Annex B

Change-Log

This annex summarizes changes from the previous version of the MPI standard to the version presented by this document. Only significant changes (i.e., clarifications and new features) that might either require implementation effort in the MPI libraries or change the understanding of MPI from a user's perspective are presented. Editorial modifications, formatting, typo corrections and minor clarifications are not shown.

B.1 Changes from Version 3.0 to Version 3.1

B.1.1 Fixes to Errata in Previous Versions of MPI

#388 1. Chapters 3–13, Annex A.3 on page 712, and Example 5.21 on page 187, and
MPI-3.0 Chapters 3-17, Annex A.3 on page 707, and Example 5.21 on page 187.
Within the `mpi_f08` Fortran support method, `BIND(C)` was removed from all `SUBROUTINE`, `FUNCTION`, and `ABSTRACT INTERFACE` definitions.

#415 2. Section 3.2.5 on page 30, and
MPI-3.0 Section 3.2.5 on page 30.
The three public fields MPI_SOURCE, MPI_TAG, and MPI_ERROR of the Fortran derived type `TYPE(MPI_Status)` must be of type `INTEGER`.

#424 3. Section 3.8.2 on page 67, and
MPI-3.0 Section 3.8.2 on page 67.
The flag arguments of the Fortran interfaces of MPI_IMPROBE were originally incorrectly defined as `INTEGER` (instead as `LOGICAL`). xxx

#345 4. Section 6.4.2 on page 237, and
MPI-3.0 Section 6.4.2 on page 237.
In the `mpi_f08` binding of MPI_COMM_IDUP, the output argument newcomm is declared as `ASYNCHRONOUS`.

#345 5. Section 6.4.4 on page 248, and
MPI-3.0 Section 6.4.4 on page 248.
In the `mpi_f08` binding of MPI_COMM_SET_INFO, the intent of comm is `IN`, and the optional output argument ierror was missing.

#419 6. Section 7.6 on page 314, and
MPI-3.0 Sections 7.6, on pages 314.

In the case of virtual general graph topolgies (created with MPI_CART_CREATE), the use of neighborhood collective communication is restricted to adjacency matrices with the number of edges between any two processes is defined to be the same for both processes (i.e., with a symmetric adjacency matrix).

#345 7. Section 8.1.1 on page 333, and
MPI-3.0 Section 8.1.1 on page 335.
In the `mpi_f08` binding of MPI_GET_LIBRARY_VERSION, a typo in the resultlen argument was corrected.

#388 8. MPI-3.0 Sections 8.2 (MPI_ALLOC_MEM and MPI_ALLOC_MEM_CPTR),
11.2.2 (MPI_WIN_ALLOCATE and MPI_WIN_ALLOCATE_CPTR),
11.2.3 (MPI_WIN_ALLOCATE_SHARED and MPI_WIN_ALLOCATE_SHARED_CPTR),
11.2.3 (MPI_WIN_SHARED_QUERY and MPI_WIN_SHARED_QUERY_CPTR),
14.2.1 and 14.2.7 (Profiling interface), and corresponding sections in the current version of this standard.
The linker name concept was substituted by defining specific procedure names.

#362 9. Section 11.2.2 on page 405, and
MPI-3.0 Section 11.2.2 on page 407.
The same_size info key can be used with all window flavors.

#350 10. Section 11.3.4 on page 423, and
MPI-3.0 Section 11.3.4 on page 424.
Origin buffer arguments to MPI_GET_ACCUMULATE are ignored when the MPI_NO_OP operation is used.

#355 11. Section 11.3.4 on page 423, and
MPI-3.0 Section 11.3.4 on page 424.
Clarify the roles of origin, result, and target communication parameters in MPI_GET_ACCUMULATE.

#383 12. Section 14.3 on page 565, and
MPI-3.0 Section 14.3 on page 561
New paragraph and advice to users clarifying intent of variable names in the tools information interface.

#383 13. Section 14.3.3 on page 567, and
MPI-3.0 Section 14.3.3 on page 563.
New paragraph clarifying variable name equivalence in the tools information interface.

#383 14. Sections 14.3.6, 14.3.7, and 14.3.8 on pages 571, 578, and 590, and
MPI-3.0 Sections 14.3.6, 14.3.7, and 14.3.8 on pages 567, 573, and 584.
In functions MPI_T_CVAR_GET_INFO, MPI_T_PVAR_GET_INFO, and MPI_T_CATEGORY_GET_INFO, clarification of parameters that must be identical for equivalent control variable / performance variable / category names across connected processes.

#391 15. Section 14.3.7 on page 578, and
MPI-3.0 Section 14.3.7 on page 573.
Clarify return code of MPI_T_PVAR_{START,STOP,RESET} routines.

#386 16. Section 14.3.7 on page 580, and
MPI-3.0 Section 14.3.7 on page 579, line 7.
Clarify the return code when bad handle is passed to an MPI_T_PVAR_* routine.

#388 17. Section 17.1.3 on page 609, and
MPI-3.0 Section 17.1.4 on page 603.
The advice to implementors at the end of the section was rwritten and moved into the following section.

#388 18. Section 17.1.5 on page 612, and
MPI-3.0 Section 17.1.5 on page 605.
The section was fully rewritten. The linker name concept was substituted by defining specific procedure names.

#388 19. Section 17.1.6 on page 617, and
MPI-3.0 Section 17.1.6 on page 611.
The requirements on `BIND(C)` procedure interfaces are removed.

#389 20. Annexes A.2, A.3, and A.4 on pages 692, 714, and 763, and
MPI-3.0 Annexes A.2, A.3, and A.4 on pages 685, 707, and 756.
The predefined callback MPI_CONVERSION_FN_NULL was added to all three annexes.

#345 21. Annex A.3.4 on page 731, and MPI-3.0 Annex A.3.4 on page 724.
In the `mpi_f08` binding of
MPI_{COMM|TYPE|WIN}_{DUP|NULL_COPY|NULL_DELETE}_FN, all `INTENT(...)` information must be removed.

B.1.2 Changes in MPI-3.1

#349+ #402+ #404+ #421 1. Sections 2.6.4 and 4.1.5 on pages 20 and 101.
The use of the intrinsic operators "+" and "−" for absolute addresses is substituted by MPI_AINT_ADD and MPI_AINT_DIFF. In C, they can be implemented as macros.

#357 2. Sections 8.7 and 12.4 on pages 355 and 484.
The routines MPI_INITIALIZED, MPI_FINALIZED, MPI_QUERY_THREAD, and MPI_IS_THREAD_MAIN are callable from threads without restriction (in the sense of MPI_THREAD_MULTIPLE), irrespective of the actual level of thread support provided, in the case where the implementation supports threads.

#369 3. Section 11.2.1 on page 403.
The same_disp_unit info key was added for use in RMA window creation routines.

#273 4. Sections 13.4.2 and 13.4.3 on pages 509 and 514.
Added MPI_File_iread_at_all, MPI_File_iwrite_at_all, MPI_File_iread_all, and MPI_File_iwrite_all

#378 5. Sections 14.3.6, 14.3.7, and 14.3.8 on pages 573, 580, and 592.
Clarified that `NULL` parameters can be provided in
MPI_T_{CVAR|PVAR|CATEGORY}_GET_INFO routines.

#377+#400

6. Sections 14.3.6, 14.3.7, 14.3.8, and 14.3.9 on pages 573, 580, 592, and 596.
   New routines MPI_T_CVAR_GET_INDEX, MPI_T_PVAR_GET_INDEX, MPI_T_CATEGORY_GET_INDEX, were added to support retrieving indices of variables and categories. The error codes MPI_T_ERR_INVALID and MPI_T_ERR_INVALID_NAME were added to indicate invalid uses of the interface.

B.2 Changes from Version 2.2 to Version 3.0

B.2.1 Fixes to Errata in Previous Versions of MPI

1. Sections 2.6.2 and 2.6.3 on pages 19 and 19, and
   MPI-2.2 Section 2.6.2 on page 17, lines 41-42, Section 2.6.3 on page 18, lines 15-16, and Section 2.6.4 on page 18, lines 40-41.
   This is an MPI-2 erratum: The scope for the reserved prefix `MPI_` and the C++ namespace `MPI` is now any name as originally intended in MPI-1.

2. Sections 3.2.2, 5.9.2, 13.6.2 Table 13.2, and Annex A.1.1 on pages 25, 176, 540, and 669, and
   MPI-2.2 Sections 3.2.2, 5.9.2, 13.5.2 Table 13.2, 16.1.16 Table 16.1, and Annex A.1.1 on pages 27, 164, 433, 472 and 513
   This is an MPI-2.2 erratum: New named predefined datatypes MPI_CXX_BOOL, MPI_CXX_FLOAT_COMPLEX, MPI_CXX_DOUBLE_COMPLEX, and MPI_CXX_LONG_DOUBLE_COMPLEX were added in C and Fortran corresponding to the C++ types `bool`, `std::complex<float>`, `std::complex<double>`, and `std::complex<long double>`. These datatypes also correspond to the deprecated C++ predefined datatypes MPI::BOOL, MPI::COMPLEX, MPI::DOUBLE_COMPLEX, and MPI::LONG_DOUBLE_COMPLEX, which were removed in MPI-3.0. The non-standard C++ types `Complex<...>` were substituted by the standard types `std::complex<...>`.

3. Sections 5.9.2 on pages 176 and MPI-2.2 Section 5.9.2, page 165, line 47.
   This is an MPI-2.2 erratum: MPI_C_COMPLEX was added to the "Complex" reduction group.

4. Section 7.5.5 on page 302, and
   MPI-2.2, Section 7.5.5 on page 257, C++ interface on page 264, line 3.
   This is an MPI-2.2 erratum: The argument rank was removed and in/outdegree are now defined as `int& indegree` and `int& outdegree` in the C++ interface of MPI_DIST_GRAPH_NEIGHBORS_COUNT.

5. Section 13.6.2, Table 13.2 on page 540, and
   MPI-2.2, Section 13.5.3, Table 13.2 on page 433.
   This was an MPI-2.2 erratum: The MPI_C_BOOL "external32" representation is corrected to a 1-byte size.

6. MPI-2.2 Section 16.1.16 on page 471, line 45.
   This is an MPI-2.2 erratum: The constant MPI::_LONG_LONG should be MPI::LONG_LONG.