

# How To: Use 'apt-cache' to Find Homepage for Your Installed Apps

It can come in pretty handy to know for certain the homepage for the applications you have installed. You can do this with 'apt-cache'. I'll show you how. This is a pretty easy article to follow and just another tool to add to your toolbox.

**NOTE:** This is only valid for systems that use apt. As the title indicates, it requires 'apt-cache'. Without apt-cache, this page will do you no good. None good. That's how much it will do you. None!

Why would you want to know the homepage – and, more so, the preferred homepage? For starters, in the days of GitHub and everyone forking, and awkward application names that aren't easily searched for, it's hard to know which site is the correct one.

Maybe you want to report a bug? Maybe you want to request a feature? Maybe you want to make a donation? Maybe you just want to thank the author for writing such awesome software? Maybe you want to know where the homepage is because you need support and you're not sure where to turn to?

There are all sorts of reasons why you might want to know the homepage of a piece of software. It's actually something that's important. It's also something your system already knows and will happily show you if you know the proper magical incantation.

Like many other articles, you're gonna want the terminal for this. Let's go ahead and get that opened by using your keyboard and pressing CTRL + ALT + T.

Got your terminal emulator open? Good! Let's start with the command.

```
[code]apt-cache show inxi[/code]
```

If you do not have 'inxi' installed, feel free to use a different application. Note that you do not need to use sudo for this. Not all apt commands require sudo. You only need sudo when you're actually doing administrative tasks. See? I saved you some typing!

Anyhow, in the text output from the above command you'll see a line that starts with "Homepage:". If you hadn't already guessed it, that's the line that tells you the authors homepage. This, of course, only works on installed applications. For sanity and space sake, it's not like your system has all that information downloaded for all the possible packages. Thus, it works on naught but the apps you have already installed.

So, let's go ahead and make the command a little more precise. We'll pipe the output through grep and get rid of the cruft we don't actually need. In that same terminal, go ahead and enter:

```
[code]apt-cache show inxi | grep Homepage[/code]
```

**NOTE:** The command contains a capitalized letter H because Linux is often case-sensitive and is certainly case-sensitive in this case. If you don't believe me, try it with a lowercase h!

But wait, there's more!

Not only is there homepage information in there, there's sometimes some useful nuggets of information in there. If you have LibreOffice installed, go ahead and check (skip the pipe and grepping) to see what the output is. Inside, it has a ton of additional information, including listing ways that you can

extend LibreOffice by installing more software.

And there you have it. You can now easily find the homepage for the applications you have installed. Should you need to contact the author, check for information, or just see if they did anything else, you now know how to do that. It's a little hidden nugget that most folks don't seem to know. Well, now they do...

Yay! You made it all the way to the bottom. You deserve a treat. Seeing as you've already got the terminal open, and seeing as we're dealing with apt-cache, let's just get some pretty neat stats with:

```
[code]apt-cache stats[/code]
```

That's it and thanks for reading. I appreciate the audience and am happy that I finally am putting some effort into this project. I've been meaning to do this for years, but something always got in the way. If you want to get notified when new articles are posted, just scroll up and sign up for the newsletter. I promise not to send you any commercial emails and I won't give any of your private data away.

---

## Let's Count the Installed Packages!

Have you ever wanted to know how many packages you've installed? It's actually a really simple process. I'll show you!

This is only useful if your distro uses 'dpkg', which is actually quite a few distros. If you remember correctly, I've

previously told you how to use 'dpkg' to get a list of installed applications. It was a pretty simple command.

Well, so isn't this a simple command.

Like pretty much every time, let's get that terminal opened up. (CTRL + ALT + T)

Now, if you'd followed the above link's directions, you'd have a file called 'installed\_apps.txt' and it would be in your Documents directory. If you open that file, you'll see that not all the lines are actually installed applications. There are some lines that contain data that isn't an installed application.

But, if you look carefully, you'll see that the lines with the applications all begin with 'ii'. We can use that to make our work more accurate.

Now, you could try this:

```
[code]cat Documents/installed_apps.txt | grep ii | wc -l[/code]
```

If you're curious, 'cat' stands for concatenate. It has been around since pretty much Unix v. 1. It basically reads a file and spits out the content. The man page describes it as thus:

*cat – concatenate files and print on the standard output*

See? Pretty simple.

Next, you're telling it which file to work on (installed\_apps.txt in the Documents directory). Then you're telling it to look for the letters 'ii' and to count the lines that contain them.

**Note:** In some cases, this will not be 100% accurate. If you have something installed with 'ii' in the name then it will count that as well. However, the goal here isn't actually 100%

accuracy, the goal here is to help you get familiar with some of these terminal commands.

So, what if you haven't followed along and don't actually have that file? Can you still do this? Absolutely! Watch this:

```
[code]dpkg -l | grep ii | wc -l[/code]
```

To be a bit more clear, that little '|' character is called a pipe. You'll see it fairly often. It's used to take the commands from one command and use them in another. It goes back to the philosophy of 'hiding the internals', with the goal being simplicity and clarity. But, you never have to make the text file to perform this counting exercise.

Again, this command will give you an inaccurate result if you happen to have an application that has an 'ii' in the name. That's fine. This is great for estimation and you really don't need a hard number for anything. In a quick look, I have exactly zero apps with 'ii' in the name. So, in my case the count should be spot on. The goal is to help you get more comfortable in the terminal and get used to some of these commands. They're surprisingly useful, even in day-to-day operations.

Like always, thanks for reading. If you want, you can sign up for the email newsletter. It's over there, in the right sidebar. I promise, I won't send you any spam, I won't sell your email address, I won't give your email address away, and I won't send you pics of my dinner. So, sign up and be the first person on the block to read new articles!

---

# How To: See Installed Applications

Here's a nice and quick way to see what software you have installed – assuming you're using a distro with Aptitude.

Crack open your terminal. You can use CTRL + ALT + T. Enter this:

```
[code]dpkg -l[/code]
```

That doesn't do you a whole lot of good, but it works. So, let's make it more usable.

```
[code]dpkg -l > Documents/installed_apps.txt[/code]
```

Now, open your Documents directory and you'll have a handy list of your installed apps in a file with an intuitive name.