NAME

PCRE2 - Perl-compatible regular expressions (revised API)

SYNOPSIS

#include <pcre2.h>

int pcre2_substitute(const pcre2_code ****code*, PCRE2_SPTR** *subject***,**
PCRE2_SIZE *length***, PCRE2_SIZE** *startoffset***,**
uint32_t *options***, pcre2_match_data ****match_data***,**
pcre2_match_context ****mcontext*, PCRE2_SPTR** *replacement***,**
PCRE2_SIZE *rlength***, PCRE2_UCHAR ****outputbuffer***,**
PCRE2_SIZE ****outlengthptr*);**

DESCRIPTION

This function matches a compiled regular expression against a given subject string, using a matching algorithm that is similar to Perl's. It then makes a copy of the subject, substituting a replacement string for what was matched. Its arguments are:

code Points to the compiled pattern
subject Points to the subject string
length Length of the subject string
startoffset Offset in the subject at which to start matching
options Option bits
match_data Points to a match data block, or is NULL
mcontext Points to a match context, or is NULL
replacement Points to the replacement string
rlength Length of the replacement string
outputbuffer Points to the output buffer
outlengthptr Points to the length of the output buffer

A match data block is needed only if you want to inspect the data from the final match that is returned in that block or if PCRE2_SUBSTITUTE_MATCHED is set. A match context is needed only if you want to:

Set up a callout function
Set a matching offset limit
Change the backtracking match limit
Change the backtracking depth limit
Set custom memory management in the match context

The *length*, *startoffset* and *rlength* values are code units, not characters, as is the contents of the variable pointed at by *outlengthptr*. This variable must contain the length of the output buffer when the function is called. If the function is successful, the value is changed to the length of the new string, excluding the trailing zero that is automatically added.

The subject and replacement lengths can be given as PCRE2_ZERO_TERMINATED for zero-terminated strings. The options are:

PCRE2_ANCHORED Match only at the first position
PCRE2_ENDANCHORED Pattern can match only at end of subject
PCRE2_NOTBOL Subject is not the beginning of a line
PCRE2_NOTEOL Subject is not the end of a line
PCRE2_NOTEMPTY An empty string is not a valid match
PCRE2_NOTEMPTY_ATSTART An empty string at the start of the subject is not a valid match
PCRE2_NO_JIT Do not use JIT matching

PCRE2_NO_UTF_CHECK Do not check the subject or replacement
for UTF validity (only relevant if
PCRE2_UTF was set at compile time)
PCRE2_SUBSTITUTE_EXTENDED Do extended replacement processing
PCRE2_SUBSTITUTE_GLOBAL Replace all occurrences in the subject
PCRE2_SUBSTITUTE_LITERAL The replacement string is literal
PCRE2_SUBSTITUTE_MATCHED Use pre-existing match data for 1st match
PCRE2_SUBSTITUTE_OVERFLOW_LENGTH If overflow, compute needed length
PCRE2_SUBSTITUTE_REPLACEMENT_ONLY Return only replacement string(s)
PCRE2_SUBSTITUTE_UNKNOWN_UNSET Treat unknown group as unset
PCRE2_SUBSTITUTE_UNSET_EMPTY Simple unset insert = empty string

If PCRE2_SUBSTITUTE_LITERAL is set, PCRE2_SUBSTITUTE_EXTENDED, PCRE2_SUBSTITUTE_UNKNOWN_UNSET, and PCRE2_SUBSTITUTE_UNSET_EMPTY are ignored.

If PCRE2_SUBSTITUTE_MATCHED is set, *match_data* must be non-zero; its contents must be the result of a call to **pcre2_match()** using the same pattern and subject.

The function returns the number of substitutions, which may be zero if there are no matches. The result may be greater than one only when PCRE2_SUBSTITUTE_GLOBAL is set. In the event of an error, a negative error code is returned.

There is a complete description of the PCRE2 native API in the **pcre2api** page and a description of the POSIX API in the **pcre2posix** page.