NAME

debi − install current version of generated Debian package

SYNOPSIS

debi [*options*] [*changes file*] [*package ...*]

DESCRIPTION

debi figures out the current version of a package and installs it. If a *.changes* file is specified on the command line, the filename must end with *.changes*, as this is how the program distinguishes it from package names. If not, then **debi** has to be called from within the source code directory tree. In this case, it will look for the *.changes* file corresponding to the current package version (by determining the name and version number from the changelog, and the architecture in the same way as **dpkg-buildpackage**(1) does). It then runs **debpkg −i** on every *.deb* archive listed in the *.changes* file to install them, assuming that all of the *.deb* archives live in the same directory as the *.changes* file. Note that you probably don't want to run this program on a *.changes* file relating to a different architecture after cross-compiling the package!

If a list of packages is given on the command line, then only those debs with names in this list of packages will be installed.

Since installing a package requires root privileges, **debi** calls **debpkg** rather than **dpkg** directly. Thus **debi** will only be useful if it is either being run as root or **debpkg** can be run as root. See **debpkg**(1) for more details.

Directory name checking

In common with several other scripts in the **devscripts** package, **debi** will climb the directory tree until it finds a *debian/changelog* file. As a safeguard against stray files causing potential problems, it will examine the name of the parent directory once it finds the *debian/changelog* file, and check that the directory name corresponds to the package name. Precisely how it does this is controlled by two configuration file variables **DEVSCRIPTS_CHECK_DIRNAME_LEVEL** and **DEVSCRIPTS_CHECK_DIRNAME_REGEX**, and their corresponding command-line options **−−check-dirname-level** and **−−check-dirname-regex**.

DEVSCRIPTS_CHECK_DIRNAME_LEVEL can take the following values:

0 Never check the directory name.

1 Only check the directory name if we have had to change directory in our search for *debian/changelog*. This is the default behaviour.

2 Always check the directory name.

The directory name is checked by testing whether the current directory name (as determined by **pwd**(1)) matches the regex given by the configuration file option **DEVSCRIPTS_CHECK_DIRNAME_REGEX** or by the command line option **−−check-dirname-regex** *regex*. Here *regex* is a Perl regex (see **perlre**(3perl)), which will be anchored at the beginning and the end. If *regex* contains a '/', then it must match the full directory path. If not, then it must match the full directory name. If *regex* contains the string ´PACKAGE', this will be replaced by the source package name, as determined from the changelog. The default value for the regex is: ´PACKAGE(-.+)?', thus matching directory names such as PACKAGE and PACKAGE-version.

OPTIONS

−a*debian-architecture*, **−t***GNU-system-type*

See **dpkg-architecture**(1) for a description of these options. They affect the search for the *.changes* file. They are provided to mimic the behaviour of **dpkg-buildpackage** when determining the name of the *.changes* file.

−−debs−dir *directory*

Look for the *.changes* and *.deb* files in *directory* instead of the parent of the source directory. This should either be an absolute path or relative to the top of the source directory.

−m, **−−multi**
Search for a multiarch *.changes* file, as created by **dpkg-cross**.

−u, **−−upgrade**
Only upgrade packages already installed on the system, rather than installing all packages listed in the *.changes* file. Useful for multi-binary packages when you don't want to have all the binaries installed at once.

−−check-dirname-level *N*
See the above section **Directory name checking** for an explanation of this option.

−−check-dirname-regex *regex*
See the above section **Directory name checking** for an explanation of this option.

−−with-depends
Attempt to satisfy the *Depends* of a package when installing it.

−−tool *tool*
Use the specified *tool* for installing the dependencies of the package(s) to be installed. By default, **apt-get** is used.

−−no-conf, **−−noconf**
Do not read any configuration files. This can only be used as the first option given on the command-line.

−−help, **−−version**
Show help message and version information respectively.

CONFIGURATION VARIABLES

The two configuration files */etc/devscripts.conf* and *~/.devscripts* are sourced in that order to set configuration variables. Command line options can be used to override configuration file settings. Environment variable settings are ignored for this purpose. The currently recognised variables are:

DEBRELEASE_DEBS_DIR
This specifies the directory in which to look for the *.changes* and *.deb* files, and is either an absolute path or relative to the top of the source tree. This corresponds to the **−−debs−dir** command line option. This directive could be used, for example, if you always use **pbuilder** or **svn-buildpackage** to build your packages. Note that it also affects **debrelease**(1) in the same way, hence the strange name of the option.

DEVSCRIPTS_CHECK_DIRNAME_LEVEL, **DEVSCRIPTS_CHECK_DIRNAME_REGEX**
See the above section **Directory name checking** for an explanation of these variables. Note that these are package-wide configuration variables, and will therefore affect all **devscripts** scripts which check their value, as described in their respective manpages and in **devscripts.conf**(5).

SEE ALSO

debpkg(1), **devscripts.conf**(5)

AUTHOR

debi was originally written by Christoph Lameter <clameter@debian.org>. The now-defunct script **debit** was originally written by James R. Van Zandt <jrv@vanzandt.mv.com>. They have been moulded into one script together with **debc**(1) and parts extensively modified by Julian Gilbey <jdg@debian.org>.