

**NAME**

PCRE - Perl-compatible regular expressions

**SYNOPSIS**

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

```
int pcre_dfa_exec(const pcre *code, const pcre_extra *extra,
    const char *subject, int length, int startoffset,
    int options, int *ovector, int ovecsz,
    int *workspace, int wscount);
```

```
int pcre16_dfa_exec(const pcre16 *code, const pcre16_extra *extra,
    PCRE_SPTR16 subject, int length, int startoffset,
    int options, int *ovector, int ovecsz,
    int *workspace, int wscount);
```

```
int pcre32_dfa_exec(const pcre32 *code, const pcre32_extra *extra,
    PCRE_SPTR32 subject, int length, int startoffset,
    int options, int *ovector, int ovecsz,
    int *workspace, int wscount);
```

**DESCRIPTION**

This function matches a compiled regular expression against a given subject string, using an alternative matching algorithm that scans the subject string just once (*not* Perl-compatible). Note that the main, Perl-compatible, matching function is **pcre[16|32]\_exec()**. The arguments for this function are:

<i>code</i>	Points to the compiled pattern
<i>extra</i>	Points to an associated <b>pcre[16 32]_extra</b> structure, or is NULL
<i>subject</i>	Points to the subject string
<i>length</i>	Length of the subject string
<i>startoffset</i>	Offset in the subject at which to start matching
<i>options</i>	Option bits
<i>ovector</i>	Points to a vector of ints for result offsets
<i>ovecsz</i>	Number of elements in the vector
<i>workspace</i>	Points to a vector of ints used as working space
<i>wscount</i>	Number of elements in the vector

The units for *length* and *startoffset* are bytes for **pcre\_exec()**, 16-bit data items for **pcre16\_exec()**, and 32-bit items for **pcre32\_exec()**. The options are:

PCRE_ANCHORED	Match only at the first position
PCRE_BSR_ANYCRLF	\R matches only CR, LF, or CRLF
PCRE_BSR_UNICODE	\R matches all Unicode line endings
PCRE_NEWLINE_ANY	Recognize any Unicode newline sequence
PCRE_NEWLINE_ANYCRLF	Recognize CR, LF, & CRLF as newline sequences
PCRE_NEWLINE_CR	Recognize CR as the only newline sequence
PCRE_NEWLINE_CRLF	Recognize CRLF as the only newline sequence
PCRE_NEWLINE_LF	Recognize LF as the only newline sequence
PCRE_NOTBOL	Subject is not the beginning of a line
PCRE_NOTEOL	Subject is not the end of a line
PCRE_NOTEMPTY	An empty string is not a valid match
PCRE_NOTEMPTY_ATSTART	An empty string at the start of the subject is not a valid match

**PCRE\_NO\_START\_OPTIMIZE** Do not do "start-match" optimizations  
**PCRE\_NO\_UTF16\_CHECK** Do not check the subject for UTF-16 validity (only relevant if **PCRE\_UTF16** was set at compile time)  
**PCRE\_NO\_UTF32\_CHECK** Do not check the subject for UTF-32 validity (only relevant if **PCRE\_UTF32** was set at compile time)  
**PCRE\_NO\_UTF8\_CHECK** Do not check the subject for UTF-8 validity (only relevant if **PCRE\_UTF8** was set at compile time)  
**PCRE\_PARTIAL** ) Return **PCRE\_ERROR\_PARTIAL** for a partial  
**PCRE\_PARTIAL\_SOFT** ) match if no full matches are found  
**PCRE\_PARTIAL\_HARD** Return **PCRE\_ERROR\_PARTIAL** for a partial match even if there is a full match as well  
**PCRE\_DFA\_SHORTEST** Return only the shortest match  
**PCRE\_DFA\_RESTART** Restart after a partial match

There are restrictions on what may appear in a pattern when using this matching function. Details are given in the **pcrematching** documentation. For details of partial matching, see the **pcrepartial** page.

A **pcre[16|32]\_extra** structure contains the following fields:

*flags* Bits indicating which fields are set  
*study\_data* Opaque data from **pcre[16|32]\_study()**  
*match\_limit* Limit on internal resource use  
*match\_limit\_recursion* Limit on internal recursion depth  
*callout\_data* Opaque data passed back to callouts  
*tables* Points to character tables or is NULL  
*mark* For passing back a \*MARK pointer  
*executable\_jit* Opaque data from JIT compilation

The flag bits are **PCRE\_EXTRA\_STUDY\_DATA**, **PCRE\_EXTRA\_MATCH\_LIMIT**, **PCRE\_EXTRA\_MATCH\_LIMIT\_RECURSION**, **PCRE\_EXTRA\_CALLOUT\_DATA**, **PCRE\_EXTRA\_TABLES**, **PCRE\_EXTRA\_MARK** and **PCRE\_EXTRA\_EXECUTABLE\_JIT**. For this matching function, the *match\_limit* and *match\_limit\_recursion* fields are not used, and must not be set. The **PCRE\_EXTRA\_EXECUTABLE\_JIT** flag and the corresponding variable are ignored.

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.