Vergleich Shell-Globbing mit Regulären Ausdrücken

(C) 2008-2024 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>
OSTC Open Source Training and Consulting GmbH
http://www.ostc.de

$Id: shell-glob-regex-cmp-HOWTO.txt,v 1.14 2025/02/18 10:08:31 tsbirn Exp $

Dieses Dokument beschreibt die Gemeinsamkeiten und Unterschiede des "Shell-Globbing" (d.h. der Shell-Muster zur Dateinamen-Expansion) und der "Regulären Ausdrücke" (d.h. der Muster zur Texterkennung und -ersetzung in Programmen).

INHALTSVERZEICHNIS

1) Einführung
2) Vergleichstabelle
3) Besonderheiten
4) Beispiele

---

1) Einführung

Obwohl die Sonderzeichen beim Shell-Globbing und in Regulären Ausdrücken teilweise identisch sind, gibt es doch eine Reihe von Unterschieden, die leicht zu Verwechslungen führen.

Shell-Globbing ist "schwächer" in seinen Ausdrucksmöglichkeiten als Reguläre Ausdrücke. Der größte Unterschied ist die "automatische Verankerung" bei den Shell-Mustern (Std: Dateiname muss VOLLSTÄNDIG zum Suchmuster passen), die bei den Reguläre Ausdrücken durch "^" und "$" erzwungen werden muss (Std: nur TEIL eines Textes muss zum Suchmuster passen).

Die Unterschiede zwischen den beiden Verfahren existieren aus pragmatischen Gründen:

  * Shell-Globbing bezieht sich auf Dateinamen
    --> Arbeitsgeschwindigkeit auf der Shell-Kommandozeile soll hoch sein,
        d.h. häufig benötigte Aufgaben einfach + schnell einzutippen sein.

  * Reguläre Audrücke beziehen sich auf Textzeilen/Zeichenketten/Datensätze
    --> Sollen möglichst leistungsfähig sein,
        daher dürfen sie auch komplizierter + langsamer einzutippen sein.

2) Vergleichstabelle

	Shell ("glob")	RegExp /.../
Verankerung	Automatisch	Nicht automatisch
GROSS/kleinscheibung	Betriebssystemabhängig	Automatisch
GROSS/kl. ignorieren	Betriebssystemabhängig	/.../i oder -i
Quotierung	\ "..." '...'	\ (NUR!)
Zeilenanfang (Anker)	Automatisch	^
Zeilenende (Anker)	Automatisch	$
1 belieb. Zeichen	? (außer "/")	. (außer "\n")
Bel. viele bel. Z.	* (außer "/")	.* (außer "\n")
Analog ALLE Zeichen	** (inkl. "/")	[\s\S]* (inkl. "/n")
Zeichenwiederh. 0-N	*(...|...)	x*

Zeichenwiederh. 1-N	+(...|...)	x+
Zeichenwiederh. M-N	Nicht möglich	x{M,N}
Optional (0-1)	?(...|...)	x?
Ein Element in Liste	@(...|...)	Nicht möglich
Alle außer einem Elem.	!(...|...)	Nicht möglich
Zeichenklasse	[abc]	[abc]
Zeichenbereich	[a-z]	[a-z]
Neg. Zeichenklasse	[!abc] [^abc]	[^abc]
ZKl. wiederh. (0-N)	Nicht möglich	[abc]*
ZKl. wiederh. (1-N)	Nicht möglich	[abc]+
ZKl. wiederh. (M-N)	Nicht möglich	[abc]{M,N}
ZKl. optional (0-1)	Nicht möglich	[abc]?
Alternative/Oder	{M1,M2,...}	M1|M2|...
Klammern	Nicht möglich	(...)

Metazeichen als normale Zeichen behandeln:

	Shell ("glob")	RegExp /.../
Stern	*	*
Fragezeichen	\?	\?
Punkt	.	\.
Caret	^	\^
Dollarzeichen	\$	\$
Leerzeichen	\ oder " " oder ' '	" " oder [ ]
Eckige Klammern	\[\]	\[\]

Vollständiger Match notwendig bei Shell automatisch, bei Regex manuell:

	Shell ("glob")	RegExp /.../
Keine Verankerung	*abc*	abc
Links verankert	abc*	^abc
Rechts verankert	*abc	abc$
Voll verankert	abc	^abc$

3) Besonderheiten

Das LEERZEICHEN ist beim Shell-Globbing ein Sonderzeichen, um mehrere Muster zu trennen. Soll es Teil des Dateinamens sein, muss es mit "\" oder "..." oder '...' geschützt werden.

In Regulären Ausdrücken ist das Leerzeichen ein ganz normales Zeichen (außer im Extended/Verbose-Modus /.../x).

Ein "." am Dateinamen-Anfang wird beim Shell-Globbing besonders behandelt, indem er von "?" und "*" nicht gefunden wird (versteckte Dateien).

Das Zeichen "/" (Verzeichnistrenner) wird beim Shell-Globbing von "*" und "?" nicht gematcht (nur von "**").

Das Zeichen "\n" (Zeilenvorschub) wird bei Regulären Ausdrücken von "." nicht gematcht (durch Tricks wie "\n", "[\s\S]" oder Option "s" (single line) in PCRE aber dennoch möglich).

4) Beispiele

* Beispiele für Regex-Muster und ihre Treffer (*) bzw. Nicht-Treffer (-)
  (Strings, Zeilen)

	/a/	/./	/[abc]/	/[a-z]/	/[^a-z]/	/^[a-z]/	/[a-z]$/	/^[a-z]$/
d	-	*	-	*	-	*	*	*
made	*	*	*	*	-	*	*	-
abc	*	*	*	*	-	*	*	-
aaaa	*	*	*	*	-	*	*	-
xyz	-	*	-	*	-	*	*	-
""	-	-	-	-	-	-	-	-
" "	-	*	-	-	*	-	-	-
" "	-	*	-	-	*	-	-	-
*	-	*	-	-	*	-	-	-

* Beispiel für Shell-Muster und ihre Treffer (*) bzw. Nicht-Treffer (-)
  (Dateinamen, glob)

	a	*?*	*[abc]*	*[a-z]*	*[!a-z]*	[a-z]*	*[a-z]	[a-z]
d	-	*	-	*	-	*	*	*
made	*	*	*	*	-	*	*	-
abc	*	*	*	*	-	*	*	-
aaaa	*	*	*	*	-	*	*	-
xyz	-	*	-	*	-	*	*	-
""	-	-	-	-	-	-	-	-
" "	-	*	-	-	*	-	-	-
" "	-	*	-	-	*	-	-	-
*	-	*	-	-	*	-	-	-