

Chapter 1

MPI Errata Ballot 3

1. MPI-2, page 179, lines 4-5 change

Thus, the names of MPI_COMM_WORLD, MPI_COMM_SELF, and MPI_COMM_PARENT will have the default of MPI_COMM_WORLD, MPI_COMM_SELF, and MPI_COMM_PARENT.

to

Thus, the names of MPI_COMM_WORLD, MPI_COMM_SELF, and the communicator returned by MPI_COMM_GET_PARENT (if not MPI_COMM_NULL) will have the default of MPI_COMM_WORLD, MPI_COMM_SELF, and MPI_COMM_PARENT.

MPI-2, page 94, line 3-5, change

* The manager is represented as the process with rank 0 in (the remote * group of) MPI_COMM_PARENT. If the workers need to communicate among * themselves, they can use MPI_COMM_WORLD.

to

* The manager is represented as the process with rank 0 in (the remote * group of) the parent communicator. If the workers need to communicate * among themselves, they can use MPI_COMM_WORLD.

2. MPI2, page 79, Line 11 is

MPI_UNPACK_EXTERNAL (datarep, inbuf, incount, datatype, outbuf, outsize, position)

but should be

MPI_UNPACK_EXTERNAL (datarep, inbuf, insize, position, outbuf, outcount, datatype)

3. MPI-2, page 337, line 31-32 reads

```
bool MPI::Win::Get_attr(const MPI::Win&win, int win_keyval,
                        void* attribute_val) const
```

1 but should read

2
3 `bool MPI::Win::Get_attr(int win_keyval, void* attribute_val) const`

4
5 4. On page 172, line 37 in section 8.2, change `MPI_REQUEST_CANCEL` to `MPI_CANCEL`.

6
7 5. MPI-2, page 163, line 22 reads

8
9 Within each group, all processes provide the same `recvcounts` argument,
10 and the sum of the `recvcounts` entries should be the same for the two
11 groups.

12
13 but should read

14
15 Within each group, all processes provide the same `recvcounts` argument,
16 and the sum of the `recvcounts` entries and `datatype` should specify the
17 same type signature for the two groups.

18 6. MPI-2, page 345, line 37: Remove the `const` from `const MPI::Op`.

19 MPI-2, page 346, line 20: Remove the `const` from `const MPI::Group`.

20
21 MPI-2, page 346, add after line 34:

22
23 *Advice to implementors.* If an implementation does not change the value of
24 predefined handles while execution of `MPI_Init`, the implementation is free to
25 define the predefined operation handles as `const MPI::Op` and the predefined
26 group handle `MPI::GROUP_EMPTY` as `const MPI::Group`. Other predefined
27 handles must not be "const" because they are allowed as `INOUT` argument in
28 the `MPI_COMM_SET_NAME/ATTR` and `MPI_TYPE_SET_NAME/ATTR` rou-
29 tines. (*End of advice to implementors.*)

30 7. MPI-1, page 128, line 11, in MPI-1.1 has an extraneous `root` argument. That line
31 should be

32
33 `MPI_Scan(a, answer, 1, sspair, myOp, comm);`

34
35 8. MPI-2, page 223, line 19. Change

36 `MPI_FILE_GET_VIEW(FH, DISP, ETYPE, FILETYPE, DATAREP, IERROR)`
37 `INTEGER FH, ETYPE, FILETYPE, IERROR`
38 `CHARACTER*(*) DATAREP, INTEGER(KIND=MPI_OFFSET_KIND) DISP`

39
40 to

41
42 `MPI_FILE_GET_VIEW(FH, DISP, ETYPE, FILETYPE, DATAREP, IERROR)`
43 `INTEGER FH, ETYPE, FILETYPE, IERROR`
44 `CHARACTER*(*) DATAREP`
45 `INTEGER(KIND=MPI_OFFSET_KIND) DISP`

46
47 in `io-2.tex`. (Replace the comma after the declaration of `datarep`)

9. MPI-2, page 66, line 26, change

```
MPI_TYPE_CREATE_HVECTOR(COUNT, BLOCKLENGTH, STIDE, OLDTYPE, NEWTYPE,
IERROR)
    INTEGER COUNT, BLOCKLENGTH, OLDTYPE, NEWTYPE, IERROR
    INTEGER(KIND=MPI_ADDRESS_KIND) STRIDE
```

to

```
MPI_TYPE_CREATE_HVECTOR(COUNT, BLOCKLENGTH, STRIDE, OLDTYPE, NEWTYPE,
IERROR)
    INTEGER COUNT, BLOCKLENGTH, OLDTYPE, NEWTYPE, IERROR
    INTEGER(KIND=MPI_ADDRESS_KIND) STRIDE
```

in misc-2.tex (Replace STIDE with STRIDE).

10. Examples in Chapter 3 of MPI 1.1 require several fixes.

MPI 1.1, Example 3.12, page 43, line 47 and page 44, lines 1, 5, 8, 10, and 13, the communicator argument `comm` must be added before the `req` argument.

The `ierr` argument must be added at the end of the argument list in the calls to `MPI_COMM_RANK` and `MPI_WAIT` in MPI 1.1, page 43, line 43, and page 44, lines 6 and 14.

The `ierr` argument must be added at the end of the argument list in the calls to `MPI_WAIT` in MPI 1.1, page 44, lines 35 and 36.

The lines in MPI 1.1, page 52, line 45, and page 53, line 17

```
IF (status(MPI_SOURCE) = 0) THEN
```

should be

```
IF (status(MPI_SOURCE) .EQ. 0) THEN
```

11. MPI 1.1, page 80, line 2, The variable `base` should be declared as `MPI_Aint`, not `int`, in Example 3.34.

12. Change MPI-2, page 343, lines 22-23

```
// Type: const void *
MPI::BOTTOM
```

to

```
// Type: void * const
MPI::BOTTOM
```

13. In MPI 1.1, page 16, line 23, use `strlen(message) + 1` instead of `strlen(message)` in the `MPI_Send` call.

14. A LaTeX line break is needed in MPI 1.1, page 58, line 44, in Section 3.9. The text should read

1 be invoked in a sequence of the form,

2 Create (Start Complete)* Free

3 where * indicates zero or more repetitions. If the same communication ...
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48