



Search:

Go

Not logged in

Reference

<sstream>

stringbuf

stringbuf

register

log in

C++

Information
Tutorials
Reference
Articles
Forum

Reference

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
wstringstream

stringbuf

stringbuf::stringbuf
public members:
stringbuf::str
virtual members:
stringbuf::overflow
stringbuf::pbackfail
stringbuf::seekoff
stringbuf::seekpos
stringbuf::setbuf
stringbuf::underflow

Answers to C++ Questions
Free Answers to Your Programming
Language Questions. Register Now!

Networking 4 Game Devs

64 Network DO's and DON'Ts for Game Developers



public member function

std::stringbuf::stringbuf

<sstream>

C++98

C++11



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

Construct a string stream buffer object

Constructs a `stringbuf` object:

(1) empty constructor (default constructor)

Constructs a `stringbuf` object with an empty sequence as content, and argument *which* as open mode.

(2) initialization constructor

Constructs a `stringbuf` object with a copy of *str* as content, and argument *which* as open mode.

(3) copy constructor (deleted)

Deleted (no copy constructor).

(4) move constructor

Acquires the contents of *x*.*x* is left in an unspecified but valid state.It is unspecified whether the internal sequence is the one in *x* before the call, or a copy of it. In any case, both objects use independent sequences after the call.

Parameters

str

A `string` object, whose content is copied.

x

A `stringbuf` object, whose value is moved.

which

Open mode: Access given to the internal sequence of characters through the internal pointers that define the stream buffer's *input sequence* and *output sequence*. It is an object of type `ios_base::openmode` for which any combination of the following constant values is significant:

C++98

C++11



value	stands for	access
<code>ios_base::in</code>	input	The sequence can be read: Members <code>eback</code> , <code>gptr</code> and <code>egptr</code> are maintained with values pointing to elements of the internal character sequence.
<code>ios_base::out</code>	output	The sequence can be written: Members <code>pbase</code> , <code>pptr</code> and <code>pptr</code> are maintained with values pointing to elements of the internal character sequence.

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

Example

```
1 // stringbuf example
2 #include <iostream>      // std::cout, std::ostream, std::hex
3 #include <sstream>       // std::stringbuf
4 #include <string>        // std::string
5
6 int main ()
7 {
8     std::stringbuf buffer;    // empty stringbuf
9
10    std::ostream os (&buffer); // associate stream buffer to stream
11
12    // mixing output to buffer with inserting to associated stream:
13    buffer.sputn ("255 in hexadecimal: ",20);
14    os << std::hex << 255;
15
16    std::cout << buffer.str();
17
18    return 0;
19 }
```

Output:

255 in hexadecimal: ff

• Data races

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

• Exception safety

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

🔗 See also

stringbuf::str

Get/set the string content ([public member function](#))

[Home page](#) | [Privacy policy](#)
© cplusplus.com, 2000-2015 - All rights reserved - v3.1
[Spotted an error? contact us](#)