public member function

std::filebuf::open

<fstream>

```
filebuf* open (const char* filename,  ios_base::openmode mode);
```

Open file

Opens the file identified by argument *filename*, associating its content with the *file stream buffer* object to perform input/output operations on it. The operations allowed and some operating details depend on parameter *mode*.

If the object is already associated with a file (i.e., it is already *open*), this function fails.

Parameters

`filename`
String with the name of the file to open.

`mode`
Flags describing the requested input/output mode for the file.
This is an object of the bitmask type `ios_base::openmode` that consists of a combination of the following constants:

value	stands for	access
`ios_base::in`	**in**put	File open for reading, supporting input operations.
`ios_base::out`	**out**put	File open for writing, supporting output operations.
`ios_base::binary`	**binary**	Operations are performed in binary mode rather than text.
`ios_base::ate`	**at e**nd	The *put pointer* (`pptr`) starts at the end of the controlled output sequence.
`ios_base::app`	**app**end	All output operations happen at the end of the file, appending to its existing contents.
`ios_base::trunc`	**trunc**ate	Any contents that existed in the file before it is open are discarded.

These flags can be combined with the bitwise OR operator (|).

If the mode has both `ios_base::trunc` and `ios_base::app` set, the opening operation fails. It also fails if either is set but `ios_base::out` is not, or if both `ios_base::app` and `ios_base::in` are set.
If the mode has both `ios_base::trunc` and `ios_base::app` set, the opening operation fails. It also fails if `ios_base::trunc` is set but `ios_base::out` is not.

Return Value

The function returns `this` if successful.
In case of failure, the file is not open, and a *null pointer* is returned.

Example

```
1  // filebuf::open()
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::out|std::ios::app);
10
11   // >> appending operations here <<
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.

See also

filebuf::is_open	Check if a file is open (public member function)
filebuf::close	Close file (public member function)