

Op:regex.h

regex.h - regular expression matching types

```
#include <regex.h>
```

Contents [1 DESCRIPTION](#) [2 APPLICATION USAGE](#) [3 RATIONALE](#) [4 FUTURE DIRECTIONS](#) [5 RELATED](#)
[6 COPYRIGHT](#) [7 CATEGORY](#)

DESCRIPTION

The `<regex.h>` header shall define the structures and symbolic constants used by the `regcomp()`, `regexec()`, `regerror()`, and `regfree()` functions.

The structure type `regex_t` shall contain at least the following member:

<code>size_t</code>	<code>re_nsub</code>	Number of parenthesized subexpressions.
---------------------	----------------------	---

The type `size_t` shall be defined as described in `<sys/types.h>` .

The type `regoff_t` shall be defined as a signed integer type that can hold the largest value that can be stored in

either a type `off_t` or type `ssize_t`. The structure type `regmatch_t` shall contain at least the following members:

<code>regoff_t</code>	<code>rm_so</code>	Byte offset from start of string to start of substring.
<code>regoff_t</code>	<code>rm_eo</code>	Byte offset from start of string of the first character after the end of substring.

Values for the `cflags` parameter to the `regcomp()` function are as follows:

`REG_EXTENDED`

Use Extended Regular Expressions.

REG_ICASE

Ignore case in match.

REG_NOSUB

Report only success or fail in **regexec()**.

REG_NEWLINE

Change the handling of <newline>.

Values for the **eflags** parameter to the **regexec()** function are as follows:

REG_NOTBOL

The circumflex character ('^'), when taken as a special character, does not match the beginning of **string**.

REG_NOTEOL

The dollar sign ('\$'), when taken as a special character, does not match the end of **string**.

The following constants shall be defined as error return values:

REG_NOMATCH

regexec() failed to match.

REG_BADPAT

Invalid regular expression.

REG_ECOLLATE

Invalid collating element referenced.

REG_ECTYPE

Invalid character class type referenced.

REG_EESCAPE

Trailing '\' in pattern.

REG_ESUBREG

Number in **\digit** invalid or in error.

REG_EBRACK

"[]" imbalance.

REG_EPAREN

"\(\)" or "()" imbalance.

REG_EBRACE

"\{\}" imbalance.

REG_BADBR

Content of "\{\}" invalid: not a number, number too large, more than two numbers, first larger than sec-

ond.

REG_ERANGE

Invalid endpoint in range expression.

REG_ESPACE

Out of memory.

REG_BADRPT

'?' , '*' , or '+' not preceded by valid regular expression.

REG_ENOSYS

Reserved.

The following shall be declared as functions and may also be defined as macros. Function prototypes shall be provided.

```
int    regcomp(regex_t *restrict, const char *restrict, int);
size_t regerror(int, const regex_t *restrict, char *restrict, size_t);
int    regexec(const regex_t *restrict, const char *restrict, size_t,
             regmatch_t[restrict], int);
void   regfree(regex_t *);
```

The implementation may define additional macros or constants using names beginning with REG_.

The following sections are informative.

APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

RELATED

<sys/types.h> , the System Interfaces volume of IEEE Std 1003.1-2001, *regcomp()*, the Shell and Utilities volume of IEEE Std 1003.1-2001

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1, 2003 Edition, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 6, Copyright (C) 2001-2003 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html> .

