

**NAME**

**pkg**, **pkg-static** - manipulate packages

**SYNOPSIS**

**pkg** [-v] [-d] [-l] [-N] [-j <jail name or id> | -c <chroot path> | -r <root directory>]  
 [-C <configuration file>] [-R <repository configuration directory>] [-4 | -6] <command> <flags>

**pkg** [--version] [--debug] [--list] [-N]  
 [--jail <jail name or id> | --chroot <chroot path> | --rootdir <root directory>]  
 [--config <configuration file>] [--repo-conf-dir <repository configuration directory>] [-4 | -6]  
 <command> <flags>

**DESCRIPTION**

**pkg** provides an interface for manipulating packages: registering, adding, removing and upgrading packages. **pkg-static** is a statically linked variant of **pkg** typically only used for the initial installation of **pkg**. There are some differences in functionality. See pkg.conf(5) for details.

**OPTIONS**

The following options are supported by **pkg**:

**-v, --version**

Display the current version of **pkg**.

**-d, --debug**

Show debug information.

**-l, --list**

List all the available command names, and exit without performing any other action. The **-v** option takes precedence over **-l** but **-l** will override any other command line arguments.

**-o <option=value>, --option <option=value>**

Set configuration option for **pkg** from the command line. Options that are set from the environment are redefined. It is permitted to specify this option multiple times.

**-N**

Activation status check mode. Prevent **pkg** from automatically creating or initializing the SQLite database in `/var/db/pkg/local.sqlite` if it does not already exist.

Prevent **pkg** from performing any actions if no packages are currently installed, on the basis that a correctly initialised system using **pkg** will always have at least the **pkg** package itself registered.

If used without any other arguments, **pkg -N** will run the sanity tests and if successful print out a short message showing how many packages are currently installed. The exit status should be a reliable indication of whether a system is configured to use **pkg** as its package management system or not.

Example usage:

```
if pkg -N >/dev/null 2>&1; then
    # pkgng-specifics
else
    # pkg_install-specifics
fi
```

The **-N** flag was first released in the `/usr/sbin/pkg` bootstrapper in FreeBSD 8.4, but was missing from FreeBSD 9.1. It may not be enough to just call **pkg -N**, as the bootstrapper may be invoked, or an error returned from **pkg**. The following script is the safest way to detect if **pkg** is installed and activated:

```
if TMPDIR=/dev/null ASSUME_ALWAYS_YES=yes \
    PACKAGESITE=file:///nonexistent \
    pkg info -x 'pkg(-devel)?$' >/dev/null 2>&1; then
    # pkgng-specifics
else
    # pkg_install-specifics
fi
```

**-j** <jail name or id>, **--jail** <jail name or id>

**pkg** will execute in the given <jail name or id>, where *name* matches "**jls name**" and *id* matches "**jls jid**". See `jail(8)` and `jls(8)`.

**-c** <chroot path>, **--chroot** <chroot path>

**pkg** will chroot in the <chroot path> environment.

**-r** <root directory>, **--rootdir** <root directory>

**pkg** will install all packages within the specified <root directory>.

**-C** <configuration file>, **--config** <configuration file>

**pkg** will use the specified file as a configuration file.

**-R** <repo conf dir>, **--repo-conf-dir** <repo conf dir>

**pkg** will search the directory for per-repository configuration files. This overrides any value of `REPOS_DIR` specified in the main configuration file.

**-4**     **pkg** will use IPv4 for fetching repository and packages.

**-6**     **pkg** will use IPv6 for fetching repository and packages.

## COMMANDS

The following commands (or their unambiguous abbreviations) are supported by **pkg**:

### **help** *command*

Display usage information of the specified command.

**add**    Install a package from either a local source or a remote one.

When installing from remote source you need to specify the protocol to use when fetching the package.

Currently supported protocols are FTP, HTTP and HTTPS.

### **annotate**

Add, modify or delete tag-value style annotations on packages.

**alias**   List the command line aliases.

**audit**   Audit installed packages against known vulnerabilities.

### **autoremove**

Delete packages which were automatically installed as dependencies and are not required any more.

### **bootstrap**

This is for compatibility with the `pkg(7)` bootstrapper. If **pkg** is already installed, nothing is done.

If invoked with the **-f** flag an attempt will be made to reinstall **pkg** from remote repository.

**check**   Sanity checks installed packages.

**clean**   Clean the local cache of fetched remote packages.

**create** Create a package.

**delete** Delete a package from the database and the system.

**fetch** Fetch packages from a remote repository.

**info** Display information about installed packages and package files.

**install** Install a package from a remote package repository. If a package is found in more than one remote repository, then installation happens from the first one. Downloading a package is tried from each package repository in turn, until the package is successfully fetched.

**lock** Prevent modification or deletion of a package.

**plugins**

List the available plugins.

**query** Query information about installed packages and package files.

**register**

Register a package in the database.

**repo** Create a local package repository for remote usage.

**rquery**

Query information for remote repositories.

**search**

Search for the given pattern in the remote package repositories.

**set** Modify information in the installed database.

**shell** Open a SQLite shell to the local or remote database. Extreme care should be taken when using this command.

**shlib** Displays which packages link to a specific shared library.

**stats** Display package database statistics.

**unlock**

Unlocks packages, allowing them to be modified or deleted.

**update**

Update the available remote repositories as listed in pkg.conf(5).

**updating**

Display UPDATING entries of installed packages.

**upgrade**

Upgrade a package to a newer version.

**version**

Summarize installed versions of packages.

**which** Query the database for package(s) that installed a specific file.

**ENVIRONMENT**

All configuration options from pkg.conf(5) can be passed as environment variables.

Extra environment variables are:

**INSTALL\_AS\_USER** Allow all manipulation to be done as a regular user instead of checking for root credentials when appropriate.

It is expected that the user will ensure that every file and directory manipulated by **pkg** are readable (or writable where appropriate) by the user.

**FILES**

See pkg.conf(5).

**EXAMPLES**

Search for a package:

```
$ pkg search perl
```

Install a package:

Installing must specify a unique origin or version otherwise it will try installing all matches.

```
% pkg install perl-5.14
```

List installed packages:

```
$ pkg info
```

Upgrade from remote repository:

```
% pkg upgrade
```

Change the origin for an installed package:

```
% pkg set -o lang/perl5.12:lang/perl5.14
```

```
% pkg install -Rf lang/perl5.14
```

List non-automatic packages:

```
$ pkg query -e '%a = 0' %o
```

List automatic packages:

```
$ pkg query -e '%a = 1' %o
```

Delete an installed package:

```
% pkg delete perl-5.14
```

Remove unneeded dependencies:

```
% pkg autoremove
```

Change a package from automatic to non-automatic, which will prevent **autoremove** from removing it:

```
% pkg set -A 0 perl-5.14
```

Change a package from non-automatic to automatic, which will make **autoremove** allow it be removed once nothing depends on it:

```
% pkg set -A 1 perl-5.14
```

Create package file from an installed package:

```
% pkg create -o /usr/ports/packages/All perl-5.14
```

Determine which package installed a file:

```
$ pkg which /usr/local/bin/perl
```

Audit installed packages for security advisories:

```
$ pkg audit
```

Check installed packages for checksum mismatches:

```
# pkg check -s -a
```

Check for missing dependencies:

```
# pkg check -d -a
```

Show the pkg-message of a package:

```
# pkg info -D perl-5.14
```

Restore a backup database:

```
% rm /var/db/pkg/local.sqlite  
% xzcat /var/backups/pkg.sql.xz | pkg shell
```

## SEE ALSO

pkg\_create(3), pkg\_printf(3), pkg\_repos(3), pkg\_keywords(5), pkg\_lua-script(5), pkg-repository(5), pkg-script(5), pkg-triggers(5), pkg.conf(5), pkg-add(8), pkg-alias(8), pkg-annotate(8), pkg-audit(8), pkg-autoremove(8), pkg-check(8), pkg-clean(8), pkg-config(8), pkg-create(8), pkg-delete(8), pkg-fetch(8), pkg-info(8), pkg-install(8), pkg-lock(8), pkg-query(8), pkg-register(8), pkg-repo(8), pkg-rquery(8), pkg-search(8), pkg-set(8), pkg-shell(8), pkg-shlib(8), pkg-ssh(8), pkg-stats(8), pkg-triggers(8), pkg-update(8), pkg-updating(8), pkg-upgrade(8), pkg-version(8), pkg-which(8)

To build your own package set for one or multiple servers see poudriere(8) (*ports/ports-mgmt/poudriere*).

*FreeBSD pkg mirror:* <https://pkg.freebsd.org>

Your closest pkg mirror based on MaxMind GeoLite geo-DNS.

## HISTORY

The **pkg** command first appeared in FreeBSD 9.1.

## AUTHORS AND CONTRIBUTORS

Baptiste Daroussin <bapt@FreeBSD.org>, Julien Laffaye <jlaffaye@FreeBSD.org>, Philippe Pepiot <phil@philpep.org>, Will Andrews <will@FreeBSD.org>, Marin Atanasov Nikolov <dnaeon@gmail.com>, Yuri Pankov <yuri.pankov@gmail.com>, Alberto Villa <avilla@FreeBSD.org>, Brad Davis <brd@FreeBSD.org>, Matthew Seaman <matthew@FreeBSD.org>, Bryan Drewery <bryan@shatow.net>, Eitan Adler <eadler@FreeBSD.org>, Romain Tarti  re <romain@FreeBSD.org>, Vsevolod Stakhov <vsevolod@FreeBSD.org>, Alexandre Perrin <alex@kaworu.ch>

## BUGS

See the issue tracker at <https://github.com/freebsd/pkg/issues>.

Please direct questions and issues to the pkg@FreeBSD.org mailing list.