NAME

aptitude − high−level interface to the package manager

SYNOPSIS

aptitude [<*options*>...] {autoclean | clean | forget−new | keep−all | update}

aptitude [<*options*>...] {full−upgrade | safe−upgrade} [<*packages*>...]

aptitude [<*options*>...] {build−dep | build−depends | changelog | download | forbid−version | hold | install | markauto | purge | reinstall | remove | show | showsrc | source | unhold | unmarkauto | versions} <*packages*>...

aptitude extract−cache−subset <*output−directory*> <*packages*>...

aptitude [<*options*>...] search <*patterns*>...

aptitude [<*options*>...] {add−user−tag | remove−user−tag} <*tag*> <*packages*>...

aptitude [<*options*>...] {why | why−not} [<*patterns*>...] <*package*>

aptitude [−S <*fname*>] [−−autoclean−on−startup | −−clean−on−startup | −i | −u]

aptitude help

DESCRIPTION

aptitude is a text−based interface to the Debian GNU/Linux package system.

It allows the user to view the list of packages and to perform package management tasks such as installing, upgrading, and removing packages. Actions may be performed from a visual interface or from the command−line.

COMMAND−LINE ACTIONS

The first argument which does not begin with a hyphen ("**−**") is considered to be an action that the program should perform. If an action is not specified on the command−line, **aptitude** will start up in visual mode.

The following actions are available:

install

Install one or more packages. The packages should be listed after the "install" command; if a package name contains a tilde character ("**~**") or a question mark ("**?**"), it will be treated as a search pattern and every package matching the pattern will be installed (see the section "Search Patterns" in the **aptitude** reference manual).

To select a particular version of the package, append "**=<***version***>**" to the package name: for instance, "**aptitude install apt=0.3.1**". Similarly, to select a package from a particular archive, append "**/<***archive***>**" to the package name: for instance, "**aptitude install apt/experimental**". You cannot specify both an archive and a version for a package.

Not every package listed on the command line has to be installed; you can tell **aptitude** to do something different with a package by appending an "override specifier" to the name of the package. For example, **aptitude remove wesnoth+** will install **wesnoth**, not remove it. The following override specifiers are available:

<*package*>**+**

Install <*package*>.

If the package was not installed, it is marked as manually installed, and the dependencies newly installed are marked with the automatic flag. If the package or the dependencies were already installed, the automatic flag is preserved. See the section about automatic installations in the documentation for more information.

<*package*>**+M**

Install <*package*> and immediately mark it as automatically installed (note that if nothing depends on <*package*>, this will cause it to be immediately removed).

<*package*>**−**
: Remove <*package*>.

<*package*>**_**
: Purge <*package*>: remove it and all its associated configuration and data files.

<*package*>**=**
: Place <*package*> on hold: cancel any active installation, upgrade, or removal, and prevent this package from being automatically upgraded in the future.

<*package*>**:**
: Keep <*package*> at its current version: cancel any installation, removal, or upgrade. Unlike "hold" (above) this does not prevent automatic upgrades in the future.

<*package*>**&M**
: Mark <*package*> as having been automatically installed.

<*package*>**&m**
: Mark <*package*> as having been manually installed.

<*package*>**&BD**
: Install the build−dependencies of a <*package*>.

As a special case, "**install**" with no arguments will act on any stored/pending actions.

Note

Once you enter **Y** at the final confirmation prompt, the "**install**" command will modify **aptitude**'s stored information about what actions to perform. Therefore, if you issue (e.g.) the command "**aptitude install foo bar**" on packages previously uninstalled, and then the installation fails once **aptitude** has started downloading and installing packages, you will need to run "**aptitude remove foo bar**" to go back to the previous state (and possibly undo installations or upgrades to other packages that were affected by the "**install**" action).

remove, **purge**, **reinstall**
: These commands are the same as "**install**", but apply the named action to all packages given on the command line for which it is not overridden.

For instance, "**aptitude remove '~ndeity'**" will remove all packages whose name contains "**deity**".

build−depends, **build−dep**
: Satisfy the build−dependencies of a package. Each package name may be a source package, in which case the build dependencies of that source package are installed; otherwise, binary packages are found in the same way as for the "**install**" command, and the build−dependencies of the source packages that build those binary packages are satisfied.

If the command−line parameter **−−arch−only** is present, only architecture−dependent build dependencies (i.e., not **Build−Depends−Indep** or **Build−Conflicts−Indep**) will be obeyed.

markauto, **unmarkauto**
: Mark packages as automatically installed or manually installed, respectively. Packages are specified in exactly the same way as for the "**install**" command. For instance, "**aptitude markauto '~slibs'**" will mark all packages in the "**libs**" section as having been automatically installed.

For more information on automatically installed packages, see the section "Managing Automatically Installed Packages" in the **aptitude** reference manual.

hold, **unhold**, **keep**
: Mark packages to be on hold, remove this property, or set to keep in the current state. Packages are specified in exactly the same way as for the "**install**" command. For instance, "**aptitude hold '~e^dpkg$'**" will mark all packages coming from the source package "**dpkg**" to be on hold.

The difference between **hold** and **keep** is that **hold** will cause a package to be ignored by future **safe−upgrade** or **full−upgrade** commands, while **keep** merely cancels any scheduled actions on the package. **unhold** will allow a package to be upgraded by future **safe−upgrade** or **full−upgrade** commands, without otherwise altering its state.

keep−all
Cancels all scheduled actions on all packages; any packages whose sticky state indicates an installation, removal, or upgrade will have this sticky state cleared.

forget−new
Forgets all internal information about what packages are "new" (equivalent to pressing "f" when in visual mode).

This command accepts package names or patterns as arguments. If the string contains a tilde character ("**~**") or a question mark ("**?**"), it will be treated as a search pattern and every package matching the pattern will be considered (see the section "Search Patterns" in the **aptitude** reference manual).

forbid−version
Forbid a package from being upgraded to a particular version, while allowing automatic upgrades to future versions. This is useful for example to avoid a known broken version of a package, without having to set and clear manual holds.

By default, **aptitude** will select the forbidden version to be the one which the package would normally be upgraded (the candidate version). This may be overridden by appending "**=**<*version*>" to the package name: for instance, "**aptitude forbid−version vim=1.2.3.broken−4**".

To revert the action, "**aptitude install** <*package*>" will remove the ban. To remove the forbidden version without installing the candidate version, the current version should be appended: "install <*package*>**=**<*version*>".

update
Updates the list of available packages from the apt sources (this is equivalent to "**apt−get update**")

safe−upgrade
Upgrades installed packages to their most recent version. Installed packages will not be removed unless they are unused (see the section "Managing Automatically Installed Packages" in the **aptitude** reference manual). Packages which are not currently installed may be installed to resolve dependencies unless the **−−no−new−installs** command−line option is supplied.

If no <*package*>s are listed on the command line, **aptitude** will attempt to upgrade every package that can be upgraded. Otherwise, **aptitude** will attempt to upgrade only the packages which it is instructed to upgrade. The <*package*>s can be extended with suffixes in the same manner as arguments to **aptitude install**, so you can also give additional instructions to **aptitude** here; for instance, **aptitude safe−upgrade bash dash−** will attempt to upgrade the bash package and remove the dash package.

It is sometimes necessary to remove one package in order to upgrade another; this command is not able to upgrade packages in such situations. Use the **full−upgrade** command to upgrade as many packages as possible.

full−upgrade
Upgrades installed packages to their most recent version, removing or installing packages as necessary. It also installs new Essential or Required packages. This command is less conservative than **safe−upgrade** and thus more likely to perform unwanted actions. However, it is capable of upgrading packages that **safe−upgrade** cannot upgrade.

If no <*package*>s are listed on the command line, **aptitude** will attempt to upgrade every package that can be upgraded. Otherwise, **aptitude** will attempt to upgrade only the packages which it is instructed to upgrade. The <*package*>s can be extended with suffixes in the same manner as arguments to

aptitude install, so you can also give additional instructions to **aptitude** here; for instance, **aptitude full−upgrade bash dash−** will attempt to upgrade the bash package and remove the dash package.

Note

This command was originally named **dist−upgrade** for historical reasons, and **aptitude** still recognizes **dist−upgrade** as a synonym for **full−upgrade**.

search

Searches for packages matching one of the patterns supplied on the command line. All packages which match any of the given patterns will be displayed; for instance, “**aptitude search '~N' edit**” will list all “new” packages and all packages whose name contains “edit”. For more information on search patterns, see the section “Search Patterns” in the **aptitude** reference manual.

Note

In the example above, “**aptitude search '~N' edit**” has two arguments after **search** and thus is searching for *two* patterns: “**~N**” and “**edit**”. As described in the search pattern reference, a *single* pattern composed of two sub−patterns separated by a space (such as “**~N edit**”) matches only if *both* patterns match. Thus, the command “**aptitude search '~N edit'**” will only show “new” packages whose name contains “edit”.

Unless you pass the **−F** option, the output of **aptitude search** will look something like this:

```
i   apt                  − Advanced front−end for dpkg
pi  apt−build            − frontend to apt to build, optimize and in
cp  apt−file             − APT package searching utility −− command−
ihA raptor−utils         − Raptor RDF Parser utilities
```

Each search result is listed on a separate line. The first character of each line indicates the current state of the package: the most common states are **p**, meaning that no trace of the package exists on the system, **c**, meaning that the package was deleted but its configuration files remain on the system, **i**, meaning that the package is installed, and **v**, meaning that the package is virtual. The second character indicates the stored action (if any; otherwise a blank space is displayed) to be performed on the package, with the most common actions being **i**, meaning that the package will be installed, **d**, meaning that the package will be deleted, and **p**, meaning that the package and its configuration files will be removed. If the third character is **A**, the package was automatically installed.

For a complete list of the possible state and action flags, see the section “Accessing Package Information” in the **aptitude** reference guide. To customize the output of **search**, see the command−line options **−F** and **−−sort**.

show

Displays detailed information about one or more packages. If a package name contains a tilde character (“**~**”) or a question mark (“**?**”), it will be treated as a search pattern and all matching packages will be displayed (see the section “Search Patterns” in the **aptitude** reference manual).

If the verbosity level is 1 or greater (i.e., at least one **−v** is present on the command−line), information about all versions of the package is displayed. Otherwise, information about the “candidate version” (the version that “**aptitude install**” would download) is displayed.

You can display information about a different version of the package by appending **=<***version***>** to the package name; you can display the version from a particular archive or release by appending **/<***archive***>** or **/<***release***>** to the package name: for instance, **/unstable** or **/sid**. If either of these is present, then only the version you request will be displayed, regardless of the verbosity level.

If the verbosity level is 1 or greater, the package's architecture, compressed size, filename, and md5sum fields will be displayed. If the verbosity level is 2 or greater, the select version or versions will be displayed once for each archive in which they are found.

showsrc

Displays detailed information about one or more source packages.

This is a thin wrapper over **apt**(8).

source

Downloads one or more source packages.

This is a thin wrapper over **apt**(8).

versions

Displays the versions of the packages listed on the command−line.

```
$ aptitude versions wesnoth
p   1:1.4.5−1                                       100
p   1:1.6.5−1                    unstable           500
p   1:1.7.14−1                   experimental       1
```

Each version is listed on a separate line. The leftmost three characters indicate the current state, planned state (if any), and whether the package was automatically installed; for more information on their meanings, see the documentation of **aptitude search**. To the right of the version number you can find the releases from which the version is available, and the pin priority of the version.

If a package name contains a tilde character ("**~**") or a question mark ("**?**"), it will be treated as a search pattern and all matching *versions* will be displayed (see the section "Search Patterns" in the **aptitude** reference manual). This means that, for instance, **aptitude versions '~i'** will display all the versions that are currently installed on the system and nothing else, not even other versions of the same packages.

```
$ aptitude versions '~nexim4−daemon−light'
Package exim4−daemon−light:
i   4.71−3                                          100
p   4.71−4                       unstable           500

Package exim4−daemon−light−dbg:
p   4.71−4                       unstable           500
```

If the input is a search pattern, or if more than one package's versions are to be displayed, **aptitude** will automatically group the output by package, as shown above. You can disable this via **−−group−by=none**, in which case **aptitude** will display a single list of all the versions that were found and automatically include the package name in each output line:

```
$ aptitude versions −−group−by=none '~nexim4−daemon−light'
i   exim4−daemon−light 4.71−3                              100
p   exim4−daemon−light 4.71−4             unstable         500
p   exim4−daemon−light−dbg 4.71−4         unstable         500
```

To disable the package name, pass **−−show−package−names=never**:

```
$ aptitude versions −−show−package−names=never −−group−by=none '~nexim4−daemon−light'
i   4.71−3                                          100
p   4.71−4                       unstable           500
p   4.71−4                       unstable           500
```

In addition to the above options, the information printed for each version can be controlled by the command−line option **−F**. The order in which versions are displayed can be controlled by the command−line option **−−sort**. To prevent **aptitude** from formatting the output into columns, use **−−disable−columns**.

add−user−tag, **remove−user−tag**

Adds a user tag to or removes a user tag from the selected group of packages. If a package name contains a tilde ("**~**") or question mark ("**?**"), it is treated as a search pattern and the tag is added to or removed from all the packages that match the pattern (see the section "Search Patterns" in the **aptitude** reference manual).

User tags are arbitrary strings associated with a package. They can be used with the **?user−tag(<***tag***>)** search term, which will select all the packages that have a user tag matching <*tag*>.

why, **why−not**

Explains the reason that a particular package should or cannot be installed on the system.

This command searches for packages that require or conflict with the given package. It displays a sequence of dependencies leading to the target package, along with a note indicating the installed state of each package in the dependency chain:

```
$ aptitude why kdepim
i   nautilus−data Recommends nautilus
i A nautilus      Recommends desktop−base (>= 0.2)
i A desktop−base  Suggests   gnome | kde | xfce4 | wmaker
p   kde           Depends    kdepim (>= 4:3.4.3)
```

The command **why** finds a dependency chain that installs the package named on the command line, as above. Note that the dependency that **aptitude** produced in this case is only a suggestion. This is because no package currently installed on this computer depends on or recommends the kdepim package; if a stronger dependency were available, **aptitude** would have displayed it.

In contrast, **why−not** finds a dependency chain leading to a conflict with the target package:

```
$ aptitude why−not textopo
i   ocaml−core          Depends   ocamlweb
i A ocamlweb            Depends   tetex−extra | texlive−latex−extra
i A texlive−latex−extra Conflicts textopo
```

If one or more <*pattern*>s are present (in addition to the mandatory last argument, which should be a valid <*package*> name), then **aptitude** will begin its search at these patterns. That is, the first package in the chain it prints to explain why <*package*> is or is not installed, will be a package matching the pattern in question. The patterns are considered to be package names unless they contain a tilde character ("**~**") or a question mark ("**?**"), in which case they are treated as search patterns (see the section "Search Patterns" in the **aptitude** reference manual).

If no patterns are present, then **aptitude** will search for dependency chains beginning at manually installed packages. This effectively shows the packages that have caused or would cause a given package to be installed.

Note

aptitude why does not perform full dependency resolution; it only displays direct relationships between packages. For instance, if A requires B, C requires D, and B and C conflict, "**aptitude why−not D**" will not produce the answer "A depends on B, B conflicts with C, and D depends on C".

By default **aptitude** outputs only the "most installed, strongest, tightest, shortest" dependency chain. That is, it looks for a chain that only contains packages which are installed or will be installed; it looks for the strongest possible dependencies under that restriction; it looks for chains that avoid ORed dependencies and Provides; and it looks for the shortest dependency chain meeting those criteria. These rules are progressively weakened until a match is found.

If the verbosity level is 1 or more, then *all* the explanations **aptitude** can find will be displayed, in inverse order of relevance. If the verbosity level is 2 or more, a truly excessive amount of debugging information will be printed to standard output.

This command returns 0 if successful, 1 if no explanation could be constructed, and −1 if an error occurred.

clean
Removes all previously downloaded **.deb** files from the package cache directory (usually /var/cache/apt/archives).

autoclean
Removes any cached packages which can no longer be downloaded. This allows you to prevent a cache from growing out of control over time without completely emptying it.

changelog
Downloads and displays the Debian changelog for each of the given source or binary packages.

By default, the changelog for the version which would be installed with "**aptitude install**" is downloaded. You can select a particular version of a package by appending **=<***version***>** to the package name; you can select the version from a particular archive or release by appending **/<***archive***>** or **/<***release***>** to the package name (for instance, **/unstable** or **/sid**).

download
Downloads the **.deb** file for the given package to the current directory.

This is a thin wrapper over **apt**(8).

extract−cache−subset
Copy the apt configuration directory (**/etc/apt**) and a subset of the package database to the specified directory. If no packages are listed, the entire package database is copied; otherwise only the entries corresponding to the named packages are copied. Each package name may be a search pattern, and all the packages matching that pattern will be selected (see the section "Search Patterns" in the **aptitude** reference manual). Any existing package database files in the output directory will be overwritten.

Dependencies in binary package stanzas will be rewritten to remove references to packages not in the selected set.

help
Displays a brief summary of the available commands and options.

OPTIONS

The following options may be used to modify the behavior of the actions described above. Note that while all options will be accepted for all commands, some options don't apply to particular commands and will be ignored by those commands.

−−add−user−tag <*tag***>**
For **full−upgrade**, **safe−upgrade**, **forbid−version**, **hold**, **install**, **keep−all**, **markauto**, **unmarkauto**, **purge**, **reinstall**, **remove**, **unhold**, and **unmarkauto**: add the user tag <*tag*> to all packages that are installed, removed, or upgraded by this command as if with the **add−user−tag** command.

−−add−user−tag−to <*tag***>,<***pattern***>**
For **full−upgrade**, **safe−upgrade**, **forbid−version**, **hold**, **install**, **keep−all**, **markauto**, **unmarkauto**, **purge**, **reinstall**, **remove**, **unhold**, and **unmarkauto**: add the user tag <*tag*> to all packages that match <*pattern*> as if with the **add−user−tag** command. The pattern is a search pattern as described in the section "Search Patterns" in the **aptitude** reference manual.

For instance, **aptitude safe−upgrade −−add−user−tag−to "new−installs,?action(install)"** will add the tag **new−installs** to all the packages installed by the **safe−upgrade** command.

−−allow−new−upgrades
When the safe resolver is being used (i.e., **−−safe−resolver** was passed, the action is **safe−upgrade**, or

Aptitude::Always−Use−Safe−Resolver is set to **true**), allow the dependency resolver to install upgrades for packages regardless of the value of **Aptitude::Safe−Resolver::No−New−Upgrades**.

−−allow−new−installs
Allow the **safe−upgrade** command to install new packages; when the safe resolver is being used (i.e., **−−safe−resolver** was passed, the action is **safe−upgrade**, or **Aptitude::Always−Use−Safe−Resolver** is set to **true**), allow the dependency resolver to install new packages. This option takes effect regardless of the value of **Aptitude::Safe−Resolver::No−New−Installs**.

−−allow−untrusted
Install packages from untrusted sources without prompting. You should only use this if you know what you are doing, as it could easily compromise your system's security.

−−disable−columns
This option causes **aptitude search** and **aptitude versions** to output their results without any special formatting. In particular: normally **aptitude** will add whitespace or truncate search results in an attempt to fit its results into vertical "columns". With this flag, each line will be formed by replacing any format escapes in the format string with the corresponding text; column widths will be ignored.

For instance, the first few lines of output from "**aptitude search −F '%p %V' −−disable−columns libedataserver**" might be:

```
disksearch 1.2.1−3
hp−search−mac 0.1.3
libbsearch−ruby 1.5−5
libbsearch−ruby1.8 1.5−5
libclass−dbi−abstractsearch−perl 0.07−2
libdbix−fulltextsearch−perl 0.73−10
```

As in the above example, **−−disable−columns** is often useful in combination with a custom display format set using the command−line option **−F**.

This corresponds to the configuration option **Aptitude::CmdLine::Disable−Columns**.

−D, **−−show−deps**
For commands that will install or remove packages (**install**, **full−upgrade**, etc), show brief explanations of automatic installations and removals.

This corresponds to the configuration option **Aptitude::CmdLine::Show−Deps**.

−d, **−−download−only**
Download packages to the package cache as necessary, but do not install or remove anything. By default, the package cache is stored in /var/cache/apt/archives.

This corresponds to the configuration option **Aptitude::CmdLine::Download−Only**.

−F <*format*>, **−−display−format** <*format*>
Specify the format which should be used to display output from the **search** and **versions** commands. For instance, passing "**%p %v %V**" for <*format*> will display a package's name, followed by its currently installed version and its candidate version (see the section "Customizing how packages are displayed" in the **aptitude** reference manual for more information).

The command−line option **−−disable−columns** is often useful in combination with **−F**.

For **search**, this corresponds to the configuration option **Aptitude::CmdLine::Package−Display−Format**; for **versions**, this corresponds to the configuration option **Aptitude::CmdLine::Version−Display−Format**.

−f

Try hard to fix the dependencies of broken packages, even if it means ignoring the actions requested on the command line.

This corresponds to the configuration item **Aptitude::CmdLine::Fix−Broken**.

−−full−resolver
When package dependency problems are encountered, use the default “full” resolver to solve them. Unlike the “safe” resolver activated by **−−safe−resolver**, the full resolver will happily remove packages to fulfill dependencies. It can resolve more situations than the safe algorithm, but its solutions are more likely to be undesirable.

This option can be used to force the use of the full resolver even when **Aptitude::Always−Use−Safe−Resolver** is true.

−−group−by <*grouping−mode*>
Control how the **versions** command groups its output. The following values are recognized:

- **archive** to group packages by the archive they occur in (“**stable**”, “**unstable**”, etc). If a package occurs in several archives, it will be displayed in each of them.
- **auto** to group versions by their package unless there is exactly one argument and it is not a search pattern.
- **none** to display all the versions in a single list without any grouping.
- **package** to group versions by their package.
- **source−package** to group versions by their source package.
- **source−version** to group versions by their source package and source version.

This corresponds to the configuration option **Aptitude::CmdLine::Versions−Group−By**.

−h, **−−help**
Display a brief help message. Identical to the **help** action.

−−log−file=<*file***>**
If <*file*> is a nonempty string, log messages will be written to it, except that if <*file*> is “**−**”, the messages will be written to standard output instead. If this option appears multiple times, the last occurrence is the one that will take effect.

This does not affect the log of installations that **aptitude** has performed (/var/log/aptitude); the log messages written using this configuration include internal program events, errors, and debugging messages. See the command−line option **−−log−level** to get more control over what gets logged.

This corresponds to the configuration option **Aptitude::Logging::File**.

−−log−level=<*level***>**, **−−log−level=<***category***>:<***level***>**
−−log−level=<*level***>** causes **aptitude** to only log messages whose level is <*level*> or higher. For instance, setting the log level to **error** will cause only messages at the log levels **error** and **fatal** to be displayed; all others will be hidden. Valid log levels (in descending order) are **off**, **fatal**, **error**, **warn**, **info**, **debug**, and **trace**. The default log level is **warn**.

−−log−level=<*category***>:<***level***>** causes messages in <*category*> to only be logged if their level is <*level*> or higher.

−−log−level may appear multiple times on the command line; the most specific setting is the one that takes effect, so if you pass **−−log−level=aptitude.resolver:fatal** and **−−log−level=aptitude.resolver.hints.match:trace**, then messages in **aptitude.resolver.hints.parse** will only be printed if their level is **fatal**, but all messages in **aptitude.resolver.hints.match** will be printed. If you set the level of the same category two or more times, the last setting is the one that will take effect.

This does not affect the log of installations that **aptitude** has performed (/var/log/aptitude); the log messages written using this configuration include internal program events, errors, and debugging messages. See the command−line option **−−log−file** to change where log messages go.

This corresponds to the configuration group **Aptitude::Logging::Levels**.

−−log−resolver
Set some standard log levels related to the resolver, to produce logging output suitable for processing with automated tools. This is equivalent to the command−line options **−−log−level=aptitude.resolver.search:trace −−log−level=aptitude.resolver.search.tiers:info**.

−−no−new−installs
Prevent **safe−upgrade** from installing any new packages; when the safe resolver is being used (i.e., **−−safe−resolver** was passed or **Aptitude::Always−Use−Safe−Resolver** is set to **true**), forbid the dependency resolver from installing new packages. This option takes effect regardless of the value of **Aptitude::Safe−Resolver::No−New−Installs**.

This mimics the historical behavior of **apt−get upgrade**.

−−no−new−upgrades
When the safe resolver is being used (i.e., **−−safe−resolver** was passed or **Aptitude::Always−Use−Safe−Resolver** is set to **true**), forbid the dependency resolver from installing upgrades for packages regardless of the value of **Aptitude::Safe−Resolver::No−New−Upgrades**.

−−no−show−resolver−actions
Do not display the actions performed by the “safe” resolver, overriding any configuration option or earlier **−−show−resolver−actions**.

−O <*order*>, **−−sort** <*order*>
Specify the order in which output from the **search** and **versions** commands should be displayed. For instance, passing “**installsize**” for <*order*> will list packages in order according to their size when installed (see the section “Customizing how packages are sorted” in the **aptitude** reference manual for more information).

Prepending the order keyword with a tilde character (˜) reverses the order from ascending to descending.

The default sort order is **name,version**.

−o <*key*>**=**<*value*>
Set a configuration file option directly; for instance, use **−o Aptitude::Log=/tmp/my−log** to log **aptitude**'s actions to /tmp/my−log. For more information on configuration file options, see the section “Configuration file reference” in the **aptitude** reference manual.

−P, **−−prompt**
Always display a prompt before downloading, installing or removing packages, even when no actions other than those explicitly requested will be performed.

This corresponds to the configuration option **Aptitude::CmdLine::Always−Prompt**.

−−purge−unused
If **Aptitude::Delete−Unused** is set to “**true**” (its default), then in addition to removing each package that is no longer required by any installed package, **aptitude** will also purge them, removing their configuration files and perhaps other important data. For more information about which packages are considered to be “unused”, see the section “Managing Automatically Installed Packages” in the **aptitude** reference manual. *THIS OPTION CAN CAUSE DATA LOSS! DO NOT USE IT UNLESS YOU KNOW WHAT YOU ARE DOING!*

This corresponds to the configuration option **Aptitude::Purge−Unused**.

−q[=<*n*>], **−−quiet[=<*n*>]**
Suppress all incremental progress indicators, thus making the output loggable. This may be supplied multiple times to make the program quieter, but unlike **apt−get**, **aptitude** does not enable **−y** when **−q** is supplied more than once.

The optional **=<*n*>** may be used to directly set the amount of quietness (for instance, to override a setting in /etc/apt/apt.conf); it causes the program to behave as if **−q** had been passed exactly <*n*> times.

−R, **−−without−recommends**
Do *not* treat recommendations as dependencies when installing new packages (this overrides settings in /etc/apt/apt.conf and ~/.aptitude/config). Packages previously installed due to recommendations will not be removed.

This corresponds to the pair of configuration options **APT::Install−Recommends** and **APT::AutoRemove::RecommendsImportant**.

−r, **−−with−recommends**
Treat recommendations as dependencies when installing new packages (this overrides settings in /etc/apt/apt.conf and ~/.aptitude/config).

This corresponds to the configuration option **APT::Install−Recommends**

−−remove−user−tag <*tag*>
For **full−upgrade**, **safe−upgrade forbid−version**, **hold**, **install**, **keep−all**, **markauto**, **unmarkauto**, **purge**, **reinstall**, **remove**, **unhold**, and **unmarkauto**: remove the user tag <*tag*> from all packages that are installed, removed, or upgraded by this command as if with the **add−user−tag** command.

−−remove−user−tag−from <*tag*>,<*pattern*>
For **full−upgrade**, **safe−upgrade forbid−version**, **hold**, **install**, **keep−all**, **markauto**, **unmarkauto**, **purge**, **reinstall**, **remove**, **unhold**, and **unmarkauto**: remove the user tag <*tag*> from all packages that match <*pattern*> as if with the **remove−user−tag** command. The pattern is a search pattern as described in the section “Search Patterns” in the **aptitude** reference manual.

For instance, **aptitude safe−upgrade −−remove−user−tag−from "not−upgraded,?action(upgrade)"** will remove the **not−upgraded** tag from all packages that the **safe−upgrade** command is able to upgrade.

−s, **−−simulate**
In command−line mode, print the actions that would normally be performed, but don't actually perform them. This does not require root privileges. In the visual interface, always open the cache in read−only mode regardless of whether you are root.

This corresponds to the configuration option **Aptitude::Simulate**.

−−safe−resolver
When package dependency problems are encountered, use a “safe” algorithm to solve them. This resolver attempts to preserve as many of your choices as possible; it will never remove a package or install a version of a package other than the package's default candidate version. It is the same algorithm used in **safe−upgrade**; indeed, **aptitude −−safe−resolver full−upgrade** is equivalent to **aptitude safe−upgrade**. Because **safe−upgrade** always uses the safe resolver, it does not accept the **−−safe−resolver** flag.

This option is equivalent to setting the configuration variable **Aptitude::Always−Use−Safe−Resolver** to **true**.

−−schedule−only

For commands that modify package states, schedule operations to be performed in the future, but don't perform them. You can execute scheduled actions by running **aptitude install** with no arguments. This is equivalent to making the corresponding selections in visual mode, then exiting the program normally.

For instance, **aptitude −−schedule−only install evolution** will schedule the **evolution** package for later installation.

−−show−package−names <*when*>

Controls when the **versions** command shows package names. The following settings are allowed:

- **always**: display package names every time that **aptitude versions** runs.
- **auto**: display package names when **aptitude versions** runs if the output is not grouped by package, and either there is a pattern−matching argument or there is more than one argument.
- **never**: never display package names in the output of **aptitude versions**.

This option corresponds to the configuration item **Aptitude::CmdLine::Versions−Show−Package−Names**.

−−show−resolver−actions

Display the actions performed by the "safe" resolver and by **safe−upgrade**.

When executing the command **safe−upgrade** or when the option −−safe−resolver is present, **aptitude** will display a summary of the actions performed by the resolver before printing the installation preview. This is equivalent to the configuration option **Aptitude::Safe−Resolver::Show−Resolver−Actions**.

−−show−summary[=<*MODE*>]

Changes the behavior of "**aptitude why**" to summarize each dependency chain that it outputs, rather than displaying it in long form. If this option is present and <*MODE*> is not "**no−summary**", chains that contain Suggests dependencies will not be displayed: combine **−−show−summary** with **−v** to see a summary of all the reasons for the target package to be installed.

<*MODE*> can be any one of the following:

1. **no−summary**: don't show a summary (the default behavior if **−−show−summary** is not present).
2. **first−package**: display the first package in each chain. This is the default value of <*MODE*> if it is not present.
3. **first−package−and−type**: display the first package in each chain, along with the strength of the weakest dependency in the chain.
4. **all−packages**: briefly display each chain of dependencies leading to the target package.
5. **all−packages−with−dep−versions**: briefly display each chain of dependencies leading to the target package, including the target version of each dependency.

This option corresponds to the configuration item **Aptitude::CmdLine::Show−Summary**; if **−−show−summary** is present on the command−line, it will override **Aptitude::CmdLine::Show−Summary**.

Example 12. Usage of −−show−summary −−show−summary used with **−v** to display all the reasons a package is installed:

```
$ aptitude −v −−show−summary why foomatic−db
Packages requiring foomatic−db:
  cupsys−driver−gutenprint
  foomatic−db−engine
  foomatic−db−gutenprint
  foomatic−db−hpijs
  foomatic−filters−ppds
  foomatic−gui
  kde
  printconf
  wine

$ aptitude −v −−show−summary=first−package−and−type why foomatic−db
Packages requiring foomatic−db:
  [Depends] cupsys−driver−gutenprint
  [Depends] foomatic−db−engine
  [Depends] foomatic−db−gutenprint
  [Depends] foomatic−db−hpijs
  [Depends] foomatic−filters−ppds
  [Depends] foomatic−gui
  [Depends] kde
  [Depends] printconf
  [Depends] wine

$ aptitude −v −−show−summary=all−packages why foomatic−db
Packages requiring foomatic−db:
  cupsys−driver−gutenprint D: cups−driver−gutenprint D: cups R: foomatic−filters R: foomatic−db−engine D: foomatic−
  foomatic−filters−ppds D: foomatic−filters R: foomatic−db−engine D: foomatic−db
  kde D: kdeadmin R: system−config−printer−kde D: system−config−printer R: hal−cups−utils D: cups R: foomatic−filte
  wine D: libwine−print D: cups−bsd R: cups R: foomatic−filters R: foomatic−db−engine D: foomatic−db
  foomatic−db−engine D: foomatic−db
  foomatic−db−gutenprint D: foomatic−db
  foomatic−db−hpijs D: foomatic−db
  foomatic−gui D: python−foomatic D: foomatic−db−engine D: foomatic−db
  printconf D: foomatic−db

$ aptitude −v −−show−summary=all−packages−with−dep−versions why foomatic−db
Packages requiring foomatic−db:
  cupsys−driver−gutenprint D: cups−driver−gutenprint (>= 5.0.2−4) D: cups (>= 1.3.0) R: foomatic−filters (>= 4.0) R: fo
  foomatic−filters−ppds D: foomatic−filters R: foomatic−db−engine (>= 4.0) D: foomatic−db (>= 20090301)
  kde D: kdeadmin (>= 4:3.5.5) R: system−config−printer−kde (>= 4:4.2.2−1) D: system−config−printer (>= 1.0.0) R: ha
  wine D: libwine−print (= 1.1.15−1) D: cups−bsd R: cups R: foomatic−filters (>= 4.0) R: foomatic−db−engine (>= 4.0)
  foomatic−db−engine D: foomatic−db
  foomatic−db−gutenprint D: foomatic−db
  foomatic−db−hpijs D: foomatic−db
  foomatic−gui D: python−foomatic (>= 0.7.9.2) D: foomatic−db−engine D: foomatic−db (>= 20090301)
  printconf D: foomatic−db
```

−−show−summary used to list a chain on one line:

```
$ aptitude −−show−summary=all−packages why aptitude−gtk libglib2.0−data
Packages requiring libglib2.0−data:
  aptitude−gtk D: libglib2.0−0 R: libglib2.0−data
```

−t <*release*>, **−−target−release** <*release*>

Set the release from which packages should be installed. For instance, "**aptitude −t experimental ...**" will install packages from the experimental distribution unless you specify otherwise.

This will affect the default candidate version of packages according to the rules described in **apt_preferences**(5).

This corresponds to the configuration item **APT::Default−Release**.

−V, **−−show−versions**
Show which versions of packages will be installed.

This corresponds to the configuration option **Aptitude::CmdLine::Show−Versions**.

−v, **−−verbose**
Causes some commands (for instance, **show**) to display extra information. This may be supplied multiple times to get more and more information.

This corresponds to the configuration option **Aptitude::CmdLine::Verbose**.

−−version
Display the version of **aptitude** and some information about how it was compiled.

−−visual−preview
When installing or removing packages from the command line, instead of displaying the usual prompt, start up the visual interface and display its preview screen.

−W, **−−show−why**
In the preview displayed before packages are installed or removed, show which manually installed package requires each automatically installed package. For instance:

```
$ aptitude −−show−why install mediawiki
...
The following NEW packages will be installed:
  libapache2−mod−php5{a} (for mediawiki)  mediawiki  php5{a} (for mediawiki)
  php5−cli{a} (for mediawiki)  php5−common{a} (for mediawiki)
  php5−mysql{a} (for mediawiki)
```

When combined with **−v** or a non−zero value for **Aptitude::CmdLine::Verbose**, this displays the entire chain of dependencies that lead each package to be installed. For instance:

```
$ aptitude −v −−show−why install libdb4.2−dev
The following NEW packages will be installed:
  libdb4.2{a} (libdb4.2−dev D: libdb4.2)  libdb4.2−dev
The following packages will be REMOVED:
  libdb4.4−dev{a} (libdb4.2−dev C: libdb−dev P<− libdb−dev)
```

This option will also describe why packages are being removed, as shown above. In this example, libdb4.2−dev conflicts with libdb−dev, which is provided by libdb−dev.

This argument corresponds to the configuration option **Aptitude::CmdLine::Show−Why** and displays the same information that is computed by **aptitude why** and **aptitude why−not**.

−w <*width*>, **−−width** <*width*>
Specify the display width which should be used for output from the **search** and **versions** commands (in the command line).

By default and when the output is seen directly in a terminal, the terminal width is used. When the output is redirected or piped, a very large "unlimited" line width is used, and this option is ignored.

This corresponds to the configuration option **Aptitude::CmdLine::Package−Display−Width**

−y, **−−assume−yes**

When a yes/no prompt would be presented, assume that the user entered “yes”. In particular, suppresses the prompt that appears when installing, upgrading, or removing packages. Prompts for “dangerous” actions, such as removing essential packages, will still be displayed. This option overrides **−P**.

This corresponds to the configuration option **Aptitude::CmdLine::Assume−Yes**.

−Z

Show how much disk space will be used or freed by the individual packages being installed, upgraded, or removed.

This corresponds to the configuration option **Aptitude::CmdLine::Show−Size−Changes**.

The following options apply to the visual mode of the program, but are primarily for internal use; you generally won't need to use them yourself.

−−autoclean−on−startup

Deletes old downloaded files when the program starts (equivalent to starting the program and immediately selecting Actions → Clean obsolete files). You cannot use this option and “**−−clean−on−startup**”, “**−i**”, or “**−u**” at the same time.

−−clean−on−startup

Cleans the package cache when the program starts (equivalent to starting the program and immediately selecting Actions → Clean package cache). You cannot use this option and “**−−autoclean−on−startup**”, “**−i**”, or “**−u**” at the same time.

−i

Displays a download preview when the program starts (equivalent to starting the program and immediately pressing “g”). You cannot use this option and “**−−autoclean−on−startup**”, “**−−clean−on−startup**”, or “**−u**” at the same time.

−S <*fname*>

Loads the extended state information from <*fname*> instead of the standard state file.

−u

Begins updating the package lists as soon as the program starts. You cannot use this option and “**−−autoclean−on−startup**”, “**−−clean−on−startup**”, or “**−i**” at the same time.

ENVIRONMENT

HOME

If $HOME/.aptitude exists, **aptitude** will store its configuration file in $HOME/.aptitude/config. Otherwise, it will look up the current user's home directory using **getpwuid**(2) and place its configuration file there.

PAGER

If this environment variable is set, **aptitude** will use it to display changelogs when “**aptitude changelog**” is invoked. If not set, it defaults to **more**.

TMP

If **TMPDIR** is unset, **aptitude** will store its temporary files in **TMP** if that variable is set. Otherwise, it will store them in /tmp.

TMPDIR

aptitude will store its temporary files in the directory indicated by this environment variable. If **TMPDIR** is not set, then **TMP** will be used; if **TMP** is also unset, then **aptitude** will use /tmp.

FILES

/var/lib/aptitude/pkgstates

The file in which stored package states and some package flags are stored.

/etc/apt/apt.conf, /etc/apt/apt.conf.d/*, ~/.aptitude/config

The configuration files for **aptitude**. ~/.aptitude/config overrides /etc/apt/apt.conf. See **apt.conf**(5) for documentation of the format and contents of these files.

SEE ALSO

apt-get(8), **apt**(8), /usr/share/doc/aptitude/html/<*lang*>/index.html from the package aptitude−doc−<*lang*>

AUTHORS

Daniel Burrows <dburrows@debian.org>
Main author of the document.

Manuel A. Fernandez Montecelo <mafm@debian.org>
Main maintainer after Daniel Burrows, documentation about new features, corrections and formatting.

COPYRIGHT