

NAME

apt-file -- APT package searching utility -- command-line interface

SYNOPSIS

apt-file [*options*] search *pattern*

apt-file [*options*] show *package*

DESCRIPTION

apt-file is a command line tool for searching files in packages for the APT package management system.

Some actions are required to run the search:

find

Alias for search.

list List the contents of a package. This action is very close to the **dpkg -L** command except the package does not need to be installed or fetched.

By default, the **list** action interprets its pattern as if **--fixed-string** was specified.

list-indices

List the known Content indices and their status in a human readable format. These indices can be searched via the **-I** option (when enabled and after fetching the indices). The output is subject to change without notice and is therefore unsuited for script usage / automation.

For a machine-readable formats (e.g. automation), please use **apt-config dump** (search for options starting with **Acquire::IndexTargets**) and **apt-get indextargets** for checking the cache (Look for entries with an *Identifier* field starting with *Contents-*).

Note that individual sources.list entries can override the global default. Please consult **sources.list** (5) for more information on advanced configuration of the **sources.list** files.

search

Search in which package a file is included. A list of all packages containing the pattern *pattern* is returned.

Since Contents files does not contain directories, the pattern must match (part of a) file name.

By default, the **search** action interprets its pattern as if **--substring-match** was specified.

show

Alias for list.

update

This action that just calls **apt update** or **apt-get update** (depending on whether a tty is available).

The only advantage using this over a regular **apt update** or **apt-get update** directly is for the case where you have configured an apt-file specific configuration (via the **Dir::Etc::apt-file-main** configuration option). In that case, said configuration will be included automatically.

OPTIONS

The following options are available:

-a, --architecture *architecture*[,...]

This option is useful if you search a package for a different architecture from the one installed on your system.

It can be a comma-separated list for searching on multiple architectures.

-c, --config-file *APT config-file*

Configuration File; Specify a configuration file to use. The program will read the default configuration file and then this configuration file. If configuration settings need to be set before the default configuration files are parsed specify a file with the **APT_CONFIG** environment variable. See **apt.conf** (5) for syntax information.

The configuration file will be read relative to when it appears on the command line and can overwrite options that appear before it.

Note that the config file will also be passed to all APT tools called by **apt-file**.

-D, --from-deb

Use contents of the given .deb archives(s) as patterns. Useful for searching for file conflicts with other packages. Implies **-F**.

-f, --from-file

Read patterns from the given file(s), one per line. Use **-** as filename for stdin. If no files are given, then the list will be read from stdin. This is much faster than invoking apt-file many times.

--filter-origins *origin*[,...]

Only search indices from the listed origins (e.g. “Debian”).

This filter matches against the name listed in the *Origin* field from the Release file.

If set to *****, this filter will be disabled (mostly useful for overriding the setting in a configuration file)

Alias of the APT config option: **apt-file::Search-Filter::Origin**

--filter-suites *suite*[,...]

Only search indices from the listed suites or codenames (e.g. “unstable”).

This filter matches against the name listed in the *Codename* and *Suite* fields from the Release file. This means that either “unstable” or “sid” will match Debian’s unstable suite.

If set to *****, this filter will be disabled (mostly useful for overriding the setting in a configuration file)

Alias of the APT config option: **apt-file::Search-Filter::Suite**

-F, --fixed-string

Do not expand search pattern with generic characters at pattern’s start and end.

This is default for **show** and **list** actions.

--index-names *type*[,...], -I *type*[,...]

Only search indexes of the given name(s). If set to the special value *ALL* (case-sensitive), then all **apt-file** indices are searched.

The name(s) must match one or more of the identifiers used in the APT configuration (minus leading “Contents-”). Example if the configuration has the following snippets:

```
Acquire::IndexTargets::deb::Contents-deb { ... };
Acquire::IndexTargets::deb-src::Contents-dsc { ... };
Acquire::IndexTargets::deb::Contents-udeb { ... };
Acquire::IndexTargets::deb::Contents-deb-legacy {
    # Explicitly named to "Contents-deb"
    Identifier "Contents-deb";
    ...;
};
```

Then, apt-file will recognise “deb”, “dsc” and “udeb” as index names.

This option defaults to the value of the “apt-file::Index-Names” apt config option (or “deb” if omitted).

-i, --ignore-case

Ignore case when searching for pattern.

-l, --package-only

Only display package name; do not display file names.

- o, --option APT::Option=Value**
 Set a Configuration Option; This will set an arbitrary configuration option. The syntax is **-o APT::Option=Value**. **-o** and **--option** can be used multiple times to set different options.

This option *can* be used to overwrite other command line options (e.g. “**-o apt-file::Search-Filter::Origin=Debian**” is effectively the same as “**--filter-origins Debian**”).

Note that the config options passed via this option will also be passed to all APT tools called by **apt-file**.
- substring-match**
 Match if the given search *pattern* is a substring of a path or package.

This is default for **search** and **find** actions.
- v, --verbose**
 Run apt-file in verbose/debug mode.
- x, --regexp**
 Treat pattern as a (perl) regular expression. See **perlref**(1) for details. Without this option, pattern is treated as a literal string to search for.

Be advised that this option can be rather slow. If performance is an issue, consider giving apt-file non-regex pattern matching too much and pipe the output to **perl -ne '/<pattern-here>/'**. This enables apt-file to use more optimizations and leaves less work to the “slower” regex.
- h, --help**
 Display a short help screen.

CONFIGURATION FILES

The apt-file command relies on the APT configuration. Notably, the default configuration makes **apt** fetch Contents files by default during a call to **apt update**.

For information on how to configure APT to fetch more or fewer Contents files, please refer to */usr/share/doc/apt-file/README.md.gz*.

The following files are notably interesting:

/etc/apt/apt-file.conf

Note this path is actually configurable by changing the value of the APT configuration called “Dir::Etc::apt-file-main”. The listed value is merely the default value of that option.

If this file is present, **apt-file** will read this file *after* all default APT configuration files. Any config file **-c** or option (**-o**) will be evaluated *before* this file (and can override options set in it).

The file will also be passed on to all APT tools called by **apt-file**.

/etc/apt/apt.conf.d/50apt-file.conf

Chooses which Contents files to download. Note that apt-file recognises only “Acquire” targets that starts with “Contents-”.

Special configurations similar to apt-file 2

Here are some **apt-file 2** related work flows and how to (almost) emulate them with **apt-file 3**. They are documented as a *starting point* for people, who are attached to these.

The emulation may not be perfect for you out of the box. Patches are welcome to keep the examples updated as long as the examples remain “trivial”.

- Only make **apt-file update** fetch Contents files

If you are accustomed to **apt update** *not* fetching Contents files, then you can run */usr/share/doc/apt-file/examples/apt-file-2-update.sh*. This script will configure **apt** and **apt-file** accordingly after best effort.

Please read the resulting */etc/apt/apt-file.conf*.

- Creating/using “user” caches

Previous versions of **apt-file** had a “`--cache`” option, which could be used to denote a directory to store the Contents files. This can be emulated by doing:

```
# Setup
$ mkdir -p ~/.cache/apt-file ~/.config
$ touch ~/.cache/apt-file/dpkg-status
$ sed '/^Dir::State/ d; /^Dir::Cache/ d;' \
    /usr/share/doc/apt-file/examples/apt-file.conf \
    > ~/.config/apt-file.conf
$ cat <<EOF >> ~/.config/apt-file.conf
Dir::State "$HOME/.cache/apt-file";
Dir::Cache "$HOME/.cache/apt-file";
Dir::State::status "$HOME/.cache/apt-file/dpkg-status";
EOF

# Update the cache
$ apt-file -c ~/.config/apt-file.conf update

# Search using the cache
$ apt-file -c ~/.config/apt-file.conf show apt-file

# Removal of the cache + config
$ rm -fr ~/.cache/apt-file ~/.config/apt-file.conf
```

(You will probably want to add an alias `apt-file` in your `~/.bashrc`)

Please read the resulting `./apt-file-user-cache.conf`.

BUGS, QUIRKS

There are some known issues or “quirks” that are good to keep in mind.

- The Contents files do **not** include a leading slash on paths. This means that `/bin/ls` is listed as `bin/ls` in the Contents file. If you are looking for something in a top-level directory, it is often better to omit the leading slash.

The search algorithm will attempt to work around the leading slash, but it will not work in all cases. As a workaround, try to pull the leading slash to the beginning of regular expressions. For example, use “`/(:usr/bin/vim|sbin/lvm)`” instead of “`/usr/bin/vim|sbin/lvm`”.

- When a new line has been added to the `sources.list` and `apt update` has not been run, `apt-file` does not print a warning message.
- By default, **apt-file** assumes that Contents files do *not* include a descriptive header (explaining what the file is and how to interpret it). However, some tools have generated them with such a header (e.g. for old versions of Contents files for the Debian archive or Contents files generated by `reprepro` prior to version 5.2.0).

If you search such files, you will want to set **apt-file::Parser::Check-For-Description-Header** to true (e.g. in `/etc/apt/apt.conf.d/50apt-file.conf`) to have **apt-file** properly filter out the headers to avoid false matches.

The reason this is not the default is that it costs a factor 2 in overhead while most common Contents files no longer have the header. (see #881405 for more details).

- Not all APT repositories have Contents files. Notably common install media (CDs etc.) may omit them to conserve space.

The default configuration by **apt-file** marks Contents files as optional and will just silently fail to search in Contents files in such repositories.

EXIT CODES

apt-file has the following defined exit codes, which can be used for scripting purposes.

- 0** **apt-file** returned successfully. If the command was a search, there was at least one result.
- 1** **apt-file** completed a search successfully, but it had no results.
- 2** An error occurred (including invalid/conflicting user options).
- 3** **apt-file** could not complete the command because the cache was empty. Please ensure there are indices enabled in the APT config and run **apt update** to fetch them.
- 4** **apt-file** could not complete the command because the cache does not have any files matching the restrictions. Either change the restrictions (e.g. **--index-names**) or configure apt to fetch the relevant files and run **apt update**.
- 255** There was an internal errors / uncaught exception in **apt-file**. Please file a bug against **apt-file**.

Any other exit code is reserved for future use.

SEE ALSO

apt(1), **apt-cache**(8), **apt.conf**(5)

The APT users guide in `/usr/share/doc/apt/`

The example config in `/usr/share/doc/apt-file/examples`

The README at `/usr/share/doc/apt-file/README.md.gz`