

GCC Code Coverage Report

Directory: **./**

File: **storage/filesystem/source/File.cpp**

Date: **2021-05-06 12:39:05**

	Exec	Total	Coverage
Lines:	29	50	58.0 %
Branches:	9	28	32.1 %

Line	Branch	Exec	Source
1			/* mbed Microcontroller Library
2			* Copyright (c) 2015 ARM Limited
3			* SPDX-License-Identifier: Apache-2.0
4			*
5			* Licensed under the Apache License, Version 2.0 (the "License");
6			* you may not use this file except in compliance with the License.
7			* You may obtain a copy of the License at
8			*
9			* http://www.apache.org/licenses/LICENSE-2.0
10			*
11			* Unless required by applicable law or agreed to in writing, software
12			* distributed under the License is distributed on an "AS IS" BASIS,
13			* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14			* See the License for the specific language governing permissions and
15			* limitations under the License.
16			*/
17			
18			#include "filesystem/File.h"
19			#include <errno.h>
20			
21			namespace mbed {
22			
23		304	File::File()
24		304	: _fs(0), _file(0)
25			{
26		304	}
27			
28			File::File(FileSystem *fs, const char *path, int flags)
29			: _fs(0), _file(0)
30			{

```

31 } open(fs, path, flags);
32
33
34 711 File::~File()
35 {
36 x✓ 304 if (_fs) {
37     close();
38 }
39 407 }
40
41 407 int File::open(FileSystem *fs, const char *path, int flags)
42 {
43 x✓ 407 if (_fs) {
44     return -EINVAL;
45 }
46
47 407 int err = fs->file_open(&_file, path, flags);
48 ✓✓ 407 if (!err) {
49 304     _fs = fs;
50 }
51
52 407 return err;
53 }
54
55 304 int File::close()
56 {
57 x✓ 304 if (!_fs) {
58     return -EINVAL;
59 }
60
61 304 int err = _fs->file_close(_file);
62 304 _fs = 0;
63 304 return err;
64 }
65
66 302 ssize_t File::read(void *buffer, size_t len)
67 {
68 x✓ 302 MBED_ASSERT(_fs);
69 302 return _fs->file_read(_file, buffer, len);
70 }

```

```

71
72 206 ssize_t File::write(const void *buffer, size_t len)
73 {
74 x✓ 206 MBED_ASSERT(_fs);
75 206 return _fs->file_write(_file, buffer, len);
76 }
77
78 int File::sync()
79 {
80 MBED_ASSERT(_fs);
81 return _fs->file_sync(_file);
82 }
83
84 int File::isatty()
85 {
86 MBED_ASSERT(_fs);
87 return _fs->file_isatty(_file);
88 }
89
90 101 off_t File::seek(off_t offset, int whence)
91 {
92 x✓ 101 MBED_ASSERT(_fs);
93 101 return _fs->file_seek(_file, offset, whence);
94 }
95
96 off_t File::tell()
97 {
98 MBED_ASSERT(_fs);
99 return _fs->file_tell(_file);
100 }
101
102 void File::rewind()
103 {
104 MBED_ASSERT(_fs);
105 return _fs->file_rewind(_file);
106 }
107
108 101 off_t File::size()
109 {
110 x✓ 101 MBED_ASSERT(_fs);

```

111			
112		101	} return _fs->file_size(_file);
113			
114			int File::truncate(off_t length)
115			{
116			MBED_ASSERT(_fs);
117			return _fs->file_truncate(_file, length);
118			}
119			
120			} // namespace mbed

Generated by: [GCOVR \(Version 4.2\)](#)