NAME

pristine−tar − regenerate pristine tarballs

SYNOPSIS

pristine-tar [OPTIONS] gendelta *tarball delta*

pristine-tar [OPTIONS] gentar *delta tarball*

pristine-tar [OPTIONS] commit *tarball* [*upstream*]

pristine-tar [OPTIONS] checkout *tarball*

pristine-tar [OPTIONS] list

pristine-tar [OPTIONS] verify *tarball*

DESCRIPTION

pristine-tar can regenerate an exact copy of a pristine upstream tarball using only a small binary *delta* file and the contents of the tarball, which are typically kept in an *upstream* branch in version control.

The *delta* file is designed to be checked into version control along-side the *upstream* branch, thus allowing Debian packages to be built entirely using sources in version control, without the need to keep copies of upstream tarballs.

pristine-tar supports compressed tarballs, calling out to **pristine−gz**(1), **pristine−bz2**(1), and **pristine−xz**(1) to produce the pristine gzip, bzip2, and xz files.

COMMANDS

pristine-tar gendelta *tarball delta*

This takes the specified upstream *tarball*, and generates a small binary delta file that can later be used by pristine-tar gentar to recreate the tarball.

If the delta filename is ''−'', it is written to standard output.

pristine-tar gentar *delta tarball*

This takes the specified *delta* file, and the files in the current directory, which must have identical content to those in the upstream tarball, and uses these to regenerate the pristine upstream *tarball*.

If the delta filename is ''−'', it is read from standard input.

pristine-tar commit *tarball* **[***upstream***]**

pristine-tar commit generates a pristine-tar delta file for the specified *tarball*, and commits it to version control. The **pristine-tar checkout** command can later be used to recreate the original tarball based only on the information stored in version control.

The *upstream* parameter specifies the tag or branch that contains the same content that is present in the tarball. This defaults to ''refs/heads/upstream'', or if there's no such branch, any branch matching ''upstream''. The name of the tree it points to will be recorded for later use by **pristine-tar checkout**. Note that the content does not need to be 100% identical to the content of the tarball, but if it is not, additional space will be used in the delta file.

The delta files are stored in a branch named ''pristine-tar'', with filenames corresponding to the input tarball, with ''.delta'' appended. This branch is created or updated as needed to add each new delta.

If *tarball* already exists previously, it will only be overwritten if it does not match a hash of the tarball that has been committed to version control.

pristine-tar checkout *tarball*

This regenerates a copy of the specified *tarball* using information previously saved in version control by **pristine-tar commit**.

pristine-tar list

This lists tarballs that pristine-tar is able to checkout from version control.

pristine-tar verify *tarball*

Verifies whether an existing *tarball* matches the one that has been committed to version control.

OPTIONS

−v
−−verbose

Verbose mode, show each command that is run.

−d
−−debug

Debug mode.

−k
−−keep

Don’t clean up the temporary directory on exit.

−m message
−−message=message

Use this option to specify a custom commit message to pristine-tar commit.

Applies to the **commit** command.

−s signaturefile
−−signature−file=signaturefile

Use this option to optionally commit or checkout an upstream signature file for the tarball. Note that extraction of signatures is not performed by default.

Applies to the **commit** and **checkout** commands.

−r
−−recompress

Use this option to tell pristine-tar that it is OK to recompress the tarball if 1) the tarball can’t be regenerated or 2) the delta file produced from the original tarball is larger than a given threshold (see also *−B*/*−−recompress−threshold−bytes* and *−T*/*−−recompress−threshold−percent*).

Note that this modifies the original tarball on disk, and you probably shouldn’t use it if you are also storing a upstream GPG signature of the original tarball (see *−−signature−file*).

On the other hand, the actual contents of the tarball stored by pristine-tar **will** be identical to the content of the upstream original tarbal even if the compressed tarball itself won’t be bit-by-bit identical to the one released by upstream.

A copy of the original tarball will be saved with ‘‘.backup’’ added to its filename.

Applies to the **commit** and **gendelta** commands.

−B N
−−recompress−threshold−bytes=N

Recompress original tarball if the generated delta is larger then *N* bytes. Default: 524288000 (500 kB).

If this option is passed explicitly in the command line, it implies *−−recompress*.

Applies to the **commit** and **gendelta** commands.

−P N
−−recompress−threshold−percent=N

Recompress original tarball if the generated delta is larger than *N%* of the size of the original tarball. Default: 30.

If this option is passed explicitly in the command line, it implies *−−recompress*.

Applies to the **commit** and **gendelta** commands.

EXAMPLES

Suppose you maintain the hello package, in a git repository. You have just created a tarball of the release, *hello−1.0.tar.gz*, which you will upload to a "forge" site.

You want to ensure that, if the "forge" loses the tarball, you can always recreate exactly that same tarball. And you'd prefer not to keep copies of tarballs for every release, as that could use a lot of disk space when hello gets the background mp3s and user-contributed levels you are planning for version 2.0.

The solution is to use pristine-tar to commit a delta file that efficiently stores enough information to reproduce the tarball later.

```
cd hello
git tag -s 1.0
pristine-tar commit ../hello-1.0.tar.gz 1.0
```

Remember to tell git to push both the pristine-tar branch, and your tag:

```
git push --all --tags
```

Now it is a year later. The worst has come to pass; the "forge" lost all its data, you deleted the tarballs to make room for bug report emails, and you want to regenerate them. Happily, the git repository is still available.

```
git clone git://github.com/joeyh/hello.git
cd hello
pristine-tar checkout ../hello-1.0.tar.gz
```

LIMITATIONS

Only tarballs, gzipped tarballs, bzip2ed tarballs, and xzed tarballs are currently supported.

Currently only the git revision control system is supported by the "checkout" and "commit" commands. It's ok if the working copy is not clean or has uncommitted changes, or has changes staged in the index; none of that will be touched by "checkout" or "commit".

ENVIRONMENT

TMPDIR

Specifies a location to place temporary files, other than the default.

PRISTINE_TAR

Defines command line options to be assumed by pristine-tar. Any options passed explicitly on the command line will override those.

Options will be split on whitespaces, so if you want to pass an option that needs an argument, use the *−−opt=arg* syntax instead of *−−opt arg*.

PRISTINE_ALL_XDELTA

Defines the underlying binary delta tool to be used for new deltas. Supported values are "xdelta3" (default) and "xdelta".

Existing deltas will be handled with the original tool that was used to create them, regardless of the value of **`$PRISTINE_ALL_XDELTA`**.

AUTHOR

Joey Hess <joeyh@debian.org>

Licensed under the GPL, version 2 or above.