



Search:

[Reference](#)
[<sstream>](#)
[stringstream](#)
[swap](#)

[register](#)
[log in](#)

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\_istream

basic\_ostringstream

basic\_stringbuf

basic\_stringstream

classes:

istream

ostream

stringbuf

stringstream

wstringstream

wstringstream

wstringbuf

wstringstream

stringstream

stringstream::stringstream

public member functions:

stringstream::operator=

stringstream::rdbuf

stringstream::str

stringstream::swap

non-member overloads:

swap (stringstream)

PatchIT Updating Libraries

PatchIT offers fully automated updating libraries for coding

public member function

std::stringstream::swap

<sstream>

void swap (stringstream& x);

Swap internals

Exchanges all internal data between x and \*this.

Internally, the function calls `istream::swap` and then calls the member `swap` of the associated `stringbuf` object.

Parameters

x

Another `stringstream` object.

Return Value

none

Example

```

1 // swapping stringstream objects
2 #include <string>      // std::string
3 #include <iostream>    // std::cout
4 #include <sstream>     // std::stringstream
5
6 int main () {
7
8     std::stringstream foo;
9     std::stringstream bar;
10
11     foo << 100;
12     bar << 200;
13
14     foo.swap(bar);
15
16     int val;
17
18     foo >> val; std::cout << "foo: " << val << '\n';
19     bar >> val; std::cout << "bar: " << val << '\n';
20
21     return 0;
22 }

```

Output:

foo: 200

bar: 100

Data races

Modifies both stream objects (\*this and x).

Exception safety

**No-throw guarantee:** this member function never throws exceptions.

See also

stringstream::operator=

Move assignment (public member function )

stream::swap

Home page | Privacy policy

© cplusplus.com, 2000-2015 - All rights reserved - v3.1

Spotted an error? contact us