WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools

What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export

Create a book
Download as PDF
Printable version

Languages

한국어
Italiano
日本語
Polski
Русский
Edit links

glob (programming)

From Wikipedia, the free encyclopedia

In computer programming, **glob** patterns specify sets of filenames with wildcard characters. For example, the Unix command `mv *.txt textfiles/` moves (`mv`) all files with names ending in `.txt` from the current directory to the directory `textfiles` . Here, `*` is a wildcard standing for "any string of characters" and `*.txt` is a glob pattern. The other common wildcard is the question mark (`?`), which stands for one character.

Contents

Origin [edit]

The command interpreters of the early versions of Unix (1st through 6th Editions, 1969–75) relied on a separate program to expand wildcard characters in unquoted arguments to a command: */etc/glob*.[1] That program performed the expansion and supplied the expanded list of file paths to the command for execution. Its name is an abbreviation for "global command".[2] Later, this functionality was provided as a library function, **glob()**, used by programs such as the shell.

Syntax [edit]

The most common wildcards are `*` , `?` , and `[…]` .

Wildcard	Description	Example	Matches	Does not match
`*`	matches any number of any characters including none	`Law*`	`Law` , `Laws` , or `Lawyer`	`GrokLaw` , `La` , or `aw`
		`*Law*`	`Law` , `GrokLaw` , or `Lawyer` .	`La` , or `aw`
`?`	matches any single character	`?at`	`Cat` , `cat` , `Bat` or `bat`	`at`
`[abc]`	matches one character given in the bracket	`[CB]at`	`Cat` or `Bat`	`cat` or `bat`
`[a-z]`	matches one character from the (locale-dependent) range given in the bracket	`Letter[0-9]`	`Letter0` , `Letter1` , `Letter2` up to `Letter9`	`Letters` , `Letter` or `Letter10`

In all cases the path separator character (`/` on Unix or `\` on Windows) will never be matched.

Unix [edit]

On Linux and POSIX systems `*` , `?` is defined as above while `[…]` has two additional meanings:[3][4]

Wildcard	Description	Example	Matches	Does not match
`[!abc]`	matches one character that is not given in the bracket	`[!C]at`	`Bat`, `bat`, or `cat`	`Cat`
`[!a-z]`	matches one character that is not from the range given in the bracket	`Letter[!3-5]`	`Letter1`, `Letter2`, `Letter6` up to `Letter9` and `Letterx` etc.	`Letter3`, `Letter4`, `Letter5` or `Letterxx`

Some shells (such as the C shell and Bash) support additional syntax known as alternation or brace expansion.

The Bash shell also supports **Extended Globbing** which allows other pattern matching operators to be used to match multiple occurrences of a pattern enclosed in parentheses. It can be enabled by setting the `extglob` shell option.[5]

Windows PowerShell [edit]

Windows PowerShell has all the common syntax defined as stated above without any additions.[6]

DOS COMMAND.COM and Windows cmd.exe [edit]

COMMAND.COM and cmd.exe have most of the common syntax with some limitations: There is no `[…]` and for COMMAND.COM the `*` may only appear at the end of the pattern, not at the beginning.

SQL [edit]

The SQL LIKE operator has an equivalent of `?` and `*`. There is no equivalent of `[…]`.

Common wildcard	SQL wildcard
`?`	`_`
`*`	`%`

Standard SQL uses a glob-like syntax for simple string matching in its `LIKE` operator. The percent sign (%) matches zero or more characters, and the underscore matches exactly one character. The term "glob" is not generally used in the SQL community, however. Many implementations of SQL have extended the `LIKE` operator to allow a richer pattern-matching language incorporating elements of regular expressions.

Some proprietary extensions such as Transact-SQL provide the `[…]` functionality, e.g., `[`*`characters`*`]` and `[^`*`characters`*`]`.[7]

Compared to regular expressions [edit]

Globs do not include syntax for the Kleene star which allows multiple repetitions of the preceding part of the expression; thus they are not considered regular expressions, which can describe the full set of regular languages over any given finite alphabet.[*citation needed*]

Common wildcard	Equivalent regular expression
`?`	`.`
`*`	`.*`

Globs attempt to match the entire string (for example, `S*.DOC` matches S.DOC and SA.DOC, but not POST.DOC or SURREY.DOCKS), whereas regular expressions match a substring unless the expression is enclosed with `^` and `$` (so the equivalent of `S*.DOC` is `^S.*\.DOC$` [8]).

Implementations [edit]

Unix shells such as Bash, tcsh, and zsh provide globbing on filenames at the command line and in shell scripts.[9]

The Windows command interpreter cmd.exe relies on a runtime function in applications to perform globbing.[10][11] Windows PowerShell Cmdlets support globbing.[12]

The term "glob" is also used to refer more generally to limited pattern-matching facilities of this kind, in other contexts:

- D has a `globMatch` function in the `std.path` module.[13]
- NodeJS has a library called `minimatch` which is used internally by npm, and `micromatch`, a purportedly more optimized, accurate and safer globbing implementation.[14][15][16][17]
- Go has a `Glob` function in the `filepath` package.[18]
- Java has a `Files` class containing methods that operate on glob patterns.[19]
- Haskell has a `Glob` package with the main module `System.FilePath.Glob`. The pattern syntax is based on a subset of Zsh's. It tries to optimize the given pattern and should be noticeably faster than a naïve character-by-character matcher.[20]
- Perl has both a `glob` function (as discussed in Larry Wall's book *Programming Perl*) and a *Glob* extension which mimics the BSD glob routine.[21] Perl's angle brackets can be used to glob as well: `<*.log>`.
- PHP has a `glob` function.[22]
- Python has a `glob` module in the standard library which performs wildcard pattern matching on filenames,[23] and an `fnmatch` module with functions for matching strings or filtering lists based on these same wildcard patterns[24] Guido van Rossum, author of the Python programming language, wrote and contributed a `glob` routine to BSD Unix in 1986.[25] There were previous implementations of `glob`, e.g., in the ex and ftp programs in previous releases of BSD.
- Ruby has a `glob` method for the `Dir` class which performs wildcard pattern matching on filenames.[26] Several libraries such as Rant and Rake provide a `FileList` class which has a glob method or use the method `FileList.[]` identically.
- SQLite has a `GLOB` function.
- Tcl contains both true regular expression matching facilities and a more limited kind of pattern matching often described as globbing.[27]

See also [edit]

- Regular expression
- Wildcard character
- Matching wildcards

References [edit]

1. ^ "First Edition Unix manual 'Miscellaneous' section (PDF)" (PDF). Archived from the original (PDF) on 2000-08-29. Retrieved 2011-05-11.
2. ^ *1st Edition UNIX*, code.google.com, src/cmd/glob.c, archived from the original on 2013-05-13
3. ^ "The Open Group Base Specifications Issue 7 IEEE Std 1003.1, 2013 Edition, 2.13. Pattern Matching Notation".
4. ^ "Linux Programmer's Manual, GLOB(7)".
5. ^ "Pattern Matching". *Bash Reference Manual*.
6. ^ "Supporting Wildcard Characters in Cmdlet Parameters". *MSDN*.
7. ^ "LIKE (Transact-SQL)".
8. ^ Strictly, `.` does not match a newline. To match newlines, the equivalents are `[\s\S]` and `[\s\S]*` or similar complementary pairs, respectively.
9. ^ The "Advanced Bash-Scripting Guide, Chapter 19.2: Globbing" (Mendel Cooper, 2003) has a concise set of examples of filename globbing patterns.
10. ^ "Wildcard Expansion". Microsoft Developer Network. 2013.
11. ^ "Expanding Wildcard Arguments". Microsoft Developer Network. 2013.
12. ^ "Supporting Wildcard Characters in Cmdlet Parameters". Microsoft Developer Network. 2013.
13. ^ "std.path - D Programming Language - Digital Mars". dlang.org. Retrieved 2014-09-08.
14. ^ "isaacs/minimatch". *GitHub*. Retrieved 2016-08-10.
15. ^ "minimatch". *npm*. Retrieved 2016-08-10.
16. ^ "jonschlinkert/micromatch". *GitHub*. Retrieved 2017-04-04.
17. ^ "micromatch". *npm*. Retrieved 2017-04-04.
18. ^ "Package filepath - The Go Programming Language". Golang.org. Retrieved 2011-05-11.
19. ^ "File Operations". Oracle. Retrieved 2013-12-16.
20. ^ "Glob-0.7.4: Globbing library". Retrieved 2014-05-07.
21. ^ Contact details. "File::Glob - Perl extension for BSD glob routine". perldoc.perl.org. Retrieved 2011-05-11.
22. ^ "glob - Manual". PHP. 2011-05-06. Retrieved 2011-05-11.

22. "glob - Manual". PHP. 2011-05-06. Retrieved 2011-05-11.
23. ^ "10.7. glob — Unix style pathname pattern expansion — Python v2.7.1 documentation". Docs.python.org. Retrieved 2011-05-11.
24. ^ "10.8 fnmatch Unix filename pattern matching -- Python v2.7.7 documentation". Docs.python.org. Retrieved 2014-06-28.
25. ^ " 'Globbing' library routine". Archived from the original on 2007-12-19. Retrieved 2011-05-11.
26. ^ "Class: Dir". Ruby-doc.org. Retrieved 2011-05-11.
27. ^ "TCL glob manual page". Retrieved 16 November 2011.

Categories: C POSIX library | Pattern matching | Unix programming tools

This page was last edited on 17 May 2018, at 06:25.