

Find String Table of Contents

General:

[Application Synopsis](#)

[Registration Form](#)

[Tool Bar](#)

[Display Window](#)

Search Aids:

[Regular Expressions](#)

[Wild Cards](#)

Dialogs:

[File Filter](#)

[Find String](#)

[Preferences](#)

[Select Files](#)

[Select Files to Search by Starting Directory and File Filter](#)

Application Synopsis

The multiline **Find String** application was built to allow application developers, html writers, system/network administrators, and others to find a key word or phrase in many lines of many text files quickly and efficiently. The application supports DOS file like wild cards or a limited set of UNIX like Regular Expressions in the **Find String**. Application developers can, for example, wire it into the 16/32 bit version of Microsoft C++ visual work bench through its Options/Tools menu to enrich the editing capability by allowing them to find all instances of a key word or phrase in their project. It can also be used stand-alone by developers, html writers, administrators et. al. who need to find a piece of information in one or more lines of many files located in a multiple directories. Once found, by double clicking on the file or file line you can launch the editor of your choice (notepad by default) to manipulate the data. **Find String**, will in addition to launching an editor, supply the line number to the editor if instructed to do so (see Preferences Dialog) and will allow command line parameters if required. For those folks like me, who can not remember where they defined a variable 20 minutes ago, it's essential.

Selecting Files to Search

Find String supports two methods of defining the file set to search. The first method allows you to select as many files from as many directories via an extended file dialog as you want the application to search. This method is supported whether or not you have registered the application. The second method allows you to specify a start directory and file filter. This method is only supported for registered customers. If you have registered the application, you can use both methods to specify the file set to search.

Multiline Searches

Find String supports multiline searches and searching for special characters. The supported special characters are displayed in a list in the lower left corner of the Find String Dialog. Just double click on the special character you wish to search for and it is automatically added to the end of the **Find String Line**. To search for a literal (e.g., \n) use the \\ (e.g., \\n).

Special Characters

Find String supports usage of a set of non-printable characters that may be embedded within the string pattern. These special characters are:

\r\n DOS end-of-line;
\n new line which is the UNIX end-of-line;
\r carriage return;
\t tab;
\v vertical tab;
\f formfeed (also paragraph indicator)
\a bell; and
\b backspace.

Regular Expressions

Find String supports the definition of group/set/character expressions for licensed customers. The supported regular expressions conform closely but not exactly to the UNIX standard. The application *does not* support:

- o nested groups within groups or sets within sets;
- o the beginning (\<) or end of word (\>) constraints;
- o definition of your own range constraints (\{m,n\});
- o definition of substring registers (\n});
- o non-escaped literal brackets [or] within a set;
- o non-escaped literal parentheses (or) (interpreted as group indicators unless escaped

with a \); and

- o iteration characters associated with groups.

DOS file like Wild Cards

Find String supports the definition of the ? and * file like wild cards for licensed customers. The ? means match any single character while the * means match one or more characters until succeeding match characters are encountered (e.g., ab*v) or, if the last character, to the end-of-line (e.g., ab*).

Display Limitation

The application was designed to handle a huge number of files from multiple directories. There is a display limitation that will limit the number of files you can search at one time. The display limitation is 128,000 lines, consequently, if 95 lines are found per file (96 display lines), the file limitation would be about 1,300 files. The application will stop the search operation when the line limitation is reached.

Windows 95 Customers

With Windows 95, I had hoped that the display limitations for list boxes et. al. would no longer be a concern. In fact, I assumed that only to find out that I can only display 32767 lines in a list box. As a result, only 32000 lines are displayed per display frame. To overcome this limitation, I allow you to switch display frames using the Up (↑) and Down (↓) arrow buttons on the tool bar or selecting the **Next/Previous Display Frame** option from the **View Menu**. For those of you who have NT, forget this paragraph.

Find String Remembers

Find String remembers the first directory searched and will default to that directory on the next invocation. Additionally, **Find String** will save up to the last 100 (20 by default) search strings and allow you to reuse those strings on subsequent invocations. For those of you who want a constant start position on your desktop, **Find String** remembers its last size and desktop position and will restore itself to that size and position until you change it. If you position it partially or fully off the desktop, it will restore itself fully on the desktop.

Newline in Files Required

The application expects to find a newline character, as a minimum, at the end of each line and will not process files that are not constructed in this manner. Files generated by word processors are not guaranteed to fall into this category unless they are saved in ASCII/text. When the application encounters such a file, it warns you and excludes that file from any further search. The search string is limited to 1024 characters and the maximum number of characters per searched line is limited to 2048.

Multiple Editor

If you wish to maintain multiple editor capabilities and segregate your remembered find strings, you may pass **Find String** a command line argument specifying the full path to a different *ini* file (unless the *ini* file is in the windows directory and then only the name of the *ini* file is required (e.g., *my.ini*)).

UNIX File Types

In today's networking environment, many files can be manipulated on foreign machines. The UNIX file type is supported automatically for those who need to remotely find a key word or phrase in text files that conform to the UNIX file standard, assuming of course, that you have that capability. UNIX file types are NOT supported automatically when you choose regular expressions. The application must know how to treat the \$ end-of-line character prior to starting the search. When you check the regular expression box, the UNIX File Type box will become active. Check it if you are using the \$ in the expression.

The application is easy and safe to use, just type in or select a previous string, select whether or not you want it to match case and/or be a whole word, select one or more files you want it to search, and Begin!

Registration Form

Your Name: _____

Note: Type/print your name and company below **EXACTLY** as it appears in the **Registration** dialog in the **Help Menu** of the Application.

Registration Dialog Name: _____

Registration Dialog Company: _____

Please type or print clearly the following information:

Address: _____

City: _____ State: _____ Zip: _____

E-Mail Address: _____

Check the programs you wish to Register.

Search/Replace String	_____	\$15.00
Find String	_____	\$15.00
Touch File Date/Time	_____	\$10.00
Both Search/Replace & Find String	_____	\$25.00
All (Search/Replace, Find String, & Touch File Date/Time)	_____	\$30.00

I have enclosed a

Check Money Order Cash (Check One)

Preferred Registration Code delivery:

Mail E-mail (Check One)

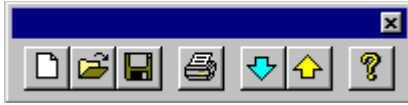
Please include a self addressed envelope for mail delivery to speed up the response.

Please make checks payable to Richard Floersch.








Mail this form and your check or money order to:

Richard Floersch
7307 W 89th Terr
Overland Park, Ks 66212

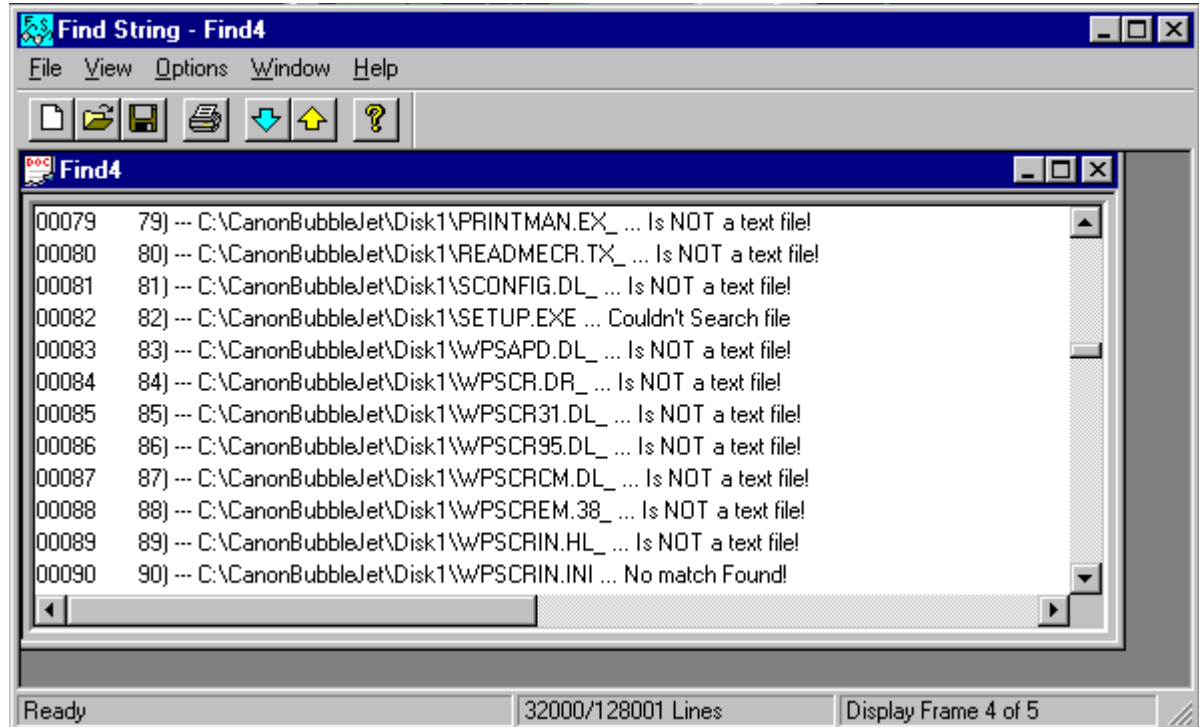
Tool Bar



The tool bar provides a quick means to perform a task. Simply select the appropriate button and click on it. From left to right the buttons are:

-  Start a new find operation.
-  Open a previously saved find or ASCII file.
-  Save the current find operation results to file.
-  Display next page.
-  Display previous page.
-  Print the find operation results for the window currently on top.
-  Get HELP!!


Display Window




The display window illustrated above is composed of an outer main window and inner scroll windows. The strings that match your **Find String** criteria are displayed in the inner scroll windows. Use the scroll bar or Page Up/Down, Cursor Up/Down, Home, or End keys to view the results. One inner window will appear for each new operation you do within a session. The menu selections and tool bar buttons on the outer window apply to the inner window that is on top (has focus).

At the bottom of the outer window is a status area with three panes. These panes will display pertinent information, during and after, the find operation.

Windows 95 Customers

With Windows 95, I had hoped that the display limitations for list boxes et. al. would no longer be a concern. In fact, I assumed that, only to find out that I can only display 32767 lines in a list box. As a result, only 32,000 lines are displayed per display page. To overcome this limitation, you may switch display pages using the **Next Page**  and **Previous Page**

 **Buttons** on the tool bar or selecting the **Next/Previous Display Page** options from the **View Menu**. For those of you who are running NT, you get all the data displayed in one display page.

Regular Expressions

General

Find String supports the definition of group (BOOL|int|void), set [ABCAbc], and single character expressions but only for licensed customers. Regular expressions are a powerful notation for describing string matching patterns.

Supported Characters

The *supported* regular expressions conform, with some variation, to the UNIX standard. The following enumerates the supported characters along with their meaning.

- .** matches any single character except for newline (\n).
- *** matches 0 or more occurrences of a set or character expression.
- +** matches 1 or more occurrences of a set or character expression.
- ?** matches 0 or 1 occurrence of a set or character expression.
- [and]** defines the beginning and end of a character set expression.
- (and)** defines the beginning and end of a group expression.
- defines a range of characters in a set expression.
- |** alteration operator which allow matching substrings on its left or right (restricted to group expressions).
- \$** matches the end-of-line.
- ^** match beginning of line if defined as the 1st character or compliment operator if defined 1st in a set expression.
- ** escape character required when you want to use any of the above characters as a literal character.

Non-Supported Operations

The application does not support:

- o nested groups within groups or sets within sets;
- o the beginning (\<) or end of word (\>) constraints;
- o definition of your own range constraints (\{m,n\});
- o definition of substring registers (\n});
- o non-escaped literal brackets [] within a set;
- o non-escaped literal parentheses () (interpreted as group indicators unless escaped with a \); and
- o iteration characters associated with groups.

Matching a Character

The basic unit of a regular expression is matching a single character. You match characters:

- o literally (using the character itself);
- o ambiguously (using the **.** wild card); or
- o with a character set (**[AEIOUaeiou]**).

Character Sets

A character set is a series of characters enclosed by square brackets **[and]**. It specifies a set of characters any one of which constitutes a match. In the above example, the character set matches any upper or lower case vowel. You may use the **-** operator within a set to specify a range of characters (e.g., **[A-Za-z0-9]** specifies all the letters in the alphabet and the digits 0 through 9). You may use the **^** as the first character in a set to compliment the ensuing character set (e.g., **[^1-36-9]** matches all characters except the numbers 1 through 3 and 6 through 9).

Beginning and End of Line

If you define the **^** as the first character of your expression, you are specifying that whatever characters/expressions follow, must occur at the beginning of the line. Likewise, you may end your expression with a **\$**, which specifies that any preceding characters/expressions must occur at the end of the line for a match to occur.

Iteration Characters

The iteration characters *****, **+**, and **?** may be used to modify a single character or character set. Without an iteration modifier, an expression will match exactly one occurrence of the character or set. As an example, if you wanted to match one or more spaces or tabs at the beginning of a line, your expression would look like this:

^[\t]+

Group Expressions

Group expressions are supported to allow usage of the alteration (**|**) operator. You can think of the alteration (**|**) operator as you would the conjunction or. It allows you to match one or more groups of characters. A group definition begins with (**(** and ends with a matching **)**. If I wanted to find Cowboys or Indians my group expression would look like this:

(Cowboys|Indians)

Character set definitions within groups *are* supported, however, iteration operators for a group are *not* supported.

Compound Expressions

The power of regular expressions comes from stringing together characters, groups, and sets to form a compound expression. For example, if I wanted to find all the view and document function declarations in my application, I would construct the following compound expression:

^[\t]*(void|int|double|BOOL|LONG)[\t]?CFind(View|Doc)::

This expression would find all function declarations for the CFindView and CFindDoc classes beginning the line with or without a space or tab, having return types of void or int or double or BOOL or LONG, followed by zero or one space or tab, and having a double colon following the w or c in CFindView or CFindDoc.

Wild Cards

Find String supports the definition of the **?** and ***** file like wild cards for licensed customers. The usage of these wild cards follows the same principles as their usage with DOS and Windows files. The **?** means match any single character while the ***** means match one or more characters until a succeeding match character is encountered (e.g., **ab*v**) or, if the last character, to the end-of-line (e.g., **ab***). Wild cards are not as powerful as regular expressions but can come in handy from time-to-time.

Usage Examples:

The following examples illustrate usage of wild cards.

Example 1:

Suppose I wanted to find all my application's function declarations. My wild card expression would be:

CFind*::

This declaration would find all occurrences where the characters CFind are embedded within a line, followed by one or more additional unspecified characters and finally containing a double colon after the CFind characters.

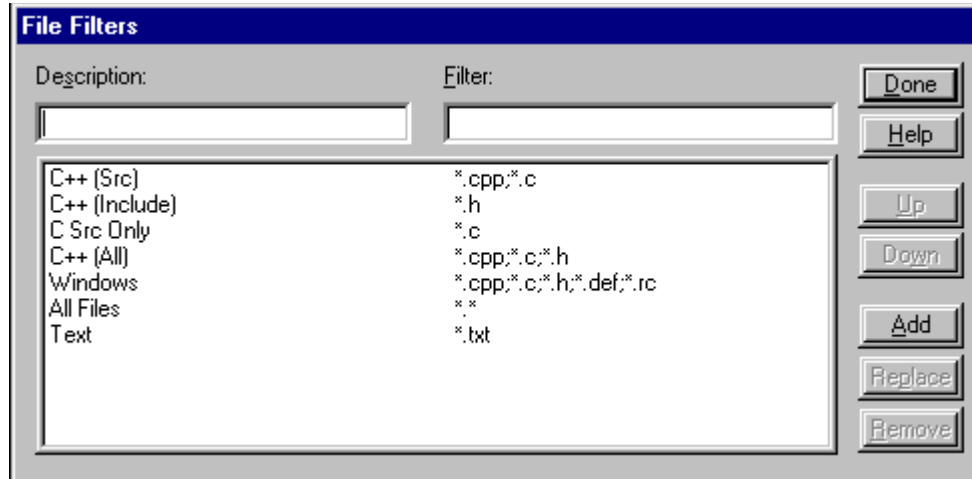
Example 2:

Suppose I wanted to find all my application's function declarations having three characters following my main application class. My wild card expression would be:

CFind***::

This declaration would find all occurrences where the characters CFind are embedded within a line, followed by three additional unspecified characters and finally containing a double colon after the CFind characters.

File Filter Dialog



The **File Filter** dialog allows you to customize the types of files you typically search. It will allow you to add new, replace existing, or remove old file types from your search options. Additionally, by moving a specific filter to the top, you can dictate to **Find String** which files types will appear by default in the [Select Files Dialog](#) when you click the [Select Files](#) button on the [Find String Dialog](#). To add a new file filter, type an appropriate description in the Description control, then move to the Filter control and enter the file type filter appropriate for the description. Alternately, you may double left mouse click an existing filter (which causes that filter to appear in the Description and Filter controls) where you can make changes and either replace the existing filter or make the modified filter a new option. To remove or move a filter, left mouse click on the appropriate filter which will highlight it and then click the appropriate button.

Actions that will result from clicking a button are:

Done: Indicates that you are finished making changes.

Help: That's how you got here.

Up: Causes the selected file filter to take the place of the filter just above it.

Down: Causes the selected file filter to take the place of the filter just below it.

Add: Adds the filter in the edit controls to the bottom of the list.

Replace: Replaces the selected filter in the list box to be replaced by the filter defined in the edit controls.

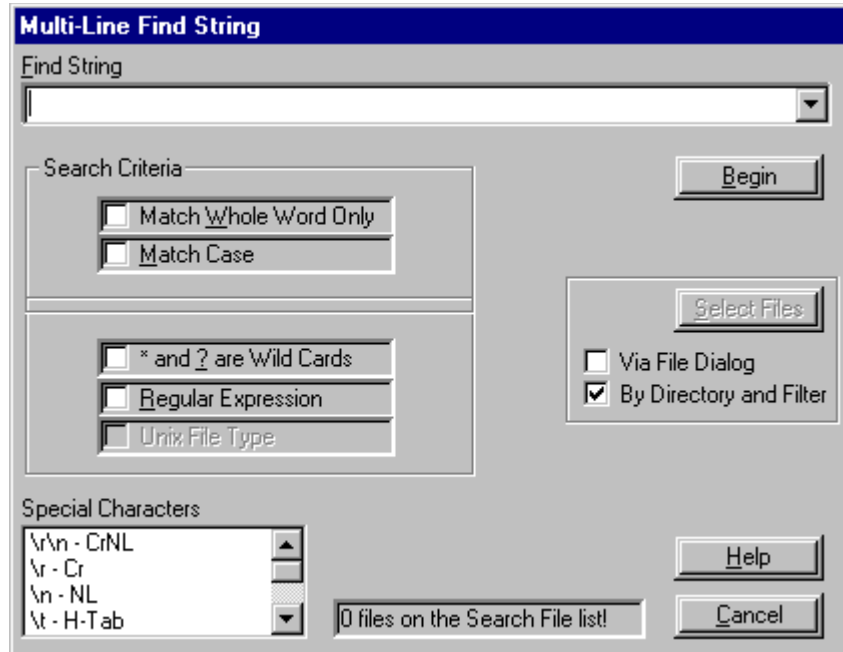
Remove: Causes the highlighted filter in the list box to be deleted.

Filter definition rules:

- Generally the file type description should not be more than 20 characters. Most applications use the common file dialog for display and so does this application although it has been extended. Any characters entered beyond 20 may **NOT** appear in the drop down list;
- You must define at least one File Filter (e.g., *.cpp) per Description (e.g., C++ Source);

- File Filters should contain one or more wild cards (* or ?); and
- Multiple File Filters (e.g., *.txt;*.c;*.htm) are allowed for a single description (e.g., My Std Files) but they must be delimited by ONLY a semicolon (New with Windows 95 since spaces are valid file characters).

Find String Dialog



The **Find String** dialog allows you to type in or select a previously typed in **Find String**. To select a previously typed in string, click the button to the right of the edit control which will cause a drop down list to appear, find the appropriate string, and click on it.

Actions that will result from checking/clicking a box/button are:

Match Case: If you want your search to be case sensitive, make sure you type the find string case explicitly and check the **Match Case** control, otherwise, case is ignored.

Match Whole Word Only: Checking the **Match Whole Word Only** control effectively causes a space to be added before the first character and also after the last character during the search (will not find it if the word/string ends a line).

*** and ? are Wild Cards:** Checking this box causes the application to treat each non-escaped ***** and **?** as a wild card. The **?** means match any single character while the ***** means match one or more characters until succeeding match characters are encountered (e.g., ab*v) or, if the last character, to the end-of-line (e.g., ab*). Wild cards are only supported for licensed customers.

Regular Expressions: **Find String** supports the definition of group/set/character expressions for licensed customers. The supported regular expressions conform to the UNIX standard with some exceptions (see [Regular Expressions](#)). Regular expressions allow you considerable flexibility in defining string matching patterns.

UNIX File Types: The UNIX file type is supported automatically for those who need to remotely find a key word or phrase in text files that conform to the UNIX file standard. UNIX file types are NOT supported automatically when you choose regular expressions. The application must know how to treat the **\$** end-of-line character prior to starting the search. When you check the regular expression box, the **UNIX File Type** box will become active. Check it if you are using the **\$** in the expression.

Special Characters: A number of special characters can be included in the search operations. You may either type in or *select* a special character from the list. All special characters begin with a backslash (\) followed by a single character. To literally search for a backslash if it is followed by one of the special characters, use a double backslash (i.e., \\t instead of \t). Special characters supported are \t-horizontal tab, \f-formfeed, \n-newline, \r-carriage return, \a-bell, \b-backspace, \v-vertical tab, and \r\n-carriage return-newline combination

Via File Dialog: If you wish to select files individually from one or more directory's check this box which will enable the [Select Files](#) button and disable the [Begin](#) button until you have added files to the **Search List**.

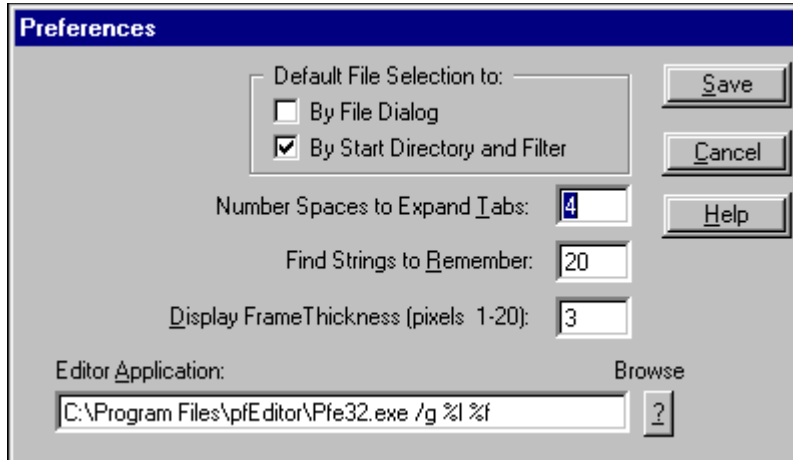
By Directory and Filter: If you wish to let the application automatically select the files to add to the **Search List**, check this box. When checked, the [Select Files](#) button is disabled and the [Begin](#) button is enabled. When you select the [Begin](#) button, you will get a **Select Files to Search by Starting Directory and Filter** dialog that lets you specify (or select) the start directory and lets you choose a file filter from those you have defined under the **Options/File Filters** menu. The first directory searched on the previous invocation will be the default starting directory. **Be aware that the Start button on the Select Files to Search by Starting Directory and Filter dialog is disabled for non-registered customers. Your only option is to Cancel which will result in no search since no files are added to the list.**

Select Files: If you have checked the [Via File Dialog](#) box, this button will become active. Click the [Select Files](#) button to display the extended [Select Files Dialog](#) that lets you individually select the files you wish to include in the search. The first directory searched on the previous invocation will be the default starting directory. You may include files from multiple directories in the operation. To do so, select the appropriate files and press the [Add](#) button to add them to the list. Change directories and repeat the process until you are done. You need not add the last directory's files, just click the [OK](#) button after making your selections. The [Cancel](#) button does **NOT** cancel any file additions. You must remove any unwanted files from the search list by selecting them and clicking the [Remove](#) button

Begin: This button will be active if you have checked the [By Directory and Filter](#) box or the [Via File Dialog](#) box and selected at least on file. This button will begin the Search operation. Before selecting the [Begin](#) button you must specify a search string and select the various available options. If the [By Directory and Filter](#) box is checked at the time you select the [Begin](#) button, the [Select Files to Search by Starting Directory and File Filter Dialog](#) will be displayed that lets you enter/select a starting directory and specify a file filter.

Aborting a Search: While the application is searching files, you may abort by double clicking the left mouse button and a displayed line in the active window.

Preferences Dialog



By File Dialog: Check this box if you want the **Via File Dialog** box on **Find String Dialog** to be checked by default:

By Start Directory and Filter: Check this box if you want the **By Start Directory and Filter** box on **Find String Dialog** to be checked by default:

Tab Characters: When a tab character is encountered in a line being displayed, it appears as a non-printable character (square box). Consequently, tab characters are expanded using spaces to preserve appearance. You may define how many spaces will be displayed in lieu of the tab by entering the number of your choice between 1 and 24. The default is 4.

Remember Find Strings: You define the number of **Find Strings** you want the application to remember from previous invocations. The default is 20 and the maximum is 100. This is handy if you find yourself constantly searching for the same string fragment over and over or would rather edit just a few characters from similar previous strings rather than typing in the new fragment.

Display Frame Thickness You can dictate to the application how thick you want the display frame enclosing the output to be. The minimum is 1 pixel and the maximum is 20 pixels. It is set by default at 5.

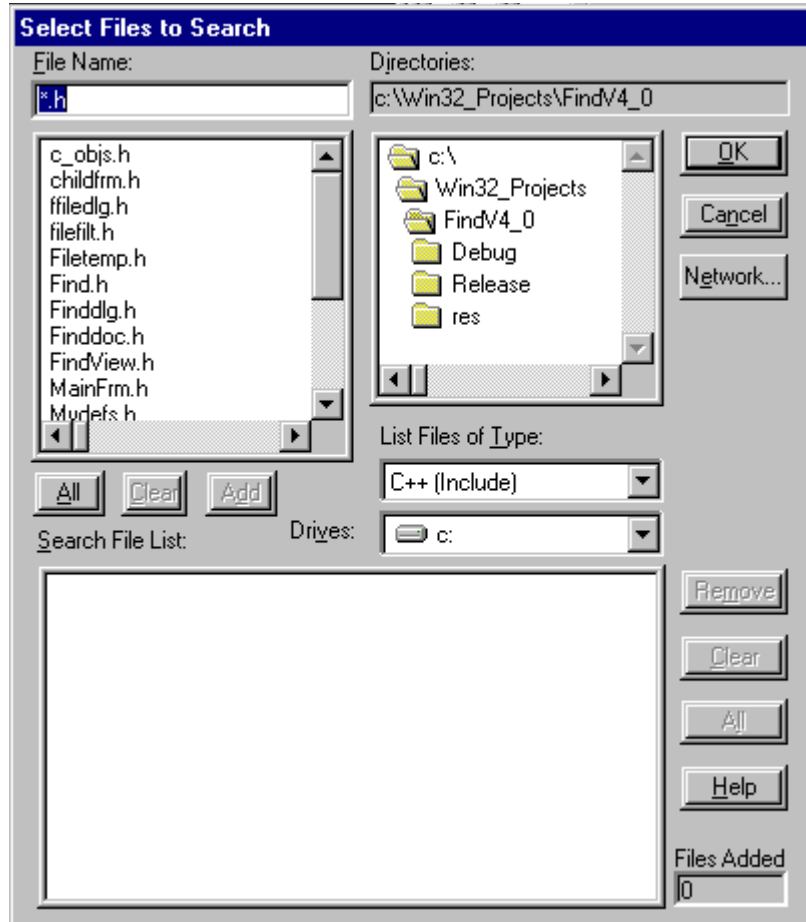
Editor Application: Double clicking on a file or found file line causes **Find String** to launch an editor. By default the editor is notepad. You may change the editor to the one of your choice by typing in the full path and file name or pressing the browse button, finding the editor on your hard drive, and selecting that editor. Two tokens are recognized in the editor application line. The tokens are **%f** and **%l** (f & l are both lower case letters) representing the file name and line number respectively. You may use the **%f** token to tell **Find String** where to place the file name if you need to specify additional command line parameters. By default, the file name is placed after the editor application string. Your editor may allow you to specify the line number where it will position the cursor when started. If so, you can take advantage of that option by specifying both the **%f** and **%l** tokens. **Find String** will supply that line number if you double click on a line it found. The following are examples of how the Editor Application line may be configured.

- c:\windows\editor (in this case line numbers are not supported)
- c:\windows\editor %f /g /r (in this case the command line switches follow the file name)

- c:\windows\editor %f %l (in this case no switch is required to specify go to line number)
- c:\window\editor %f /l %l (in this case a /l tells the editor a line number follows)
- c:\window\editor /G %l /N %f (in this case a /G tells the editor a line number follows)

If you specify a %l in the line and do not double click on a found line, **Find String** will supply a one for the line number, consequently, you will start at the beginning of the file.

Select Files Dialog



You may search for a string or string fragment in files located in multiple directories. When you have selected the file names in the current directory that you want to search, click the **Add** button to add them to the **Search File List**. Change to the next directory and repeat the process. If you are searching files in a single directory, simply select the ones you want and click **OK**. Clicking the **OK** button automatically adds any selected files to the **Search File List**, consequently, you need NOT add the files by clicking the **Add** button when you are in your last directory. If you do inadvertently click the **Add** button in the last directory, click the **OK** or **Cancel** button to terminate the **Select Files Dialog**. **Be aware that once files have been added to the Search List they will remain on the list until removed by you, the Find operation is initiated, or the Find operation is canceled.**

The buttons to the right of the **Search File List** do the following.

Remove: Removes any selected files from the Search File List.

Clear: Deselects all files on the Search File List that are selected.

All/Invert: Selects all files in the Search File List if nothing is selected. Once one or more search files are selected, the button will change to **Invert** which deselects files currently selected and selects files that were previously deselected.

Help: Got you here!

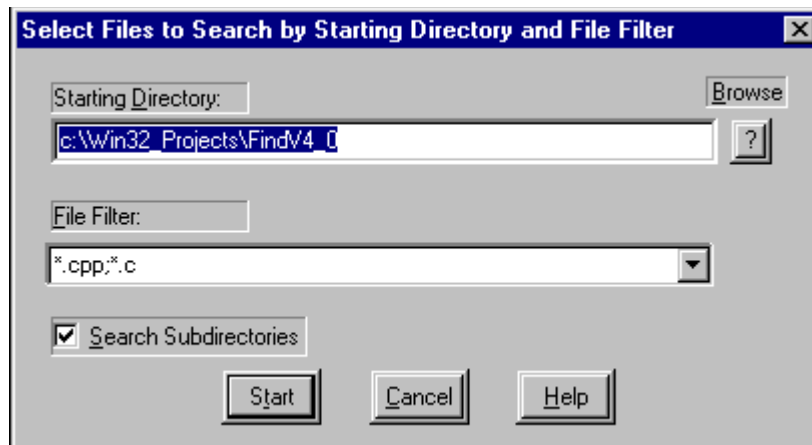
The buttons below the **File Names List** do the following.

Add: Adds all the files selected in the file name list to the Search File List.

All/Invert: Selects all files in the File Name List if nothing is selected. Once one or more file names are selected, the button will change to **Invert** which deselects files currently selected and selects files that were previously deselected.

Clear: Deselects all file names that are selected.

Select Files to Search by Starting Directory and File Filter Dialog



If the **By Directory and Filter** check box is checked when the **Begin** button is clicked, you are allowed to add additional files to the **Search List** or all the files you want searched to the **Search List**. This dialog allows you to specify a starting directory and file filter to use for automatic file addition. If you check the **Search Subdirectories** box, all subdirectories below the start directory are also searched for files to add. Only files meeting one or more of the file filter specifications are added.

Starting Directory:

The first directory searched on previous operations will be the default directory listed when this dialog appears. You may enter a new starting directory or select the **Browse** button to invoke the **Select Start Directory** dialog which allows you to navigate the available drives on your system to find and select the start directory.

Browse:

Invokes a the **Select Start Directory** dialog which allows you to navigate the available drives on your system to find and select the start directory.

File Filter:

By default the last file filter you used will be displayed in this control. To change the filter to one you have defined under **Options/File Filters**, click the drop down button to the right of the field and select the new filter. You are NOT restricted to file filters on that list. You may define, from scratch, one or more filters by entering their wildcard based specifications or you may extend an existing filter by the entering additional filter specifications, separated by semicolons (no spaces) anywhere it is appropriate in the filter field.

Search Subdirectories:

Check this box when you want all subdirectories below the start directory searched for files meeting the file filter specifications.

Start:

When you are ready to begin, click the this button. **Note:** The **Start** button is disabled for non-registered customers.

