pacman/Rosetta

wiki.archlinux.org/title/Pacman/Rosetta

Tip: Arch users having to temporarily deal with another Linux distribution can use pacapt, a simple wrapper around other package managers.
Note: Some of the tools described here are specific to a certain version of *pacman*. The `-Qk` option is new in *pacman* 4.1.

Basic operations

Action	Arch	Red Hat/Fedora	Debian/Ubuntu	SLES/openSUSE	Gentoo
Install a package(s) by name	`pacman -S`	`dnf install`	`apt install`	`zypper install` or `zypper in`	`emerge`
Remove a package(s) by name	`pacman -Rs`	`dnf remove`	`apt remove`	`zypper remove` or `zypper rm`	`emerge`
Search for package(s) by searching the expression in name, description, short description. What exact fields are being searched by default varies in each tool. Mostly options bring tools on par.	`pacman -Ss`	`dnf search`	`apt search`	`zypper search` or `zypper se [-s]`	`emerge`
Upgrade Packages - Install packages which have an older version already installed	`pacman -Syu`	`dnf upgrade`	`apt update` and then `apt upgrade`	`zypper update` or `zypper up`	`emerge @world`
Upgrade Packages - Another form of the update command, which can perform more complex updates -- like distribution upgrades. When the usual update command will omit package updates, which include changes in dependencies, this command can perform those updates.	`pacman -Syu`	`dnf distro-sync`	`apt update` and then `apt dist-upgrade`	`zypper dup`	`emerge @world`
Clean up all local caches. Options might limit what is actually cleaned.	`pacman -Sc` or `pacman -Scc`	`dnf clean all`	`apt autoclean` removes only unneeded, obsolete information or `apt clean`	`zypper clean`	`eclean`

Remove dependencies that are no longer needed, because e.g. the package which needed the dependencies was removed.	`pacman -Qdtq | pacman -Rs -`	`dnf autoremove`	`apt autoremove`	`zypper rm -u` (just for removing a package) or `zypper packages --unneeded` (listing only and without recursion)	`emerge depclean`
Remove packages no longer included in any repositories.	`pacman -Qmq | pacman -Rs -`	`dnf repoquery --extras`	`aptitude purge '~o'`		
Mark a package previously installed as a dependency as explicitly required.	`pacman -D --asexplicit`	`dnf mark install`	`apt-mark manual`	`zypper install --force` (workaround which needs to reinstall the package)	`emerge`
Install package(s) as dependency / without marking as explicitly required.	`pacman -S --asdeps`	`dnf install` and then `dnf mark remove`	`apt-mark auto`	n/a (feature request)	`emerge oneshot [-a] -1`
Only downloads the given package(s) without unpacking or installing them	`pacman -Sw`	`dnf download`	`apt install --download-only` (into the package cache) or `apt download` (bypass the package cache)	`zypper --download-only`	`emerge fetchonl`
Start a shell to enter multiple commands in one session		`dnf shell`	`apt-config shell`	`zypper shell`	
Show a log of actions taken by the software management.	read `/var/log/pacman.log`	`dnf history`	read `/var/log/dpkg.log`	read `/var/log/zypp/history`	read `/var/lo`
Get a dump of the whole system information - Prints, Saves or similar the current state of the package management system. Preferred output is text or XML. (Note: Why either-or here? No tool offers the option to choose the output format.)	see `/var/lib/pacman/local`	see `/var/lib/rpm/Packages`	`apt-cache stats`		`emerge`
e-mail delivery of package changes			`apt install apt-listchanges`		

Querying specific packages

Action	Arch	Red Hat/Fedora	Debian/Ubuntu	SLES/openSUSE	Gentoo

Show all or most information about a package. The tools' verbosity for the default command vary. But with options, the tools are on par with each other.	`pacman -Si` or `pacman -Qi`	`dnf list` or `dnf info`	`apt show` or `apt-cache policy`	`zypper info` or `zypper if`	`emerge -S` , `emerge -pv` or `eix`
Display local package information: Name, version, description, etc.	`pacman -Qi`	`rpm -qi` / `dnf info installed`	`dpkg -s` or `aptitude show`	`zypper info` or `rpm -qi`	`emerge -pv` or `emerge -S`
Display remote package information: Name, version, description, etc.	`pacman -Si`	`dnf info`	`apt-cache show` or `aptitude show`	`zypper info`	`emerge -pv` and `emerge -S` or `equery meta`
Display files provided by local package	`pacman -Ql`	`rpm -ql`	`dpkg -L`	`rpm -ql`	`equery files` or `qlist`
Display files provided by a remote package	`pacman -Fl`	`dnf repoquery -l` or `repoquery -l` (from package yum-utils)	`apt-file list`		`pfl`
Query the package which provides FILE	`pacman -Qo`	`rpm -qf` (installed only) or `dnf provides` (everything) or `repoquery -f` (from package yum-utils)	`dpkg -S` or `dlocate`	`zypper search -f`	`equery belongs` or `qfile`
List the files that the package holds. Again, this functionality can be mimicked by other more complex commands.	`pacman -Ql` or `pacman -Fl`	`dnf repoquery -l`	`dpkg-query -L`	`rpm -ql`	`equery files` or `qlist`
Displays packages which provide the given exp. aka reverse provides. Mainly a shortcut to search a specific field. Other tools might offer this functionality through the search command.	`pacman -F`	`dnf provides`	`apt-file search`	`zypper what-provides` or `zypper wp`	`equery belongs` (only installed packages) or `pfl`
Search all packages to find the one which holds the specified file.	`pacman -F`	`dnf provides`	`apt-file search` or `auto-apt` is using this functionality.	`zypper search -f`	`equery belongs` or `qfile`
Show the changelog of a package	`pacman -Qc`	`dnf changelog`	`apt-get changelog`	`rpm -q --changelog`	`equery changes -f`

Querying package lists

Action	Arch	Red Hat/Fedora	Debian/Ubuntu	SLES/openSUSE	Gentoo
Search for package(s) by searching the expression in name, description, short description. What exact fields are being searched by default varies in each tool. Mostly options bring tools on par.	`pacman -Ss`	`dnf search`	`apt search`	`zypper search` or `zypper se -s`	`emerge -S` or `eix`
Lists packages which have an update available. Note: Some provide special commands to limit the output to certain installation sources, others use options.	`pacman -Qu`	`dnf list updates` or `dnf check-update`	`apt list --upgradable`	`zypper list-updates` or `zypper patch-check` (just for patches)	`emerge -uDNp @world`
Display a list of all packages in all installation sources that are handled by the packages management. Some tools provide options or additional commands to limit the output to a specific installation source.	`pacman -Sl`	`dnf list available`	`apt-cache dumpavail` or `apt-cache dump` (Cache only) or `apt-cache pkgnames`	`zypper packages`	`portageq all_best_visible /`
Generates a list of installed packages	`pacman -Q`	`dnf list installed`	`dpkg --list | grep ^i`	`zypper search --installed-only`	`qlist -IC`
List packages that are installed but are not available in any installation source (anymore).	`pacman -Qm`	`dnf list extras`	`apt --installed list | grep ,local`	`zypper se -si | grep 'System Packages'`	`eix-test-obsolete`

List packages that were recently added to one of the installation sources, i.e. which are new to it.		`dnf list recent`	`aptitude search '~N'` or `aptitude forget-new`		`eix-diff`
List installed local packages along with version	`pacman -Q`	`rpm -qa`	`dpkg -l` or `apt list --installed`	`zypper search -s` or `rpm -qa`	`qlist -ICv`
Search locally installed package for names or descriptions	`pacman -Qs`	`rpm -qa '*<str>*'`	`aptitude search '~i(~n $name|~d $description)'`		`eix -S -I`
List packages not required by any other package	`pacman -Qtt`	`dnf leaves` or `package-cleanup --leaves --all`	`deborphan -anp1`		`emerge -pc`
List packages installed explicitly (not as dependencies)	`pacman -Qe`	`dnf history userinstalled`	`apt-mark showmanual`	grep -E '^i\+' (workaround)	`emerge -pvO @selected` or `eix --selected`
List packages installed automatically (as dependencies)	`pacman -Qd`	grep -E '^i[^+]' (workaround)	`apt-mark showauto`		

Querying package dependencies

Action	Arch	Red Hat/Fedora	Debian/Ubuntu	SLES/openSUSE	Gentoo
Display packages which require X to be installed, aka show reverse dependencies.	`pacman -Sii`	`dnf repoquery --alldeps --whatrequires` or `repoquery --whatrequires`	`apt-cache rdepends` or `aptitude search ~D$pattern`	`zypper search --requires`	`emerge -pvc`
Display packages which conflict with given expression (often package). Search can be used as well to mimic this function.		`dnf repoquery --conflicts`	`aptitude search '~C$pattern'`		
List all packages which are required for the given package, aka show dependencies.	`pacman -Si` or `pacman -Qi`	`dnf repoquery --requires` or `repoquery -R`	`apt-cache depends` or `apt-cache show`	`zypper info --requires`	`emerge -ep`
List what the current package provides		`dnf repoquery --provides`	`dpkg -s` or `aptitude show`	`zypper info --provides`	`equery files` or `qlist`
List all packages that require a particular package		`dnf repoquery --installed --alldeps --whatrequires`	`aptitude search ~D{depends,recommends,suggests}:$pattern` or `aptitude why`	`zypper search --requires`	`equery depends -a`
Display all packages that the specified packages obsoletes.		`dnf list obsoletes`	`apt-cache show`		
Generates an output suitable for processing with dotty for the given package(s).			`apt-cache dotty`		

Installation sources management

Action	Arch	Red Hat/Fedora	Debian/Ubuntu	SLES/openSUSE
Installation sources management	edit `/etc/pacman.conf`	edit `/etc/yum.repos.d/${REPO}.repo`	edit `/etc/apt/sources.list`	edit `/etc/zypp/repos.d/`

Add an installation source to the system. Some tools provide additional commands for certain sources, others allow all types of source URI for the add command. Again others, like apt and dnf force editing a sources list. apt-cdrom is a special command, which offers special options design for CDs/DVDs as source.	edit `/etc/pacman.conf`	`/etc/yum.repos.d/*.repo`	`apt-cdrom add`	`zypper service-add`
Refresh the information about the specified installation source(s) or all installation sources.	`pacman -Sy` (always upgrade the whole system afterwards)	`dnf clean expire-cache` and then `dnf check-update`	`apt-get update`	`zypper refresh` or
Prints a list of all installation sources including important information like URI, alias etc.	`cat /etc/pacman.d/mirrorlist`	`cat /etc/yum.repos.d/*`	`apt-cache policy`	`zypper service-lis`
List all packages from a certain repo	`paclist <repo>`			
Disable an installation source for an operation		`dnf --disablerepo=`		
Download packages from a different version of the distribution than the one installed.		`dnf --releasever=`	`apt-get install -t release package` or `apt-get install package/release` (dependencies not covered)	

Overrides

Action	Arch	Red Hat/Fedora	Debian/Ubuntu	SLES/openSUSE	Gentoo

Add a package lock rule to keep its current state from being changed	edit `/etc/pacman.conf` modifying IgnorePkg array	edit `dnf.conf` adding/amending the `exclude` option	`apt-mark hold pkg`	`zypper al` or put package name in `/etc/zypp/locks`	`/etc/portage/package.mask`
Delete a package lock rule	edit `/etc/pacman.conf` removing package from IgnorePkg line		`apt-mark unhold pkg`	`zypper rl` or remove package name from `/etc/zypp/locks`	`/etc/portage/package.mask` `package.unmask`)
Show a listing of all lock rules	`cat /etc/pacman.conf`		`/etc/apt/preferences`	`zypper ll` or view `/etc/zypp/locks`	`cat /etc/portage/package.`
Set the priority of the given package to avoid upgrade, force downgrade or to overwrite any default behavior. Can also be used to prefer a package version from a certain installation source.	edit `/etc/pacman.conf` modifying HoldPkg and/or IgnorePkg arrays		`/etc/apt/preferences` , `apt-cache policy`	`zypper mr -p`	edit `/etc/portage/package.acce` adding a line with `=category/` `version`
Remove a previously set priority			`/etc/apt/preferences`	`zypper mr -p`	edit `/etc/portage/package.acce` removing offending line
Show a list of set priorities			`apt-cache policy` or `/etc/apt/preferences`	`zypper lr -p`	`grep -r . /etc/portage/package.accep`
Ignore problems that priorities may trigger.				n/a	

Verification and repair

Action	Arch	Red Hat/Fedora	Debian/Ubuntu	SLES/openSUSE	Gentoo
Verify single package	`pacman -Qk` (can add another `k`)	`rpm -V`	`debsums`	`rpm -V`	`equery check`
Verify all packages	`pacman -Qk` (can add another `k`)	`rpm -Va`	`debsums`	`rpm -Va`	`equery check`
Reinstall given package; this will reinstall the given package without dependency hassle	`pacman -S`	`dnf reinstall`	`apt install --reinstall`	`zypper install --force`	`emerge -1O`
Verify dependencies of the complete system; used if installation process was forcefully killed	`pacman -Dk`	`dnf repoquery --requires`	`apt-get check`	`zypper verify`	`emerge -uDN @world`
Use some magic to fix broken dependencies in a system	for *pacman* dependency level, use `pacman -Dk` ; for shared library level, use findbrokenpkgs[AUR] or `lddd` (from devtools)	`dnf repoquery --unsatisfied`	`apt-get --fix-broken` and then `aptitude install`	`zypper verify`	`revdep-rebuild`

Add a checkpoint to the package system for later rollback		(unnecessary, it is done on every transaction)	n/a
Remove a checkpoint from the system	n/a	n/a	n/a
Provide a list of all system checkpoints	n/a	`dnf history list`	n/a
Rolls entire packages back to a certain date or checkpoint	n/a	`dnf history rollback`	n/a
Undo a single specified transaction	n/a	`dnf history undo`	n/a

Using package files and building packages

Action	Arch	Red Hat/Fedora	Debian/Ubuntu	SLES/openSUSE	Gentoo
Query a package supplied on the command line rather than an entry in the package management database	`pacman -Qp`	`rpm -qp`	`dpkg -I`		
List the contents of a package file	`pacman -Qpl`	`rpmls rpm -qpl`	`dpkg -c`	`rpm -qpl`	
Install local package file, e.g. app.rpm and uses the installation sources to resolve dependencies	`pacman -U`	`dnf install`	`apt install`	`zypper in`	`emerge`
Updates package(s) with local packages and uses the installation sources to resolve dependencies	`pacman -U`	`dnf upgrade`	`debi`		`emerge`
Add a local package to the local package cache mostly for debugging purposes.	`cp package-filename /var/cache/pacman/pkg/`		`apt-cache add package-filename`	n/a	`cp package-filename /usr/portage/distfiles`
Extract a package	`tar -Jxvf`	`rpm2cpio | cpio -vid`	`dpkg-deb -x`	`rpm2cpio | cpio -vid`	`tar -jxvf`
Install/Remove packages to satisfy build-dependencies. Uses information in the source package	Use ABS and `makepkg -seoc`	`dnf builddep`	`apt-get build-dep`	`zypper si -d`	`emerge -o`
Display the source package to the given package name(s)		`dnf repoquery -s`	`apt-cache showsrc`	n/a	
Download the corresponding source package(s) to the given package name(s)	Use ABS and `makepkg -o`	`dnf download --source`	`apt-get source` or `debcheckout`	`zypper source-install`	`emerge --fetchonly`
Build a package	`makepkg -s`	`rpmbuild -ba` (normal) or *mock* (in chroot)	`debuild`	`rpmbuild -ba`, then build, and then `osc build`	`ebuild` or `quickpkg`
Check for possible packaging issues	*namcap* (requires namcap)	*rpmlint*	*lintian*	*rpmlint*	*repoman*

Retrieved from "https://wiki.archlinux.org/index.php?title=Pacman/Rosetta&oldid=667792"

Category:

Package manager