

NAME

debsign – sign a Debian *.changes* and *.dsc* file pair using GPG

SYNOPSIS

debsign [*options*] [*changes-file*][*dsc-file*][*commands-file* ...]

DESCRIPTION

debsign mimics the signing aspects (and bugs) of **dpkg-buildpackage**(1). It takes a *.dsc*, *.buildinfo*, or *.changes* file and signs it, and any child *.dsc*, *.buildinfo*, or *.changes* files directly or indirectly referenced by it, using the GNU Privacy Guard. It is careful to calculate the size and checksums of any newly signed child files and replace the original values in the parent file.

If no file is specified, *debian/changelog* is parsed to determine the name of the *.changes* file to look for in the parent directory.

If a *.commands* file is specified it is first validated (see the details at <ftp://ftp.upload.debian.org/pub/UploadQueue/README>), and the name specified in the Uploader field is used for signing.

This utility is useful if a developer must build a package on one machine where it is unsafe to sign it; they need then only transfer the small *.dsc*, *.buildinfo* and *.changes* files to a safe machine and then use the **debsign** program to sign them before transferring them back. This process can be automated in two ways. If the files to be signed live on the **remote** machine, the **-r** option may be used to copy them to the local machine and back again after signing. If the files live on the **local** machine, then they may be transferred to the remote machine for signing using **debrsign**(1). However note that it is probably safer to have your trusted signing machine use **debsign** to connect to the untrusted non-signing machine, rather than using **debrsign** to make the connection in the reverse direction.

This program can take default settings from the **devscripts** configuration files, as described below.

OPTIONS

-r [*username@*]*remotehost*

The files to be signed live on the specified remote host. In this case, a *.dsc*, *.buildinfo* or *.changes* file must be explicitly named, with an absolute directory or one relative to the remote home directory. **scp** will be used for the copying. The [*username@*]*remotehost:filename* syntax is permitted as an alternative. Wildcards (* etc.) are allowed.

-pprognam

When **debsign** needs to execute GPG to sign it will run *prognam* (searching the **PATH** if necessary), instead of **gpg**.

-mmaintainer

Specify the maintainer name to be used for signing. (See **dpkg-buildpackage**(1) for more information about the differences between **-m**, **-e** and **-k** when building packages; **debsign** makes no use of these distinctions except with respect to the precedence of the various options. These multiple options are provided so that the program will behave as expected when called by **debuild**(1).)

-emaintainer

Same as **-m** but takes precedence over it.

-kkeyid

Specify the key ID to be used for signing; overrides any **-m** and **-e** options.

-S

Look for a source-only *.changes* file instead of a binary-build *.changes* file.

-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.

--multi

Multiarch *.changes* mode: This signifies that **debsign** should use the most recent file with the name pattern *package_version_*.changes* as the *.changes* file, allowing for the *.changes* files produced by **dpkg-cross**.

--re-sign, --no-re-sign

Recreate signature, respectively use the existing signature, if the file has been signed already. If neither option is given and an already signed file is found the user is asked if he or she likes to use the current signature.

--debs-dir DIR

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

--no-conf, --noconf

Do not read any configuration files. This can only be used as the first option given on the command-line.

--help, -h

Display a help message and exit successfully.

--version

Display version and copyright information and exit successfully.

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:

DEBSIGN_PROGRAM

Setting this is equivalent to giving a **-p** option.

DEBSIGN_MAINT

This is the **-m** option.

DEBSIGN_KEYID

And this is the **-k** option.

DEBSIGN_ALWAYS_RESIGN

Always re-sign files even if they are already signed, without prompting.

DEBRELEASE_DEBS_DIR

This specifies the directory in which to look for the files to be signed, 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.

SEE ALSO

debsign(1), **debuild(1)**, **dpkg-architecture(1)**, **dpkg-buildpackage(1)**, **gpg(1)**, **gpg2(1)**, **md5sum(1)**, **sha1sum(1)**, **sha256sum(1)**, **scp(1)**, **devscripts.conf(5)**

AUTHOR

This program was written by Julian Gilbey <jdg@debian.org> and is copyright under the GPL, version 2 or later.