GREP 8.01 Quick Reference Card

grepqrc.doc Courtesy of Matthew Miner, revised 18 Apr 2005
GREP Copyright 1986–2021 Stan Brown, BrownMath.com

usage: GREP [options] [regex] [inputfilepecs]
or: GREP [options] [regex] <inputfile

See *grepman.htm* or **"grep /? | more"** for details and examples.
Feedback and suggestions are appreciated. Send to
tech support: **brownmath.com/contact**

[inputfilespecs]

May include drive and path (separator is \ or /)
Case is ignored for file names (AbC.Ef = abc.ef)

?	matches any one char(acter), including dot (.)
*	matches zero or more chars, including dot (.)
[*c*]	matches any one of the member chars
[^*c*]	matches any one char *not* listed
[*m*-*n*]	matches range of chars

"Globbing" Examples:
[a-c]*.c all .c files that start with a b or c
[kxz]*. files that start with k x or z and have no extension
[^ab]*.?[**0-9**] files that don't begin with a or b and have an ext w/ 1 char plus a digit.

Usage: grep pattern source[0-9][0-9].hpp source[0-9][0-9].cpp *.mak
grep pattern *.hpp *.cpp *.mak
grep pattern source*.[hc]pp *.mak
grep pattern *.[hc]pp *.mak
grep pattern *.?pp *.mak
grep pattern *

GREP Options

Start with / or - and may be run together.
Options may also be stored in the ORS_GREP environment variable.
(Options marked like this work in the registered version only.)

Input File Options:

/@*file*	Take input filespecs from the *file*
/@–	Take input filespecs from the standard input
/A	Search hidden and system files when matching wildcards
/G*n*	Text read mode:
/G0	Text lines are split at the /W option width
/G1	(default) Read text lines of any length
/G2	Paragraph mode (non-blank lines separated by blank lines)
/R*n*	File read mode:
/R0	(default) Read all files as text
/R2	Read all files as record-oriented binary
/R3	Read all files as free-form binary
/R-1	Decide /R0 or /R3 based on first 256 bytes of each file
/R-2	Decide /R0 or /R3 based on all bytes of each file
/S	Search subdirectories
/W*t*,*b*	Expect lines *t* characters or shorter and read binary files *b* characters at a time (default: 4096). If *b* is omitted, *b* = *t*.
/X*pattern*	Exclude files that match *pattern*, which may include * ? and [...] wildcards but no path. Multiple /X options are allowed.

Pattern-Matching Options:

/En	Regex type (0-4); default is "/E1\")
/E0	Match literal string, not as a regex
/E1	Match as a basic regex
/E2	Match as an extended regex
/E4	Match regex as a word (see /M to define a "word")
	Add backslash (/E0\) to turn on Special Rules for Command Line
/Ffile	Read one or more regexes from file, not command line
/F–	Read one or more regexes from the standard input
/I	Ignore case when matching
/Mloc	See docs for Locale options
/V	Report or count lines that DON'T contain a match
/Y	Report only lines that match ALL regexes (when /F is used and multiple regexes are entered)

Output Options:

/B	Show headers for all files, not just files with matches
/C	Show count of matching lines, not the lines themselves
/H	Suppress headers (filepecs) in output
/Jmode	Decide what to show for a match:
/J0	(default) Show the lines that contain a match
/J1	Show just the first matching part of a line
/J2	Show just the (non-overlapping) matching parts of a line
/J3	Show just the matching parts of a line (matches can overlap)
/K*n*	Show just the first *n* matches in each file
/L	Show just the filespecs that contain matches
/N	Show line numbers
/O*t*,*b*	Output format for text and binary (If *b* is omitted, *b*=*t*):
/O–1	(default) GREP decides based on input format
/O0	Hex dump and printable characters side by side
/O1	"Naïve" text – no special processing
/O2	"Safety filter" – text but with non-printables as ^X or <nn>
/O3	"Control filter" – like /O2 but CR and LF also break lines
/P*b*,*a*	Show *b* lines before and *a* lines after any match
/U	UNIX-style output: filespecs on lines with hits

General Options:

/?	Display help message and exit
/1	Exit status 1 if matches found, 0 if none
/0	Exit status 0 if matches found, 1 if none
/3	Exit status 3 if warnings but no errors
/D*file*	Append detailed debugging information to *file*
/D	Display detailed debugging information to standard error
/D–	Display detailed debugging information to standard output
/Qn	Quietness level:
/Q0	Show program logo and all warnings (default)
/Q1	Suppress program logo
/Q2	Suppress program logo and less serious warnings
/Q3	Suppress program logo and all warnings, even missing files
/Z	Reset all options to default values (recommended in batch files)

Regular Expressions – Regexes

Basic (/E1 option, default):

x	Any character (except as noted below) matches itself
.	(dot) matches any character except newline
*	(star) after anything matches 0 or more occurrences
+	(plus) after anything matches 1 or more occurrences.
[*c*]	matches any one of the member chars
[^*c*]	matches any one char *not* listed
[*m*-*n*]	matches range of chars
^	Start of line
$	End of line
\	Before any char takes away special meaning

Extended (/E2 or /E4) – all the above, plus:

?	0 or 1 occurrence
|	alternatives
(...)	subexpressions
{*m*,*n*}	*m* to *n* occurrences (also: {m}, {m,}, {,n})
*?, +?, ??, {...}?	same as *, +, ?, {...} but "ungreedy".

and these character types and assertions:

\w, \W	any word (non-word) character
\d, \D	any decimal (non-decimal) digit (e.g., [0-9] or [^0-9])
\s, \S	any whitespace (non-whitespace) char
\b, \B	word (non-)boundary position

Extended regex support is provided by the PCRE library, written by Philip Hazel and copyright by the University of Cambridge, England. Please see docs for more features of extended regexes.

Examples:

abcd	abcd
a.c	a<any char>c: abc, acc, a4c, a%c
ab*c	a<0 or more b>c: ac, abbbc
a.+c	a<1 or more anything>c: awe#!c, but *not* ac
^ab$	ab alone on line
a*b	matches a*b (* not special because of \)
\bcat	*cat*cher but not dogcatcher
(dog|cat)catcher	dogcatcher or catcatcher, *not* mousecatcher

Special Rules for the Command Line

DOS can't take certain characters on the command line. To get them into a regex, use the /F option and enter them from a file or the keyboard. Or use the (deprecated) Special Rules for the Command Line, provided that you don't use the /E option or **do** put a \ after the /E and number (e.g., /E2\):

\" for quote "	\q for equals =
\s for the space character	\l for less <
\t for tab	\g for greater >
\c for comma ,	\v for vertical bar |
\i for semicolon ;	\e for escape

plus numeric escapes like decimal \34, hex \0x22, and octal \042.

If - or / starts the regex, add a leading \ else it looks like an option.