How to list all installed packages

Asked 8 years, 11 months ago   Active 3 months ago   Viewed 3.4m times

▲

1915

▼

★

801

I'd like to output a list of all installed packages into a text file so that I can review it and bulk-install on another system. How would I do this?

`apt`   `package-management`

edited Jun 8 '16 at 17:48   asked Dec 17 '10 at 2:08
anonymous2   **Ivan**
3,567   4   24   52   **23.1k**   60   138   200

---

7   This was already answered here. – Simon Quigley Aug 20 '15 at 6:35

2   Wow, going through the answers there seems to be so many ways to accomplish this :/ – Madivad Dec 30 '15 at 21:53

2   How do I do the bulk installation on a new system from such a text file? – Elysium Oct 30 '16 at 23:02

2   @Madivad these packages managers were once supposed to make life easier... – René Nyffenegger Nov 26 '16 at 8:21

2   Similar question at unix.stackexchange.com "List packages by installation date" – JamesThomasMoon1979 May 18 '17 at 19:14 ✏

24 Answers

▲

2169

Ubuntu 14.04 and above

The `apt` tool on Ubuntu 14.04 and above makes this very easy.

Older Versions

To get a list of packages installed locally do this in your terminal:

```
dpkg --get-selections | grep -v deinstall
```

(The `-v` tag "inverts" grep to return non-matching lines)

To get a list of a specific package installed:

```
dpkg --get-selections | grep postgres
```

To save that list to a text file called `packages` on your desktop do this in your terminal:

```
dpkg --get-selections | grep -v deinstall > ~/Desktop/packages
```

Alternatively, simply use

```
dpkg -l
```

(you don't need to run any of these commands as the superuser, so no `sudo` or any other variants necessary here)

edited Aug 2 '17 at 5:21	answered Dec 17 '10 at 3:02
fosslinux	**Sabacon**
2,501 2 22 38	**29.1k** 5 30 39

---

37   Then, when installing to the new computer, do `cat ~/Desktop/packages > sudo dpkg --set-selections && sudo apt-get dselect upgrade`
      source – koanhead Dec 17 '10 at 5:51 ✏

---

13   The syntax for dpkg may have changed slightly since 2006 because that command didn't work for me, @koanhead. `sudo dpkg --set-selections < ~/Desktop/packages && sudo apt-get -u dselect-upgrade` does the trick. – James Feb 17 '11 at 20:44

---

222   Note that this won't keep track of which packages were explicitly installed by the user and which were installed as dependencies. This means that if

73   It's sad and amazing that @intuited 's comment is not better understood by the community. Would the more "highly rated" contributors stop to think more before giving the advice that simply reloading old packages on a new ubuntu version is not a good idea(?). With all the dpkg options I still do not see one that pulls out the 'expressely' installed packages by a user in order that THAT list can be reloaded and allowed to have its dependencies installed; I would love to know it - please share that info. In the meantime, make a list of the packages you really need on a re-install and run that – Ricalsin Jul 17 '13 at 17:42

62   We all know that linux discriminates people without photographic memory, but seriously, what would be wrong with `apt list installed` . Someday we should stop systematically failing the "keep simple things easy" principle. – user77164 Nov 19 '13 at 20:31 ✏

To get just the packages which were expressly installed (not just installed as dependencies), you can run

416

```
aptitude search '~i!~M'
```

This will also include a brief description, which you may want. If not, also add the option `-F '%p'` , as mentioned by karthick87.

Yet another option seems to be to copy the file `/var/lib/apt/extended_states` , which is a text file database in this format:

```
Package: grub-common
Architecture: amd64
Auto-Installed: 0

Package: linux-headers-2.6.35-22-generic
Architecture: amd64
Auto-Installed: 1
```

`Auto-Installed: 0` indicates that the package was expressly installed and is not just a dependency.

edited Mar 7 '18 at 8:51   answered Aug 28 '12 at 21:35

Sadi   intuited
9,198   5   41   51   **5,489**   3   23   33

---

52   This should be the correct answer. Why list dependencies? – Stavros Korokithakis Dec 8 '12 at 1:44

13   This **should** be the answer, but using *aptitude* is a bit unreliable because of Multiarch *currently* (fixes on the way), unfortunately. Still +1 for pointing out only listing explicitly installed packages and a way to do this (despite it won't work on 11.10+ currently). – gertvdijk Jan 11 '13 at 15:31

11   This lists all packages, not just manually installed packages on Ubuntu 13.10. – Eamon Nerbonne Oct 31 '13 at 14:33

5   is there a way to achieve the same result with `apt-get` ? – Javier Arias Jul 10 '17 at 13:27

To list all packages **intentionally installed** (not as dependencies) by apt commands, run the following :

▲

205

▼

```
(zcat $(ls -tr /var/log/apt/history.log*.gz); cat /var/log/apt/history.log) 2>/dev/null
|
  egrep '^(Start-Date:|Commandline:)' |
  grep -v aptdaemon |
  egrep '^Commandline:'
```

This provides a reverse time based view, with older commands listed first:

```
Commandline: apt-get install k3b
Commandline: apt-get install jhead
...
```

Installation data also showing synaptic usage, but without details (the same with installation date) :

```
(zcat $(ls -tr /var/log/apt/history.log*.gz); cat /var/log/apt/history.log) 2>/dev/null
|
  egrep '^(Start-Date:|Commandline:)' |
  grep -v aptdaemon |
  egrep -B1 '^Commandline:'
```

providing the following:

```
Start-Date: 2012-09-23  14:02:14
Commandline: apt-get install gparted
Start-Date: 2012-09-23  15:02:51
Commandline: apt-get install sysstat
...
```

1   **2,075**   1   8   2

3   Or (with zgrep and removing update messages): `zgrep -hE '^(Start-Date:|Commandline:)' $(ls -tr /var/log/apt/history.log*.gz ) | egrep -v 'aptdaemon|upgrade' | egrep -B1 '^Commandline:'` — belacqua May 17 '13 at 16:16 ✎

3   Does this approach miss packages installed with `dpkg` ?? — drevicko Jun 13 '13 at 23:27

5   While the above simple answers are good for the general user. This method by far is the best for backtracking all the customizations done to the machine, as it also shows what was removed, or added, from the base image, as it list them in the sequence it was performed, and helps you remember which is the correct sequence to add them back in another system. — AllGamer Jul 24 '14 at 16:27 ✎

3   @drevicko you are correct, it does not list packages that are installed with dpkg — Steve Buzonas Apr 14 '15 at 10:39

3   This is pretty handy for keeping organized-- my brain prefers it this way for whatever reason.. — JeremyFelix Apr 8 '16 at 15:36

▲

192

▼

Create a backup of what packages are currently installed:

```
dpkg --get-selections > list.txt
```

Then (on another system) restore installations from that list:

```
dpkg --clear-selections
sudo dpkg --set-selections < list.txt
```

To get rid of stale packages:

```
sudo apt-get autoremove
```

To get installed like at backup time (i.e. to install packages set by `dpkg --set-selections`):

```
sudo apt-get dselect-upgrade
```

edited Nov 3 '16 at 18:39   answered May 16 '12 at 18:26

🛡 wjandrea   🏠 gogaman

5	Great tip about restoring on another machine. +1. — Drew Noakes Jan 17 '13 at 0:28

15	migrating between 2 different ubuntu 12.04 machines this approach broke my system. it took a while until I realized that somehow `ubuntu-desktop` got uninstalled - probably due to one of the dpkg commands. be careful, it took me hours to repair the damage! — Karl Frisk Aug 19 '13 at 16:02

1	And sometimes system updates an app that you specifically wish system to not touch it ever, do fire this at end of commands `sudo apt-mark hold name-your-package` . This will prevent apt-get from upgrading to current version which is the default for updating process. — Faron Mar 1 '16 at 0:29 ✏

▲

78

▼

```
apt-mark showmanual
```

man pages state:

> will print a list of manually installed packages

So, it should just give a list of explicitly installed packages (though this includes packages that were part of the default initial install) without all of the dependencies included due to these packages being installed.

To output the result into a text file:

```
apt-mark showmanual > list-manually-installed.txt
```

edited Jan 18 at 2:00   answered Jul 10 '14 at 13:43

Pablo A   Tim Tisdall
4,405   3   22   49   **1,151**   9   11

	On Debian "apt-mark showmanual" didn't work on "squeeze" but worked on "wheezy" — Wadih M. Oct 5 '15 at 0:07

11	Like with most other answers, `apt-mark showmanual` doesn't really. It also lists tons of automatically installed packages, probably part of the base install. — mivk Nov 29 '15 at 22:05

8	Well the question was for installed packages and this gives all installed packages minus the automatically installed dependencies. It does include the initial packages as part of the initial install. I guess you could run this on a fresh install to get a list of the default installs and then subtract that from this to see the difference. — Tim Tisdall Nov 29 '15 at 23:54

5	No one mention this alternative to list manually installed packages: `apt list --manual-installed` . — Pablo A Jan 18 at 1:59 ✏

`dpkg-query` (instead of `dpkg --get-selections`, which lists some packages that are not installed) as follows:

41

```
dpkg-query -W -f='${PackageSpec} ${Status}\n' | grep installed |  sort -u | cut -f1 -d \
> installed-pkgs
```

Or:

```
dpkg -l | grep ^ii | sed 's_  _\t_g' | cut -f 2 > installed-pkgs
```

edited Feb 15 '17 at 12:18   answered May 16 '12 at 18:40

d a i s y   **kyleN**
3,691　8　27　49   **1,165**　8　7

Typying `# for pkg in `cat installed-pkgs`; do apt-get install -y $pkg; done` in the second system I have made this so that it will install Packages. – user9869932 Mar 11 '15 at 19:46 ✏

1　@julianromera: `apt-get install -y $(< installed-pkgs)` will make it so that apt-get runs just once and takes care of all dependencies at once. – jamadagni Nov 10 '15 at 8:29

@julianromera...correct me if I'm mistaken but to build an app with dependencies before doing the install usually would be `sudo apt-get build-dep name-package; sudo apt-get install name-package;` ? – Faron Mar 1 '16 at 0:39

To list all installed packages,

39

```
dpkg -l |awk '/^[hi]i/{print $2}' > 1.txt
```

or

```
aptitude search -F '%p' '~i' > 1.txt
```

or

```
dpkg --get-selections > 1.txt
```

Note:

You will get the result 1.txt file in your home folder or you can specify your own path.

edited Feb 15 '17 at 12:19   answered Dec 17 '10 at 3:35

d a i s y   karthick87

3,691   8   27   49   **54.3k**   56   179   225

35

You can use Synaptic to save the current state of your installed packaged. In Synaptic, select "file/save markings", Enter the name of the file to save the state to, and make sure to check the "Save full state, not only changes" box.

The file saved from this can be loaded into a new machine using "file/read markings" in Synaptic.

answered Dec 17 '10 at 4:46

Nerdfest

3,395   2   24   28

34

I recommend using blueprint. Even though it is designed for servers, it can be also used from desktops as well. It will create a shell script/chef/puppet that you ca use to re-install all you packages.

answered Apr 12 '11 at 12:26

Tamer

507   4   7

> I am not trying a complete mirror, but only the fact of taking config files into account is awesome. I was looking for such a tool for a while now, thank you very much! – tbolender Sep 5 '14 at 9:58

33

You want to reinstall the packages now there on 12.04, right?

If so, it's very easy. You'll need an "Ubuntu Single Sign On account." (Create it before reinstalling so that your system is synced.)

1. Go to the Software Center and look for the "Sync Between Computers..." option under the File menu.

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

4. You just have to sign in to your Ubuntu account and your previous computer will be shown.

5. Click on it; you'll get a list of all apps. Select "install" on the app you want to install.

edited Jun 20 '13 at 1:27   answered May 16 '12 at 18:26

andyg0808   Nirmik
193   5   **4,093**   15   52   85

I set the same name for my computer on a fresh installation to prevent from huge downloading from Ubuntu One server each time. Do you think that this work for me? – Alen Aug 17 '13 at 20:29
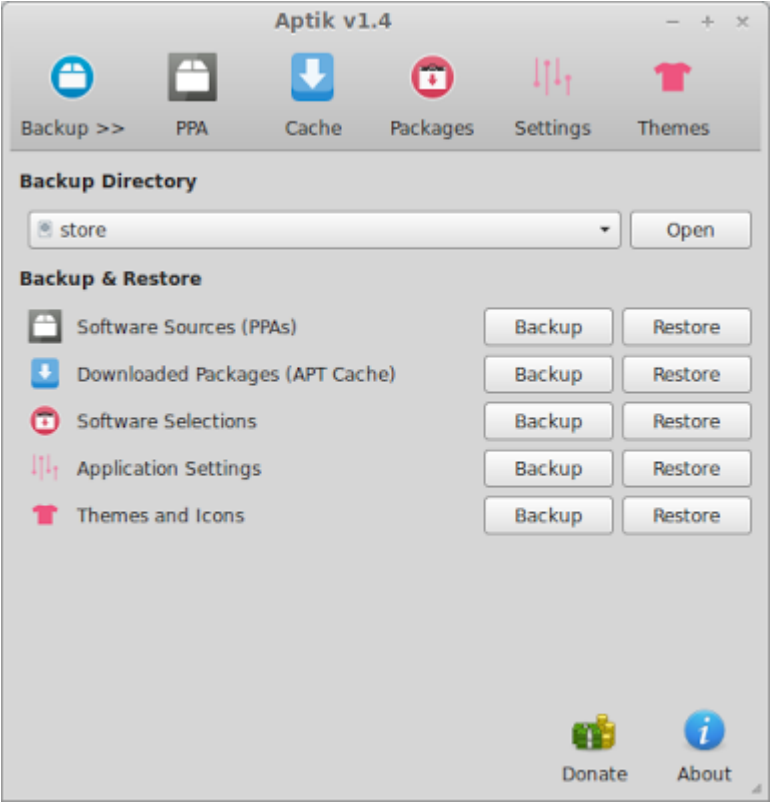
There's also a tool called **Aptik** (both command line and GUI) which can help you view a list of all installed packages, with an option to select/unselect some of them, make a backup list, and then restore the same set of packages in another system.

32

To install:

```
sudo add-apt-repository -y ppa:teejee2008/ppa
sudo apt-get update
sudo apt-get install aptik
```

Further info: https://teejeetech.in/aptik/

As can be seen in the screenshot, **Aptik** lets you **also backup and restore PPAs**, which will certainly be necessary to install some of the packages installed.

edited Apr 6 at 7:56   answered May 30 '14 at 20:22

smac89   Sadi
285   5   13   **9,198**   5   41   51

If you want the GUI, you have to install `apt-get install aptik-gtk` as well – Maduka Jayalath May 24 '18 at 6:14

APT-Clone. This package can be used to clone/restore the packages on a apt based system.

> **source:** <ins>man apt-clone</ins>

APT-Clone is used by `ubiquity` (Ubuntu installer) for upgrade process. It is much better than the `dpkg --get-selections` solution because:

1. It preserves all repositories information.
2. It keeps track of what packages were automatically installed.
3. It allows to repack locally installed DEB files.

How to Use

1. Install

```
sudo apt-get install apt-clone
```

2. Make backup

```
sudo apt-clone clone path-to/apt-clone-state-ubuntu-$(lsb_release -sr)-$(date
+%F).tar.gz
```

3. Restore backup

```
sudo apt-clone restore path-to/apt-clone-state-ubuntu.tar.gz
```

   Restore to newer release:

```
sudo apt-clone restore-new-distro path-to/apt-clone-state-ubuntu.tar.gz $(lsb_release -
sc)
```

It make simple gzipped tar file which can be easily edited and reviewed before restoring on the other machines. Here an example of its structure:

```
/
├── etc
│   └── apt
│       ├── preferences.d
```

```
│          │       ├── maarten-baert-ubuntu-simplescreenrecorder-vivid.list
│          │       └── megasync.list
│          ├── trusted.gpg
│          └── trusted.gpg.d
│              ├── anton__ubuntu_dnscrypt.gpg
│              ├── anton__ubuntu_dnscrypt.gpg~
│              ├── maarten-baert_ubuntu_simplescreenrecorder.gpg
│              └── maarten-baert_ubuntu_simplescreenrecorder.gpg~
└── var
    └── lib
        └── apt-clone
            ├── extended_states
            ├── installed.pkgs
            └── uname
```

answered Sep 5 '15 at 23:09

user.dz
37.4k   11   106   190

There's [a great explanation on Unix StackExchange](#) that describes how to use aptitude to list packages not installed as dependencies, and how to compare that list with the list of default packages for your Ubuntu release.

26

To obtain the manifest file for desktop versions of 12.04 and newer, visit [this site](#), choose your release, and scroll down below the CD images to the files section. You'll find something like "ubuntu-12.04.4-desktop-amd64+mac.manifest" that matches your architecture.

For server versions you'll need to obtain the manifest file from the ISO that was used to install the original system. For a VPS or cloud server, your provider may make the images available or you might need to contact them.

Here's an example using the code from the referenced post, along with modifications to install on the new server.

Old server (code from other post, output saved to file):

```
aptitude search '~i !~M' -F '%p' --disable-columns | sort -u > currently-installed.list
wget -qO - http://mirror.pnl.gov/releases/precise/ubuntu-12.04.3-desktop-amd64.manifest
\
  | cut -f1 | sort -u > default-installed.list
comm -23 currently-installed.list default-installed.list > user-installed.list
```

```
scp user@oldserver:user-installed.list .
sed -i 's/$/\tinstall/' user-installed.list
sudo dpkg --set-selections < user-installed.list
sudo apt-get dselect-upgrade
```

Before starting this task, I recommend reading and understanding all parts of the post mentioned in the beginning, and then consult the aptitude reference guide for details on search patterns.

edited Aug 13 '14 at 14:45   answered Feb 10 '14 at 16:35

bastistician   thinkmassive
153   1   4   613   5   6

You can look at the `apt` log under `/var/log/apt/` and the `dpkg` log under `/var/log/`

25

and you can get the list of the installed packages with just a command:

```
dpkg -l | grep '^ii '
```

edited Nov 29 '15 at 23:38   answered Mar 13 '14 at 10:36

mivk   Maythux
2,685   28   37   56k   35   188   235

I'm surprised the `apt-cache` command designed **exactly for this purpose** hasn't been mentioned above...

23

```
apt-cache pkgnames
```

For more info, run `apt-cache --help` :

```
**apt-cache is a low-level tool used to query information
from APT's binary cache files

Commands:
```

```
  stats - Show some basic statistics
  dump - Show the entire file in a terse form
  dumpavail - Print an available file to stdout
  unmet - Show unmet dependencies
  search - Search the package list for a regex pattern
  show - Show a readable record for the package
  depends - Show raw dependency information for a package
  rdepends - Show reverse dependency information for a package
  pkgnames - List the names of all packages in the system
  dotty - Generate package graphs for GraphViz
  xvcg - Generate package graphs for xvcg
  policy - Show policy settings

Options:
  -h   This help text.
  -p=? The package cache.
  -s=? The source cache.
  -q   Disable progress indicator.
  -i   Show only important deps for the unmet command.
  -c=? Read this configuration file
  -o=? Set an arbitrary configuration option, eg -o dir::cache=/tmp
See the apt-cache(8) and apt.conf(5) manual pages for more information.
**
```

answered Aug 14 '15 at 19:35

ostrokach
604 4 9

---

5   TLDR; I eventually found that `apt --installed list` gives best result (as suggested in top answer above. Although `apt-cache pkgnames` seems to do the trick at first glance, it lists "all packages in the system" (per the help text above), which also includes packages which apt knows about but aren't actually installed. There's a `--installed` option but it doesn't seem to work with `pkgnames`. — sxc731 Jan 1 '16 at 16:05 ✏

2   `apt-cache pkgnames | wc -l` gives `55909` packages (the system knows about), whereas `aptitude search '~i!~M' | wc -l` gives `2160` (packages explicitly installed, without dependencies). — knb Apr 11 '17 at 8:42

For the complete rundown see:

20

https://help.ubuntu.com/community/SwitchingToUbuntu/FromLinux/RedHatEnterpriseLinuxAndFedora#Command_Line_Tools

half way across the page:

```
dpkg --list
```

adriano72
489 4 3

The below command will also lists all the installed packages,

18

```
grep ' installed ' /var/log/dpkg.log /var/log/dpkg.log.1 | awk '{print $5}' | sort -u
```

Avinash Raj
56.2k 44 178 229

The .1 there implies the log was rotated, if that's the case then maybe it's better to grep `/var/log/dpkg.log*` to get all rotated logs. – Steve Buzonas
Apr 14 '15 at 11:55

Help out this community wiki - Add up-to-date solutions.

17

dpkg, xargs, & apt-get

This command should accomplish the creation of a text file containing installed packages:

```
dpkg -l | awk  '{print $2}' > package_list.txt
```

To accomplish the bulk installation of the listed packages you'll need to edit 'package_list.txt'. Remove the weird lines at the top of the file using a text editor. You can then use this command to install packages from the created file using:

```
xargs < package_list.txt apt-get install -y
```

apt-cache, xargs, & apt-get

Only use this method if you want **all** current packages to be installed using the list (**which includes** automatically installed, etc).

Output the response of 'apt-cache pkgnames' to a file we'll simply name "package_list.txt". You can accomplish this with:

```
apt-cache pkgnames > package_list.txt
```

Then when you want to install packages from "package_list.txt" you would use this command:

```
xargs < package_list.txt apt-get install -y
```

apt-mark, xargs, & apt-get

We can use the command `apt-mark showmanual` to give a list of packages that were manually or initially installed with Ubuntu. We'll want to output that to a file we'll just call "package-list.txt". Use this command to accomplish that:

```
apt-mark showmanual > package-list.txt
```

The command we would use to install packages from the file "package_list.txt" is below.

```
xargs < package_list.txt apt-get install -y
```

Aptik Migration Utility

> Utility to simplify re-installation of software packages after upgrading/re-installing Ubuntu-based distributions.
> [Launchpad | Aptik]

For information on Aptik, try visiting its **official page**, and for a screenshot click here or view the end of this section.
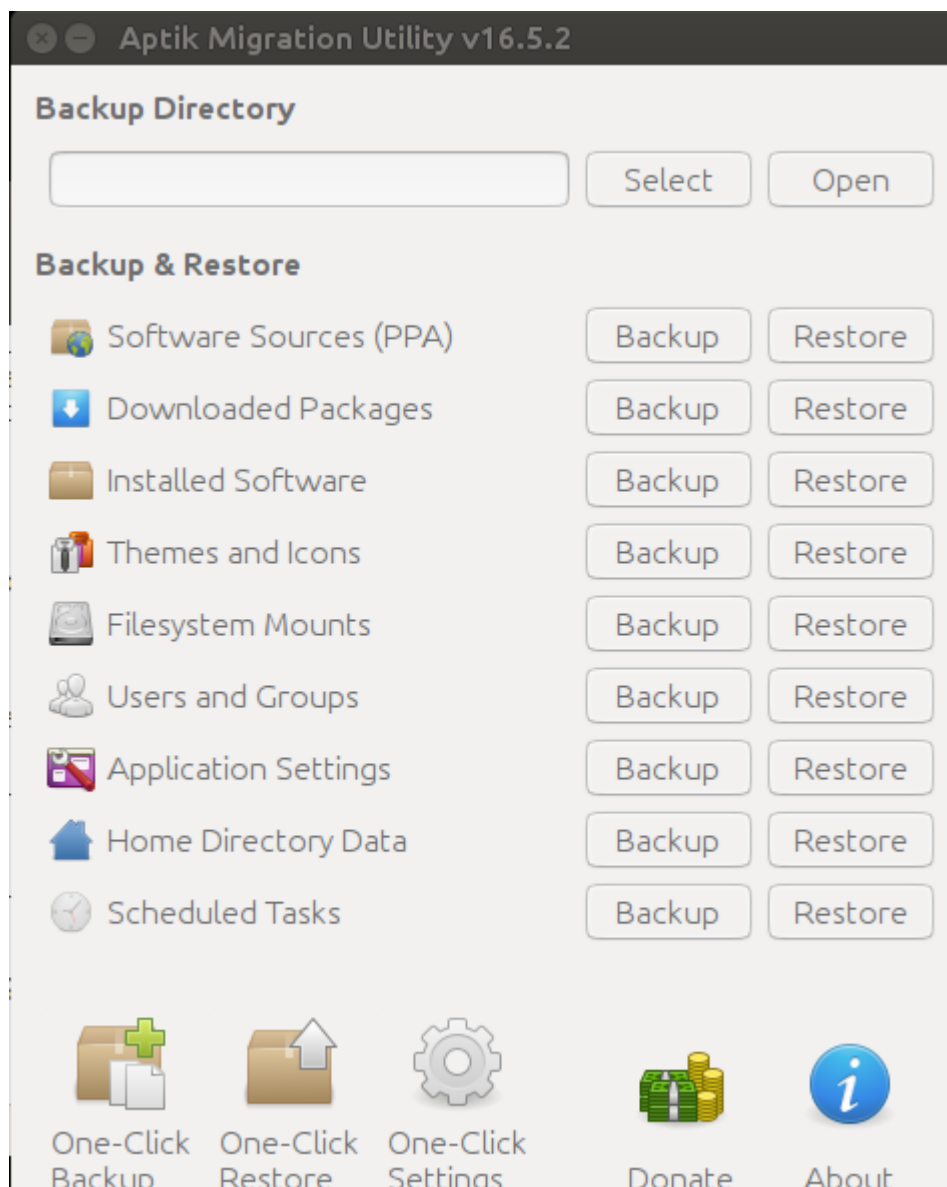
Installing Aptik is simple. Follow these steps:

2. **Update** apt with the below command.

```
sudo apt-get update
```

3. **Install** Aptik using:

```
sudo apt-get install aptik
```

edited Mar 31 at 2:43

community wiki
9 revs, 3 users 97%
David your friend

1 The output from `apt-mark showmanual` includes Bash, Unity, and Xorg, among others. Are they supposed to be there? – wjandrea Aug 16 '16 at 22:55

@wjandrea the output of 'apt-mark showmanual' includes Bash & Unity in my generated list, also. It should be normal as the command lists manually installed & initially installed w/ Ubuntu. – David your friend Aug 16 '16 at 22:59 ✎

To save a list of installed packages to a file named `installed_packages.txt` , just run:

16

```
dpkg-query --list >> installed_packages.txt
```

edited Oct 17 '16 at 16:16

answered May 12 '15 at 8:29

wb9688
852 1 13 26

I think it is interesting to note `apt list --installed` or `dpkg-query --list` actually use the file called `/var/lib/dpkg/status` in behind where all the info about the packages is beard.

3

So if you would like to deal with the super extended list of packages just `cat /var/lib/dpkg/status` .

Note: Do not alter `/var/lib/dpkg/status` file.

answered Aug 16 '18 at 19:44

prosti
449 4 10

In addition to **APT** packages, many GUI packages are nowadays distributed as **snaps**.

If your package can't be found in `apt list --installed` then try `snap list` .

```
$ snap list

Name                    Version                  Rev    Tracking  Publisher        Notes
gimp                    2.10.10                  165    stable    snapcrafters     -
gnome-calculator        3.32.1                   406    stable/…  canonical✓       -
keepassxc               2.4.1                    267    stable    keepassxreboot   -
...
```

It's also a good idea to add `/snap/bin` to the PATH so you can start those from the terminal (done automatically for non-root users).

answered Jun 8 at 8:56

rustyx
279 3 9

The package `dctrl-tools` provide the `grep-status` tool to get the list of the packages marked as installed on your system:

1

```
sudo apt install dctrl-tools
```

Usage:

```
grep-status -FStatus -sPackage -n   "install ok installed"
```

see : man dctrl-tools

answered Nov 22 '18 at 11:11

GAD3R
1,534 9 22

---

https://www.rosehosting.com/blog/list-all-installed-packages-with-apt-on-ubuntu/:

1

1. List the installed software packages on Ubuntu

```
sudo apt list --installed
```

The output of the command will be very similar to the following one, depending on which packages are currently installed:

```
Listing...
acl/xenial,now 2.2.52-3 amd64 [installed]
adduser/xenial,xenial,now 3.113+nmu3ubuntu4 all [installed]
apache2/xenial-updates,xenial-security,now 2.4.18-2ubuntu3.1 amd64 [installed]
apache2-bin/xenial-updates,xenial-security,now 2.4.18-2ubuntu3.1 amd64
[installed,automatic]
apache2-data/xenial-updates,xenial-updates,xenial-security,xenial-security,now 2.4.18-
2ubuntu3.1 all [installed,automatic]
apache2-doc/xenial-updates,xenial-updates,xenial-security,xenial-security,now 2.4.18-
2ubuntu3.1 all [installed]
apache2-utils/xenial-updates,xenial-security,now 2.4.18-2ubuntu3.1 amd64 [installed]
apparmor/xenial-updates,now 2.10.95-0ubuntu2.5 amd64 [installed,automatic]
apt/xenial-updates,now 1.2.19 amd64 [installed]
apt-utils/xenial-updates,now 1.2.19 amd64 [installed]
...
```

2. Use the LESS program

To easily read the entire output you can use the `less` program.

```
sudo apt list --installed | less
```

3. Use the GREP Command

You can look for a specific package through the output using the `grep` program.

```
sudo apt list --installed | grep -i apache
```

4. List all packages that include Apache

The output from the above command will list all packages that include `apache` in their names.

```
apache2/xenial-updates,xenial-security,now 2.4.18-2ubuntu3.1 amd64 [installed]
apache2-bin/xenial-updates,xenial-security,now 2.4.18-2ubuntu3.1 amd64
```

```
apache2-doc/xenial-updates,xenial-updates,xenial-security,xenial-security,now 2.4.18-
2ubuntu3.1 all [installed]
apache2-utils/xenial-updates,xenial-security,now 2.4.18-2ubuntu3.1 amd64 [installed]
libapache2-mod-php/xenial,xenial,now 1:7.0+35ubuntu6 all [installed,automatic]
libapache2-mod-php7.0/xenial-updates,now 7.0.13-0ubuntu0.16.04.1 amd64
[installed,automatic]
libapache2-mod-security2/xenial,now 2.9.0-1 amd64 [installed]
libapache2-modsecurity/xenial,xenial,now 2.9.0-1 all [installed]
```

Apt supports patterns to match package names and options to list installed `(--installed)` packages, upgradeable `(--upgradeable)` packages or all available `(--all-versions)` package versions.

5. Use the DPKG program

Another alternative that you can use to list the installed software packages on your **Ubuntu VPS** is the `dpkg` command.

```
sudo dpkg -l
```

The output of the command will provide you with information such as the name of the package, version, architecture and short description about the package. Of course, you can use the `grep` program again to search for a specific package.

```
sudo dpkg -l | grep -i apache
```

The output should look like the one below:

```
ii  apache2                 2.4.18-2ubuntu3.1              amd64
Apache HTTP Server
ii  apache2-bin             2.4.18-2ubuntu3.1              amd64
Apache HTTP Server (modules and other binary files)
ii  apache2-data            2.4.18-2ubuntu3.1              all
Apache HTTP Server (common files)
ii  apache2-doc             2.4.18-2ubuntu3.1              all
Apache HTTP Server (on-site documentation)
ii  apache2-utils           2.4.18-2ubuntu3.1              amd64
Apache HTTP Server (utility programs for web servers)
rc  apache2.2-common        2.2.22-6ubuntu5.1             amd64
Apache HTTP Server common files
ii  libapache2-mod-php      1:7.0+35ubuntu6               all
server-side, HTML-embedded scripting language (Apache 2 module) (default)
rc  libapache2-mod-php5     5.5.9+dfsg-1ubuntu4.16        amd64
server-side, HTML-embedded scripting language (Apache 2 module)
```

```
Tighten web applications security for Apache
ii  libapache2-modsecurity      2.9.0-1                     all
Dummy transitional package
ii  libapr1:amd64               1.5.2-3                     amd64
Apache Portable Runtime Library
ii  libaprutil1:amd64           1.5.4-1build1               amd64
Apache Portable Runtime Utility Library
ii  libaprutil1-dbd-sqlite3:amd64 1.5.4-1build1             amd64
Apache Portable Runtime Utility Library - SQLite3 Driver
ii  libaprutil1-ldap:amd64      1.5.4-1build1               amd64
Apache Portable Runtime Utility Library - LDAP Driver
.
```

With the competition of this tutorial, you have successfully learned how to list installed packages in Ubuntu.

edited Jul 17 at 12:04   answered Oct 19 '18 at 6:54

terdon ♦   M.A.K. Ripon
75.5k 14 152 238   **1,601** 1 12 29

There are many ways to do that. In case you are using CentOS like me you can use these: 1. `yum list installed` 2. `rpm -qa`

0   answered Jan 31 at 21:03

                                           saeed mohammadi
                                           24 1 6

---

protected by Mitch ♦ Nov 5 '14 at 12:53

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these unanswered questions instead?