

**DIRNOTES**  
for  
**Windows**

Corel Systems Corp, 1990  
Pat Beirne

**Description**

This is a tiny Windows app which allows the user to add annotation to his directories. For compatibility with DIRNOTES.COM, this app allows up to 38 characters to be associated with each file.

The notes are stored in a file called DIRN-xxx.DAT where xxx is the first three letters of the directory name. If the name is less than 3 letters, '-' is used. For example, for the root, the note file is called DIRN----.DAT. The format for this file is identical to that used by DIRNOTES.COM, a popular DOS program.

When the program is run, it first checks the command line for a file name. The name is split into disk+path, and filename. The disk+path is used to locate the relevent notes file. If the notes file does not exist, it is created. The filename is used to search within the notes for the relevent entry; the exception is filenames that match "DIRN-???.DAT", which not used to search.

There is an edit box at the top of the screen, into which the comments are duplicated. The user may modify the 38 characters of the comments, in this box. Any scroll activity in the listbox will take away and save the comments from the edit box.

The menu for this app is simply:

File  
    Exit/Update  
    Quit/NoUpdate  
    About  
Edit  
    Undo  
    Cut  
    Copy  
    Paste

NOTE: the edit box has been subclassed, so that the Up/Dn/PgUp/PdDn arrows scroll the list box. The Directory list box has been subclassed so that the space bar acts as if you double-clicked.

## Program Structure

### Startup

- register the windows class
- wake up the main and child windows
- fill them with data
- message loop

The main window is created through *CreateDialog()*. This allows us to specify the shape of the child elements in the .RC file. Since it should look-and-feel like a dialog, the *IsDialogMessage()* is called in the message loop.

At exit time, the sub-classing proc-instances are freed.

### Main WndProc

Since the program acts like a dialog box, we don't bother painting. The only messages we pick up are:

WM_SIZE:	to adjust the size of the list box
WM_DESTROY:	to do auto-save
WM_SETFOCUS:	so that it wakes up "ready-to-type"
WM_INITMENU:	to grey out certain entries
WM_COMMAND:	to receive messages from the children & menu

### Wakeup

The *WakeUp()* code calls the *CreateDialog()* to start up the main window and the three children. It then measures the system font and the fixed-space system font, to adjust windows sizes. It also configures the child boxes:

- edit box: limit text entry to 38 characters
- file list box: use a fixed-space font

and then it subclasses the edit box and the directory box

### FillListBox

When the correct directory is known, a dir is done, using the *\_findfirst()*, *\_findnext()*. The result is *wspri*nted into the main list box, and sorted as it's entered.

Each entry in the dir-pool has a capital 'U' appended to it (at column 40).

The relevant notes file is read into ram and parsed on the fly, into fixed length records. As the records are read in, they're compared to the entries in the main list box. Non-matching entries are discarded. Matching entries are checked for identical length and date. If the dir-pool date/time is the same as the old-pool date/time, the 'U' is cleared to a blank. Any existing comments are moved from the input file into the list box.

When all the records are moved over, the windows are displayed, showing the filename, extension, size, date, time, new-marker and comments. If the program was started with a specific file, the list box is scrolled to that entry.

Note: in this program, directories are included in the listing, where they are not in the DOS program DIRNOTES.COM.

Corel Systems Corp, 1990



## CopyComments

Every time the user scrolls the entry in the file list box, we move the comments from the edit box into the corresponding entry in the list box. The flag *fLocalDirty* indicates whether the comments have been modified. Note that when you replace an entry in a list box, you must WM\_INSERTSTRING before you WM\_DELETESTRING. Note also that we push and pop the global flag *fDirty*, since the act of stuffing the edit box will cause the *fDirty* flag to be set, which is not significant.

## NewDir

If the user changes directories, we first save the old comment file (if it has been touched: *fDirty*). Then we get the new directory and parse it for the comment-file-name, and the re-fill the list box.

The rest is self explanatory.

Typical comment file.

43	COM	24	4-11-89	11:23p	my 43 line mode change
A	DOC	2384	3-13-88	12:07a	quick subdir goto
A	EXE	11178	3-12-88	8:18p	
C	EXE	10544	7-12-89	4:56p	file print formatter
CAL	EXE	11088	7-12-89	4:56p	
CAT	COM	512	6-08-84	3:20p	dir by type
CED	COM	7040	1-01-80	2:42a	
CO	COM	3990	6-01-88	12:00p	copy files
COMM	COM	1908	6-04-85	8:37p	
CRC	COM	1536	10-21-82	5:50p	
CS	COM	119	6-01-88	12:00p	clear screen
DEBUG	COM	15552	6-23-86	1:01p	
DI	EXE	9596	1-03-89	4:51p	disk info NORTON
DIFF	COM	33776	2-02-88	8:32a	file compare
DINSTALL	COM	3988	4-15-88	12:00p	
DIRMAGIC	COM	8496	2-21-89	10:05a	
DIRN-UTI	DAT	14661	5-15-90	10:59a	
DIRNOTES	COM	1697	6-01-88	12:00p	
DM	COM	4421	4-15-88	12:00p	
DR	COM	3793	6-01-88	12:00p	
DRM	COM	3976	9-27-89	12:51a	
EXES	<DIR>		11-12-88	12:00a	my exectables
FA	EXE	9334	1-03-89	4:51p	NORTON file attrib
FC	EXE	14576	5-15-85	12:00a	file compare
FC20	ZIP	33546	4-06-90	10:23a	
FCN	EXE	44382	3-21-90	12:00a	ascii file compare
FD	EXE	10446	1-03-89	4:51p	
FF	EXE	9042	1-03-89	4:51p	
FFIND	EXE	12441	11-07-89	9:53p	function finder for C code
FI	EXE	18466	1-03-89	4:51p	file info
FILEINFO	FI	3217	1-03-89	4:51p	
FR	EXE	44670	1-03-89	4:51p	format recover NORTON
FREE	COM	343	6-05-86	10:54a	show free space on disk
FS	EXE	9242	1-03-89	4:51p	

Corel Systems Corp, 1990

FV	COM	6180	3-22-90	1:17a	view compressed file names
WSHELL1	ZIP	8915	11-26-90	10:11aU	WIN one-line shell