



Search:

Reference <fstream> filebuf sync

Not logged 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:

<fstream>

class templates:

basic_filebuf

basic_fstream

basic_ifstream

basic_ofstream

classes:

filebuf

fstream

ifstream

ofstream

wfilebuf

wfstream

wifstream

wofstream

filebuf

filebuf::filebuf

filebuf::~filebuf

public members:

filebuf::close

filebuf::is_open

filebuf::open

filebuf::operator=

filebuf::swap

protected virtual members:

filebuf::imbue

filebuf::overflow

filebuf::pbackfail

filebuf::seekoff

filebuf::seekpos

filebuf::setbuf

filebuf::showmanyc

filebuf::sync

filebuf::uflow

filebuf::underflow

non-member overloads:

swap (filebuf)

PatchIT Updating Libraries

PatchIT offers fully automated updating libraries for coding

protected virtual member function

std::filebuf::sync

<fstream>

int sync();

Synchronize buffer

Synchronizes the intermediate buffers with the associated file:

If an *intermediate output buffer* exists (as defined by `pptr`, `pptr` and `pback`), its contents are written to the file (as if `filebuf::overflow` was called).

If an intermediate input buffer exists, the effects -if any- depend on the library implementation.

This virtual member function overrides the inherited member `streambuf::sync`, called by member `pubsync` to synchronize the buffers.

Parameters

none

Return Value

If successful, it returns zero.
Otherwise, it returns -1 to indicate failure.

Data races

Modifies the `filebuf` object.
Concurrent access to the same *file stream buffer* object may introduce data races.

Exception safety

Basic guarantee: if an exception is thrown, the *file stream buffer* is in a valid state.

See also

streambuf::pubsync

Synchronize stream buffer (public member function)

streambuf::sync

Synchronize stream buffer (protected virtual member function)

Home page | Privacy policy

© cplusplus.com, 2000-2015 - All rights reserved - v3.1

Spotted an error? contact us