## NAME

PCRE2 - Perl-compatible regular expressions (revised API)

## SYNOPSIS

**#include <pcre2.h>**

**pcre2_code *pcre2_compile(PCRE2_SPTR** *pattern*, **PCRE2_SIZE** *length*,
**uint32_t** *options*, **int** ***errorcode*, **PCRE2_SIZE** ***erroroffset,*
**pcre2_compile_context** ***ccontext*);

## DESCRIPTION

This function compiles a regular expression pattern into an internal form. Its arguments are:

| | |
|---|---|
| *pattern* | A string containing expression to be compiled |
| *length* | The length of the string or PCRE2_ZERO_TERMINATED |
| *options* | Option bits |
| *errorcode* | Where to put an error code |
| *erroffset* | Where to put an error offset |
| *ccontext* | Pointer to a compile context or NULL |

The length of the pattern and any error offset that is returned are in code units, not characters. A compile context is needed only if you want to provide custom memory allocation functions, or to provide an external function for system stack size checking, or to change one or more of these parameters:

What \R matches (Unicode newlines, or CR, LF, CRLF only);
PCRE2's character tables;
The newline character sequence;
The compile time nested parentheses limit;
The maximum pattern length (in code units) that is allowed.
The additional options bits (see pcre2_set_compile_extra_options())

The option bits are:

| | |
|---|---|
| PCRE2_ANCHORED | Force pattern anchoring |
| PCRE2_ALLOW_EMPTY_CLASS | Allow empty classes |
| PCRE2_ALT_BSUX | Alternative handling of \u, \U, and \x |
| PCRE2_ALT_CIRCUMFLEX | Alternative handling of ^ in multiline mode |
| PCRE2_ALT_VERBNAMES | Process backslashes in verb names |
| PCRE2_AUTO_CALLOUT | Compile automatic callouts |

| | |
|---|---|
| PCRE2_CASELESS | Do caseless matching |
| PCRE2_DOLLAR_ENDONLY | $ not to match newline at end |
| PCRE2_DOTALL | . matches anything including NL |
| PCRE2_DUPNAMES | Allow duplicate names for subpatterns |
| PCRE2_ENDANCHORED | Pattern can match only at end of subject |
| PCRE2_EXTENDED | Ignore white space and # comments |
| PCRE2_FIRSTLINE | Force matching to be before newline |
| PCRE2_LITERAL | Pattern characters are all literal |
| PCRE2_MATCH_INVALID_UTF | Enable support for matching invalid UTF |
| PCRE2_MATCH_UNSET_BACKREF | Match unset backreferences |
| PCRE2_MULTILINE | ^ and $ match newlines within data |
| PCRE2_NEVER_BACKSLASH_C | Lock out the use of \C in patterns |
| PCRE2_NEVER_UCP | Lock out PCRE2_UCP, e.g. via (*UCP) |
| PCRE2_NEVER_UTF | Lock out PCRE2_UTF, e.g. via (*UTF) |
| PCRE2_NO_AUTO_CAPTURE | Disable numbered capturing parentheses (named ones available) |
| PCRE2_NO_AUTO_POSSESS | Disable auto-possessification |
| PCRE2_NO_DOTSTAR_ANCHOR | Disable automatic anchoring for .* |
| PCRE2_NO_START_OPTIMIZE | Disable match-time start optimizations |
| PCRE2_NO_UTF_CHECK | Do not check the pattern for UTF validity (only relevant if PCRE2_UTF is set) |
| PCRE2_UCP | Use Unicode properties for \d, \w, etc. |
| PCRE2_UNGREEDY | Invert greediness of quantifiers |
| PCRE2_USE_OFFSET_LIMIT | Enable offset limit for unanchored matching |
| PCRE2_UTF | Treat pattern and subjects as UTF strings |

PCRE2 must be built with Unicode support (the default) in order to use PCRE2_UTF, PCRE2_UCP and related options.

Additional options may be set in the compile context via the **pcre2_set_compile_extra_options** function.

If either of *errorcode* or *erroroffset* is NULL, the function returns NULL immediately. Otherwise, the yield of this function is a pointer to a private data structure that contains the compiled pattern, or NULL if an error was detected. In the error case, a text error message can be obtained by passing the value returned via the *errorcode* argument to the the **pcre2_get_error_message()** function. The offset (in code units) where the error was encountered is returned via the *erroroffset* argument.

If there is no error, the value passed via *errorcode* returns the message "no error" if passed to **pcre2_get_error_message()**, and the value passed via *erroroffset* is zero.

There is a complete description of the PCRE2 native API, with more detail on each option, in the **pcre2api** page, and a description of the POSIX API in the **pcre2posix** page.