

HOW TO INSTALL SOFTWARE

[A. Howlett](#), 8 September 2004

Debian Packages

A GNU/Linux system includes hundreds of Free and Open Source Software projects. For instance, the Linux kernel is a Free Software project, OpenOffice is an Open Source software project, the XFree86 Window system is an Open Source software project, and the K Desktop Environment is a Free Software project.. You could download the source code for each of these projects, compile the program and install it. Some people do that, but it requires a lot of skill and takes a long time.

Another way is to let someone else compile everything and you just copy the resulting “stuff” - manual pages, shared library files, executable binary files, and so on – to your computer. There are several problems with these precompiled binary files. Most importantly, the precompiled binary might require other files on the original computer. For instance, if you compile the mplayer multimedia player, then you must have aalib1, xlibs, libncurses, and about twenty other software projects on your system. If you then copy the mplayer binary to another computer, all those other projects must be copied over also. Almost every project depends upon other projects. Before you can use the binaries from one software project, you need to know which other binaries to install – the project's dependencies.

The Debian Project overcame this problem by creating the Debian Package system. Debian Project volunteers compile the software projects and build all the binaries, documentation and so forth into a debian package file. A debian package file name always ends with “.deb” therefore these files are often called deb files. Deb files also include dependency information, so if you know how to read the deb file, then you will know what software must be installed before you can use this deb file. The deb file contains other information too. Debian servers around the world (<http://www.debian.org/mirror/list>) store the packages. After you download a debian package you can read its information using the `dpkg -I packagename` command, like this:

```
andrew@samwise:~$ dpkg -I /var/cache/apt/archives/mplayer-386_1%3a1.0-pre5-sarge0.3_i386.deb
new debian package, version 2.0.
size 3094844 bytes: control archive= 4168 bytes.
   98 bytes,    4 lines   control
  2086 bytes,   24 lines   control
  5814 bytes,   80 lines   md5sums
   738 bytes,   28 lines   * postinst      #!/bin/sh
   472 bytes,   22 lines   * postrm       #!/bin/sh
  3144 bytes,  144 lines   * preinst      #!/bin/sh
Package: mplayer-386
Version: 1:1.0-pre5-sarge0.3
Section: graphics
Priority: extra
Architecture: i386
Depends: aalib1 (>= 1.2), libartsc0 (>= 1.2.3), libasound2 (>= 1.0.5), libaudio2, libaudiofile0 (>= 0.2.3-4),
libc6 (>= 2.3.2.ds1-4), libcdparanoia0 (>= 3a9.8-11), libdirectfb-0.9-20, libdivxdecod0 (>= 1:5.0.1), libdv4,
libdvdrad3, libesd0 (>= 0.2.29-1) | libesd-alsa0 (>= 0.2.29-1), libfaad2-0 (>= 2.0.0-sarge0.0), libfontconfig1
(>= 2.2.1), libfreetype6 (>= 2.1.5-1), libfridi0, libggi2 (>= 1:2.0.4), libglib1.2 (>= 1.2.0), libglib2.0-0 (>=
2.4.1), libgtk1.2 (>= 1.2.10-4), libjpeg62, liblircclient0, liblzo1, libmad0 (>= 0.15.1b), libncurses5 (>= 5.4-
1), libogg0 (>= 1.1.0), libpng12-0 (>= 1.2.5.0-4), libpostproc0 (>= 0.90rc4), libSDL1.2debian (>= 1.2.7-0),
libsmblclient (>= 3.0.2a-1), libstdc++5 (>= 1:3.3.4-1), libsvgal | svgalib-dummyg1, libtheora0, libungif4g (>=
4.1.0b1), libvorbis0a (>= 1.0.1), libx11-6 | xlibs (>= 4.1.0), libxext6 | xlibs (>= 4.1.0), libxi6 | xlibs (>=
4.1.0), libxt6 | xlibs (>= 4.1.0), libxv1, libxvidcore4 (>= 1:1.0.0-rc4-0.0), slang1 (>= 1.4.9dbs-2), xlibmesa-g1
| libgl1, zlib1g (>= 1:1.2.1)
Recommends: mplayer-fonts
Suggests: w32codecs (>= 0.91-1), libvdcoss, mplayer-doc
Conflicts: mplayer-586, mplayer-686, mplayer-k6, mplayer-k7, mplayer-custom, mplayer-nogui
Replaces: mplayer-586, mplayer-686, mplayer-k6, mplayer-k7, mplayer-custom, mplayer-nogui
Provides: mplayer
Installed-Size: 7204
Maintainer: Christian Marillat <marillat.christian@wanadoo.fr>
Source: mplayer
Description: The Ultimate Movie Player For Linux
 It plays most mpeg, avi and asf files, supported by many native and win32
 DLL codecs. You can watch VCD, DVD and even DivX movies too. The another
 big feature of mplayer is the wide range of supported output drivers. It
 works with X11, Xv, DGA, OpenGL, SVGalib, fbdev, but you can use SDL (and
 this way all drivers of SDL) and some lowlevel card-specific drivers (for
 Matrox/3dfx/SiS) too! Most of them supports software or hardware scaling,
 so you can enjoy movies in fullscreen.
.
This version is for all CPUs.
```

You can install debian packages individually using the `dpkg -i packagename` command, but we won't do that because there is an easier way.

Advanced Packaging Tool (apt)

Debian Packages helped, but as the number of packages grew and software became more modular, the number of dependencies grew enormously. You might download a package, find out that it required another package, download that other package, find out that it required a third package, etc. This could take some time, especially if there are multiple dependencies – and there usually are. Obviously it would be much faster if there were a software tool that would read all the dependencies files, figure it all out, and download all the packages automatically. This tool exists and it is called “apt”, the Advanced Packaging Tool.

To use apt, first you must define a list of debian package servers. This list is stored in the `/etc/apt/source.list` file. Here is an example of a `source.list` file:

```

# 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.

# Security updates for "stable"
deb http://security.debian.org stable/updates main contrib non-free
deb http://security.debian.org testing/updates main contrib non-free

# Stable
deb http://debian.yorku.ca/debian/non-US stable/non-US main contrib non-free
deb http://debian.yorku.ca/debian/ stable main contrib non-free

# Sources
deb-src http://debian.yorku.ca/debian/ stable main contrib non-free
deb-src http://debian.yorku.ca/debian/non-US stable/non-US main contrib non-free

# Testing
deb http://debian.yorku.ca/debian/ testing main contrib non-free
deb http://debian.yorku.ca/debian/non-US testing/non-US main contrib non-free

# Sources
deb-src http://debian.yorku.ca/debian/ testing main contrib non-free
deb-src http://debian.yorku.ca/debian/non-US testing/non-US main contrib non-free

# Unstable
deb http://debian.yorku.ca/debian/ unstable main contrib non-free
deb http://debian.yorku.ca/debian/non-US unstable/non-US main contrib non-free

# Sources
deb-src http://debian.yorku.ca/debian unstable main contrib non-free
deb-src http://debian.yorku.ca/debian/non-US unstable/non-US main contrib non-free

# Java
deb ftp://ftp.gwdg.de/pub/languages/java/linux/debian woody main non-free
deb-src ftp://ftp.gwdg.de/pub/languages/java/linux/debian woody main non-free
deb ftp://ftp.gwdg.de/pub/languages/java/linux/debian unstable main non-free
deb-src ftp://ftp.gwdg.de/pub/languages/java/linux/debian unstable main non-free

# k3b
deb http://www.planet-moll.de/debian woody main
deb http://www.planet-moll.de/debian sarge main

# mplayer and other video stuff
deb ftp://ftp.nerim.net/debian-marillat/ testing main
deb ftp://ftp.nerim.net/debian-marillat/ unstable main

#end

```

Debian has three distributions: stable (also known as woody), testing (aka sarge) and unstable (aka sid). Notice that the file defines source for each version of debian. I happen to live in Canada, near Toronto, so I have used a debian archive located in Toronto. If you live somewhere else then you could find a closer server on the list of debian mirrors (<http://www.debian.org/mirror/list>) and edit the source.list file.

Also notice that some software packages, such as Java, k3b and mplayer are not included in the standard Debian servers. But other people have put their own debian servers on the internet, and you can add their server names to your source.list file. Often these servers are not as reliable as the standard archives, so you might need to comment them out if they cause problems.

Once you have a source.list then apt must copy the package lists and the dependency lists from the source servers to the apt cache on your computer. To do this, use the **apt-get update** command, like this:

```

root@samwise:/home/andrew# apt-get update
Hit http://debian.yorku.ca stable/non-US/main Packages
Hit http://debian.yorku.ca stable/non-US/main Release
... deletia ...

Fetched 9364kB in 35s (260kB/s)
Reading Package Lists... Done

```

This can take a long time on a dial-up internet connection. And you will have to repeat the update from time to time because packages are constantly being upgraded on the debian servers.

Using apt with a CD-ROM

apt works best with high speed internet connections. But you can use cd-roms with apt. You cannot add cd-rom sources to the sources.list file manually, instead you must use the **apt-cdrom** command and follow the instructions on the screen.

Installing Software using Apt

Installing software using apt is extremely simple: just use the **apt-get install** command, like this:

```

root@test36:/home/andrew# apt-get install mplayer-686
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
 libdirectfb-0.9-20 libdiyxdec0re0 libdvdread3 libfaad2-0 libfribidi0 libggi2 libgii0 libgii0-target-x
 liblame0 libpostproc0 libtheora0 libxvidcore4 mplayer-586
Suggested packages:
 libdvcss2 libggi-target-emu libggi-target-monotext libggimisc2 w32codecs libdvcss mplayer-doc
Recommended packages:

```

```

libggi-target-x libggi-target mplayer-fonts
The following NEW packages will be installed:
libdirectfb-0.9-20 libdivxdec0 libdvdread3 libfaad2-0 libfribidi0 libggi2 libggi0 libggi0-target-x
liblame0 libpostproc0 libtheora0 libxvidcore4 mplayer-586 mplayer-686
0 upgraded, 14 newly installed, 0 to remove and 196 not upgraded.
Need to get 4713kB of archives.
After unpacking 12.0MB of additional disk space will be used.
Do you want to continue? [Y/n]

```

apt tells you which packages will be installed, what additional packages might be installed, and gives some summary information. Read the list of suggested and recommended packages carefully. For instance, notice that the **libdvdcss2** package is on the suggested list, not the “will install” list. That means that mplayer won’t be able to play commercial DVDs. So you should probably install the **libdvdcss2** package later. Same goes for **mplayer-fonts** and **mplayer-doc**. But for now we press the Enter key to continue.

```

Get:1 http://debian.yorku.ca testing/main libdirectfb-0.9-20 0.9.20-1 [466kB]
Get:2 ftp://ftp.nerim.net testing/main libdivxdec0 1:5.0.1-1 [130kB]
Get:3 http://debian.yorku.ca testing/main libdvdread3 0.9.4-5 [53.8kB]
Get:4 http://debian.yorku.ca testing/main libfribidi0 0.10.4-3 [42.4kB]
Get:5 http://debian.yorku.ca testing/main libggi0-target-x 1:0.8.5-2 [17.4kB]
Get:6 http://debian.yorku.ca testing/main libggi0 1:0.8.5-2 [126kB]
Get:7 http://debian.yorku.ca testing/main libggi2 1:2.0.4-3 [190kB]
Get:8 http://debian.yorku.ca testing/main libtheora0 0.0.0.alpha3-1 [61.4kB]

... deletia ...

Setting up libggi2 (2.0.4-3) ...
Setting up mplayer-586 (1.0-pre5-sarge0.3) ...
Setting up mplayer-686 (1.0-pre5-sarge0.2) ...

```

Installing a Specific Version of a Package

Suppose that there are many different versions of a package. Typically apt will install the most recent version of the package, but suppose that you want an older version. For instance, k3b is an excellent cd/dvd mastering software. Suppose you want to check which versions of k3b are available. Use the command **apt-cache showpkg k3b**

```

root@samwise:/home/andrew# apt-cache showpkg k3b
Package: k3b
Versions:
0.11.14-1(/var/lib/apt/lists/debian.yorku.ca_debian_dists_unstable_main_binary-i386_Packages)
0.11.12-1(/var/lib/apt/lists/debian.yorku.ca_debian_dists_testing_main_binary-i386_Packages)
0.11.12-1(/var/lib/apt/lists/www.planet-moll.de_debian_dists_woody_main_binary-i386_Packages)
0.11.12-1(/var/lib/apt/lists/www.planet-moll.de_debian_dists_sarge_main_binary-i386_Packages)
0.11.9-2(/var/lib/dpkg/status)

Reverse Depends:
k3b-i18n,k3b 0.10
k3b-i18n,k3b 0.10
kde-extras,k3b
Dependencies:
0.11.14-1 - k3blibs (2 0.11.14) kdelibs4 (2 4:3.2.3) libart-2.0-2 (2 2.3.16) libarts1 (2 1.2.3) libasound2 (4
1.0.5) libaudio2 (0 (null)) libaudiofile0 (2 0.2.3-4) libbc6 (2 2.3.2.ds1-4) libesd0 (18 0.2.29-1) libesd-alsa0 (2
0.2.29-1) libfam0c102 (0 (null)) libgcc1 (2 1:3.4.1-3) libglib2.0-0 (2 2.4.1) libice6 (16 (null)) xlibs (4 4.1.0)

... deletia ...

```

Notice the fourth line of output: Versions. There are three different versions available. Also notice the long path following the version numbers, which includes the important words “unstable”, “testing”, “woody” and “sarge”. remember that there are three different debian distributions: stable/woody, testing/sarge, and unstable/sid. You might also notice that the long path name includes the name of one of the debian archives defined in your **sources.list** file. This long pathname shows which version of k3b (if any) is available from each distribution at each source archive. So which version is currently installed on your system?

Use the **dpkg -l k3b** command

```

root@samwise:/home/andrew# dpkg -l k3b
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Installed/Config-files/Unpacked/Failed-config/Half-installed
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/ Name          Version          Description
+++-----
ii k3b             0.11.9-2         A sophisticated KDE cd burning application

```

So we are currently using version 0.11.9-2, which is the most recent version of k3b in the stable distribution. But the absolutely most recent version is 0.11.14-1 which is in the unstable branch. We decide that we want to use the 0.11.14-1 version. Then add the **=version_number** identifier after the packagename in the **apt-get** command:

```

root@samwise:/home/andrew# apt-get install k3b=0.11.14-1
Reading Package Lists... Done
Building Dependency Tree... Done
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.

Since you only requested a single operation it is extremely likely that
the package is simply not installable and a bug report against
that package should be filed.
The following information may help to resolve the situation:

```

```
The following packages have unmet dependencies:
 k3b: Depends: k3blibs (>= 0.11.14) but 0.11.12-1 is to be installed
        Depends: libqt3c102-mt (>= 3:3.3.2) but 3:3.2.3-4 is to be installed
E: Broken packages
```

In fact, it won't work. Most of the packages used by knoppix are from the stable distribution, but k3b version 0.11.14-1 is from unstable. Which means that the packages that k3b depends on are from the unstable distribution. So we have to upgrade these dependency packages also, like this

```
root@samwise:/home/andrew# apt-get --reinstall install k3b=0.11.14-1 k3blibs=0.11.14-1 libqt3c102-mt=3:3.3.3-4
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
 k3b k3blibs libgcc1 libqt3c102-mt
Suggested packages:
 normalize toolame libqt3c102-mt-psql libqt3c102-mt-mysql libqt3c102-mt-odbc
The following packages will be upgraded:
 k3b k3blibs libgcc1 libqt3c102-mt
4 upgraded, 0 newly installed, 0 to remove and 311 not upgraded.
Need to get 6497kB of archives.
After unpacking 279kB disk space will be freed.
Do you want to continue? [Y/n]
```

As you can see, it gets a little bit complicated to specify specific package versions. And it may not be wise to mix packages from different distributions. And besides, there is an easy way to upgrade all installed packages to their most recent versions.

Upgrading All Software Packages

Installing individual packages by version number might be difficult, but upgrading all packages to the most recent version is very easy. Just use the **apt-get upgrade** command. Apt will upgrade every installed package to the most recent version in whatever distribution you are running. Maybe you don't know which debian distribution you are using (sometimes debian masquerades under a different name, such as knoppix or libranet or Xandros). To find out, have a look at the `/etc/apt/apt.conf` file.

```
andrew@samwise:~$ cat /etc/apt/apt.conf
APT::Default-Release "testing";
APT::Cache-Limit 10000000;
Apt::Get::Purge;
```

Searching for Packages

Suppose you know that you want to install a certain software project but you don't know its package name. No problem. You can search the APT system for the name of the project. For instance, suppose that you want to see flash animations in Mozilla. So you type in the command **apt-get install flash**, like this:

```
root@samwise:/home/andrew# apt-get install flash
Reading Package Lists... Done
Building Dependency Tree... Done
E: Couldn't find package flash
```

Oops, it's not named flash. So you search the apt-cache for the word flash using the **apt-cache search flash** command:

```
root@samwise:/home/andrew# apt-cache search flash
avr - Programmer for Atmel AVR microcontrollers
libming - Library to generate SWF (Flash) Files
libming-dev - Library to generate SWF (Flash) Files (development files)
libming-fonts-openoffice - Fonts for use with the Ming Library for SWF Creation
libming-util - Library to generate SWF (Flash) Files - Utilities
libswf-perl - Ming (SWF) module for Perl
php4-ming - Ming (SWF) module for php4
python-ming - Ming (SWF) module for Python
python2.2-ming - Ming (SWF) module for Python 2.2
adzapper - squid advertisement zapper
aget - Multithreaded HTTP Download Accelerator
flashybird - automates use of a flash disk as the root filesystem
ftdi-eprom - Tool for reading/erasing/flashing FTDI USB chip eeproms
fujiplay - Interface for Fuji digital cameras
libflash-dev - GPL Flash (SWF) Library
libflash0 - GPL Flash (SWF) library
liborange - library to extracts CAB files from self-extracting installers
liborange-dev - development libraries for liborange
libroxen-flash2 - Flash2 module for the Roxen Challenger web server
libswfdec-dev - SWF (Macromedia Flash) decoder library
libswfdec0 - SWF (Macromedia Flash) decoder library
libtk-tablematrix-perl - Table/matrix widget extension to Perl/Tk
mathwar - A flash card game designed to teach maths
mtd-tools - Memory Technology Device Tools
newsflash - gets news with the newnews command from a server
openwinice-jtag - allows programming jtag capable devices such as CPUs or FPGAs
orange - extracts CAB files from self-extracting installers
osflash - Reflash the OS of a Palm Computing Device
ploticus - A script driven business graphics package
ratpoison - Simple window manager with no fat library dependencies.
redboot - Red Hat Embedded Debug and Bootstrap firmware
swf-player - SWF (Macromedia Flash) player
vrfash - tool to flash kernels and romdisks to Agenda VR
xmms-jess - visualization plugin for XMMS using various 2D and 3D methods
flashplugin-nonfree - Macromedia Flash Player plugin installer
microcode.ctl - Intel IA32 CPU Microcode Utility
flashkard - flash card learning tool for KDE
```

```

gstreamer-swfdec - SWF (Macromedia Flash) decoder plugin for GStreamer
gstreamer0.8-swfdec - SWF (Macromedia Flash) decoder plugin for GStreamer
ifp-line - command line tool to access iRiver iFP audio players
keduca - interactive form-based tests for KDE
kvocctrain - vocabulary trainer for KDE
op-rotation-fb - Rotation settings for Opie
flashplayer-mozilla - Macromedia Flash Player

```

So you find out that 45 different packages include the word “flash” somewhere in the description. Free software is a good thing, but it is a little annoying when you have to read through a long list of package names to find the package you want. And of course, the package that you want to install is always the last one in the list (i.e. **flashplayer-mozilla**). A true linux guru might have piped the output of the **apt-cache** command to the **grep** command, like this:

```

andrew@samwise:~$ apt-cache search flash | grep player
swf-player - SWF (Macromedia Flash) player
ifp-line - command line tool to access iRiver iFP audio players
flashplayer-mozilla - Macromedia Flash Player

```

Listing All Installed Packages

To list all installed packages use the **dpkg -l | more** command:

```

andrew@samwise:~$ dpkg -l | more
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Installed/Config-files/Unpacked/Failed-config/Half-installed
| / Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/ Name          Version          Description
++-----+-----+-----+
ii  3270-common     3.3.2p1-1       Common files for IBM 3270 emulators {x,c,s,t
ii  a2ps            4.13b-2         GNU a2ps - 'Anything to PostScript' convert
ii  aalib-bin       1.4p5-20       sample programs using aalib
ii  aalib          1.4p5-20       ascii art library
ii  ace-of-penguin  1.2-6          Solitaire-games with penguin-look
ii  acpi            0.07-3         displays information on ACPI devices
ii  acroread        4.05-5         Adobe Acrobat Reader: Portable Document Form
ii  adduser         3.57           Add and remove users and groups
ii  afio            2.5-2          Archive file manipulation program
ii  aircsnort       0.2.4a-1       WLAN sniffer
ii  alevt          1.6.1-5        X11 Teletext/Videotext browser

... deletia ...

```

The two “i”s indicate the desired status and the actual status (in this case, both status are “installed”). Then the name of the package. Then the version number of the package. And then the first line of the package description.

Removing Packages

When you install knoppix to your hard drive you automatically install all the Knoppix packages. But you might not need them all. You could free up disk space by removing packages. For instance, if you don't have a cd burner you could remove the k3b application

```

root@samwise:/home/andrew# apt-get remove k3b
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
 k3b
0 upgraded, 0 newly installed, 1 to remove and 313 not upgraded.
Need to get 0B of archives.
After unpacking 4846kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 104059 files and directories currently installed.)
Removing k3b ...

```

Orphan Packages

When you install a certain package the apt system will automatically install the dependencies. But when you remove the package apt won't automatically remove that packages dependencies, because other packages might depend on them also. It all gets a bit complicated. But you can use the **deborphan** command to search the entire list of installed packages and list those packages which are (probably!) no longer required

```

root@samwise:/home/andrew# deborphan
k3blibs

```

The **k3blibs** package is no longer required because we removed **k3b** in the last section. So we can remove **k3blibs** too.

/var/cache/apt/archive

When apt downloads packages from the source archives apt stores the packages in the **/var/cache/apt/archive** directory. After a while (especially after doing an upgrade) this directory might be very large. You can clean it out using the **apt-get clean** command.

Installing Source Packages

Suppose you install the binary, but it segfaults. You need to compile it on your own machine. No problem! First, download all the build dependencies using the **apt-get build-dep packagename** command. Then download and build the source package, **apt-get -b source packagename**. Finally, install the new deb file, **dpkg -i packagename-version.deb**. Putting it all together:

```
apt-get build-dep packagename
apt-get -b source packagename
dpkg -i docbook-utils_0.6.13-7_all.deb
```

How to copy a debian installation

Use "dpkg --get-selections > selections.txt" on the original system.

Copy the "selections.txt" file to your target system

Run "dpkg --set-selections < selections.txt" on the target system to tell it which packages you want installed.

Finally, "apt-get dselect-upgrade" will download and install the packages.

Fifteen Commands you Need to Know

Install a debian package manually	dpkg -i packagename
List of debian archives	/etc/apt/sources.list
apt config file	/etc/apt/apt.conf
add a cdrom to the sources file	apt-cdrom
Update the apt cache	apt-get update
Install a package	apt-get install packagename
What versions of a package are available?	apt-cache showpkg packagename
What version is installed?	dpkg -l packagename
Install a specific version	apt-get install packagename=version
Upgrade all packages	apt-get upgrade
Search for a package	apt-cache search searchstring
List all installed packages	dpkg -l more
Remove a package	apt-get remove packagename
List orphaned packages	debtorphan
Empty the local package cache	apt-get clean

Graphical Package Managers

Now that you know how it works, you can go ahead and use a graphical package manager. As far as I know, the best one is kpackage, shown below.

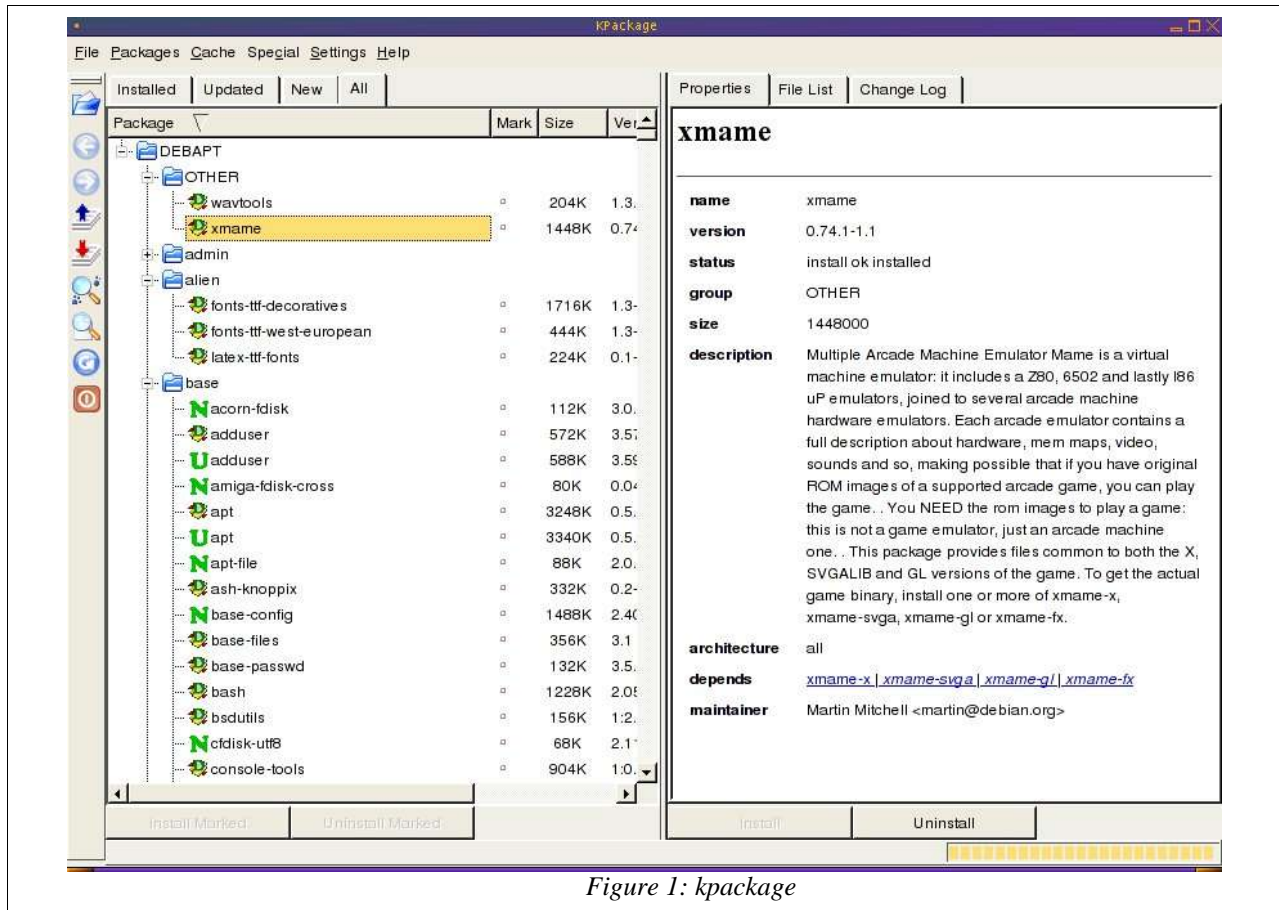


Figure 1: kpackage