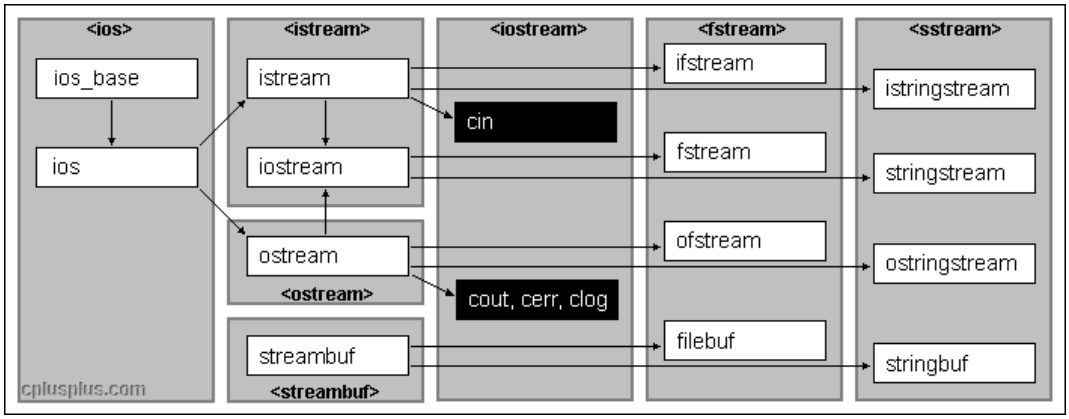
library

Input/Output

Input/Output library



The iostream library is an object-oriented library that provides input and output functionality using streams.

A stream is an abstraction that represents a device on which input and ouput operations are performed. A stream can basically be represented as a source or destination of characters of indefinite length.

Streams are generally associated to a physical source or destination of characters, like a disk file, the keyboard, or the console, so the characters gotten or written to/from our abstraction called stream are physically input/output to the physical device. For example, file streams are C++ objects to manipulate and interact with files; Once a file stream is used to open a file, any input or output operation performed on that stream is physically reflected in the file.

To operate with streams, C++ provides the standard `iostream` library, which contains the following elements:

Basic class templates

The base of the iostream library is the hierarchy of class templates. The class templates provide most of the functionality of the library in a type-independent fashion.

This is a set of class templates, each one having two template parameters: the *char type* (charT) parameter, that determines the type of elements that are going to be manipulated and the *traits* parameter, that provides additional characteristics specific for a particular type of elements.

The class templates in this class hierarchy have the same name as their char-type instantiations but with the prefix `basic_`. For example, the class template which `istream` is instantiated from is called `basic_istream`, the one from which `fstream` is is called `basic_fstream`, and so on... The only exception is `ios_base`, which is by itself type-independent, and therefore is not based on a template, but is a regular class.

Class template instantiations

The library incorporates two standard sets of instantiations of the entire iostream class template hierarchy: one is narrow-oriented, to manipulate elements of type `char` and another one, wide-oriented, to manipulate elements of type `wchar_t`.

The narrow-oriented (`char` type) instantiation is probably the better known part of the iostream library. Classes like `ios`, `istream` and `ofstream` are narrow-oriented. The diagram on top of this page shows the names and relationships of narrow-oriented classes.

The classes of the wide-oriented (`wchar_t`) instatiation follow the same naming conventions as the narrow-oriented instantiation but with the name of each class and object prefixed with a `w` character, forming `wios`, `wistream` and `wofstream`, as an example.

Standard objects

As part of the iostream library, the header file `<iostream>` declares certain objects that are used to perform input and output operations on the standard input and output.

They are divided in two sets: narrow-oriented objects, which are the popular `cin`, `cout`, `cerr` and `clog` and their wide-oriented counterparts, declared as `wcin`, `wcout`, `wcerr` and `wclog`.

Types

The iostream classes barely use fundamental types on their member's prototypes. They generally use defined types that depend on the traits used in their instantiation. For the default `char` and `wchar_t` instantiations, types `streampos`, `streamoff` and `streamsize` are used to represent positions, offsets and sizes, respectively.

Manipulators

Manipulators are global functions designed to be used together with insertion (<<) and extraction (>>) operators performed on `iostream` stream objects. They generally modify properties and formatting settings of the streams. `endl`, `hex` and `scientific` are some examples of manipulators.

Organization

The library and its hierarchy of classes is split in different files:

- `<ios>`, `<istream>`, `<ostream>`, `<streambuf>` and `<iosfwd>` aren't usually included directly in most C++ programs. They describe the base classes of the hierarchy and are automatically included by other header files of the library that contain derived classes.
- `<iostream>` declares the objects used to communicate through the standard input and output (including `cin` and `cout`).
- `<fstream>` defines the file stream classes (like the template `basic_ifstream` or the class `ofstream`) as well as the internal buffer objects used with these (`basic_filebuf`). These classes are used to manipulate files using streams.
- `<sstream>`: The classes defined in this file are used to manipulate string objects as if they were streams.
- `<iomanip>` declares some standard manipulators with parameters to be used with extraction and insertion operators to modify internal flags and formatting options.

Compatibility notes

The names, prototypes and examples included in this reference for the iostream classes mostly describe and use the `char` instantiations of the class templates instead of the templates themselves, even though these classes are only one of their possible instantiations. We believe this provides a better readability and is arguably as easy to obtain the names and prototypes of the basic template from the `char` instantiation as the opposite.

Elements of the iostream library (char instantitation)

Classes:

ios_base	Base class for streams (class)
ios	Base class for streams (type-dependent components) (class)
istream	Input stream (class)
ostream	Output Stream (class)
iostream	Input/output stream (class)
ifstream	Input file stream class (class)
ofstream	Output file stream (class)
fstream	Input/output file stream class (class)
istringstream	Input string stream (class)
ostringstream	Output string stream (class)
stringstream	Input/output string stream (class)
streambuf	Base buffer class for streams (class)
filebuf	File stream buffer (class)
stringbuf	String stream buffer (class)

Objects:

cin	Standard input stream (object)
cout	Standard output stream (object)
cerr	Standard output stream for errors (object)
clog	Standard output stream for logging (object)

Types:

fpos	Stream position class template (class template)
streamoff	Stream offset type (type)
streampos	Stream position type (type)
streamsize	Stream size type (type)

Manipulators:

boolalpha	Alphanumerical bool values (function)
dec	Use decimal base (function)
endl	Insert newline and flush (function)
ends	Insert null character (function)
fixed	Use fixed floating-point notation (function)
flush	Flush stream buffer (function)
hex	Use hexadecimal base (function)
internal	Adjust field by inserting characters at an internal position (function)
left	Adjust output to the left (function)
noboolalpha	No alphanumerical bool values (function)

noshowbase	Do not show numerical base prefixes (function)
noshowpoint	Do not show decimal point (function)
noshowpos	Do not show positive signs (function)
noskipws	Do not skip whitespaces (function)
nounitbuf	Do not force flushes after insertions (function)
nouppercase	Do not generate upper case letters (function)
oct	Use octal base (function)
resetiosflags	Reset format flags (function)
right	Adjust output to the right (function)
scientific	Use scientific floating-point notation (function)
setbase	Set basefield flag (function)
setfill	Set fill character (function)
setiosflags	Set format flags (function)
setprecision	Set decimal precision (function)
setw	Set field width (function)
showbase	Show numerical base prefixes (function)
showpoint	Show decimal point (function)
showpos	Show positive signs (function)
skipws	Skip whitespaces (function)
unitbuf	Flush buffer after insertions (function)
uppercase	Generate upper-case letters (function)
ws	Extract whitespaces (function)