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/