cplusplus.com

Search: Go

Not logged in

Reference <istream> istream readsome

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:

<istream>
class templates:
basic_iostream
basic_istream
classes:
iostream
istream
wiostream
wistream
manipulators:
ws

istream
istream::istream
istream::~istream
member classes:
istream::sentry
member functions:
istream::gcount
istream::get
istream::getline
istream::ignore
istream::operator>>
istream::peek
istream::putback
istream::read
istream::readsome
istream::seekg
istream::sync
istream::tellg
istream::unget
non-member overloads:
operator>> (istream)
protected members:
istream::operator= C++11
istream::swap C++11

public member function

std::istream::readsome

<istream> <iostream>

```
streamsize readsome (char* s, streamsize n);
```

Read data available in buffer

Extracts up to *n* characters from the stream and stores them in the array pointed by *s*, stopping as soon as the internal buffer kept by the *associated stream buffer object* (if any) runs out of characters, even if the *end-of-file* has not yet been reached.

The function is meant to be used to read data from certain types of asynchronous sources that may eventually wait for more characters, since it stops extracting characters as soon as the internal buffer is exhausted, avoiding potential delays.

Note that this function relies on internals of the particular *stream buffer* object associated to the stream whose behavior is mostly implementation-defined for standard classes.

Internally, the function accesses the input sequence by first constructing a `sentry` object (with *noskipws* set to `true`). Then (if `good`), it checks how many characters are currently available at the associated *stream buffer* object by calling its member function `in_avail` and extracts up to that many characters by calling `sbumpc` (or `sgetc`). Finally, it destroys the `sentry` object before returning.

The number of characters successfully read and stored by this function can be accessed by calling member `gcount`.

Parameters

s

Pointer to an array where the extracted characters are stored.

n

Maximum number of characters to extract.
`streamsize` is a signed integral type.

Return Value

The number of characters stored.
`streamsize` is a signed integral type.

Errors are signaled by modifying the *internal state flags*:

flag	error
`eofbit`	The input sequence has no characters available (as reported by `rdbuf()->in_avail()` returning -1).
`failbit`	The construction of `sentry` failed (such as when the *stream state* was not `good` before the call).
`badbit`	Error on stream (such as when this function catches an exception thrown by an internal operation). When set, the integrity of the stream may have been affected.

Multiple flags may be set by a single operation.

If the operation sets an *internal state flag* that was registered with member `exceptions`, the function throws an exception of member type `failure`.

Data races

Modifies the elements in the array pointed by *s* and the stream object.
Concurrent access to the same stream object may cause data races, except for the standard stream object `cin` when this is *synchronized with stdio* (in this case, no data races are initiated, although no guarantees are given on the order in which extracted characters are attributed to threads).

Exception safety

Basic guarantee: if an exception is thrown, the object is in a valid state.
It throws an exception of member type `failure` if the resulting *error state flag* is not `goodbit` and member `exceptions` was set to throw for that state.
Any exception thrown by an internal operation is caught and handled by the function, setting `badbit`. If `badbit` was set on the last call to `exceptions`, the function rethrows the caught exception.

See also

istream::read	Read block of data (public member function)
ostream::write	Write block of data (public member function)
istream::operator>>	Extract formatted input (public member function)

Home page | Privacy policy

Spotted an error? contact us