

<stdio> (stdio.h)

C library to perform Input/Output operations

Input and Output operations can also be performed in C++ using the **C** **S**tandar **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](#) objects. A pointer to a [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` , `stdout` and `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](#) 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 `<wchar>` are *wide-oriented*). See [wchar](#) 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](#) function, and can be reset by calling either to [clearerr](#) , [freopen](#) or [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](#) function, and can be reset by calling either to [clearerr](#) or [freopen](#) or by calling to any repositioning function ([rewind](#) , [fseek](#) and [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](#) and [fgetpos](#) functions, and can be changed using the repositioning functions [rewind](#) , [fseek](#) and [fsetpos](#) .

Functions

Operations on files :

remove	Remove file (function)
rename	Rename file (function)
tmpfile	Open a temporary file (function)
tmpnam	Generate temporary filename (function)

File access :

fclose	Close file (function)
fflush	Flush stream (function)
fopen	Open file (function)

freopen	Reopen stream with different file or mode (function)
setbuf	Set stream buffer (function)
setvbuf	Change stream buffering (function)

Formatted input/output :

fprintf	Write formatted data to stream (function)
fscanf	Read formatted data from stream (function)
printf	Print formatted data to stdout (function)
scanf	Read formatted data from stdin (function)
snprintf	Write formatted output to sized buffer (function)
sprintf	Write formatted data to string (function)
sscanf	Read formatted data from string (function)
vfprintf	Write formatted data from variable argument list to stream (function)
vfscanf	Read formatted data from stream into variable argument list (function)
vprintf	Print formatted data from variable argument list to stdout (function)
vscanf	Read formatted data into variable argument list (function)
vsprintf	Write formatted data from variable argument list to sized buffer (function)
vsprintf	Write formatted data from variable argument list to string (function)
vsscanf	Read formatted data from string into variable argument list (function)

Character input/output :

fgetc	Get character from stream (function)
fgets	Get string from stream (function)
fputc	Write character to stream (function)
fputs	Write string to stream (function)
getc	Get character from stream (function)
getchar	Get character from stdin (function)
gets	Get string from stdin (function)
putc	Write character to stream (function)
putchar	Write character to stdout (function)
puts	Write string to stdout (function)
ungetc	Unget character from stream (function)

Direct input/output :

fread	Read block of data from stream (function)
fwrite	Write block of data to stream (function)

File positioning :

fgetpos	Get current position in stream (function)
fseek	Reposition stream position indicator (function)
fsetpos	Set position indicator of stream (function)
ftell	Get current position in stream (function)
rewind	Set position of stream to the beginning (function)

Error-handling :

clearerr	Clear error indicators (function)
feof	Check end-of-file indicator (function)
ferror	Check error indicator (function)
perror	Print error message (function)

Macros

BUFSIZ	Buffer size (constant)
EOF	End-of-File (constant)
FILENAME_MAX	Maximum length of file names (constant)
FOPEN_MAX	Potential limit of simultaneous open streams (constant)

L_tmpnam	Minimum length for temporary file name (constant)
NULL	Null pointer (macro)
TMP_MAX	Number of temporary files (constant)

Additionally: `_IOFBF` , `_IOLBF` , `_IONBF` (used with `setvbuf`)
and `SEEK_CUR` , `SEEK_END` and `SEEK_SET` (used with `fseek`).

Types

FILE	Object containing information to control a stream (type)
fpos_t	Object containing information to specify a position within a file (type)
size_t	Unsigned integral type (type)