z/OS / 2.4.0 / Change version

Feedback Product list

regerror() — Return error message

Last Updated: 2021-06-25

Standards

Standards / Extensions	C or C++	Dependencies
XPG4 XPG4.2 Single UNIX Specification, Version 3	both	

Format

```
#include <regex.h>

size_t regerror(int errcode, const regex_t *_restrict_preg,
                char *_restrict_errbuf, size_t errbuf_size);
```

General description

Finds the description for *errcode*. (For a description of regular expressions, see Regular expressions.)

Returned value

regerror() returns the integer value that is the size of the buffer needed to hold the generated description string for the error condition corresponding to *errcode*.

regerror() returns the following messages.

errcode

Description String

REG_BADBR

Invalid \{ \} range exp

REG_BADPAT

Invalid regular expression

REG_BADRPT

?*+ not preceded by valid RE

REG_EBOL

¬ anchor and not BOL

REG_EBRACE

\{ \} or { } imbalance

REG_EBRACK

[] imbalance

REG_ECHAR

Invalid multibyte character

REG_ECOLLATE

Invalid collating element

REG_ECTYPE

Invalid character class

REG_EEOL

$ anchor and not EOL

REG_EESCAPE

Last character is \

REG_EPAREN

\(\) or () imbalance

REG_ERANGE

Invalid range exp endpoint

REG_ESPACE

Out of memory

REG_ESUBREG

Invalid number in \digit

REG_NOMATCH

RE pattern not found

The LC_SYNTAX characters in the messages will be converted to the code points from the current LC_SYNTAX category.

Example

CELEBR08

```
/* CELEBR08

   This example compiles an invalid regular expression, and
   print error message &regerror..
```

```
 */
#include <regex.h>
#include <locale.h>
#include <stdio.h>
#include <stdlib.h>

main() {
    regex_t    preg;
    char       *pattern = "a[missing.bracket";
    int        rc;
    char       buffer[100];

    if ((rc = regcomp(&preg, pattern, REG_EXTENDED)) != 0) {
       regerror(rc, &preg, buffer, 100);
       printf("regcomp() failed with '%s'\n", buffer);
       exit(1);
    }
}
```

Related information

- See the topics about internationalization in z/OS XL C/C++ Programming Guide.
- regex.h — Regular expression functions
- regcomp() — Compile regular expression
- regexec() — Execute compiled regular expression
- regfree() — Free memory for regular expression

Parent topic:

→ Library functions

Previous
regcomp() — Compile regular expression

Next
regex() — Execute compiled regular expression