errno

Defined in header <cerrno>

```
#define errno /*implementation-defined*/
```

errno is a preprocessor macro used for error indication. It expands to a static (until C++11) thread-local (since C++11) modifiable lvalue of type int .

Several standard library functions indicate errors by writing positive integers to errno. Typically, the value of errno is set to one of the error codes, listed in <cerrno> as macro constants that begin with the letter E, followed by uppercase letters or digits.

The value of errno is 0 at program startup, and although library functions are allowed to write positive integers to errno whether or not an error occurred, library functions never store 0 in errno.

Example

Run this code

```cpp
#include <iostream>
#include <cmath>
#include <cerrno>
#include <cstring>
#include <clocale>

int main()
{
    double not_a_number = std::log(-1.0);
    std::cout << not_a_number << '\n';
    if (errno == EDOM) {
        std::cout << "log(-1) failed: " << std::strerror(errno) << '\n';
        std::setlocale(LC_MESSAGES, "de_DE.utf8");
        std::cout << "Or, in German, " << std::strerror(errno) << '\n';
    }
}
```

Possible output:

```
nan
log(-1) failed: Numerical argument out of domain
Or, in German, Das numerische Argument ist ausserhalb des Definitionsbereiches
```

Defect reports

The following behavior-changing defect reports were applied retroactively to previously published C++ standards.

DR	Applied to	Behavior as published	Correct behavior
LWG 310 (https://cplusplus.github.io/LWG/issue310)	C++98	it is unspecified whether errno is a macro or an identifier with external linkage	errno must be macro

See also

E2BIG, EACCES, ..., EXDEV	macros for standard POSIX-compatible error conditions (macro constant)
perror	displays a character string corresponding of the current error to stderr (function)
strerror	returns a text version of a given error code (function)
C documentation for **errno**	