Search Subdirectories with a Macro
WordPerfect Magazine
By Gayle G. Humpherys

Call me overly excitable, but I've got a confession to make: the List Files feature of WP5.1 sends me into orbit every time I press (F5). Still, as good as it is, there's one thing that I've always wished I could do in List Files: search for a specific file in a directory and all of its subdirectories.

Of course, you can use the Find option in List Files to search through the current directory for files using a specific pattern or containing certain text. But the Find option doesn't search through any subdirectories found under the current directory. If you want to search through those subdirectories, you have to manually repeat the process in each subdirectory – a major annoyance if you have a lot of subdirectories to go through.

To avoid this vexation, just use the included macro DIRSRCH.WPM. This cunning macro searches through a given directory and all of its subdirectories for files with a specific pattern (such as *.wpm) or containing specific text. The results of the search are placed in the second document screen.

In this article I'll show you how to create and use this macro.

Using the Macro

To use the DIRSRCH.WPM macro, first make sure the second document screen is clear. Then press Macro (Alt-F10), type “dirsrch” and press (Enter). The first prompt asks for the main directory to search through. Type the directory you want to search and press (Enter). Any subdirectories under that directory will be included and searched.

The next prompt asks if you want to search for a file pattern or text in a file. Press (1) File Pattern to search for a pattern in the name of the file, such as *.wpm. Press (2) Text in File or just press (Enter) if you want to search through the entire document for a specific word or phrase.

The third prompt asks for the pattern or text to search for. Type the desired file pattern or text depending on the option you just chose and press (Enter). If you're searching for a file pattern, you can use the DOS wildcards (*) and (?) to narrow the search as much as you want. (See the section on wildcards below.)

If you're searching for text and enter more than one word, the macro finds all documents with those words directly next to each other. For example, if you enter "cats and dogs" as the text, the macro finds any files containing those three words next to each other and not files with each word in a separate sentence or on a separate page.

When the search begins, a prompt at the bottom of your screen displays the directory name the macro is currently searching in and keeps track of how many files have been found containing the desired pattern or text. If you're searching through a large directory that contains several subdirectories, the search might take some time. If you press Cancel (F1) while the macro is searching, it switches to the second document screen and creates a list of files that it has found up to the point where Cancel was pressed.

After all the directories have been searched, a list of all these files is typed into the second document screen. The list includes the total number of files found and the complete path and filename of each file. This list can then be saved or printed for reference if desired.

A Few Caveats

Although this macro works great in most cases, you should be aware of a few minor items:

1. The macro has a limit as to how many files it can find: usually between 500 and 800 files. This limit depends on how much conventional memory your PC has available. If you run out of memory before the macro has finished searching, you'll probably get a Divide Overflow error message. If you do, exit WordPerfect completely and reboot your computer to clear your PC's memory before doing any more work in WordPerfect.

   Tip: If you anticipate the macro finding more than about 400 files during the search and you run under a Shell program, consider exiting your Shell program and running WordPerfect directly from a DOS prompt.

2. If the macro finds a filename containing a tilde (~), the macro may not work properly.

3. Subdirectories that have been given an extension (characters after a period), for example, C:\GRAPHICS\CLIPART2.0\, will not work with this macro.

Understanding the Macro

To make it easier to follow the logic of DIRSRCH.WPM, think of the macro as performing four general actions: (1) initializing variables, (2) searching for subdirectories, (3) searching for files in each directory, and (4) creating the list of files found. Each of these actions is briefly described below.

Initializing Variables (lines 1-14). The first 14 lines of the macro gather and assign initial information. Line 2 assigns initial values to three variables used later in the macro. Lines 3-5 prompt the user for the needed information. In lines 6-9, the macro checks to see if the directory entered by the user (line 3) includes an ending backslash and if not, adds one (such as c:\wp51\). Lines 10-11 switch the text of the path entered on line 3 to uppercase characters. Lines 12-14 assign the parent directory of the directory entered by the user to a variable so the macro will know when to quit.

Searching for Subdirectories (lines 15-44). The second section of the macro (lines 15-44) does the searching for any subdirectories. Lines 15-22 utilize the Name Search option of List Files to find the last subdirectory listed in the current directory. The macro then works backwards through any subdirectories-searching them as it goes-until the Parent directory is reached.

Lines 23-36 contain a {WHILE} loop that "backs out" of a subdirectory to its parent directory. Since each subdirectory can have numerous subdirectories of its own and these subdirectories can have even more subdirectories and so on, the macro needs to keep track of the path to the current subdirectory. Lines 23-29 help do this with several different variables.

Lines 31-33 are executed if the parent directory happens to match the variable assigned earlier (line 13)-indicating that the macro has searched through all the subdirectories. If this is the case, these lines assign the path to a variable (line 32), call lines 45-55 to search through that directory and then jump to line 56 where the list of files found is created.

Back on line 30, if the macro is not currently highlighting the Parent directory, but a subdirectory name instead, lines 37-41 again use variables to add the new subdirectory name and a backslash to the current path. The two {Enter} commands at the end of line 41 then list the files in the currently highlighted subdirectory. Lines 42-43 create a prompt that keeps you informed of the directory the macro is searching through and how many files have been found.

The section of the macro that actually searches through the current directory for files is called on line 44. After this search (described below), the macro returns to line 15, where it continues its search for the next subdirectory.

Searching for Files (lines 45-55). The commands that search through the files in the current directory are found in lines 45-55. Lines 46-47 utilize WP's Find feature from the List Files screen. If files are found in that directory, WP automatically displays them in a new file list. The macro uses a {FOR} loop in lines 49-52 to assign the path and name of any found files to variables. Line 55 returns the directory to its original listing before the Find feature was used.

Creating the File List (lines 56-66). The final section of the macro is referenced from line 33 (or line 16 if Cancel is pressed) and is executed after the macro has searched through the original directory and all of its subdirectories. Line 57 exits List Files. If any files were found (line 58), lines 59-64 insert the path and filenames into the second document screen. The variables used to store the filenames are cleared on line 63, and the macro quits on line 66.

! ! !

Using Wildcards

With the DOS wildcards (*) and (?), you can narrow your search for a specific filename pattern. The asterisk (*) represents several characters in the filename; the question mark (?) represents a single character in the filename.

Note: Some of the more advanced uses of DOS wildcards (such as alt?????. and alt*.*) may produce different results when used with the Find feature in List Files (which the DIRSRCH.WPM macro uses) than they will when used to list a directory in List Files or at a DOS prompt.*