

Table 2.1 shows a list of all of the deprecated constructs. Note that the constants `MPI_LB` and `MPI_UB` are replaced by the function `MPI_TYPE_CREATE_RESIZED`; this is because their principal use was as input datatypes to `MPI_TYPE_STRUCT` to create resized datatypes. Also note that some C typedefs and Fortran subroutine names are included in this list; they are the types of callback functions.

| Deprecated | MPI-2 Replacement |
|------------------------------------|---------------------------------------------------------|
| <code>MPI_ADDRESS</code> | <code>MPI_GET_ADDRESS</code> |
| <code>MPI_TYPE_HINDEXED</code> | <code>MPI_TYPE_CREATE_HINDEXED</code> |
| <code>MPI_TYPE_HVECTOR</code> | <code>MPI_TYPE_CREATE_HVECTOR</code> |
| <code>MPI_TYPE_STRUCT</code> | <code>MPI_TYPE_CREATE_STRUCT</code> |
| <code>MPI_TYPE_EXTENT</code> | <code>MPI_TYPE_GET_EXTENT</code> |
| <code>MPI_TYPE_UB</code> | <code>MPI_TYPE_GET_EXTENT</code> |
| <code>MPI_TYPE_LB</code> | <code>MPI_TYPE_GET_EXTENT</code> |
| <code>MPI_LB</code> | <code>MPI_TYPE_CREATE_RESIZED</code> |
| <code>MPI_UB</code> | <code>MPI_TYPE_CREATE_RESIZED</code> |
| <code>MPI_ERRHANDLER_CREATE</code> | <code>MPI_COMM_CREATE_ERRHANDLER</code> |
| <code>MPI_ERRHANDLER_GET</code> | <code>MPI_COMM_GET_ERRHANDLER</code> |
| <code>MPI_ERRHANDLER_SET</code> | <code>MPI_COMM_SET_ERRHANDLER</code> |
| <code>MPI_Handler_function</code> | <code>MPI_Comm_errhandler_[ticket7.][fn]function</code> |
| <code>MPI_KEYVAL_CREATE</code> | <code>MPI_COMM_CREATE_KEYVAL</code> |
| <code>MPI_KEYVAL_FREE</code> | <code>MPI_COMM_FREE_KEYVAL</code> |
| <code>MPI_DUP_FN</code> | <code>MPI_COMM_DUP_FN</code> |
| <code>MPI_NULL_COPY_FN</code> | <code>MPI_COMM_NULL_COPY_FN</code> |
| <code>MPI_NULL_DELETE_FN</code> | <code>MPI_COMM_NULL_DELETE_FN</code> |
| <code>MPI_Copy_function</code> | <code>MPI_Comm_copy_attr_function</code> |
| <code>COPY_FUNCTION</code> | <code>COMM_COPY_ATTR_FN</code> |
| <code>MPI_Delete_function</code> | <code>MPI_Comm_delete_attr_function</code> |
| <code>DELETE_FUNCTION</code> | <code>COMM_DELETE_ATTR_FN</code> |
| <code>MPI_ATTR_DELETE</code> | <code>MPI_COMM_DELETE_ATTR</code> |
| <code>MPI_ATTR_GET</code> | <code>MPI_COMM_GET_ATTR</code> |
| <code>MPI_ATTR_PUT</code> | <code>MPI_COMM_SET_ATTR</code> |

Table 2.1: Deprecated constructs

2.6.2 Fortran Binding Issues

Originally, MPI-1.1 provided bindings for Fortran 77. These bindings are retained, but they are now interpreted in the context of the Fortran 90 standard. MPI can still be used with most Fortran 77 compilers, as noted below. When the term Fortran is used it means Fortran 90.

All MPI names have an `MPI_` prefix, and all characters are capitals. Programs must not declare variables, parameters, or functions with names beginning with the prefix `MPI_`. To avoid conflicting with the profiling interface, programs should also avoid functions with the prefix `PMPI_`. This is mandated to avoid possible name collisions.

All MPI Fortran subroutines have a return code in the last argument. A few MPI operations which are functions do not have the return code argument. The return code value

15.2 Deprecated since MPI-2.1

The entire set of C++ language bindings have been deprecated.

Rationale. The C++ bindings add minimal functionality over the C bindings while incurring a significant amount of maintenance to the MPI specification. Since the C++ bindings are effectively a one-to-one mapping of the C bindings, it should be relatively easy to convert existing C++ MPI applications to use the MPI C bindings. Additionally, there are third party packages available that provide C++ class library functionality (i.e., C++-specific functionality layered on top of the MPI C bindings) that are likely more expressive and/or natural to C++ programmers and are not suitable for standardization in this specification. (*End of rationale.*)

The following function typedefs have been deprecated and are superseded by new names. Other than the typedef names, the function signatures are exactly the same; the names were updated to match conventions of other function typedef names.

| Deprecated Name | New Name |
|---------------------------------------|---------------------------------------------|
| <code>MPI_Comm_errhandler_fn</code> | <code>MPI_Comm_errhandler_function</code> |
| <code>MPI::Comm::Errhandler_fn</code> | <code>MPI::Comm::Errhandler_function</code> |
| <code>MPI_File_errhandler_fn</code> | <code>MPI_File_errhandler_function</code> |
| <code>MPI::File::Errhandler_fn</code> | <code>MPI::File::Errhandler_function</code> |
| <code>MPI_Win_errhandler_fn</code> | <code>MPI_Win_errhandler_function</code> |
| <code>MPI::Win::Errhandler_fn</code> | <code>MPI::Win::Errhandler_function</code> |

```

MPI::Group 1
MPI::Info 2
MPI::Op 3
MPI::Request 4
MPI::Prequest 5
MPI::Grequest 6
MPI::Win 7
8
9

```

A.1.3 Prototype definitions

The following are defined C typedefs for user-defined functions, also included in the file `mpi.h`.

```

/* prototypes for user-defined functions */
typedef void MPI_User_function(void *invec, void *inoutvec, int *len,
                               MPI_Datatype *datatype);

typedef int MPI_Comm_copy_attr_function(MPI_Comm oldcomm,
                                         int comm_keyval, void *extra_state, void *attribute_val_in,
                                         void *attribute_val_out, int *flag);
typedef int MPI_Comm_delete_attr_function(MPI_Comm comm,
                                         int comm_keyval, void *attribute_val, void *extra_state);

typedef int MPI_Win_copy_attr_function(MPI_Win oldwin, int win_keyval,
                                         void *extra_state, void *attribute_val_in,
                                         void *attribute_val_out, int *flag);
typedef int MPI_Win_delete_attr_function(MPI_Win win, int win_keyval,
                                         void *attribute_val, void *extra_state);

typedef int MPI_Type_copy_attr_function(MPI_Datatype oldtype,
                                         int type_keyval, void *extra_state,
                                         void *attribute_val_in, void *attribute_val_out, int *flag);
typedef int MPI_Type_delete_attr_function(MPI_Datatype type,
                                         int type_keyval, void *attribute_val, void *extra_state);

typedef void MPI_Comm_errhandler_[ticket7.][fn]function(MPI_Comm *, int *, ...);
typedef void MPI_Win_errhandler_[ticket7.][fn]function(MPI_Win *, int *, ...);
typedef void MPI_File_errhandler_[ticket7.][fn]function(MPI_File *, int *, ...);

typedef int MPI_Grequest_query_function(void *extra_state,
                                         MPI_Status *status);
typedef int MPI_Grequest_free_function(void *extra_state);
typedef int MPI_Grequest_cancel_function(void *extra_state, int complete);

typedef int MPI_Datarep_extent_function(MPI_Datatype datatype,
                                         MPI_Aint *file_extent, void *extra_state);
typedef int MPI_Datarep_conversion_function(void *userbuf,

```

LOGICAL FLAG

```
SUBROUTINE TYPE_DELETE_ATTR_FN(TYPE, TYPE_KEYVAL, ATTRIBUTE_VAL,
                                EXTRA_STATE, IERROR)
  INTEGER TYPE, TYPE_KEYVAL, IERROR
  INTEGER(KIND=MPI_ADDRESS_KIND) ATTRIBUTE_VAL, EXTRA_STATE
```

The handler-function argument to MPI_COMM_CREATE_ERRHANDLER should be declared like this:

```
SUBROUTINE COMM_ERRHANDLER_[ticket7.] [FN]FUNCTION(COMM, ERROR_CODE[ticket1.] [, ..])
  INTEGER COMM, ERROR_CODE
```

The handler-function argument to MPI_WIN_CREATE_ERRHANDLER should be declared like this:

```
SUBROUTINE WIN_ERRHANDLER_[ticket7.] [FN]FUNCTION(WIN, ERROR_CODE[ticket1.] [, ..])
  INTEGER WIN, ERROR_CODE
```

The handler-function argument to MPI_FILE_CREATE_ERRHANDLER should be declared like this:

```
SUBROUTINE FILE_ERRHANDLER_[ticket7.] [FN]FUNCTION(FILE, ERROR_CODE[ticket1.] [, ..])
  INTEGER FILE, ERROR_CODE
```

The query, free, and cancel function arguments to MPI_GREQUEST_START should be declared like these:

```
SUBROUTINE GREQUEST_QUERY_FUNCTION(EXTRA_STATE, STATUS, IERROR)
  INTEGER STATUS(MPI_STATUS_SIZE), IERROR
  INTEGER(KIND=MPI_ADDRESS_KIND) EXTRA_STATE
```

```
SUBROUTINE GREQUEST_FREE_FUNCTION(EXTRA_STATE, IERROR)
  INTEGER IERROR
  INTEGER(KIND=MPI_ADDRESS_KIND) EXTRA_STATE
```

```
SUBROUTINE GREQUEST_CANCEL_FUNCTION(EXTRA_STATE, COMPLETE, IERROR)
  INTEGER IERROR
  INTEGER(KIND=MPI_ADDRESS_KIND) EXTRA_STATE
  LOGICAL COMPLETE
```

The extend and conversion function arguments to MPI_REGISTER_DATAREP should be declared like these:

```
SUBROUTINE DATAREP_EXTENT_FUNCTION(DATATYPE, EXTENT, EXTRA_STATE, IERROR)
  INTEGER DATATYPE, IERROR
  INTEGER(KIND=MPI_ADDRESS_KIND) EXTENT, EXTRA_STATE
```

```
SUBROUTINE DATAREP_CONVERSION_FUNCTION(USERBUF, DATATYPE, COUNT, FILEBUF,
```

```

1      POSITION, EXTRA_STATE, IERROR)
2      <TYPE> USERBUF(*), FILEBUF(*)
3      INTEGER COUNT, DATATYPE, IERROR
4      INTEGER(KIND=MPI_OFFSET_KIND) POSITION
5      INTEGER(KIND=MPI_ADDRESS_KIND) EXTRA_STATE

```

The following are defined C++ typedefs, also included in the file `mpi.h`.

```

8
9 namespace MPI {
10     typedef void User_function(const void* invec, void *inoutvec,
11                               int len, const Datatype& datatype);
12
13     typedef int Comm::Copy_attr_function(const Comm& oldcomm,
14     int comm_keyval, void* extra_state, void* attribute_val_in,
15     void* attribute_val_out, bool& flag);
16     typedef int Comm::Delete_attr_function(Comm& comm, int
17     comm_keyval, void* attribute_val, void* extra_state);
18
19     typedef int Win::Copy_attr_function(const Win& oldwin,
20     int win_keyval, void* extra_state, void* attribute_val_in,
21     void* attribute_val_out, bool& flag);
22     typedef int Win::Delete_attr_function(Win& win, int
23     win_keyval, void* attribute_val, void* extra_state);
24
25     typedef int Datatype::Copy_attr_function(const Datatype& oldtype,
26     int type_keyval, void* extra_state,
27     const void* attribute_val_in, void* attribute_val_out,
28     bool& flag);
29     typedef int Datatype::Delete_attr_function(Datatype& type,
30     int type_keyval, void* attribute_val, void* extra_state);
31
32     typedef void Comm::Errhandler_[ticket7.] [fn]function(Comm &, int *, ...);
33     typedef void Win::Errhandler_[ticket7.] [fn]function(Win &, int *, ...);
34     typedef void File::Errhandler_[ticket7.] [fn]function(File &, int *, ...);
35
36     typedef int Grequest::Query_function(void* extra_state, Status& status);
37     typedef int Grequest::Free_function(void* extra_state);
38     typedef int Grequest::Cancel_function(void* extra_state, bool complete);
39
40     typedef void Datarep_extent_function(const Datatype& datatype,
41     Aint& file_extent, void* extra_state);
42     typedef void Datarep_conversion_function(void* userbuf,
43     Datatype& datatype, int count, void* filebuf,
44     Offset position, void* extra_state);
45 }

```

MPI_DIST_GRAPH_CREATE, the constants MPI_UNWEIGHTED, and the derived C++ class Distgraphcomm were added.

18. Section 7.5.4 on page 264.

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.

19. Section 7.5.4 on page 264.

Remove ambiguity regarding duplicated neighbors with MPI_GRAPH_NEIGHBORS and MPI_GRAPH_NEIGHBORS_COUNT.

20. Section 8.1.1 on page 275.

The subversion number changed from 1 to 2.

21. Section 8.3 on page 280, Section 15.2 on page 468, and Annex A.1.3 on page 523.

Changed function pointer typedef names MPI_{Comm,File,Win}_errhandler_fn to MPI_{Comm,File,Win}_errhandler_function. Deprecated old “_fn” names.

22. Section 8.7.1 on page 299.

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.

23. Section 11.3.4 on page 349.

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.

24. Section 12.2 on page 375.

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.

25. Section 13.5.2 on page 433, and Table 13.2 on page 435.

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.

26. Section 16.3.7 on page 506.

The description was modified that it only describes how an MPI implementation behaves, but not how MPI stores attributes internally. The erroneous MPI-2.1 Example 16.17 was replaced with three new examples ??, ??, and ?? on pages ??-?? explicitly detailing cross-language attribute behavior. Implementations that matched the behavior of the old example will need to be updated.

27. Annex A.1.1 on page 511.

Removed type MPI::Fint (compare MPI_Fint in Section A.1.2 on page 522).

MPI Callback Function Prototype Index

This index lists the C typedef names for callback routines, such as those used with attribute caching or user-defined reduction operations. C++ names for these typedefs and Fortran example prototypes are given near the text of the C name.

MPI_Comm_copy_attr_function, [17](#), [239](#),
[518](#), [523](#)
MPI_Comm_delete_attr_function, [17](#), [239](#),
[518](#), [523](#)
MPI_Comm_errhandler_fn, [282](#), [468](#), [592](#)
MPI_Comm_errhandler_function, [17](#), [468](#),
[523](#), [592](#)
MPI_Copy_function, [17](#), [464](#), [518](#), [527](#)
MPI_Datarep_conversion_function, [436](#), [523](#)
MPI_Datarep_extent_function, [436](#), [523](#)
MPI_Delete_function, [17](#), [465](#), [518](#), [527](#)
MPI_File_errhandler_fn, [285](#), [468](#), [592](#)
MPI_File_errhandler_function, [468](#), [523](#),
[592](#)
MPI_Grequest_cancel_function, [378](#), [523](#)
MPI_Grequest_free_function, [377](#), [523](#)
MPI_Grequest_query_function, [376](#), [523](#)
MPI_Handler_function, [17](#), [466](#), [527](#)
MPI_Type_copy_attr_function, [246](#), [518](#),
[523](#)
MPI_Type_delete_attr_function, [246](#), [518](#),
[523](#)
MPI_User_function, [174](#), [523](#)
MPI_Win_copy_attr_function, [243](#), [518](#),
[523](#)
MPI_Win_delete_attr_function, [243](#), [518](#),
[523](#)
MPI_Win_errhandler_fn, [284](#), [468](#), [592](#)
MPI_Win_errhandler_function, [468](#), [523](#),
[592](#)