

Error handling

Error numbers

Defined in header `<errno.h>`

errno	macro which expands to POSIX-compatible thread-local error number variable (macro variable)
E2BIG, EACCES, ..., EXDEV	macros for standard POSIX-compatible error conditions (macro constant)

Assertions

Defined in header `<assert.h>`

assert	aborts the program if the user-specified condition is not <code>true</code> . May be disabled for release builds (function macro)
static_assert (C11)	issues a compile-time diagnostic if the value of a constant expression is false (keyword macro)

Bounds checking

The standard library provides bounds-checked versions of some existing functions (`gets_s`, `fopen_s`, `printf_s`, `strcpy_s`, `wcscpy_s`, `mbstowcs_s`, `qsort_s`, `getenv_s`, etc). This functionality is *optional* and is only available if `__STDC_LIB_EXT1__` is defined. The following macros and functions support this functionality.

Defined in header `<errno.h>`
Defined in header `<stdio.h>`

errno_t (C11)	a typedef for the type <code>int</code> , used to self-document functions that return errno values (typedef)	
Defined in header <code><stddef.h></code> Defined in header <code><stdio.h></code> Defined in header <code><stdlib.h></code> Defined in header <code><string.h></code> Defined in header <code><time.h></code> Defined in header <code><wchar.h></code>		
rsize_t (C11)	a typedef for the same type as <code>size_t</code> , used to self-document functions that range-check their parameters at runtime (typedef)	(since C11)
Defined in header <code><stdint.h></code>		
RSIZE_MAX (C11)	largest acceptable size for bounds-checked functions, expands to either constant or variable which may change at runtime (e.g. as the currently allocated memory size changes) (macro variable)	
Defined in header <code><stdlib.h></code>		
set_constraint_handler_s (C11)	set the error callback for bounds-checked functions (function)	
abort_handler_s (C11)	abort callback for the bounds-checked functions (function)	
ignore_handler_s (C11)	ignore callback for the bounds-checked functions (function)	

Note: implementations of bounds-checked functions are available as open-source libraries Safe C (<https://github.com/rurban/safeclib/>) and Slibc (<https://code.google.com/archive/p/slibc/>), and as part of Watcom C. There is also an incompatible set of bounds-checked functions available in Visual Studio.

References

- C11 standard (ISO/IEC 9899:2011):
 - 7.2 Diagnostics `<assert.h>` (p: 186–187)
 - 7.5 Errors `<errno.h>` (p: 205)
 - 7.19 Common definitions `<stddef.h>` (p: 288)
 - 7.20 Integer types `<stdint.h>` (p: 289–295)
 - 7.21 Input/output `<stdio.h>` (p: 296–339)
 - 7.22 General utilities `<stdlib.h>` (p: 340–360)

- K.3.1.3 Use of errno (p: 584)
- K.3.2/2 errno_t (p: 585)
- K.3.3/2 rsize_t (p: 585)
- K.3.4/2 RSIZE_MAX (p: 585)
- 7.31.3 Errors <errno.h> (p: 455)
- 7.31.10 Integer types <stdint.h> (p: 456)
- 7.31.11 Input/output <stdio.h> (p: 456)
- 7.31.12 General utilities <stdlib.h> (p: 456)
- C99 standard (ISO/IEC 9899:1999):
 - 7.2 Diagnostics <assert.h> (p: 169)
 - 7.5 Errors <errno.h> (p: 186)
 - 7.26.3 Errors <errno.h> (p: 401)
 - 7.26.8 Integer types <stdint.h> (p: 401)
 - 7.26.9 Input/output <stdio.h> (p: 402)
 - 7.26.10 General utilities <stdlib.h> (p: 402)
- C89/C90 standard (ISO/IEC 9899:1990):
 - 4.2 DIAGNOSTICS <assert.h>
 - 4.1.3 Errors <errno.h>
 - 4.13.1 Errors <errno.h>
 - 4.13.6 Input/output <stdio.h>
 - 4.13.7 General utilities <stdlib.h>

See also

math_errhandling (C99) defines the error handling mechanism used by the common mathematical functions
MATH_ERRNO (C99) (macro constant)
MATH_ERREXCEPT (C99)

C++ documentation for Error handling

Retrieved from "https://en.cppreference.com/mwiki/index.php?title=c/error&oldid=117920"
