況可使用-vv選項）。看出下的例子：

```
# netselect ftp.debian.org http.us.debian.org ftp.at.debian.org download.une
  365 ftp.debian.org.br
#
```

它表示，在netselect後列出的所有主機中，`ftp.debian.org.br`是下載速度最快的主機，其得分爲365。(注意！！這是在我電腦上的測試結果，不同的網路節點网速會大不相同，所以這個分值不一定適用於其它電腦)

現在將netselect找到的連線速度最快的鏡像站台加入`/etc/apt/sources.list`檔案(參考'/etc/apt/sources.list檔案' 3)並按照'套裝軟體管理' 7中的技巧來做。

注意：鏡像站台列表通常包含在檔案 `http://www.debian.org/mirror/mirrors_full`中。

從0.3.ds1版開始，netselect來源碼包中包含了**netselect-apt**二進制包，它使上述操作自動完成。只需將發佈目錄樹做爲參數(預設爲stable)輸入，`sources.list`檔案就會產生速度最快的main和non-US鏡像站台列表，並保存在目前目錄下。下面的例子產生一個包含stable發佈鏡像站台列表的sources.list：

```
# ls sources.list
ls: sources.list: File or directory not found
```

```
# netselect-apt stable
(...)
# ls -l sources.list
sources.list
#
```

記住：sources.list產生在目前目錄下，必須將其移 至/etc/apt目錄。

接著，按照'套裝軟體管理' 7中的技巧來做。

2.4 將CD-ROM加入source.list檔案

如果你用APT從CD-ROM上安裝及升級套裝軟體，你可以將它加入到sources.list檔案中。完成該操作，可使用apt-cdrom程序：

```
# apt-cdrom add
```

將Debian光碟放入光碟機，它將加載光碟目錄，並在光碟上尋找套裝軟體資訊。如果你 的光碟機需要額外設定，可使用以下選項：

```
-h            - program help
-d directory - CD-ROM mount point
-r            - Rename a recognized CD-ROM
-m            - No mounting
-f            - Fast mode, don't check package files
-a            - Thorough scan mode
```

例如：

```
# apt-cdrom -d /home/kov/mycdrom add
```

你還可以掃瞄一張光碟，但不將其加入列表：

```
# apt-cdrom ident
```

注意，只有當你在系統的/etc/fstab中正確設定了光碟機後，這個程式才會工作。

Chapter 3

套裝軟體管理

3.1 更新可用套裝軟體列表

套裝軟體管理系統使用一個私有資料庫追蹤列表中套裝軟體的目前狀態：已安裝、未安裝或可安裝。apt-get通過該資料庫來確定如何安裝使用者想用的軟體包以及正常執行該套裝軟體所必須的其它關聯包。

你可以使用apt-get update來更新資料庫列表。這個命令將掃瞄/etc/apt/sources.list檔案中所指路徑中的套裝軟體列表檔案。有關該列表檔案的更多資訊請參考'/etc/apt/sources.list檔案'3。

定時執行這個程序是個好主意，它將使你和你的系統獲得最新的套裝軟體更新和安全更新等資訊。

3.2 安裝套裝軟體

現在，終於到了你一直期待的階段！準備好了sources.list和最新版的的可用軟體包，你所需做的就是執行apt-get來安裝你渴望已久的軟體了。例如，你可以這樣：

```
# apt-get install xchat
```

APT會掃瞄它的資料庫找到最新的版本的套裝軟體，並將它從sources.list中所指的地方下載到本地。如果該套裝軟體需要其它套裝軟體才能正常執行——如本例一樣——APT會做關聯性檢查並自動安裝所關聯套裝軟體。如下所示：

```
# apt-get install nautilus
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  bonobo libmedusa0 libnautilus0
```

```
The following NEW packages will be installed:
  bonobo libmedusa0 libnautilus0 nautilus
0 packages upgraded, 4 newly installed, 0 to remove and 1  not upgraded.
Need to get 8329kB of archives. After unpacking 17.2MB will be used.
Do you want to continue? [Y/n]
```

nautilus套裝軟體需要參照共享函式庫，因此APT會從鏡像來源處下載相關共 享函式庫，如果你在apt-get命令行中手動指定了這些共享函式庫的名稱， APT不會詢問你是否要繼續；它會自動認爲你希望安裝所有這些套裝軟體。

也就是說APT只會在安裝那些沒有在命令行中指定的套裝軟體時提示確認。

下列apt-get選項也許對你有用：

```
-h   這個幫助資訊
-d   只下載——不安裝或解壓檔案
-f   即便完整性檢查失敗了仍然繼續
-s   不做什麼。只是按順序類比
-y   對於所有問題都假定爲Yes，不詢問
-u   顯示一系列已經將要更新的包
```

可以用一條命令安裝多個套裝軟體。包檔案從網路上下載到本地/var/cache/apt/archives目錄，稍後再安裝。

你可以用同樣的命令行刪除指定套裝軟體，只需在套裝軟體名稱後緊跟一個“-”，如下所示：

```
# apt-get install nautilus gnome-panel-
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  bonobo libmedusa0 libnautilus0
The following packages will be REMOVED:
  gnome-applets gnome-panel gnome-panel-data gnome-session
The following NEW packages will be installed:
  bonobo libmedusa0 libnautilus0 nautilus
0 packages upgraded, 4 newly installed, 4 to remove and 1  not upgraded.
Need to get 8329kB of archives. After unpacking 2594kB will be used.
Do you want to continue? [Y/n]
```

參考‘移除套裝軟體’9一節以獲得更多的關於刪除套裝軟體的資訊。

假如你不小心損壞了已安裝的套裝軟體而想修復它，或是只有只有想重新安裝套裝軟體中某 些檔案的最新版本，這是可以做到的，你可以用如下的--reinstall選項：

```
# apt-get --reinstall install gdm
Reading Package Lists... Done
```

```
Building Dependency Tree... Done
0 packages upgraded, 0 newly installed, 1 reinstalled, 0 to remove and 1  no
Need to get 0B/182kB of archives. After unpacking 0B will be used.
Do you want to continue? [Y/n]
```

3.3 移除套裝軟體

如果你不再使用某些套裝軟體，你可以用APT將其從系統中刪除。要刪除套裝軟體只需 輸入：apt-get remove package。如下所示：

```
# apt-get remove gnome-panel
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
  gnome-applets gnome-panel gnome-panel-data gnome-session
0 packages upgraded, 0 newly installed, 4 to remove and 1  not upgraded.
Need to get 0B of archives. After unpacking 14.6MB will be freed.
Do you want to continue? [Y/n]
```

由上例可知，APT會關注那些與被刪除的套裝軟體有依賴關係的套裝軟體。使用APT刪除 一個套裝軟體將會連帶刪除那些與該套裝軟體有依賴關係的套裝軟體。

上例中執行apt-get會刪除指定套裝軟體以及與之有依賴關係的軟體 包，但它們的組態檔案，如果有的話，會完好無損地保留在系統裡。如果想徹底刪 除這些包及其組態檔案，執行：

```
# apt-get --purge remove gnome-panel
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
  gnome-applets* gnome-panel* gnome-panel-data* gnome-session*
0 packages upgraded, 0 newly installed, 4 to remove and 1  not upgraded.
Need to get 0B of archives. After unpacking 14.6MB will be freed.
Do you want to continue? [Y/n]
```

注意：套裝軟體名字後面的*表示該套裝軟體所有的組態檔案也將被刪除。

就像install時一樣，你可以在remove命令中用一個符號來指 定安裝某個套裝軟體。在刪除套裝軟體時，如果你在套裝軟體名字後面緊跟一個 “+” ，那麼該套裝軟體就會被安裝而不是刪除。

```
# apt-get --purge remove gnome-panel nautilus+
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
```

```
  bonobo libmedusa0 libnautilus0 nautilus
The following packages will be REMOVED:
  gnome-applets* gnome-panel* gnome-panel-data* gnome-session*
The following NEW packages will be installed:
  bonobo libmedusa0 libnautilus0 nautilus
0 packages upgraded, 4 newly installed, 4 to remove and 1  not upgraded.
Need to get 8329kB of archives. After unpacking 2594kB will be used.
Do you want to continue? [Y/n]
```

注意，apt-get列出了那些將要被安裝的額外套裝軟體(即保證該軟體 包正常執行的其它套裝軟體)和將要被刪除關聯套裝軟體，然後，再次列出了將要被安 裝的套裝軟體(包括了額外的包)。

3.4 更新套裝軟體

套裝軟體更新是APT最成功的特點。只需一條命令即可完成更新： apt-get upgrade。你可以使用這條命令從相同版本號的發佈版中更新 套裝軟體，也可以從新版本號的發佈版中更新套裝軟體，盡管實作後一種更新的推薦 命令爲apt-get dist-upgrade；詳情請參考'升級到新版本' 11。

在執行該命令時加上-u選項很有用。這個選項讓APT顯示完整的可更新 套裝軟體列表。不加這個選項，你就只能盲目地更新。APT會下載每個套裝軟體的最新 更新版本，然後以合理的次序安裝它們。注意在執行該命令前應先執行 apt-get update更新資料庫。詳情請參考'更新可用套裝軟體列表' 7。請看下 面的例子：

```
# apt-get -u upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages have been kept back
  cpp gcc lilo
The following packages will be upgraded
  adduser ae apt autoconf debhelper dpkg-dev esound esound-common ftp indent
  ipchains isapnptools libaudiofile-dev libaudiofile0 libesd0 libesd0-dev
  libgtk1.2 libgtk1.2-dev liblockfile1 libnewt0 liborbit-dev liborbit0
  libstdc++2.10-glibc2.2 libtiff3g libtiff3g-dev modconf orbit procps psmisc
29 packages upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Need to get 5055B/5055kB of archives. After unpacking 1161kB will be used.
Do you want to continue? [Y/n]
```

整個更新程序非常簡單。注意在本例中頭幾行，apt-get報告有些套裝軟體 的更新被kept back，這表明這些套裝軟體的更新版本因故無法安裝，可能 的原因有關聯不同步(目前沒有供下載的新版本關聯包)或關聯延伸(需要安裝新的 關聯包以配合新版套裝軟體)。

對於第一種原因沒有很好的解決方法，對於第二次原因，執行apt-get intall安裝所需的新關聯包就可以。另一個更好的解決方法就是使用dist-upgrade。詳情請參考'升級到新版本' 11。

3.5 升級到新版本

APT的絕活就是讓你一次就完成整個系統的更新，不論是通過Internet還是通過光 槃檔案(購買的碟片或下載的ISO鏡像檔案)。

它也可以用來更新那些關聯關係發生改變的套裝軟體。即如前所述的那些使用apt-get upgrade時被不被更新(kept back)的套裝軟體。

例如，假設你目前使用的Debian爲stable revision 0，而你購買了revision 3的 新版Debian，你可以使用APT從新光碟上升級你的系統。使用apt-cdrom(參考'將CD-ROM加入source.list檔案' 6)將光碟加載到/etc/apt/sources.list中，然後執行apt-get dist-upgrade。

請注意，APT總是搜索最新版本的套裝軟體，因此，如果一個套裝軟體在你的/etc/apt/sources.list中所列的版本比光碟上所列的版本要新，那麼 APT會下載其中的套裝軟體而不是使用光碟上的套裝軟體。

在'更新套裝軟體' 10節的例子中，我們看到有些包被kept back了，現 在我們就用dist-upgrade方法來解決這個問題：

```
# apt-get -u dist-upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
Calculating Upgrade... Done
The following NEW packages will be installed:
  cpp-2.95 cron exim gcc-2.95 libident libopenldap-runtime libopenldap1
  libpcre2 logrotate mailx
The following packages have been kept back
  lilo
The following packages will be upgraded
  adduser ae apt autoconf cpp debhelper dpkg-dev esound esound-common ftp gc
  indent ipchains isapnptools libaudiofile-dev libaudiofile0 libesd0
  libesd0-dev libgtk1.2 libgtk1.2-dev liblockfile1 libnewt0 liborbit-dev
  liborbit0 libstdc++2.10-glibc2.2 libtiff3g libtiff3g-dev modconf orbit
  procps psmisc
31 packages upgraded, 10 newly installed, 0 to remove and 1 not upgraded.
Need to get 0B/7098kB of archives. After unpacking 3118kB will be used.
Do you want to continue? [Y/n]
```

注意現在那些套裝軟體將會被更新，那些新的關聯套裝軟體也會被安裝。但是lilo仍被 kept back，可能還存在一些比建立新關聯更棘手的問題，我們通過如下 方法確定問題所在：

```
# apt-get -u install lilo
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  cron debconf exim libident libopenldap-runtime libopenldap1 libpcre2
  logrotate mailx
```

```
The following packages will be REMOVED:
  debconf-tiny
The following NEW packages will be installed:
  cron debconf exim libident libopenldap-runtime libopenldap1 libpcre2
  logrotate mailx
The following packages will be upgraded
  lilo
1 packages upgraded, 9 newly installed, 1 to remove and 31 not upgraded.
Need to get 225kB/1179kB of archives. After unpacking 2659kB will be used.
Do you want to continue? [Y/n]
```

檢視上述提示資訊可知，lilo與debconf-tiny包產生了一個 新衝突，這表明除非刪除debconf-tiny，否則將無法安裝(或更新)lilo。

想知道該保留或刪除哪些套裝軟體，你可以使用：

```
# apt-get -o Debug::pkgProblemResolver=yes dist-upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
Calculating Upgrade... Starting
Starting 2
Investigating python1.5
Package python1.5 has broken dep on python1.5-base
  Considering python1.5-base 0 as a solution to python1.5 0
  Holding Back python1.5 rather than change python1.5-base
Investigating python1.5-dev
Package python1.5-dev has broken dep on python1.5
  Considering python1.5 0 as a solution to python1.5-dev 0
  Holding Back python1.5-dev rather than change python1.5
 Try to Re-Instate python1.5-dev
Done
Done
The following packages have been kept back
  gs python1.5-dev
0 packages upgraded, 0 newly installed, 0 to remove and 2  not upgraded.
```

現在，你很容易就知道不能安裝python1.5-dev套裝軟體是因爲無法滿足另一個軟體包python1.5的關聯要求。

3.6 移除無用套裝軟體檔案：apt-get clean and autoclean

當你需要安裝某個套裝軟體時，APT從/etc/apt/sources.list中所列的主機下載所 需的檔案，將它們保存到本機軟體庫(/var/cache/apt/archives/)， 然後開始安裝，參考'安裝套裝軟體' 7。

本地軟體庫會不斷膨脹占用大量硬碟空間，幸運的是，APT提供了工具來管理本地軟體庫：apt-get的clean方法和autoclean方法。

apt-get clean將刪除/var/cache/apt/archives目錄和/var/cache/apt/archives/partial目錄下鎖檔案以外的所有檔案。這樣以來，當你需要再次安裝某個套裝軟體時，APT將重新下載它。

apt-get autoclean只有刪除那些不需要再次下載的檔案。

下面這個例子顯示了apt-get autoclean如何工作：

```
# ls /var/cache/apt/archives/logrotate* /var/cache/apt/archives/gpm*
logrotate_3.5.9-7_i386.deb
logrotate_3.5.9-8_i386.deb
gpm_1.19.6-11_i386.deb
```

在/var/cache/apt/archives目錄下有兩個不同版本的logrotate 套裝軟體檔案以及一個gpm套裝軟體檔案。

```
# apt-show-versions -p logrotate
logrotate/stable uptodate 3.5.9-8
# apt-show-versions -p gpm
gpm/stable upgradeable from 1.19.6-11 to 1.19.6-12
```

apt-show-versions顯示logrotate_3.5.9-8_i386.deb 提供了logrotate的升級版本，所以 logrotate_3.5.9-7_i386.deb沒用了。同樣 gpm_1.19.6-11_i386.deb也沒有用了，因爲可以下載該套裝軟體的更新版本。

```
# apt-get autoclean
Reading Package Lists... Done
Building Dependency Tree... Done
Del gpm 1.19.6-11 [145kB]
Del logrotate 3.5.9-7 [26.5kB]
```

總之，apt-get autoclean只有刪除那些過時的檔案。參考 '如何從Debian的專用版本下升級套裝軟體' 15以了解apt-show-versions的更多詳情。

3.7 在dselect中操作APT

dselect工具幫助使用者選取想要安裝的Debian套裝軟體。它有點複雜甚至令人望而生厭，但經過實踐你就能掌握它恐怖的終端機界面。

dselect進階功能之一就是它知道利用Debian套裝軟體的“推薦”和“建議”能力。(Debian軟件包有一種能力：推薦或建議系統在安裝自己的同時，安裝別的套裝軟體以配合自身的工

作，當然這些推薦的套裝軟體不一定是必須的；而dselect工具可以識別和利用這個能 力，使用dselect時你就能體會到。譯者注)以root身份執行dselect，進入程 序後選擇apt作爲連線方式(access)。該步驟不是必須的，但如果你沒有光碟機而且想通 過Internet下載安裝套裝軟體，這是使用dselect的最好方法。

想深入學習dselect的使用方式，請到Debian网站查閱dselect文件頁面http://www.debian.org/doc/ddp.

在dselect中選好了套裝軟體後，執行：

```
# apt-get -u dselect-upgrade
```

如下例所示：

```
# apt-get -u dselect-upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
  lbxproxy
The following NEW packages will be installed:
  bonobo console-tools-libs cpp-3.0 enscript expat fingerd gcc-3.0
  gcc-3.0-base icepref klogd libdigest-md5-perl libfnlib0 libft-perl
  libgc5-dev libgcc300 libhtml-clean-perl libltdl0-dev libsasl-modules
  libstdc++3.0 metamail nethack proftpd-doc psfontmgr python-newt talk tidy
  util-linux-locales vacation xbill xplanet-images
The following packages will be upgraded
  debian-policy
1 packages upgraded, 30 newly installed, 1 to remove and 0  not upgraded.
Need to get 7140kB of archives. After unpacking 16.3MB will be used.
Do you want to continue? [Y/n]
```

比較一下我們在相同系統上執行apt-get dist-upgrade時的情形：

```
# apt-get -u dist-upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
Calculating Upgrade... Done
The following packages will be upgraded
  debian-policy
1 packages upgraded, 0 newly installed, 0 to remove and 0  not upgraded.
Need to get 421kB of archives. After unpacking 25.6kB will be freed.
Do you want to continue? [Y/n]
```

我們看到在前例中許多套裝軟體被安裝是其它套裝軟體“推薦”或“建議”的結果。另外一些套裝軟體被安裝或刪除(例如lbxproxy套裝軟體)是我們通過dselect工具作出的決定。 由此可見dselect與APT結合起來將是一個功能強大的工具。

3.8 如何保持一個混合系統

人們有時會對這種情況有興趣——使用一個版本的Debain作爲其主發行版，但從另一 個分支上安裝一個或多個包。

要設定你的Debian主版本，應當修改/etc/apt/apt.conf檔案，並加 入：

```
APT::Default-Release "version";
```

其 中*version*是 你 希 望 作 爲 主 發 行 版 使 用 的Debian版 本 。 你 可 以 使 用 的 版 本 有stable、testing和unstable。 要從另外一個版本中安裝套裝軟體，你必須按照如下方式執行APT：

```
# apt-get -t distribution install package
```

爲了使其可以個工作，在你的/etc/apt/sources.list中至少有一行 是關於你要使用的那個版本的，而且要使用的套裝軟體也必須存在於該版本中。

你也可以要求使用某個特定把本的套裝軟體，如下所示：

```
# apt-get install package=version
```

例如，下面的命令將會安裝2.2.4-1版的nautilus軟 件包：

```
# apt-get install nautilus=2.2.4-1
```

重要資訊：最新版的Debian套裝軟體首先會上傳到“unstable”發佈版中， 這個發佈版包含了套裝軟體所有變更階段，無論是小修小補還是影響到眾多套裝軟體 乃至整個系統的重大修改。所以，新手和那些強調系統穩定性的使用者不 會使用這個發佈版。

“testing”發佈版比起“unstable”發佈版，多注重了些系統穩定性，但正式執行的 系統應當使用“stable”發佈版。

3.9 如何從Debian的專用版本下升級套裝軟體

apt-show-versions提供了一個安全的途徑，讓那些使用混合系統的 使用者放心升級他們的系統，不必擔心升級會將原來屬於stable的包升級成了unstable包。 例如，在安裝了apt-show-versions套裝軟體之後， 使用這條命令將只升級你的unstable套裝軟體：

```
# apt-get install `apt-show-versions -u -b | grep unstable | cut -d ' ' -f 1
```

3.10 如何維護已安裝套裝軟體的多個版本（複雜） How to keep specific versions of packages installed (complex)

你可能會遇到這種情況，變更了某個套裝軟體中的一些檔案，但你沒有時間或根本就 不想將這些變更引入到新版本中。或是，你將系統升級到3.0，但仍想繼續使用Debian 2.2下的某個套裝軟體。你可以“釘住”這個版本，這樣它就不會被更新了。

操作起來十分簡單，你只需編輯/etc/apt/preferences檔案。/etc/apt/preferences.

檔案格式很簡單：

```
Package: <package>
Pin: <pin definition>
Pin-Priority: <pin's priority>
```

每個條目都要以空白行與其它條目分割開。例如，我對sylpheed 套裝軟體做了某些修改以使用“reply-to-list”功能，其版本爲0.4.99。我想保留這 些修改不被更新，可加上：

```
Package: sylpheed
Pin: version 0.4.99*
```

注意我用了一個*(星號)。這是一個“萬用字元”；它表明我希望“釘住”所有 以0.4.99打頭的版本(以防它們被下載並安裝到我機器上。pin控制的是伺服器端的 更新套裝軟體而非本地的已安裝套裝軟體。譯者注)。因爲Debian使用“Debian版本號” 爲其套裝軟體定版本，我不想進行所有這些版本的升級，如果不用萬用字元， 那麼0.4.99-1版或0.4.99-10版只要一出爐系統就會安裝它們。如果你修改了套裝軟體， 你一定不希望這麼做。

Pin的優先等級幫助我們檢查一個與“Packages:”和“Pin:”相符合的套裝軟體是否應該被 安裝。當優先等級比較高時，符合的套裝軟體將會被安裝。你可以查閱apt_preferences(7)，其中有關於優先等級的詳細討論，但通過 一些簡單的例子也可以了解基本的概念。下面就說明了在上面的sylpheed例子中設 定優先等級網域的效果。

1001 Sylpheed 0.4.99永遠不會被apt取代。如果可能，apt甚至會用0.4.99版取代已經 安裝的更高的版本呢。只有那些優先等級比1000大的套裝軟體才會降級。

1000 除了不會將高版本降級以外，與1001的效果相同。

990 版本0.4.99只會被偏好發佈系列中高版本的的套裝軟體取代，偏好發行版由變數“APT::Default-Release”定義(參考‘如何保持一個混合系統’ 15。

500 任何發佈系列中比0.4.99版本高的sylpheed都會被安裝，但相對於低版本而言，仍然 建議使用0.4.99。

100 任何發佈系列中高版本的sylpheed都會被安裝；因此只有沒有其它版本可以安裝時 才會安裝0.4.99。已安裝包的優先等級。

-1 負的優先等級也是允許的，它會組織0.4.99版被安裝。

釘子也可以用來指定套裝軟體的version、release或 origin。

我們已經看到，釘在一個version上，可以使用具體的版本號，也可以使 用萬用字元一次指定多個版本。

release選項依賴於APT倉庫上的或是CD中的Release檔案。如果你使用的 APT倉庫並沒有提供這個檔案，這個選項就沒有任何用處了。你可以在 /var/lib/apt/lists/中看到Release檔案的內容。release的參數是： a(存檔)、c(部件)、v(版本)、o(起來源) 和l(標籤)。

例如：

```
Package: *
Pin: release v=2.2*,a=stable,c=main,o=Debian,l=Debian
Pin-Priority: 1001
```

在這個例子中，我們選擇了Debian版本2.2*(可以是2.2r2、2.2r3——這些版本中通常 包含了對安全問題的修復和其它重要更新)，stable倉庫，main (相應的還有contrib或是non-free)區段、起來源和標籤都是 Debian。origin(o=)定義了誰製作了這個Release檔案，label(l=)定義了發行版的 名字：Debian自己就使用Debian而Progeny則使用Progeny。Release檔案的例子如下 所示：

```
$ cat /var/lib/apt/lists/ftp.debian.org.br_debian_dists_potato_main_binary-i
Archive: stable
Version: 2.2r3
Component: main
Origin: Debian
Label: Debian
Architecture: i386
```

Chapter 4

幾個非常有用的工具

4.1 如何安裝本地編譯的套裝軟體：equivs

有時，使用者想使用某些軟體的特殊版本，它們只以來源程式碼的形式存在，沒有現成的Debian套裝軟體。套裝軟體管理系統在處理這類事務時可能會出問題。假設你想編譯新版本的信件伺服器，所有的事情都很正常，但是Debian中的很多套裝軟體是依賴於MTA(信件傳輸代理)的。由於你是自己手動編譯安裝軟體，套裝軟體管理系統對此一無所知。

現在是`equivs`登台的時候了。用它來安裝套裝軟體，它所做的工作就是建立一個新的空套裝軟體來實作關聯，讓套裝軟體管理系統相信所有的依賴關係都可以滿足。

在我們開始以前，我必須提醒你，編譯某個軟體最安全的方法是對該軟體現有的Debian套裝軟體進行修改後重新編譯，如果你並不知道你正在干什麼，勸你不要使用equivs取代關聯包。更多資訊請參考'來源碼包操作' 29。

繼續上面的例子，你安裝好了新編譯的`postfix`，接下來打算安裝`mutt`。突然你發現`mutt`想安裝另外一個MTA，但實際上你已經有了你的MTA。

前往某個目錄(例如`/tmp`)執行：

```
# equivs-control name
```

將*name*取代爲你建立的控制檔案，控制檔案按如下格式建立：

```
Section: misc
Priority: optional
Standards-Version: 3.0.1

Package: <enter package name; defaults to equivs-dummy>
Version: <enter version here; defaults to 1.0>
Maintainer: <your name and email address; defaults to username>
Pre-Depends: <packages>
Depends: <packages>
```

```
Recommends: <packages>
Suggests: <package>
Provides: <(virtual)package>
Architecture: all
Copyright: <copyright file; defaults to GPL2>
Changelog: <changelog file; defaults to a generic changelog>
Readme: <README.Debian file; defaults to a generic one>
Extra-Files: <additional files for the doc directory, comma-separated>
Description: <short description; defaults to some wise words>
 long description and info
 .
 second paragraph
```

我們只需按自己的需要修改相關專案就行了。檔案中每個項目都描述得很清楚，我 們不必在此逐行解釋它們。現在開始修改吧：

```
Section: misc
Priority: optional
Standards-Version: 3.0.1

Package: mta-local
Provides: mail-transport-agent
```

行了，就是這樣。mutt依賴於mail-transport-agent，這是所有MTA共同提供的一個虛擬包，我可以簡單地將這個套裝軟體命名 爲mail-transport-agent，不過我更愿意使用系統的虛擬包方 案，使用Provides選項。

現在你可以開始組建套裝軟體了：

```
# equivs-build name
dh_testdir
touch build-stamp
dh_testdir
dh_testroot
dh_clean -k
# Add here commands to install the package into debian/tmp.
touch install-stamp
dh_testdir
dh_testroot
dh_installdocs
dh_installchangelogs
dh_compress
dh_fixperms
dh_installdeb
dh_gencontrol
```

```
dh_md5sums
dh_builddeb
dpkg-deb: building package `name' in `../name_1.0_all.deb'.
```

套裝軟體已經被建立了，注意，套裝軟體是建立在目前目錄中的。

然後安裝這個.deb檔案。

眾所周知，equivs的使用方式很多，譬如你可以建立一個my-favorites套裝軟體，它依賴於你通常安裝的套裝軟體。盡情發揮你的想像力吧，當然還是要小心。

重要提示：在/usr/share/doc/equivs/examples目錄下有控制檔案 的例子，最好看一下。

4.2 移除無用的地區組態(locale)檔案：localepurge

許多Debian使用者只有在固定地區使用Debian。例如，在巴西的Debian使用者，通常使用pt_BR地區組態檔案而不會關心es地區組態檔案。

對於這類使用者而言localepurge是一個非常有用的工具，你可 以只有保留你目前所用的地區組態檔案，刪除其它無用的檔案，從而釋放大量硬碟空 間。執行apt-get install localepurge就行了。

它組態起來非常容易，debconf的提問將引導使用者一步一步完成設定。在回答第一 個問題時請務必謹慎，如果回答錯了，系統可能刪掉所有的地區組態檔案，包括你 正在使用的這個。複原它們的唯一方法就是重裝那些套裝軟體。

4.3 如何知曉哪些套裝軟體可以升級

apt-show-versions工具可以告訴你系統中哪些包可以更新以及其它 一些有用的資訊。-u選項可以顯示可更新套裝軟體列表：

```
$ apt-show-versions -u
libeel0/unstable upgradeable from 1.0.2-5 to 1.0.2-7
libeel-data/unstable upgradeable from 1.0.2-5 to 1.0.2-7
```

Chapter 5

獲取套裝軟體資訊

有些基於APT系統的前端程序，能十分方便地獲得系統套裝軟體列表，列表包括可安 裝或已安裝的套裝軟體，還可以顯示某套裝軟體屬於哪個section，它的優先等級是多少， 它的說明文件等等。

但是...在此我們想的學習如何使用APT本身來完成。你如何找出你想要安裝的軟體 包的名稱？

我們完成這個工作的方法有很多。我們從apt-cache開始，APT系統使用 這個程序來維護它的資料庫。下面我們通過一些實際操作來對它做個概覽。

5.1 獲得套裝軟體名稱

假設你十分懷念玩Atari 2600的好日子，你決定用APT安裝一個Atari emulator， 隨後再下載幾個游戲，你可以這樣：

```
# apt-cache search atari
atari-fdisk-cross - Partition editor for Atari (running on non-Atari)
circuslinux - The clowns are trying to pop balloons to score points!
madbomber - A Kaboom! clone
tcs - Character set translator.
atari800 - Atari emulator for svgalib/X/curses
stella - Atari 2600 Emulator for X windows
xmess-x - X binaries for Multi-Emulator Super System
```

我們找到了幾個相關的套裝軟體，以及有關的簡單描述。想進一步獲得某個套裝軟體的 詳細資訊，你可以執行：

```
# apt-cache show stella
Package: stella
Priority: extra
Section: non-free/otherosfs
```

```
Installed-Size: 830
Maintainer: Tom Lear <tom@trap.mtview.ca.us>
Architecture: i386
Version: 1.1-2
Depends: libc6 (>= 2.1), libstdc++2.10, xlib6g (>= 3.3.5-1)
Filename: dists/potato/non-free/binary-i386/otherosfs/stella_1.1-2.deb
Size: 483430
MD5sum: 11b3e86a41a60fa1c4b334dd96c1d4b5
Description: Atari 2600 Emulator for X windows
 Stella is a portable emulator of the old Atari 2600 video-game console
 written in C++.  You can play most Atari 2600 games message it.  The latest
 news, code and binaries for Stella can be found at:
 http://www4.ncsu.edu/~bwmott/2600
```

螢幕上顯示出這個套裝軟體的詳細資訊及其用途的完整描述。如果你的系統中已安裝 了某個套裝軟體而系統又搜索到它的新版本，系統會將它們的詳細資訊一並列出。如 下例：

```
# apt-cache show lilo
Package: lilo
Priority: important
Section: base
Installed-Size: 271
Maintainer: Russell Coker <russell@coker.com.au>
Architecture: i386
Version: 1:21.7-3
Depends: libc6 (>= 2.2.1-2), debconf (>=0.2.26), logrotate
Suggests: lilo-doc
Conflicts: manpages (<<1.29-3)
Filename: pool/main/l/lilo/lilo_21.7-3_i386.deb
Size: 143052
MD5sum: 63fe29b5317fe34ed8ec3ae955f8270e
Description: LInux LOader - The Classic OS loader can load Linux and others
 This Package contains lilo (the installer) and boot-record-images to
 install Linux, OS/2, DOS and generic Boot Sectors of other OSes.
 .
 You can use Lilo to manage your Master Boot Record (message a simple text s
 or call Lilo from other Boot-Loaders to jump-start the Linux kernel.

Package: lilo
Status: install ok installed
Priority: important
Section: base
Installed-Size: 190
Maintainer: Vincent Renardias <vincent@debian.org>
Version: 1:21.4.3-2
```

```
Depends: libc6 (>= 2.1.2)
Recommends: mbr
Suggests: lilo-doc
Description: LInux LOader - The Classic OS loader can load Linux and others
 This Package contains lilo (the installer) and boot-record-images to
 install Linux, OS/2, DOS and generic Boot Sectors of other OSes.
 .
 You can use Lilo to manage your Master Boot Record (message a simple text s
 or call Lilo from other Boot-Loaders to jump-start the Linux kernel.
```

注意，首先列出的是可用套裝軟體，接著列出的是已安裝套裝軟體。獲取某個套裝軟體的 一般資訊可執行：

```
# apt-cache showpkg penguin-command
Package: penguin-command
Versions:
1.4.5-1(/var/lib/apt/lists/download.sourceforge.net_debian_dists_unstable_ma

Reverse Depends:
Dependencies:
1.4.5-1 - libc6 (2 2.2.1-2) libpng2 (0 (null)) libsdl-mixer1.1 (2 1.1.0) lib
Provides:
1.4.5-1 -
Reverse Provides:
```

如果只有想了解某套裝軟體的與哪些套裝軟體關聯，可執行：

```
# apt-cache depends penguin-command
penguin-command
  Depends: libc6
  Depends: libpng2
  Depends: libsdl-mixer1.1
  Depends: libsdl1.1
  Depends: zlib1g
```

總之，有一系列工具可幫助我們找到我們想要的套裝軟體。

5.2 使用dpkg尋找套裝軟體名稱

另一個定位套裝軟體的方法是知道套裝軟體中某個關鍵檔案的名稱。例如，你編譯時需 要某個“.h”頭檔案，尋找提供該檔案的套裝軟體，你可以執行：

```
# dpkg -S stdio.h
libc6-dev: /usr/include/stdio.h
libc6-dev: /usr/include/bits/stdio.h
perl: /usr/lib/perl/5.6.0/CORE/nostdio.h
```

或是：

```
# dpkg -S /usr/include/stdio.h
libc6-dev: /usr/include/stdio.h
```

解系統中已安裝軟體的套裝軟體名稱十分有用，譬如當你想清理硬碟空間時，可以執行：

```
# dpkg -l | grep mozilla
ii  mozilla-browse 0.9.6-7        Mozilla Web Browser
```

這個命令的缺點是它會“截斷”套裝軟體的名字。在上例中，套裝軟體的全稱是mozilla-browser，解決這個問題可以使用COLUMNS環境變數：

```
[kov]@[couve] $ COLUMNS=132 dpkg -l | grep mozilla
ii  mozilla-browser             0.9.6-7                      Mozilla Web Brow
```

或顯示成這樣：

```
# apt-cache search "Mozilla Web Browser"
mozilla-browser - Mozilla Web Browser
```

5.3 如何“按需”安裝套裝軟體

你正在編譯某段程序，突然，停住了！一條錯誤資訊報告說你沒有它需要的.h頭檔案。讓auto-apt來救你吧，它問你是否要安裝需要 的套裝軟體，然後挂起編譯進程，安裝好套裝軟體後再恢復編譯進程。

你所要做的只有只有是：

```
# auto-apt run command
```

這裡“command”指在執行程序中可能出現“需求檔案不存在”問題的命令。例如：

```
# auto-apt run ./configure
```

一會兒，它就會告訴你要安裝所需的套裝軟體並自動前往apt-get處理。如果你正在 執行X，就會一個圖形界面提示視窗。

爲了提高效率auto-apt所用的資料庫需要實時更新。可調用 auto-apt update，auto-apt updatedb和 auto-apt update-local來完成更新。

5.4 如何知道檔案屬於哪個套裝軟體

如果你想安裝某個套裝軟體，但用apt-cache查不出它的名稱，不過你知道這個程序的檔案名稱，或這個套裝軟體中某些檔案的檔案名稱，那麼你可以用apt-file來尋找套裝軟體名稱。如下所示：

```
$ apt-file search filename
```

它用起來很象dpkg -S，不過它還會列出包含該檔案的已刪除套裝軟體。它也可以用來尋找哪個套裝軟體包含編譯時所缺的檔案，當然，解決這類問題auto-apt可能是更好的方案，請參考'如何“按需”安裝套裝軟體' 26。

用這個命令，你可以列出套裝軟體的內容：

```
$ apt-file list packagename
```

apt-file用一個資料庫來存放所有套裝軟體的內容資訊，和auto-apt一樣，這個資料庫也需要實時更新，完成更新可以執行：

```
# apt-file update
```

預設情況下，apt-file和auto-apt使用同一個資料庫，參考'如何“按需”安裝套裝軟體' 26。

5.5 如何掌握套裝軟體的變化情況

在每個套裝軟體被安裝以後，都會在文件目錄(/usr/share/doc/packagename)產生一個changelog.Debian.gz的檔案，這個檔案記錄了該套裝軟體最後一次更新對系統做了哪些修改，你可以用zless閱讀這些資訊。不過當你對整個系統進行升級以後，逐個檢視套裝軟體的更新資訊可不是件容易事。

有一個工具能幫你完成這項工作，它就是apt-listchanges。首先你要裝上apt-listchanges套裝軟體。在安裝的程序中，爲了進行配置，Debconf會問你一些問題，按你的要求回答它們就行了。

第一個問題是問你希望apt-listchanges如何來顯示修改日誌。你可以讓它把資訊通過信件的方式發送給你，這對於自動更新是非常有用的。或是你可以讓它在less等程序中顯示修改日誌，這樣在繼續升級前你就可以檢視它們了。如果你不希望apt-listchanges在升級的時候自動的執行，可以回答none。

安裝了apt-listchanges後，每當apt下載套裝軟體之後(不論來來源是Internet、光碟或是硬碟)都會顯示這些套裝軟體的系統更新資訊。

Chapter 6

來源碼包操作

6.1 下載來源碼包

在自由軟體的世界裡，經常需要學習來源碼或爲程序除錯，所以你需要下載它們。APT提供了一套簡便的方法幫你獲得發佈版中眾多程序的來源程式碼以及建立一個.debs所需的所有檔案。

Debian來源碼的另一個普遍用途是將unstable發佈版的新版程序進行改寫以供別的發布版使用。例如，從stable發佈版外引入新的套裝軟體，需要重新產生.debs將它在原發佈版中的關聯關係遷移到新的發佈版。

要完成這些工作，/etc/apt/sources.list檔案中deb-src所指參照鏡像來源應該是unstable，別忘了將行首的註解符去掉。詳情參考'/etc/apt/sources.list檔案'3。

用下面的命令下載來源碼包：

```
$ apt-get source packagename
```

通常會下載三個檔案：一個.orig.tar.gz、一個.dsc和一個.diff.gz。對於Debian專用的套裝軟體，不會下載最後一個檔案，第一個檔案的文件名中沒有“orig”項。

dpkg-source通過.dsc檔案中的資訊，將來源碼包解包到 *packagename-version* 目錄，下載下來的來源碼包中有一個 debian/目錄，裡面是建立.deb包所需的檔案。

想要下載的來源碼包自動編譯成套裝軟體，只需在命令行中加上-b，如下：

```
$ apt-get -b source packagename
```

如果你不打算在下載後就立刻建立.deb檔案，你可以在之後用下面的命令建立：

```
$ dpkg-buildpackage -rfakeroot -uc -b
```

上述命令應當在下載後爲套裝軟體建立的目錄中執行。要安裝用這種方式組建好的 套裝軟體，只能直接使用套裝軟體管理器，例如：

```
# dpkg -i file.deb
```

apt-get的source命令與它的其它命令有所不同，普通用 户就可以執行source命令。檔案被下載到使用者調用apt-source package 命令時所處的目錄中。

6.2 編譯來源碼包所需的套裝軟體

通常，編譯來源碼包時要用到某些頭檔案和共享庫，所有的來源碼包的控制檔案中都 有一個網域“Build-Depends：”，網域中指出了編譯該來源碼包需要哪些附加包。

APT提供了一個簡單的方法下載這些附加包，你只需執行apt-get build-dep package，其中“package”就是你打算編譯的來源碼包 名稱。見下例：

```
# apt-get build-dep gmc
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  comerr-dev e2fslibs-dev gdk-imlib-dev imlib-progs libgnome-dev libgnorba-d
  libgpmg1-dev
0 packages upgraded, 7 newly installed, 0 to remove and 1  not upgraded.
Need to get 1069kB of archives. After unpacking 3514kB will be used.
Do you want to continue? [Y/n]
```

這些將要被安裝的包是用於正確編譯gmc的。注意這個命令不 能用來搜索某個軟體的來源碼包，你得另外執行apt-get source下載來源碼包。

如果你想做的是檢查要編譯一個套裝軟體需要哪些其它的套裝軟體，apt-cache show可以顯示它(從那考‘獲取套裝軟體資訊’ 23，在眾多資訊之中，Build-Depends一行會列出那些需要的套裝軟體。

```
# apt-cache showsrc package
```

Chapter 7

如何處理錯誤

7.1 一般錯誤

錯誤總是發生，大部分是因爲使用者的粗心，下面列舉一些常見錯誤及處理方法。

如果在執行apt-get install package時，你的系統報告如下資訊：

```
Reading Package Lists... Done
Building Dependency Tree... Done
W: Couldn't stat source package list 'http://people.debian.org unstable/ Pac
W: You may want to run apt-get update to correct these missing files
E: Couldn't find package penguineyes
```

上次你修改/etc/apt/sources.list後，忘了執行apt-get update更新。

如果出現這樣的資訊：

```
E: Could not open lock file /var/lib/dpkg/lock - open (13 Permission denied)
E: Unable to lock the administration directory (/var/lib/dpkg/), are you roo
```

如果你沒有root權限，執行除source外的其它apt-get命 令，如會出現上面的錯誤資訊。這是因爲你是普通用户。

當你同時執行兩個apt-get進程，或是當你試圖執行apt-get時已有一個的dpkg進程處於活化狀態，系統也 會報告與上面相似的錯誤資訊。唯一能與其它命令同時執行的只有source 命令。

如果在安裝程序中出現插斷，然後你發現該套裝軟體既不能重裝又不能刪除，試試下 面兩個命令：

```
# apt-get -f install
# dpkg --configure -a
```

重試著安裝那個套裝軟體，如果不行再次執行上述命令後重試。這兩個命令對於那些使用unstable的玩家非常有用。

如果你在執行`apt-get update`時看到“E: Dynamic MMap ran out of room”，那麼在`/etc/apt/apt.conf`加入如下內容：

```
APT::Cache-Limit 10000000;
```

7.2 在哪兒獲得幫助？

如果你發現自己有太多疑問，沒關係，有大量的Debian套裝軟體管理系統文件供你參考。`--help`和幫助文件能爲你提供巨大的幫助，這些文件位元於`/usr/share/doc`目錄中，如`/usr/share/doc/apt`。

如果本文件沒法幫你排憂解難，可以去Debian信件列表找找答案，在相關欄目內你會獲得更多資訊。Debian的網址是：http://www.debian.org。

記住只有Debian使用者才能這些信件列表和資源，其它作業系統的使用者請到相關係統發佈者建立的社區中獲取更多資源。

Chapter 8

哪些發行版支援APT？

下面是部分使用APT系統的發行版的名稱：

Debian GNU/Linux (http://www.debian.org) - APT正是爲這個發行版開發的

Conectiva (http://www.conectiva.com.br) - 這是第一個將APT移植到rpm上的發行版

Libranet (http://www.libranet.com)

Mandrake (http://www.mandrake.com)

PLD (http://www.pld.org.pl)

Vine (http://www.vinelinux.org)

APT4RPM (http://apt4rpm.sf.net)

Alt Linux (http://www.altlinux.ru/)

Red Hat (http://www.redhat.com/)

Sun Solaris (http://www.sun.com/)

SuSE (http://www.suse.de/)

Yellow Dog Linux (http://www.yellowdoglinux.com/)

Chapter 9

致謝

非常感謝你們，我Debian-BR項目組的好朋友們！還有你Debian，始終在我身邊幫助 我，給我動力不停工作爲全人類做出貢獻，幫助我樹立拯救世界的理想。:)

我還要感謝CIPSGA，他們給予我們項目組乃至整個自由軟體項目巨大的幫助，是我 們靈感的來源泉。

特別感謝：

Yooseong Yang <yooseong@debian.org>

Michael Bramer <grisu@debian.org>

Bryan Stillwell <bryan@bokeoa.com>

Pawel Tecza <pawel.tecza@poczta.fm>

Hugo Mora <h.mora@melix.com.mx>

Luca Monducci <luca.mo@tiscali.it>

Tomohiro KUBOTA <kubota@debian.org>

Pablo Lorenzzoni <spectra@debian.org>

Steve Langasek <vorlon@netexpress.net>

Arnaldo Carvalho de Melo <acme@conectiva.com.br>

Erik Rossen <rossen@freesurf.ch>

Ross Boylan <RossBoylan@stanfordalumni.org>

Matt Kraai <kraai@debian.org>

Aaron M. Ucko <ucko@debian.org>

Jon Åslund <d98-jas@nada.kth.se>

Chapter 10

本使用指南的新版本

本操作手冊由Debian-BR項目組 (http://www.debian-br.org)撰寫，我們希望它能爲Debian使用者提供有效的幫助。

你可以從Debian文件項目頁面獲得本文件的新版本：http://www.debian.org/doc/ddp。

對本文件有任何意見或建議可直接發email給我：<kov@debian.org>。(華文使用者請發給譯者)