

## Wild Cards Reference





The shell understands a number of special characters when expanding filenames with wild cards. The case construct and if condition used in shell programming can also make use of these as well. These are summarised here.

<b>*</b>	matches any string of characters, including the null string <i>e.g.</i> file* matches file, file4, fileabc, file.c
<b>?</b>	matches any single character <i>e.g.</i> file? matches file4, filex, file1
<b>[abcd]</b>	matches any of the characters a,b,c,d <i>e.g.</i> file[xyz] matches filex, filey, filez
<b>[a-f]</b>	matches any single character lexically between a and f <i>e.g.</i> file[0-3] matches file0, file1, file2, file3
<b>[!abc]</b>	file[a-d7-9] matches filea, fileb, filec, filed, file7, file8, file9 matches any single character, <i>except</i> a, b, c <i>e.g.</i> file[!a-h] matches filej, filek, ... but not any of filea, ..., fileh
<b>"xyz"</b>	matches xyz exactly, except for shell variable expansion <i>e.g.</i> "file*" matches file*
<b>'xyz'</b>	matches xyz exactly, suppressing all special character <i>e.g.</i> "\$file" matches \$file
<b>\c</b>	matches any special character c exactly <i>e.g.</i> file\? matches file?
<b>a b</b>	only interpreted in case expressions - matches a or b

### SHELL SPECIAL CHARACTERS

<b>`</b>	command substitution (back quotes)
<b>\$( )</b>	command substitution (Korn Shell only)
<b>\$</b>	var variable substitution
<b>' '</b>	disable all special characters (single quotes)
<b>" "</b>	parameter and command substitution only (double quotes)
<b>* ? []</b>	filename generation
<b> </b>	pipeline (also ^ in older systems)
<b>;</b>	command separator
<b>&amp;</b>	run command in background
<b>( )</b>	run commands in sub-shell
<b>&lt;file</b>	redirect standard input
<b>&gt;file</b>	redirect standard output
<b>&gt;&gt;file</b>	append standard output
<b>2&gt;file</b>	redirect standard error
<b>2&gt;&amp;1</b>	duplicate standard error to standard output
<b>\ch</b>	escape special character <i>ch</i>

## REGULAR EXPRESSION CHARACTERS

.	matches any single character <i>e.g.</i> a.c matches abc, adc, a4c
[abcd]	matches any single character of a, b, c, d/ <i>e.g.</i> patt[abc] matches patta, pattb, pattc
[a-f]	matches any single character lexically between a and f <i>e.g.</i> patt[0-3] matches patt0, patt1, patt2, patt3
[^abc]	matches any single character, <i>except</i> a, b, c <i>e.g.</i> patt[!a-h] matches pattj, pattk, ... but not any of patta, ..., patth
^	matches the beginning of the line <i>e.g.</i> ^foo matches foo only if it occurs at the start of a line
\$	matches the end of the line <i>e.g.</i> foo\$ matches foo only if it occurs at the end of a line
f*	matches any number of occurrences of 'f', including 0.
f{5\}	matches 'f' exactly five times
f{5,\}	'f' 5 or more times
f{5,30\}	matches 'f' between 5 and 30 times