# GCC Code Coverage Report

| | | | Exec | Total | Coverage |
|---|---|---|---|---|---|
| Directory: | ./ | | | | |
| File: | storage/blockdevice/source/ObservingBlockDevice.cpp | Lines: | 41 | 44 | 93.2 % |
| Date: | 2021-05-06 12:39:05 | Branches: | 7 | 14 | 50.0 % |

| Line | Branch | Exec | Source |
|---|---|---|---|
| 1 | | | /* mbed Microcontroller Library |
| 2 | | | * Copyright (c) 2017-2020 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 "blockdevice/ObservingBlockDevice.h" |
| 19 | | | #include "blockdevice/ReadOnlyBlockDevice.h" |
| 20 | | | |
| 21 | | | namespace mbed { |
| 22 | | | |
| 23 | | 2 | ObservingBlockDevice::ObservingBlockDevice(BlockDevice *bd) |
| 24 | | 2 | :  _bd(bd) |
| 25 | | | { |
| 26 | | | // Does nothing |
| 27 | | 2 | } |
| 28 | | | |
| 29 | | 2 | ObservingBlockDevice::~ObservingBlockDevice() |
| 30 | | | { |

```
31          2      // Does nothing
32          2  }
33
34          1  void ObservingBlockDevice::attach(mbed::Callback<void(BlockDevice *)> cb)
35             {
36          1      _change = cb;
37          1  }
38
39          2  int ObservingBlockDevice::init()
40             {
41          2      return _bd->init();
42             }
43
44          2  int ObservingBlockDevice::deinit()
45             {
46          2      return _bd->deinit();
47             }
48
49             int ObservingBlockDevice::sync()
50             {
51                 return _bd->sync();
52             }
53
54          1  int ObservingBlockDevice::read(void *buffer, bd_addr_t addr, bd_size_t size)
55             {
56          1      return _bd->read(buffer, addr, size);
57             }
58
59          1  int ObservingBlockDevice::program(const void *buffer, bd_addr_t addr, bd_size_t size)
60             {
61          1      int res = _bd->program(buffer, addr, size);
62    ✓✗    1      if (_change) {
63    ✓✗    2          ReadOnlyBlockDevice dev(_bd);
64    ✓✗    1          _change(&dev);
65             }
66          1      return res;
67             }
68
69          1  int ObservingBlockDevice::erase(bd_addr_t addr, bd_size_t size)
70             {
```

```
71      1     int res = _bd->erase(addr, size);
72  ✓✗  1     if (_change) {
73  ✓✗  2         ReadOnlyBlockDevice dev(_bd);
74  ✓✗  1         _change(&dev);
75          }
76      1     return res;
77      }
78
79      1 bd_size_t ObservingBlockDevice::get_read_size() const
80      {
81      1     return _bd->get_read_size();
82      }
83
84      1 bd_size_t ObservingBlockDevice::get_program_size() const
85      {
86      1     return _bd->get_program_size();
87      }
88
89      1 bd_size_t ObservingBlockDevice::get_erase_size() const
90      {
91      1     return _bd->get_erase_size();
92      }
93
94      1 bd_size_t ObservingBlockDevice::get_erase_size(bd_addr_t addr) const
95      {
96      1     return _bd->get_erase_size(addr);
97      }
98
99      1 int ObservingBlockDevice::get_erase_value() const
100     {
101     1     return _bd->get_erase_value();
102     }
103
104     1 bd_size_t ObservingBlockDevice::size() const
105     {
106     1     return _bd->size();
107     }
108
109     1 const char *ObservingBlockDevice::get_type() const
110     {
```

```
111  ✓✗  1      if (_bd != NULL) {
112      1          return _bd->get_type();
113              }
114
115              return NULL;
116          }
117
118          } // namespace mbed
```

Generated by: GCOVR (Version 4.2)