18. Section **??** on page **??**.
    New functions for a scalable distributed graph topology interface has been added. In this section, the functions MPI_DIST_GRAPH_CREATE_ADJACENT and MPI_DIST_GRAPH_CREATE, the constants MPI_UNWEIGHTED, and the derived C++ class Distgraphcomm were added.

19. Section 7.5.4 on page 248.
    For the scalable distributed graph topology interface, the functions MPI_DIST_NEIGHBORS_COUNT and MPI_DIST_NEIGHBORS and the constant MPI_DIST_GRAPH were added.

20. Section 7.5.4 on page 248.
    Remove ambiguity regarding duplicated neighbors with MPI_GRAPH_NEIGHBORS and MPI_GRAPH_NEIGHBORS_COUNT.

21. Section 8.1.1 on page 259.
    The subversion number changed from 1 to 2.

22. Section 8.3 on page 264, Section **??** on page **??**, and Annex A.1.3 on page 499.
    Changed function pointer typedef names MPI_{Comm,File,Win}_errhandler_fn to MPI_{Comm,File,Win}_errhandler_function. Deprecated old "_fn" names.

23. Section **??** on page **??**, and Section **??** on page **??**.
    Added const versions of MPI::File::Set_errhandler and MPI::Win::Set_errhandler. This allows MPI::FILE_NULL and MPI::WIN_NULL – regardless of whether they are const or not – to invoke these functions.

24. Section **??** on page **??**.
    Attribute deletion callbacks on MPI_COMM_SELF are now called in LIFO order. Implementors must now also register all implementation-internal attribute deletion callbacks on MPI_COMM_SELF before returning from MPI_INIT/MPI_INIT_THREAD.

25. Section 11.3.4 on page 331.
    The restriction added in MPI 2.1 that the operation MPI_REPLACE in MPI_ACCUMULATE can be used only with predefined datatypes has been removed. MPI_REPLACE can now be used even with derived datatypes, as it was in MPI 2.0. Also, a clarification has been made that MPI_REPLACE can be used only in MPI_ACCUMULATE, not in collective operations that do reductions, such as MPI_REDUCE and others.

26. Section 12.2 on page 357.
    Add "*" to the query_fn, free_fn, and cancel_fn arguments to the C++ binding for MPI::Grequest::Start() for consistency with the rest of MPI functions that take function pointer arguments.

27. Section 13.5.2 on page 414, and Table **??** on page **??**.
    MPI_(U)INT{8,16,32,64}_T, MPI_AINT, MPI_OFFSET, MPI_C_COMPLEX, MPI_C_FLOAT_COMPLEX, MPI_C_DOUBLE_COMPLEX, MPI_C_LONG_DOUBLE_COMPLEX, and MPI_C_BOOL are added as predefined datatypes in the external32 representation.