

BBEdit grep

quick reference

support@barebones.com
www.barebones.com/support

by Charles Poynton, PhD

A character in a grep pattern generally matches itself. Meta-characters (summarized to the right) must be preceded by backslash (\) to match literally.

PATTERN MODIFIERS (?on-off)	
(?i)	case insensitive
(?m)	multiline: ^ and \$ match begin/end of file
(?s)	. matches newline, but consider (. \n) instead
(?x)	ignore whitespace; allow # comments

A character class [abc] matches exactly one character, but [abc] may be qualified by +, *, ?, {n} etc. A-Z indicates a range. Within a character class, characters such as . * | match themselves with no special interpretation. To match a hyphen, place it first or backslash-escape it. Backslash escape \] matches], \\ matches \.

POSIX CHARACTER CLASSES match 1 char *within* a character class set

[:alnum:]	alphanumeric
[:alpha:]	including Unicode
[:ascii:]	[\x00-\x7f]
[:blank:]	horiz whitespace
[:cntrl:]	control codes
[:digit:]	[0-9], same as \d
[:graph:]	printable not spaces
[:lower:]	lowercase letter
[:print:]	printable incl spaces
[:punct:]	punctuation
[:space:]	same as \s
[:upper:]	uppercase letter
[:word:]	same as \w
[:xdigit:]	hex digit [0-9A-Fa-f]
[:^class:]	invert the class

REPLACEMENT PATTERN	
escape	is backslash in BBEdition; other systems may use \ or \$
& or \0	entire match
\1..\99	k th subpattern
\P<a>	named subpattern
\u	case specifier: make next character upper/lowercase
\l	
\U	make characters upper/lowercase until next case specifier
\L	
\E	end case transform

PRIMITIVES			
\A	0 chars at start of file	\z	...at end of file
\Z	0 chars at end of file or in front of newline immediately before eof		
^	0 chars at start of line	\$...at end of line
\b	0 chars at word boundary	\B	...not at a word boundary
.	dot/period matches any 1 character except newline, but see (?s)		
\\ \^ \.	match literal backslash, caret, period; backslash escape is also necessary to match \$ () * ? { } [] +		
\t	tab; equivalently, \x09		
\n	newline/"linefeed"	\r	in BBEdition, identical to \n
\0377	(backslash, zero) octal code [deprecated, use hex codes instead]		
\xff	hexadecimal code	\x{ffff}	hexadecimal code
\s	whitespace character	\S	non-whitespace char
\w	word char, equivalent to [a-zA-Z0-9etc]	\W	non-word char (incl. newline)
\d	digit	\D	nondigit (incl. newline)
[abc]	any single char in set (see also POSIX CLASSES)	[a-z]	any single character in specified range
[^abcx-z]	any single char (possibly \n) <i>not</i> in specified characters or range[s]		
ALTERNATIVES/SUBPATTERNS			
p q	alternation, match p or match q; also, p q r s		
(p)	subpattern (capture)	\1 .. \99	backreference to subpattern capture
(?:p)	subpattern (noncapture / cluster-only)		
(?P<a>p)	name a for subpattern p (named capture)	(?P=a)	backreference to named capture
QUANTIFIERS			
p?	zero or one (greedy)	p??	zero or one (nongreedy)
p*	zero or more (greedy)	p*?	zero or more (nongreedy)
p+	one or more (greedy)	p+?	one or more (nongreedy)
p{n}	exactly n (greedy)	p{n}?	exactly n (nongreedy)
p{m,}	at least m (greedy)	p{m,}?	at least m (nongreedy)
p{m,n}	at least m, but no more than n (greedy)	p{m,n}?	at least m, but no more than n (nongreedy)
POSITIONAL ASSERTIONS match 0 characters			
(?!x)	negative lookbehind: match subpattern only if <i>not</i> preceded by x	(?!x)	negative lookahead: match subpattern only if <i>not</i> followed by x
(?<=x)	positive lookbehind: match subpattern only if preceded by x	(?=x)	positive lookahead: match subpattern only if followed by x
CONDITIONAL SUBPATTERNS			
(?(k)y)	if subpattern number k matched, attempt match of y-clause, else, skip; for named subpattern, (?(P<a>)y)		
(?(k)y n)	if subpattern number k matched, attempt match of y-clause; else, attempt match of n-clause		
ONCE-ONLY SUBPATTERNS (NONCAPTURING)			
(?>p)	If p matches, prevent backtracking in p		
COMMENT in (?x) mode, # starts comment until and including newline; also ...			
(?#comment)	enclosed text is ignored, whether or not x-mode is set		