NAME

debcommit – commit changes to a package

SYNOPSIS

debcommit [*options*] [**−−all** | *files to commit*]

DESCRIPTION

debcommit generates a commit message based on new text in **debian/changelog**, and commits the change to a package's repository. It must be run in a working copy for the package. Supported version control systems are: **cvs**, **git**, **hg** (mercurial), **svk**, **svn** (Subversion), **baz**, **bzr**, **tla** (arch), **darcs**.

OPTIONS

−c, **−−changelog** *path*
Specify an alternate location for the changelog. By default debian/changelog is used.

−r, **−−release**
Commit a release of the package. The version number is determined from debian/changelog, and is used to tag the package in the repository.

Note that svn/svk tagging conventions vary, so debcommit uses **svnpath**(1) to determine where the tag should be placed in the repository.

−R, **−−release−use−changelog**
When used in conjunction with **−−release**, if there are uncommitted changes to the changelog then derive the commit message from those changes rather than using the default message.

−m *text*, **−−message** *text*
Specify a commit message to use. Useful if the program cannot determine a commit message on its own based on debian/changelog, or if you want to override the default message.

−n, **−−noact**
Do not actually do anything, but do print the commands that would be run.

−d, **−−diff**
Instead of committing, do print the diff of what would have been committed if this option were not given. A typical usage scenario of this option is the generation of patches against the current working copy (e.g. when you don't have commit access right).

−C, **−−confirm**
Display the generated commit message and ask for confirmation before committing it. It is also possible to edit the message at this stage; in this case, the confirmation prompt will be re-displayed after the editing has been performed.

−e, **−−edit**
Edit the generated commit message in your favorite editor before committing it.

−a, **−−all**
Commit all files. This is the default operation when using a VCS other than git.

−s, **−−strip−message**, **−−no−strip−message**
If this option is set and the commit message has been derived from the changelog, the characters "* " will be stripped from the beginning of the message.

This option is set by default and ignored if more than one line of the message begins with "[*+−] ".

−−sign−commit, **−−no−sign−commit**
If this option is set, then the commits that debcommit creates will be signed using gnupg. Currently this is only supported by git, hg, and bzr.

−−sign−tags, **−−no−sign−tags**
If this option is set, then tags that debcommit creates will be signed using gnupg. Currently this is only supported by git.

−−changelog−info
If this option is set, the commit author and date will be determined from the Maintainer and Date field of the first paragraph in *debian/changelog*. This is mainly useful when using **debchange**(1) with the **−−no−mainttrailer** option.

CONFIGURATION VARIABLES

The two configuration files */etc/devscripts.conf* and *~/.devscripts* are sourced by a shell 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:

DEBCOMMIT_STRIP_MESSAGE
If this is set to *no*, then it is the same as the **−−no−strip−message** command line parameter being used. The default is *yes*.

DEBCOMMIT_SIGN_TAGS
If this is set to *yes*, then it is the same as the **−−sign−tags** command line parameter being used. The default is *no*.

DEBCOMMIT_SIGN_COMMITS
If this is set to *yes*, then it is the same as the **−−sign−commit** command line parameter being used. The default is *no*.

DEBCOMMIT_RELEASE_USE_CHANGELOG
If this is set to *yes*, then it is the same as the **−−release−use−changelog** command line parameter being used. The default is *no*.

DEBSIGN_KEYID
This is the key id used for signing tags. If not set, a default will be chosen by the revision control system.

VCS SPECIFIC FEATURES

tla / baz
If the commit message contains more than 72 characters, a summary will be created containing as many full words from the message as will fit within 72 characters, followed by an ellipsis.

Each of the features described below is applicable only if the commit message has been automatically determined from the changelog.

git If only a single change is detected in the changelog, **debcommit** will unfold it to a single line and behave as if **−−strip−message** was used.

Otherwise, the first change will be unfolded and stripped to form a summary line and a commit message formed using the summary line followed by a blank line and the changes as extracted from the changelog. **debcommit** will then spawn an editor so that the message may be fine-tuned before committing.

hg / darcs
The first change detected in the changelog will be unfolded to form a single line summary. If multiple changes were detected then an editor will be spawned to allow the message to be fine-tuned.

bzr If the changelog entry used for the commit message closes any bugs then **−−fixes** options to "bzr commit" will be generated to associate the revision and the bugs.

LICENSE

This code is copyright by Joey Hess <joeyh@debian.org>, all rights reserved. This program comes with ABSOLUTELY NO WARRANTY. You are free to redistribute this code under the terms of the GNU General Public License, version 2 or later.

AUTHOR

Joey Hess <joeyh@debian.org>

SEE ALSO

debchange(1), **svnpath**(1)