.

Tardy

Reference Manual

Peter Miller
pmiller@opensource.org.au

.

This document describes Tardy version 1.28
and was prepared 22 January 2018.

This document describing the Tardy program, and the Tardy program itself, are
Copyright © 1993, 1994, 1995, 1996, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2006, 2008, 2009, 2010, 2011, 2012, 2013 Peter Miller

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

NAME

tardy version 1.28 − a tar post-processor

Copyright © 1993, 1994, 1995, 1996, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2006, 2008, 2009, 2010, 2011, 2012, 2013 Peter Miller

The *tardy* program is distributed under the terms of the GNU General Public License. See the LICENSE section, below, for more details.

tardy *a.*, slow to act, behind time.

DESCRIPTION

The *tardy* program is a *tar*(1) post-processor. It may be used to manipulate the file headers in *tar*(5) archive files in various ways.

The reason the *tardy* program was written was because the author wanted to "spruce up" his tar files before posting them to the net, mostly to remove artefacts of the development environment, without introducing more.

The *tardy* program was designed to allow you to alter certain characteristics of files AFTER they have been included in the TAR file. Among them are:

- change file owner (by NUMBER or NAME)
- change file group (by NUMBER or NAME)
- add directory prefix (dot is a really useful prefix)
- change file protections (for example from 600 to 644)

Note that all of these affect ALL files in the archive.

BUILDING

Instructions on how to build and test the *tardy* program are to be found in the *BUILDING* file included in this distribution.

LICENSE

The *tardy* program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

The *tardy* program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

It should be in the *LICENSE* file included in this distribution.

AUTHOR

Peter Miller EMail: pmiller@opensource.org.au
/\/* WWW: http://miller.emu.id.au/pmiller/

RELEASE NOTES

Version 1.28 (2013-Jun-27)

• jari <jari.aalto@cante.net> discovered a false negative from test/00/t0075a.sh, causd by a difference in time zones.

• jari <jari.aalto@cante.net> discovered that three files were mysteriously absent from the tarball causing FTBFS.

• jari <jari.aalto@cante.net> discovered a problem in the download pages, causing the debian-less orig.tar.gz to be improperly linked.

Version 1.27 (2013-May-28)

- There is now a tarball on the web site with no debian/ directory, to make it easier for downstream packagers.
- There is now a debian/watch file, in main tarball.
- There is now a test tarballs > 2GB. This was fixed back in i1.25.D012 (12-Jun-2012). Debian: Closes: #659630

Version 1.26 (2012-Jul-09)

- The tardy(1) command now understands ZX compression for both reading and writing, symmetric with how it handles gzip comporession.
- A memory scribble deep within the string code has been fixed. It was related to how it split the hash table when growing the size.
- Tardy now uses 64-bit file sizes consistently.
- The tardy(1) command is now able to use bzip2 compression on input and output, symmetric with its gzip compression handling.
- Emanuel Haupt <ehaupt@FreeBSD.org> discovered an incorrect function prototype, that caused build problems on some 32-bit systems.
- Dagobert Michelsen <dam@opencsw.org> discovered that Solaris needs more #include <stdarg.h> lines.
- Dagobert Michelsen <dam@opencsw.org> discovered a build problem with the memmem replacement function.

Version 1.25 (2011-Nov-07)

- Dagobert Michelsen <dam@opencsw.org> discovered that several string functions were undeclared on Solaris, causing build problems; and that there was no replacement memmem() function for systems that didn't have one.

Version 1.24 (2011-Oct-03)

- The *tardy*(1) command is now more tolerant of *tar*(5) archives that contain regular files labeled LD_DIRECTORY instead of LD_NORMAL.

Version 1.23 (2011-Sep-09)

- Thomas <metaf4@users.askja.de> discovered that the V7 tar output was using an incorrect link flag when writing file headers for directories, causing *pax*(1) and *star*(1) to complain.
- Thomas <metaf4@users.askja.de> discovered that the BSD tar output was using an incorrect link flag when writing file headers for directories, causing *mc*(1) and *Xarchiver*(1) to complain.
- Dagobert Michelsen <dam@opencsw.org> discovered a build problem on systems without the *strverscmp*(3) function.

Version 1.22 (2011-May-14)

- Thomas <metaf4@users.askja.de> contributed a patch to detect reading from a terminal, or writing to a terminal. These are usually indications that the user has failed to supply a command line argument.
- Tardy is now able to read and write long file names in *ar*(1) archives.
- A bug has been fixed in the BSD ar(1) format archives, the file padding has been corrected to 2-byte boundaries.
- The *ar*(1) padding now uses newlines, to be compatible with GNU binutils.
- The ./configure script will now fail if libexplain has not been installed, rather than failing later in the build.

Version 1.21 (2011-Mar-21)

- Tardy is now able to automatically *gunzip*(1) input files *gzip*(1) output files.
- Tardy is now able to read and write several *ar*(1) file formats.
- Tardy is now able to read and write several *cpio*(1) file formats.
- There is a new tardy **−relative−paths** filter, it may be used to ensure that all file names in the archive are relative (do not start with a '/' slash character). This is tidier that the "**−prefix .**" alternative.
- Tardy now uses **libexplain** for improved error messages.
  http://libexplain.sourceforge.net/
- There is a new Tardy **−extract** option, that may be used to extract the contents of an archive, in any format that Tardy understands.
- Thomas <metaf4@users.askja.de> submitted a patch that added better output support for the USTAR archive format on.
- Jari Aalto <jari.aalto@cante.net> reported several spelling errors im the man page, these have been corrected.

Version 1.20 (2010-May-22)

- A build problem on FreeBSD has been fixed.
- A build problem on Debian has been fixed.

Version 1.19 (2010-Mar-10)

- The Makefile.in file has been fixed so the package now builds correctly. Additional support for building Debian packages has been added.

Version 1.18 (2009-Nov-29)

- Numerous build glitches and warnings have been fixed.
- There is a new tardy -exclude option that may be used to exclude (remove) files from the archive.

Version 1.17 (2008-Jun-23)

- A build problem with GCC 4.3 has been fixed [Gentoo Bug 226861].

Version 1.16 (2008-Apr-29)

- The tabs in the Makefile.in file have been fixed.

Version 1.15 (2008-Apr-14)

- The license has been updated to GNU GPL version 3.

Version 1.12 (2004-Aug-17)

- Tardy has a new home page, http://tardy.sourceforge.net/
- The tardy --remove-prefix option has been extended to accept a numeric argument, giving the count of directories to be removed.
- A bug has been fixed in the tardy --reomve-prefix option. If there is a directory named exactly for the prefix being removed, it is now renamed to "." rather than being left unaltered.
- The tar output has been altered to be consistent with GNU Tar 1.14 and later. As of GNU Tar 1.14, it warns if the block size on input is not what it expects. Tardy now uses the same default block size as GNU

Tar when writing tarballs, which silences the GNU Tar warning.

Version 1.11 (2002-Oct-20)

• A bug has been fixed with the processing of very long directory names.

Version 1.10 (2002-Oct-16)

• Some documentation errors have been corrected.

• Some build problems have been fixed.

Version 1.9 (27-Aug-2001)

• A number fo build problems have been fixed.

• Support has been added for long names.

Version 1.8 (24-Oct-2001)

• The generated *Makefile.in* file was missing some dependencies. This has been fixed.

Version 1.7 (23-Oct-2001)

• A number of files were missing from the tarball. These have been added.

Version 1.6 (23-Oct-2001)

• Build and test problems caused by the recent RH7 release have been fixed.

Version 1.5 (13-Sep-2000)

• Tardy has been entirely re-written in C++.

• Tardy now understands cpio formats in addition to tar formats.

• Tardy now comes with an RPM package spec.

Version 1.4 (1998-Jan-25)

Never had a public release.

Version 1.3

Never had a public release.

Version 1.2

• This version of tardy now uses a GNU autoconf generated *configure* script to automatically configure the tardy program.

NAME

How to build Tardy

Copyright © 1993, 1994, 1995, 1996, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2006, 2008, 2009, 2010, 2011, 2012, 2013 Peter Miller

The *tardy* program is distributed under the terms of the GNU General Public License. See the LICENSE section, below, for more details.

tardy *a.*, slow to act, behind time.

BEFORE YOU START

There are a few pieces of software you may want to fetch and install before you proceed with your installation of Tardy.

Boost Library
: You will need the C++ Boost Library. If you are using a package based system, you will need the libboost-devel package, or one named something very similar.
http://boost.org/

BZ2 library
: You will need the bzip2 library. If you are using a package based system, you will need the libbz2-devel package, or one named something very similar.

LZMA library
: You will need the lzma library, for .xz compression. If you are using a package based system, you will need the liblzma-devel package, or one named something very similar.

cpio
: The test suite uses the *cpio*(1) command to create CPIO archives for testing.

GNU Groff
: The documentation for the *Tardy* package was prepared using the GNU Groff package (version 1.14 or later). This distribution includes full documentation, which may be processed into PostScript or DVI files at install time − if GNU Groff has been installed.

GCC
: You may also want to consider fetching and installing the GNU C Compiler if you have not done so already. This is not essential. Tardy was developed using the GNU C++ compiler, and the GNU C++ libraries.

libexplain
: The *libexplain* project provides a library if system-call-specific sterror(3) replacements, for more informative error messages.
http://libexplain.sourceforge.net/

zlib
: zlib is a library implementing the deflate compression method (and inflate decompression method) found in *gzip*(1) and PKZIP.
http://zlib.net/

SITE CONFIGURATION

The **tardy** program is configured using the *configure* shell script included in this distribution.

The *configure* shell script attempts to guess correct values for various system-dependent variables used during compilation, and creates the *Makefile* and *libtardy/config.h* files. It also creates a shell script *config.status* that you can run in the future to recreate the current configuration.

Normally, you just *cd* to the directory containing *tardy*'s source code and type

```
% ./configure
...lots of output...
%
```

If you're using *csh* on an old version of System V, you might need to type

```
% sh configure
...lots of output...
%
```

instead to prevent *csh* from trying to execute *configure* itself.

Running *configure* takes a minute or two. While it is running, it prints some messages that tell what it is doing. If you don't want to see the messages, run *configure* using the --quiet option; for example,

```
% ./configure --quiet
%
```

By default, *configure* will arrange for the *make install* command to install the **tardy** program's files in */usr/local/bin* and */usr/local/man*. You can specify an installation prefix other than */usr/local* by giving *configure* the option `--prefix=`*PATH*.

You can specify separate installation prefixes for architecture-specific files and architecture-independent files. If you give *configure* the option `--exec-prefix=`*PATH* the **tardy** package will use *PATH* as the prefix for installing programs and libraries. Data files and documentation will still use the regular prefix. Normally, all files are installed using the same prefix.

configure ignores any other arguments that you give it.

On systems that require unusual options for compilation or linking that the *tardy* package's *configure* script does not know about, you can give *configure* initial values for variables by setting them in the environment. In Bourne-compatible shells, you can do that on the command line like this:

```
$ CC='gcc -traditional' LIBS=-lposix ./configure
...lots of output...
$
```

Here are the *make* variables that you might want to override with environment variables when running *configure*.

Variable: CC
: C compiler program. The default is *cc*.

Variable: INSTALL
: Program to use to install files. The default is *install* if you have it, *cp* otherwise.

Variable: LIBS
: Libraries to link with, in the form `-l`*foo* `-l`*bar*. The *configure* script will append to this, rather than replace it.

If you need to do unusual things to compile the package, the author encourages you to figure out how *configure* could check whether to do them, and mail diffs or instructions to the author so that they can be included in the next release.

BUILDING TARDY

All you should need to do is use the

```
% make
...lots of output...
%
```

command and wait. When this finishes you should see a directory called *bin* containing one file: *tardy*. The *tardy* program is a tar post-processor.

You can remove the program binaries and object files from the source directory by using the

```
% make clean
...lots of output...
%
```

command. To remove all of the above files, and also remove the *Makefile* and *libtardy/config.h* and *config.status* files, use the

```
% make distclean
...lots of output...
%
```

command.

The file *etc/configure.ac* is used to create *configure* by a GNU program called *autoconf*. You only need to know this if you want to regenerate *configure* using a newer version of *autoconf*.

TESTING TARDY

The *tardy* program comes with a test suite. To run this test suite, use the command

```
% make sure
...lots of output...
Passed All Tests
%
```

The tests take a about a minute each, with a few very fast, and a couple very slow, but it varies greatly depending on your CPU.

INSTALLING TARDY

The *tardy* program is installed under the */usr/local* tree by default. Use the `--prefix=`*PATH* option to *configure* if you want some other path.

All that is required to install the *tardy* program is to use the

```
% make install
...lots of output...
%
```

command. Control of the directories used may be found in the first few lines of the *Makefile* file if you want to bypass the *configure* script.

The above procedure assumes that the *soelim*(1) command is somewhere in the command search *PATH*. The *soelim*(1) command is available as part of the *GNU Roff* package.

The above procedure also assumes that the *$(prefix)/man/man1* and *$(prefix)/man/man5* directories already exist. If they do not, you will need to *mkdir* them manually.

PRINTED MANUALS

This distribution contains the sources to all of the documentation for *tardy*. The author used the GNU groff package and a postscript printer to prepare the documentation. If you do not have this software, you will need to substitute commands appropriate to your site.

To print copies of the *README*, and *BUILDING* files, the following commands may be used

```
% groff -t -man etc/*.man | lpr
%
```

This will produce about 4 pages. The "-t" flag means preprocess with *tbl*(1).

To print copies of the manual entry, the following commands may be used

```
% cd man1
% groff -s -t -man *.1 | lpr
% cd ..
%
```

This will produce about 3 pages. The "-s" flag means preprocess with *soelim*(1), and the "-t" flag means preprocess with *tbl*(1).

GETTING HELP

If you need assistance with the *tardy* program, please do not hesitate to contact the author at

Peter Miller <pmiller@opensource.org.au>

Any and all feedback is welcome.

When reporting problems, please include the version number given by the

```
% tardy -version
tardy version a.b.cccc
...
%
```

command.

LICENSE

The *tardy* program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.

The *tardy* program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

It should be in the *LICENSE* file included in this distribution.

AUTHOR

Peter Miller EMail: pmiller@opensource.org.au
/\/* WWW: http://miller.emu.id.au/pmiller/

NAME

tardy − a tar post-processor

SYNOPSIS

tardy [*option*...][*infile* [*outfile*]]
tardy −Help
tardy −VERSion

DESCRIPTION

The *tardy* program is used to manipulate the file headers within archive files. Supported archive formats include *tar*(1), *cpio*(1) and *ar*(1).

This is useful when preparing sources for Internet archive sites, and you don't want to included details of your development environment in the *tar*(1) file.

One of the most useful features is the ability to add a prefix to the names of the files within the *tar*(1) file. An example is adding the prefix "./", for when you have a *tar*(1) file with absolute paths in it, and need to extract the file with relative paths.

The GNU tar format headers are understood, including extended names.

If no input file is named, or the special name "−" is used, input will be read from the standard input. Similarly, if no output file is named, or the special name "-" is used, output will be written to the standard output. Note that *tardy* will refuse to read from or write to a terminal, as this indicates that the user has made a mistake.

If the input is in *gzip*(1) format, it will be automatically *gunzip*(1)ed before processing. Similarly, if the output file name indicates that *gzip*(1) should be used, this will be done automatically on output.

OPTIONS

The following options are understood:

−Block_Size *number*
: This option many be used to specify the block size, as a multiple of 512 bytes. The default is `−bs=20` resulting in a 10kB block size.

−Clean_Meta
: This option may be used to ensure that file names do not contain shell meta characters. If any are found, they are replaced by a hyphen (−) character. Warning: duplicate file names may result.

-Clean_Print
: This option may be used to ensure that file names only contain printable ASCII characters. If any non-printable characters are found, they are replaced by a hyphen (−) character. Warning: duplicate file names may result.

−Clean_Space
: This option may be used to ensure that file names contain no white space characters. If any are found, they are replaced by a hyphen (−) character. Warning: duplicate file names may result.

−Clean
: This option is a terse way of specifying all of the above 3 options.

−DownCase
: This option may be used to force file names to be all lower case. Warning: duplicate file names may result.

−EXclude *pattern*
: This option may be used to completely remove files from the archive. All files that have a filename patching the given shell filename pattern will be removed. Be careful with wild cards, they match everything, including slash (/) characters. The exclude pattern is applied before all filename manipulating patterns.

−eXtract
This option may be used to extract the contents of an archive.

−Group *arg*
This option may be used to set both the group name and group number fields in the header of every file in the archive file. The argument may be either a string or a number. It is an error if a corresponding entry cannot be found in the */etc/group* file.

-Group_NAme *string*
This option is used to set the group name field in the header of every file in the archive file. The string may be any arbitrary string, it is not restricted to a known group.

−Group_NAme *number*
This option is used to set the group name field in the header of every file in the archive file. The number is mapped to a group name through the */etc/group* file. It is an error if a corresponding group cannot be found.

−Group_NUmber *string*
This option is used to set the group number field in the header of every file in the archive file. The string is mapped to a group number through the */etc/group* file. It is an error if a corresponding group cannot be found.

−Group_NUmber *number*
This option is used to set the group number field in the header of every file in the archive file. The number may be any arbitrary number, it is not restricted to a known group.

−Help
This option may be used to obtain more information about how to use the *tardy* program.

−Input_ForMaT *name*
By default, the input file is examined to automatically detect the input archive file's format. This option may be used to specify the input format to be used. The input format *name*s are

tar This format understands all of the various *tar*(1) formats. This is the default.

tar-bsd This format expects BSD *tar*(1) archives.

ustar This format expects *ustar*(1) format archives.

list The input is a text file which contains one file name per line. The named files are read as if they were the input. It is common for *find*(1) to be used to form the list of file names.

directory
The input is a text file which contains one file name per line. The named files are read as if they were an archive input. If any directories are found, they will be descended recursively.

cpio The file is expected to be in one of the *cpio*(5) formats, and will be examined to figure out which.

cpio-bin This format expects *cpio*(5) "old binary" format archives.

cpio-crc This format expects *cpio*(5) "crc" format archives.

cpio-new-ascii
This format expects *cpio*(5) "new ascii" format archives.

cpio-old-ascii
This format expects *cpio*(5) "old ascii" format archives.

ar The file is expected to be in one of the *ar*(1) formats, and will be examined to figure out which.

ar-bsd This format expects *ar*(1) "BSD" format archives.

ar-pdp11
: This format expects *ar*(1) archives, in the ancient pre-V7 format.

ar-v7 This format expects *ar*(1) “Unix Version 7” format archives.

Any other format name will produce an error.

−List
: Give a terse listing of the file headers on the standard error as they are written to the output.

−Mode_Clear *bits*
: This option may be used to set the mode of each file in the archive file. The *bits* specified are cleared in the mode. You should use an octal number with a leading zero as the argument; if you omit the leading zero it will be interpreted as decimal. The **−Mode_Set** option is applied first, the **−Mode_Clear** option is applied second; if neither is specified the mode of each file is unaltered.

−Mode_Set *bits*
: This option may be used to set the mode of each file in the archive file. The *bits* specified are set in the mode. You should use an octal number with a leading zero as the argument; if you omit the leading zero it will be interpreted as decimal.

−No_Directories
: This option may be used to suppress directories from the output. This can be useful when the archive has been created with directories in useless modes.

−Now
: Set the last-time-modified field of the headers to the current time, all files will be given the same time.

−Old_Type
: By default, tardy corrects the old normal file type into the modern normal file type. This option requests that old normal file type indicators remain unaltered.

−Output_ForMaT *name*
: By default, *tardy* tries to write the same format on output as it sees on input. This option may be used to specify the output format to be used. The output format *name*s are

  ar The archive will be written using the *ar*(1) format, BSD variant. Note that this format only supports simple files; it silently discards directories, and even discards directory portions of file names.

  ar-bsd The archive will be written using the *ar*(1) format, BSD variant. Long file names are handled by a special file at the start of the archive full of file names.

  ar-bsd-l2
  The archive will be written using the *ar*(1) format, BSD variant. Long file names are handled by special file name mangling in the archive, with the actual file name transparently injected into the start of the file data.

  ar-pdp11
  The archive will be written using the *ar*(1) format, in the most ancient pre-V7 variant. (There are also *ar-pdp11-le* and *ar-pdp11-le* names, if you need to select the byte order.)

  ar-port5 The archive will be written using the *ar*(1) format, in the PORT5 variant. (There are also *ar-port5-le* and *ar-port5-le* names, if you need to select the byte order.)

  ar-v7 The archive will be written using the *ar*(1) format, ancient Unix Version 7 variant. (There are also *ar-v7-le* and

  posix The format that GNU tar uses.

  tar-bsd The format that BSD *tar*(1) uses.

cpio The new *cpio*(5) ASCII format.

cpio-bin The *cpio*(5) "old binary" format, using host byte order.

cpio-bin-le
The *cpio*(5) "old binary" format, using littl--endian byte order.

cpio-bin-be
The *cpio*(5) "old binary" format, using big-endian byte order.

cpio-crc The *cpio*(5) "crc ascii" format.

cpio-old-ascii
The old *cpio*(5) ASCII format.

ustar This is the format that Joerg Schilling's *ustar*(1) uses.

v7 The ancient UNIX V7 *tar*(1) format.

Any other format name will produce an error.

−Prefix *string*
This option is used to add a prefix directory name to the name of every file in the archive file.

−PROgress
This option is used to obtain a progress indicator. Only works in combination with the *−ifmt=list* option.

−Remove_Prefix *number*
This option is used to remove a number of leading directories from the name of every file in the archive file, if present. The prefixes will be removed before any prefix specified by the **−Prefix** option is prepended.

−Remove_Prefix *string*
This option is used to remove a prefix directory name from the name of every file in the archive file, if present. This option may be given more than once, and as many of the prefixes as appear will be removed. The prefixes will be removed before any prefix specified by the **−Prefix** option is prepended.

This option is potentially ambiguous with the preceding option, if you wish to remove a leading prefix which looks like a number. If this is the case, add a slash ("/") to the end of the prefix to stop it looking like a number.

−UpCase
This option may be used to force file names to be all upper case. Warning: duplicate file names may result.

−User *arg*
This option may be used to set both the user name and user number fields in the header of every file in the archive file. The argument may be either a string or a number. It is an error if a corresponding entry cannot be found in the */etc/passwd* file.

−User_NAme *string*
This option is used to set the user name field in the header of every file in the archive file. The string may be any arbitrary string, it is not restricted to a known user.

−User_NAme *number*
This option is used to set the user name field in the header of every file in the archive file. The number is mapped to a user name through the */etc/passwd* file. It is an error if a corresponding user cannot be found.

−User_NUmber *string*
This option is used to set the user number field in the header of every file in the archive file. The string is mapped to a user number through the */etc/passwd* file. It is an error if a corresponding user cannot be found.

−User_NUmber *number*
This option is used to set the user number field in the header of every file in the archive file. The number may be any arbitrary number, it is not restricted to a known user.

−VERSion
Print the version of the *tardy* program being executed.

All options may be abbreviated; the abbreviation is documented as the upper case letters, all lower case letters and underscores (_) are optional. You must use consecutive sequences of optional letters.

All options are case insensitive, you may type them in upper case or lower case or a combination of both, case is not important.

For example: the arguments "−help", "−HELP" and "−h" are all interpreted to mean the **−Help** option. The argument "-hlp" will not be understood, because consecutive optional characters were not supplied.

Options and other command line arguments may be mixed arbitrarily on the command line, after the function selectors.

The GNU long option names are understood. Since all option names for *tardy* are long, this means ignoring the extra leading '-'. The "*--option=value*" convention is also understood.

EXIT STATUS

The *tardy* command will exit with a status of 1 on any error. The *tardy* command will only exit with a status of 0 if there are no errors.

SEE ALSO

ar(1) Create, modify, and extract from archives

ar(5) Format of *ar*(1) archive files (not all systems have this man page).

cpio(1) Copy files to and from archives

cpio(5) Format of *cpio*(1) archive files

find(1) Search for files in a directory hierarchy

tar(1) Store and extract files from a tape or disk archive

tar(5) Format of *tar*(1) archive files

COPYRIGHT

tardy version 1.28.D001
Copyright © 1993, 1994, 1995, 1996, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2006, 2008, 2009, 2010, 2011, 2012, 2013 Peter Miller

The tardy program comes with ABSOLUTELY NO WARRANTY; for details use the "*tardy−VERSion License*" command. This is free software and you are welcome to redistribute it under certain conditions; for details use the "*tardy−VERSion License*" command.

AUTHOR

Peter Miller EMail: pmiller@opensource.org.au
/\/* WWW: http://miller.emu.id.au/pmiller/

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program -- to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of

the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your

copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply

to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license,

or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor’s

essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient’s use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others’ Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section

13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> *one line to give the program's name and a brief idea of what it does.*
> Copyright (C) *year name of author*
>
> This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
>
> This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
>
> You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

> <program> Copyright (C) <year> <name of author>
> This program comes with ABSOLUTELY NO WARRANTY; for details type "show w". This is free software, and you are welcome to redistribute it under certain conditions; type "show c" for details.

The hypothetical commands "show w" and "show c" should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

Table of Contents(Tardy)

	The README file	1
	Release Notes	2
	How to build Tardy	5
tardy(1)	a tar(1) post-processor	9
tardy_license(1)	GNU General Public License	14

tardy(1) 9 tardy - a tar post-processor
tardy(1) 9 tardy - a tar post- processor
tardy(1) 9 tardy - a tar post-processor
tardy(1) 9 tardy - a tar post-processor