

NAME

apt-patterns – Syntax and semantics of apt search patterns

DESCRIPTION

Starting with version 2.0, **APT** provides support for patterns, which can be used to query the apt cache for packages.

LOGIC PATTERNS

These patterns provide the basic means to combine other patterns into more complex expressions, as well as ?true and ?false patterns.

?and(PATTERN, PATTERN, ...)

Selects objects where all specified patterns match.

?false

Selects nothing.

?not(PATTERN)

Selects objects where PATTERN does not match.

?or(PATTERN, PATTERN, ...)

Selects objects where at least one of the specified patterns match.

?true

Selects all objects.

PACKAGE PATTERNS

These patterns select specific packages.

?architecture(WILDCARD)

Selects packages matching the specified architecture, which may contain wildcards using any.

?automatic

Selects packages that were installed automatically.

?broken

Selects packages that have broken dependencies.

?config-files

Selects packages that are not fully installed, but have solely residual configuration files left.

?essential

Selects packages that have Essential: yes set in their control file.

?exact-name(NAME)

Selects packages with the exact specified name.

?garbage

Selects packages that can be removed automatically.

?installed

Selects packages that are currently installed.

?name(REGEX)

Selects packages where the name matches the given regular expression.

?obsolete

Selects packages that no longer exist in repositories.

?upgradable

Selects packages that can be upgraded (have a newer candidate).

?virtual

Selects all virtual packages; that is packages without a version. These exist when they are referenced somewhere in the archive, for example because something depends on that name.

EXAMPLES

`apt remove ?garbage`

Remove all packages that are automatically installed and no longer needed – same as `apt autoremove`

`apt purge ?config-files`

Purge all packages that only have configuration files left

MIGRATING FROM APTITUDE

Patterns in `apt` are heavily inspired by patterns in `aptitude`, but with some tweaks:

- Only long forms — the ones starting with `?` — are supported
- Syntax is uniform: If there is an opening parenthesis after a term, it is always assumed to be the beginning of an argument list.

In `aptitude`, a syntactic form `"?foo(bar)"` could mean `"?and(?foo,bar)"` if `foo` does not take an argument. In `APT`, this will cause an error.

- Not all patterns are supported.
- Some additional patterns are available, for example, for finding `gststreamer` codecs.
- Escaping terms with `~` is not supported.
- A trailing comma is allowed in argument lists

SEE ALSO

`apt-get(8)`, `apt(8)`

BUGS

[APT bug page](#)^[1]. If you wish to report a bug in `APT`, please see `/usr/share/doc/debian/bug-reporting.txt` or the `reportbug(1)` command.

AUTHOR

`APT` was written by the `APT` team <apt@packages.debian.org>.

AUTHORS

Jason Gunthorpe

APT team

NOTES

1. `APT` bug page
<http://bugs.debian.org/src:apt>