Findstr

Aggiornamento: settembre 2007

Si applica a: Windows Server 2008, Windows Vista

Searches for patterns of text in files.

For examples of how to use this command, see Examples.

Syntax

```
findstr [/b] [/e] [/l | /r] [/s] [/i] [/x] [/v] [/n] [/m] [/o] [/p] [/f:<File>]
[/c:<String>] [/g:<File>] [/d:<DirList>] [/a:<ColorAttribute>] [/off[line]] <Strings>
[<Drive>:][<Path>]<FileName>[ ...]
```

Parameters

Parameter	Description
/b	Matches the text pattern if it is at the beginning of a line.
/e	Matches the text pattern if it is at the end of a line.
/l	Processes search strings literally.
/r	Processes search strings as regular expressions. This is the default setting.
/s	Searches the current directory and all subdirectories.
/i	Ignores the case of the characters when searching for the string.
/x	Prints lines that match exactly.
/v	Prints only lines that do not contain a match.
/n	Prints the line number of each line that matches.
/m	Prints only the file name if a file contains a match.
/o	Prints character offset before each matching line.
/p	Skips files with non-printable characters.
/off[line]	Does not skip files that have the offline attribute set.

/f:<File>	Gets a file list from the specified file.
/c:<String>	Uses the specified text as a literal search string.
/g:<File>	Gets search strings from the specified file.
/d:<DirList>	Searches the specified list of directories. Each directory must be separated with a semicolon (;), for example `dir1;dir2;dir3`.
/a:<ColorAttribute>	Specifies color attributes with two hexadecimal digits. Type `color /?` for additional information.
<Strings>	Specifies the text to search for in *FileName*. Required.
[<Drive>:][<Path>] <FileName>[...]	Specifies the location and file or files to search. At least one file name is required.
/?	Displays Help at the command prompt.

Remarks

- All **findstr** command-line options must precede *Strings* and *FileName* in the command string.
- Regular expressions use both literal characters and metacharacters to find patterns of text, rather than exact strings of characters. A literal character is a character that does not have a special meaning in the regular-expression syntax—it matches an occurrence of that character. For example, letters and numbers are literal characters. A metacharacter is a symbol with special meaning (an operator or delimiter) in the regular-expression syntax.

  The following table lists the metacharacters that **findstr** accepts.

Metacharacter	Value
.	Wildcard: any character
*	Repeat: zero or more occurrences of the previous character or class
^	Line position: beginning of the line
$	Line position: end of the line
[class]	Character class: any one character in a set
[^class]	Inverse class: any one character not in a set
[x-y]	Range: any characters within the specified range
\x	Escape: literal use of a metacharacter x

\<string	Word position: beginning of the word
string\>	Word position: end of the word

The special characters in regular expression syntax have the most power when you use them together. For example, use the following combination of the wildcard character (.) and repeat (*) character to match any string of characters:

```
.*
```

Use the following expression as part of a larger expression to match any string beginning with "b" and ending with "ing":

```
b.*ing
```

Examples

Use spaces to separate multiple search strings unless the argument is prefixed with **/c**.

To search for "hello" or "there" in file x.y, type:

```
findstr "hello there" x.y
```

To search for "hello there" in file x.y, type:

```
findstr /c:"hello there" x.y
```

To find all occurrences of the word "Windows" (with an initial capital letter W) in the file Proposal.txt, type:

```
findstr Windows proposal.txt
```

To search every file in the current directory and all subdirectories that contained the word Windows, regardless of the letter case, type:

```
findstr /s /i Windows *.*
```

To find all occurrences of lines that begin with "FOR" and are preceded by zero or more spaces (as in a computer program loop), and to display the line number where each occurrence is found, type:

```
findstr /b /n /r /c:"^ *FOR" *.bas
```

To search for multiple strings in a set of files, create a text file that contains each search criterion on a separate line. You can also list the exact files that you want to search in a text file. For example, to use the search criteria in the file Stringlist.txt, search the files listed in Filelist.txt, and then store the results in the file Results.out, type:

```
findstr /g:stringlist.txt /f:filelist.txt > results.out
```

To list every file containing the word "computer" within the current directory and all subdirectories, regardless of case, type:

```
findstr /s /i /m "\<computer\>" *.*
```

To list every file containing the word "computer" and any other words that begin with "comp", (such as "compliment" and "compete"), type:

```
findstr /s /i /m "\<comp.*" *.*
```

Additional references

Command-Line Syntax Key

Aggiunte alla community