

Search: 

Go

Reference

&lt;fstream&gt;

ofstream

close

## C++

Information

Tutorials

Reference

Articles

Forum

## Reference

## C library:

&lt;cassert&gt; (assert.h)

&lt;cctype&gt; (ctype.h)

&lt;cerrno&gt; (errno.h)

&lt;cfenv&gt; (fenv.h)

&lt;cfloat&gt; (float.h)

&lt; cinttypes&gt; (inttypes.h)

&lt;ciso646&gt; (iso646.h)

&lt;climits&gt; (limits.h)

&lt;locale&gt; (locale.h)

&lt;cmath&gt; (math.h)

&lt;csetjmp&gt; (setjmp.h)

&lt;csignal&gt; (signal.h)

&lt;csdarg&gt; (stdarg.h)

&lt;csdbool&gt; (stdbool.h)

&lt;csddef&gt; (stddef.h)

&lt;csdint&gt; (stdint.h)

public member function

**std::ofstream::close**

&lt;fstream&gt;

void close();

**Close file**

Closes the file currently associated with the object, disassociating it from the stream.

Any pending output sequence is written to the file.

If the stream is currently not associated with any file (i.e., no file has successfully been *open* with it), calling this function fails.The file association of a stream is kept by its *internal stream buffer*:Internally, the function calls `rdbuf()->close()`, and sets *failbit* in case of failure.Note that any open file is automatically closed when the *ofstream* object is destroyed. **Return Value**

none

If the operation fails (including if no file was *open* before the call), the *failbit state flag* is set for the stream (which may throw `ios_base::failure` if that *state flag* was registered using member *exceptions*). **Example**

```

1 // ofstream::open / ofstream::close
2 #include <fstream>           // std::ofstream
3
4 int main () {
5
6     std::ofstream ofs;
7     ofs.open ("test.txt", std::ofstream::out | std::ofstream::app);
8
9     ofs << " more lorem ipsum";
10
11     ofs.close();
12
13     return 0;
14 }

```

**Data races**Modifies the *ofstream* object.Concurrent access to the same *stream* may introduce data races. **Exception safety****Basic guarantee:** if an exception is thrown, the *stream* is in a valid state.

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

It throws an exception of member type *failure* if the function fails (setting the *failbit* state flag) and member *exceptions* was set to throw for that state. **See also**

<b>ofstream::open</b>	Open file (public member function )
<b>ofstream::is_open</b>	Check if file is open (public member function )
<b>filebuf::close</b>	Close file (public member function )