header

**<cstdio> (stdio.h)**

**C library to perform Input/Output operations**

Input and Output operations can also be performed in C++ using the **C** **St**andar**d** **I**nput and **O**utput Library (**cstdio**, known as stdio.h in the C language). This library uses what are called *streams* to operate with physical devices such as keyboards, printers, terminals or with any other type of files supported by the system. Streams are an abstraction to interact with these in an uniform way; All streams have similar properties independently of the individual characteristics of the physical media they are associated with.  
  
Streams are handled in the cstdio library as pointers to [FILE](http://www.cplusplus.com/FILE) objects. A pointer to a [FILE](http://www.cplusplus.com/FILE) object uniquely identifies a stream, and is used as a parameter in the operations involving that stream.  
  
There also exist three standard streams: [stdin](http://www.cplusplus.com/stdin), [stdout](http://www.cplusplus.com/stdout) and [stderr](http://www.cplusplus.com/stderr), which are automatically created and opened for all programs using the library.

**Stream properties**

Streams have some properties that define which functions can be used on them and how these will treat the data input or output through them. Most of these properties are defined at the moment the stream is associated with a file (opened) using the [fopen](http://www.cplusplus.com/fopen) function:

***Read/Write Access***

Specifies whether the stream has read or write access (or both) to the physical media they are associated with.

***Text / Binary***

Text streams are thought to represent a set of text lines, each one ending with a new-line character. Depending on the environment where the application is run, some character translation may occur with text streams to adapt some special characters to the text file specifications of the environment. A binary stream, on the other hand, is a sequence of characters written or read from the physical media with no translation, having a one-to-one correspondence with the characters read or written to the stream.

***Buffer***

A buffer is a block of memory where data is accumulated before being physically read or written to the associated file or device. Streams can be either *fully buffered*, *line buffered* or *unbuffered*. On fully buffered streams, data is read/written when the buffer is filled, on line buffered streams this happens when a new-line character is encountered, and on unbuffered streams characters are intended to be read/written as soon as possible.

***Orientation***

On opening, streams have no orientation. As soon as an input/output operation is performed on them, they become either *byte-oriented* or *wide-oriented*, depending on the operation performed (generally, functions defined in <cstdio> are *byte-oriented*, while functions in [<cwchar>](http://www.cplusplus.com/cwchar) are *wide-oriented*). See [cwchar](http://www.cplusplus.com/cwchar) for more info.

**Indicators**

Streams have certain internal indicators that specify their current state and which affect the behavior of some input and output operations performed on them:

Error indicator

This indicator is set when an error has occurred in an operation related to the stream. This indicator can be checked with the [ferror](http://www.cplusplus.com/ferror) function, and can be reset by calling either to [clearerr](http://www.cplusplus.com/clearerr), [freopen](http://www.cplusplus.com/freopen) or [rewind](http://www.cplusplus.com/rewind).

End-Of-File indicator

When set, indicates that the last reading or writing operation performed with the stream reached the *End of File*. It can be checked with the [feof](http://www.cplusplus.com/feof) function, and can be reset by calling either to [clearerr](http://www.cplusplus.com/clearerr) or [freopen](http://www.cplusplus.com/freopen) or by calling to any repositioning function ([rewind](http://www.cplusplus.com/rewind), [fseek](http://www.cplusplus.com/fseek) and [fsetpos](http://www.cplusplus.com/fsetpos)).

Position indicator

It is an internal pointer of each stream which points to the next character to be read or written in the next I/O operation. Its value can be obtained by the [ftell](http://www.cplusplus.com/ftell) and [fgetpos](http://www.cplusplus.com/fgetpos) functions, and can be changed using the repositioning functions [rewind](http://www.cplusplus.com/rewind), [fseek](http://www.cplusplus.com/fseek) and [fsetpos](http://www.cplusplus.com/fsetpos).

**Functions**

**Operations on files**:

[**remove**](http://www.cplusplus.com/reference/cstdio/remove/)

Remove file (function )

[**rename**](http://www.cplusplus.com/reference/cstdio/rename/)

Rename file (function )

[**tmpfile**](http://www.cplusplus.com/reference/cstdio/tmpfile/)

Open a temporary file (function )

[**tmpnam**](http://www.cplusplus.com/reference/cstdio/tmpnam/)

Generate temporary filename (function )

**File access**:

[**fclose**](http://www.cplusplus.com/reference/cstdio/fclose/)

Close file (function )

[**fflush**](http://www.cplusplus.com/reference/cstdio/fflush/)

Flush stream (function )

[**fopen**](http://www.cplusplus.com/reference/cstdio/fopen/)

Open file (function )

[**freopen**](http://www.cplusplus.com/reference/cstdio/freopen/)

Reopen stream with different file or mode (function )

[**setbuf**](http://www.cplusplus.com/reference/cstdio/setbuf/)

Set stream buffer (function )

[**setvbuf**](http://www.cplusplus.com/reference/cstdio/setvbuf/)

Change stream buffering (function )

**Formatted input/output**:

[**fprintf**](http://www.cplusplus.com/reference/cstdio/fprintf/)

Write formatted data to stream (function )

[**fscanf**](http://www.cplusplus.com/reference/cstdio/fscanf/)

Read formatted data from stream (function )

[**printf**](http://www.cplusplus.com/reference/cstdio/printf/)

Print formatted data to stdout (function )

[**scanf**](http://www.cplusplus.com/reference/cstdio/scanf/)

Read formatted data from stdin (function )

[**snprintf**](http://www.cplusplus.com/reference/cstdio/snprintf/)

Write formatted output to sized buffer (function )

[**sprintf**](http://www.cplusplus.com/reference/cstdio/sprintf/)

Write formatted data to string (function )

[**sscanf**](http://www.cplusplus.com/reference/cstdio/sscanf/)

Read formatted data from string (function )

[**vfprintf**](http://www.cplusplus.com/reference/cstdio/vfprintf/)

Write formatted data from variable argument list to stream (function )

[**vfscanf**](http://www.cplusplus.com/reference/cstdio/vfscanf/)

Read formatted data from stream into variable argument list (function )

[**vprintf**](http://www.cplusplus.com/reference/cstdio/vprintf/)

Print formatted data from variable argument list to stdout (function )

[**vscanf**](http://www.cplusplus.com/reference/cstdio/vscanf/)

Read formatted data into variable argument list (function )

[**vsnprintf**](http://www.cplusplus.com/reference/cstdio/vsnprintf/)

Write formatted data from variable argument list to sized buffer (function )

[**vsprintf**](http://www.cplusplus.com/reference/cstdio/vsprintf/)

Write formatted data from variable argument list to string (function )

[**vsscanf**](http://www.cplusplus.com/reference/cstdio/vsscanf/)

Read formatted data from string into variable argument list (function )

**Character input/output**:

[**fgetc**](http://www.cplusplus.com/reference/cstdio/fgetc/)

Get character from stream (function )

[**fgets**](http://www.cplusplus.com/reference/cstdio/fgets/)

Get string from stream (function )

[**fputc**](http://www.cplusplus.com/reference/cstdio/fputc/)

Write character to stream (function )

[**fputs**](http://www.cplusplus.com/reference/cstdio/fputs/)

Write string to stream (function )

[**getc**](http://www.cplusplus.com/reference/cstdio/getc/)

Get character from stream (function )

[**getchar**](http://www.cplusplus.com/reference/cstdio/getchar/)

Get character from stdin (function )

[**gets**](http://www.cplusplus.com/reference/cstdio/gets/)

Get string from stdin (function )

[**putc**](http://www.cplusplus.com/reference/cstdio/putc/)

Write character to stream (function )

[**putchar**](http://www.cplusplus.com/reference/cstdio/putchar/)

Write character to stdout (function )

[**puts**](http://www.cplusplus.com/reference/cstdio/puts/)

Write string to stdout (function )

[**ungetc**](http://www.cplusplus.com/reference/cstdio/ungetc/)

Unget character from stream (function )

**Direct input/output**:

[**fread**](http://www.cplusplus.com/reference/cstdio/fread/)

Read block of data from stream (function )

[**fwrite**](http://www.cplusplus.com/reference/cstdio/fwrite/)

Write block of data to stream (function )

**File positioning**:

[**fgetpos**](http://www.cplusplus.com/reference/cstdio/fgetpos/)

Get current position in stream (function )

[**fseek**](http://www.cplusplus.com/reference/cstdio/fseek/)

Reposition stream position indicator (function )

[**fsetpos**](http://www.cplusplus.com/reference/cstdio/fsetpos/)

Set position indicator of stream (function )

[**ftell**](http://www.cplusplus.com/reference/cstdio/ftell/)

Get current position in stream (function )

[**rewind**](http://www.cplusplus.com/reference/cstdio/rewind/)

Set position of stream to the beginning (function )

**Error-handling**:

[**clearerr**](http://www.cplusplus.com/reference/cstdio/clearerr/)

Clear error indicators (function )

[**feof**](http://www.cplusplus.com/reference/cstdio/feof/)

Check end-of-file indicator (function )

[**ferror**](http://www.cplusplus.com/reference/cstdio/ferror/)

Check error indicator (function )

[**perror**](http://www.cplusplus.com/reference/cstdio/perror/)

Print error message (function )

**Macros**

[**BUFSIZ**](http://www.cplusplus.com/reference/cstdio/BUFSIZ/)

Buffer size (constant )

[**EOF**](http://www.cplusplus.com/reference/cstdio/EOF/)

End-of-File (constant )

[**FILENAME\_MAX**](http://www.cplusplus.com/reference/cstdio/FILENAME_MAX/)

Maximum length of file names (constant )

[**FOPEN\_MAX**](http://www.cplusplus.com/reference/cstdio/FOPEN_MAX/)

Potential limit of simultaneous open streams (constant )

[**L\_tmpnam**](http://www.cplusplus.com/reference/cstdio/L_tmpnam/)

Minimum length for temporary file name (constant )

[**NULL**](http://www.cplusplus.com/reference/cstdio/NULL/)

Null pointer (macro )

[**TMP\_MAX**](http://www.cplusplus.com/reference/cstdio/TMP_MAX/)

Number of temporary files (constant )

Additionally: [\_IOFBF](http://www.cplusplus.com/setvbuf), [\_IOLBF](http://www.cplusplus.com/setvbuf), [\_IONBF](http://www.cplusplus.com/setvbuf) (used with [setvbuf](http://www.cplusplus.com/setvbuf))  
and [SEEK\_CUR](http://www.cplusplus.com/fseek), [SEEK\_END](http://www.cplusplus.com/fseek) and [SEEK\_SET](http://www.cplusplus.com/fseek) (used with [fseek](http://www.cplusplus.com/fseek)).

**Types**

[**FILE**](http://www.cplusplus.com/reference/cstdio/FILE/)

Object containing information to control a stream (type )

[**fpos\_t**](http://www.cplusplus.com/reference/cstdio/fpos_t/)

Object containing information to specify a position within a file (type )

[**size\_t**](http://www.cplusplus.com/reference/cstdio/size_t/)

Unsigned integral type (type )

header

# <cassert> (assert.h)

**C Diagnostics Library**

assert.h defines one macro function that can be used as a standard debugging tool:

### Macro functions

[**assert**](http://www.cplusplus.com/reference/cassert/assert/)

Evaluate assertion (macro )

header

# <cctype> (ctype.h)

**Character handling functions**

This header declares a set of functions to classify and transform individual characters.

### Functions

These functions take the int equivalent of one character as parameter and return an int that can either be another character or a value representing a boolean value: an int value of 0 means false, and an int value different from 0represents true.  
  
There are two sets of functions:

#### Character classification functions

They check whether the character passed as parameter belongs to a certain category:

[**isalnum**](http://www.cplusplus.com/reference/cctype/isalnum/)

Check if character is alphanumeric (function )

[**isalpha**](http://www.cplusplus.com/reference/cctype/isalpha/)

Check if character is alphabetic (function )

[**isblank**](http://www.cplusplus.com/reference/cctype/isblank/)

Check if character is blank (function )

[**iscntrl**](http://www.cplusplus.com/reference/cctype/iscntrl/)

Check if character is a control character (function )

[**isdigit**](http://www.cplusplus.com/reference/cctype/isdigit/)

Check if character is decimal digit (function )

[**isgraph**](http://www.cplusplus.com/reference/cctype/isgraph/)

Check if character has graphical representation (function )

[**islower**](http://www.cplusplus.com/reference/cctype/islower/)

Check if character is lowercase letter (function )

[**isprint**](http://www.cplusplus.com/reference/cctype/isprint/)

Check if character is printable (function )

[**ispunct**](http://www.cplusplus.com/reference/cctype/ispunct/)

Check if character is a punctuation character (function )

[**isspace**](http://www.cplusplus.com/reference/cctype/isspace/)

Check if character is a white-space (function )

[**isupper**](http://www.cplusplus.com/reference/cctype/isupper/)

Check if character is uppercase letter (function )

[**isxdigit**](http://www.cplusplus.com/reference/cctype/isxdigit/)

Check if character is hexadecimal digit (function )

#### Character conversion functions

Two functions that convert between letter cases:

[**tolower**](http://www.cplusplus.com/reference/cctype/tolower/)

Convert uppercase letter to lowercase (function )

[**toupper**](http://www.cplusplus.com/reference/cctype/toupper/)

Convert lowercase letter to uppercase (function )

For the first set, here is a map of how the original 127-character ASCII set is considered by each function (an x indicates that the function returns true on that character)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ASCII values** | **characters** | [**iscntrl**](http://www.cplusplus.com/iscntrl) | [**isblank**](http://www.cplusplus.com/isblank) | [**isspace**](http://www.cplusplus.com/isspace) | [**isupper**](http://www.cplusplus.com/isupper) | [**islower**](http://www.cplusplus.com/islower) | [**isalpha**](http://www.cplusplus.com/isalpha) | [**isdigit**](http://www.cplusplus.com/isdigit) | [**isxdigit**](http://www.cplusplus.com/isxdigit) | [**isalnum**](http://www.cplusplus.com/isalnum) | [**ispunct**](http://www.cplusplus.com/ispunct) | [**isgraph**](http://www.cplusplus.com/isgraph) | [**isprint**](http://www.cplusplus.com/isprint) |
| 0x00 .. 0x08 | NUL, (other control codes) | x |  |  |  |  |  |  |  |  |  |  |  |
| 0x09 | tab ('\t') | x | x | x |  |  |  |  |  |  |  |  |  |
| 0x0A .. 0x0D | (white-space control codes:'\f','\v','\n','\r') | x |  | x |  |  |  |  |  |  |  |  |  |
| 0x0E .. 0x1F | (other control codes) | x |  |  |  |  |  |  |  |  |  |  |  |
| 0x20 | space (' ') |  | x | x |  |  |  |  |  |  |  |  | x |
| 0x21 .. 0x2F | !"#$%&'()\*+,-./ |  |  |  |  |  |  |  |  |  | x | x | x |
| 0x30 .. 0x39 | 0123456789 |  |  |  |  |  |  | x | x | x |  | x | x |
| 0x3a .. 0x40 | :;<=>?@ |  |  |  |  |  |  |  |  |  | x | x | x |
| 0x41 .. 0x46 | ABCDEF |  |  |  | x |  | x |  | x | x |  | x | x |
| 0x47 .. 0x5A | GHIJKLMNOPQRSTUVWXYZ |  |  |  | x |  | x |  |  | x |  | x | x |
| 0x5B .. 0x60 | [\]^\_` |  |  |  |  |  |  |  |  |  | x | x | x |
| 0x61 .. 0x66 | abcdef |  |  |  |  | x | x |  | x | x |  | x | x |
| 0x67 .. 0x7A | ghijklmnopqrstuvwxyz |  |  |  |  | x | x |  |  | x |  | x | x |
| 0x7B .. 0x7E | {|}~ |  |  |  |  |  |  |  |  |  | x | x | x |
| 0x7F | (DEL) | x |  |  |  |  |  |  |  |  |  |  |  |

The characters in the extended character set (above 0x7F) may belong to diverse categories depending on the locale and the platform. As a general rule, [ispunct](http://www.cplusplus.com/ispunct), [isgraph](http://www.cplusplus.com/isgraph) and [isprint](http://www.cplusplus.com/isprint) return true on these for the standard C locale on most platforms supporting extended character sets.

header

# <cerrno> (errno.h)

**C Errors**

C Header that defines the following macro:

[**errno**](http://www.cplusplus.com/reference/cerrno/errno/)

Last error number (macro )

plus at least three additional macro constants: EDOM, ERANGE and EILSEQ (see [errno](http://www.cplusplus.com/errno) for more details).

header

# <cfenv> (fenv.h)

**Floating-point environment**

This header declares a set of functions and macros to access the *floating-point environment*, along with specific types.  
  
The floating-point environment maintains a series of *status flags* and specific *control modes*. Specific about the contents of the *floating-point environment* depend on the implementation, but the *status flags* generally include the *floating-point exceptions* and their associated information, and the *control modes* include at least the *rounding direction*.

### Functions

#### Floating-point exceptions

[**feclearexcept**](http://www.cplusplus.com/reference/cfenv/feclearexcept/)

Clear floating-point exceptions (function )

[**feraiseexcept**](http://www.cplusplus.com/reference/cfenv/feraiseexcept/)

Raise floating-point exception (function )

[**fegetexceptflag**](http://www.cplusplus.com/reference/cfenv/fegetexceptflag/)

Get floating-point exception flags (function )

[**fesetexceptflag**](http://www.cplusplus.com/reference/cfenv/fesetexceptflag/)

Set floating-point exception flags (function )

#### Rounding direction

[**fegetround**](http://www.cplusplus.com/reference/cfenv/fegetround/)

Get rounding direction mode (function )

[**fesetround**](http://www.cplusplus.com/reference/cfenv/fesetround/)

Set rounding direction mode (function )

#### Entire environment

[**fegetenv**](http://www.cplusplus.com/reference/cfenv/fegetenv/)

Get floating-point environment (function )

[**fesetenv**](http://www.cplusplus.com/reference/cfenv/fesetenv/)

Set floating-point environment (function )

[**feholdexcept**](http://www.cplusplus.com/reference/cfenv/feholdexcept/)

Hold floating-point exceptions (function )

[**feupdateenv**](http://www.cplusplus.com/reference/cfenv/feupdateenv/)

Update floating-point environment (function )

#### Other

[**fetestexcept**](http://www.cplusplus.com/reference/cfenv/fetestexcept/)

Test for floating-point exceptions (function )

### Types

[**fenv\_t**](http://www.cplusplus.com/reference/cfenv/fenv_t/)

Floating-point environment type (type )

[**fexcept\_t**](http://www.cplusplus.com/reference/cfenv/fexcept_t/)

Floating-point exceptions type (type )

### Macro constants

#### Floating-point exceptions

[**FE\_DIVBYZERO**](http://www.cplusplus.com/reference/cfenv/FE_DIVBYZERO/)

Pole error exception (macro )

[**FE\_INEXACT**](http://www.cplusplus.com/reference/cfenv/FE_INEXACT/)

Inexact result exception (macro )

[**FE\_INVALID**](http://www.cplusplus.com/reference/cfenv/FE_INVALID/)

Invalid argument exception (macro )

[**FE\_OVERFLOW**](http://www.cplusplus.com/reference/cfenv/FE_OVERFLOW/)

Overflow range error exception (macro )

[**FE\_UNDERFLOW**](http://www.cplusplus.com/reference/cfenv/FE_UNDERFLOW/)

Underflow range error exception (macro )

[**FE\_ALL\_EXCEPT**](http://www.cplusplus.com/reference/cfenv/FE_ALL_EXCEPT/)

All exceptions (macro )

#### Rounding directions

[**FE\_DOWNWARD**](http://www.cplusplus.com/reference/cfenv/FE_DOWNWARD/)

Downward rounding direction mode (macro )

[**FE\_TONEAREST**](http://www.cplusplus.com/reference/cfenv/FE_TONEAREST/)

To-nearest rounding direction mode (macro )

[**FE\_TOWARDZERO**](http://www.cplusplus.com/reference/cfenv/FE_TOWARDZERO/)

Toward-zero rounding direction mode (macro )

[**FE\_UPWARD**](http://www.cplusplus.com/reference/cfenv/FE_UPWARD/)

Upward rounding direction mode (macro )

#### Entire evnironment

[**FE\_DFL\_ENV**](http://www.cplusplus.com/reference/cfenv/FE_DFL_ENV/)

Default environment (macro )

### Pragmas

[**FENV\_ACCESS**](http://www.cplusplus.com/reference/cfenv/FENV_ACCESS/)

Access to Floating-point environment (pragma )

header

**<cfloat> (float.h)**

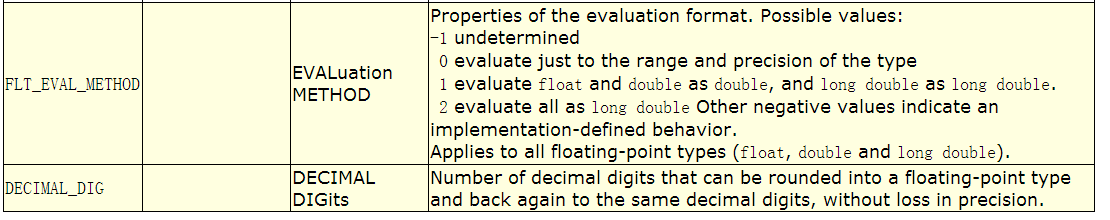
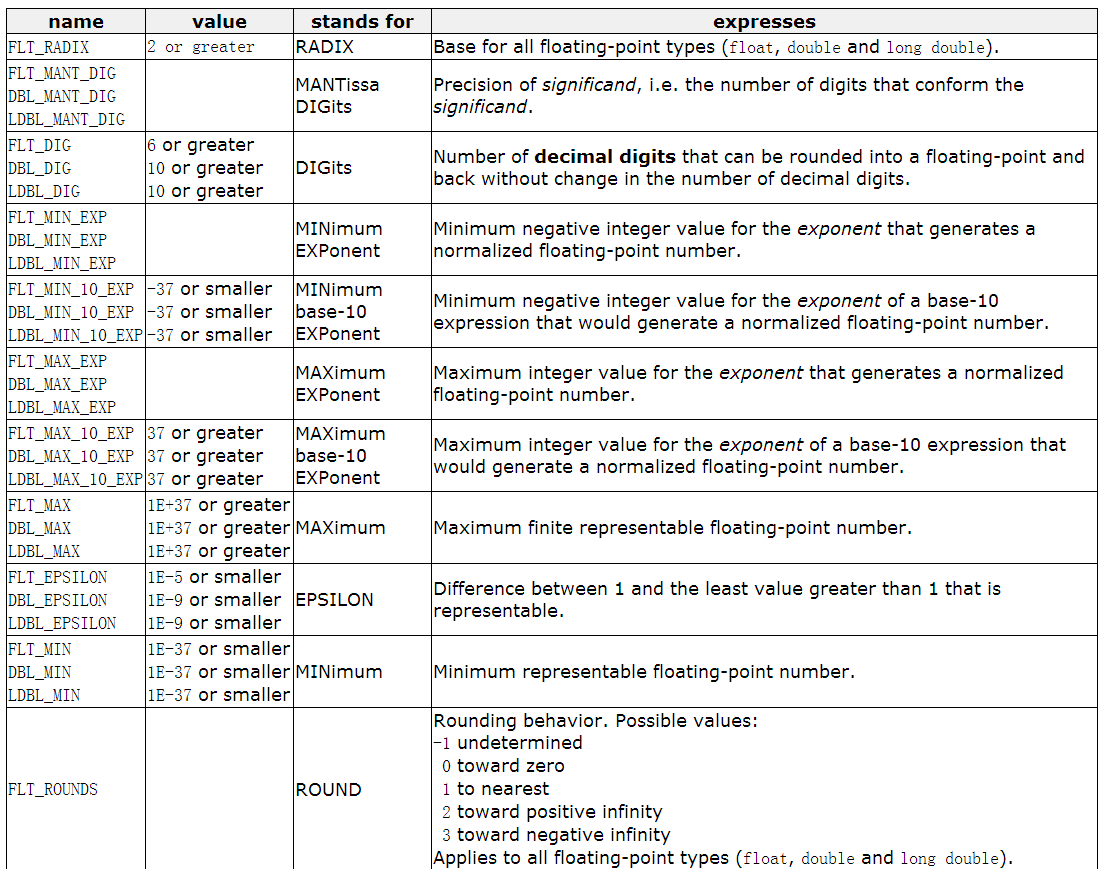
**Characteristics of floating-point types**

This header describes the characteristics of floating types for the specific system and compiler implementation used.  
  
A floating-point number is composed of four elements:

* a sign: either negative or non-negative
* a base (or radix): which expresses the different numbers that can be represented with a single digit (2 for binary, 10 for decimal, 16 for hexadecimal, and so on...)
* a significand (or mantissa): which is a series of digits of the aforementioned base. The number of digits in this series is what is known as *precision*.
* an exponent (also known as characteristic, or scale): which represents the offset of the significand, affecting the value in the following way:  
  *value of floating-point* = *significand* x *baseexponent*, with its corresponding sign.

**Macro constants**

The following panel shows the name of the different values defined in this header and their minimal or maximal values for all implementations (particula rimplementations may have values greater or smaller, as specified):  
  
When a group of macros exists prefixed by FLT\_, DBL\_ and LDBL\_, the one beginning with FLT\_ applies to the float type, the one with DBL\_ to double and the one with LDBL\_ to long double.



header

# <cinttypes> (inttypes.h)

**C integer types**

Header with library support for [*width-based integral types*](http://www.cplusplus.com/%3Ccstdint%3E).  
  
Including this header automatically includes also [<cstdint>](http://www.cplusplus.com/%3Ccstdint%3E) (which defines *width-based integral types*).

### Macros

The following macros expand to character string literals that contain a [printf](http://www.cplusplus.com/printf) or [scanf](http://www.cplusplus.com/scanf) specifier suitable to be used with specific [*width-based integral types*](http://www.cplusplus.com/%3Ccstdint%3E):

|  |  |  |
| --- | --- | --- |
| **macro** | **description** | **example** |
| PRI***x***MAX | [printf](http://www.cplusplus.com/printf) specifier for intmax\_t | PRIiMAX is the equivalent of i (in "%i") for intmax\_t values |
| PRI***xN*** | [printf](http://www.cplusplus.com/printf) specifier for int***N***\_t | PRId16 is the equivalent of d (in "%d") for int16\_t values |
| PRI***x***LEAST***N*** | [printf](http://www.cplusplus.com/printf) specifier for int\_least***N***\_t | PRIuLEAST32 is the equivalent of u (in "%u") for uint32\_t values |
| PRI***x***FAST***N*** | [printf](http://www.cplusplus.com/printf) specifier for int\_fast***N***\_t | PRIxFAST8 is the equivalent of x (in "%x") for uint8\_t values |
| PRI***x***PTR | [printf](http://www.cplusplus.com/printf) specifier for intptr\_t | PRIuPTR is the equivalent of u (in "%u") for uintptr\_t values |
| SCN***x***MAX | [scanf](http://www.cplusplus.com/scanf) specifier for intmax\_t | SCNiMAX is the equivalent of i (in "%i") for intmax\_t values |
| SCN***xN*** | [scanf](http://www.cplusplus.com/scanf) specifier for int***N***\_t | SCNd16 is the equivalent of d (in "%d") for int16\_t values |
| SCN***x***LEAST***N*** | [scanf](http://www.cplusplus.com/scanf) specifier for int\_least***N***\_t | SCNuLEAST32 is the equivalent of u (in "%u") for uint32\_t values |
| SCN***x***FAST***N*** | [scanf](http://www.cplusplus.com/scanf) specifier for int\_fast***N***\_t | SCNxFAST8 is the equivalent of x (in "%x") for uint8\_t values |
| SCN***x***PTR | [scanf](http://www.cplusplus.com/scanf) specifier for intptr\_t | SCNuPTR is the equivalent of u (in "%u") for uintptr\_t values |

Where:

* ***x*** is one of d, i, o,u or x (for the [printf](http://www.cplusplus.com/printf) specifiers this can also be an uppercase X).\*
* ***N*** is 8, 16, 32, 64, or any other type width supported by the library in [<cstdint>](http://www.cplusplus.com/%3Ccstdint%3E).

\* The specifier assumes the type to be signed for i and d, and unsigned for o, u, x and X.

### Functions

This header also declares the following functions, that adapt functions in [<cstdlib>](http://www.cplusplus.com/%3Ccstdlib%3E) and [<cwchar>](http://www.cplusplus.com/%3Ccwchar%3E) for [intmax\_t](http://www.cplusplus.com/intmax_t):

|  |  |
| --- | --- |
| **function** | **description** |
| imaxabs | equivalent to [abs](http://www.cplusplus.com/abs) for [intmax\_t](http://www.cplusplus.com/intmax_t): intmax\_t imaxabs (intmax\_t n); |
| imaxdiv | equivalent to [div](http://www.cplusplus.com/div) for [intmax\_t](http://www.cplusplus.com/intmax_t): imaxdiv\_t imaxdiv (intmax\_t numer, intmax\_t denom); |
| strtoimax | equivalent to [strtol](http://www.cplusplus.com/strtol) for [intmax\_t](http://www.cplusplus.com/intmax_t): intmax\_t strtoimax (const char\* str, char\*\* endptr, int base); |
| strtoumax | equivalent to [strtoul](http://www.cplusplus.com/strtoul) for [uintmax\_t](http://www.cplusplus.com/uintmax_t): uintmax\_t strtoumax (const char\* str, char\*\* endptr, int base); |
| wcstoimax | equivalent to [wcstol](http://www.cplusplus.com/wcstol) for [intmax\_t](http://www.cplusplus.com/intmax_t): intmax\_t wcstoimax (const wchar\_t\* wcs, wchar\_t\*\* endptr, int base); |
| wcstoumax | equivalent to [wcstoul](http://www.cplusplus.com/wcstoul) for [uintmax\_t](http://www.cplusplus.com/uintmax_t): uintmax\_t wcstoumax (const wchar\_t\* wcs, wchar\_t\*\* endptr, int base); |

In C++, including this header also provides overloads of [abs](http://www.cplusplus.com/abs) and [div](http://www.cplusplus.com/div) for [intmax\_t](http://www.cplusplus.com/intmax_t) (equivalent to the functions above) unless intmax\_t is an alias of a [*fundamental type*](http://www.cplusplus.com/is_fundamental).

### Types

|  |  |
| --- | --- |
| **Type** | **description** |
| imaxdiv\_t | Type returned by imaxdiv, which is the [div\_t](http://www.cplusplus.com/div_t) equivalent for [intmax\_t](http://www.cplusplus.com/intmax_t). |

### Example

|  |  |  |
| --- | --- | --- |
| 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 | /\* example of <inttypes.h> \*/  #include <stdio.h> /\* printf, scanf, fgets, stdin, NULL \*/  #include <stdint.h> /\* intmax\_t \*/  #include <inttypes.h> /\* strtoimax, PRIdMAX, SCNdMAX \*/  int main ()  {  char buffer[80];  intmax\_t foo,bar;  printf ("Please, enter a number: ");  fgets (buffer,80,stdin);  foo = strtoimax (buffer,NULL,10);  printf ("Thanks for entering %" PRIdMAX ".\n", foo);  printf ("Please, enter another number: ");  scanf ("%" SCNdMAX,&bar);  printf ("%" PRIdMAX " by %" PRIdMAX " is %" PRIdMAX, foo, bar, foo\*bar);  return 0;  } | [Edit & Run](http://www.cplusplus.com/reference/cinttypes/) |

Possible output:

|  |
| --- |
| Please, enter a number: 10  Thanks for entering 10.  Please, enter another number: 20  10 by 20 is 200 |

header

# <ciso646> (iso646.h)

**ISO 646 Alternative operator spellings**

This header defines eleven macro constants with alternative spellings for those C++ operators not supported by the ISO646 standard character set:

|  |  |
| --- | --- |
| **macro** | **operator** |
| and | && |
| and\_eq | &= |
| bitand | & |
| bitor | | |
| compl | ~ |
| not | ! |
| not\_eq | != |
| or | || |
| or\_eq | |= |
| xor | ^ |
| xor\_eq | ^= |

In C++, reserved words exist with the same names as these macros and are treated as aliases of their respective operator. Therefore the inclusion of this header has no effect in C++, and is not necessary in order to use the alternative names.

header

# <climits> (limits.h)

**Sizes of integral types**

This header defines constants with the limits of fundamental integral types for the specific system and compiler implementation used.  
  
The limits for fundamental floating-point types are defined in [<cfloat>](http://www.cplusplus.com/%3Ccfloat%3E) (<float.h>).  
The limits for width-specific integral types and other typedef types are defined in [<cstdint>](http://www.cplusplus.com/%3Ccstdint%3E) (<stdint.h>).

### Macro constants

|  |  |  |
| --- | --- | --- |
| **name** | **expresses** | **value\*** |
| CHAR\_BIT | Number of bits in a char object (byte) | 8 or greater\* |
| SCHAR\_MIN | Minimum value for an object of type signed char | -127 (-27+1) or less\* |
| SCHAR\_MAX | Maximum value for an object of type signed char | 127 (27-1) or greater\* |
| UCHAR\_MAX | Maximum value for an object of type unsigned char | 255 (28-1) or greater\* |
| CHAR\_MIN | Minimum value for an object of type char | either SCHAR\_MIN or 0 |
| CHAR\_MAX | Maximum value for an object of type char | either SCHAR\_MAX or UCHAR\_MAX |
| MB\_LEN\_MAX | Maximum number of bytes in a multibyte character, for any locale | 1 or greater\* |
| SHRT\_MIN | Minimum value for an object of type short int | -32767 (-215+1) or less\* |
| SHRT\_MAX | Maximum value for an object of type short int | 32767 (215-1) or greater\* |
| USHRT\_MAX | Maximum value for an object of type unsigned short int | 65535 (216-1) or greater\* |
| INT\_MIN | Minimum value for an object of type int | -32767 (-215+1) or less\* |
| INT\_MAX | Maximum value for an object of type int | 32767 (215-1) or greater\* |
| UINT\_MAX | Maximum value for an object of type unsigned int | 65535 (216-1) or greater\* |
| LONG\_MIN | Minimum value for an object of type long int | -2147483647 (-231+1) or less\* |
| LONG\_MAX | Maximum value for an object of type long int | 2147483647 (231-1) or greater\* |
| ULONG\_MAX | Maximum value for an object of type unsigned long int | 4294967295 (232-1) or greater\* |
| LLONG\_MIN | Minimum value for an object of type long long int | -9223372036854775807 (-263+1) or less\* |
| LLONG\_MAX | Maximum value for an object of type long long int | 9223372036854775807 (263-1) or greater\* |
| ULLONG\_MAX | Maximum value for an object of type unsigned long long int | 18446744073709551615 (264-1) or greater\* |

\* the actual value depends on the particular system and library implementation, but shall reflect the limits of these types in the target platform.

### Compatibility

LLONG\_MIN, LLONG\_MAX and ULLONG\_MAX are defined for libraries complying with the C standard of 1999 or later (which only includes the C++ standard since 2011: C++11).

header

# <clocale> (locale.h)

**C localization library**

The C language supports localization specific settings, such as culture-specific date formats or country-specific currency symbols.  
  
Each system and specific compiler implementation may provide different choices of locales to be selected (using function[setlocale](http://www.cplusplus.com/setlocale)), but at least two locales are available to choose for any C program:

* The "C" locale is the minimal locale. It is a rather neutral locale which has the same settings across all systems and compilers, and therefore the exact results of a program using this locale are predictable. This is the locale used by default on all C programs.
* The system's default locale (which is specified by an empty C-string: ""). It is the locale configuration provided by the environment where the application is running. This usually contains more localization information than the "C" locale.

This header declares the struct [lconv](http://www.cplusplus.com/lconv) and the functions [setlocale](http://www.cplusplus.com/setlocale) and [localeconv](http://www.cplusplus.com/localeconv), along with several macros to be used with them. These are used to define locale-specific information.  
  
Notice that locale configuration affects the behavior of many functions within the standard C library:

* In [<cstring>](http://www.cplusplus.com/%3Ccstring%3E) (<string.h>), functions [strcoll](http://www.cplusplus.com/strcoll) and [strxfrm](http://www.cplusplus.com/strxfrm) are affected by character transformation rules.
* In [<cctype>](http://www.cplusplus.com/%3Ccctype%3E) (<ctype.h>), all functions except [isdigit](http://www.cplusplus.com/isdigit) and [isxdigit](http://www.cplusplus.com/isxdigit) are affected by the extended character set selected.
* In [<cstdio>](http://www.cplusplus.com/%3Ccstdio%3E) (<stdio.h>), formatted input/output operations are affected by character transformation rules and decimal-point character set in the numeric formatting settings.
* In [<ctime>](http://www.cplusplus.com/%3Cctime%3E) (<time.h>), the function [strftime](http://www.cplusplus.com/strftime) is affected by the time formatting settings.
* In this header, it affects the value returned by its functions [setlocale](http://www.cplusplus.com/setlocale) and [localeconv](http://www.cplusplus.com/localeconv).

Some locale information is also provided to programs to better render its output in a locale-sensitive manner ([localeconv](http://www.cplusplus.com/localeconv)).

### Types

[**struct lconv**](http://www.cplusplus.com/reference/clocale/lconv/)

Formatting info for numeric values (type )

### Functions

[**setlocale**](http://www.cplusplus.com/reference/clocale/setlocale/)

Set or retrieve locale (function )

[**localeconv**](http://www.cplusplus.com/reference/clocale/localeconv/)

Get locale formatting parameters for quantities (function )

header

# <cmath> (math.h)

**C numerics library**

Header <cmath> declares a set of functions to compute common mathematical operations and transformations:

### Functions

#### Trigonometric functions

[**cos**](http://www.cplusplus.com/reference/cmath/cos/)

Compute cosine (function )

[**sin**](http://www.cplusplus.com/reference/cmath/sin/)

Compute sine (function )

[**tan**](http://www.cplusplus.com/reference/cmath/tan/)

Compute tangent (function )

[**acos**](http://www.cplusplus.com/reference/cmath/acos/)

Compute arc cosine (function )

[**asin**](http://www.cplusplus.com/reference/cmath/asin/)

Compute arc sine (function )

[**atan**](http://www.cplusplus.com/reference/cmath/atan/)

Compute arc tangent (function )

[**atan2**](http://www.cplusplus.com/reference/cmath/atan2/)

Compute arc tangent with two parameters (function )

#### Hyperbolic functions

[**cosh**](http://www.cplusplus.com/reference/cmath/cosh/)

Compute hyperbolic cosine (function )

[**sinh**](http://www.cplusplus.com/reference/cmath/sinh/)

Compute hyperbolic sine (function )

[**tanh**](http://www.cplusplus.com/reference/cmath/tanh/)

Compute hyperbolic tangent (function )

[**acosh**](http://www.cplusplus.com/reference/cmath/acosh/)

Compute arc hyperbolic cosine (function )

[**asinh**](http://www.cplusplus.com/reference/cmath/asinh/)

Compute arc hyperbolic sine (function )

[**atanh**](http://www.cplusplus.com/reference/cmath/atanh/)

Compute arc hyperbolic tangent (function )

#### Exponential and logarithmic functions

[**exp**](http://www.cplusplus.com/reference/cmath/exp/)

Compute exponential function (function )

[**frexp**](http://www.cplusplus.com/reference/cmath/frexp/)

Get significand and exponent (function )

[**ldexp**](http://www.cplusplus.com/reference/cmath/ldexp/)

Generate value from significand and exponent (function )

[**log**](http://www.cplusplus.com/reference/cmath/log/)

Compute natural logarithm (function )

[**log10**](http://www.cplusplus.com/reference/cmath/log10/)

Compute common logarithm (function )

[**modf**](http://www.cplusplus.com/reference/cmath/modf/)

Break into fractional and integral parts (function )

[**exp2**](http://www.cplusplus.com/reference/cmath/exp2/)

Compute binary exponential function (function )

[**expm1**](http://www.cplusplus.com/reference/cmath/expm1/)

Compute exponential minus one (function )

[**ilogb**](http://www.cplusplus.com/reference/cmath/ilogb/)

Integer binary logarithm (function )

[**log1p**](http://www.cplusplus.com/reference/cmath/log1p/)

Compute logarithm plus one (function )

[**log2**](http://www.cplusplus.com/reference/cmath/log2/)

Compute binary logarithm (function )

[**logb**](http://www.cplusplus.com/reference/cmath/logb/)

Compute floating-point base logarithm (function )

[**scalbn**](http://www.cplusplus.com/reference/cmath/scalbn/)

Scale significand using floating-point base exponent (function )

[**scalbln**](http://www.cplusplus.com/reference/cmath/scalbln/)

Scale significand using floating-point base exponent (long) (function )

#### Power functions

[**pow**](http://www.cplusplus.com/reference/cmath/pow/)

Raise to power (function )

[**sqrt**](http://www.cplusplus.com/reference/cmath/sqrt/)

Compute square root (function )

[**cbrt**](http://www.cplusplus.com/reference/cmath/cbrt/)

Compute cubic root (function )

[**hypot**](http://www.cplusplus.com/reference/cmath/hypot/)

Compute hypotenuse (function )

#### Error and gamma functions

[**erf**](http://www.cplusplus.com/reference/cmath/erf/)

Compute error function (function )

[**erfc**](http://www.cplusplus.com/reference/cmath/erfc/)

Compute complementary error function (function )

[**tgamma**](http://www.cplusplus.com/reference/cmath/tgamma/)

Compute gamma function (function )

[**lgamma**](http://www.cplusplus.com/reference/cmath/lgamma/)

Compute log-gamma function (function )

#### Rounding and remainder functions

[**ceil**](http://www.cplusplus.com/reference/cmath/ceil/)

Round up value (function )

[**floor**](http://www.cplusplus.com/reference/cmath/floor/)

Round down value (function )

[**fmod**](http://www.cplusplus.com/reference/cmath/fmod/)

Compute remainder of division (function )

[**trunc**](http://www.cplusplus.com/reference/cmath/trunc/)

Truncate value (function )

[**round**](http://www.cplusplus.com/reference/cmath/round/)

Round to nearest (function )

[**lround**](http://www.cplusplus.com/reference/cmath/lround/)

Round to nearest and cast to long integer (function )

[**llround**](http://www.cplusplus.com/reference/cmath/llround/)

Round to nearest and cast to long long integer (function )

[**rint**](http://www.cplusplus.com/reference/cmath/rint/)

Round to integral value (function )

[**lrint**](http://www.cplusplus.com/reference/cmath/lrint/)

Round and cast to long integer (function )

[**llrint**](http://www.cplusplus.com/reference/cmath/llrint/)

Round and cast to long long integer (function )

[**nearbyint**](http://www.cplusplus.com/reference/cmath/nearbyint/)

Round to nearby integral value (function )

[**remainder**](http://www.cplusplus.com/reference/cmath/remainder/)

Compute remainder (IEC 60559) (function )

[**remquo**](http://www.cplusplus.com/reference/cmath/remquo/)

Compute remainder and quotient (function )

#### Floating-point manipulation functions

[**copysign**](http://www.cplusplus.com/reference/cmath/copysign/)

Copy sign (function )

[**nan**](http://www.cplusplus.com/reference/cmath/nan-function/)

Generate quiet NaN (function )

[**nextafter**](http://www.cplusplus.com/reference/cmath/nextafter/)

Next representable value (function )

[**nexttoward**](http://www.cplusplus.com/reference/cmath/nexttoward/)

Next representable value toward precise value (function )

#### Minimum, maximum, difference functions

[**fdim**](http://www.cplusplus.com/reference/cmath/fdim/)

Positive difference (function )

[**fmax**](http://www.cplusplus.com/reference/cmath/fmax/)

Maximum value (function )

[**fmin**](http://www.cplusplus.com/reference/cmath/fmin/)

Minimum value (function )

#### Other functions

[**fabs**](http://www.cplusplus.com/reference/cmath/fabs/)

Compute absolute value (function )

[**abs**](http://www.cplusplus.com/reference/cmath/abs/)

Compute absolute value (function )

[**fma**](http://www.cplusplus.com/reference/cmath/fma/)

Multiply-add (function )

### Macros / Functions

These are implemented as macros in C and as functions in C++:

#### Classification macro / functions

[**fpclassify**](http://www.cplusplus.com/reference/cmath/fpclassify/)

Classify floating-point value (macro/function )

[**isfinite**](http://www.cplusplus.com/reference/cmath/isfinite/)

Is finite value (macro )

[**isinf**](http://www.cplusplus.com/reference/cmath/isinf/)

Is infinity (macro/function )

[**isnan**](http://www.cplusplus.com/reference/cmath/isnan/)

Is Not-A-Number (macro/function )

[**isnormal**](http://www.cplusplus.com/reference/cmath/isnormal/)

Is normal (macro/function )

[**signbit**](http://www.cplusplus.com/reference/cmath/signbit/)

Sign bit (macro/function )

#### Comparison macro / functions

[**isgreater**](http://www.cplusplus.com/reference/cmath/isgreater/)

Is greater (macro )

[**isgreaterequal**](http://www.cplusplus.com/reference/cmath/isgreaterequal/)

Is greater or equal (macro )

[**isless**](http://www.cplusplus.com/reference/cmath/isless/)

Is less (macro )

[**islessequal**](http://www.cplusplus.com/reference/cmath/islessequal/)

Is less or equal (macro )

[**islessgreater**](http://www.cplusplus.com/reference/cmath/islessgreater/)

Is less or greater (macro )

[**isunordered**](http://www.cplusplus.com/reference/cmath/isunordered/)

Is unordered (macro )

### Macro constants

[**math\_errhandling**](http://www.cplusplus.com/reference/cmath/math_errhandling/)

Error handling (macro )

[**INFINITY**](http://www.cplusplus.com/reference/cmath/INFINITY/)

Infinity (constant )

[**NAN**](http://www.cplusplus.com/reference/cmath/NAN/)

Not-A-Number (constant )

[**HUGE\_VAL**](http://www.cplusplus.com/reference/cmath/HUGE_VAL/)

Huge value (constant )

[**HUGE\_VALF**](http://www.cplusplus.com/reference/cmath/HUGE_VALF/)

Huge float value

[**HUGE\_VALL**](http://www.cplusplus.com/reference/cmath/HUGE_VALL/)

Huge long double value (constant )

This header also defines the following macro constants (since C99/C++11):

|  |  |  |
| --- | --- | --- |
| **macro** | **type** | **description** |
| [MATH\_ERRNO](http://www.cplusplus.com/MATH_ERRNO)  [MATH\_ERREXCEPT](http://www.cplusplus.com/MATH_ERREXCEPT) | int | Bitmask value with the possible values [math\_errhandling](http://www.cplusplus.com/math_errhandling) can take. |
| [FP\_FAST\_FMA](http://www.cplusplus.com/FP_FAST_FMA)  [FP\_FAST\_FMAF](http://www.cplusplus.com/FP_FAST_FMAF)  [FP\_FAST\_FMAL](http://www.cplusplus.com/FP_FAST_FMAL) | int | Each, if defined, identifies for which type [fma](http://www.cplusplus.com/fma) is at least as efficient as x\*y+z. |
| [FP\_INFINITE](http://www.cplusplus.com/FP_INFINITE)  [FP\_NAN](http://www.cplusplus.com/FP_NAN)  [FP\_NORMAL](http://www.cplusplus.com/FP_NORMAL)  [FP\_SUBNORMAL](http://www.cplusplus.com/FP_SUBNORMAL)  [FP\_ZERO](http://www.cplusplus.com/FP_ZERO) | int | The possible values returned by [fpclassify](http://www.cplusplus.com/fpclassify). |
| [FP\_ILOGB0](http://www.cplusplus.com/FP_ILOGB0)  [FP\_ILOGBNAN](http://www.cplusplus.com/FP_ILOGBNAN) | int | Special values the [ilogb](http://www.cplusplus.com/ilogb) function may return. |

### Types

[**double\_t**](http://www.cplusplus.com/reference/cmath/double_t/)

Floating-point type (type )

[**float\_t**](http://www.cplusplus.com/reference/cmath/float_t/)

Floating-point type (type )

header

# <csetjmp> (setjmp.h)

**Non local jumps**

The tools provided through this header file allow the programmer to bypass the normal function call and return discipline, by providing the means to perform jumps preserving the calling environment.  
  
The header provides, a function, a macro with functional form and a specific type:

### Functions

[**longjmp**](http://www.cplusplus.com/reference/csetjmp/longjmp/)

Long jump (function )

### Macro functions

[**setjmp**](http://www.cplusplus.com/reference/csetjmp/setjmp/)

Save calling environment for long jump (macro )

### Types

[**jmp\_buf**](http://www.cplusplus.com/reference/csetjmp/jmp_buf/)

Type to hold information to restore calling environment (type )

header

# <csignal> (signal.h)

**C library to handle signals**

Some running environments use *signals* to inform running processes of certain events. These events may be related to errors performed by the program code, like a wrong arithmetical operation or to exceptional situations, such as a request to interrupt the program.  
  
Signals generally represent situations where the program has either been requested to terminate or an unrecoverable error has happened, therefore handling a signal allows for either perform pre-termination cleanup operations or try to recover from the error in some way.  
  
Not all running environments are required to generate automatic signals in the cases for which they are designed in the standard C library, and some other environments not only generate these but also many more specific signals. But in any case, all signals generated explicitly with a call to function [raise](http://www.cplusplus.com/raise) are delivered to its corresponding signal handler.

### Functions

[**signal**](http://www.cplusplus.com/reference/csignal/signal/)

Set function to handle signal (function )

[**raise**](http://www.cplusplus.com/reference/csignal/raise/)

Generates a signal (function )

### Types

[**sig\_atomic\_t**](http://www.cplusplus.com/reference/csignal/sig_atomic_t/)

Integral type (type )

### Macro constants

|  |  |  |
| --- | --- | --- |
| **type** | **macro** | **signal** |
| int(signals) | SIGABRT | (Signal Abort) Abnormal termination, such as is initiated by the [abort](http://www.cplusplus.com/abort) function. |
| SIGFPE | (Signal Floating-Point Exception) Erroneous arithmetic operation, such as zero divide or an operation resulting in overflow (not necessarily with a floating-point operation). |
| SIGILL | (Signal Illegal Instruction) Invalid function image, such as an illegal instruction. This is generally due to a corruption in the code or to an attempt to execute data. |
| SIGINT | (Signal Interrupt) Interactive attention signal. Generally generated by the application user. |
| SIGSEGV | (Signal Segmentation Violation) Invalid access to storage: When a program tries to read or write outside the memory it has allocated. |
| SIGTERM | (Signal Terminate) Termination request sent to program. |
| *functions*(handlers) | SIG\_DFL | Default handling: The signal is handled by the default action for that particular signal. |
| SIG\_IGN | Ignore Signal: The signal is ignored. |
| SIG\_ERR | Special return value indicating failure. |

See function [signal](http://www.cplusplus.com/signal) for more information.

header

# <cstdarg> (stdarg.h)

**Variable arguments handling**

This header defines macros to access the individual arguments of a list of unnamed arguments whose number and types are not known to the called function.  
  
A function may accept a varying number of additional arguments without corresponding parameter declarations by including a comma and three dots (,...) after its regular named parameters:  
  
return\_type function\_name ( parameter\_declarations , ... );   
To access these additional arguments the macros [va\_start](http://www.cplusplus.com/va_start), [va\_arg](http://www.cplusplus.com/va_arg) and [va\_end](http://www.cplusplus.com/va_end), declared in this header, can be used:

* First, [va\_start](http://www.cplusplus.com/va_start) initializes the list of variable arguments as a [va\_list](http://www.cplusplus.com/va_list).
* Subsequent executions of [va\_arg](http://www.cplusplus.com/va_arg) yield the values of the additional arguments in the same order as passed to the function.
* Finally, [va\_end](http://www.cplusplus.com/va_end) shall be executed before the function returns.

### Types

[**va\_list**](http://www.cplusplus.com/reference/cstdarg/va_list/)

Type to hold information about variable arguments (type )

### Macro functions

[**va\_start**](http://www.cplusplus.com/reference/cstdarg/va_start/)

Initialize a variable argument list (macro )

[**va\_arg**](http://www.cplusplus.com/reference/cstdarg/va_arg/)

Retrieve next argument (macro )

[**va\_end**](http://www.cplusplus.com/reference/cstdarg/va_end/)

End using variable argument list (macro )

[**va\_copy**](http://www.cplusplus.com/reference/cstdarg/va_copy/)

Copy variable argument list (macro )

header

# <cstdbool> (stdbool.h)

**Boolean type**

The purpose in C of this header is to add a bool type and the true and false values as macro definitions.  
  
In C++, which supports those directly, the header simply contains a macro that can be used to check if the type is supported:

### Macro constants

|  |  |  |
| --- | --- | --- |
| **Macro** | **description** | **defined as** |
| \_\_bool\_true\_false\_are\_defined | Specifies whether bool, true and false are defined | 1 |

header

# <cstddef> (stddef.h)

**C Standard definitions**

This header defines several types implicitly generated or used by certain language expressions.

### Types

[**ptrdiff\_t**](http://www.cplusplus.com/reference/cstddef/ptrdiff_t/)

Result of pointer subtraction (type )

[**size\_t**](http://www.cplusplus.com/reference/cstddef/size_t/)

Unsigned integral type (type )

[**max\_align\_t**](http://www.cplusplus.com/reference/cstddef/max_align_t/)

Type with widest scalar alignment (type )

[**nullptr\_t**](http://www.cplusplus.com/reference/cstddef/nullptr_t/)

Null pointer type (C++) (type )

In C, this header also includes the declaration of the wchar\_t type (wide character type), which in C++ is a *language keyword* that identifies a distinct fundamental type (no header inclusion required in C++).

### Macro functions

[**offsetof**](http://www.cplusplus.com/reference/cstddef/offsetof/)

Return member offset (macro )

### Macro constants

[**NULL**](http://www.cplusplus.com/reference/cstddef/NULL/)

Null pointer (macro )

header

# <cstdint> (stdint.h)

**Integer types**

This header defines a set of integral type aliases with specific width requirements, along with macros specifying their limits and macro functions to create values of these types.

### Types

The following are typedefs of *fundamental integral types* or *extended integral types*.

|  |  |  |
| --- | --- | --- |
| **signed type** | **unsigned type** | **description** |
| intmax\_t | uintmax\_t | Integer type with the maximum width supported. |
| int8\_t | uint8\_t | Integer type with a width of exactly 8, 16, 32, or 64 bits. For signed types, negative values are represented using 2's complement. No padding bits. **Optional:** These typedefs are not defined if no types with such characteristics exist.\* |
| int16\_t | uint16\_t |
| int32\_t | uint32\_t |
| int64\_t | uint64\_t |
| int\_least8\_t | uint\_least8\_t | Integer type with a minimum of 8, 16, 32, or 64 bits. No other integer type exists with lesser size and at least the specified width. |
| int\_least16\_t | uint\_least16\_t |
| int\_least32\_t | uint\_least32\_t |
| int\_least64\_t | uint\_least64\_t |
| int\_fast8\_t | uint\_fast8\_t | Integer type with a minimum of 8, 16, 32, or 64 bits. At least as fast as any other integer type with at least the specified width. |
| int\_fast16\_t | uint\_fast16\_t |
| int\_fast32\_t | uint\_fast32\_t |
| int\_fast64\_t | uint\_fast64\_t |
| intptr\_t | uintptr\_t | Integer type capable of holding a value converted from a void pointer and then be converted back to that type with a value that compares equal to the original pointer. **Optional:** These typedefs may not be defined in some library implementations.\* |

Some of these typedefs may denote the same types. Therefore, function overloads should not rely on these being different.  
  
\* Notice that some types are optional (and thus, with no portability guarantees). A particular library implementation may also define additional types with other widths supported by its system. In any case, if either the signed or the unsigned version is defined, both the signed and unsigned versions are defined.

### Macros

#### Limits of cstdint types

|  |  |  |
| --- | --- | --- |
| **Macro** | **description** | **defined as** |
| INTMAX\_MIN | Minimum value of intmax\_t | -(263-1), or lower |
| INTMAX\_MAX | Maximum value of intmax\_t | 263-1, or higher |
| UINTMAX\_MAX | Maximum value of uintmax\_t | 264-1, or higher |
| INT*N*\_MIN | Minimum value of exact-width signed type | Exactly -2(*N*-1) |
| INT*N*\_MAX | Maximum value of exact-width signed type | Exactly 2(*N*-1)-1 |
| UINT*N*\_MAX | Maximum value of exact-width unsigned type | Exactly 2*N*-1 |
| INT\_LEAST*N*\_MIN | Minimum value of minimum-width signed type | -(2(*N*-1)-1), or lower |
| INT\_LEAST*N*\_MAX | Maximum value of minimum-width signed type | 2(*N*-1)-1, or higher |
| UINT\_LEAST*N*\_MAX | Maximum value of minimum-width unsigned type | 2*N*-1, or higher |
| INT\_FAST*N*\_MIN | Minimum value of fastest minimum-width signed type | -(2(*N*-1)-1), or lower |
| INT\_FAST*N*\_MAX | Maximum value of fastest minimum-width signed type | 2(*N*-1)-1, or higher |
| UINT\_FAST*N*\_MAX | Maximum value of fastest minimum-width unsigned type | 2*N*-1, or higher |
| INTPTR\_MIN | Minimum value of intptr\_t | -(215-1), or lower |
| INTPTR\_MAX | Maximum value of intptr\_t | 215-1, or higher |
| UINTPTR\_MAX | Maximum value of uintptr\_t | 216-1, or higher |

Where *N* is one in 8, 16, 32, 64, or any other type width supported by the library.  
  
Only the macros corresponding to types supported by the library are defined.

#### Limits of other types

Limits of other standard integral types:

|  |  |  |
| --- | --- | --- |
| **Macro** | **description** | **defined as** |
| SIZE\_MAX | Maximum value of [size\_t](http://www.cplusplus.com/size_t) | 264-1, or higher |
| PTRDIFF\_MIN | Minimum value of [ptrdiff\_t](http://www.cplusplus.com/ptrdiff_t) | -(216-1), or lower |
| PTRDIFF\_MAX | Maximum value of [ptrdiff\_t](http://www.cplusplus.com/ptrdiff_t) | 216-1, or higher |
| SIG\_ATOMIC\_MIN | Minimum value of [sig\_atomic\_t](http://www.cplusplus.com/sig_atomic_t) | if [sig\_atomic\_t](http://www.cplusplus.com/sig_atomic_t) is signed: -127, or lower if [sig\_atomic\_t](http://www.cplusplus.com/sig_atomic_t) is unsigned: 0 |
| SIG\_ATOMIC\_MAX | Maximum value of [sig\_atomic\_t](http://www.cplusplus.com/sig_atomic_t) | if [sig\_atomic\_t](http://www.cplusplus.com/sig_atomic_t) is signed: 127, or higher if [sig\_atomic\_t](http://www.cplusplus.com/sig_atomic_t) is unsigned: 255, or higher |
| WCHAR\_MIN | Minimum value of wchar\_t | if wchar\_t is signed: -127, or lower if wchar\_t is unsigned: 0 |
| WCHAR\_MAX | Maximum value of wchar\_t | if wchar\_t is signed: 127, or higher if wchar\_t is unsigned: 255, or higher |
| WINT\_MIN | Minimum value of wint\_t | if wint\_t is signed: -32767, or lower if wint\_t is unsigned: 0 |
| WINT\_MAX | Maximum value of wint\_t | if wint\_t is signed: 32767, or higher if wint\_t is unsigned: 65535, or higher |

#### Function-like macros

These function-like macros expand to integer constants suitable to initialize objects of the types above:

|  |  |
| --- | --- |
| **Macro** | **description** |
| INTMAX\_C | expands to a value of type intmax\_t |
| UINTMAX\_C | expands to a value of type uintmax\_t |
| INT*N*\_C | expands to a value of type int\_least*N*\_t |
| UINT*N*\_C | expands to a value of type uint\_least*N*\_t |

For example:

|  |  |  |
| --- | --- | --- |
|  | INTMAX\_C(2012) // expands to 2012LL or similar |  |

header

# <cstdio> (stdio.h)

**C library to perform Input/Output operations**

Input and Output operations can also be performed in C++ using the **C** **St**andar**d** **I**nput and **O**utput Library (**cstdio**, known as stdio.h in the C language). This library uses what are called *streams* to operate with physical devices such as keyboards, printers, terminals or with any other type of files supported by the system. Streams are an abstraction to interact with these in an uniform way; All streams have similar properties independently of the individual characteristics of the physical media they are associated with.  
  
Streams are handled in the cstdio library as pointers to [FILE](http://www.cplusplus.com/FILE) objects. A pointer to a [FILE](http://www.cplusplus.com/FILE) object uniquely identifies a stream, and is used as a parameter in the operations involving that stream.  
  
There also exist three standard streams: [stdin](http://www.cplusplus.com/stdin), [stdout](http://www.cplusplus.com/stdout) and [stderr](http://www.cplusplus.com/stderr), which are automatically created and opened for all programs using the library.

### Stream properties

Streams have some properties that define which functions can be used on them and how these will treat the data input or output through them. Most of these properties are defined at the moment the stream is associated with a file (opened) using the [fopen](http://www.cplusplus.com/fopen) function:

***Read/Write Access***

Specifies whether the stream has read or write access (or both) to the physical media they are associated with.

***Text / Binary***

Text streams are thought to represent a set of text lines, each one ending with a new-line character. Depending on the environment where the application is run, some character translation may occur with text streams to adapt some special characters to the text file specifications of the environment. A binary stream, on the other hand, is a sequence of characters written or read from the physical media with no translation, having a one-to-one correspondence with the characters read or written to the stream.

***Buffer***

A buffer is a block of memory where data is accumulated before being physically read or written to the associated file or device. Streams can be either *fully buffered*, *line buffered* or *unbuffered*. On fully buffered streams, data is read/written when the buffer is filled, on line buffered streams this happens when a new-line character is encountered, and on unbuffered streams characters are intended to be read/written as soon as possible.

***Orientation***

On opening, streams have no orientation. As soon as an input/output operation is performed on them, they become either *byte-oriented* or *wide-oriented*, depending on the operation performed (generally, functions defined in <cstdio> are *byte-oriented*, while functions in [<cwchar>](http://www.cplusplus.com/cwchar) are *wide-oriented*). See [cwchar](http://www.cplusplus.com/cwchar) for more info.

### Indicators

Streams have certain internal indicators that specify their current state and which affect the behavior of some input and output operations performed on them:

Error indicator

This indicator is set when an error has occurred in an operation related to the stream. This indicator can be checked with the [ferror](http://www.cplusplus.com/ferror) function, and can be reset by calling either to [clearerr](http://www.cplusplus.com/clearerr), [freopen](http://www.cplusplus.com/freopen) or [rewind](http://www.cplusplus.com/rewind).

End-Of-File indicator

When set, indicates that the last reading or writing operation performed with the stream reached the *End of File*. It can be checked with the [feof](http://www.cplusplus.com/feof) function, and can be reset by calling either to [clearerr](http://www.cplusplus.com/clearerr) or [freopen](http://www.cplusplus.com/freopen) or by calling to any repositioning function ([rewind](http://www.cplusplus.com/rewind), [fseek](http://www.cplusplus.com/fseek) and [fsetpos](http://www.cplusplus.com/fsetpos)).

Position indicator

It is an internal pointer of each stream which points to the next character to be read or written in the next I/O operation. Its value can be obtained by the [ftell](http://www.cplusplus.com/ftell) and [fgetpos](http://www.cplusplus.com/fgetpos) functions, and can be changed using the repositioning functions [rewind](http://www.cplusplus.com/rewind), [fseek](http://www.cplusplus.com/fseek) and [fsetpos](http://www.cplusplus.com/fsetpos).

### Functions

**Operations on files**:

[**remove**](http://www.cplusplus.com/reference/cstdio/remove/)

Remove file (function )

[**rename**](http://www.cplusplus.com/reference/cstdio/rename/)

Rename file (function )

[**tmpfile**](http://www.cplusplus.com/reference/cstdio/tmpfile/)

Open a temporary file (function )

[**tmpnam**](http://www.cplusplus.com/reference/cstdio/tmpnam/)

Generate temporary filename (function )

**File access**:

[**fclose**](http://www.cplusplus.com/reference/cstdio/fclose/)

Close file (function )

[**fflush**](http://www.cplusplus.com/reference/cstdio/fflush/)

Flush stream (function )

[**fopen**](http://www.cplusplus.com/reference/cstdio/fopen/)

Open file (function )

[**freopen**](http://www.cplusplus.com/reference/cstdio/freopen/)

Reopen stream with different file or mode (function )

[**setbuf**](http://www.cplusplus.com/reference/cstdio/setbuf/)

Set stream buffer (function )

[**setvbuf**](http://www.cplusplus.com/reference/cstdio/setvbuf/)

Change stream buffering (function )

**Formatted input/output**:

[**fprintf**](http://www.cplusplus.com/reference/cstdio/fprintf/)

Write formatted data to stream (function )

[**fscanf**](http://www.cplusplus.com/reference/cstdio/fscanf/)

Read formatted data from stream (function )

[**printf**](http://www.cplusplus.com/reference/cstdio/printf/)

Print formatted data to stdout (function )

[**scanf**](http://www.cplusplus.com/reference/cstdio/scanf/)

Read formatted data from stdin (function )

[**snprintf**](http://www.cplusplus.com/reference/cstdio/snprintf/)

Write formatted output to sized buffer (function )

[**sprintf**](http://www.cplusplus.com/reference/cstdio/sprintf/)

Write formatted data to string (function )

[**sscanf**](http://www.cplusplus.com/reference/cstdio/sscanf/)

Read formatted data from string (function )

[**vfprintf**](http://www.cplusplus.com/reference/cstdio/vfprintf/)

Write formatted data from variable argument list to stream (function )

[**vfscanf**](http://www.cplusplus.com/reference/cstdio/vfscanf/)

Read formatted data from stream into variable argument list (function )

[**vprintf**](http://www.cplusplus.com/reference/cstdio/vprintf/)

Print formatted data from variable argument list to stdout (function )

[**vscanf**](http://www.cplusplus.com/reference/cstdio/vscanf/)

Read formatted data into variable argument list (function )

[**vsnprintf**](http://www.cplusplus.com/reference/cstdio/vsnprintf/)

Write formatted data from variable argument list to sized buffer (function )

[**vsprintf**](http://www.cplusplus.com/reference/cstdio/vsprintf/)

Write formatted data from variable argument list to string (function )

[**vsscanf**](http://www.cplusplus.com/reference/cstdio/vsscanf/)

Read formatted data from string into variable argument list (function )

**Character input/output**:

[**fgetc**](http://www.cplusplus.com/reference/cstdio/fgetc/)

Get character from stream (function )

[**fgets**](http://www.cplusplus.com/reference/cstdio/fgets/)

Get string from stream (function )

[**fputc**](http://www.cplusplus.com/reference/cstdio/fputc/)

Write character to stream (function )

[**fputs**](http://www.cplusplus.com/reference/cstdio/fputs/)

Write string to stream (function )

[**getc**](http://www.cplusplus.com/reference/cstdio/getc/)

Get character from stream (function )

[**getchar**](http://www.cplusplus.com/reference/cstdio/getchar/)

Get character from stdin (function )

[**gets**](http://www.cplusplus.com/reference/cstdio/gets/)

Get string from stdin (function )

[**putc**](http://www.cplusplus.com/reference/cstdio/putc/)

Write character to stream (function )

[**putchar**](http://www.cplusplus.com/reference/cstdio/putchar/)

Write character to stdout (function )

[**puts**](http://www.cplusplus.com/reference/cstdio/puts/)

Write string to stdout (function )

[**ungetc**](http://www.cplusplus.com/reference/cstdio/ungetc/)

Unget character from stream (function )

**Direct input/output**:

[**fread**](http://www.cplusplus.com/reference/cstdio/fread/)

Read block of data from stream (function )

[**fwrite**](http://www.cplusplus.com/reference/cstdio/fwrite/)

Write block of data to stream (function )

**File positioning**:

[**fgetpos**](http://www.cplusplus.com/reference/cstdio/fgetpos/)

Get current position in stream (function )

[**fseek**](http://www.cplusplus.com/reference/cstdio/fseek/)

Reposition stream position indicator (function )

[**fsetpos**](http://www.cplusplus.com/reference/cstdio/fsetpos/)

Set position indicator of stream (function )

[**ftell**](http://www.cplusplus.com/reference/cstdio/ftell/)

Get current position in stream (function )

[**rewind**](http://www.cplusplus.com/reference/cstdio/rewind/)

Set position of stream to the beginning (function )

**Error-handling**:

[**clearerr**](http://www.cplusplus.com/reference/cstdio/clearerr/)

Clear error indicators (function )

[**feof**](http://www.cplusplus.com/reference/cstdio/feof/)

Check end-of-file indicator (function )

[**ferror**](http://www.cplusplus.com/reference/cstdio/ferror/)

Check error indicator (function )

[**perror**](http://www.cplusplus.com/reference/cstdio/perror/)

Print error message (function )

### Macros

[**BUFSIZ**](http://www.cplusplus.com/reference/cstdio/BUFSIZ/)

Buffer size (constant )

[**EOF**](http://www.cplusplus.com/reference/cstdio/EOF/)

End-of-File (constant )

[**FILENAME\_MAX**](http://www.cplusplus.com/reference/cstdio/FILENAME_MAX/)

Maximum length of file names (constant )

[**FOPEN\_MAX**](http://www.cplusplus.com/reference/cstdio/FOPEN_MAX/)

Potential limit of simultaneous open streams (constant )

[**L\_tmpnam**](http://www.cplusplus.com/reference/cstdio/L_tmpnam/)

Minimum length for temporary file name (constant )

[**NULL**](http://www.cplusplus.com/reference/cstdio/NULL/)

Null pointer (macro )

[**TMP\_MAX**](http://www.cplusplus.com/reference/cstdio/TMP_MAX/)

Number of temporary files (constant )

Additionally: [\_IOFBF](http://www.cplusplus.com/setvbuf), [\_IOLBF](http://www.cplusplus.com/setvbuf), [\_IONBF](http://www.cplusplus.com/setvbuf) (used with [setvbuf](http://www.cplusplus.com/setvbuf))  
and [SEEK\_CUR](http://www.cplusplus.com/fseek), [SEEK\_END](http://www.cplusplus.com/fseek) and [SEEK\_SET](http://www.cplusplus.com/fseek) (used with [fseek](http://www.cplusplus.com/fseek)).

### Types

[**FILE**](http://www.cplusplus.com/reference/cstdio/FILE/)

Object containing information to control a stream (type )

[**fpos\_t**](http://www.cplusplus.com/reference/cstdio/fpos_t/)

Object containing information to specify a position within a file (type )

[**size\_t**](http://www.cplusplus.com/reference/cstdio/size_t/)

Unsigned integral type (type )

header

# <cstdlib> (stdlib.h)

**C Standard General Utilities Library**

This header defines several general purpose functions, including dynamic memory management, random number generation, communication with the environment, integer arithmetics, searching, sorting and converting.

### Functions

#### String conversion

[**atof**](http://www.cplusplus.com/reference/cstdlib/atof/)

Convert string to double (function )

[**atoi**](http://www.cplusplus.com/reference/cstdlib/atoi/)

Convert string to integer (function )

[**atol**](http://www.cplusplus.com/reference/cstdlib/atol/)

Convert string to long integer (function )

[**atoll**](http://www.cplusplus.com/reference/cstdlib/atoll/)

Convert string to long long integer (function )

[**strtod**](http://www.cplusplus.com/reference/cstdlib/strtod/)

Convert string to double (function )

[**strtof**](http://www.cplusplus.com/reference/cstdlib/strtof/)

Convert string to float (function )

[**strtol**](http://www.cplusplus.com/reference/cstdlib/strtol/)

Convert string to long integer (function )

[**strtold**](http://www.cplusplus.com/reference/cstdlib/strtold/)

Convert string to long double (function )

[**strtoll**](http://www.cplusplus.com/reference/cstdlib/strtoll/)

Convert string to long long integer (function )

[**strtoul**](http://www.cplusplus.com/reference/cstdlib/strtoul/)

Convert string to unsigned long integer (function )

[**strtoull**](http://www.cplusplus.com/reference/cstdlib/strtoull/)

Convert string to unsigned long long integer (function )

#### Pseudo-random sequence generation

[**rand**](http://www.cplusplus.com/reference/cstdlib/rand/)

Generate random number (function )

[**srand**](http://www.cplusplus.com/reference/cstdlib/srand/)

Initialize random number generator (function )

#### Dynamic memory management

[**calloc**](http://www.cplusplus.com/reference/cstdlib/calloc/)

Allocate and zero-initialize array (function )

[**free**](http://www.cplusplus.com/reference/cstdlib/free/)

Deallocate memory block (function )

[**malloc**](http://www.cplusplus.com/reference/cstdlib/malloc/)

Allocate memory block (function )

[**realloc**](http://www.cplusplus.com/reference/cstdlib/realloc/)

Reallocate memory block (function )

#### Environment

[**abort**](http://www.cplusplus.com/reference/cstdlib/abort/)

Abort current process (function )

[**atexit**](http://www.cplusplus.com/reference/cstdlib/atexit/)

Set function to be executed on exit (function )

[**at\_quick\_exit**](http://www.cplusplus.com/reference/cstdlib/at_quick_exit/)

Set function to be executed on quick exit (function )

[**exit**](http://www.cplusplus.com/reference/cstdlib/exit/)

Terminate calling process (function )

[**getenv**](http://www.cplusplus.com/reference/cstdlib/getenv/)

Get environment string (function )

[**quick\_exit**](http://www.cplusplus.com/reference/cstdlib/quick_exit/)

Terminate calling process quick (function )

[**system**](http://www.cplusplus.com/reference/cstdlib/system/)

Execute system command (function )

[**\_Exit**](http://www.cplusplus.com/reference/cstdlib/_Exit/)

Terminate calling process (function )

#### Searching and sorting

[**bsearch**](http://www.cplusplus.com/reference/cstdlib/bsearch/)

Binary search in array (function )

[**qsort**](http://www.cplusplus.com/reference/cstdlib/qsort/)

Sort elements of array (function )

#### Integer arithmetics

[**abs**](http://www.cplusplus.com/reference/cstdlib/abs/)

Absolute value (function )

[**div**](http://www.cplusplus.com/reference/cstdlib/div/)

Integral division (function )

[**labs**](http://www.cplusplus.com/reference/cstdlib/labs/)

Absolute value (function )

[**ldiv**](http://www.cplusplus.com/reference/cstdlib/ldiv/)

Integral division (function )

[**llabs**](http://www.cplusplus.com/reference/cstdlib/llabs/)

Absolute value (function )

[**lldiv**](http://www.cplusplus.com/reference/cstdlib/lldiv/)

Integral division (function )

#### Multibyte characters

[**mblen**](http://www.cplusplus.com/reference/cstdlib/mblen/)

Get length of multibyte character (function )

[**mbtowc**](http://www.cplusplus.com/reference/cstdlib/mbtowc/)

Convert multibyte sequence to wide character (function )

[**wctomb**](http://www.cplusplus.com/reference/cstdlib/wctomb/)

Convert wide character to multibyte sequence (function )

#### Multibyte strings

[**mbstowcs**](http://www.cplusplus.com/reference/cstdlib/mbstowcs/)

Convert multibyte string to wide-character string (function )

[**wcstombs**](http://www.cplusplus.com/reference/cstdlib/wcstombs/)

Convert wide-character string to multibyte string (function )

### Macro constants

[**EXIT\_FAILURE**](http://www.cplusplus.com/reference/cstdlib/EXIT_FAILURE/)

Failure termination code (macro )

[**EXIT\_SUCCESS**](http://www.cplusplus.com/reference/cstdlib/EXIT_SUCCESS/)

Success termination code (macro )

[**MB\_CUR\_MAX**](http://www.cplusplus.com/reference/cstdlib/MB_CUR_MAX/)

Maximum size of multibyte characters (macro )

[**NULL**](http://www.cplusplus.com/reference/cstdlib/NULL/)

Null pointer (macro )

[**RAND\_MAX**](http://www.cplusplus.com/reference/cstdlib/RAND_MAX/)

Maximum value returned by rand (macro )

### Types

[**div\_t**](http://www.cplusplus.com/reference/cstdlib/div_t/)

Structure returned by div (type )

[**ldiv\_t**](http://www.cplusplus.com/reference/cstdlib/ldiv_t/)

Structure returned by ldiv (type )

[**lldiv\_t**](http://www.cplusplus.com/reference/cstdlib/lldiv_t/)

Structure returned by lldiv (type )

[**size\_t**](http://www.cplusplus.com/reference/cstddef/size_t/)

Unsigned integral type (type )

header

# <cstring> (string.h)

**C Strings**

This header file defines several functions to manipulate *C strings* and arrays.

### Functions

**Copying**:

[**memcpy**](http://www.cplusplus.com/reference/cstring/memcpy/)

Copy block of memory (function )

[**memmove**](http://www.cplusplus.com/reference/cstring/memmove/)

Move block of memory (function )

[**strcpy**](http://www.cplusplus.com/reference/cstring/strcpy/)

Copy string (function )

[**strncpy**](http://www.cplusplus.com/reference/cstring/strncpy/)

Copy characters from string (function )

**Concatenation**:

[**strcat**](http://www.cplusplus.com/reference/cstring/strcat/)

Concatenate strings (function )

[**strncat**](http://www.cplusplus.com/reference/cstring/strncat/)

Append characters from string (function )

**Comparison**:

[**memcmp**](http://www.cplusplus.com/reference/cstring/memcmp/)

Compare two blocks of memory (function )

[**strcmp**](http://www.cplusplus.com/reference/cstring/strcmp/)

Compare two strings (function )

[**strcoll**](http://www.cplusplus.com/reference/cstring/strcoll/)

Compare two strings using locale (function )

[**strncmp**](http://www.cplusplus.com/reference/cstring/strncmp/)

Compare characters of two strings (function )

[**strxfrm**](http://www.cplusplus.com/reference/cstring/strxfrm/)

Transform string using locale (function )

**Searching**:

[**memchr**](http://www.cplusplus.com/reference/cstring/memchr/)

Locate character in block of memory (function )

[**strchr**](http://www.cplusplus.com/reference/cstring/strchr/)

Locate first occurrence of character in string (function )

[**strcspn**](http://www.cplusplus.com/reference/cstring/strcspn/)

Get span until character in string (function )

[**strpbrk**](http://www.cplusplus.com/reference/cstring/strpbrk/)

Locate characters in string (function )

[**strrchr**](http://www.cplusplus.com/reference/cstring/strrchr/)

Locate last occurrence of character in string (function )

[**strspn**](http://www.cplusplus.com/reference/cstring/strspn/)

Get span of character set in string (function )

[**strstr**](http://www.cplusplus.com/reference/cstring/strstr/)

Locate substring (function )

[**strtok**](http://www.cplusplus.com/reference/cstring/strtok/)

Split string into tokens (function )

**Other**:

[**memset**](http://www.cplusplus.com/reference/cstring/memset/)

Fill block of memory (function )

[**strerror**](http://www.cplusplus.com/reference/cstring/strerror/)

Get pointer to error message string (function )

[**strlen**](http://www.cplusplus.com/reference/cstring/strlen/)

Get string length (function )

### Macros

[**NULL**](http://www.cplusplus.com/reference/cstring/NULL/)

Null pointer (macro )

### Types

[**size\_t**](http://www.cplusplus.com/reference/cstring/size_t/)

Unsigned integral type (type )

header

# <ctgmath> (tgmath.h)

**Type-generic math**

* [C](javascript:switch1.select(1))
* [C++](javascript:switch1.select(2))

This header defines macro functions that correspond to the functions in [<math.h>](http://www.cplusplus.com/cmath), but which can take other non-*floating point* types as arguments:  
  
Every function in [<math.h>](http://www.cplusplus.com/cmath) that takes at least one double as argument (except [modf](http://www.cplusplus.com/modf)) is defined in [<tgmath.h>](http://www.cplusplus.com/ctgmath) as a macro with the same semantics but taking *generic parameters* instead:  
  
Each of the arguments provided for these *generic parameters* that is of an integer type is casted to a double; Arguments of *floating-point types* are used without transformation (directly as float, double or long double).  
  
This header automatically includes [<math.h>](http://www.cplusplus.com/cmath) and <complex.h>: The type-generic function can also take *complex* values if the function exists in <complex.h> (prefixed with a "c" character).

header

# <ctime> (time.h)

**C Time Library**

This header file contains definitions of functions to get and manipulate date and time information.

### Functions

#### Time manipulation

[**clock**](http://www.cplusplus.com/reference/ctime/clock/)

Clock program (function )

[**difftime**](http://www.cplusplus.com/reference/ctime/difftime/)

Return difference between two times (function )

[**mktime**](http://www.cplusplus.com/reference/ctime/mktime/)

Convert tm structure to time\_t (function )

[**time**](http://www.cplusplus.com/reference/ctime/time/)

Get current time (function )

#### Conversion

[**asctime**](http://www.cplusplus.com/reference/ctime/asctime/)

Convert tm structure to string (function )

[**ctime**](http://www.cplusplus.com/reference/ctime/ctime/)

Convert time\_t value to string (function )

[**gmtime**](http://www.cplusplus.com/reference/ctime/gmtime/)

Convert time\_t to tm as UTC time (function )

[**localtime**](http://www.cplusplus.com/reference/ctime/localtime/)

Convert time\_t to tm as local time (function )

[**strftime**](http://www.cplusplus.com/reference/ctime/strftime/)

Format time as string (function )

### Macro constants

[**CLOCKS\_PER\_SEC**](http://www.cplusplus.com/reference/ctime/CLOCKS_PER_SEC/)

Clock ticks per second (macro )

[**NULL**](http://www.cplusplus.com/reference/ctime/NULL/)

Null pointer (macro )

### types

[**clock\_t**](http://www.cplusplus.com/reference/ctime/clock_t/)

Clock type (type )

[**size\_t**](http://www.cplusplus.com/reference/ctime/size_t/)

Unsigned integral type (type )

[**time\_t**](http://www.cplusplus.com/reference/ctime/time_t/)

Time type (type )

[**struct tm**](http://www.cplusplus.com/reference/ctime/tm/)

Time structure (type )

header

# <cuchar> (uchar.h)

**Unicode characters**

This header provides support for 16-bit and 32-bit characters, suitable to be encoded using UTF-16 and UTF-32.

### Types

In C, this header defines two macros: char16\_t and char32\_t, which map to unsigned integral types of the appropriate size (the same as [uint\_least16\_t](http://www.cplusplus.com/cstdint) and [uint\_least32\_t](http://www.cplusplus.com/cstdint), respectively).  
  
In C++, char16\_t and char32\_t are fundamental types (and thus this header does not define such macros in C++).

### Macros

In C++, the following macros are defined by this header:

|  |  |
| --- | --- |
| **Macro** | **description** |
| \_\_STD\_UTF\_16\_\_ | If defined, values of type char16\_t have UTF-16 encoding. Otherwise, the encoding of char16\_t is unspecified. (In C11, the macro expands to 1 when defined) |
| \_\_STD\_UTF\_32\_\_ | If defined, values of type char32\_t have UTF-32 encoding. Otherwise, the encoding of char32\_t is unspecified. (In C11, the macro expands to 1 when defined) |

### Functions

[**c16rtomb**](http://www.cplusplus.com/reference/cuchar/c16rtomb/)

Convert 16-bit character to multibyte sequence (function )

[**c32rtomb**](http://www.cplusplus.com/reference/cuchar/c32rtomb/)

Convert 32-bit character to multibyte sequence (function )

[**mbrtoc16**](http://www.cplusplus.com/reference/cuchar/mbrtoc16/)

Convert multibyte sequence to 16-bit character (function )

[**mbrtoc32**](http://www.cplusplus.com/reference/cuchar/mbrtoc32/)

Convert multibyte sequence to 32-bit character (function )

### Compatibility

This header, first introduced in C by a Technical Report extending C99, is fully supported by the latest standards of C and C++ (both published in 2011).

header

# <cwchar> (wchar.h)

**Wide characters**

This header file defines several functions to work with *C wide strings*.

### Functions

**Input/Output:** (mostly wide versions of [<cstdio>](http://www.cplusplus.com/cstdio) functions)

[**fgetwc**](http://www.cplusplus.com/reference/cwchar/fgetwc/)

Get wide character from stream (function )

[**fgetws**](http://www.cplusplus.com/reference/cwchar/fgetws/)

Get wide string from stream (function )

[**fputwc**](http://www.cplusplus.com/reference/cwchar/fputwc/)

Write wide character to stream (function )

[**fputws**](http://www.cplusplus.com/reference/cwchar/fputws/)

Write wide string to stream (function )

[**fwide**](http://www.cplusplus.com/reference/cwchar/fwide/)

Stream orientation (function )

[**fwprintf**](http://www.cplusplus.com/reference/cwchar/fwprintf/)

Write formatted data to stream (function )

[**fwscanf**](http://www.cplusplus.com/reference/cwchar/fwscanf/)

Read formatted data from stream (function )

[**getwc**](http://www.cplusplus.com/reference/cwchar/getwc/)

Get wide character from stream (function )

[**getwchar**](http://www.cplusplus.com/reference/cwchar/getwchar/)

Get wide character from stdin (function )

[**putwc**](http://www.cplusplus.com/reference/cwchar/putwc/)

Write wide character to stream (function )

[**putwchar**](http://www.cplusplus.com/reference/cwchar/putwchar/)

Write wide character to stdout (function )

[**swprintf**](http://www.cplusplus.com/reference/cwchar/swprintf/)

Write formatted data to wide string (function )

[**swscanf**](http://www.cplusplus.com/reference/cwchar/swscanf/)

Read formatted data from string (function )

[**ungetwc**](http://www.cplusplus.com/reference/cwchar/ungetwc/)

Unget wide character from stream (function )

[**vfwprintf**](http://www.cplusplus.com/reference/cwchar/vfwprintf/)

Write formatted data from variable argument list to stream (function )

[**vfwscanf**](http://www.cplusplus.com/reference/cwchar/vfwscanf/)

Read formatted data from stream into variable argument list (function )

[**vswprintf**](http://www.cplusplus.com/reference/cwchar/vswprintf/)

Write formatted data from variable argument list to sized buffer (function )

[**vswscanf**](http://www.cplusplus.com/reference/cwchar/vswscanf/)

Read formatted data from wide string into variable argument list (function )

[**vwprintf**](http://www.cplusplus.com/reference/cwchar/vwprintf/)

Print formatted data from variable argument list to stdout (function )

[**vwscanf**](http://www.cplusplus.com/reference/cwchar/vwscanf/)

Read formatted data into variable argument list (function )

[**wprintf**](http://www.cplusplus.com/reference/cwchar/wprintf/)

Print formatted data to stdout (function )

[**wscanf**](http://www.cplusplus.com/reference/cwchar/wscanf/)

Read formatted data from stdin (function )

**General utilities:** (wide versions of [<cstdlib>](http://www.cplusplus.com/cstdlib) functions)

[**wcstod**](http://www.cplusplus.com/reference/cwchar/wcstod/)

Convert wide string to double (function )

[**wcstof**](http://www.cplusplus.com/reference/cwchar/wcstof/)

Convert wide string to float (function )

[**wcstol**](http://www.cplusplus.com/reference/cwchar/wcstol/)

Convert wide string to long integer (function )

[**wcstold**](http://www.cplusplus.com/reference/cwchar/wcstold/)

Convert wide string to long double (function )

[**wcstoll**](http://www.cplusplus.com/reference/cwchar/wcstoll/)

Convert wide string to long long integer (function )

[**wcstoul**](http://www.cplusplus.com/reference/cwchar/wcstoul/)

Convert wide string to unsigned long integer (function )

[**wcstoull**](http://www.cplusplus.com/reference/cwchar/wcstoull/)

Convert wide string to unsigned long long integer (function )

**Character/string conversion:** (mostly extended versions of [<cstdlib>](http://www.cplusplus.com/cstdlib) functions)

[**btowc**](http://www.cplusplus.com/reference/cwchar/btowc/)

Convert single byte character to wide character (function )

[**mbrlen**](http://www.cplusplus.com/reference/cwchar/mbrlen/)

Get length of multibyte character (function )

[**mbrtowc**](http://www.cplusplus.com/reference/cwchar/mbrtowc/)

Convert multibyte sequence to wide character (function )

[**mbsinit**](http://www.cplusplus.com/reference/cwchar/mbsinit/)

Check if initial conversion state (function )

[**mbsrtowcs**](http://www.cplusplus.com/reference/cwchar/mbsrtowcs/)

Convert multibyte string to wide-character string (function )

[**wcrtomb**](http://www.cplusplus.com/reference/cwchar/wcrtomb/)

Convert wide character to multibyte sequence (function )

[**wctob**](http://www.cplusplus.com/reference/cwchar/wctob/)

Convert wide character to single byte (function )

[**wcsrtombs**](http://www.cplusplus.com/reference/cwchar/wcsrtombs/)

Convert wide-character string to multibyte string (function )

**Strings:** (wide versions of [<cstring>](http://www.cplusplus.com/cstring) functions)

[**wcscat**](http://www.cplusplus.com/reference/cwchar/wcscat/)

Concatenate wide strings (function )

[**wcschr**](http://www.cplusplus.com/reference/cwchar/wcschr/)

Locate first occurrence of character in wide string (function )

[**wcscmp**](http://www.cplusplus.com/reference/cwchar/wcscmp/)

Compare two strings (function )

[**wcscoll**](http://www.cplusplus.com/reference/cwchar/wcscoll/)

Compare two wide strings using locale (function )

[**wcscpy**](http://www.cplusplus.com/reference/cwchar/wcscpy/)

Copy wide string (function )

[**wcscspn**](http://www.cplusplus.com/reference/cwchar/wcscspn/)

Get span until character in wide string (function )

[**wcslen**](http://www.cplusplus.com/reference/cwchar/wcslen/)

Get wide string length (function )

[**wcsncat**](http://www.cplusplus.com/reference/cwchar/wcsncat/)

Append characters from wide string (function )

[**wcsncmp**](http://www.cplusplus.com/reference/cwchar/wcsncmp/)

Compare characters of two wide strings (function )

[**wcsncpy**](http://www.cplusplus.com/reference/cwchar/wcsncpy/)

Copy characters from wide string (function )

[**wcspbrk**](http://www.cplusplus.com/reference/cwchar/wcspbrk/)

Locate characters in wide string (function )

[**wcsrchr**](http://www.cplusplus.com/reference/cwchar/wcsrchr/)

Locate last occurrence of character in wide string (function )

[**wcsspn**](http://www.cplusplus.com/reference/cwchar/wcsspn/)

Get span of character set in wide string (function )

[**wcsstr**](http://www.cplusplus.com/reference/cwchar/wcsstr/)

Locate substring of wide string (function )

[**wcstok**](http://www.cplusplus.com/reference/cwchar/wcstok/)

Split wide string into tokens (function )

[**wcsxfrm**](http://www.cplusplus.com/reference/cwchar/wcsxfrm/)

Transform wide string using locale (function )

[**wmemchr**](http://www.cplusplus.com/reference/cwchar/wmemchr/)

Locate character in block of wide characters (function )

[**wmemcmp**](http://www.cplusplus.com/reference/cwchar/wmemcmp/)

Compare two blocks of wide characters (function )

[**wmemcpy**](http://www.cplusplus.com/reference/cwchar/wmemcpy/)

Copy block of wide characters (function )

[**wmemmove**](http://www.cplusplus.com/reference/cwchar/wmemmove/)

Move block of wide characters (function )

[**wmemset**](http://www.cplusplus.com/reference/cwchar/wmemset/)

Fill array of wide characters (function )

**Time:** (a wide version of a [<ctime>](http://www.cplusplus.com/%3Cctime%3E) function)

[**wcsftime**](http://www.cplusplus.com/reference/cwchar/wcsftime/)

Format time as wide string (function )

### Types

[**mbstate\_t**](http://www.cplusplus.com/reference/cwchar/mbstate_t/)

Multibyte conversion state (type )

[**size\_t**](http://www.cplusplus.com/reference/cwchar/size_t/)

Unsigned integral type (type )

[**struct tm**](http://www.cplusplus.com/reference/cwchar/tm/)

Time structure (type )

[**wchar\_t**](http://www.cplusplus.com/reference/cwchar/wchar_t/)

Wide character (type )

[**wint\_t**](http://www.cplusplus.com/reference/cwchar/wint_t/)

Wide int type (type )

### Macro constants

[**NULL**](http://www.cplusplus.com/reference/cwchar/NULL/)

Null pointer (macro )

[**WCHAR\_MAX**](http://www.cplusplus.com/reference/cwchar/WCHAR_MAX/)

Maximum value of wchar\_t (constant )

[**WCHAR\_MIN**](http://www.cplusplus.com/reference/cwchar/WCHAR_MIN/)

Minimum value of wchar\_t (constant )

[**WEOF**](http://www.cplusplus.com/reference/cwchar/WEOF/)

Wide end-of-file (constant )

header

# <cwctype> (wctype.h)

**Wide character type**

This header declares a set of functions to classify and transform individual wide characters.  
  
For more info on how the standard ASCII character set is classified using the "C" locale, see [<cctype>](http://www.cplusplus.com/cctype).

### Functions

#### Character classification functions

They check whether the character passed as parameter belongs to a certain category:

[**iswalnum**](http://www.cplusplus.com/reference/cwctype/iswalnum/)

Check if wide character is alphanumeric (function )

[**iswalpha**](http://www.cplusplus.com/reference/cwctype/iswalpha/)

Check if wide character is alphabetic (function )

[**iswblank**](http://www.cplusplus.com/reference/cwctype/iswblank/)

Check if wide character is blank (function )

[**iswcntrl**](http://www.cplusplus.com/reference/cwctype/iswcntrl/)

Check if wide character is a control character (function )

[**iswdigit**](http://www.cplusplus.com/reference/cwctype/iswdigit/)

Check if wide character is decimal digit (function )

[**iswgraph**](http://www.cplusplus.com/reference/cwctype/iswgraph/)

Check if wide character has graphical representation (function )

[**iswlower**](http://www.cplusplus.com/reference/cwctype/iswlower/)

Check if wide character is lowercase letter (function )

[**iswprint**](http://www.cplusplus.com/reference/cwctype/iswprint/)

Check if wide character is printable (function )

[**iswpunct**](http://www.cplusplus.com/reference/cwctype/iswpunct/)

Check if wide character is punctuation character (function )

[**iswspace**](http://www.cplusplus.com/reference/cwctype/iswspace/)

Check if wide character is a white-space (function )

[**iswupper**](http://www.cplusplus.com/reference/cwctype/iswupper/)

Check if wide character is uppercase letter (function )

[**iswxdigit**](http://www.cplusplus.com/reference/cwctype/iswxdigit/)

Check if wide character is hexadecimal digit (function )

#### Character conversion functions

Two functions that convert between letter cases:

[**towlower**](http://www.cplusplus.com/reference/cwctype/towlower/)

Convert uppercase wide character to lowercase (function )

[**towupper**](http://www.cplusplus.com/reference/cwctype/towupper/)

Convert lowercase wide character to uppercase (function )

#### Extensible classification/conversion functions

[**iswctype**](http://www.cplusplus.com/reference/cwctype/iswctype/)

Check if wide character has property (function )

[**towctrans**](http://www.cplusplus.com/reference/cwctype/towctrans/)

Convert using transformation (function )

[**wctrans**](http://www.cplusplus.com/reference/cwctype/wctrans/)

Return character transformation (function )

[**wctype**](http://www.cplusplus.com/reference/cwctype/wctype/)

Return character property (function )

### Types

[**wctrans\_t**](http://www.cplusplus.com/reference/cwctype/wctrans_t/)

Wide character transformation (type )

[**wctype\_t**](http://www.cplusplus.com/reference/cwctype/wctype_t/)

Wide character type (type )

[**wint\_t**](http://www.cplusplus.com/reference/cwctype/wint_t/)

Wide character integral type (type )

### Constants

[**WEOF**](http://www.cplusplus.com/reference/cwctype/WEOF/)

Wide End-of-File (constant )