Contents

Foreword vi

1 Scope 1

2 Normative references 2

3 Terms and definitions 3

4 General Principles 4
4.1 Conformance . . . 4
4.2 Acknowledgments . . . 4

5 Namespaces and headers 5

6 Future plans (Informative) 6

7 Feature test macros (Informative) 7

8 Method of description (Informative) 8
8.1 Structure of each clause . . . 8
8.2 Other conventions . . . 8

9 Error reporting 9
9.1 Synchronous operations . . . 9
9.2 Asynchronous operations . . . 10
9.3 Error conditions . . . 10
9.4 Suppression of signals . . . 10

10 Library summary 11

11 Convenience header 13
11.1 Header `<experimental/net>` synopsis . . . 13

12 Forward declarations 14
12.1 Header `<experimental/netfwd>` synopsis . . . 14

13 Asynchronous model 16
13.1 Header `<experimental/executor>` synopsis . . . 16
13.2 Requirements . . . 19
13.3 Class template `async_result` . . . 27
13.4 Class template `async_completion` . . . 28
13.5 Class template `associated_allocator` . . . 29
13.6 Function `get_associated_allocator` . . . 30
13.7 Class `execution_context` . . . 30
13.8 Class `execution_context::service` . . . 32
13.9 Class template `is_executor` . . . 33
13.10 Executor argument tag . . . 33
13.11 `uses_executor` . . . 34

13.12 Class template `associated_executor` . . . 34
13.13 Function `get_associated_executor` . . . 35
13.14 Class template `executor_binder` . . . 36
13.15 Function `bind_executor` . . . 39
13.16 Class template `executor_work_guard` . . . 40
13.17 Function `make_work_guard` . . . 41
13.18 Class `system_executor` . . . 42
13.19 Class `system_context` . . . 43
13.20 Class `bad_executor` . . . 44
13.21 Class `executor` . . . 45
13.22 Function `dispatch` . . . 49
13.23 Function `post` . . . 50
13.24 Function `defer` . . . 51
13.25 Class template `strand` . . . 52
13.26 Class template `use_future_t` . . . 56
13.27 Partial specialization of `async_result` for `packaged_task` . . . 59

14 Basic I/O services **61**
14.1 Header `<experimental/io_context>` synopsis . . . 61
14.2 Class `io_context` . . . 61
14.3 Class `io_context::executor_type` . . . 65

15 Timers **67**
15.1 Header `<experimental/timer>` synopsis . . . 67
15.2 Requirements . . . 67
15.3 Class template `wait_traits` . . . 68
15.4 Class template `basic_waitable_timer` . . . 69

16 Buffers **73**
16.1 Header `<experimental/buffer>` synopsis . . . 73
16.2 Requirements . . . 78
16.3 Error codes . . . 82
16.4 Class `mutable_buffer` . . . 82
16.5 Class `const_buffer` . . . 83
16.6 Buffer type traits . . . 84
16.7 Buffer sequence access . . . 85
16.8 Function `buffer_size` . . . 85
16.9 Function `buffer_copy` . . . 85
16.10 Buffer arithmetic . . . 86
16.11 Buffer creation functions . . . 86
16.12 Class template `dynamic_vector_buffer` . . . 88
16.13 Class template `dynamic_string_buffer` . . . 89
16.14 Dynamic buffer creation functions . . . 91

17 Buffer-oriented streams **92**
17.1 Requirements . . . 92
17.2 Class `transfer_all` . . . 94
17.3 Class `transfer_at_least` . . . 95
17.4 Class `transfer_exactly` . . . 95
17.5 Synchronous read operations . . . 96
17.6 Asynchronous read operations . . . 98

17.7 Synchronous write operations . . . 99
17.8 Asynchronous write operations . . . 100
17.9 Synchronous delimited read operations . . . 102
17.10 Asynchronous delimited read operations . . . 102

18 Sockets **104**
18.1 Header `<experimental/socket>` synopsis . . . 104
18.2 Requirements . . . 106
18.3 Error codes . . . 115
18.4 Class `socket_base` . . . 116
18.5 Socket options . . . 118
18.6 Class template `basic_socket` . . . 121
18.7 Class template `basic_datagram_socket` . . . 131
18.8 Class template `basic_stream_socket` . . . 139
18.9 Class template `basic_socket_acceptor` . . . 145

19 Socket iostreams **157**
19.1 Class template `basic_socket_streambuf` . . . 157
19.2 Class template `basic_socket_iostream` . . . 161

20 Socket algorithms **164**
20.1 Synchronous connect operations . . . 164
20.2 Asynchronous connect operations . . . 165

21 Internet protocol **167**
21.1 Header `<experimental/internet>` synopsis . . . 167
21.2 Requirements . . . 171
21.3 Error codes . . . 173
21.4 Class `ip::address` . . . 174
21.5 Class `ip::address_v4` . . . 177
21.6 Class `ip::address_v6` . . . 181
21.7 Class `ip::bad_address_cast` . . . 186
21.8 Hash support . . . 187
21.9 Class template `ip::basic_address_iterator` specializations . . . 187
21.10 Class template `ip::basic_address_range` specializations . . . 188
21.11 Class template `ip::network_v4` . . . 190
21.12 Class template `ip::network_v6` . . . 193
21.13 Class template `ip::basic_endpoint` . . . 195
21.14 Class template `ip::basic_resolver_entry` . . . 199
21.15 Class template `ip::basic_resolver_results` . . . 201
21.16 Class `ip::resolver_base` . . . 204
21.17 Class template `ip::basic_resolver` . . . 205
21.18 Host name functions . . . 211
21.19 Class `ip::tcp` . . . 211
21.20 Class `ip::udp` . . . 212
21.21 Internet socket options . . . 214

Index **219**

Index of library names **221**

Index of implementation-defined behavior **227**