C++
Information
Tutorials
Reference
Articles
Forum

Reference
C library:
Containers:
Input/Output:
<fstream>
<iomanip>
<ios>
<iosfwd>
<iostream>
<istream>
<ostream>
<sstream>
<streambuf>
Multi-threading:
Other:

<sstream>
class templates:
basic_istringstream
basic_ostringstream
basic_stringbuf
basic_stringstream
classes:
istringstream
ostringstream
stringbuf
stringstream
wistringstream
wostringstream
wstringbuf
wstringstream

istringstream
istringstream::istringstream
public member functions:
istringstream::operator=
istringstream::rdbuf
istringstream::str
istringstream::swap
non-member overloads:
swap (istringstream)

public member function

std::istringstream::istringstream

<sstream>

C++98 C++11

```
default (1)        explicit istringstream (ios_base::openmode which = ios_base::in);
initialization (2) explicit istringstream (const string& str,
                                           ios_base::openmode which = ios_base::in);
```

Construct object

Constructs a `istringstream` object:

(1) empty constructor (default constructor)

Constructs an `istringstream` object with an empty sequence as content.
Internally, its `istream` base constructor is passed a pointer to a `stringbuf` object constructed with an argument based on *which*.

(2) initialization constructor

Constructs an `istringstream` object with a copy of *str* as content.
Internally, its `istream` base constructor is passed a pointer to a `stringbuf` object constructed with arguments based on *str* and *which*.

(3) copy constructor (deleted)

Deleted (no copy constructor).

(4) move constructor

Acquires the contents of *x*.
First, the function move-constructs both its base `istream` class from *x* and a `stringbuf` object from *x*'s internal `streambuf` object, and then associates them by calling member `set_rdbuf`.
x is left in an unspecified but valid state.
It is unspecified whether the sequence controlled by the internal `stringbuf` object is the one in *x* before the call, or a copy of it. In any case, both objects have internal *string buffers* that use independent sequences after the call.

The internal `stringbuf` object has at least the same duration as the `istringstream` object.

Parameters

`str`
A `string` object, whose content is copied.

`x`
A `istringstream` object, whose value is moved.

`which`
Open mode: Access given by the internal `stringbuf` object to its internal sequence of characters. It is an object of member type `openmode` for which any combination of the following member values is significant:

C++98 C++11

member constant	stands for	access
`ios_base::in`*	**in**put	The sequence supports input operations.
`ios_base::out`	**out**put	The sequence supports output operations.

Other values of type `ios_base::openmode` may also be specified, although whether they have an effect on `istringstream` objects depends on the library implementation.

* `ios_base::in` is always set for `istringstream` objects (even if explicitly not set in argument *which*).
Note that even though `istringstream` is an input stream, its internal `stringbuf` object may be set to also support output operations. This influences certain operations, such as `putback`, that in `istringstream` may alter the contents of the sequence.

Example

```
1  // istringstream constructors.
2  #include <iostream>     // std::cout
3  #include <sstream>      // std::istringstream
4  #include <string>       // std::string
5
6  int main () {
7
8    std::string stringvalues = "125 320 512 750 333";
9    std::istringstream iss (stringvalues);
10
11   for (int n=0; n<5; n++)
12   {
13     int val;
14     iss >> val;
15     std::cout << val*2 << '\n';
```

```
  }

  return 0;
}
```

Output:

```
250
640
1024
1500
666
```

Data races

The *move constructor (4)* modifies *x*.

Exception safety

Strong guarantee: if an exception is thrown, there are no side effects.

See also

istringstream::str	Get/set content (public member function)