

**NAME**

cowpoke – Build a Debian source package in a remote cowbuilder instance

**SYNOPSIS**

**cowpoke** [*options*] *packagename.dsc*

**DESCRIPTION**

Uploads a Debian source package to a **cowbuilder** host and builds it, optionally also signing and uploading the result to an incoming queue.

**OPTIONS**

The following options are available:

**--arch=architecture**

Specify the Debian architecture(s) to build for. A space separated list of architectures may be used to build for all of them in a single pass. Valid arch names are those returned by **dpkg-architecture(1)** for **DEB\_BUILD\_ARCH**.

**--dist=distribution**

Specify the Debian distribution(s) to build for. A space separated list of distributions may be used to build for all of them in a single pass. Either codenames (such as **sid**, or **squeeze**) or distribution names (such as **unstable**, or **experimental**) may be used, but you should usually stick to using one or the other consistently as this name may be used in file paths and to locate old packages for comparison reporting.

It is now also possible to use locally defined names with this option, when used in conjunction with the **BASE\_DIST** option in a configuration file. This permits the maintenance and use of specially configured build chroots, which can source package dependencies from the backports archives or a local repository, or have other unusual configuration options set, without polluting the chroots you use for clean package builds intended for upload to the main repositories. See the description of **BASE\_DIST** below.

**--buildd=host**

Specify the remote host to build on.

**--buildd-user=name**

Specify the remote user to build as.

**--create**

Create the remote **cowbuilder** root if it does not already exist. If this option is not passed it is an error for the specified **--dist** or **--arch** to not have an existing **cowbuilder** root in the expected location.

The **--buildd-user** must have permission to create the **RESULT\_DIR** on the build host, or an admin with the necessary permission must first create it and give that user (or some group they are in) write access to it, for this option to succeed.

**--return=[path]**

Copy results of the build to *path*. If *path* is not specified, then return them to the current directory. The given *path* must exist, it will not be created.

**--no-return**

Do not copy results of the build to **RETURN\_DIR** (overriding a path set for it in the configuration files).

**--dpkg-opts=*opt1 opt2 ...***

Specify additional options to be passed to **dpkg-buildpackage**(1). Multiple options are delimited with spaces. This will override any options specified in **DEBBUILD\_OPTS** in the build host's *pbuilder*rc.

**--create-opts=*cowbuilder option***

Specify additional arguments to be passed verbatim to **cowbuilder** when a chroot is first created (using the **--create** option above). If multiple arguments need to be passed, this option should be specified separately for each of them.

E.g., **--create-opts "--othermirror" --create-opts "deb http:// ..."**

This option will override any **CREATE\_OPTS** specified for a chroot in the cowpoke configuration files.

**--update-opts=*cowbuilder option***

Specify additional arguments to be passed verbatim to **cowbuilder** if the base of the chroot is updated. If multiple arguments need to be passed, this option should be specified separately for each of them.

This option will override any **UPDATE\_OPTS** specified for a chroot in the cowpoke configuration files.

**--build-opts=*cowbuilder option***

Specify additional arguments to be passed verbatim to **cowbuilder** when a package build is performed. If multiple arguments need to be passed, this option should be specified separately for each of them.

This option will override any **BUILD\_OPTS** specified for a chroot in the cowpoke configuration files.

**--sign=*keyid***

Specify the key to sign packages with. This will override any **SIGN\_KEYID** specified for a chroot in the cowpoke configuration files.

**--upload=*queue***

Specify the dput queue to upload signed packages to. This will override any **UPLOAD\_QUEUE** specified for a chroot in the cowpoke configuration files.

**--help** Display a brief summary of the available options and current configuration.

**--version**

Display the current version information.

## CONFIGURATION OPTIONS

When **cowpoke** is run the following configuration options are read from global, per-user, and per-project configuration files if present. File paths may be absolute or relative, the latter being relative to the **BUILDD\_USER**'s home directory. Since the paths are typically quoted when used, tilde expansion will **not** be performed on them.

### Global defaults

These apply to every *arch* and *dist* in a single cowpoke invocation.

#### **BUILDD\_HOST**

The network address or fqdn of the build machine where **cowbuilder** is configured. This may be overridden by the **--buldd** command line option.

#### **BUILDD\_USER**

The unprivileged user name for operations on the build machine. This defaults to the local name of the user executing **cowpoke** (or to a username that is specified in your SSH configuration for **BUILDD\_HOST**), and may be overridden by the **--buldd-user** command line option.

#### **BUILDD\_ARCH**

The Debian architecture(s) to build for. This must match the **DEB\_BUILD\_ARCH** of the build chroot being used. It defaults to the local machine architecture where **cowpoke** is executed, and may be overridden by the **--arch** command line option. A (quoted) space separated list of architectures may be used here to build for all of them in a single pass.

#### **BUILDD\_DIST**

The Debian distribution(s) to build for. A (quoted) space separated list of distributions may be used to build for all of them in a single pass. This may be overridden by the **--dist** command line option.

#### **INCOMING\_DIR**

The directory path on the build machine where the source package will initially be placed. This must be writable by the **BUILDD\_USER**.

#### **PBUILDER\_BASE**

The filesystem root for all pbuilder CoW and result files. *Arch* and *dist* specific subdirectories will normally be created under this. The apt cache and temporary build directory will also be located under this path.

#### **SIGN\_KEYID**

If this option is set, it is expected to contain the gpg key ID to pass to **debsign**(1) if the packages are to be remotely signed. You will be prompted to confirm whether you wish to sign the packages after all builds are complete. If this option is unset or an empty string, no attempt to sign packages will be made. It may be overridden on an *arch* and *dist* specific basis using the *arch\_dist\_SIGN\_KEYID* option described below, or per-invocation with the **--sign** command line option.

#### **UPLOAD\_QUEUE**

If this option is set, it is expected to contain a 'host' specification for **dput**(1) which will be used to upload them after they are signed. You will be prompted to confirm whether you wish to upload the packages after they are signed. If this option is unset or an empty string, no attempt to upload packages will be made. If **SIGN\_KEYID** is not set, this option will be ignored entirely. It may be overridden on an *arch* and *dist* specific basis using the *arch\_dist\_UPLOAD\_QUEUE* option described below, or per-invocation with the **--upload** command line option.

**BUILDD\_ROOTCMD**

The command to use to gain root privileges on the remote build machine. If unset the default is **sudo(8)**. This is only required to invoke **cowbuilder** and allow it to enter its chroot, so you may restrict this user to only being able to run that command with escalated privileges. Something like this in sudoers will enable invoking **cowbuilder** without an additional password entry required:

```
youruser ALL = NOPASSWD: /usr/sbin/cowbuilder
```

Alternatively you could use SSH with a forwarded key, or whatever other mechanism suits your local access policy. Using **su -c** isn't really suitable here due to its quoting requirements being somewhat different to the rest.

**DEBOOTSTRAP**

The utility to use when creating a new build root. Alternatives are **debootstrap** or **cdebootstrap**.

**RETURN\_DIR**

If set, package files resulting from the build will be copied to the path (local or remote) that this is set to, after the build completes. The path must exist, it will not be created. This option is unset by default and can be overridden with **--return** or **--no-return**.

**Arch and dist specific options**

These are variables of the form: **\$arch\_\$dist\_VAR** which apply only for a particular target arch/dist build.

*arch\_dist\_RESULT\_DIR*

The directory path on the build machine where the resulting packages (source and binary) will be found, and where older versions of the package that were built previously may be found. If any such older packages exist, **debdiff** will be used to compare the new package with the previous version after the build is complete, and the result will be included in the build log. Files in it must be readable by the **BUILDD\_USER** for sanity checking with **lintian(1)** and **debdiff(1)**, and for upload with **dput(1)**. If this option is not specified for some arch and dist combination then it will default to **\$PBUILDER\_BASE/\$arch/\$dist/result**

*arch\_dist\_BASE\_PATH*

The directory where the CoW master files are to be found (or created if the **--create** command line option was passed). If this option is not specified for some arch or dist then it will default to **\$PBUILDER\_BASE/\$arch/\$dist/base.cow**

*arch\_dist\_BASE\_DIST*

The code name to pass as the **--distribution** option for cowbuilder instead of *dist*. This is necessary when *dist* is a locally significant name assigned to some specially configured build chroot, such as 'wheezy\_backports', and not the formal suite name of a distro release known to debootstrap. This option cannot be overridden on the command line, since it would rarely, if ever, make any sense to change it for individual invocations of **cowpoke**. If this option is not specified for an arch and dist combination then it will default to *dist*.

*arch\_dist\_CREATE\_OPTS*

A bash array containing additional options to pass verbatim to **cowbuilder** when this chroot is created for the first time (using the **--create** option). This is useful when options like **--othermirror** are wanted to create specialised chroot configurations such as 'wheezy\_backports'. By default this is unset. All values set in it will be overridden if the **--create-opts** option is passed on the command line.

Each element in this array corresponds to a single argument (in the ARGV sense) that will be passed to **cowbuilder**. This ensures that arguments which may contain whitespace or have strange quoting requirements or other special characters will not be mangled before they get to **cowbuilder**.

Bash arrays are initialised using the following form:

```
OPTS=( "arg1" "arg 2" "--option" "value" "--opt=val" "etc. etc." )
```

#### *arch\_dist\_UPDATE\_OPTS*

A bash array containing additional options to pass verbatim to **cowbuilder** each time the base of this chroot is updated. It behaves similarly to the **CREATE\_OPTS** option above, except for acting when the chroot is updated.

#### *arch\_dist\_BUILD\_OPTS*

A bash array containing additional options to pass verbatim to **cowbuilder** each time a package build is performed in this chroot. This is useful when you want to use some option like **--twice** which **cowpoke** does not directly need to care about. It otherwise behaves similarly to **UPDATE\_OPTS** above except that it acts during the build phase of **cowbuilder**.

#### *arch\_dist\_SIGN\_KEYID*

An optional arch and dist specific override for the global **SIGN\_KEYID** option.

#### *arch\_dist\_UPLOAD\_QUEUE*

An optional arch and dist specific override for the global **UPLOAD\_QUEUE** option.

## CONFIGURATION FILES

*/etc/cowpoke.conf*

Global configuration options. Will override hardcoded defaults.

*~/cowpoke*

Per-user configuration options. Will override any global configuration.

*.cowpoke*

Per-project configuration options. Will override any per-user or global configuration if **cowpoke** is called from the directory where they exist.

If the environment variable **COWPOKE\_CONF** is set, it specifies an additional configuration file which will override all of those above. Options specified explicitly on the command line override all configuration files.

## COWBUILDER CONFIGURATION

There is nothing particularly special required to configure a **cowbuilder** instance for use with **cowpoke**. Simply create them in the flavour you require with '**cowbuilder --create**' according to the **cowbuilder** documentation, then configure **cowpoke** with the user, arch, and path information required to access it, on the machines you wish to invoke it from (or alternatively configure **cowpoke** with the path, arch and distribution information and pass the **--create** option to it on the first invocation). The build host running **cowbuilder** does not require **cowpoke** installed locally.

The build machine should have the **lintian** and **devscripts** packages installed for post-build sanity checking. Upon completion, the build log and the results of automated checks will be recorded in the

**INCOMING\_DIR**. If you wish to upload signed packages the build machine will also need **dput(1)** installed and configured to use the *'host'* alias specified by **UPLOAD\_QUEUE**. If **rsync(1)** is available on both the local and build machine, then it will be used to transfer the source package (this may save on some transfers of the *orig.tar.\** when building subsequent Debian revisions).

The user executing **cowpoke** must have SSH access to the build machine as the **BUILDD\_USER**. That user must be able to invoke **cowbuilder** as root by using the **BUILDD\_ROOTCMD**. Signing keys are not required to be installed on the build machine (and will be ignored there if they are). If the package is signed, keys will be expected on the machine that executes **cowpoke**.

When **cowpoke** is invoked, it will first attempt to update the **cowbuilder** image if that has not already been done on the same day. This is checked by the presence or absence of a *cowbuilder-\$arch-\$dist-update-log-\$date* file in the **INCOMING\_DIR**. You may move, remove, or touch this file if you wish the image to be updated more or less often than that. Its contents log the output of **cowbuilder** during the update (or creation) of the build root.

## NOTES

Since **cowbuilder** creates a chroot, and to do that you need root, **cowpoke** also requires some degree of root access. So all the horrible things that can go wrong with that may well one day rain down upon you. **cowbuilder** has been known to accidentally wipe out bind-mounted filesystems outside the chroot, and worse than that can easily happen. So be careful, keep good backups of things you don't want to lose on your build machine, and use **cowpoke** to keep all that on a machine that isn't your bleeding edge dev box with your last few hours of uncommitted work.

## SEE ALSO

**cowbuilder(1)**, **pbuilder(1)**, **ssh-agent(1)**, **sudoers(5)**

## AUTHOR

**cowpoke** was written by Ron <ron@debian.org>.