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::sync

<istream> <iostream>

`int sync();`

Synchronize input buffer

Synchronizes the associated *stream buffer* with its controlled input sequence.

Specifics of the operation depend on the particular implementation of the *stream buffer* object associated to the stream.

Internally, the function accesses the input sequence by first constructing a `sentry` object (with *noskipws* set to `true`). Then (if `good`), it calls `pubsync` on its associated *stream buffer* object (if `rdbuf` is null, the function returns `-1` instead). Finally, it destroys the `sentry` object before returning.

If the call to `pubsync` fails (i.e., it returns `-1`), the function sets the `badbit` flag, and returns `-1`. Otherwise it returns zero, indicating success.

Notice that the called function may succeed when no action is performed, if that is the behavior defined for the *stream buffer* object on synchronization.

Calling this function does not alter the value returned by `gcount`.

Parameters

none

Return Value

If the function fails, either because no *stream buffer* object is associated to the stream (`rdbuf` is null), or because the call to its `pubsync` member fails, it returns `-1`.
Otherwise, it returns zero, indicating success.

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

flag	error
`eofbit`	-
`failbit`	The construction of `sentry` failed (such as when the *stream state* was not `good` before the call).
`badbit`	Either the internal call to `pubsync` returned `-1`, or some other error occurred on the stream (such as when the 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`.

Example

```cpp
// syncing input stream
#include <iostream>     // std::cin, std::cout

int main () {
  char first, second;

  std::cout << "Please, enter a word: ";
  first = std::cin.get();
  std::cin.sync();

  std::cout << "Please, enter another word: ";
  second = std::cin.get();

  std::cout << "The first word began by " << first << '\n';
  std::cout << "The second word began by " << second << '\n';

  return 0;
}
```

This example demonstrates how `sync` behaves on certain implementations of `cin`, removing any unread character from the standard input queue of characters.

Possible output:

```
Please, enter a word: test
Please enter another word: text
The first word began by t
The second word began by t
```

Data races

Modifies 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 characters are extracted or synchronized between 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

ostream::flush	Flush output stream buffer (public member function)
streambuf::pubsync	Synchronize stream buffer (public member function)