

Installing Open-Source Software on Linux

RPM LOADING

CHRIS BROWN

Everyone needs to update their system with new programs from time to time. Here, we show you how to find and install applications simply and painlessly.



When you install a Linux distribution, the installation software will ask you to specify what pieces you want to include. Usually it provides default selections such as a 'standard workstation' install, and if you're new to Linux, chances are you'll go with the default. You'll be

guaranteed a working, useful system with a reasonable collection of tools. Sooner or later, though, you're going to want to use something that the default installation didn't include. Or you may want a package that isn't part of your Linux distribution at all. Or you might want to install a more recent version of a package than the one you're running now, perhaps because it fixes bugs which are troubling you, or has new features you need, or closes security holes which have been discovered.

In this article we're going to learn how we can obtain and install new software packages on Linux. More often than not, finding the package in the first place is the hardest part. So we'll look at ways of searching for software, too.

Figure 1: Finding RPMs.

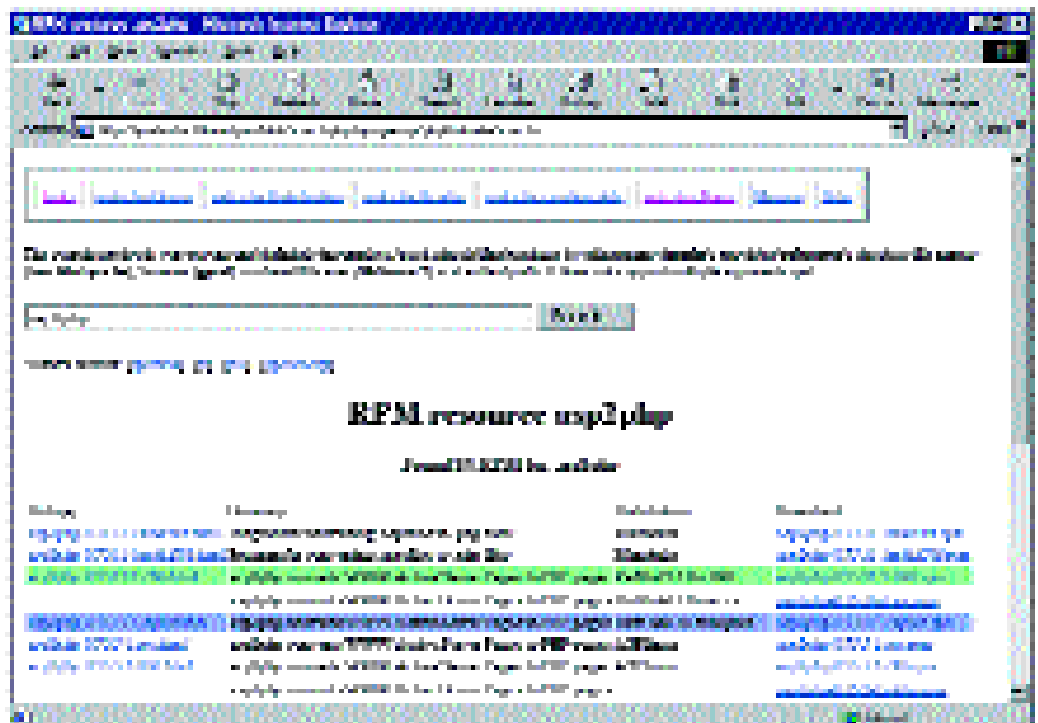




Figure 2: Check the information before you download

The sheer wealth of free software available for Linux can be overwhelming - what one might call an 'embarrassment of riches', to quote Abbe D'Allainval. Well, actually, he said it in French, and since he died in 1753 he most likely wasn't talking about software. But the sense of the phrase is that the more alternatives you have, the more difficult it is to choose.

Distribution formats

Software packaged for Linux is available in a number of formats. The two you'll most often encounter (and the two we're going to look at) are RPM packages, and a format rather quaintly known as a 'tarball'. RPM packages usually provide compiled binary code for a specific platform. Tarballs usually provide source code that can be compiled and run on different hardware architectures and a variety of versions of Linux and UNIX. There are other packages formats, which we don't deal with here, such as the Gnu/Debian DEB format.

What's RPM?

RPM stands for Red Hat Package Manager, but its use is not restricted to Red Hat Linux. The SuSE and Mandrake distributions, for example, also use it. RPM defines a format for parcelling up an installable piece of software. The parcel (a '.rpm' file) includes the compiled binary code of the software itself, scripts which do whatever needs to be done to install the software, and information about other software packages which this particular package requires. One of the biggest advantages of using

RPM is that it keeps a database of packages already installed on your system, and if you want to install a new package it will tell you if any of the pre-requisite packages required by the new package aren't present. Missing pre-requisites are the most likely cause of failure when installing a new package. Although RPM doesn't magically fix the problem, it does at least give you a clear statement of what's missing.

Installing an RPM

Let's take an example of locating and installing an RPM package. Suppose I'm interested in migrating a website, developed using active server pages, to my Linux machine. I have heard that there's a tool called asp2php that claims to translate asp code into php which I can run in my Apache Web server. I'd like to give it a try.

Looking locally

The first thing to figure out is, do I already have it? That's not such a stupid question. There's an awful lot of stuff on those CDs that came with my Linux distribution I don't know about. I'm a great believer in trying obvious things first, so I'll open a terminal window and try to run it:

```
$ asp2php
bash: asp2php: command not found
$
```

Not a good start. That doesn't prove it's not installed, but it does show that it's not in one of the directories on my search path. We can do an exhaustive search of the file system with the find

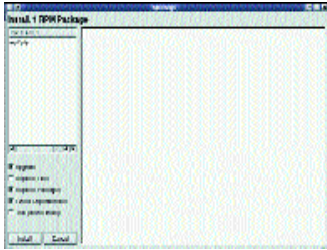


Figure 3: Using kpackage to install

command. It's best to do this as root, otherwise you'll get lots of messages about directories that find doesn't have permission to look in. Try the command:

```
# find / -name asp2php -print
#
```

which says "search the whole file system, starting at the root directory, for a file called asp2php, and if you find it, print its name". This might take a minute or two. In my case, it didn't find anything. I just got a prompt back. This is the usual Linux/UNIX philosophy - as Thumper said in a rare moment of lucidity in the film 'Bambi': "If you can't say nutt'n nice, don't say nutt'n at all".

If your system uses RPM for its package management, you could have found this out a little faster by using the rpm command to query the list of installed packages:

```
$ rpm -q asp2php
package asp2php is not installed
$
```



Figure 4: Showing the dependencies needed

Of course, the fact that it isn't installed doesn't necessarily mean that it isn't included in my Linux distro. We need to take a look at the RPM files on the distribution CDs. In the case of a Red Hat distribution, for example, the RPMs live in a directory called Red Hat/RPMS on each CD. So, with one of the CDs mounted on /mnt/cdrom, for example, I can look for the package with a command such as:

```
$ ls /mnt/cdrom/RedHat/RPMS/asp2php*
```

In this way, I discovered that in the Red Hat 7.1 distribution CD2 contains two files:

```
asp2php-0.75.11-1.i386.rpm
asp2php-gtk-0.75.11-1.i386.rpm
```

These filenames, whilst admittedly messy, tell you quite a bit: asp2php is the name of the package. 0.75.11 is the version number. The '-1' tells you that this is the first build of an RPM for that package and version. The 'i386' means it's compiled for the Intel architecture. (Remember, this RPM contains compiled code so it's specific to the processor architecture for which it was built. If you were using a Sun sparystation, for example, you'd be looking for a name ending in '.sparc.rpm').

Installing the RPM

Now we've found the rpms, we can install both of them with the command:

```
# rpm -i /mnt/cdrom/RedHat/RPMS/asp2php*
#
```

You have to do this as root because the installation needs to copy the program into a system directory (/usr/bin in this case) and update the RPM database.

If asp2php required any packages which weren't present, it would have told you at this stage. For example, if I had tried to install asp2php-gtk-0.75.11-1.i386.rpm by itself, I would see this:

```
# rpm -i /mnt/cdrom/RedHat/RPMS/asp2php-gtk-0.75.11-1.i386.rpm
error: failed dependencies:
  asp2php = 0.75.11 needed by asp2php-gtk-0.75.11-1.i386.rpm
#
```

Typing these really long command lines and filenames (and getting them right!) is a pain, even for die-hard command line users like me. If you're using bash as your shell, it's much easier if you turn on filename completion. To generate the command line just shown, what I actually typed was:

```
rpm -i /mnt/c <T>/Re <T>/R <T>/asp <T>-g<T>
```

where <T> means press the TAB key. At each stage, bash will complete the filename for you.

If you don't like working at the command line, you can do all these things with graphical tools. For example, clicking on the RPM file within the KDE browser will bring up a kpackage install screen from where the package can be installed simply by clicking the Install button. See Figure 1.

Alternatively, kpackage can be started directly, and pointed at the directory where the RPM files are (click on Settings --> Location of uninstalled packages), and used to browse, select, examine, and install any package. See Figure 2.

Net gain

OK, what if we can't find the package we want on our distribution CDs? A great place to start searching for RPMs on the Internet is <http://rpmfind.net/linux/RPM>. Just type the package name into the search box on that page (see Figure 3).

You'll get a listing of all known RPMs for the package, with the most recent version numbers listed first. The third line in this list is the same Red Hat rpm we just installed off our local disk.

Look carefully at the filenames. The fifth line is an entry for the same package compiled for the Sparc architecture. Clicking on the filename link at the start of the line will display basic information about the package. See Figure 4. Clicking on the .rpm file name at the end of the line will download the rpm which can then be installed as previously described. ■

