



Search:

[Reference](#)
[<fstream>](#)
[filebuf](#)
[close](#)

[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:

<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 Librar

PatchIT offers fully automated updating libraries for coding

public member function

std::filebuf::close

<fstream>

```
filebuf* close();
```

Close file

Closes the file currently associated with the object, disassociating them.

Any pending output sequence is written to the file (as if calling virtual member `overflow`), along with any unshift characters (`codecvt::unshift`), when needed.

The function fails if any of these operations fail, or if no file is currently open.

Parameters

none

Return Value

The function returns this, if successful.  
In case of failure (including when the file is *not open*), a *null pointer* is returned.

Example

```
1 // filebuf::close()
2 #include <iostream>
3 #include <fstream>
4
5 int main () {
6     std::ifstream is;
7     std::filebuf * fb = is.rdbuf();
8
9     fb->open ("test.txt", std::ios::in);
10
11     // appending operations
12
13     fb->close();
14
15     return 0;
16 }
```

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.  
Any exception thrown by an internal operation is caught by the function and rethrown after closing the file.

See also

[filebuf::open](#)
Open file (public member function )

[filebuf::is\\_open](#)
Check if a file is open (public member function )

Home page | Privacy policy

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

Spotted an error? contact us