

NAME

PCRE - Perl-compatible regular expressions

SYNOPSIS

```
#include <pcre.h>

int pcre_config(int what, void *where);

int pcre16_config(int what, void *where);

int pcre32_config(int what, void *where);
```

DESCRIPTION

This function makes it possible for a client program to find out which optional features are available in the version of the PCRE library it is using. The arguments are as follows:

what A code specifying what information is required
where Points to where to put the data

The *where* argument must point to an integer variable, except for PCRE_CONFIG_MATCH_LIMIT, PCRE_CONFIG_MATCH_LIMIT_RECURSION, and PCRE_CONFIG_PARENS_LIMIT, when it must point to an unsigned long integer, and for PCRE_CONFIG_JITTARGET, when it must point to a const char*. The available codes are:

PCRE_CONFIG_JIT	Availability of just-in-time compiler support (1=yes 0=no)
PCRE_CONFIG_JITTARGET	String containing information about the target architecture for the JIT compiler, or NULL if there is no JIT support
PCRE_CONFIG_LINK_SIZE	Internal link size: 2, 3, or 4
PCRE_CONFIG_PARENS_LIMIT	Parentheses nesting limit
PCRE_CONFIG_MATCH_LIMIT	Internal resource limit
PCRE_CONFIG_MATCH_LIMIT_RECURSION	Internal recursion depth limit
PCRE_CONFIG_NEWLINE	Value of the default newline sequence:
	13 (0x000d) for CR
	10 (0x000a) for LF
	3338 (0x0d0a) for CRLF
	-2 for ANYCRLF
	-1 for ANY
PCRE_CONFIG_BSR	Indicates what \R matches by default:
	0 all Unicode line endings
	1 CR, LF, or CRLF only
PCRE_CONFIG_POSIX_MALLOC_THRESHOLD	Threshold of return slots, above which malloc() is used by the POSIX API
PCRE_CONFIG_STACKRECURSE	Recursion implementation (1=stack 0=heap)
PCRE_CONFIG_UTF16	Availability of UTF-16 support (1=yes 0=no); option for pcre16_config()
PCRE_CONFIG_UTF32	Availability of UTF-32 support (1=yes 0=no); option for pcre32_config()
PCRE_CONFIG_UTF8	Availability of UTF-8 support (1=yes 0=no); option for pcre_config()
PCRE_CONFIG_UNICODE_PROPERTIES	Availability of Unicode property support

(1=yes 0=no)

The function yields 0 on success or PCRE_ERROR_BADOPTION otherwise. That error is also given if PCRE_CONFIG_UTF16 or PCRE_CONFIG_UTF32 is passed to **pcre_config()**, if PCRE_CONFIG_UTF8 or PCRE_CONFIG_UTF32 is passed to **pcre16_config()**, or if PCRE_CONFIG_UTF8 or PCRE_CONFIG_UTF16 is passed to **pcre32_config()**.

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.