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
`ios_base::in`	**in**put	The sequence can be read: Members `eback`, `gptr` and `egptr` are maintained with values pointing to elements of the internal character sequence.
`ios_base::out`	**out**put	The sequence can be written: Members `pbase`, `pptr` and `epptr` 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

```
// stringbuf example
#include <iostream>      // std::cout, std::ostream, std::hex
#include <sstream>       // std::stringbuf
#include <string>        // std::string

int main ()
{
  std::stringbuf buffer;      // empty stringbuf

  std::ostream os (&buffer);  // associate stream buffer to stream

  // mixing output to buffer with inserting to associated stream:
  buffer.sputn ("255 in hexadecimal: ",20);
  os << std::hex << 255;

  std::cout << buffer.str();

  return 0;
}
```

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)