



Linux Ubuntu 26.04 Manual Pages on command 'pkgcli.1'

\$ man pkgcli.1

PKGCLI(1) pkgcli PKGCLI(1)

NAME

pkgcli - command-line client for PackageKit

SYNOPSIS

pkgcli [GLOBAL-OPTIONS...] COMMAND [COMMAND-OPTIONS...] [ARGUMENTS...]

DESCRIPTION

pkgcli is a command line client for PackageKit, a distribution-neutral interface to the operating system's package manager. It provides a set of subcommands for querying and managing software packages, repositories and transaction history.

Use pkgcli --help to get a list of available commands, and pkgcli COMMAND --help for command-specific help.

GLOBAL OPTIONS

The following options are parsed before the command name and affect most subcommands:

-v, --version

Show the pkgcli version and exit.

-h, --help

Show generic help and a list of available commands.

-q, --quiet

Only provide minimal output.

--verbose

Show more detailed and debug-style output.

--json

Output results in JSON format suitable for consumption by other tools. Output is

generated in the JSONL (JSON Lines) format, with one JSON object per line.

`--no-color`

Disable colored terminal output. This is also enabled when the `NO_COLOR` environment variable is set.

`-y, --yes`

Answer `?yes?` to all questions and run in non-interactive mode.

`-f, --filter=FILTER`

Apply one or more filters when querying packages, such as restricting results to installed or available packages. You can view all supported filter options by running `pkgcli organization`.

`-n, --background`

Hint that the command is not urgent and may use idle network bandwidth and power-saving settings where supported.

COMMANDS

The following commands are available:

`backend`

Display information about the PackageKit backend currently in use, including the backend name, version, and the roles it supports.

`history`

Show a list of recent package management transactions (like installation, removal, and update operations performed on the system).

`search`

Search for packages matching a given pattern. The first argument may optionally be one of `name`, `details`, `file` or `group`, to limit the search to the selected domain. If only one argument is given, a search across package details (name, summary, and description) is performed.

`list`

List packages, optionally filtered by installation status or name pattern. Use with `--filter` to show only installed, available, or other categories of packages.

`show`

Display detailed information about one or more packages, including version, size, description, license, and repository information.

`list-depends`

List all packages that the specified package depends on. Use the --recursive flag to list dependencies of the entire chain.

list-required-by

List packages that depend on or require the specified package.

what-provides

Find and list packages that provide a given capability, shared library, or file path.

files

Display a list of all files contained in the specified package.

list-updates

List all available package updates for installed software.

show-update

Display detailed information about a specific update, including changelog entries, security advisories, and other update metadata.

resolve

Resolve package names or identifiers to their package state, showing whether they are installed, available, or unavailable.

organization

Display available package filters (such as installed, available, newest), groups (such as system, development, graphics), and categories that can be used to query packages.

show-os-upgrade

Check for and display information about available operating system or distribution upgrades, such as upgrades to a newer OS release.

last-time

Show the elapsed time since the last transaction of a specified role occurred, such as the last package update or cache refresh.

refresh

Refresh the local package metadata cache by downloading the latest package information from configured repositories. This should be run before checking for updates or searching for new packages.

install

Install one or more packages by name, or install local package files if a path is provided instead of a package name.

remove

Remove one or more installed packages from the system. This may also remove dependent packages if they are no longer needed. Use with the `--autoremove` option to clean up unused packages as well.

update

Install available updates for all installed packages on the system, or for specific packages if specified as arguments.

upgrade

Perform a system upgrade to newer package versions or a new distribution release. This may involve major version updates and significant changes.

download

Download package files to the local system without installing them. This is useful for offline installation or package inspection.

offline-update

Prepare, trigger, cancel, or inspect offline system updates. Offline updates are applied during system boot before the desktop environment starts, allowing critical system packages to be updated safely.

install-sig

Install a cryptographic signature for a package to verify its authenticity and integrity.

repair

Attempt to automatically repair problems with the package management system, such as broken dependencies, corrupted databases, or interrupted transactions.

quit

Request the PackageKit daemon to terminate gracefully after completing any ongoing operations. The daemon will restart automatically when needed.

repo-list

Display a list of all configured software repositories, showing their identifiers, descriptions, and enabled/disabled status.

repo-enable

Enable a previously disabled repository, allowing packages from that repository to be installed and updated.

repo-disable

Disable a repository, preventing packages from that source from being installed or

updated while keeping the repository configuration.

repo-remove

Completely remove a repository from the system configuration. This is permanent and the repository will need to be re-added if needed later. Takes a repository-ID as parameter.

monitor

Monitor and display real-time PackageKit transaction events and status changes on the bus. This is useful for debugging or observing package management activity.

Use pkgcli COMMAND --help for detailed usage information including supported options and arguments for each command.

EXIT STATUS

The following exit codes are used by pkgcli:

0

The command completed successfully.

1

A non-specific error occurred.

2

Invalid syntax or command-line options were supplied.

3

Insufficient permissions to perform the requested operation.

4

Required information or packages could not be found.

5

The transaction failed.

Additional exit codes may be used for more specific failure conditions by individual commands.

ENVIRONMENT

NO_COLOR

When set to any non-empty value, disables colored output, equivalent to passing --no-color.

EXAMPLES

Search for packages related to text editors

```
pkgcli search editor
```

Search only in package names

```
pkgcli search name python3
```

Show detailed information about a package

```
pkgcli show nano
```

Install a package non-interactively

```
pkgcli --yes install nano
```

List all installed packages

```
pkgcli --filter=installed list
```

Check for available updates

```
pkgcli refresh && pkgcli list-updates
```

List dependencies of a package

```
pkgcli list-depends inkscape
```

Find which package provides the AV1 GStreamer codec

```
pkgcli what-provides "gststreamer1(decoder-video/x-av1)"
```

SEE ALSO

appstreamcli(1), and the PackageKit documentation at

<https://www.freedesktop.org/software/PackageKit/>.

AUTHORS

Matthias Klumpp <matthias@tenstral.net>

Developer

Richard Hughes <richard@hughsie.com>

Developer

COPYRIGHT

Copyright ? 2012-2026 Matthias Klumpp

PackageKit

26. November 2025

PKGCLI(1)