README: Installation instruction for dired-4.07

Last update: **Fri Jun 17 17:17:02 2011**

Table of contents

Jump start

As with most GNUware, you can build, test, and install this program on most UNIX systems by these simple steps

```
csh et amici:
        setenv CC ...your favorite C or C++ compiler...
        ./configure && make all check install

sh et amici:
        CC=...your favorite C or C++ compiler...
        export CC
        ./configure && make all check install
```

Or in *one* line, if you have `env` (most modern UNIX systems do):

```
        env CC=... ./configure && make all check install
```

If you don't set the `CC` environment variable, then `gcc` (or `cc`, if `gcc` is not available) will be assumed.

If you wish to undo a *make install*, just do *make uninstall*; this will remove any files in system directories put there by *make install*.

See below for further details, and for instructions for non-UNIX systems.

Introduction

Please report all problems, suggestions, and comments to the maintainer and co-author:

Nelson H. F. Beebe
University of Utah
Department of Mathematics, 110 LCB
155 S 1400 E RM 233

Salt Lake City, UT 84112-0090
USA
Tel: +1 801 581 5254
FAX: +1 801 581 4148
Email: beebe@math.utah.edu , beebe@acm.org , beebe@computer.org , (Internet)
WWW URL: http://www.math.utah.edu/~beebe/

The principal author is no longer able to maintain this program.

Installation

`dired` 4.00 has been updated to use the GNU `autoconf` automatic configuration system for UNIX installations.

GNU `autoconf` is run at the maintainer's site to produce the `configure` script from `configure.in` .

The `configure` script is run at each installer's UNIX site to produce `Makefile` from `Makefile.in` , and `config.h` from `config.hin`. The `configure` script is a large (4100+ lines) Bourne shell program that investigates various aspects of the local C implementation, and records its conclusions in `config.h`.

For convenience and safety, the distribution includes a subdirectory named `save` that contains read-only copies of the files `Makefile`, `config.h`, and `configure` created by `autoconf` and *make configure*. This will allow recovery from a lost or damaged `configure` file.

Should you do a *make maintainer-clean* [**not** recommended, except at the maintainer's site], the configure script will be deleted, and you will need recent versions of both GNU `m4` and `autoconf` correctly installed to reconstruct things, which can be done this way:

```
make -f save/Makefile reconfigure
```

Suitable hand-crafted `config.h` files are provided for non-UNIX systems, and in the unlikely event of a failure of the `configure` script on a UNIX system, `config.h` can be manually produced from a copy of `config.hin` with a few seconds of editing work. If you do this, remember to save a copy of your `config.h` under a different name, because running `configure` will destroy it. If you have GNU `autoconf` installed (the installation is very simple and source code is available from `ftp://prep.ai.mit.edu/pub/gnu/autoconf-` *`x.y`* `.tar.gz`), you might try augmenting `configure.in` instead, then run `autoconf`, `autoheader`, and `configure`.

Thus, on UNIX, installation normally consists of just two steps (assuming a `csh` -compatible shell):

```
setenv CC ...your favorite C or C++ compiler...
./configure && make all check install
```

If you like, add `OPT='your favorite optimization flags'` to the `make` command; by default, no optimization flags are set.

The GNU standard installation directories `/usr/local/bin` for binaries, and `/usr/local/man/man1` for manual pages are assumed. The prefix `/usr/local` can be overridden by providing an alternate definition on the command line:

```
make prefix=/some/other/path install
```

After installation, you can do

```
make distclean
```

to restore the directories to their distribution state. You should also do this between builds for different architectures from the same source tree; *neglecting to do so will almost certainly lead to failure*, because the `config.cache` file created by `configure` will lead to an incorrect `config.h` for the next build.

UNIX Systems

The code can be compiled with either C (K&R or ISO/ANSI Standard C) or C++ compilers. With some C++ compilers, it may be necessary to supply additional switches for force the compiler to stay in C++ mode, rather than reverting to C mode (e.g., on DEC Alpha OSF/1, you must do `setenv CC "cxx -x cxx"`).

On UNIX systems, the only changes that you are likely to need in the `Makefile` are the settings of `CC` and `CFLAGS`, and possibly, `DEFINES`, and if you wish to do *make install*, the settings of `bindir`, `MANDIR`, and `MANEXT`.

These programs have been successfully built and tested with C and C++ compilers and tested on these systems for the 4.00 release (106 builds):

Machine and model	**O/S**	**Compilers**
DEC Alpha 2100-5/250	OSF/1 3.2	/bin/c89, /bin/cc, /bin/cxx -x cxx, /usr/bin/c89, /usr/bin/cc, /usr/ccs/bin/c89, /usr/ccs/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc, /usr/ucb/cc
DECstation 5000/200	ULTRIX 4.3	/bin/cc, /usr/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc, /usr/local/bin/lcc -A -A
HP 9000/735	HP-UX 10.01	/bin/CC, /bin/c89, /bin/cc, /usr/bin/CC, /usr/bin/c89, /usr/bin/cc, /usr/ccs/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc
IBM PowerPC 43P	AIX 4.1	/bin/c89, /bin/cc, /bin/xlC, /usr/bin/c89, /usr/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc
IBM PowerPC 43P	AIX 4.2	/bin/c89, /bin/cc, /bin/xlC, /usr/bin/c89, /usr/bin/cc
IBM RS/6000-370	AIX 3.2.5	/bin/c89, /bin/cc, /usr/bin/c89, /usr/bin/cc, /usr/local/bin/gcc

Intel Pentium II MMX (300MHz)	Linux 2.0.33	/usr/bin/cc, /usr/bin/g++, /usr/bin/gcc
Intel Pentium MMX (200MHz)	Linux 2.0.30	/usr/bin/cc, /usr/bin/g++, /usr/bin/gcc
NeXT Turbostation	Mach 3.3	/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc
SGI Challenge L	IRIX 5.3	/bin/CC, /bin/cc, /usr/bin/CC, /usr/bin/DCC, /usr/bin/NCC, /usr/bin/cc, /usr/bin/ncc, /usr/local/bin/g++, /usr/local/bin/gcc
SGI O2 R10000-SC	IRIX 6.3	/bin/CC, /bin/c89, /bin/cc, /usr/bin/CC, /usr/bin/DCC, /usr/bin/NCC, /usr/bin/c89, /usr/bin/cc, /bin/cc -n32, /usr/bin/cc -n32, /bin/DCC -32, /bin/NCC -32, /bin/cc -32, /usr/bin/DCC -32, /usr/bin/NCC -32, /usr/bin/cc -32
SGI Origin/200-4	IRIX 6.4	/bin/CC, /bin/c89, /bin/cc, /usr/bin/CC, /usr/bin/DCC, /usr/bin/NCC, /usr/bin/c89, /usr/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc, /bin/cc -n32, /usr/bin/cc -n32, /bin/DCC -32, /bin/NCC -32, /bin/cc -32, /usr/bin/DCC -32, /usr/bin/NCC -32, /usr/bin/cc -32, /bin/cc -64, /usr/bin/cc -64
Sun SPARC 20/512	Solaris 2.6	/opt/SUNWspro/bin/CC, /opt/SUNWspro/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc, /usr/local/bin/lcc -A -A
Sun SPARC 4/380	SunOS 4.1.3	/bin/cc, /usr/bin/cc, /usr/lang/acc, /usr/local/bin/g++, /usr/local/bin/gcc, /usr/ucb/cc

A tour of `configure.in`, or why porting `dired` is hard

As of early July, 1998, I have now adapted 18 of my software packages to use GNU `autoconf` and `autoheader` to generate a `configure` script that can be run on any UNIX or POSIX system to examine the environment, set various flags in `config.h`, and produce a customized `Makefile` from `Makefile.in`.

When this process is successful, it is of enormous value to end users, because installation becomes a trivial one-line command. However, it is a painful process for software developers, and each package adaptation has taken me longer than I expected: for `dired`, my labors stretched over four long days, and involved more than a thousand builds on the systems listed above. The resulting `configure.in` script, at 521 lines, is almost twice as long as any of the others that I've written so far; the others range from 54 to 281 lines, with an average of 132 lines. Fortunately, very little of this labor is specific to `dired`: I can use much of `configure.in` for the next package that I adapt to GNU-style configuration.

Despite the fact that the C programming language has had American and international standards since December 14, 1989, and the draft of the C Standard was stable for at least two years before that, many vendors continue to use non-standard extensions in their header files, making compilation with a strict Standard C compiler, like the excellent `lcc` compiler, impossible.

There are several reasons why `dired` so hard to port to new architectures, reflecting the sorry divergences of almost 30 years of UNIX development, vendors' differing attempts to deal with the divergences, and frequently, careless coding and inadequate testing by vendor programming staff. The points in the list below follow the tests in `configure.in` approximately; those 500+ lines of tests expand into 4100+ lines of Bourne shell script to implement them. For comparison, the `dired` source code amounts to about 7700 lines of C (also compilable with C++).

- There are three existing UNIX terminal interfaces, corresponding to the header files `sgtty.h` (Berkeley UNIX), `termio.h` (AT&T System V), and `termios.h` (XPG3, POSIX.1, FIPS 151-1).
- For terminal screen access, `dired` uses the `curses` library, and its underlying terminal support which is found in three different libraries, `-lcurses`, `-ltermcap`, and `-ltermlib` .
- `dired` uses `signal` to catch the interrupts `SIGINT`, `SIGQUIT`, and `SIGTSTP`. Depending on the system, UNIX signal handlers may be of type `int` or `void`, and may take zero, one, or an indefinite number of arguments. Because `signal` takes a function argument, the type of that function must be perfectly correct, or else cast to the correct type, in C++ compilation.
- `dired` uses `wait` to wait for subprocesses; that function takes an argument of type `union wait *` in old Berkeley-style systems, and of type `int *` in newer ones.
- `dired` provides its own regular-expression support, under the same function names as implemented on some UNIX systems; sometimes, the argument types differ.
- C++ compilation on any system must deal with numerous deficiencies in vendor-provided header files, either from omission of required library function prototypes, or from incorrect ones. These deficiencies are particularly severe with the `curses` support. There are about a dozen library functions for which prototypes are missing on one of more systems; these prototypes *must* be provided for successful compilation under C++. One of them, `tputs` , takes a function argument whose type differs between systems.
- `dired` reads file directories. There are at least five approaches to this: reading the directory directly (very old UNIX systems, but the code to do so remains inside `dired`, showing its origins on the venerable PDP-11 of the mid 1970s), `sys/dir.h` , `sys/ndir.h`, `ndir.h`, and `dirent.h`. Some of these use a `struct direct` , and others a `struct dirent`. On some systems, the filename stored in that structure may not be NUL-terminated.
- Several vendors hide common library functions, exposing them only in response to definition of preprocessor symbols, such as `_ALL_SOURCE`, `_HPUX_SOURCE`, `_POSIX_SOURCE`, and `_XOPEN_SOURCE`. On at least one system, NeXT Mach 3.x, use of `_POSIX_SOURCE` also requires a special compiler option, `-posix`, which changes the run-time libraries used. Failure to supply this flag results in either core dumps at run time, or incorrect reading of directories.
- `dired` uses `stat` to retrieve a directory entry structure; the flags and macros needed to do so are capriciously hidden on some systems.
- The type of the argument passed to `ctime` may be either `long *` or `time_t *`, depending on the system.

IBM PC DOS

`dired` has not yet been ported to the IBM PC DOS platform.

Sample build output for UNIX

Here is a log of a successful build on Sun Solaris 2.6 using the native C++ compiler, `CC`:

```
% env CC=cc ./configure && make all check install
creating cache ./config.cache
checking whether make sets ${MAKE}... yes
checking for gcc... cc
checking whether the C compiler  cc    works... yes
checking whether the C compiler  cc    is a cross-compiler... no
checking whether we are using GNU C... no
checking for a BSD compatible install... /usr/local/bin/install -c
checking whether ln -s works... yes
checking for col... col -x -b
checking for gawk... gawk
checking for chmod... chmod
checking for checksum... checksum
checking for rcp... rcp
checking for cmp... cmp
checking for deroff... deroff
checking for diff... diff
checking for distill... distill
checking for strip... strip
checking for less... /usr/local/bin/less
checking for dw... dw
checking for geqn... geqn
checking for gzip... gzip
checking for ispell... ispell
checking for ln... ln
checking for lpr... lpr
checking for ls... ls
checking for man2html... man2html
checking for mkdir... mkdir
checking for mv... mv
checking for groff... groff
checking for rm... rm
checking for rmdir... rmdir
checking for sed... sed
checking for shar... shar
checking for sort... sort
checking for spell... spell
checking for strip...  cached  strip
checking for gnutar... no
checking for gtar... no
checking for tar... tar
checking for gtbl... gtbl
checking for tr... tr
checking for tex... /usr/local/lib/tex
checking for touch... touch
checking for unzip... unzip
checking for zip... zip
checking for zoo... zoo
checking for Standard C/C++ function declarations... yes
checking how to run the C preprocessor... cc -E
checking for ANSI C header files... yes
checking for assert.h... yes
checking for config.h... no
checking for ctype.h... yes
checking for curses.h... yes
checking for dirent.h... yes
checking for errno.h... yes
checking for fcntl.h... yes
checking for grp.h... yes
checking for limits.h... yes
checking for memory.h... yes
checking for pwd.h... yes
checking for re_comp.h... yes
checking for signal.h... yes
checking for sgtty.h... yes
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for sys/43ioctl.h... no
checking for sys/dir.h... no
checking for sys/ioctl.h... yes
checking for sys/mkdev.h... yes
checking for sys/param.h... yes
checking for sys/stat.h... yes
checking for sys/sysmacros.h... yes
checking for sys/types.h... yes
checking for sys/wait.h... yes
checking for term.h... yes
checking for termio.h... yes
checking for termios.h... yes
checking for time.h... yes
checking for unistd.h... yes
checking return type of signal handlers... void
checking for SIG_PF typedef... no
checking for re_comp   prototype... yes
checking for memset   prototype... yes
checking for tgoto   prototype... no
checking for tgetent   prototype... yes
checking for tgetnum   prototype... yes
checking for tgetstr   prototype... yes
checking for tputs   function final argument type... char
checking for tputs   prototype... yes
checking for ioctl   prototype... yes
checking whether sys/types.h defines makedev... no
checking for sys/mkdev.h...  cached  yes
checking for tcgetattr   prototype... no
checking for strchr   prototype... no
checking for strcpy   prototype... no
checking for strncpy   prototype... no
checking for strrchr   prototype... no
checking for strcat   prototype... no
checking if _ALL_SOURCE needed to expose S_IREAD in <sys/stat.h>... no
checking if _HPUX_SOURCE needed to expose S_IREAD in <sys/stat.h>... no
checking if _XOPEN_SOURCE needed to expose S_IFMT in <sys/stat.h>... no
checking for Standard C/C++ prototype support... yes
checking for working const... yes
checking for size_t... yes
checking for atoi... yes
checking for isatty... yes
checking for getgrgid... yes
checking for getpwuid... yes
checking for opendir... yes
checking for remove... yes
checking for unlink... yes
checking if -lcurses available... yes
checking for argument type of ctime  ... const time_t *
checking for struct stat... yes
checking if _POSIX_SOURCE needed for struct dirent... no
checking if -posix needed... no
updating cache ./config.cache
creating ./config.status
creating Makefile
creating config.h
cc  -DV4P2 -DHAVE_CONFIG_H  -I. -DDIREDFILE= "/usr/local/bin/dired " -DHELPFILE= "/usr/local/bin/dired.hlp " -DMAXFILES=10240 -DMOREPGM= "/usr/local/bin/less " -c dired.c
cc  -DV4P2 -DHAVE_CONFIG_H  -I. -c color.c
cc  -DV4P2 -DHAVE_CONFIG_H  -I. -c cshsystem.c
cc  -DV4P2 -DHAVE_CONFIG_H  -I. -c regexpr.c
"regexpr.c", line 1383: warning: statement not reached
cc  -DV4P2 -DHAVE_CONFIG_H  -I. -o dired dired.o color.o cshsystem.o regexpr.o -lcurses
There is no validation suite for dired...you must run it manually
rm -f /usr/local/bin/dired /usr/local/bin/dired-4.00 /usr/local/bin/dired.hlp
rcp dired /usr/local/bin/dired
ln /usr/local/bin/dired /usr/local/bin/dired-4.00
strip /usr/local/bin/dired
chmod 755 /usr/local/bin/dired
rcp dired.hlp /usr/local/bin/dired.hlp
```

```
chmod 644 /usr/local/bin/dired.hlp
rm -f /usr/local/man/man1/../cat1/dired.1
rm -f /usr/local/man/man1/dired.1
rm -f /usr/local/man/man1/../cat1/dired.1
sed -e "s@DIRED@dired@g" -e "s@DIRED@`echo dired | tr a-z A-Z`@g" -e "s@dired-x@dired-x@g" -e "s@<STRONG>more</STRONG>[ ]1[ ]@<STRONG>/usr/local/bin/less</STRONG>@" -e "s@[.]BR more [ ]1[ ]@.B /usr/local/bin/less@" dired.man > /usr/local/man/man1/dired.1
chmod 644 /usr/local/man/man1/dired.1

Installed files...
-rwxr-xr-x   1 beebe    staff     129128 Jul 10 10:03 /usr/local/bin/dired
-rwxrwxr-x   1 beebe    staff     140832 Jul 10 10:03 /usr/local/bin/dired-4.00
-rw-r--r--   1 beebe    staff      25549 Jul 10 10:03 /usr/local/man/man1/dired.1
========================================================================
========================================================================
== Finally, you may wish to install the HTML form of the dired        ==
== documentation.  Try "make -n install-html" to see if that is       ==
== appropriate for your system.                                       ==
========================================================================
========================================================================
```