NAME

uupdate − upgrade a source code package from an upstream revision

SYNOPSIS

uupdate [*options*] *new_upstream_archive* [*version*]
uupdate [*options*] **−−find|−f**
uupdate [*options*] **−−patch|−p** *patch_file*

DESCRIPTION

uupdate modifies an existing Debian source code archive to reflect an upstream update supplied as a patch or from a wholly new source code archive. The utility needs to be invoked from the top directory of the old source code directory, and if a relative name is given for the new archive or patch file, it will be looked for first relative to the execution directory and then relative to the parent of the source tree. (For example, if the changelog file is */usr/local/src/foo/foo-1.1/debian/changelog*, then the archive or patch file will be looked for relative to */usr/local/src/foo*.) Note that the patch file or archive cannot be within the source tree itself. The full details of what the code does are given below.

Currently supported source code file types are *.tar.gz*, *.tar.bz2*, *.tar.Z*, *.tgz*, *.tar*, *.tar.lzma*, *.tar.xz*, *.7z* and *.zip* archives. Also supported are already unpacked source code archives; simply give the path of the source code directory. Supported patch file types are **gzip**-compressed, **bzip2**-compressed, **lzma**-compressed, **xz**-compressed and uncompressed patch files. The file types are identified by the file names, so they must use the standard suffixes.

Usually **uupdate** will be able to deduce the version number from the source archive name (as long as it only contains digits and periods). If that fails, you need to specify the version number explicitly (without the Debian release number which will always be initially "1", or "0ubuntu1" on Ubuntu-detected systems). This can be done with an initial **−−upstream-version** or **−v** option, or in the case of an archive, with a version number after the filename. (The reason for the latter is so that **uupdate** can be called directly from **uscan**.)

Since **uupdate** uses **debuild** to clean the current archive before trying to apply a patch file, it accepts a **−−rootcmd** or **−r** option allowing the user to specify a gain-root command to be used. The default is to use **fakeroot**.

If an archive is being built, the pristine upstream source should be used to create the *.orig.tar.gz* file wherever possible. This means that MD5 sums or other similar methods can be used to easily compare the upstream source to Debian's copy of the upstream version. This is the default behaviour, and can be switched off using the **−−no−pristine** option below.

OPTIONS

This is a summary of what was explained above.

−−no-conf, **−−noconf**
: Do not read any configuration files. This can only be used as the first option given on the command-line.

−−upstream−version *version*, **−v** *version*
: Specify the version number of the upstream package explicitly.

−−force−bad−version, −b
: Force a version number to be less than the current one (e.g., when backporting).

−−rootcmd *gain-root-command*, **−r** *gain-root-command*
: Specify the command to be used to become root to build the package and is passed onto **debuild**(1) if it is specified.

−−pristine, **−u**
: Treat the source as pristine upstream source and symlink to it from *<package>_<version>.orig.tar.gz* whenever possible. This option has no meaning for patches. This is the default behaviour.

−−no−pristine
Do not attempt to make a *<package>_<version>.orig.tar.gz* symlink.

−−symlink, **−s**
Simply create a symlink when moving a new upstream *.tar.gz* archive to the new *<package>_<version>.orig.tar.gz* location. This is the default behaviour.

−−no−symlink
Copy the upstream *.tar.gz* to the new location instead of making a symlink, if *<package>_<version>.orig.tar.gz* is missing. Otherwise, do nothing.

−−find, −f
Find all upstream tarballs in *../* which match *<pkg>_<version>.orig.tar.{gz|bz2|lzma|xz}* or *<pkg>_<version>.orig−<component>.tar.{gz|bz2|lzma|xz}* ; **−−upstream−version** required; pristine source required; not valid for **−−patch**; This option uses **dpkg−source** as the backend to enable support for the multiple upstream tarballs and to resolve minor bugs reported previously. The use of this option is highly recommended.

−−verbose
Give verbose output.

−−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:

UUPDATE_PRISTINE
If this is set to *no*, then it is the same as the **−−no−pristine** command line parameter being used.

UUPDATE_SYMLINK_ORIG
If this is set to *no*, then it is the same as the **−−no−symlink** command line parameter being used.

UUPDATE_ROOTCMD
This is equivalent to the **−−rootcmd** option.

ACTIONS TAKEN ON AN ARCHIVE

Figure out new version number
Unless an explicit version number is provided, the archive name is analyzed for a sequence of digits separated by dots. If something like that is found, it is taken to be the new upstream version number. If not, processing is aborted.

Create the .orig.tar.gz archive
If the **−−pristine** or **−u** option is specified and the upstream archive is a *.tar.gz* or *.tgz* archive, then this will be copied directly to *<package>_<version>.orig.tar.gz*.

Unpacking
The archive is unpacked and placed in a directory with the correct name according to Debian policy: package-upstream_version.orig. Processing is aborted if this directory already exists.

Patching
The *.diffs.gz* from the current version are applied to the unpackaged archive. A non-zero exit status and warning message will occur if the patches did not apply cleanly or if no patch file was found. Also, the list of rejected patches will be shown. The file *debian/rules* is made executable and all of the *.orig* files created by **patch** are deleted.

Changelog update

A changelog entry with the new version number is generated with the text "New upstream release".

When used on Ubuntu systems, **dpkg-vendor** detection is used to set the Debian revision to "0ubuntu1". You may change *debian/changelog* manually afterwards.

ACTIONS TAKEN ON A PATCH FILE

Figure out new version number

Unless an explicit version number is provided, the patch file name is analyzed for a sequence of digits separated by dots. If something like that is found, it is taken to be the new upstream version number. If not, processing is aborted.

Clean the current source tree

The command **debuild clean** is executed within the current Debian source archive to clean it. If a **−r** option is given to **uupdate**, it is passed on to **debuild**.

Patching

The current source archive (*.orig.tar.gz*) is unpacked and the patch applied to the original sources. If this is successful, then the *.orig* directory is renamed to reflect the new version number and the current Debian source directory is copied to a directory with the new version number, otherwise processing is aborted. The patch is then applied to the new copy of the Debian source directory. The file *debian/rules* is made executable and all of the *.orig* files created by **patch** are deleted. If there was a problem with the patching, a warning is issued and the program will eventually exit with non-zero exit status.

Changelog update

A changelog entry with the new version number is generated with the text "New upstream release".

When used on Ubuntu systems, **dpkg-vendor** detection is used to set the Debian revision to "0ubuntu1". You may change *debian/changelog* manually afterwards.

SEE ALSO

debuild(1), **fakeroot**(1), **patch**(1), **devscripts.conf**(5)

The Debian Policy Manual

AUTHOR

The original version of **uupdate** was written by Christoph Lameter <clameter@debian.org>. Several changes and improvements have been made by Julian Gilbey <jdg@debian.org>.