NAME

pcregrep - a grep with Perl-compatible regular expressions.

SYNOPSIS

pcregrep [options] [long options] [pattern] [path1 path2 ...]

DESCRIPTION

pcregrep searches files for character patterns, in the same way as other grep commands do, but it uses the PCRE regular expression library to support patterns that are compatible with the regular expressions of Perl 5. See **pcresyntax**(3) for a quick-reference summary of pattern syntax, or **pcrepattern**(3) for a full description of the syntax and semantics of the regular expressions that PCRE supports.

Patterns, whether supplied on the command line or in a separate file, are given without delimiters. For example:

```
pcregrep Thursday /etc/motd
```

If you attempt to use delimiters (for example, by surrounding a pattern with slashes, as is common in Perl scripts), they are interpreted as part of the pattern. Quotes can of course be used to delimit patterns on the command line because they are interpreted by the shell, and indeed quotes are required if a pattern contains white space or shell metacharacters.

The first argument that follows any option settings is treated as the single pattern to be matched when neither **-e** nor **-f** is present. Conversely, when one or both of these options are used to specify patterns, all arguments are treated as path names. At least one of **-e**, **-f**, or an argument pattern must be provided.

If no files are specified, **pcregrep** reads the standard input. The standard input can also be referenced by a name consisting of a single hyphen. For example:

```
pcregrep some-pattern /file1 - /file3
```

By default, each line that matches a pattern is copied to the standard output, and if there is more than one file, the file name is output at the start of each line, followed by a colon. However, there are options that can change how **pcregrep** behaves. In particular, the **-M** option makes it possible to search for patterns that span line boundaries. What defines a line boundary is controlled by the **-N** (**--newline**) option.

The amount of memory used for buffering files that are being scanned is controlled by a parameter that can be set by the **--buffer-size** option. The default value for this parameter is specified when **pcregrep**

is built, with the default default being 20K. A block of memory three times this size is used (to allow for buffering "before" and "after" lines). An error occurs if a line overflows the buffer.

Patterns can be no longer than 8K or BUFSIZ bytes, whichever is the greater. BUFSIZ is defined in **<stdio.h>**. When there is more than one pattern (specified by the use of **-e** and/or **-f**), each pattern is applied to each line in the order in which they are defined, except that all the **-e** patterns are tried before the **-f** patterns.

By default, as soon as one pattern matches a line, no further patterns are considered. However, if **--colour** (or **--color**) is used to colour the matching substrings, or if **--only-matching**, **--file-offsets**, or **--line-offsets** is used to output only the part of the line that matched (either shown literally, or as an offset), scanning resumes immediately following the match, so that further matches on the same line can be found. If there are multiple patterns, they are all tried on the remainder of the line, but patterns that follow the one that matched are not tried on the earlier part of the line.

This behaviour means that the order in which multiple patterns are specified can affect the output when one of the above options is used. This is no longer the same behaviour as GNU grep, which now manages to display earlier matches for later patterns (as long as there is no overlap).

Patterns that can match an empty string are accepted, but empty string matches are never recognized. An example is the pattern "(super)?(man)?", in which all components are optional. This pattern finds all occurrences of both "super" and "man"; the output differs from matching with "super|man" when only the matching substrings are being shown.

If the **LC_ALL** or **LC_CTYPE** environment variable is set, **pcregrep** uses the value to set a locale when calling the PCRE library. The **--locale** option can be used to override this.

SUPPORT FOR COMPRESSED FILES

It is possible to compile **pcregrep** so that it uses **libz** or **libbz2** to read files whose names end in **.gz** or **.bz2**, respectively. You can find out whether your binary has support for one or both of these file types by running it with the **--help** option. If the appropriate support is not present, files are treated as plain text. The standard input is always so treated.

BINARY FILES

By default, a file that contains a binary zero byte within the first 1024 bytes is identified as a binary file, and is processed specially. (GNU grep also identifies binary files in this manner.) See the **--binary-files** option for a means of changing the way binary files are handled.

OPTIONS

The order in which some of the options appear can affect the output. For example, both the **-h** and **-l**

options affect the printing of file names. Whichever comes later in the command line will be the one that takes effect. Similarly, except where noted below, if an option is given twice, the later setting is used. Numerical values for options may be followed by K or M, to signify multiplication by 1024 or 1024*1024 respectively.

-- This terminates the list of options. It is useful if the next item on the command line starts with a hyphen but is not an option. This allows for the processing of patterns and filenames that start with hyphens.

-A *number*, **--after-context=***number*

Output *number* lines of context after each matching line. If filenames and/or line numbers are being output, a hyphen separator is used instead of a colon for the context lines. A line containing "--" is output between each group of lines, unless they are in fact contiguous in the input file. The value of *number* is expected to be relatively small. However, **pcregrep** guarantees to have up to 8K of following text available for context output.

-a, **--text** Treat binary files as text. This is equivalent to **--binary-files**=*text*.

-B *number*, **--before-context=***number*

Output *number* lines of context before each matching line. If filenames and/or line numbers are being output, a hyphen separator is used instead of a colon for the context lines. A line containing "--" is output between each group of lines, unless they are in fact contiguous in the input file. The value of *number* is expected to be relatively small. However, **pcregrep** guarantees to have up to 8K of preceding text available for context output.

--binary-files=*word*

Specify how binary files are to be processed. If the word is "binary" (the default), pattern matching is performed on binary files, but the only output is "Binary file <name> matches" when a match succeeds. If the word is "text", which is equivalent to the **-a** or **--text** option, binary files are processed in the same way as any other file. In this case, when a match succeeds, the output may be binary garbage, which can have nasty effects if sent to a terminal. If the word is "without-match", which is equivalent to the **-I** option, binary files are not processed at all; they are assumed not to be of interest.

--buffer-size=*number*

Set the parameter that controls how much memory is used for buffering files that are being scanned.

-C *number*, **--context=***number*

Output *number* lines of context both before and after each matching line. This is equivalent to setting both **-A** and **-B** to the same value.

-c, **--count** Do not output individual lines from the files that are being scanned; instead output the number of lines that would otherwise have been shown. If no lines are selected, the number zero is output. If several files are are being scanned, a count is output for each of them. However, if the **--files-with-matches** option is also used, only those files whose counts are greater than zero are listed. When **-c** is used, the **-A**, **-B**, and **-C** options are ignored.

--colour, **--color**

If this option is given without any data, it is equivalent to "--colour=auto". If data is required, it must be given in the same shell item, separated by an equals sign.

--colour=*value*, **--color=***value*

This option specifies under what circumstances the parts of a line that matched a pattern should be coloured in the output. By default, the output is not coloured. The value (which is optional, see above) may be "never", "always", or "auto". In the latter case, colouring happens only if the standard output is connected to a terminal. More resources are used when colouring is enabled, because **pcregrep** has to search for all possible matches in a line, not just one, in order to colour them all.

The colour that is used can be specified by setting the environment variable PCREGREP_COLOUR or PCREGREP_COLOR. The value of this variable should be a string of two numbers, separated by a semicolon. They are copied directly into the control string for setting colour on a terminal, so it is your responsibility to ensure that they make sense. If neither of the environment variables is set, the default is "1;31", which gives red.

-D *action*, **--devices=***action*

If an input path is not a regular file or a directory, "action" specifies how it is to be processed. Valid values are "read" (the default) or "skip" (silently skip the path).

-d *action*, **--directories=***action*

If an input path is a directory, "action" specifies how it is to be processed. Valid values are "read" (the default in non-Windows environments, for compatibility with GNU grep), "recurse" (equivalent to the **-r** option), or "skip" (silently skip the path, the default in Windows environments). In the "read" case, directories are read as if they were ordinary files. In some operating systems the effect of reading a directory like this is an immediate end-of-file; in others it may provoke an error.

-e *pattern*, **--regex=***pattern*, **--regexp=***pattern*

Specify a pattern to be matched. This option can be used multiple times in order to specify several patterns. It can also be used as a way of specifying a single pattern that starts with a hyphen. When **-e** is used, no argument pattern is taken from the command line; all arguments are treated as file names. There is no limit to the number of patterns. They are applied to each line in the order in which they are defined until one matches.

If **-f** is used with **-e**, the command line patterns are matched first, followed by the patterns from the file(s), independent of the order in which these options are specified. Note that multiple use of **-e** is not the same as a single pattern with alternatives. For example, X|Y finds the first character in a line that is X or Y, whereas if the two patterns are given separately, with X first, **pcregrep** finds X if it is present, even if it follows Y in the line. It finds Y only if there is no X in the line. This matters only if you are using **-o** or **--colo(u)r** to show the part(s) of the line that matched.

--exclude=*pattern*

Files (but not directories) whose names match the pattern are skipped without being processed. This applies to all files, whether listed on the command line, obtained from **--file-list**, or by scanning a directory. The pattern is a PCRE regular expression, and is matched against the final component of the file name, not the entire path. The **-F**, **-w**, and **-x** options do not apply to this pattern. The option may be given any number of times in order to specify multiple patterns. If a file name matches both an **--include** and an **--exclude** pattern, it is excluded. There is no short form for this option.

--exclude-from=*filename*

Treat each non-empty line of the file as the data for an **--exclude** option. What constitutes a newline when reading the file is the operating system’s default. The **--newline** option has no effect on this option. This option may be given more than once in order to specify a number of files to read.

--exclude-dir=*pattern*

Directories whose names match the pattern are skipped without being processed, whatever the setting of the **--recursive** option. This applies to all directories, whether listed on the command line, obtained from **--file-list**, or by scanning a parent directory. The pattern is a PCRE regular expression, and is matched against the final component of the directory name, not the entire path. The **-F**, **-w**, and **-x** options do not apply to this pattern. The option may be given any number of times in order to specify more than one pattern. If a directory matches both **--include-dir** and **--exclude-dir**, it is excluded. There is no short form for this option.

-F, **--fixed-strings**

Interpret each data-matching pattern as a list of fixed strings, separated by newlines, instead of as a regular expression. What constitutes a newline for this purpose is controlled by the **--newline** option. The **-w** (match as a word) and **-x** (match whole line) options can be used with **-F**. They apply to each of the fixed strings. A line is selected if any of the fixed strings are found in it (subject to **-w** or **-x**, if present). This option applies only to the patterns that are matched against the contents of files; it does not apply to patterns specified by any of the **--include** or **--exclude** options.

-f *filename*, **--file=***filename*

Read patterns from the file, one per line, and match them against each line of input. What constitutes a newline when reading the file is the operating system's default. The **--newline** option has no effect on this option. Trailing white space is removed from each line, and blank lines are ignored. An empty file contains no patterns and therefore matches nothing. See also the comments about multiple patterns versus a single pattern with alternatives in the description of **-e** above.

If this option is given more than once, all the specified files are read. A data line is output if any of the patterns match it. A filename can be given as "-" to refer to the standard input. When **-f** is used, patterns specified on the command line using **-e** may also be present; they are tested before the file's patterns. However, no other pattern is taken from the command line; all arguments are treated as the names of paths to be searched.

--file-list=*filename*

Read a list of files and/or directories that are to be scanned from the given file, one per line. Trailing white space is removed from each line, and blank lines are ignored. These paths are processed before any that are listed on the command line. The filename can be given as "-" to refer to the standard input. If **--file** and **--file-list** are both specified as "-", patterns are read first. This is useful only when the standard input is a terminal, from which further lines (the list of files) can be read after an end-of-file indication. If this option is given more than once, all the specified files are read.

--file-offsets

Instead of showing lines or parts of lines that match, show each match as an offset from the start of the file and a length, separated by a comma. In this mode, no context is shown. That is, the **-A**, **-B**, and **-C** options are ignored. If there is more than one match in a line, each of them is shown separately. This option is mutually exclusive with **--line-offsets** and **--only-matching**.

-H, **--with-filename**

Force the inclusion of the filename at the start of output lines when searching a single file. By default, the filename is not shown in this case. For matching lines, the filename is followed by a colon; for context lines, a hyphen separator is used. If a line number is also being output, it follows the file name.

-h, **--no-filename**

Suppress the output filenames when searching multiple files. By default, filenames are shown when multiple files are searched. For matching lines, the filename is followed by a colon; for context lines, a hyphen separator is used. If a line number is also being output, it follows the file name.

--help Output a help message, giving brief details of the command options and file type support, and then exit. Anything else on the command line is ignored.

-I Treat binary files as never matching. This is equivalent to **--binary-files**=*without-match*.

-i, **--ignore-case**

Ignore upper/lower case distinctions during comparisons.

--include=*pattern*

If any **--include** patterns are specified, the only files that are processed are those that match one of the patterns (and do not match an **--exclude** pattern). This option does not affect directories, but it applies to all files, whether listed on the command line, obtained from **--file-list**, or by scanning a directory. The pattern is a PCRE regular expression, and is matched against the final component of the file name, not the entire path. The **-F**, **-w**, and **-x** options do not apply to this pattern. The option may be given any number of times. If a file name matches both an **--include** and an **--exclude** pattern, it is excluded. There is no short form for this option.

--include-from=*filename*

Treat each non-empty line of the file as the data for an **--include** option. What constitutes a newline for this purpose is the operating system's default. The **--newline** option has no effect on this option. This option may be given any number of times; all the files are read.

--include-dir=*pattern*

If any **--include-dir** patterns are specified, the only directories that are processed are those that match one of the patterns (and do not match an **--exclude-dir** pattern). This applies to all directories, whether listed on the command line, obtained from **--file-list**, or by scanning a parent directory. The pattern is a PCRE regular expression, and is matched against the final component of the directory name, not the entire path. The **-F**, **-w**, and **-x**

options do not apply to this pattern. The option may be given any number of times. If a directory matches both **--include-dir** and **--exclude-dir**, it is excluded. There is no short form for this option.

-L, **--files-without-match**

Instead of outputting lines from the files, just output the names of the files that do not contain any lines that would have been output. Each file name is output once, on a separate line.

-l, **--files-with-matches**

Instead of outputting lines from the files, just output the names of the files containing lines that would have been output. Each file name is output once, on a separate line. Searching normally stops as soon as a matching line is found in a file. However, if the **-c** (count) option is also used, matching continues in order to obtain the correct count, and those files that have at least one match are listed along with their counts. Using this option with **-c** is a way of suppressing the listing of files with no matches.

--label=*name*

This option supplies a name to be used for the standard input when file names are being output. If not supplied, "(standard input)" is used. There is no short form for this option.

--line-buffered

When this option is given, input is read and processed line by line, and the output is flushed after each write. By default, input is read in large chunks, unless **pcregrep** can determine that it is reading from a terminal (which is currently possible only in Unix-like environments). Output to terminal is normally automatically flushed by the operating system. This option can be useful when the input or output is attached to a pipe and you do not want **pcregrep** to buffer up large amounts of data. However, its use will affect performance, and the **-M** (multiline) option ceases to work.

--line-offsets

Instead of showing lines or parts of lines that match, show each match as a line number, the offset from the start of the line, and a length. The line number is terminated by a colon (as usual; see the **-n** option), and the offset and length are separated by a comma. In this mode, no context is shown. That is, the **-A**, **-B**, and **-C** options are ignored. If there is more than one match in a line, each of them is shown separately. This option is mutually exclusive with **--file-offsets** and **--only-matching**.

--locale=*locale-name*

This option specifies a locale to be used for pattern matching. It overrides the value in the

LC_ALL or **LC_CTYPE** environment variables. If no locale is specified, the PCRE library's default (usually the "C" locale) is used. There is no short form for this option.

--match-limit=*number*

Processing some regular expression patterns can require a very large amount of memory, leading in some cases to a program crash if not enough is available. Other patterns may take a very long time to search for all possible matching strings. The **pcre_exec()** function that is called by **pcregrep** to do the matching has two parameters that can limit the resources that it uses.

The **--match-limit** option provides a means of limiting resource usage when processing patterns that are not going to match, but which have a very large number of possibilities in their search trees. The classic example is a pattern that uses nested unlimited repeats. Internally, PCRE uses a function called **match()** which it calls repeatedly (sometimes recursively). The limit set by **--match-limit** is imposed on the number of times this function is called during a match, which has the effect of limiting the amount of backtracking that can take place.

The **--recursion-limit** option is similar to **--match-limit**, but instead of limiting the total number of times that **match()** is called, it limits the depth of recursive calls, which in turn limits the amount of memory that can be used. The recursion depth is a smaller number than the total number of calls, because not all calls to **match()** are recursive. This limit is of use only if it is set smaller than **--match-limit**.

There are no short forms for these options. The default settings are specified when the PCRE library is compiled, with the default default being 10 million.

-M, **--multiline**

Allow patterns to match more than one line. When this option is given, patterns may usefully contain literal newline characters and internal occurrences of ^ and $ characters. The output for a successful match may consist of more than one line, the last of which is the one in which the match ended. If the matched string ends with a newline sequence the output ends at the end of that line.

When this option is set, the PCRE library is called in "multiline" mode. There is a limit to the number of lines that can be matched, imposed by the way that **pcregrep** buffers the input file as it scans it. However, **pcregrep** ensures that at least 8K characters or the rest of the document (whichever is the shorter) are available for forward matching, and similarly the previous 8K characters (or all the previous characters, if fewer than 8K) are guaranteed to be available for lookbehind assertions. This option does not work when input is read

line by line (see **--line-buffered**.)

-N *newline-type*, **--newline**=*newline-type*

The PCRE library supports five different conventions for indicating the ends of lines. They are the single-character sequences CR (carriage return) and LF (linefeed), the two-character sequence CRLF, an "anycrlf" convention, which recognizes any of the preceding three types, and an "any" convention, in which any Unicode line ending sequence is assumed to end a line. The Unicode sequences are the three just mentioned, plus VT (vertical tab, U+000B), FF (form feed, U+000C), NEL (next line, U+0085), LS (line separator, U+2028), and PS (paragraph separator, U+2029).

When the PCRE library is built, a default line-ending sequence is specified. This is normally the standard sequence for the operating system. Unless otherwise specified by this option, **pcregrep** uses the library's default. The possible values for this option are CR, LF, CRLF, ANYCRLF, or ANY. This makes it possible to use **pcregrep** to scan files that have come from other environments without having to modify their line endings. If the data that is being scanned does not agree with the convention set by this option, **pcregrep** may behave in strange ways. Note that this option does not apply to files specified by the **-f**, **--exclude-from**, or **--include-from** options, which are expected to use the operating system's standard newline sequence.

-n, **--line-number**

Precede each output line by its line number in the file, followed by a colon for matching lines or a hyphen for context lines. If the filename is also being output, it precedes the line number. This option is forced if **--line-offsets** is used.

--no-jit If the PCRE library is built with support for just-in-time compiling (which speeds up matching), **pcregrep** automatically makes use of this, unless it was explicitly disabled at build time. This option can be used to disable the use of JIT at run time. It is provided for testing and working round problems. It should never be needed in normal use.

-o, **--only-matching**

Show only the part of the line that matched a pattern instead of the whole line. In this mode, no context is shown. That is, the **-A**, **-B**, and **-C** options are ignored. If there is more than one match in a line, each of them is shown separately. If **-o** is combined with **-v** (invert the sense of the match to find non-matching lines), no output is generated, but the return code is set appropriately. If the matched portion of the line is empty, nothing is output unless the file name or line number are being printed, in which case they are shown on an otherwise empty line. This option is mutually exclusive with **--file-offsets** and **--line-offsets**.

-o*number*, **--only-matching**=*number*

Show only the part of the line that matched the capturing parentheses of the given number. Up to 32 capturing parentheses are supported, and -o0 is equivalent to **-o** without a number. Because these options can be given without an argument (see above), if an argument is present, it must be given in the same shell item, for example, -o3 or --only-matching=2. The comments given for the non-argument case above also apply to this case. If the specified capturing parentheses do not exist in the pattern, or were not set in the match, nothing is output unless the file name or line number are being printed.

If this option is given multiple times, multiple substrings are output, in the order the options are given. For example, -o3 -o1 -o3 causes the substrings matched by capturing parentheses 3 and 1 and then 3 again to be output. By default, there is no separator (but see the next option).

--om-separator=*text*

Specify a separating string for multiple occurrences of **-o**. The default is an empty string. Separating strings are never coloured.

-q, **--quiet** Work quietly, that is, display nothing except error messages. The exit status indicates whether or not any matches were found.

-r, **--recursive**

If any given path is a directory, recursively scan the files it contains, taking note of any **--include** and **--exclude** settings. By default, a directory is read as a normal file; in some operating systems this gives an immediate end-of-file. This option is a shorthand for setting the **-d** option to "recurse".

--recursion-limit=*number*

See **--match-limit** above.

-s, **--no-messages**

Suppress error messages about non-existent or unreadable files. Such files are quietly skipped. However, the return code is still 2, even if matches were found in other files.

-u, **--utf-8** Operate in UTF-8 mode. This option is available only if PCRE has been compiled with UTF-8 support. All patterns (including those for any **--exclude** and **--include** options) and all subject lines that are scanned must be valid strings of UTF-8 characters.

-V, **--version**

Write the version numbers of **pcregrep** and the PCRE library to the standard output and

then exit. Anything else on the command line is ignored.

-v, **--invert-match**

Invert the sense of the match, so that lines which do *not* match any of the patterns are the ones that are found.

-w, **--word-regex**, **--word-regexp**

Force the patterns to match only whole words. This is equivalent to having \b at the start and end of the pattern. This option applies only to the patterns that are matched against the contents of files; it does not apply to patterns specified by any of the **--include** or **--exclude** options.

-x, **--line-regex**, **--line-regexp**

Force the patterns to be anchored (each must start matching at the beginning of a line) and in addition, require them to match entire lines. This is equivalent to having ^ and $ characters at the start and end of each alternative branch in every pattern. This option applies only to the patterns that are matched against the contents of files; it does not apply to patterns specified by any of the **--include** or **--exclude** options.

ENVIRONMENT VARIABLES

The environment variables **LC_ALL** and **LC_CTYPE** are examined, in that order, for a locale. The first one that is set is used. This can be overridden by the **--locale** option. If no locale is set, the PCRE library's default (usually the "C" locale) is used.

NEWLINES

The **-N** (**--newline**) option allows **pcregrep** to scan files with different newline conventions from the default. Any parts of the input files that are written to the standard output are copied identically, with whatever newline sequences they have in the input. However, the setting of this option does not affect the interpretation of files specified by the **-f**, **--exclude-from**, or **--include-from** options, which are assumed to use the operating system's standard newline sequence, nor does it affect the way in which **pcregrep** writes informational messages to the standard error and output streams. For these it uses the string "\n" to indicate newlines, relying on the C I/O library to convert this to an appropriate sequence.

OPTIONS COMPATIBILITY

Many of the short and long forms of **pcregrep**'s options are the same as in the GNU **grep** program. Any long option of the form **--xxx-regexp** (GNU terminology) is also available as **--xxx-regex** (PCRE terminology). However, the **--file-list**, **--file-offsets**, **--include-dir**, **--line-offsets**, **--locale**, **--match-limit**, **-M**, **--multiline**, **-N**, **--newline**, **--om-separator**, **--recursion-limit**, **-u**, and **--utf-8** options are specific to **pcregrep**, as is the use of the **--only-matching** option with a capturing parentheses number.

Although most of the common options work the same way, a few are different in **pcregrep**. For example, the **--include** option's argument is a glob for GNU **grep**, but a regular expression for **pcregrep**. If both the **-c** and **-l** options are given, GNU grep lists only file names, without counts, but **pcregrep** gives the counts.

OPTIONS WITH DATA

There are four different ways in which an option with data can be specified. If a short form option is used, the data may follow immediately, or (with one exception) in the next command line item. For example:

```
-f/some/file
-f /some/file
```

The exception is the **-o** option, which may appear with or without data. Because of this, if data is present, it must follow immediately in the same item, for example -o3.

If a long form option is used, the data may appear in the same command line item, separated by an equals character, or (with two exceptions) it may appear in the next command line item. For example:

```
--file=/some/file
--file /some/file
```

Note, however, that if you want to supply a file name beginning with ~ as data in a shell command, and have the shell expand ~ to a home directory, you must separate the file name from the option, because the shell does not treat ~ specially unless it is at the start of an item.

The exceptions to the above are the **--colour** (or **--color**) and **--only-matching** options, for which the data is optional. If one of these options does have data, it must be given in the first form, using an equals character. Otherwise **pcregrep** will assume that it has no data.

MATCHING ERRORS

It is possible to supply a regular expression that takes a very long time to fail to match certain lines. Such patterns normally involve nested indefinite repeats, for example: (a+)*\d when matched against a line of a's with no final digit. The PCRE matching function has a resource limit that causes it to abort in these circumstances. If this happens, **pcregrep** outputs an error message and the line that caused the problem to the standard error stream. If there are more than 20 such errors, **pcregrep** gives up.

The **--match-limit** option of **pcregrep** can be used to set the overall resource limit; there is a second option called **--recursion-limit** that sets a limit on the amount of memory (usually stack) that is used (see the discussion of these options above).

DIAGNOSTICS

Exit status is 0 if any matches were found, 1 if no matches were found, and 2 for syntax errors, overlong lines, non-existent or inaccessible files (even if matches were found in other files) or too many matching errors. Using the **-s** option to suppress error messages about inaccessible files does not affect the return code.

SEE ALSO

pcrepattern(3), **pcresyntax**(3), **pcretest**(1).

AUTHOR

Philip Hazel
University Computing Service
Cambridge CB2 3QH, England.

REVISION

Last updated: 03 April 2014