

NAME

PCRE - Perl-compatible regular expressions

SYNOPSIS

```
#include <pcre.h>
```

```
pcre *pcre_compile2(const char *pattern, int options,
    int *errorcodeptr,
    const char **errptr, int *erroffset,
    const unsigned char *tableptr);
```

```
pcre16 *pcre16_compile2(PCRE_SPTR16 pattern, int options,
    int *errorcodeptr,
    const char **errptr, int *erroffset,
    const unsigned char *tableptr);
```

```
pcre32 *pcre32_compile2(PCRE_SPTR32 pattern, int options,
    int *errorcodeptr,
    const char **errptr, int *erroffset,
    const unsigned char *tableptr);
```

DESCRIPTION

This function compiles a regular expression into an internal form. It is the same as `pcre[16|32]_compile()`, except for the addition of the `errorcodeptr` argument. The arguments are:

pattern A zero-terminated string containing the regular expression to be compiled
options Zero or more option bits
errorcodeptr Where to put an error code
errptr Where to put an error message
erroffset Offset in pattern where error was found
tableptr Pointer to character tables, or NULL to use the built-in default

The option bits are:

PCRE_ANCHORED	Force pattern anchoring
PCRE_AUTO_CALLOUT	Compile automatic callouts
PCRE_BSR_ANYCRLF	\R matches only CR, LF, or CRLF
PCRE_BSR_UNICODE	\R matches all Unicode line endings

PCRE_CASELESS	Do caseless matching
PCRE_DOLLAR_ENDONLY	\$ not to match newline at end
PCRE_DOTALL	. matches anything including NL
PCRE_DUPNAMES	Allow duplicate names for subpatterns
PCRE_EXTENDED	Ignore white space and # comments
PCRE_EXTRA	PCRE extra features (not much use currently)
PCRE_FIRSTLINE	Force matching to be before newline
PCRE_JAVASCRIPT_COMPAT	JavaScript compatibility
PCRE_MULTILINE	^ and \$ match newlines within data
PCRE_NEVER_UTF	Lock out UTF, e.g. via (*UTF)
PCRE_NEWLINE_ANY	Recognize any Unicode newline sequence
PCRE_NEWLINE_ANYCRLF	Recognize CR, LF, and CRLF as newline sequences
PCRE_NEWLINE_CR	Set CR as the newline sequence
PCRE_NEWLINE_CRLF	Set CRLF as the newline sequence
PCRE_NEWLINE_LF	Set LF as the newline sequence
PCRE_NO_AUTO_CAPTURE	Disable numbered capturing paren- theses (named ones available)
PCRE_NO_AUTO_POSSESS	Disable auto-possessification
PCRE_NO_START_OPTIMIZE	Disable match-time start optimizations
PCRE_NO_UTF16_CHECK	Do not check the pattern for UTF-16 validity (only relevant if PCRE_UTF16 is set)
PCRE_NO_UTF32_CHECK	Do not check the pattern for UTF-32 validity (only relevant if PCRE_UTF32 is set)
PCRE_NO_UTF8_CHECK	Do not check the pattern for UTF-8 validity (only relevant if PCRE_UTF8 is set)
PCRE_UCP	Use Unicode properties for \d, \w, etc.
PCRE_UNGREEDY	Invert greediness of quantifiers
PCRE_UTF16	Run pcre16_compile() in UTF-16 mode
PCRE_UTF32	Run pcre32_compile() in UTF-32 mode
PCRE_UTF8	Run pcre_compile() in UTF-8 mode

PCRE must be built with UTF support in order to use PCRE_UTF8/16/32 and PCRE_NO_UTF8/16/32_CHECK, and with UCP support if PCRE_UCP is used.

The yield of the function is a pointer to a private data structure that contains the compiled pattern, or

NULL if an error was detected. Note that compiling regular expressions with one version of PCRE for use with a different version is not guaranteed to work and may cause crashes.

There is a complete description of the PCRE native API in the **pcreapi** page and a description of the POSIX API in the **pcreposix** page.