

**NAME**

**debc** – view contents of a generated Debian package

**SYNOPSIS**

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

**DESCRIPTION**

**debc** figures out the current version of a package and displays information about the *.deb* and *.udeb* files which have been generated in the current build process. 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 **debc** 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 **dpkg-deb -I** and **dpkg-deb -c** on every *.deb* and *.udeb* archive listed in the *.changes* file to display information about the contents of the *.deb* / *.udeb* files. It precedes every *.deb* or *.udeb* file with the name of the file. It assumes that all of the *.deb* / *.udeb* archives live in the same directory as the *.changes* file. It is useful for ensuring that the expected files have ended up in the Debian package.

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

**Directory name checking**

In common with several other scripts in the **devscripts** package, **debc** 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**

**-adebian-architecture, -tGNU-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*, *.deb* and *.udeb* 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.

**--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.

**--list-changes**

List the filename of the `.changes` file, and do not display anything else. This option only makes sense if a `.changes` file is NOT passed explicitly in the command line. This can be used for example in a script that needs to reference the `.changes` file, without having to duplicate the heuristics for finding it that `debc` already implements.

**--list-debs**

List the filenames of the `.deb` packages, and do not display their contents.

**--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:

**DEBREASE\_DEBS\_DIR**

This specifies the directory in which to look for the `.changes`, `.deb` and `.udeb` 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**

**debdiff(1)**, **dpkg-deb(1)**, **devscripts.conf(5)**

**AUTHOR**

Julian Gilbey <jdg@debian.org>, based on an original script by Christoph Lameter <clameter@debian.org>.