Wildcard

Updated: 12/31/2022 by Computer Hope

Alternatively called a **wild character** or **wildcard character**, a **wildcard** is a symbol used to replace or represent one or more characters. The most common wildcards are the asterisk (*), which represents one or more characters, and the question mark (?), which represents a single character. In the examples below of how a wildcard may be used, realize that wildcards are relatively universal but may not work in every program.

Wildcard basics

Percent (%) in a wildcard

The percent symbol is used in SQL (Structured Query Language) to match any character (including an underscore) zero or more times.

Asterisk (*) in a wildcard

The asterisk in a wildcard matches any character zero or more times. For example, "comp*" matches anything beginning with "comp," which means "comp," "complete," and "computer" are all matched.

Question mark (?) in a wildcard

A question mark matches a single character once. For example, "c?mp" matches "camp" and "comp." The question mark can also be used more than once. For example, "c??p" would match the above examples. In MS-DOS and the Windows command line, the question mark can also match any trailing question mark zero or one time. For example, "co??" would match all of the above matches, but because they are trailing question marks, it would also match "cop" even though it's not four characters.

Tip: With regular expressions, a period (.) is a wildcard for a single character.

Open and close brackets ([ ]) in a wildcard

With Unix shells, Windows PowerShell, and programming languages that support regular expressions, the open and close bracket wildcards match a single character in a range. For example, [a-z] matches any character "a" through "z," which means anything not in that range, like a number, would not be matched.

Tip: Adding an exclamation mark in places that support the brackets as a wildcard tells the program NOT to match.

MS-DOS and Windows command line wildcard examples

```
dir c?mp
```

List files in MS-DOS using the dir command containing c, mp, and any other character between. For example, comp, camp, c2mp, and c-mp would all be matched.

```
dir *.mp3
```

In this next example, the dir command would only list files that end with .MP3 file extension.

```
dir *data
```

List any file that ends with data using the dir command. For example, the files "appdata," "mydata," and "123data" would all be matched.

```
dir he??.*
```

List any file four characters long that begins with he, and has any extension. For example, help.txt, help.mp3, and heck.jpg would all be matched.

```
rename *.txt *.jpg
```

Rename all files in the current directory that end with the file extension .txt to .jpg. For example, the file test.txt would become test.jpg.

```
del comp*.txt
```

Deleting files in MS-DOS that begin with comp and end with a ".txt" extension.

Find and replace using wildcard examples

Find and replace features that support wildcards, like Microsoft Word, that allow searches to contain wildcards. Below are examples of how to use wildcards in Find and Replace. Remember that for any of these to work, you must have the **Use wildcards** option checked in Find and Replace.

```
comp*r
```

Match anything starting with "comp" and ending with "r." In other words, this would find "computer" and "compiler" in your document. However, remember that "*" is greedy, meaning everything is matched up to "r." In other words, if there's an "r" anywhere after comp, it's matched. So, "computer your" is matched since it begins with "comp" and "your" ends with "r."

```
d[eo]ll
```

Using brackets, indicate to Microsoft Word to look for any of the letters contained in the brackets. In this example, "e" or "o" are matched, so find would match either "dell" or "doll."

```
d[o-u]ll
```

The brackets can also be used to search for a range of characters. In the above example, this range includes the letters from "o" to "u." This range matches words like "doll" and "dull" in your document.

```
d[!e]ll
```

Using an exclamation mark in the brackets tells the Find not to match any of the characters in the bracket. In the above example, this wildcard tells the Find not to match "dell," but to match anything else beginning with "d" and ending in 'll.'

```
d?ll
```

The question mark only matches one character. In the above example, this would match "dall," "dell," "dill," "doll," and "dull" since they contain a "d" at the first and "ll" at the end.

```
se{2}d
```

Using a curly bracket in your *Find*, look for the number of characters preceding the brackets. In the above example, Find matches "seed," but not match "sed."

```
<(comp)
```

A find starting with a less than and containing text in parentheses tells Find to look for any word beginning with whatever is contained in the parentheses. In the example above, this would find any words beginning with "comp."

```
(er)>
```

A string starting with characters in a parenthesis and ending a greater than tells Find to look for any word ending with whatever is contained in the parentheses. In the example above, this would find any words ending with "er."

Linux and Unix wildcard examples

```
ls comp*
```

This command uses the ls command to list all files and directories in the working directory that begin with the letters "comp" in a Linux variant.

```
rm c?mp
```

Deleting files using the rm command in a Linux variant containing c, mp, and any character between.

Microsoft Excel wildcard examples

```
=SUMIF(A1:A6,"*",B1:B6)
```

Excel formula to search for any character using the * wildcard in cells B1 through B6, and if found, use SUM to add all values between A1 and A6.

Finding text containing a wildcard character

When performing a find or search, you may not want a wildcard character (.*?) to be treated as a wildcard. In this situation, you want the character to be treated as a literal character. For example, in Excel, to find cells containing a question mark, you would look for "~?" and not "?" in the *Find what* box. When the tilde is added before the wildcard in Excel, it treats it as a literal character, not a wildcard. In other programs and programming languages, you would escape the character.

© 2024 Computer Hope