NAME

dired − file directory editor

SYNOPSIS

dired [**−?**] [**−b**] [**−d**] [**−f** *firstfile*] [**−F** *firstfile*] [**−g***[gid-or-groupname]*] [**−h**] [**−m**] [**−**[**s**|**r**][**c**|**g**|**G**|**i**|**l**|**n**|**N**|**r**|**s**|**t**|**u**|**U**|**w**]] [**−S**] [**−u***[uid-or-username]*] [**−V**] [**−v**] [**−w**[**f**|**h**|**number**]] [*directory-list*|*file-list*]

DESCRIPTION

dired displays a long-form **ls**(1) directory listing on the screen of a display terminal and allows you to 'edit' and peruse that listing by moving up and down, deleting, undeleting, editing, and displaying entries. The **emacs**(1) text editor provides a similar capability in an editor window. The shell *TERM* variable should be set to the standard string which the Berkeley *termcap* library uses for distinguishing terminals. With no argument, the connected directory is used. With only one argument, if that argument is a directory, it is used. With multiple arguments, (or a single non-directory argument) the argument(s) are interpreted as filenames. **dired** then types 'Reading' and gets information about the various files/directories in your specification. This may take a short while (depending on how many you give it), so it types one period (.) after the word 'Reading' for every 10 files it has gathered information about. With this, you can keep track of its progress. Interrupts, hangups, and the like are disabled since your terminal is put into a special mode that is only changed when you quit with the *q* command.

The format of the screen is as follows: each line represents a file (or directory), the name of which is rightmost. From left the fields are: mode, link count, owner, size, write date and name. See **ls**(1) for a description of what each of these mean. You move up and down the column immediately left of the filename. The bottom half of the screen is used for displaying files via the **t**ype command. If there are too many files to all fit on one window, more windows are allocated. The **f** and **b** commands can be used for stepping forward and backward windows. The last screen line is used as an 'echo' line for displaying error messages and reading arguments. It also displays the full directory name if 'direding' a directory. When in split screen mode, the divider serves also as a 'linear indicator' showing where the current window is relative to the entire list of files. The symbols '(' and ')' denote the window. Square brackets replace '(' and/or ')' when the window is the first and/or last window. A single 'o' is used to represent the window when the window size is small compared to the total number of files.

OPTIONS

Command-line options, which are inherited by recursive invocations of **dired**, are:

−? Display a brief help message on *stderr* and quit with a success return code.

−b Select batch mode: **dired** then sends its output to *stdout* without reading keyboard input, and without colorizing filenames. See the **COLOR SUPPORT** section below for details.

dired will normally select batch mode automatically when its output is not a terminal, but this option provides a way to force that for systems where **dired** is unable to detect non-terminal output.

Use this option in conjunction with the sorting options described below as an alternative to **ls**(1).

−d This flag is normally not specified by users; **dired** sets it for **dired** subprocesses.

−f *firstfile*

Start the output with the specified file on the first line of the display, but if that file is a directory, prevent attempts to edit it recursively with the 'e' command.

This option is ignored in batch mode.

−F *firstfile*

Start the output with the specified file on the first line of the display.

This option is ignored in batch mode.

−g*[gid-or-groupname]*

Show only files belonging to the specified group id number or group name. If the number or name is omitted, then the group id of the current user is assumed.

−[s|r][c|g|G|i|l|n|N|r|s|t|u|U|w]
Forward or reverse sort by *inode-change date*, *group name*, *group number*, *link count*, *name* (lexicographic) , *name* (numeric) , *read date*, *size*, *type*, *user name*, *user number*, or *write date* respectively. ''Normal sort'' is the order most often desired; it is descending for size and link counts, from newest to oldest for date sorts, and ascending for other sorts. The default is to sort by name.

For equal sort keys, do a secondary sort to put the filenames in ascending order.

Numeric sorting of filenames with **−sN** is useful in directories such as */proc* and Usenet news trees, where files are named 1, 2, 3, ...

−h Display a brief help message on *stderr* and quit with a success return code.

−m Monochrome mode: suppress the default colorization of filenames. Colorization can still be toggled on and off in the **dired** session with the *C* command. See the **COLOR SUPPORT** section below for details.

−u*[uid-or-username]*
Show only files belonging to the specified user id number or user name. If the number or name is omitted, then the user id of the current user is assumed.

−S Do not show symbolic link targets, reducing output line width requirements. The directory status display then reflects the file that the symbolic link points to, rather than the link itself. Without this option, the display corresponds to the link.

−V Same as **−v**.

−v Show the program version on *stderr*, and quit immediately.

−w[f|h|number]
Use **number** lines for the directory index window, reserving the other half for quick file display. **f** means use the full screen for the index. **h** means use half of the screen for the index. **f** is the default.

KEYBOARD COMMANDS

Commands consist of single characters, with any necessary arguments prompted for, and displayed in the echo line. Several commands take an optional non-negative integer count argument, as in **vi**(1).

Here are the keyboard commands that control **dired**, organized into several categories:

Help

?, **h** Display a help message summarizing the keyboard commands.

Quitting dired

a Abort out of the current directory. No deletions are done.

A Abort completely out of **dired**, with no deletions.

q Exit **dired**, displaying files marked for deletion and requiring confirmation before deleting them. If no confirmation is given (typing anything other than *y*), **dired** goes back to its display.

Q Quit **dired**, with no deletions.

Actions on files

! Prompt for a system command to invoke. The command is executed, and confirmation is required before returning to the display. All **%** characters in the command are replaced with the full pathname of the current entry, and all **#** chars are replaced with just the trailing filename component (what you see on the screen).

. Repeat the previous **!** shell command, substituting the current entry for any special chars (%#) in the original command.

e Run the editor defined in the *EDITOR* environment variable upon the current file. If *EDITOR* is not defined, **vi**(1) is used. However, if the current file is a directory,

it is not edited, but rather, **dired** forks a copy of itself upon that directory. In this manner, you can examine the contents of that directory and thus move down the directory hierarchy.

m Run the program defined by the *PAGER* environment variable program on the current entry. If *PAGER* is not defined, run **more**(1).

P Print the current file on the lineprinter.

t Type the file to the terminal; this is considerably faster than starting an editor on the file. In two-window mode, the bottom window is used, pausing after each screenful. The type-out may be interrupted by **<control C>** or **q**.

T Same as **t**, but without any pauses at end-of-screen.

Deletion and undeletion

d Mark the current entry for deletion. Upon exit and confirmation (or re-reading using the **'R'** command), this entry will be deleted. *Warning:* this includes directories! If it is a directory, everything in it and underneath it will be removed.

D Mark *all* files for deletion. (**'U'** undoes this).

~ Mark files with names ending in ~ for deletion.

Mark files with names beginning with # for deletion.

u Undelete the current entry, if it was previously marked for deletion.

U Cancel all deletion requests.

Display

<space> Re-print the directory path name.

c Refresh the current line.

**** Toggle between split-screen mode and full-screen mode.

<control G> Show the current file number, the total number of files, and the percentage through the file. Useful in full screen mode when there is no linear indicator.

l, **<control L>**
Refresh the current window.

p Display the full path name of the current file; embedded control characters are shown with graphics.

R Re-read the directory or file list. If files are marked for deletion, **dired** will first ask for confirmation and then delete them before re-reading. This is useful after operations done during shell escapes (e.g., **chmod**(1)).

r, **s** Sort the file list by various fields, with the field selected by the next input character. If that character is **<ESC>**, then cancel the sort request. See the documentation of the **−s** option above for a description of the sort fields and sort orders.

Miscellaneous

<ESC> Cancel a sort or a count.

C Toggle colorization of filenames.

Moving around

<down arrow>, **<lf>**,**^N**, **j**
Step to the next file. If this crosses a window boundary, the next window is displayed with a one-line overlap. May be preceded by a count.

<up arrow>, **^**, **-**, **k**, **<backspace>**, **^P**
Step to previous file. If this crosses a window boundary, the previous window is displayed with a one-line overlap. May be preceded by a count.

<PageDown>, **<right arrow>**, **f**
Go forward a window, leaving a one-line overlap. May be preceded by a count.

<PageUp>,**<left arrow>**, **b**
Go backward a window, leaving a one-line overlap. May be preceded by a count.

<, **[**, **(**, **{** Go to the start of the file list. With an argument *n*, go *n/10* of the way from the start.

>, **]**, **)**, **}** Go to the end of the file list. With an argument *n*, go *n/10* of the way from the end.

E Go up to the next higher level directory. In the case of an argument list of files to **dired**, go to the parent of the directory which contains the current file.

G Go to the file number given by the preceding count. With no count, go to the last file as in **vi**(1).

M Remember the current entry on the mark stack. You can later return to it with the **J** (jump) command.

J Pop the top entry from the mark stack, and jump to it.

L Pop the top entry from the view stack, and jump to it. Each screen display adds an entry to the view stack, except for views created by this command, so you can use **L** to display views in reverse order.

Searching

/<regular-expression>
Locate a file matching the given regular expression, as defined in **re_comp**(3) and **re_exec**(3). The search is in the forward direction. The regular expression is remembered for subsequent use by the **n** and **N** commands. If the regular expression is empty, the last one remembered is used.

n Find the next instance in the forward direction of the regular expression previously-defined by a '/' command.

N Find the next instance in the backward direction of the regular expression previously-defined by a '/' command.

COLOR SUPPORT

From version 4.00, **dired** supports color coding of files by type and by extension.

The color support is identical to that provided by the GNU **ls**(1) command: if the terminal type defined by the *TERM* environment variable is known to support color (i.e., is one of *con132x25*, *con132x30*, *con132x43*, *con132x60*, *con80x25*, *con80x28*, *con80x30*, *con80x43*, *con80x50*, *con80x60*, *console*, *linux*, *vt100*, or *xterm*), then built-in defaults determine file colors.

Do not be disappointed if your terminal type is one of the above, but the output is still monochrome: only recent versions of UNIX terminal emulators, such as **xterm**(1), contain support for text color beyond the normal foreground and background colors. However, all should have at least bold, flashing, underlined, and reverse video capability.

The defaults can be overridden by settings of the *LS_COLORS* (or *LS_COLOURS*) environment variable; that variable augments, but does not eliminate, the internal defaults. Thus, for customization, the user need only supply changed values.

The value of the *LS_COLORS* variable is a **termcap**(1)-like capability list: a colon-separated list of *key=value* pairs. As in computer programming languages, when there are repeated assignments to the same key name, only the last is effective.

Keys may be one of these file types or commands:

bd block device,

cd character device,

di directory,

ec end control sequence code (replaces *lc+no+rc*),

ex executable file,

fi regular file,

lc left control sequence code,

ln symbolic link,

mi missing file (defaults to *fi*),

no normal (non-filename) text,

or orphaned symbolic link (defaults to *ln*),

pi named pipe (FIFO),

rc right control sequence code,

so socket.

Keys may also be of the form **.ext* to select files by dotted extension.

Values are terminal color control sequences, usually semicolon-separated lists of numbers, as follows:

0 restore default color

1 brighter colors

4 underlined text

5 flashing text

7 reverse video

30 black foreground

31 red foreground

32 green foreground

33 yellow (or brown) foreground

34 blue foreground

35 purple foreground

36 cyan foreground

37 white (or gray) foreground

40 black background

41 red background

42 green background

43 yellow (or brown) background

44 blue background

45 purple background

46 cyan background

47 white (or gray) background

For convenience, here is the same data, sorted by color names instead of color numbers:

40	black background
30	black foreground
44	blue background
34	blue foreground
46	cyan background
36	cyan foreground
42	green background
32	green foreground
45	purple background
35	purple foreground
41	red background
31	red foreground
47	white (or gray) background
37	white (or gray) foreground
43	yellow (or brown) background
33	yellow (or brown) foreground

Any required special characters in capability values can be represented by backslash or caret escape sequences:

\?	rubout (ASCII DELete)
_	space
a	alert (ASCII BEL)
b	backspace (ASCII BS)
e	escape (ASCII ESCape)
f	formfeed (ASCII FF)
n	newline (ASCII NL)
ooo	3-octal-digit character value
r	carriage return (ASCII CR)
t	horizontal tab (ASCII HT)
v	vertical tab (ASCII VT)
xhh	2-or-more-hexadecimal-digit character value
^*x*	Control character formed from the five low-order bits of the character ‘x’

Backslash can also be used to protect other characters, notably, backslash, colon, and equals, from misinterpretation: ‘\\’ reduces to a single ‘\’, and ‘e:’ to a non-separating colon.

The GNU **dircolors**(1) command can be used to display the default setting of *LS_COLORS* used by **dired**. For convenience, the default is displayed here, but for readability, newlines have been inserted after the colons separating *key=value* pairs, and the pairs have been sorted into ascending order:

**.arj=01;31:*
**.avi=01;37:*
**.bmp=01;35:*
**.deb=01;31:*
**.dl=01;37:*
**.gif=01;35:*

**.gl=01;37:*
**.gz=01;31:*
**.jpg=01;35:*
**.lzh=01;31:*
**.mpg=01;37:*
**.ppm=01;35:*
**.tar=01;31:*
**.taz=01;31:*
**.tga=01;35:*
**.tgz=01;31:*
**.tif=01;35:*
**.xbm=01;35:*
**.xpm=01;35:*
**.Z=01;31:*
**.z=01;31:*
**.zip=01;31:*
bd=40;33;01:
cd=40;33;01:
di=01;34:
ex=01;32:
fi=0:
lc=\e[:
ln=01;36:
no=0:
or=40;31;01:
pi=40;33:
rc=m:
so=01;35:

dired writes each filename to the screen in the form <*lc*><*color*code><rc><filename><ec>. If the *ec* command is undefined, the sequence <*lc*><*no*><*rc*> is used instead. For example, the default setting shown above for the *di* directory file capability, *01;34*, produces bright blue text on the screen background, and the character string that is output to the screen is *\e[01;34mfilename\e[0m*.

If you routinely use colored screen backgrounds, you should define background, as well as foreground, colors in each color capability value; otherwise, you may find that the color selected for certain files is difficult, or impossible, to see against the colored background.

DIAGNOSTICS

The error messages should be self-explanatory.

The exit codes returned to the invoking program are as follows:

0 success (normal termination);

1 failure (usually accompanied by an explanatory message);

2 failed to open `termcap` file;

3 unknown terminal type;

69 user requested abort with *A* or *Q*, or else editor invocation failed.

ENVIRONMENT VARIABLES

LS_COLORS
color capability list; see the **COLOR SUPPORT** section above.

LS_COLOURS
alternate spelling of *LS_COLORS*, used if that name is not defined.

TERM standard UNIX terminal type.

FILES

/usr/local/bin/dired	Executable program.
/usr/local/src/dired	Source directory.
/usr/local/bin/dired.hlp	Help file for **?** and **h**.

SEE ALSO

emacs(1), **ls**(1). **more**(1), **re_comp**(3), **re_exec**(3), **termcap**(1). **vi**(1), **xterm**(1).

AUTHORS

Stuart Mclure Cracraft.

Enhancements by Jay Lepreau <lepreau@cs.utah.edu>.

Fixes and enhancements by Charles Hill.

Regular-expression support code by Tatu Ylonen <ylo@ngs.fi>.

Many fixes and enhancements, major revision of manual pages, support for GNU *autoconfigure* (for simple installation), batch mode, and color support, by Nelson H. F. Beebe <beebe@math.utah.edu>, who is the current maintainer of **dired**.

AVAILABILITY

dired is freely available; its master distribution can be found at

```
ftp://ftp.math.utah.edu/pub/misc
```

in the file *dired-x.yy.tar.gz* where *x.yy* is the current version. Other distribution formats are usually available in the same location.

That site is mirrored to several other Internet archives, so you may also be able to find it elsewhere on the Internet; try searching for the string *dired* at one or more of the popular Web search sites, such as

```
http://altavista.digital.com/
http://www.hotbot.com/
http://www.stpt.com/
http://www.yahoo.com/
```

BUGS

If lines wrap or overprint, it is likely that **dired** has an incorrect notion of the screen size. This can happen on older UNIX systems where the kernel does not track screen sizes, but instead, leaves it up to the shell, which uses environment variables named *LINES* (or *ROWS*) and *COLUMNS* to record the screen dimensions. Print those variables (e.g., *echo $COLUMNS*), and use the **stty -a** command to see the kernel's settings, if any, correct them if needed, and restart **dired**.

On some systems, if **dired** is running under a **script**(1) session, the screen dimensions may be incorrect, and can be reset in the same way as above.