NAME

devscripts − scripts to ease the lives of Debian developers

DESCRIPTION

The **devscripts** package provides a collection of scripts which may be of use to Debian developers and others wishing to build Debian packages. For a summary of the available scripts, please see the file */usr/share/doc/devscripts/README.gz*, and for full details, please see the individual manpages. They are contributed by multiple developers; for details of the authors, please see the code or manpages.

Also, the directory */usr/share/doc/devscripts/examples* contains examples of **procmail** and **exim** scripts for sorting mail arriving to Debian mailing lists.

ENVIRONMENT

Several scripts of the devscripts suite use the following environment variables. Check the man pages of individual scripts for more details on how the variables are used.

DEBEMAIL
: Email of the person acting on a given Debian package via devscripts.

DEBFULLNAME
: Full name (first + family) of the person acting on a given Debian package via devscripts.

SCRIPTS

Here is the complete list of available devscripts. See their man pages for additional documentation.

annotate-output(1)
: run a command and prepend time and stream (O for stdout, E for stderr) for every line of output.

archpath(1)
: Prints arch (tla/Bazaar) package names. Also supports calculating the package names for other branches. [tla | bazaar]

bts(1)
: A command-line tool for accessing the Debian Bug Tracking System, both to send mails to control@bts.debian.org and to access the web pages and SOAP interface of the BTS. [www-browser, libauthen-sasl-perl, libnet-smtps-perl, libsoap-lite-perl, liburi-perl, libwww-perl, bsd-mailx | mailx]

build-rdeps(1)
: Searches for all packages that build-depend on a given package. [dctrl-tools, dose-extra, libdpkg-perl]

chdist(1)
: tool to easily play with several distributions. [dctrl-tools]

checkbashisms(1)
: check whether a /bin/sh script contains any common bash-specific constructs.

cowpoke(1)
: upload a Debian source package to a cowbuilder host and build it, optionally also signing and uploading the result to an incoming queue. [ssh-client]

cvs-debi, cvs-debc(1)
: wrappers around debi and debc respectively (see below) which allow them to be called from the CVS working directory. [cvs-buildpackage]

cvs-debrelease(1)
: wrapper around debrelease which allows it to be called from the CVS working directory. [cvs-buildpackage, dupload | dput, ssh-client]

cvs-debuild(1)
: A wrapper for cvs-buildpackage to use debuild as its package building program. [cvs-buildpackage, fakeroot, lintian, gnupg |gnupg2]

dcmd(1)
: run a given command replacing the name of a .changes or .dsc file with each of the files referenced therein. *

dcontrol(1)
: remotely query package and source control files for all Debian distributions. [liburl-perl, libwww-perl]

dd-list(1)
: given a list of packages, pretty-print it ordered by maintainer. *

debc(1) List contents of current package. Do this after a successful "debuild" to see if the package looks all right.

debchange (abbreviation dch)(1)
: Modifies debian/changelog and manages version numbers for you. It will either increment the version number or add an entry for the current version, depending upon the options given to it. [libdistro-info-perl, libsoap-lite-perl]*

debcheckout(1)
: checkout the development repository of a Debian package. *

debclean(1)
: Clean a Debian source tree. Debclean will clean all Debian source trees below the current directory, and if requested, also remove all files that were generated from these source trees (that is .deb, .dsc and .changes files). It will keep the .diffs and original files, though, so that the binaries and other files can be rebuilt if necessary. [fakeroot]*

debcommit(1)
: Commits changes to cvs, darcs, svn, svk, tla, bzr, git, or hg, using new entries in debian/changelog as the commit message. Also supports tagging Debian package releases. [cvs | darcs | subversion | svk | tla | bzr | git-core | mercurial, libtimedate-perl]

debdiff(1)
: A program which examines two .deb files or two .changes files and reports on any difference found in their file lists. Useful for ensuring that no files were inadvertently lost between versions. Can also examine two .dsc files and report on the changes between source versions. For a deeper comparison one can use the diffoscope package. [wdiff, patchutils]*

debdiff-apply(1)
: Apply unified diffs of two Debian source packages, such as those generated by debdiff, to a target Debian source package. Any changes to debian/changelog are dealt with specially, to avoid the conflicts that changelog diffs typically produce when applied naively. May be used to check that old patches still apply to newer versions of those packages. [python3-debian, python3-unidiff, quilt]

debi(1) Installs the current package by using the setuid root debpkg script described below. It assumes that the current package has just been built (for example by debuild), and the .deb lives in the parent directory, and will effectively run dpkg -i on the .deb. The ability to install the package with a very short command is very useful when troubleshooting packages.

debpkg(1)
: A wrapper for dpkg used by debi to allow convenient testing of packages. For debpkg to work, it needs to be made setuid root, and this needs to be performed by the sysadmin -- it is not installed as setuid root by default. (Note that being able to run a setuid root debpkg is effectively the same as having root access to the system, so this should be done with caution.) Having debpkg as a wrapper for dpkg can be a Good Thing (TM), as it decreases the potential for damage by accidental wrong use of commands in superuser mode (e.g., an inadvertent rm -rf * in the wrong directory is disastrous as many can attest to).

debrelease(1)
: A wrapper around dupload or dput which figures out which version to upload, and then calls dupload or dput to actually perform the upload. [dupload | dput, ssh-client]

debrepro(1)
: A script that tests reproducibility of Debian packages. It will build a given source directory twice, with a set of variation between the first and second build, and compare the binary packages produced. If diffoscope is installed, it is used to compare non-matching binaries. If disorderfs is installed, it is used during the build to inject non-determinism in filesystem listing operations. [faketime, diffoscope, disorderfs]

debrsign(1)
: This transfers a .changes/.dsc pair to a remote machine for signing, and runs debsign on the remote machine over an SSH connection. [gnupg | gnupg2, debian-keyring, ssh-client]

debsign(1)
: Use GNU Privacy Guard to sign the changes (and possibly dsc) files created by running dpkg-buildpackage with no-sign options. Useful if you are building a package on a remote machine and wish to sign it on a local one. This script is capable of automatically downloading the .changes and .dsc files from a remote machine. [gnupg |gnupg2, debian-keyring, ssh-client]*

debsnap(1)
: grab packages from https://snapshot.debian.org [libwww-perl, libjson-perl]

debuild(1)
: A wrapper for building a package (i.e., dpkg-buildpackage) to avoid problems with insufficient permissions and wrong paths etc. Debuild will set up the proper environment for building a package. Debuild will use the fakeroot program to build the package by default, but can be instructed to use any other gain-root command, or can even be installed setuid root. Debuild can also be used to run various of the debian/rules operations with the same root-gaining procedure. Debuild will also run lintian to check that the package does not have any major policy violations. [fakeroot, lintian, gnupg | gnupg2]*

deb-reversion(1)
: increases a binary package version number and repacks the package, useful for porters and the like.

dep3changelog(1)
: generate a changelog entry from a DEP3-style patch header.

desktop2menu(1)
: given a freedesktop.org desktop file, generate a skeleton for a menu file. [libfile-desktopentry-perl]

dget(1) Downloads Debian source and binary packages. Point at a .changes or .dsc to download all references files. Specify a package name to download it from the configured apt repository. [wget | curl]

diff2patches(1)
: extracts patches from a .diff.gz file placing them under debian/ or, if present, debian/patches. [patchutils]

dpkg-depcheck, dpkg-genbuilddeps(1)
: Runs a specified command (such as debian/rules build) or dpkg-buildpackage, respectively, to determine the packages used during the build process. This information can be helpful when trying to determine the packages needed in the Build-Depends etc. lines in the debian/control file. [build-essential, strace]

dscextract(1)
: extract a single file from a Debian source package. [patchutils]

dscverify(1)
: check the signature and MD5 sums of a dsc file against the most current Debian keyring on your system. [gnupg | gnupg2, debian-keyring]

edit-patch(1)
: add/edit a patch for a source package and commit the changes. [quilt | dpatch | cdbs]

getbuildlog(1)
: download package build logs from Debian auto-builders. [wget]

git-deborig(1)
: try to produce Debian orig.tar using git-archive(1). [libdpkg-perl, libgit-wrapper-perl, liblist-compare-perl, libstring-shellquote-perl, libtry-tiny-perl]

grep-excuses(1)
: grep britney's excuses to find out what is happening to your packages. [libdbd-pg-perl, libterm-size-perl, libyaml-syck-perl, wget, w3m]

hardening-check(1)
: report the hardening characteristics of a set of binaries.

list-unreleased(1)
: searches for packages marked UNRELEASED in their changelog.

ltnu (Long Time No Upload)(1)
: List all uploads of packages by the given uploader or maintainer and display them ordered by the last upload of that package, oldest uploads first.

manpage-alert(1)
: locate binaries without corresponding manpages. [man-db]

mass-bug(1)
: mass-file bug reports. [bsd-mailx | mailx]

mergechanges(1)
: merge .changes files from the same release but built on different architectures.

mk-build-deps(1)
: Given a package name and/or control file, generate a binary package which may be installed to satisfy the build-dependencies of the given package. [equivs]

mk-origtargz(1)
: Rename upstream tarball, optionally changing the compression and removing unwanted files. [libfile-which-perl, unzip, xz-utils, file]

namecheck(1)
: Check project names are not already taken.

nmudiff(1)
: prepare a diff of this version (presumably an NMU against the previously released version (as per the changelog) and submit the diff to the BTS. [patchutils, mutt]

origtargz(1)
: fetch the orig tarball of a Debian package from various sources, and unpack it.

plotchangelog(1)
: display information from a changelog graphically using gnuplot. [libtimedate-perl, gnuplot]

pts-subscribe(1)
: subscribe to the PTS (Package Tracking System) for a limited period of time. [bsd-mailx | mailx, at]

rc-alert(1)
: list installed packages which have release-critical bugs. [wget | curl]

rmadison(1)
: remotely query the Debian archive database about packages. [liburi-perl, wget | curl]

sadt(1) run DEP-8 tests. [python3-debian]

salsa(1)
: manipulates salsa.debian.org repositories and users [libgitlab-api-v4-perl]

suspicious-source(1)
: output a list of files which are not common source files. [python3-magic]

svnpath(1)
: Prints the path to the Subversion repository of a Subversion checkout. Also supports calculating the paths for branches and tags in a repository independent fashion. Used by debcommit to generate svn tags. [subversion]

tagpending(1)
: runs from a Debian source tree and tags bugs that are to be closed in the latest changelog as pending. [libsoap-lite-perl]

transition-check(1)
: Check a list of source packages for involvement in transitions for which uploads to unstable are currently blocked. [libwww-perl, libyaml-syck-perl]

uscan(1)
: Automatically scan for and download upstream updates. Uscan can also call a program such as uupdate to attempt to update the Debianised version based on the new update. Whilst uscan could be used to release the updated version automatically, it is probably better not to without testing it first. Uscan can also verify detached OpenPGP signatures if upstream's signing key is known. [file, gpgv | gpgv2, gnupg | gnupg2, libfile-which-perl, liblwp-protocol-https-perl, libmoo-perl, libwww-perl, unzip, xz-utils]*

uupdate(1)
: Update the package with an archive or patches from an upstream author. This will be of help if you have to update your package. It will try to apply the latest diffs to your package and tell you how successful it was. [patch]

what-patch(1)
: determine what patch system, if any, a source package is using. [patchutils]

whodepends(1)
: check which maintainers' packages depend on a package.

who-permits-upload(1)
: Retrieve information about Debian Maintainer access control lists. [gnupg | gnupg2, libencode-locale-perl, libwww-perl, debian-keyring]

who-uploads(1)
: determine the most recent uploaders of a package to the Debian archive. [gnupg | gnupg2, debian-keyring, debian-maintainers, wget]

wnpp-alert(1)
: list installed packages which are orphaned or up for adoption. [wget | curl]

wnpp-check(1)
: check whether there is an open request for packaging or intention to package bug for a package. [wget | curl]

wrap-and-sort(1)
: wrap long lines and sort items in packaging files. [python3-debian]