# Pacman Package Manager: Core Functionality

Pacman is the simple but surprisingly effective (IMHO) command-line package manager of the Arch Linux project. At it’s core, it has the ability to sync with an external repository and install packages from it, upgrade updated packages, search for packages, uninstall packages, more. Pacman does automatic package signing and dependency management. It does core tasks extremely quickly, and this fact is a key part of its user experience.

Pacman has a number of commands, each with their own set of subcommands.

## Sync

For your standard daily use, the “-S” (Sync) command will often be sufficient. -S communicates with an external software repository to search for packages (-Ss package-name), download packages (-S package-name), update the local database (-Sy), upgrade the locally installed packages (-Su), and get details about individual packages (-Si). There are other options to alter the operation, but we can get to those later.

## Query

the “-Q” (Query) command is used to inspect the *local* database. The most command operations used are getting the info about an installed package (-Qi), listing the files owned by a specific package (-Ql) often useful for finding the binary files included with a package, searching the local database (-Qs), and listing outdated packages (-Qu).

This command is noticibly similar to Sync, with the main difference being working with the local database rather than an external one.

## Files

Get information about which package owns specific files with the “-F” (“Files”) command, especially useful for when you need to install a program and don’t know which package it is in. It also allows you to list the files owned by a package (-Fl). The “File Database” is seperate from the core package database, and can be updated seperately (-Fy).

## Removing

Remove packages from the local system with the -R (“Remove”) command. Useful subcommands include also removing associated configuration files (-Rn) and also recursively removing dependencies that aren’t needed by anything else (-Rs). A very common command combines these two subcommands (-Rns).

## Installing From Other Sources

Packages can be installed from other sources (URL or file) with the -U command. This command will also install dependencies from the standard external repositories

## Searching For Packages

Pacman has a very simple searching mechanism, searching by a string with one of the “s” subcommands. Only packages whose name contains that string will be returned. Pacman Apparently has the ability to search via regexp, but it is not explained anywhere the format of the regular expressions (I’m guessing “standard”). This is an area of pacman that could *very much* use some improvement with more complex searches, fuzzy searching, etc.

## Makepkg

In addition to the standard binary package repositories, pacman also has the community-driven “Arch User Repository” (AUR) where users can download, compile, and install from source code. This is done with the “makepkg” command and then installed with pacman. While not strictly part of pacman, makepkg and the AUR is closely linked to it and often used in the management of an Arch Linux system. If we end up lacking screens in the UI, it might be a good idea to include this functionality.

# Examples With Output

## Installation

sudo pacman -S atom  
resolving dependencies...  
looking for conflicting packages...  
  
Packages (9) apm-2.4.5-1 c-ares-1.15.0-1 electron4-4.2.12-2 node-gyp-6.1.0-1  
 nodejs-13.9.0-1 npm-6.14.1-1 ripgrep-11.0.2-1 semver-7.1.3-1  
 atom-1.44.0-1  
  
Total Installed Size: 410.48 MiB  
  
:: Proceed with installation? [Y/n]   
(9/9) checking keys in keyring [######################] 100%  
(9/9) checking package integrity [######################] 100%  
(9/9) loading package files [######################] 100%  
(9/9) checking for file conflicts [######################] 100%  
(9/9) checking available disk space [######################] 100%  
:: Processing package changes...  
(1/9) installing c-ares [######################] 100%  
(2/9) installing nodejs [######################] 100%  
Optional dependencies for nodejs  
 npm: nodejs package manager [pending]  
(3/9) installing semver [######################] 100%  
(4/9) installing node-gyp [######################] 100%  
(5/9) installing npm [######################] 100%  
(6/9) installing apm [######################] 100%  
(7/9) installing electron4 [######################] 100%  
Optional dependencies for electron4  
 kde-cli-tools: file deletion support (kioclient5)  
 trash-cli: file deletion support (trash-put)  
 xdg-utils: open URLs with desktop's default (xdg-email, xdg-open)  
 [installed]  
(8/9) installing ripgrep [######################] 100%  
(9/9) installing atom [######################] 100%  
Optional dependencies for atom  
 ctags: symbol indexing support  
 git: Git and GitHub integration [installed]  
:: Running post-transaction hooks...  
(1/3) Arming ConditionNeedsUpdate...  
(2/3) Updating icon theme caches...  
(3/3) Updating the desktop file MIME type cache...

## Removal

thelink% sudo pacman -Rns atom   
checking dependencies...  
  
Packages (9) apm-2.4.5-1 c-ares-1.15.0-1 electron4-4.2.12-2 node-gyp-6.1.0-1  
 nodejs-13.9.0-1 npm-6.14.1-1 ripgrep-11.0.2-1 semver-7.1.3-1  
 atom-1.44.0-1  
  
Total Removed Size: 410.48 MiB  
  
:: Do you want to remove these packages? [Y/n]   
:: Processing package changes...  
(1/9) removing atom [############################] 100%  
(2/9) removing ripgrep [############################] 100%  
(3/9) removing electron4 [############################] 100%  
(4/9) removing apm [############################] 100%  
(5/9) removing npm [############################] 100%  
(6/9) removing node-gyp [############################] 100%  
(7/9) removing semver [############################] 100%  
(8/9) removing nodejs [############################] 100%  
(9/9) removing c-ares [############################] 100%  
:: Running post-transaction hooks...  
(1/3) Arming ConditionNeedsUpdate...  
(2/3) Updating icon theme caches...  
(3/3) Updating the desktop file MIME type cache...

## Searching for a Package

thelink% pacman -Ss atom  
extra/atomix 3.34.0-1  
 Build molecules out of single atoms  
extra/boost-libs 1.72.0-1 [installed]  
 Free peer-reviewed portable C++ source libraries - runtime libraries  
extra/katomic 19.12.2-1 (kde-applications kdegames)  
 A fun and educational game built around molecular geometry  
extra/libatomic\_ops 7.6.10-1  
 Provides semi-portable access to hardware provided atomic memory operations  
extra/libgrss 0.7.0+16+g971c421-2  
 Glib-based library to manage RSS and Atom feeds  
extra/python-feedparser 5.2.1-5  
 Parse RSS and Atom feeds in Python  
extra/python2-feedparser 5.2.1-5 [installed]  
 Parse RSS and Atom feeds in Python2  
extra/syndication 5.67.0-1 (kf5)  
 RSS/Atom parser library  
extra/xorg-xlsatoms 1.1.3-1 (xorg-apps xorg) [installed]  
 List interned atoms defined on server  
community/apm 2.4.5-1  
 Atom package manager  
community/atom 1.44.0-1  
 A hackable text editor for the 21st Century

## Seaching for a File

thelink% pacman -F atom   
community/atom 1.41.0-1  
 usr/bin/atom  
 usr/lib/atom/atom  
community/lv2 1.16.0-3  
 usr/include/lv2/atom  
 usr/include/lv2/lv2plug.in/ns/ext/atom  
community/wpscan 1:3.7.5-1  
 opt/wpscan/spec/fixtures/finders/wp\_version/atom\_generator/feed/atom

thelink% pacman -F atom   
community/atom 1.41.0-1  
 usr/bin/atom  
 usr/lib/atom/atom  
community/lv2 1.16.0-3  
 usr/include/lv2/atom  
 usr/include/lv2/lv2plug.in/ns/ext/atom  
community/wpscan 1:3.7.5-1  
 opt/wpscan/spec/fixtures/finders/wp\_version/atom\_generator/feed/atom  
thelink% pacman -F wifi-menu  
core/netctl 1.20-2 [installed]  
 usr/bin/wifi-menu  
 usr/share/bash-completion/completions/wifi-menu

## Dependencies and Conflicts

Pacman performs automatic dependency resolution. When a user installs a package, the system will also add dependencies to the list of installed packages (recursively); These dependencies will be recorded in the manifest as “implicitly installed”, meaning that they were installed to be a dependency of another package and will be removed (if requested) when that package is removed. Pacman requires all dependencies of a package to be installed before that package is installed, otherwise it will not go ahead with the operation; this prevents the system from entering an invalid (unsafe) state.

Arch Linux is a “rolling release” distribution, meaning that the system as a whole is not versioned, and devices running the system are intended to update the whole set of packaged at once rather than piecemeal, always keeping each package updated to its latest version. As such, the versioning conflicts encountered in other distributions are rare or absent from Arch systems. Nevertheless, pacman does understand version dependencies (say that package A requires package B with version >= 1.52) and enforces them.

A more common conflict is when two packages provide the same program (like MySQL and MariaDB, which is an open-source implementation of MySQL). In this situation, the user will be asked whether they want the new package or the old package, since both cannot exist on the system simultaneously. This occasionally happens in the course of upgrading the system, as the package tree is rearranged to better organize its contents.