How To Use Arch Linux Package Management

Authored by: **ASPHostServer Administrator** *[asphostserver@gmail.com]*
Saved From: http://faq.asphosthelpdesk.com/article.php?id=170

Arch Linux provides package management facilities similar to those found in other modern Linux distributions. This is a guide to common package management operations.

Before proceeding further, make sure your package databases are up to date with:

```
sudo pacman -Sy
```

Searching

Favoring brevity over intuitiveness, most package management operations in Arch Linux appear in the format:

```
pacman -<a-z><a-z>.
```

For example, the normal command for searching pacman repositories is

```
pacman -Ss <package>
```

Like most pacman commands, it is not particularly obvious. pacman also ships with a pacsearch utility. It works similarly, but it has some enhancements over pacman -Ss:

- color highlighting
- installed packages category ('local')

Here is how you would list all standard packages with the keyword 'linux.'

```
pacsearch linux
```

Sometimes there will be a lot of packages matching your search criteria. This is especially true when using an AUR helper like yaourt. Regular expressions can help narrow down the list. For example, let's say you are trying to find anything related to the "ack" tool. A regular search for ack will inevitably return a lot of results to sift through, due to words like "package," "hack," or "playback."

```
pacsearch ack
```

However, not many words begin with ack. Adding a ^ to the search word will only include results that start with those letters. This works with at least pacsearch, pacman, and yaourt.

```
pacsearch ^ack
```

To guarantee a regular expression works as intended, surround it with quotes.

```
pacsearch 'c\+\+'
```

If you want to get a list of installed packages matching a search, pipe the results to "grep local." The "-A 1" option tells grep to include one line after each match (the package description, in this case).

```
pacsearch linux | grep local -A 1
```

Alternatively, you can use pacman -Qs with the downside of no color highlighting.

```
pacman -Qs linux
```

If you want color highlighting with pacman -Ss, you can uncomment Color from /etc/pacman.conf. It highlights different things than pacsearch, though, so you might want to try both and see which one you like better.

```
sudo vi /etc/pacman.conf
...
# Misc options
#UseSyslog
Color
#TotalDownload
CheckSpace
#VerbosePkgLists
...
```

Getting Information

pacman -Qi displays basic information about an installed package.

```
pacman -Qi linux
Name            : linux
Version         : 3.8.4-1
Description     : The linux kernel and modules
Architecture    : x86_64
URL             : http://www.kernel.org/
Licenses        : GPL2
Groups          : base
Provides        : kernel26=3.8.4
Depends On      : coreutils  linux-firmware  kmod  mkinitcpio>=0.7
Optional Deps   : crda: to set the correct wireless channels of your country
Required By     : None
Optional For    : None
Conflicts With  : kernel26
Replaces        : kernel26
Installed Size  : 64728.00 KiB
Packager        : Tobias Powalowski <tpowa@archlinux.org>
Build Date      : Wed Mar 20 21:16:17 2013
Install Date    : Fri Mar 29 01:02:14 2013
Install Reason  : Explicitly installed
Install Script  : Yes
Validated By    : Unknown
```

If pacman -Qi is passed no arguments, it returns all packages in the system. You can search this output to get specialized information about installed packages.

For example, if you wanted to get each package and its size:

```
pacman -Qi | grep -e "Name" -e "Installed Size"
Name            : a52dec
Installed Size : 244.00 KiB
Name            : aalib
Installed Size : 768.00 KiB
...
```

While pacman -Qi provides information about installed packages, pacman -Si queries the database for the most recently retrieved information about a package.

```
pacman -Si linux
Repository      : core
Name            : linux
Version         : 3.11.6-1
Description     : The Linux kernel and modules
Architecture    : x86_64
URL             : http://www.kernel.org/
Licenses        : GPL2
Groups          : base
Provides        : kernel26=3.11.6
Depends On      : coreutils  linux-firmware  kmod  mkinitcpio>=0.7
Optional Deps   : crda: to set the correct wireless channels of your country
Conflicts With  : kernel26
Replaces        : kernel26
Download Size   : 47855.98 KiB
Installed Size  : 64493.00 KiB
Packager        : Thomas Bächler <thomas@archlinux.org>
Build Date      : Fri 18 Oct 2013 05:25:12 PM EDT
Validated By    : MD5 Sum  SHA256 Sum  Signature
```

pacman -Ql lists all files associated with a package.

```
pacman -Ql vi
vi /usr/
vi /usr/bin/
vi /usr/bin/edit
vi /usr/bin/ex
vi /usr/bin/vedit
vi /usr/bin/vi
...
```

The package name on each line can make the output more difficult to use in a script. pacman -Qlq (i.e. "pacman query list, quiet") will not print the package name.

```
pacman -Qlq vi
/usr/
/usr/bin/
/usr/bin/edit
```

```
/usr/bin/ex
/usr/bin/vedit
/usr/bin/vi
...
```

You can use pacman -Qlq | grep bin to find all files in that package that are in a bin folder (and thus are likely executable files). This is handy when the command associated with a package is different from the package name.

```
pacman -Qql pacman | grep bin
/usr/bin/
/usr/bin/bacman
/usr/bin/checkupdates
/usr/bin/cleanupdelta
/usr/bin/makepkg
/usr/bin/paccache
/usr/bin/pacdiff
/usr/bin/paclist
/usr/bin/paclog-pkglist
/usr/bin/pacman
/usr/bin/pacman-db-upgrade
/usr/bin/pacman-key
/usr/bin/pacman-optimize
/usr/bin/pacscripts
/usr/bin/pacsearch
/usr/bin/pacsort
/usr/bin/pacsysclean
/usr/bin/pactree
/usr/bin/pkgdelta
/usr/bin/rankmirrors
/usr/bin/repo-add
/usr/bin/repo-elephant
/usr/bin/repo-remove
/usr/bin/testdb
/usr/bin/testpkg
/usr/bin/updpkgsums
/usr/bin/vercmp
```

Storage Consumption

The simplest way to see which packages are taking up the most space on your system is pacsysclean which ships with pacman.

```
pacsysclean
```

If you want something a little more complicated, the pacgraph utility can produce a dependency word cloud of the largest packages in SVG and PNG format.

pacgraph is found in the community repository.

```
sudo pacman -S pacgraph
```

pacgraph -c will output to the console like pacsysclean. If there are a lot of packages, pacgraph's output can get unwieldy since it is sorted from largest to smallest, unlike pacsysclean. You can pipe it to head to just see biggest packages.

```
pacgraph -c | head
Autodetected Arch.
Loading package info
Total size: 730MB
114MB linux
103MB libtool
84MB pacgraph
40MB imagemagick
25784kB ppl
22264kB nmap
17295kB pkgfile
```

In the Arch Linux server instance I tested, some errors were produced before this output. To remove the errors, you will need to correct the locale settings from the default "C".

```
export LANG=en_US.UTF-8
```

Run without any arguments, pacgraph produces an SVG and, if possible, a PNG file. If you don't have a file sharing system set up with your virtual server, the simplest way to access the images generated by pacgraph is to use Secure Copy.

```
scp <user>@<addr>:<remote_path> <local_path>
```

Note: Although pacgraph will claim to render a PNG file after imagemagick and libpng are installed, at the time of this writing it will not actually generate a PNG without inkscape, which has quite a few dependencies.

Downloads / Installation

The typical way to install or upgrade a standard package is pacman -S.

```
sudo pacman -S <package>
```

Packages often have a similar pattern in their names. Brackets can be used as a shortcut.

```
sudo pacman -S lua-{sec,socket,zlib}
```

Sometimes you might want to just download a package for archival purposes without installing it. pacman -Sw will download a package to the cache folder.

```
sudo pacman -Sw <package>
```

If a package has been downloaded, or if you know the url, you can install it directly.

```
sudo pacman -U <package_path>
```

Upgrades

Since pacman revolves around the idea of "syncing" a package with the remote repository, pacman -S will upgrade a package if it is already installed.

Pacman can list packages that are out of date.

```
pacman -Qu
```

Unfortunately, it doesn't display the version of the package in the repository, so you won't be able to tell how out of date each package is. If this is important to you, you might want to look into 3rd party package managers or write a script that ties together the package version obtained from pacman -Qi and pacman -Si.

You can get the version from pacman -Qi or pacman -Si using a regular expression.

```
pacman -Qi linux | grep "Version" | sed 's/^Version\s*:\s//'
3.8.4-1
```

And then write a script that shows the versions side by side.

```
for i in $(pacman -Qqu)
do
  printf "$i: "
  printf "$(pacman -Qi "$i" | grep 'Version' | sed 's/^Version\s*:\s//') "
  echo   "$(pacman -Si "$i" | grep 'Version' | sed 's/^Version\s*:\s//')"
done
```

Pacman provides a way to upgrade all of your packages at once, but it is not recommended because Arch is a rolling release distribution. If problems arise, it can take time to determine what the causes are.

```
sudo pacman -Syu
```

Downgrades

Arch Linux does not officially maintain deprecated packages. Instead, you will need to rely on your package cache and places like the Arch Rollback Machine.

Cache

Every package downloaded with pacman is stored in /var/cache/pacman/pkg.

```
ls /var/cache/pacman/pkg | grep linux
linux-3.11.6-1-x86_64.pkg.tar.xz
linux-3.9.7-1-x86_64.pkg.tar.xz
...
```

If you would like to revert to a package in your cache, just install it directly.

```
pacman -U <path_to_cached_file>
```

Arch Rollback Machine

The Arch Rollback Machine is a collection of snapshots of the official Arch Linux mirror. As of this writing, it goes back four months. The ARM is currently hosted at http://seblu.net/a/arm, though this could change in the future.

ARM packages can be installed remotely using pacman -U .

```
pacman -U
```

```
http://seblu.net/a/arm/2013/08/31/extra/os/x86_64/apache-2.2.25-1-x86_64.pkg.tar.xz
```

It isn't exactly convenient to browse the ARM for older packages. Fortunately, there are tools that make this easier:

- downgrade
- downgrader

They search for older versions in the cache and the ARM. Their usage is what you would expect.

```
downgrade <package>
downgrader <package>
```

Both tools are in the AUR, so the easiest way to install them will be with a helper like yaourt.

Note: The ARM is an unofficial project and has been closed in the past, so it might be a good idea to avoid clearing your cache in case the ARM goes down or changes locations again. If you would like to roll your own ARM, there appears to be a NodeJS project on github.

Removal

Remove a package, provided nothing is depending on it. Leave its dependencies installed.

```
sudo pacman -R <package>
```

Remove a package, provided nothing is depending on it. Remove its dependencies that aren't required by anything else.

```
sudo pacman -Rs <package>
```

Force removal of a package. This is the command you will reach for to just wipe the package from your system and reinstall when reinstalling alone isn't enough.

```
sudo pacman -Rdd <package>
```

Unofficial Packages

The vast majority of Arch Linux packages are located in the [Arch User Repository](). The best way to search for and download packages from the AUR is to use a [helper tool]().