

Search:  Go

Not logged in  
register log in

Reference
<streambuf>
streambuf

C++

Information

Tutorials

Reference

Articles

Forum

Reference

C library:

Containers:

Input/Output:

<fstream>

<iomanip>

<ios>

<iosfwd>

<iostream>

<istream>

<ostream>

<sstream>

<streambuf>

Multi-threading:

Other:

<streambuf>

basic\_streambuf

streambuf

wstreambuf

streambuf

streambuf::streambuf

streambuf::~~streambuf

public members:

streambuf::getloc

streambuf::in\_avail

streambuf::pubimbue

streambuf::pubseekoff

streambuf::pubseekpos

streambuf::pubsetbuf

streambuf::pubsync

streambuf::sbumpc

streambuf::sgetc

streambuf::sgetn

streambuf::snextc

streambuf::sputbackc

streambuf::sputc

streambuf::sputn

streambuf::sungetc

protected members:

streambuf::eback

streambuf::egptr

streambuf::epptr

streambuf::gbump

streambuf::gptr

streambuf::operator=

streambuf::pbase

streambuf::pbump

streambuf::pptr

streambuf::setg

streambuf::setp

streambuf::swap

virtual protected members:

streambuf::imbue

streambuf::overflow

streambuf::pbackfail

streambuf::seekoff

streambuf::seekpos

streambuf::setbuf

streambuf::showmanyc

streambuf::sync

streambuf::uflow

streambuf::underflow

streambuf::xsgetn

streambuf::xspn

class

<streambuf> <iostream>

std::streambuf

typedef basic\_streambuf<char> streambuf;

Base buffer class for streams

streambuf

←

filebuf

←

stringbuf

This template is designed as base virtual class for all *stream buffer* classes that handle narrow characters (of type char).

It is an instantiation of `basic_streambuf` with the following template parameters:

template parameter	definition	comments
charT	char	Aliased as member <code>char_type</code>
traits	<code>char_traits&lt;char&gt;</code>	Aliased as member <code>traits_type</code>

A *stream buffer* is an object in charge of performing the reading and writing operations of the *stream* object it is associated with: the stream delegates all such operations to its associated *stream buffer* object, which is an intermediary between the *stream* and its *controlled input and output sequences*.

All *stream* objects, no matter whether buffered or unbuffered, have an associated *stream buffer*: Some *stream buffer* types may then be set to either use an intermediate *buffer* or not.

*Stream buffer* objects keep internally, at least:

- A `locale` object, used for locale-dependent operations.
- A set of internal pointers to keep an input buffer: `eback`, `gptr`, `egptr`.
- A set of internal pointers to keep an output buffer: `pbase`, `pptr`, `epptr`.

Internally, the `streambuf` class is an elaborated base class designed to provide a uniform public interface for all derived classes: These public functions call virtual protected members that derived classes may override to implement specific behavior. These overridden virtual functions have access to the internals of the `streambuf` class by means of a set of protected functions (see below).

Member types

member type	definition
<code>char_type</code>	<code>char</code>
<code>traits_type</code>	<code>char_traits&lt;char&gt;</code>
<code>int_type</code>	<code>int</code>
<code>pos_type</code>	<code>streampos</code>
<code>off_type</code>	<code>streamoff</code>

Public member functions

The common functionality for all stream buffers is provided through the following public member functions:

(constructor)	Construct object (public member function)
(destructor)	Destroy object (public member function)

**Locales:**

pubimbue	Imbue locale (public member function)
getloc	Get current locale (public member function)

**Buffer management and positioning:**

pubsetbuf	Set buffer array (public member function)
pubseekoff	Set internal position pointer to relative position (public member function)
pubseekpos	Set internal position pointer to absolute position (public member function)
pubsync	Synchronize stream buffer (public member function)

**Input functions (get):**

in_avail	Get number of characters available to read (public member function)
snextc	Advance to next position and get character (public member function)
sbumpc	Get current character and advance to next position (public member function)
sgetc	Get current character (public member function)

AddressSanitizer  
Detects buffer overflows and use-after-free in C++. [Open source](#)



<b>sgetn</b>	Get sequence of characters ( <a href="#">public member function</a> )
<b>sputbackc</b>	Put character back ( <a href="#">public member function</a> )
<b>sungetc</b>	Decrease current position ( <a href="#">public member function</a> )

**Output functions (put):**

<b>sputc</b>	Store character at current put position and increase put pointer ( <a href="#">public member function</a> )
<b>sputn</b>	Put sequence of characters ( <a href="#">public member function</a> )

***fx* Protected member functions**

The public functions do not perform their operations directly on the *controlled input and output sequences*, but mostly rely on two arrays accessible by a set of internal pointers:

	<b>beginning</b> <i>(beginning pointers)</i>	<b>current position</b> <i>(get/put pointer)</i>	<b>end</b> <i>(end pointers)</i>
Input sequence	<a href="#">eback</a>	<a href="#">gptr</a>	<a href="#">egptr</a>
Output sequence	<a href="#">pbase</a>	<a href="#">pptr</a>	<a href="#">epptr</a>

The following protected member functions provide access to these pointers:

**Input sequence (get):**

<b>eback</b>	Pointer to beginning of input sequence ( <a href="#">protected member function</a> )
<b>gptr</b>	Pointer to current position of input sequence ( <a href="#">protected member function</a> )
<b>egptr</b>	Pointer to end of input sequence ( <a href="#">protected member function</a> )
<b>gbump</b>	Increase get pointer ( <a href="#">protected member function</a> )
<b>setg</b>	Set input sequence pointers ( <a href="#">protected member function</a> )

**Output sequence (put):**

<b>pbase</b>	Pointer to beginning of output sequence ( <a href="#">protected member function</a> )
<b>pptr</b>	Pointer to current position of output sequence ( <a href="#">protected member function</a> )
<b>epptr</b>	Pointer to end of output sequence ( <a href="#">protected member function</a> )
<b>pbump</b>	Increase put pointer ( <a href="#">protected member function</a> )
<b>setp</b>	Set output sequence pointers ( <a href="#">protected member function</a> )

**Copying:**

<b>operator=</b> <small>C++11</small>	Streambuf assignment ( <a href="#">public member function</a> )
<b>swap</b> <small>C++11</small>	Swap stream buffers ( <a href="#">public member function</a> )

***fx* Virtual protected member functions**

Each streambuf-derived class shall define members that keep the validity of the pointers above with respect to their own type of *controlled sequence*; Modifying the values of the pointers, reallocating the sequences themselves and performing all necessary synchronizations with the *associated character sequence*.

With this design, the core functionality involving the process of reading and writing directly to the specific *associated character sequence* and to manage the *controlled sequences* is provided by means of virtual functions, which are overridden as necessary by derived classes:

**Locales:**

<b>imbue</b>	Imbue locale ( <a href="#">protected virtual member function</a> )
--------------	--

**Buffer management and positioning:**

<b>setbuf</b>	Set buffer ( <a href="#">protected virtual member function</a> )
<b>seekoff</b>	Set internal position pointer to relative position ( <a href="#">protected virtual member function</a> )
<b>seekpos</b>	Set internal position pointer to absolute position ( <a href="#">protected virtual member function</a> )
<b>sync</b>	Synchronize stream buffer ( <a href="#">protected virtual member function</a> )

**Input functions (get):**

<b>showmanyc</b>	Get number of characters available ( <a href="#">protected virtual member function</a> )
<b>xsgetn</b>	Get sequence of characters ( <a href="#">protected virtual member function</a> )
<b>underflow</b>	Get character on underflow ( <a href="#">protected virtual member function</a> )
<b>uflow</b>	Get character on underflow and advance position ( <a href="#">protected virtual member function</a> )
<b>pbackfail</b>	Put character back in the case of backup underflow ( <a href="#">protected virtual member function</a> )

**Output functions (put):**

<b>xspn</b>	Put sequence of characters ( <a href="#">protected virtual member function</a> )
<b>overflow</b>	Put character on overflow ( <a href="#">protected virtual member function</a> )

## PatchIT Updating Library

PatchIT offers fully automated updating libraries for coding



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