HOWTO zum Shell-Globbing

(C) 2006-2017 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>
OSTC Open Source Training and Consulting GmbH
http://www.ostc.de

$Id: shell-globbing-HOWTO.txt,v 1.12 2017/08/07 15:49:28 tsbirn Exp $

Dieses Dokument beschreibt die Shell-Dateinamen-Expansion ("Globbing").

INHALTSVERZEICHNIS

---

1) Globbing

Um an ein Kommando eine Liste von Dateinamen zur Verarbeitung zu übergeben, können diese vollständig angegeben werden. Oder man gibt ein oder mehrere "Suchmuster" (Pattern) an, die von der aktuellen Shell in passende Dateinamen umgewandelt werden.

Die Shell expandiert zuerst alle Suchmuster zu einer Liste von passenden Dateinamen ("filename globbing"), setzt diese dann anstelle der Suchmuster ein und führt schließlich das Kommando aus (d.h. das Kommando "sieht" die Suchmuster nicht!).

Unabhängig vom auszuführenden Kommando wird die Dateinamenexpansion immer von der Shell durchgeführt, BEVOR ein Kommando gestartet wird. Außer der Shell befasst sich also kein Kommando mit dem Aspekt der Dateinamenexpansion, diese Funktionalität ist somit EINHEITLICH für alle Kommandos in der Shell realisiert.

2) Syntax

Folgende "Metazeichen" können zur Angabe von Suchmustern verwendet werden:

Metazeichen	Bedeutung
/	Verzeichnistrenner "/"
?	Genau ein beliebiges Zeichen (außer Verz.trenner "/")
*	0 oder mehr beliebige Zeichen (außer Verz.trenner "/")
**	0 oder mehr beliebige Zeichen (inkl. Verz.trenner "/")
**/	Pfad ohne Dateiname (nach letztem "/")
\C	Metazeichen C selbst (Backslash quotiert nächstes Zeichen)
\\	Zeichen "°\" selbst
[abc] [a-z]	Genau ein Zeichen aus Liste/Bereich (Zeichenmenge)
[!abc] [!a-z]	Genau ein Zeichen NICHT aus Liste/Bereich (sh, bash, ksh)
[^abc] [^a-z]	Genau ein Zeichen NICHT aus Liste/Bereich (csh, bash, ksh)
~	Home-Verzeichnis des aktuellen Benutzers
~USER	Home-Verzeichnis des Benutzers USER
{abc,def,...}	Liste der angegebenen Zeichenketten (csh, bash, ksh)
?(...|...)	0 oder 1 Vorkommen der Elemente in Musterliste (bash, ksh)
*(...|...)	0 oder mehr Vorkommen der ... (bash, ksh)
+(...|...)	1 oder mehr Vorkommen der ... (bash, ksh)
@(...|...)	Eines der Elemente in Musterliste (bash, ksh)
!(...|...)	Alles außer einem der Elemente der Musterliste (bash, ksh)

Diese Metazeichen können zu beliebig komplexen Suchmustern zusammengesetzt werden, passende Datei- und/oder Verzeichnisnamen müssen das Suchmuster vollständig erfüllen. Eine Längenbeschränkung oder eine Beschränkung in der Anzahl der verwendeten Metazeichen gibt es nicht.

```
ls *-{jan,feb,mar,apr,may,jun,jul,aug,sep,oct,nov,dec}-*   # -MMM-
ls [a-z][a-zA-Z0-9_]*.txt                                  #
ls *_{19,20}[0-9][0-9]-[0123][0-9]-[0123][0-9]_*           # _YYYY-MM-DD_
ls *-[012][0-9]:[0-5][0-9]:[0-5][0-9]-*                    # -HH:MM:SS-
```

Die sich ergebende Kommandozeile ist allerdings in ihrer Länge beschränkt (z.B. 2 Mio Zeichen bei der bash). Überschreitet man diese erlaubte Maximallänge, so wird das Kommando nicht ausgeführt und man erhält statt dessen eine

```
Fehlermeldung der Form:

  argument list too long
  command line too long

Den genauen Wert der maximalen Kommandozeilenlänge erhält man per "xargs --show-limits":

  Your environment variables take up 1733 bytes
  POSIX upper limit on argument length (this system): 2093371
  POSIX smallest allowable upper limit on argument length (all systems): 4096
  Maximum length of command we could actually use: 2091638
  Size of command buffer we are actually using: 131072
  Maximum parallelism (--max-procs must be no greater): 2147483647

Soll kein Globbing eines Suchmusters erfolgen (weil z.B. ein Dateiname
Metazeichen enthält), so ist es durch Quotierung aller oder einzelner
Metazeichen mit "...", '...' oder "\" zu schützen.

  ls "*?[]"
  ls '*?[]'
  ls \*\?\[\]

Mehrere Muster sind durch Leerraum zu trennen und werden von der Shell getrennt
expandiert und das Ergebnis in die endgültige Kommandozeile eingesetzt.

  ls *.sh *.awk *.pl tmp*
  ls    *.sh    *.awk    *.pl    tmp*

Dateinamen mit führendem Punkt ("versteckte Dateien") werden nur dann gefunden,
wenn der Punkt explizit angegeben wird. D.h. die Muster "*" und "?" matchen
keinen führenden Punkt "." in einem Dateinamen.

  ls .[^.]*   # Alle Dateinamen ausser "." und ".."

ACHTUNG: Die Verzeichnisse "." (aktuelles Verzeichnis und ".."
(Elternverzeichnis) beginnen ebenfalls mit einem Punkt!

  ls -d .*    # Alle Datei- und Verz.namen mit führendem Punkt

Auch Muster für ganze Dateipfade (Verzeichnis + Dateiname) sind angebbar, die
von der Shell bis zum (bitteren) Ende expandiert werden.

  ls *.sh
  ls /*/*.sh
  ls /*/*/*.sh
  ls /*/*/*/*.sh
  ls *.sh /*/*.sh /*/*/*.sh /*/*/*/*.sh
  ls /usr/local/{bin,sbin}/*.sh

Solange die Länge der generierten Liste von Dateinamen unter der maximal
möglichen der aktuellen Shell liegt (etwa 2 Mio Zeichen), werden alle passenden
Dateinamen dafür von der Shell eingesetzt.


3) Tipps
--------
* Die Angabe der Option "-d" (directory) verhindert bei "ls" das Auflisten des
  INHALTS von Verzeichnissen, es wird nur der Verzeichnisname ausgegeben
  (Beispiel: "X11" in "/bin").

* Statt "ls" kann also auch "echo" zum Auflisten der zu einem Suchmuster
  passenden Dateinamen verwendet werden.

* Das Verzeichnis "/usr/bin" eignet sich aufgrund der vielen darin enthaltenen
  Dateien sehr gut zum Ausprobieren der Suchmuster.


4) Beispiele
------------
Immer "ls -d" oder "echo" voransetzen:

  +--------------+-------------------------------------------------------------+
  | Muster       | Dateinamen ...                                              |
  +--------------+-------------------------------------------------------------+
  | a*           | ... mit "a" am Anfang (auch "a")                            |
  | *a           | ... mit "a" am Ende (auch "a")                              |
  | *a*          | ... mit mind. einem "a" (auch "a")                          |
  | *a*a*        | ... mit mind. zwei "a" (auch "aa")                          |
  | *aa*         | ... mit mind. zwei "a" direkt hintereinander (auch "aa")    |
  | [a-z][a-z]   | Kleingeschriebene zweibuchstabige ...                       |
  | *[0-9]*[0-9]*| ... mit mind. zwei Ziffern (nicht "0".."9")                 |
  | *a*e*i*o*u*  | ... mit mind. 5 Vokalen in angegebener Reihenfolge          |
```

?	Einbuchstabige ...
??	Zweibuchstabige ...
?a?	... mit 3 Buchstaben und "a" als zweitem Buchstaben
*.c	... mit Endung ".c"
/*/*.c	... mit Endung ".c" in Unterverz. des Root-Verz.
/*/*/*.c	... mit Endung ".c" in Unter-Unterverz. des Root-Verz.
*.[ch]	... die auf ".c" oder ".h" enden
*[!.][!ch]	... die als vorletztes Zeichen nicht "." und als letztes nicht "c" oder "h" haben (z.B. "a.", ".b", "aa")
*.{c,h,sh}	... die auf ".c", ".h" oder ".sh" enden
*.[ch] *.sh	... (analog)
.[!.]*	Versteckte ... (aber nicht "." und "..")