

Ask Ubuntu is a question and answer site for Ubuntu users and developers. Join them; it only takes a minute:

Here's how it works:

Sign up

Anybody can ask a question

Anybody can answer

The best answers are voted up and rise to the top

How to list all installed packages

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



anonymous2

3,015 3 17 46

asked Dec 17 '10 at 2:08



Ivan

19.3k 54 121 193

- 4 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
- 1 How do I do the bulk installation on a new system from such a text file? – [Elysium](#) Oct 30 '16 at 23:02
- 1 @Madivad these packages managers were once supposed to make life easier... – [René Nyffenegger](#) Nov 26 '16 at 8:21
- 1 Similar question at unix.stackexchange.com "List packages by installation date" – [JamesThomasMoon1979](#) May 18 '17 at 19:14

19 Answers

Ubuntu 14.04 and above

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

```
apt list --installed
```

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



ubashu

2,049 1 15 31

answered Dec 17 '10 at 3:02



Sabacon

22.7k 4 27 37

- 34 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
-
- 185 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 you use this method to recreate your setup on another machine, apt won't be able to remove unneeded dependencies when you remove a given package. – intuited Aug 28 '12 at 21:03
-
- 63 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
-
- 57 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. – nus Nov 19 '13 at 20:31
-

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

```
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.

<p>edited Mar 7 at 8:51</p>  <p>Sadi 8,317 4 37 45</p>	<p>answered Aug 28 '12 at 21:35</p>  <p>intuited 4,829 3 20 29</p>
--	--

- 41 This should be the correct answer. Why list dependencies? – Stavros Korokithakis Dec 8 '12 at 1:44
-
- 12 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
-
- 4 @intuited "Fix Released" for Precise (some time ago).: – gertvdijk Jun 20 '13 at 20:10
-
- 9 This lists all packages, not just manually installed packages on Ubuntu 13.10. – Eamon Nerbonne Oct 31 '13 at 14:33
-
- 3 on Ubuntu-15.10, must install aptitude first *eg*: `sudo apt-get install aptitude` – Mark Mikofski Apr 9 '16 at 16:00
-

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
```

<p>edited Nov 3 '16 at 18:39</p>  <p>wjandrea 5,946 4 17 50</p>	<p>answered May 16 '12 at 18:26</p>  <p>gogaman 1,885 1 7 3</p>
---	---

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

12 sudo is unnecessary to list packages – Nick T Jul 11 '13 at 21:17

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

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

```
(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
...
```

edited Feb 23 '17 at 2:41

 **muru**
120k 18 248 429

answered Feb 2 '13 at 10:39

 **bcl**
1,705 1 7 2

1 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

2 Does this approach miss packages installed with dpkg ?? – drevicko Jun 13 '13 at 23:27

4 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

@AllGamer My only problem is that /var/log/ is a memory mapped directory on my machine, so I cannot use this approach... – Ali Aug 16 '14 at 20:44

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

```
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-installed.txt
```

edited Nov 29 '15 at 23:55

answered Jul 10 '14 at 13:43



Tim Tisdall
811 7 10

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

- 9 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
- 4 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

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

```
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

 **daisy**
2,692 5 20 35

answered May 16 '12 at 18:40

 **kyleN**
1,115 8 7

Typing # 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. – rxx Mar 11 '15 at 19:46

- 1 @julianromera: `apt-get install -y $(cat 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,

```
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

 **daisy**
2,692 5 20 35

answered Dec 17 '10 at 3:35

 **karthick87**
42k 49 160 214

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**
3,137 2 21 26

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**
487 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

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.
2. When you click on it you will see your computer registered and a list of all apps on your computer.
3. When you will install fresh, that computer will be considered a new computer.
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



[andyg0808](#)
193 5

answered May 16 '12 at 18:26



[Nirmik](#)
3,604 13 49 81

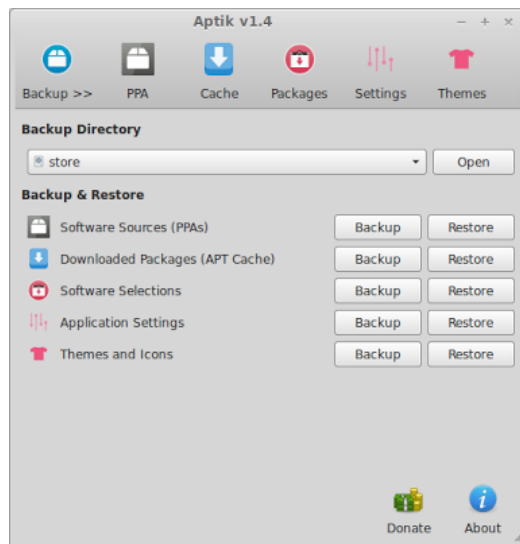
I set the same name for my computer on a fresh installation to prevent from huge downloading from Ubuntu One server each time. Do you think that this work for me? – [AliNā](#) Aug 17 '13 at 20:29

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.

To install:

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

Further info: <http://www.teejeetech.in/2014/04/aptik-v14.html>



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.

answered May 30 '14 at 20:22



[Sadi](#)
8,317 4 37 45

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

- It will save/restore the packages, sources.list, keyring and automatic-installed states.
- It can also save/restore no longer downloadable packages using dpkg-repack.

source: [man apt-clone](#)

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
│       ├── sources.list
│       ├── sources.list.d
│       │   ├── anton__ubuntu-dnscrypt-vivid.list
│       │   ├── 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](#)

32.3k 11 79 157

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.

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
```

On the new server, copy the file using scp, then use sed to append 'install' to every line (-i performs an inline-replace). Then you can use the list as input to 'dpkg --set-selections' and install the packages with apt-get:

```
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

 **Wastl**
153 1 4

answered Feb 10 '14 at 16:35

 **thinkmassive**
483 4 5

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

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

 **mivk**
2,034 21 29

answered Mar 13 '14 at 10:36

 **Maythux**
44.6k 27 148 199

For the complete rundown see:

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

half way across the page:

```
dpkg --list
```

answered Jul 31 '13 at 13:21

 **adriano72**
409 4 3

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

```
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:

```
gencaches - Build both the package and source cache
showpkg - Show some general information for a single package
showsrc - Show source records
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
456 3 7

- 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

The below command will also lists all the installed packages,

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

answered May 30 '14 at 18:36



Avinash Raj
47.4k 38 156 200

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

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

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

edited Oct 17 '16 at 16:16

answered May 12 '15 at 8:29



wb9688
716 1 9 23

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

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

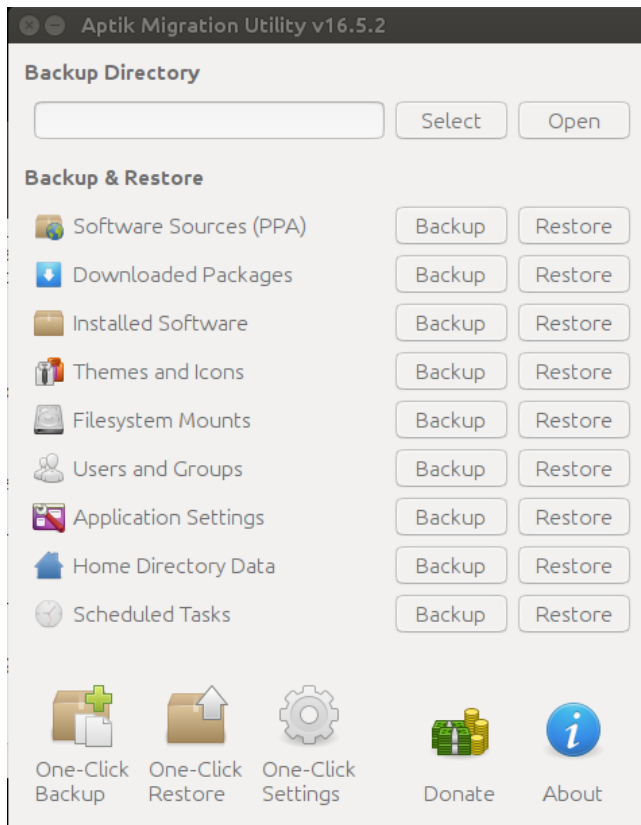
Utility to simplify re-installation of software packages after upgrading/re-installing Ubuntu-based distributions.

[\[Launchpad | Aptik\]](#)

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:

1. **Add** the PPA with:
`sudo add-apt-repository -y ppa:teejee2008/ppa`
2. **Update** apt with the below command.
`sudo apt-get update`
3. **Install** Aptik using:
`sudo apt-get install aptik`



edited Sep 5 '17 at 9:39

community wiki
7 revs
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

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?

