

## A.3 Fortran 2008 Bindings with the mpi\_f08 Module

#388

### A.3.1 Point-to-Point Communication Fortran 2008 Bindings

```

MPI_Bsend(buf, count, datatype, dest, tag, comm, ierror)
  TYPE(*), DIMENSION(..), INTENT(IN) :: buf
  INTEGER, INTENT(IN) :: count, dest, tag
  TYPE(MPI_Datatype), INTENT(IN) :: datatype
  TYPE(MPI_Comm), INTENT(IN) :: comm
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Bsend_init(buf, count, datatype, dest, tag, comm, request, ierror)
  TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
  INTEGER, INTENT(IN) :: count, dest, tag
  TYPE(MPI_Datatype), INTENT(IN) :: datatype
  TYPE(MPI_Comm), INTENT(IN) :: comm
  TYPE(MPI_Request), INTENT(OUT) :: request
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Buffer_attach(buffer, size, ierror)
  TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buffer
  INTEGER, INTENT(IN) :: size
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Buffer_detach(buffer_addr, size, ierror)
  USE, INTRINSIC :: ISO_C_BINDING, ONLY : C_PTR
  TYPE(C_PTR), INTENT(OUT) :: buffer_addr
  INTEGER, INTENT(OUT) :: size
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Cancel(request, ierror)
  TYPE(MPI_Request), INTENT(IN) :: request
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Get_count(status, datatype, count, ierror)
  TYPE(MPI_Status), INTENT(IN) :: status
  TYPE(MPI_Datatype), INTENT(IN) :: datatype
  INTEGER, INTENT(OUT) :: count
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Ibsend(buf, count, datatype, dest, tag, comm, request, ierror)
  TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
  INTEGER, INTENT(IN) :: count, dest, tag
  TYPE(MPI_Datatype), INTENT(IN) :: datatype
  TYPE(MPI_Comm), INTENT(IN) :: comm
  TYPE(MPI_Request), INTENT(OUT) :: request
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Improbe(source, tag, comm, flag, message, status, ierror)
  INTEGER, INTENT(IN) :: source, tag
  TYPE(MPI_Comm), INTENT(IN) :: comm

```

← BIND(C) is  
removed from all  
mpi\_f08 interfaces



```

LOGICAL, INTENT(OUT) :: flag
TYPE(MPI_Message), INTENT(OUT) :: message
TYPE(MPI_Status) :: status
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Imrecv(buf, count, datatype, message, request, ierror)
TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buf
INTEGER, INTENT(IN) :: count
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Message), INTENT(INOUT) :: message
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Iprobe(source, tag, comm, flag, status, ierror)
INTEGER, INTENT(IN) :: source, tag
TYPE(MPI_Comm), INTENT(IN) :: comm
LOGICAL, INTENT(OUT) :: flag
TYPE(MPI_Status) :: status
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Irecv(buf, count, datatype, source, tag, comm, request, ierror)
TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buf
INTEGER, INTENT(IN) :: count, source, tag
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Comm), INTENT(IN) :: comm
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Irsend(buf, count, datatype, dest, tag, comm, request, ierror)
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
INTEGER, INTENT(IN) :: count, dest, tag
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Comm), INTENT(IN) :: comm
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Isend(buf, count, datatype, dest, tag, comm, request, ierror)
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
INTEGER, INTENT(IN) :: count, dest, tag
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Comm), INTENT(IN) :: comm
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Issend(buf, count, datatype, dest, tag, comm, request, ierror)
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
INTEGER, INTENT(IN) :: count, dest, tag
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Comm), INTENT(IN) :: comm
TYPE(MPI_Request), INTENT(OUT) :: request

```

```

1      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
2
3      MPI_Mprobe(source, tag, comm, message, status, ierror)
4          INTEGER, INTENT(IN) :: source, tag
5          TYPE(MPI_Comm), INTENT(IN) :: comm
6          TYPE(MPI_Message), INTENT(OUT) :: message
7          TYPE(MPI_Status) :: status
8          INTEGER, OPTIONAL, INTENT(OUT) :: ierror
9
10     MPI_Mrecv(buf, count, datatype, message, status, ierror)
11         TYPE(*), DIMENSION(..) :: buf
12         INTEGER, INTENT(IN) :: count
13         TYPE(MPI_Datatype), INTENT(IN) :: datatype
14         TYPE(MPI_Message), INTENT(INOUT) :: message
15         TYPE(MPI_Status) :: status
16         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
17
18     MPI_Probe(source, tag, comm, status, ierror)
19         INTEGER, INTENT(IN) :: source, tag
20         TYPE(MPI_Comm), INTENT(IN) :: comm
21         TYPE(MPI_Status) :: status
22         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
23
24     MPI_Recv(buf, count, datatype, source, tag, comm, status, ierror)
25         TYPE(*), DIMENSION(..) :: buf
26         INTEGER, INTENT(IN) :: count, source, tag
27         TYPE(MPI_Datatype), INTENT(IN) :: datatype
28         TYPE(MPI_Comm), INTENT(IN) :: comm
29         TYPE(MPI_Status) :: status
30         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
31
32     MPI_Recv_init(buf, count, datatype, source, tag, comm, request, ierror)
33         TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buf
34         INTEGER, INTENT(IN) :: count, source, tag
35         TYPE(MPI_Datatype), INTENT(IN) :: datatype
36         TYPE(MPI_Comm), INTENT(IN) :: comm
37         TYPE(MPI_Request), INTENT(OUT) :: request
38         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
39
40     MPI_Request_free(request, ierror)
41         TYPE(MPI_Request), INTENT(INOUT) :: request
42         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
43
44     MPI_Request_get_status(request, flag, status, ierror)
45         TYPE(MPI_Request), INTENT(IN) :: request
46         LOGICAL, INTENT(OUT) :: flag
47         TYPE(MPI_Status) :: status
48         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
49
50     MPI_Rsend(buf, count, datatype, dest, tag, comm, ierror)
51         TYPE(*), DIMENSION(..), INTENT(IN) :: buf

```

```

    INTEGER, INTENT(IN) :: count, dest, tag
    TYPE(MPI_Datatype), INTENT(IN) :: datatype
    TYPE(MPI_Comm), INTENT(IN) :: comm
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Rsend_init(buf, count, datatype, dest, tag, comm, request, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
    INTEGER, INTENT(IN) :: count, dest, tag
    TYPE(MPI_Datatype), INTENT(IN) :: datatype
    TYPE(MPI_Comm), INTENT(IN) :: comm
    TYPE(MPI_Request), INTENT(OUT) :: request
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Send(buf, count, datatype, dest, tag, comm, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN) :: buf
    INTEGER, INTENT(IN) :: count, dest, tag
    TYPE(MPI_Datatype), INTENT(IN) :: datatype
    TYPE(MPI_Comm), INTENT(IN) :: comm
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Send_init(buf, count, datatype, dest, tag, comm, request, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
    INTEGER, INTENT(IN) :: count, dest, tag
    TYPE(MPI_Datatype), INTENT(IN) :: datatype
    TYPE(MPI_Comm), INTENT(IN) :: comm
    TYPE(MPI_Request), INTENT(OUT) :: request
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Sendrecv_replace(buf, count, datatype, dest, sendtag, source, recvtag,
    comm, status, ierror)
    TYPE(*), DIMENSION(..) :: buf
    INTEGER, INTENT(IN) :: count, dest, sendtag, source, recvtag
    TYPE(MPI_Datatype), INTENT(IN) :: datatype
    TYPE(MPI_Comm), INTENT(IN) :: comm
    TYPE(MPI_Status) :: status
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Sendrecv(sendbuf, sendcount, sendtype, dest, sendtag, recvbuf,
    recvcount, recvtype, source, recvtag, comm, status, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
    TYPE(*), DIMENSION(..) :: recvbuf
    INTEGER, INTENT(IN) :: sendcount, dest, sendtag, recvcount, source,
    recvtag
    TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
    TYPE(MPI_Comm), INTENT(IN) :: comm
    TYPE(MPI_Status) :: status
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Ssend(buf, count, datatype, dest, tag, comm, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN) :: buf

```

```

1      INTEGER, INTENT(IN) :: count, dest, tag
2      TYPE(MPI_Datatype), INTENT(IN) :: datatype
3      TYPE(MPI_Comm), INTENT(IN) :: comm
4      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
5
6      MPI_Ssend_init(buf, count, datatype, dest, tag, comm, request, ierror)
7      TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
8      INTEGER, INTENT(IN) :: count, dest, tag
9      TYPE(MPI_Datatype), INTENT(IN) :: datatype
10     TYPE(MPI_Comm), INTENT(IN) :: comm
11     TYPE(MPI_Request), INTENT(OUT) :: request
12     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
13
14     MPI_Startall(count, array_of_requests, ierror)
15     INTEGER, INTENT(IN) :: count
16     TYPE(MPI_Request), INTENT(INOUT) :: array_of_requests(count)
17     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
18
19     MPI_Start(request, ierror)
20     TYPE(MPI_Request), INTENT(INOUT) :: request
21     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
22
23     MPI_Testall(count, array_of_requests, flag, array_of_statuses, ierror)
24     INTEGER, INTENT(IN) :: count
25     TYPE(MPI_Request), INTENT(INOUT) :: array_of_requests(count)
26     LOGICAL, INTENT(OUT) :: flag
27     TYPE(MPI_Status) :: array_of_statuses(*)
28     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
29
30     MPI_Testany(count, array_of_requests, index, flag, status, ierror)
31     INTEGER, INTENT(IN) :: count
32     TYPE(MPI_Request), INTENT(INOUT) :: array_of_requests(count)
33     INTEGER, INTENT(OUT) :: index
34     LOGICAL, INTENT(OUT) :: flag
35     TYPE(MPI_Status) :: status
36     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
37
38     MPI_Test_cancelled(status, flag, ierror)
39     TYPE(MPI_Status), INTENT(IN) :: status
40     LOGICAL, INTENT(OUT) :: flag
41     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
42
43     MPI_Test(request, flag, status, ierror)
44     TYPE(MPI_Request), INTENT(INOUT) :: request
45     LOGICAL, INTENT(OUT) :: flag
46     TYPE(MPI_Status) :: status
47     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
48
49     MPI_Testsome(incount, array_of_requests, outcount, array_of_indices,
50                 array_of_statuses, ierror)
51     INTEGER, INTENT(IN) :: incount

```

```

TYPE(MPI_Request), INTENT(INOUT) :: array_of_requests(incount)
INTEGER, INTENT(OUT) :: outcount, array_of_indices(*)
TYPE(MPI_Status) :: array_of_statuses(*)
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Waitall(count, array_of_requests, array_of_statuses, ierror)
INTEGER, INTENT(IN) :: count
TYPE(MPI_Request), INTENT(INOUT) :: array_of_requests(count)
TYPE(MPI_Status) :: array_of_statuses(*)
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Waitany(count, array_of_requests, index, status, ierror)
INTEGER, INTENT(IN) :: count
TYPE(MPI_Request), INTENT(INOUT) :: array_of_requests(count)
INTEGER, INTENT(OUT) :: index
TYPE(MPI_Status) :: status
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Wait(request, status, ierror)
TYPE(MPI_Request), INTENT(INOUT) :: request
TYPE(MPI_Status) :: status
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Waitsome(incount, array_of_requests, outcount, array_of_indices,
             array_of_statuses, ierror)
INTEGER, INTENT(IN) :: incount
TYPE(MPI_Request), INTENT(INOUT) :: array_of_requests(incount)
INTEGER, INTENT(OUT) :: outcount, array_of_indices(*)
TYPE(MPI_Status) :: array_of_statuses(*)
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

### A.3.2 Datatypes Fortran 2008 Bindings

```

INTEGER(KIND=MPI_ADDRESS_KIND) MPI_Aint_add(base, disp)
INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: base, disp

INTEGER(KIND=MPI_ADDRESS_KIND) MPI_Aint_diff(addr1, addr2)
INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: addr1, addr2

MPI_Get_address(location, address, ierror)
TYPE(*), DIMENSION(..), ASYNCHRONOUS :: location
INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) :: address
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Get_elements(status, datatype, count, ierror)
TYPE(MPI_Status), INTENT(IN) :: status
TYPE(MPI_Datatype), INTENT(IN) :: datatype
INTEGER, INTENT(OUT) :: count
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Get_elements_x(status, datatype, count, ierror)

```

```

1      TYPE(MPI_Status), INTENT(IN) :: status
2      TYPE(MPI_Datatype), INTENT(IN) :: datatype
3      INTEGER(KIND = MPI_COUNT_KIND), INTENT(OUT) :: count
4      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
5
6      MPI_Pack_external(datarep, inbuf, incount, datatype, outbuf, outsize,
7          position, ierror)
8      CHARACTER(LEN=*), INTENT(IN) :: datarep
9      TYPE(*), DIMENSION(..), INTENT(IN) :: inbuf
10     TYPE(*), DIMENSION(..) :: outbuf
11     INTEGER, INTENT(IN) :: incount
12     TYPE(MPI_Datatype), INTENT(IN) :: datatype
13     INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: outsize
14     INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(INOUT) :: position
15     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
16
17     MPI_Pack_external_size(datarep, incount, datatype, size, ierror)
18     TYPE(MPI_Datatype), INTENT(IN) :: datatype
19     INTEGER, INTENT(IN) :: incount
20     CHARACTER(LEN=*), INTENT(IN) :: datarep
21     INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) :: size
22     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
23
24     MPI_Pack(inbuf, incount, datatype, outbuf, outsize, position, comm, ierror)
25     TYPE(*), DIMENSION(..), INTENT(IN) :: inbuf
26     TYPE(*), DIMENSION(..) :: outbuf
27     INTEGER, INTENT(IN) :: incount, outsize
28     TYPE(MPI_Datatype), INTENT(IN) :: datatype
29     INTEGER, INTENT(INOUT) :: position
30     TYPE(MPI_Comm), INTENT(IN) :: comm
31     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
32
33     MPI_Pack_size(incount, datatype, comm, size, ierror)
34     INTEGER, INTENT(IN) :: incount
35     TYPE(MPI_Datatype), INTENT(IN) :: datatype
36     TYPE(MPI_Comm), INTENT(IN) :: comm
37     INTEGER, INTENT(OUT) :: size
38     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
39
40     MPI_Type_commit(datatype, ierror)
41     TYPE(MPI_Datatype), INTENT(INOUT) :: datatype
42     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
43
44     MPI_Type_contiguous(count, oldtype, newtype, ierror)
45     INTEGER, INTENT(IN) :: count
46     TYPE(MPI_Datatype), INTENT(IN) :: oldtype
47     TYPE(MPI_Datatype), INTENT(OUT) :: newtype
48     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
49
50     MPI_Type_create_darray(size, rank, ndims, array_of_gsizes,
51         array_of_distribs, array_of_dargs, array_of_psize, order,

```

```

        oldtype, newtype, ierror)
    INTEGER, INTENT(IN) :: size, rank, ndims, array_of_gsizes(ndims),
    array_of_distribs(ndims), array_of_dargs(ndims),
    array_of_psize(ndims), order
    TYPE(MPI_Datatype), INTENT(IN) :: oldtype
    TYPE(MPI_Datatype), INTENT(OUT) :: newtype
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Type_create_hindexed_block(count, blocklength, array_of_displacements,
        oldtype, newtype, ierror)
    INTEGER, INTENT(IN) :: count, blocklength
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) ::
    array_of_displacements(count)
    TYPE(MPI_Datatype), INTENT(IN) :: oldtype
    TYPE(MPI_Datatype), INTENT(OUT) :: newtype
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Type_create_hindexed(count, array_of_blocklengths,
        array_of_displacements, oldtype, newtype, ierror)
    INTEGER, INTENT(IN) :: count, array_of_blocklengths(count)
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) ::
    array_of_displacements(count)
    TYPE(MPI_Datatype), INTENT(IN) :: oldtype
    TYPE(MPI_Datatype), INTENT(OUT) :: newtype
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Type_create_hvector(count, blocklength, stride, oldtype, newtype,
        ierror)
    INTEGER, INTENT(IN) :: count, blocklength
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: stride
    TYPE(MPI_Datatype), INTENT(IN) :: oldtype
    TYPE(MPI_Datatype), INTENT(OUT) :: newtype
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Type_create_indexed_block(count, blocklength, array_of_displacements,
        oldtype, newtype, ierror)
    INTEGER, INTENT(IN) :: count, blocklength,
    array_of_displacements(count)
    TYPE(MPI_Datatype), INTENT(IN) :: oldtype
    TYPE(MPI_Datatype), INTENT(OUT) :: newtype
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Type_create_resized(oldtype, lb, extent, newtype, ierror)
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: lb, extent
    TYPE(MPI_Datatype), INTENT(IN) :: oldtype
    TYPE(MPI_Datatype), INTENT(OUT) :: newtype
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Type_create_struct(count, array_of_blocklengths,
        array_of_displacements, array_of_types, newtype, ierror)

```



```

1      INTEGER, INTENT(IN) :: count, array_of_blocklengths(count)
2      INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) ::
3      array_of_displacements(count)
4      TYPE(MPI_Datatype), INTENT(IN) :: array_of_types(count)
5      TYPE(MPI_Datatype), INTENT(OUT) :: newtype
6      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
7
8      MPI_Type_create_subarray(ndims, array_of_sizes, array_of_subsizes,
9                              array_of_starts, order, oldtype, newtype, ierror)
10     INTEGER, INTENT(IN) :: ndims, array_of_sizes(ndims),
11     array_of_subsizes(ndims), array_of_starts(ndims), order
12     TYPE(MPI_Datatype), INTENT(IN) :: oldtype
13     TYPE(MPI_Datatype), INTENT(OUT) :: newtype
14     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
15
16     MPI_Type_dup(oldtype, newtype, ierror)
17     TYPE(MPI_Datatype), INTENT(IN) :: oldtype
18     TYPE(MPI_Datatype), INTENT(OUT) :: newtype
19     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
20
21     MPI_Type_free(datatype, ierror)
22     TYPE(MPI_Datatype), INTENT(INOUT) :: datatype
23     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
24
25     MPI_Type_get_contents(datatype, max_integers, max_addresses, max_datatypes,
26                           array_of_integers, array_of_addresses, array_of_datatypes,
27                           ierror)
28     TYPE(MPI_Datatype), INTENT(IN) :: datatype
29     INTEGER, INTENT(IN) :: max_integers, max_addresses, max_datatypes
30     INTEGER, INTENT(OUT) :: array_of_integers(max_integers)
31     INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) ::
32     array_of_addresses(max_addresses)
33     TYPE(MPI_Datatype), INTENT(OUT) :: array_of_datatypes(max_datatypes)
34     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
35
36     MPI_Type_get_envelope(datatype, num_integers, num_addresses, num_datatypes,
37                           combiner, ierror)
38     TYPE(MPI_Datatype), INTENT(IN) :: datatype
39     INTEGER, INTENT(OUT) :: num_integers, num_addresses, num_datatypes,
40     combiner
41     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
42
43     MPI_Type_get_extent(datatype, lb, extent, ierror)
44     TYPE(MPI_Datatype), INTENT(IN) :: datatype
45     INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) :: lb, extent
46     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
47
48     MPI_Type_get_extent_x(datatype, lb, extent, ierror)
49     TYPE(MPI_Datatype), INTENT(IN) :: datatype
50     INTEGER(KIND = MPI_COUNT_KIND), INTENT(OUT) :: lb, extent
51     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

MPI_Type_get_true_extent(datatype, true_lb, true_extent, ierror)      1
    TYPE(MPI_Datatype), INTENT(IN) :: datatype                        2
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) :: true_lb, true_extent 3
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                          4
                                                                    5
MPI_Type_get_true_extent_x(datatype, true_lb, true_extent, ierror)    6
    TYPE(MPI_Datatype), INTENT(IN) :: datatype                        7
    INTEGER(KIND = MPI_COUNT_KIND), INTENT(OUT) :: true_lb, true_extent 8
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                          9
                                                                    10
MPI_Type_indexed(count, array_of_blocklengths, array_of_displacements, 11
    oldtype, newtype, ierror)
    INTEGER, INTENT(IN) :: count, array_of_blocklengths(count),      12
    array_of_displacements(count)                                     13
    TYPE(MPI_Datatype), INTENT(IN) :: oldtype                         14
    TYPE(MPI_Datatype), INTENT(OUT) :: newtype                       15
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                         16
                                                                    17
MPI_Type_size(datatype, size, ierror)                                  18
    TYPE(MPI_Datatype), INTENT(IN) :: datatype                       19
    INTEGER, INTENT(OUT) :: size                                     20
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                         21
                                                                    22
MPI_Type_size_x(datatype, size, ierror)                                23
    TYPE(MPI_Datatype), INTENT(IN) :: datatype                       24
    INTEGER(KIND=MPI_COUNT_KIND), INTENT(OUT) :: size                25
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                         26
                                                                    27
MPI_Type_vector(count, blocklength, stride, oldtype, newtype, ierror) 28
    INTEGER, INTENT(IN) :: count, blocklength, stride                29
    TYPE(MPI_Datatype), INTENT(IN) :: oldtype                        30
    TYPE(MPI_Datatype), INTENT(OUT) :: newtype                       31
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                         32
                                                                    33
MPI_Unpack_external(datarep, inbuf, insize, position, outbuf, outcount, 34
    datatype, ierror)
    CHARACTER(LEN=*), INTENT(IN) :: datarep                           35
    TYPE(*), DIMENSION(..), INTENT(IN) :: inbuf                      36
    TYPE(*), DIMENSION(..) :: outbuf                                 37
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: insize              38
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(INOUT) :: position         39
    INTEGER, INTENT(IN) :: outcount                                  40
    TYPE(MPI_Datatype), INTENT(IN) :: datatype                       41
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                         42
                                                                    43
MPI_Unpack(inbuf, insize, position, outbuf, outcount, datatype, comm, 44
    ierror)
    TYPE(*), DIMENSION(..), INTENT(IN) :: inbuf                      45
    TYPE(*), DIMENSION(..) :: outbuf                                 46
    INTEGER, INTENT(IN) :: insize, outcount                          47
    INTEGER, INTENT(INOUT) :: position                                48

```

```

1      TYPE(MPI_Datatype), INTENT(IN) :: datatype
2      TYPE(MPI_Comm), INTENT(IN) :: comm
3      INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

### A.3.3 Collective Communication Fortran 2008 Bindings

```

7      MPI_Allgather(sendbuf, sendcount, sendtype, recvbuf, recvcount, recvtype,
8                  comm, ierror)

```

```

9      TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
10     TYPE(*), DIMENSION(..) :: recvbuf
11     INTEGER, INTENT(IN) :: sendcount, recvcount
12     TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
13     TYPE(MPI_Comm), INTENT(IN) :: comm
14     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

16     MPI_Allgatherv(sendbuf, sendcount, sendtype, recvbuf, recvcounts, displs,
17                  recvtype, comm, ierror)

```

```

18     TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
19     TYPE(*), DIMENSION(..) :: recvbuf
20     INTEGER, INTENT(IN) :: sendcount, recvcounts(*), displs(*)
21     TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
22     TYPE(MPI_Comm), INTENT(IN) :: comm
23     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

24     MPI_Allreduce(sendbuf, recvbuf, count, datatype, op, comm, ierror)

```

```

25     TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
26     TYPE(*), DIMENSION(..) :: recvbuf
27     INTEGER, INTENT(IN) :: count
28     TYPE(MPI_Datatype), INTENT(IN) :: datatype
29     TYPE(MPI_Op), INTENT(IN) :: op
30     TYPE(MPI_Comm), INTENT(IN) :: comm
31     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

33     MPI_Alltoall(sendbuf, sendcount, sendtype, recvbuf, recvcount, recvtype,
34                  comm, ierror)

```

```

35     TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
36     TYPE(*), DIMENSION(..) :: recvbuf
37     INTEGER, INTENT(IN) :: sendcount, recvcount
38     TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
39     TYPE(MPI_Comm), INTENT(IN) :: comm
40     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

41     MPI_Alltoallv(sendbuf, sendcounts, sdispls, sendtype, recvbuf, recvcounts,
42                  rdispls, recvtype, comm, ierror)

```

```

43     TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
44     TYPE(*), DIMENSION(..) :: recvbuf
45     INTEGER, INTENT(IN) :: sendcounts(*), sdispls(*), recvcounts(*),
46     rdispls(*)
47     TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype

```

```

TYPE(MPI_Comm), INTENT(IN) :: comm
INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Alltoallw(sendbuf, sendcounts, sdispls, sendtypes, recvbuf, recvcoun
    rdispls, recvtypes, comm, ierror)
TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
TYPE(*), DIMENSION(..) :: recvbuf
INTEGER, INTENT(IN) :: sendcounts(*), sdispls(*), recvcoun
    rdispls(*)
TYPE(MPI_Datatype), INTENT(IN) :: sendtypes(*)
TYPE(MPI_Datatype), INTENT(IN) :: recvtypes(*)
TYPE(MPI_Comm), INTENT(IN) :: comm
INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Barrier(comm, ierror)
TYPE(MPI_Comm), INTENT(IN) :: comm
INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Bcast(buffer, count, datatype, root, comm, ierror)
TYPE(*), DIMENSION(..) :: buffer
INTEGER, INTENT(IN) :: count, root
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Comm), INTENT(IN) :: comm
INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Exscan(sendbuf, recvbuf, count, datatype, op, comm, ierror)
TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
TYPE(*), DIMENSION(..) :: recvbuf
INTEGER, INTENT(IN) :: count
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Op), INTENT(IN) :: op
TYPE(MPI_Comm), INTENT(IN) :: comm
INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Gather(sendbuf, sendcount, sendtype, recvbuf, recvcoun
    root, comm, ierror)
TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
TYPE(*), DIMENSION(..) :: recvbuf
INTEGER, INTENT(IN) :: sendcount, recvcoun
    root
TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
TYPE(MPI_Comm), INTENT(IN) :: comm
INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Gatherv(sendbuf, sendcount, sendtype, recvbuf, recvcoun
    recvtype, root, comm, ierror)
TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
TYPE(*), DIMENSION(..) :: recvbuf
INTEGER, INTENT(IN) :: sendcount, recvcoun
    root
TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
TYPE(MPI_Comm), INTENT(IN) :: comm

```

```

1      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
2
3  MPI_Iallgather(sendbuf, sendcount, sendtype, recvbuf, recvcount, recvtype,
4      comm, request, ierror)
5      TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
6      TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
7      INTEGER, INTENT(IN) :: sendcount, recvcount
8      TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
9      TYPE(MPI_Comm), INTENT(IN) :: comm
10     TYPE(MPI_Request), INTENT(OUT) :: request
11     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
12
13 MPI_Iallgatherv(sendbuf, sendcount, sendtype, recvbuf, recvcounts, displs,
14     recvtype, comm, request, ierror)
15     TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
16     TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
17     INTEGER, INTENT(IN) :: sendcount
18     INTEGER, INTENT(IN), ASYNCHRONOUS :: recvcounts(*), displs(*)
19     TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
20     TYPE(MPI_Comm), INTENT(IN) :: comm
21     TYPE(MPI_Request), INTENT(OUT) :: request
22     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
23
24 MPI_Iallreduce(sendbuf, recvbuf, count, datatype, op, comm, request,
25     ierror)
26     TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
27     TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
28     INTEGER, INTENT(IN) :: count
29     TYPE(MPI_Datatype), INTENT(IN) :: datatype
30     TYPE(MPI_Op), INTENT(IN) :: op
31     TYPE(MPI_Comm), INTENT(IN) :: comm
32     TYPE(MPI_Request), INTENT(OUT) :: request
33     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
34
35 MPI_Ialltoall(sendbuf, sendcount, sendtype, recvbuf, recvcount, recvtype,
36     comm, request, ierror)
37     TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
38     TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
39     INTEGER, INTENT(IN) :: sendcount, recvcount
40     TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
41     TYPE(MPI_Comm), INTENT(IN) :: comm
42     TYPE(MPI_Request), INTENT(OUT) :: request
43     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
44
45 MPI_Ialltoallv(sendbuf, sendcounts, sdispls, sendtype, recvbuf, recvcounts,
46     rdispls, recvtype, comm, request, ierror)
47     TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
48     TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
49     INTEGER, INTENT(IN), ASYNCHRONOUS :: sendcounts(*), sdispls(*),
50     recvcounts(*), rdispls(*)

```



```

1 MPI_Igatherv(sendbuf, sendcount, sendtype, recvbuf, recvcounts, displs,
2             recvtype, root, comm, request, ierror)
3     TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
4     TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
5     INTEGER, INTENT(IN) :: sendcount, root
6     INTEGER, INTENT(IN), ASYNCHRONOUS :: recvcounts(*), displs(*)
7     TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
8     TYPE(MPI_Comm), INTENT(IN) :: comm
9     TYPE(MPI_Request), INTENT(OUT) :: request
10    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
11
12 MPI_Ireduce_scatter_block(sendbuf, recvbuf, recvcount, datatype, op, comm,
13                           request, ierror)
14    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
15    TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
16    INTEGER, INTENT(IN) :: recvcount
17    TYPE(MPI_Datatype), INTENT(IN) :: datatype
18    TYPE(MPI_Op), INTENT(IN) :: op
19    TYPE(MPI_Comm), INTENT(IN) :: comm
20    TYPE(MPI_Request), INTENT(OUT) :: request
21    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
22
23 MPI_Ireduce_scatter(sendbuf, recvbuf, recvcounts, datatype, op, comm,
24                     request, ierror)
25    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
26    TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
27    INTEGER, INTENT(IN), ASYNCHRONOUS :: recvcounts(*)
28    TYPE(MPI_Datatype), INTENT(IN) :: datatype
29    TYPE(MPI_Op), INTENT(IN) :: op
30    TYPE(MPI_Comm), INTENT(IN) :: comm
31    TYPE(MPI_Request), INTENT(OUT) :: request
32    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
33
34 MPI_Ireduce(sendbuf, recvbuf, count, datatype, op, root, comm, request,
35             ierror)
36    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
37    TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
38    INTEGER, INTENT(IN) :: count, root
39    TYPE(MPI_Datatype), INTENT(IN) :: datatype
40    TYPE(MPI_Op), INTENT(IN) :: op
41    TYPE(MPI_Comm), INTENT(IN) :: comm
42    TYPE(MPI_Request), INTENT(OUT) :: request
43    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
44
45 MPI_Iscan(sendbuf, recvbuf, count, datatype, op, comm, request, ierror)
46    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
47    TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
48    INTEGER, INTENT(IN) :: count
49    TYPE(MPI_Datatype), INTENT(IN) :: datatype

```

```

TYPE(MPI_Op), INTENT(IN) :: op
TYPE(MPI_Comm), INTENT(IN) :: comm
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Isscatter(sendbuf, sendcount, sendtype, recvbuf, recvcount, recvtype,
              root, comm, request, ierror)
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
INTEGER, INTENT(IN) :: sendcount, recvcount, root
TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
TYPE(MPI_Comm), INTENT(IN) :: comm
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Isscatterv(sendbuf, sendcounts, displs, sendtype, recvbuf, recvcount,
               recvtype, root, comm, request, ierror)
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
INTEGER, INTENT(IN), ASYNCHRONOUS :: sendcounts(*), displs(*)
INTEGER, INTENT(IN) :: recvcount, root
TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
TYPE(MPI_Comm), INTENT(IN) :: comm
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Op_commutative(op, commute, ierror)
TYPE(MPI_Op), INTENT(IN) :: op
LOGICAL, INTENT(OUT) :: commute
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Op_create(user_fn, commute, op, ierror)
PROCEDURE(MPI_User_function) :: user_fn
LOGICAL, INTENT(IN) :: commute
TYPE(MPI_Op), INTENT(OUT) :: op
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Op_free(op, ierror)
TYPE(MPI_Op), INTENT(INOUT) :: op
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Reduce_local(inbuf, inoutbuf, count, datatype, op, ierror)
TYPE(*), DIMENSION(..), INTENT(IN) :: inbuf
TYPE(*), DIMENSION(..) :: inoutbuf
INTEGER, INTENT(IN) :: count
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Op), INTENT(IN) :: op
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Reduce_scatter_block(sendbuf, recvbuf, recvcount, datatype, op, comm,
                         ierror)

```



```

1      TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
2      TYPE(*), DIMENSION(..) :: recvbuf
3      INTEGER, INTENT(IN) :: recvcoun
4      TYPE(MPI_Datatype), INTENT(IN) :: datatype
5      TYPE(MPI_Op), INTENT(IN) :: op
6      TYPE(MPI_Comm), INTENT(IN) :: comm
7      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
8
9      MPI_Reduce_scatter(sendbuf, recvbuf, recvcoun, datatype, op, comm,
10         ierror)
11      TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
12      TYPE(*), DIMENSION(..) :: recvbuf
13      INTEGER, INTENT(IN) :: recvcoun(*)
14      TYPE(MPI_Datatype), INTENT(IN) :: datatype
15      TYPE(MPI_Op), INTENT(IN) :: op
16      TYPE(MPI_Comm), INTENT(IN) :: comm
17      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
18
19      MPI_Reduce(sendbuf, recvbuf, count, datatype, op, root, comm, ierror)
20      TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
21      TYPE(*), DIMENSION(..) :: recvbuf
22      INTEGER, INTENT(IN) :: count, root
23      TYPE(MPI_Datatype), INTENT(IN) :: datatype
24      TYPE(MPI_Op), INTENT(IN) :: op
25      TYPE(MPI_Comm), INTENT(IN) :: comm
26      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
27
28      MPI_Scan(sendbuf, recvbuf, count, datatype, op, comm, ierror)
29      TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
30      TYPE(*), DIMENSION(..) :: recvbuf
31      INTEGER, INTENT(IN) :: count
32      TYPE(MPI_Datatype), INTENT(IN) :: datatype
33      TYPE(MPI_Op), INTENT(IN) :: op
34      TYPE(MPI_Comm), INTENT(IN) :: comm
35      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
36
37      MPI_Scatter(sendbuf, sendcount, sendtype, recvbuf, recvcoun, recvtype,
38         root, comm, ierror)
39      TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
40      TYPE(*), DIMENSION(..) :: recvbuf
41      INTEGER, INTENT(IN) :: sendcount, recvcoun, root
42      TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
43      TYPE(MPI_Comm), INTENT(IN) :: comm
44      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
45
46      MPI_Scatterv(sendbuf, sendcounts, displs, sendtype, recvbuf, recvcoun,
47         recvtype, root, comm, ierror)
48      TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
49      TYPE(*), DIMENSION(..) :: recvbuf
50      INTEGER, INTENT(IN) :: sendcounts(*), displs(*), recvcoun, root

```

```

TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
TYPE(MPI_Comm), INTENT(IN) :: comm
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

#### A.3.4 Groups, Contexts, Communicators, and Caching Fortran 2008 Bindings

```

MPI_Comm_compare(comm1, comm2, result, ierror)
  TYPE(MPI_Comm), INTENT(IN) :: comm1, comm2
  INTEGER, INTENT(OUT) :: result
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Comm_create(comm, group, newcomm, ierror)
  TYPE(MPI_Comm), INTENT(IN) :: comm
  TYPE(MPI_Group), INTENT(IN) :: group
  TYPE(MPI_Comm), INTENT(OUT) :: newcomm
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Comm_create_group(comm, group, tag, newcomm, ierror)
  TYPE(MPI_Comm), INTENT(IN) :: comm
  TYPE(MPI_Group), INTENT(IN) :: group
  INTEGER, INTENT(IN) :: tag
  TYPE(MPI_Comm), INTENT(OUT) :: newcomm
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Comm_create_keyval(comm_copy_attr_fn, comm_delete_attr_fn, comm_keyval,
  extra_state, ierror)
  PROCEDURE(MPI_Comm_copy_attr_function) :: comm_copy_attr_fn
  PROCEDURE(MPI_Comm_delete_attr_function) :: comm_delete_attr_fn
  INTEGER, INTENT(OUT) :: comm_keyval
  INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: extra_state
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Comm_delete_attr(comm, comm_keyval, ierror)
  TYPE(MPI_Comm), INTENT(IN) :: comm
  INTEGER, INTENT(IN) :: comm_keyval
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Comm_dup(comm, newcomm, ierror)
  TYPE(MPI_Comm), INTENT(IN) :: comm
  TYPE(MPI_Comm), INTENT(OUT) :: newcomm
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_COMM_DUP_FN(oldcomm, comm_keyval, extra_state, attribute_val_in,
  attribute_val_out, flag, ierror)
  TYPE(MPI_Comm) :: oldcomm
  INTEGER :: comm_keyval
  INTEGER(KIND=MPI_ADDRESS_KIND) :: extra_state, attribute_val_in
  INTEGER(KIND=MPI_ADDRESS_KIND) :: attribute_val_out
  LOGICAL :: flag
  INTEGER :: ierror

```

```

1  MPI_Comm_dup_with_info(comm, info, newcomm, ierror)
2      TYPE(MPI_Comm), INTENT(IN) :: comm
3      TYPE(MPI_Info), INTENT(IN) :: info
4      TYPE(MPI_Comm), INTENT(OUT) :: newcomm
5      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
6
7  MPI_Comm_free(comm, ierror)
8      TYPE(MPI_Comm), INTENT(INOUT) :: comm
9      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
10
11 MPI_Comm_free_keyval(comm_keyval, ierror)
12     INTEGER, INTENT(INOUT) :: comm_keyval
13     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
14
15 MPI_Comm_get_attr(comm, comm_keyval, attribute_val, flag, ierror)
16     TYPE(MPI_Comm), INTENT(IN) :: comm
17     INTEGER, INTENT(IN) :: comm_keyval
18     INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) :: attribute_val
19     LOGICAL, INTENT(OUT) :: flag
20     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
21
22 MPI_Comm_get_info(comm, info_used, ierror)
23     TYPE(MPI_Comm), INTENT(IN) :: comm
24     TYPE(MPI_Info), INTENT(OUT) :: info_used
25     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
26
27 MPI_Comm_get_name(comm, comm_name, resultlen, ierror)
28     TYPE(MPI_Comm), INTENT(IN) :: comm
29     CHARACTER(LEN=MPI_MAX_OBJECT_NAME), INTENT(OUT) :: comm_name
30     INTEGER, INTENT(OUT) :: resultlen
31     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
32
33 MPI_Comm_group(comm, group, ierror)
34     TYPE(MPI_Comm), INTENT(IN) :: comm
35     TYPE(MPI_Group), INTENT(OUT) :: group
36     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
37
38 MPI_Comm_idup(comm, newcomm, request, ierror)
39     TYPE(MPI_Comm), INTENT(IN) :: comm
40     TYPE(MPI_Comm), INTENT(OUT), ASYNCHRONOUS :: newcomm
41     TYPE(MPI_Request), INTENT(OUT) :: request
42     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
43
44 MPI_COMM_NULL_COPY_FN(oldcomm, comm_keyval, extra_state, attribute_val_in,
45     attribute_val_out, flag, ierror)
46     TYPE(MPI_Comm) :: oldcomm
47     INTEGER :: comm_keyval
48     INTEGER(KIND=MPI_ADDRESS_KIND) :: extra_state, attribute_val_in
49     INTEGER(KIND=MPI_ADDRESS_KIND) :: attribute_val_out
50     LOGICAL :: flag
51     INTEGER :: ierror

```

```

MPI_COMM_NULL_DELETE_FN(comm, comm_keyval, attribute_val, extra_state,      1
                          ierror)                                           2
    TYPE(MPI_Comm) :: comm                                                  3
    INTEGER :: comm_keyval                                                  4
    INTEGER(KIND=MPI_ADDRESS_KIND) :: attribute_val, extra_state           5
    INTEGER :: ierror                                                       6
                                                                              7
MPI_Comm_rank(comm, rank, ierror)                                           8
    TYPE(MPI_Comm), INTENT(IN) :: comm                                     9
    INTEGER, INTENT(OUT) :: rank                                           10
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                               11
                                                                              12
MPI_Comm_remote_group(comm, group, ierror)                                  13
    TYPE(MPI_Comm), INTENT(IN) :: comm                                     14
    TYPE(MPI_Group), INTENT(OUT) :: group                                  15
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                               16
                                                                              17
MPI_Comm_remote_size(comm, size, ierror)                                    18
    TYPE(MPI_Comm), INTENT(IN) :: comm                                     19
    INTEGER, INTENT(OUT) :: size                                           20
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                               21
                                                                              22
MPI_Comm_set_attr(comm, comm_keyval, attribute_val, ierror)                23
    TYPE(MPI_Comm), INTENT(IN) :: comm                                     24
    INTEGER, INTENT(IN) :: comm_keyval                                     25
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: attribute_val           26
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                               27
                                                                              28
MPI_Comm_set_info(comm, info, ierror)                                       29
    TYPE(MPI_Comm), INTENT(IN) :: comm                                     30
    TYPE(MPI_Info), INTENT(IN) :: info                                     31
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                               32
                                                                              33
MPI_Comm_set_name(comm, comm_name, ierror)                                  34
    TYPE(MPI_Comm), INTENT(IN) :: comm                                     35
    CHARACTER(LEN=*), INTENT(IN) :: comm_name                             36
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                               37
                                                                              38
MPI_Comm_size(comm, size, ierror)                                           39
    TYPE(MPI_Comm), INTENT(IN) :: comm                                     40
    INTEGER, INTENT(OUT) :: size                                           41
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                               42
                                                                              43
MPI_Comm_split(comm, color, key, newcomm, ierror)                          44
    TYPE(MPI_Comm), INTENT(IN) :: comm                                     45
    INTEGER, INTENT(IN) :: color, key                                       46
    TYPE(MPI_Comm), INTENT(OUT) :: newcomm                                  47
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                               48
                                                                              49
MPI_Comm_split_type(comm, split_type, key, info, newcomm, ierror)          50
    TYPE(MPI_Comm), INTENT(IN) :: comm                                     51
    INTEGER, INTENT(IN) :: split_type, key                                  52

```

```

1      TYPE(MPI_Info), INTENT(IN) :: info
2      TYPE(MPI_Comm), INTENT(OUT) :: newcomm
3      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
4
5      MPI_Comm_test_inter(comm, flag, ierror)
6      TYPE(MPI_Comm), INTENT(IN) :: comm
7      LOGICAL, INTENT(OUT) :: flag
8      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
9
10     MPI_Group_compare(group1, group2, result, ierror)
11     TYPE(MPI_Group), INTENT(IN) :: group1, group2
12     INTEGER, INTENT(OUT) :: result
13     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
14
15     MPI_Group_difference(group1, group2, newgroup, ierror)
16     TYPE(MPI_Group), INTENT(IN) :: group1, group2
17     TYPE(MPI_Group), INTENT(OUT) :: newgroup
18     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
19
20     MPI_Group_excl(group, n, ranks, newgroup, ierror)
21     TYPE(MPI_Group), INTENT(IN) :: group
22     INTEGER, INTENT(IN) :: n, ranks(n)
23     TYPE(MPI_Group), INTENT(OUT) :: newgroup
24     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
25
26     MPI_Group_free(group, ierror)
27     TYPE(MPI_Group), INTENT(INOUT) :: group
28     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
29
30     MPI_Group_incl(group, n, ranks, newgroup, ierror)
31     TYPE(MPI_Group), INTENT(IN) :: group
32     INTEGER, INTENT(IN) :: n, ranks(n)
33     TYPE(MPI_Group), INTENT(OUT) :: newgroup
34     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
35
36     MPI_Group_intersection(group1, group2, newgroup, ierror)
37     TYPE(MPI_Group), INTENT(IN) :: group1, group2
38     TYPE(MPI_Group), INTENT(OUT) :: newgroup
39     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
40
41     MPI_Group_range_excl(group, n, ranges, newgroup, ierror)
42     TYPE(MPI_Group), INTENT(IN) :: group
43     INTEGER, INTENT(IN) :: n, ranges(3,n)
44     TYPE(MPI_Group), INTENT(OUT) :: newgroup
45     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
46
47     MPI_Group_range_incl(group, n, ranges, newgroup, ierror)
48     TYPE(MPI_Group), INTENT(IN) :: group
49     INTEGER, INTENT(IN) :: n, ranges(3,n)
50     TYPE(MPI_Group), INTENT(OUT) :: newgroup
51     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

MPI_Group_rank(group, rank, ierror)                                1
    TYPE(MPI_Group), INTENT(IN) :: group                          2
    INTEGER, INTENT(OUT) :: rank                                  3
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                      4
                                                                    5
MPI_Group_size(group, size, ierror)                                6
    TYPE(MPI_Group), INTENT(IN) :: group                          7
    INTEGER, INTENT(OUT) :: size                                  8
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                      9
                                                                    10
MPI_Group_translate_ranks(group1, n, ranks1, group2, ranks2, ierror) 11
    TYPE(MPI_Group), INTENT(IN) :: group1, group2                12
    INTEGER, INTENT(IN) :: n, ranks1(n)                          13
    INTEGER, INTENT(OUT) :: ranks2(n)                            14
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                      15
                                                                    16
MPI_Group_union(group1, group2, newgroup, ierror)                 17
    TYPE(MPI_Group), INTENT(IN) :: group1, group2                18
    TYPE(MPI_Group), INTENT(OUT) :: newgroup                     19
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                      20
                                                                    21
MPI_Intercomm_create(local_comm, local_leader, peer_comm, remote_leader, 22
    tag, newintercomm, ierror)                                    23
    TYPE(MPI_Comm), INTENT(IN) :: local_comm, peer_comm          24
    INTEGER, INTENT(IN) :: local_leader, remote_leader, tag      25
    TYPE(MPI_Comm), INTENT(OUT) :: newintercomm                  26
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                      27
                                                                    28
MPI_Intercomm_merge(intercomm, high, newintracomm, ierror)        29
    TYPE(MPI_Comm), INTENT(IN) :: intercomm                       30
    LOGICAL, INTENT(IN) :: high                                   31
    TYPE(MPI_Comm), INTENT(OUT) :: newintracomm                  32
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                      33
                                                                    34
MPI_Type_create_keyval(type_copy_attr_fn, type_delete_attr_fn, type_keyval, 35
    extra_state, ierror)                                          36
    PROCEDURE(MPI_Type_copy_attr_function) :: type_copy_attr_fn 37
    PROCEDURE(MPI_Type_delete_attr_function) :: type_delete_attr_fn 38
    INTEGER, INTENT(OUT) :: type_keyval                           39
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: extra_state    40
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                      41
                                                                    42
MPI_Type_delete_attr(datatype, type_keyval, ierror)               43
    TYPE(MPI_Datatype), INTENT(IN) :: datatype                   44
    INTEGER, INTENT(IN) :: type_keyval                            45
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                      46
                                                                    47
MPI_Type_DUP_FN(oldtype, type_keyval, extra_state, attribute_val_in, 48
    attribute_val_out, flag, ierror)
    TYPE(MPI_Datatype) :: oldtype
    INTEGER :: type_keyval

```

```

1      INTEGER(KIND=MPI_ADDRESS_KIND) :: extra_state, attribute_val_in
2      INTEGER(KIND=MPI_ADDRESS_KIND) :: attribute_val_out
3      LOGICAL :: flag
4      INTEGER :: ierror
5
6      MPI_Type_free_keyval(type_keyval, ierror)
7          INTEGER, INTENT(INOUT) :: type_keyval
8          INTEGER, OPTIONAL, INTENT(OUT) :: ierror
9
10     MPI_Type_get_attr(datatype, type_keyval, attribute_val, flag, ierror)
11         TYPE(MPI_Datatype), INTENT(IN) :: datatype
12         INTEGER, INTENT(IN) :: type_keyval
13         INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) :: attribute_val
14         LOGICAL, INTENT(OUT) :: flag
15         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
16
17     MPI_Type_get_name(datatype, type_name, resultlen, ierror)
18         TYPE(MPI_Datatype), INTENT(IN) :: datatype
19         CHARACTER(LEN=MPI_MAX_OBJECT_NAME), INTENT(OUT) :: type_name
20         INTEGER, INTENT(OUT) :: resultlen
21         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
22
23     MPI_TYPE_NULL_COPY_FN(oldtype, type_keyval, extra_state, attribute_val_in,
24         attribute_val_out, flag, ierror)
25         TYPE(MPI_Datatype) :: oldtype
26         INTEGER :: type_keyval
27         INTEGER(KIND=MPI_ADDRESS_KIND) :: extra_state, attribute_val_in
28         INTEGER(KIND=MPI_ADDRESS_KIND) :: attribute_val_out
29         LOGICAL :: flag
30         INTEGER :: ierror
31
32     MPI_TYPE_NULL_DELETE_FN(datatype, type_keyval, attribute_val, extra_state,
33         ierror)
34         TYPE(MPI_Datatype) :: datatype
35         INTEGER :: type_keyval
36         INTEGER(KIND=MPI_ADDRESS_KIND) :: attribute_val, extra_state
37         INTEGER, INTENT(OUT) :: ierror
38
39     MPI_Type_set_attr(datatype, type_keyval, attribute_val, ierror)
40         TYPE(MPI_Datatype), INTENT(IN) :: datatype
41         INTEGER, INTENT(IN) :: type_keyval
42         INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: attribute_val
43         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
44
45     MPI_Type_set_name(datatype, type_name, ierror)
46         TYPE(MPI_Datatype), INTENT(IN) :: datatype
47         CHARACTER(LEN=*), INTENT(IN) :: type_name
48         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
49
50     MPI_Win_create_keyval(win_copy_attr_fn, win_delete_attr_fn, win_keyval,
51         extra_state, ierror)

```

```

PROCEDURE(MPI_Win_copy_attr_function) :: win_copy_attr_fn      1
PROCEDURE(MPI_Win_delete_attr_function) :: win_delete_attr_fn  2
INTEGER, INTENT(OUT) :: win_keyval                             3
INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: extra_state      4
INTEGER, OPTIONAL, INTENT(OUT) :: ierror                       5
                                                                6
MPI_Win_delete_attr(win, win_keyval, ierror)                   7
  TYPE(MPI_Win), INTENT(IN) :: win                             8
  INTEGER, INTENT(IN) :: win_keyval                             9
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror                     10
                                                                11
MPI_WIN_DUP_FN(oldwin, win_keyval, extra_state, attribute_val_in,
               attribute_val_out, flag, ierror)                 12
  TYPE(MPI_Win) :: oldwin                                       13
  INTEGER :: win_keyval                                          14
  INTEGER(KIND=MPI_ADDRESS_KIND) :: extra_state, attribute_val_in 15
  INTEGER(KIND=MPI_ADDRESS_KIND) :: attribute_val_out           16
  LOGICAL :: flag                                               17
  INTEGER :: ierror                                              18
                                                                19
MPI_Win_free_keyval(win_keyval, ierror)                        20
  INTEGER, INTENT(INOUT) :: win_keyval                          21
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror                      22
                                                                23
MPI_Win_get_attr(win, win_keyval, attribute_val, flag, ierror) 24
  TYPE(MPI_Win), INTENT(IN) :: win                             25
  INTEGER, INTENT(IN) :: win_keyval                             26
  INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) :: attribute_val 27
  LOGICAL, INTENT(OUT) :: flag                                  28
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror                      29
                                                                30
MPI_Win_get_name(win, win_name, resultlen, ierror)             31
  TYPE(MPI_Win), INTENT(IN) :: win                             32
  CHARACTER(LEN=MPI_MAX_OBJECT_NAME), INTENT(OUT) :: win_name 33
  INTEGER, INTENT(OUT) :: resultlen                             34
  INTEGER, OPTIONAL, INTENT(OUT) :: ierror                      35
                                                                36
MPI_WIN_NULL_COPY_FN(oldwin, win_keyval, extra_state, attribute_val_in,
                    attribute_val_out, flag, ierror)            37
  TYPE(MPI_Win) :: oldwin                                       38
  INTEGER :: win_keyval                                          39
  INTEGER(KIND=MPI_ADDRESS_KIND) :: extra_state, attribute_val_in 40
  INTEGER(KIND=MPI_ADDRESS_KIND) :: attribute_val_out           41
  LOGICAL :: flag                                               42
  INTEGER :: ierror                                              43
                                                                44
MPI_WIN_NULL_DELETE_FN(win, win_keyval, attribute_val, extra_state, ierror)
  TYPE(MPI_Win) :: win                                           45
  INTEGER :: win_keyval                                           46
  INTEGER(KIND=MPI_ADDRESS_KIND) :: attribute_val, extra_state 47
  INTEGER :: ierror                                              48

```



```

1  MPI_Win_set_attr(win, win_keyval, attribute_val, ierror)
2      TYPE(MPI_Win), INTENT(IN) :: win
3      INTEGER, INTENT(IN) :: win_keyval
4      INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: attribute_val
5      INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

6  MPI_Win_set_name(win, win_name, ierror)
7      TYPE(MPI_Win), INTENT(IN) :: win
8      CHARACTER(LEN=*), INTENT(IN) :: win_name
9      INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

### A.3.5 Process Topologies Fortran 2008 Bindings

```

14 MPI_Cart_coords(comm, rank, maxdims, coords, ierror)
15     TYPE(MPI_Comm), INTENT(IN) :: comm
16     INTEGER, INTENT(IN) :: rank, maxdims
17     INTEGER, INTENT(OUT) :: coords(maxdims)
18     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

19 MPI_Cart_create(comm_old, ndims, dims, periods, reorder, comm_cart, ierror)
20     TYPE(MPI_Comm), INTENT(IN) :: comm_old
21     INTEGER, INTENT(IN) :: ndims, dims(ndims)
22     LOGICAL, INTENT(IN) :: periods(ndims), reorder
23     TYPE(MPI_Comm), INTENT(OUT) :: comm_cart
24     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

26 MPI_Cartdim_get(comm, ndims, ierror)
27     TYPE(MPI_Comm), INTENT(IN) :: comm
28     INTEGER, INTENT(OUT) :: ndims
29     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

30 MPI_Cart_get(comm, maxdims, dims, periods, coords, ierror)
31     TYPE(MPI_Comm), INTENT(IN) :: comm
32     INTEGER, INTENT(IN) :: maxdims
33     INTEGER, INTENT(OUT) :: dims(maxdims), coords(maxdims)
34     LOGICAL, INTENT(OUT) :: periods(maxdims)
35     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

37 MPI_Cart_map(comm, ndims, dims, periods, newrank, ierror)
38     TYPE(MPI_Comm), INTENT(IN) :: comm
39     INTEGER, INTENT(IN) :: ndims, dims(ndims)
40     LOGICAL, INTENT(IN) :: periods(ndims)
41     INTEGER, INTENT(OUT) :: newrank
42     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

43 MPI_Cart_rank(comm, coords, rank, ierror)
44     TYPE(MPI_Comm), INTENT(IN) :: comm
45     INTEGER, INTENT(IN) :: coords(*)
46     INTEGER, INTENT(OUT) :: rank
47     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

MPI_Cart_shift(comm, direction, disp, rank_source, rank_dest, ierror) 1
    TYPE(MPI_Comm), INTENT(IN) :: comm 2
    INTEGER, INTENT(IN) :: direction, disp 3
    INTEGER, INTENT(OUT) :: rank_source, rank_dest 4
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror 5
6
MPI_Cart_sub(comm, remain_dims, newcomm, ierror) 7
    TYPE(MPI_Comm), INTENT(IN) :: comm 8
    LOGICAL, INTENT(IN) :: remain_dims(*) 9
    TYPE(MPI_Comm), INTENT(OUT) :: newcomm 10
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror 11
12
MPI_Dims_create(nnodes, ndims, dims, ierror) 13
    INTEGER, INTENT(IN) :: nnodes, ndims 14
    INTEGER, INTENT(INOUT) :: dims(ndims) 15
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror 16
17
MPI_Dist_graph_create_adjacent(comm_old, indegree, sources, sourceweights,
    outdegree, destinations, destweights, info, reorder,
    comm_dist_graph, ierror) 18
    TYPE(MPI_Comm), INTENT(IN) :: comm_old 19
    INTEGER, INTENT(IN) :: indegree, sources(indegree), outdegree,
    destinations(outdegree) 20
    INTEGER, INTENT(IN) :: sourceweights(*), destweights(*) 21
    TYPE(MPI_Info), INTENT(IN) :: info 22
    LOGICAL, INTENT(IN) :: reorder 23
    TYPE(MPI_Comm), INTENT(OUT) :: comm_dist_graph 24
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror 25
26
MPI_Dist_graph_create(comm_old, n, sources, degrees, destinations, weights,
    info, reorder, comm_dist_graph, ierror) 27
    TYPE(MPI_Comm), INTENT(IN) :: comm_old 28
    INTEGER, INTENT(IN) :: n, sources(n), degrees(n), destinations(*) 29
    INTEGER, INTENT(IN) :: weights(*) 30
    TYPE(MPI_Info), INTENT(IN) :: info 31
    LOGICAL, INTENT(IN) :: reorder 32
    TYPE(MPI_Comm), INTENT(OUT) :: comm_dist_graph 33
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror 34
35
MPI_Dist_graph_neighbors(comm, maxindegree, sources, sourceweights,
    maxoutdegree, destinations, destweights, ierror) 36
    TYPE(MPI_Comm), INTENT(IN) :: comm 37
    INTEGER, INTENT(IN) :: maxindegree, maxoutdegree 38
    INTEGER, INTENT(OUT) :: sources(maxindegree),
    destinations(maxoutdegree) 39
    INTEGER :: sourceweights(*), destweights(*) 40
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror 41
42
MPI_Dist_graph_neighbors_count(comm, indegree, outdegree, weighted, ierror) 43
    TYPE(MPI_Comm), INTENT(IN) :: comm 44
45
46
47
48

```

```

1      INTEGER, INTENT(OUT) :: indegree, outdegree
2      LOGICAL, INTENT(OUT) :: weighted
3      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
4
5      MPI_Graph_create(comm_old, nnodes, index, edges, reorder, comm_graph,
6                      ierror)
7      TYPE(MPI_Comm), INTENT(IN) :: comm_old
8      INTEGER, INTENT(IN) :: nnodes, index(nnodes), edges(*)
9      LOGICAL, INTENT(IN) :: reorder
10     TYPE(MPI_Comm), INTENT(OUT) :: comm_graph
11     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
12
13     MPI_Graphdims_get(comm, nnodes, nedges, ierror)
14     TYPE(MPI_Comm), INTENT(IN) :: comm
15     INTEGER, INTENT(OUT) :: nnodes, nedges
16     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
17
18     MPI_Graph_get(comm, maxindex, maxedges, index, edges, ierror)
19     TYPE(MPI_Comm), INTENT(IN) :: comm
20     INTEGER, INTENT(IN) :: maxindex, maxedges
21     INTEGER, INTENT(OUT) :: index(maxindex), edges(maxedges)
22     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
23
24     MPI_Graph_map(comm, nnodes, index, edges, newrank, ierror)
25     TYPE(MPI_Comm), INTENT(IN) :: comm
26     INTEGER, INTENT(IN) :: nnodes, index(nnodes), edges(*)
27     INTEGER, INTENT(OUT) :: newrank
28     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
29
30     MPI_Graph_neighbors(comm, rank, maxneighbors, neighbors, ierror)
31     TYPE(MPI_Comm), INTENT(IN) :: comm
32     INTEGER, INTENT(IN) :: rank, maxneighbors
33     INTEGER, INTENT(OUT) :: neighbors(maxneighbors)
34     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
35
36     MPI_Graph_neighbors_count(comm, rank, nneighbors, ierror)
37     TYPE(MPI_Comm), INTENT(IN) :: comm
38     INTEGER, INTENT(IN) :: rank
39     INTEGER, INTENT(OUT) :: nneighbors
40     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
41
42     MPI_Ineighbor_allgather(sendbuf, sendcount, sendtype, recvbuf, recvcount,
43                             recvtype, comm, request, ierror)
44     TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
45     TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
46     INTEGER, INTENT(IN) :: sendcount, recvcount
47     TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
48     TYPE(MPI_Comm), INTENT(IN) :: comm
49     TYPE(MPI_Request), INTENT(OUT) :: request
50     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

MPI_Ineighbor_allgatherv(sendbuf, sendcount, sendtype, recvbuf, recvcounts,
    displs, recvtype, comm, request, ierror)
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
INTEGER, INTENT(IN) :: sendcount
INTEGER, INTENT(IN), ASYNCHRONOUS :: recvcounts(*), displs(*)
TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
TYPE(MPI_Comm), INTENT(IN) :: comm
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Ineighbor_alltoall(sendbuf, sendcount, sendtype, recvbuf, recvcount,
    recvtype, comm, request, ierror)
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
INTEGER, INTENT(IN) :: sendcount, recvcount
TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
TYPE(MPI_Comm), INTENT(IN) :: comm
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Ineighbor_alltoallv(sendbuf, sendcounts, sdispls, sendtype, recvbuf,
    recvcounts, rdispls, recvtype, comm, request, ierror)
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
INTEGER, INTENT(IN), ASYNCHRONOUS :: sendcounts(*), sdispls(*),
    recvcounts(*), rdispls(*)
TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvtype
TYPE(MPI_Comm), INTENT(IN) :: comm
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Ineighbor_alltoallw(sendbuf, sendcounts, sdispls, sendtypes, recvbuf,
    recvcounts, rdispls, recvtypes, comm, request, ierror)
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: sendbuf
TYPE(*), DIMENSION(..), ASYNCHRONOUS :: recvbuf
INTEGER, INTENT(IN), ASYNCHRONOUS :: sendcounts(*), recvcounts(*)
INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN), ASYNCHRONOUS ::
    sdispls(*), rdispls(*)
TYPE(MPI_Datatype), INTENT(IN), ASYNCHRONOUS :: sendtypes(*),
    recvtypes(*)
TYPE(MPI_Comm), INTENT(IN) :: comm
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Neighbor_allgather(sendbuf, sendcount, sendtype, recvbuf, recvcount,
    recvtype, comm, ierror)
TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
TYPE(*), DIMENSION(..) :: recvbuf

```

```

1      INTEGER, INTENT(IN) :: sendcount, recvcount
2      TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvttype
3      TYPE(MPI_Comm), INTENT(IN) :: comm
4      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
5
6  MPI_Neighbor_allgatherv(sendbuf, sendcount, sendtype, recvbuf, recvcounts,
7      displs, recvttype, comm, ierror)
8      TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
9      TYPE(*), DIMENSION(..) :: recvbuf
10     INTEGER, INTENT(IN) :: sendcount, recvcounts(*), displs(*)
11     TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvttype
12     TYPE(MPI_Comm), INTENT(IN) :: comm
13     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
14
15 MPI_Neighbor_alltoall(sendbuf, sendcount, sendtype, recvbuf, recvcount,
16     recvttype, comm, ierror)
17     TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
18     TYPE(*), DIMENSION(..) :: recvbuf
19     INTEGER, INTENT(IN) :: sendcount, recvcount
20     TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvttype
21     TYPE(MPI_Comm), INTENT(IN) :: comm
22     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
23
24 MPI_Neighbor_alltoallv(sendbuf, sendcounts, sdispls, sendtype, recvbuf,
25     recvcounts, rdispls, recvttype, comm, ierror)
26     TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
27     TYPE(*), DIMENSION(..) :: recvbuf
28     INTEGER, INTENT(IN) :: sendcounts(*), sdispls(*), recvcounts(*),
29     rdispls(*)
30     TYPE(MPI_Datatype), INTENT(IN) :: sendtype, recvttype
31     TYPE(MPI_Comm), INTENT(IN) :: comm
32     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
33
34 MPI_Neighbor_alltoallw(sendbuf, sendcounts, sdispls, sendtypes, recvbuf,
35     recvcounts, rdispls, recvtypes, comm, ierror)
36     TYPE(*), DIMENSION(..), INTENT(IN) :: sendbuf
37     TYPE(*), DIMENSION(..) :: recvbuf
38     INTEGER, INTENT(IN) :: sendcounts(*), recvcounts(*)
39     INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: sdispls(*), rdispls(*)
40     TYPE(MPI_Datatype), INTENT(IN) :: sendtypes(*), recvtypes(*)
41     TYPE(MPI_Comm), INTENT(IN) :: comm
42     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
43
44 MPI_Topo_test(comm, status, ierror)
45     TYPE(MPI_Comm), INTENT(IN) :: comm
46     INTEGER, INTENT(OUT) :: status
47     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
48

```

## A.3.6 MPI Environmental Management Fortran 2008 Bindings

```

DOUBLE PRECISION MPI_Wtick()
DOUBLE PRECISION MPI_Wtime()
MPI_Abort(comm, errorcode, ierror)
    TYPE(MPI_Comm), INTENT(IN) :: comm
    INTEGER, INTENT(IN) :: errorcode
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Add_error_class(errorclass, ierror)
    INTEGER, INTENT(OUT) :: errorclass
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Add_error_code(errorclass, errorcode, ierror)
    INTEGER, INTENT(IN) :: errorclass
    INTEGER, INTENT(OUT) :: errorcode
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Add_error_string(errorcode, string, ierror)
    INTEGER, INTENT(IN) :: errorcode
    CHARACTER(LEN=*), INTENT(IN) :: string
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Alloc_mem(size, info, baseptr, ierror)
    USE, INTRINSIC :: ISO_C_BINDING, ONLY : C_PTR
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: size
    TYPE(MPI_Info), INTENT(IN) :: info
    TYPE(C_PTR), INTENT(OUT) :: baseptr
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Comm_call_errhandler(comm, errorcode, ierror)
    TYPE(MPI_Comm), INTENT(IN) :: comm
    INTEGER, INTENT(IN) :: errorcode
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Comm_create_errhandler(comm_errhandler_fn, errhandler, ierror)
    PROCEDURE(MPI_Comm_errhandler_function) :: comm_errhandler_fn
    TYPE(MPI_Errhandler), INTENT(OUT) :: errhandler
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Comm_get_errhandler(comm, errhandler, ierror)
    TYPE(MPI_Comm), INTENT(IN) :: comm
    TYPE(MPI_Errhandler), INTENT(OUT) :: errhandler
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Comm_set_errhandler(comm, errhandler, ierror)
    TYPE(MPI_Comm), INTENT(IN) :: comm
    TYPE(MPI_Errhandler), INTENT(IN) :: errhandler
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Errhandler_free(errhandler, ierror)

```

```

1      TYPE(MPI_Errhandler), INTENT(INOUT) :: errhandler
2      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
3
4      MPI_Error_class(errorcode, errorclass, ierror)
5          INTEGER, INTENT(IN) :: errorcode
6          INTEGER, INTENT(OUT) :: errorclass
7          INTEGER, OPTIONAL, INTENT(OUT) :: ierror
8
9      MPI_Error_string(errorcode, string, resultlen, ierror)
10         INTEGER, INTENT(IN) :: errorcode
11         CHARACTER(LEN=MPI_MAX_ERROR_STRING), INTENT(OUT) :: string
12         INTEGER, INTENT(OUT) :: resultlen
13         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
14
15      MPI_File_call_errhandler(fh, errorcode, ierror)
16         TYPE(MPI_File), INTENT(IN) :: fh
17         INTEGER, INTENT(IN) :: errorcode
18         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
19
20      MPI_File_create_errhandler(file_errhandler_fn, errhandler, ierror)
21         PROCEDURE(MPI_File_errhandler_function) :: file_errhandler_fn
22         TYPE(MPI_Errhandler), INTENT(OUT) :: errhandler
23         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
24
25      MPI_File_get_errhandler(file, errhandler, ierror)
26         TYPE(MPI_File), INTENT(IN) :: file
27         TYPE(MPI_Errhandler), INTENT(OUT) :: errhandler
28         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
29
30      MPI_File_set_errhandler(file, errhandler, ierror)
31         TYPE(MPI_File), INTENT(IN) :: file
32         TYPE(MPI_Errhandler), INTENT(IN) :: errhandler
33         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
34
35      MPI_Finalized(flag, ierror)
36         LOGICAL, INTENT(OUT) :: flag
37         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
38
39      MPI_Finalize(ierror)
40         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
41
42      MPI_Free_mem(base, ierror)
43         TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: base
44         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
45
46      MPI_Get_library_version(version, resultlen, ierror)
47         CHARACTER(LEN=MPI_MAX_LIBRARY_VERSION_STRING), INTENT(OUT) :: version
48         INTEGER, INTENT(OUT) :: resultlen
49         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
50
51      MPI_Get_processor_name(name, resultlen, ierror)
52         CHARACTER(LEN=MPI_MAX_PROCESSOR_NAME), INTENT(OUT) :: name
53         INTEGER, INTENT(OUT) :: resultlen

```

```

    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Get_version(version, subversion, ierror)
    INTEGER, INTENT(OUT) :: version, subversion
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Initialized(flag, ierror)
    LOGICAL, INTENT(OUT) :: flag
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Init(ierror)
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Win_call_errhandler(win, errorcode, ierror)
    TYPE(MPI_Win), INTENT(IN) :: win
    INTEGER, INTENT(IN) :: errorcode
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Win_create_errhandler(win_errhandler_fn, errhandler, ierror)
    PROCEDURE(MPI_Win_errhandler_function) :: win_errhandler_fn
    TYPE(MPI_Errhandler), INTENT(OUT) :: errhandler
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Win_get_errhandler(win, errhandler, ierror)
    TYPE(MPI_Win), INTENT(IN) :: win
    TYPE(MPI_Errhandler), INTENT(OUT) :: errhandler
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Win_set_errhandler(win, errhandler, ierror)
    TYPE(MPI_Win), INTENT(IN) :: win
    TYPE(MPI_Errhandler), INTENT(IN) :: errhandler
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

### A.3.7 The Info Object Fortran 2008 Bindings

```

MPI_Info_create(info, ierror)
    TYPE(MPI_Info), INTENT(OUT) :: info
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Info_delete(info, key, ierror)
    TYPE(MPI_Info), INTENT(IN) :: info
    CHARACTER(LEN=*), INTENT(IN) :: key
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Info_dup(info, newinfo, ierror)
    TYPE(MPI_Info), INTENT(IN) :: info
    TYPE(MPI_Info), INTENT(OUT) :: newinfo
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
MPI_Info_free(info, ierror)
    TYPE(MPI_Info), INTENT(INOUT) :: info
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```



```

1  MPI_Info_get(info, key, valuelen, value, flag, ierror)
2      TYPE(MPI_Info), INTENT(IN) :: info
3      CHARACTER(LEN=*), INTENT(IN) :: key
4      INTEGER, INTENT(IN) :: valuelen
5      CHARACTER(LEN=valuelen), INTENT(OUT) :: value
6      LOGICAL, INTENT(OUT) :: flag
7      INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

8  MPI_Info_get_nkeys(info, nkeys, ierror)
9      TYPE(MPI_Info), INTENT(IN) :: info
10     INTEGER, INTENT(OUT) :: nkeys
11     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

12 MPI_Info_get_nthkey(info, n, key, ierror)
13     TYPE(MPI_Info), INTENT(IN) :: info
14     INTEGER, INTENT(IN) :: n
15     CHARACTER(LEN=*), INTENT(OUT) :: key
16     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

17 MPI_Info_get_valuelen(info, key, valuelen, flag, ierror)
18     TYPE(MPI_Info), INTENT(IN) :: info
19     CHARACTER(LEN=*), INTENT(IN) :: key
20     INTEGER, INTENT(OUT) :: valuelen
21     LOGICAL, INTENT(OUT) :: flag
22     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

23 MPI_Info_set(info, key, value, ierror)
24     TYPE(MPI_Info), INTENT(IN) :: info
25     CHARACTER(LEN=*), INTENT(IN) :: key, value
26     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

### A.3.8 Process Creation and Management Fortran 2008 Bindings

```

31 MPI_Close_port(port_name, ierror)
32     CHARACTER(LEN=*), INTENT(IN) :: port_name
33     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

34 MPI_Comm_accept(port_name, info, root, comm, newcomm, ierror)
35     CHARACTER(LEN=*), INTENT(IN) :: port_name
36     TYPE(MPI_Info), INTENT(IN) :: info
37     INTEGER, INTENT(IN) :: root
38     TYPE(MPI_Comm), INTENT(IN) :: comm
39     TYPE(MPI_Comm), INTENT(OUT) :: newcomm
40     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

41 MPI_Comm_connect(port_name, info, root, comm, newcomm, ierror)
42     CHARACTER(LEN=*), INTENT(IN) :: port_name
43     TYPE(MPI_Info), INTENT(IN) :: info
44     INTEGER, INTENT(IN) :: root
45     TYPE(MPI_Comm), INTENT(IN) :: comm

```

```

TYPE(MPI_Comm), INTENT(OUT) :: newcomm
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Comm_disconnect(comm, ierror)

TYPE(MPI_Comm), INTENT(INOUT) :: comm
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Comm_get_parent(parent, ierror)

TYPE(MPI_Comm), INTENT(OUT) :: parent
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Comm_join(fd, intercomm, ierror)

INTEGER, INTENT(IN) :: fd
TYPE(MPI_Comm), INTENT(OUT) :: intercomm
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Comm_spawn(command, argv, maxprocs, info, root, comm, intercomm,
               array_of_errcodes, ierror)

CHARACTER(LEN=*), INTENT(IN) :: command, argv(*)
INTEGER, INTENT(IN) :: maxprocs, root
TYPE(MPI_Info), INTENT(IN) :: info
TYPE(MPI_Comm), INTENT(IN) :: comm
TYPE(MPI_Comm), INTENT(OUT) :: intercomm
INTEGER :: array_of_errcodes(*)
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Comm_spawn_multiple(count, array_of_commands, array_of_argv,
                       array_of_maxprocs, array_of_info, root, comm, intercomm,
                       array_of_errcodes, ierror)

INTEGER, INTENT(IN) :: count, array_of_maxprocs(*), root
CHARACTER(LEN=*), INTENT(IN) :: array_of_commands(*)
CHARACTER(LEN=*), INTENT(IN) :: array_of_argv(count, *)
TYPE(MPI_Info), INTENT(IN) :: array_of_info(*)
TYPE(MPI_Comm), INTENT(IN) :: comm
TYPE(MPI_Comm), INTENT(OUT) :: intercomm
INTEGER :: array_of_errcodes(*)
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Lookup_name(service_name, info, port_name, ierror)

CHARACTER(LEN=*), INTENT(IN) :: service_name
TYPE(MPI_Info), INTENT(IN) :: info
CHARACTER(LEN=MPI_MAX_PORT_NAME), INTENT(OUT) :: port_name
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Open_port(info, port_name, ierror)

TYPE(MPI_Info), INTENT(IN) :: info
CHARACTER(LEN=MPI_MAX_PORT_NAME), INTENT(OUT) :: port_name
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Publish_name(service_name, info, port_name, ierror)

TYPE(MPI_Info), INTENT(IN) :: info

```

```

1      CHARACTER(LEN=*), INTENT(IN) :: service_name, port_name
2      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
3
4      MPI_Unpublish_name(service_name, info, port_name, ierror)
5      CHARACTER(LEN=*), INTENT(IN) :: service_name, port_name
6      TYPE(MPI_Info), INTENT(IN) :: info
7      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
8

```

### A.3.9 One-Sided Communications Fortran 2008 Bindings

```

10
11      MPI_Accumulate(origin_addr, origin_count, origin_datatype, target_rank,
12                    target_disp, target_count, target_datatype, op, win, ierror)
13      TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: origin_addr
14      INTEGER, INTENT(IN) :: origin_count, target_rank, target_count
15      TYPE(MPI_Datatype), INTENT(IN) :: origin_datatype, target_datatype
16      INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: target_disp
17      TYPE(MPI_Op), INTENT(IN) :: op
18      TYPE(MPI_Win), INTENT(IN) :: win
19      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
20
21      MPI_Compare_and_swap(origin_addr, compare_addr, result_addr, datatype,
22                          target_rank, target_disp, win, ierror)
23      TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: origin_addr
24      TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: compare_addr
25      TYPE(*), DIMENSION(..), ASYNCHRONOUS :: result_addr
26      TYPE(MPI_Datatype), INTENT(IN) :: datatype
27      INTEGER, INTENT(IN) :: target_rank
28      INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: target_disp
29      TYPE(MPI_Win), INTENT(IN) :: win
30      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
31
32      MPI_Fetch_and_op(origin_addr, result_addr, datatype, target_rank,
33                      target_disp, op, win, ierror)
34      TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: origin_addr
35      TYPE(*), DIMENSION(..), ASYNCHRONOUS :: result_addr
36      TYPE(MPI_Datatype), INTENT(IN) :: datatype
37      INTEGER, INTENT(IN) :: target_rank
38      INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: target_disp
39      TYPE(MPI_Op), INTENT(IN) :: op
40      TYPE(MPI_Win), INTENT(IN) :: win
41      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
42
43      MPI_Get_accumulate(origin_addr, origin_count, origin_datatype, result_addr,
44                        result_count, result_datatype, target_rank, target_disp,
45                        target_count, target_datatype, op, win, ierror)
46      TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: origin_addr
47      TYPE(*), DIMENSION(..), ASYNCHRONOUS :: result_addr
48      INTEGER, INTENT(IN) :: origin_count, result_count, target_rank,
49      target_count

```

```

TYPE(MPI_Datatype), INTENT(IN) :: origin_datatype, target_datatype,
result_datatype
INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: target_disp
TYPE(MPI_Op), INTENT(IN) :: op
TYPE(MPI_Win), INTENT(IN) :: win
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Get(origin_addr, origin_count, origin_datatype, target_rank,
        target_disp, target_count, target_datatype, win, ierror)
TYPE(*), DIMENSION(..), ASYNCHRONOUS :: origin_addr
INTEGER, INTENT(IN) :: origin_count, target_rank, target_count
TYPE(MPI_Datatype), INTENT(IN) :: origin_datatype, target_datatype
INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: target_disp
TYPE(MPI_Win), INTENT(IN) :: win
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Put(origin_addr, origin_count, origin_datatype, target_rank,
        target_disp, target_count, target_datatype, win, ierror)
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: origin_addr
INTEGER, INTENT(IN) :: origin_count, target_rank, target_count
TYPE(MPI_Datatype), INTENT(IN) :: origin_datatype, target_datatype
INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: target_disp
TYPE(MPI_Win), INTENT(IN) :: win
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Raccumulate(origin_addr, origin_count, origin_datatype, target_rank,
        target_disp, target_count, target_datatype, op, win, request,
        ierror)
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: origin_addr
INTEGER, INTENT(IN) :: origin_count, target_rank, target_count
TYPE(MPI_Datatype), INTENT(IN) :: origin_datatype, target_datatype
INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: target_disp
TYPE(MPI_Op), INTENT(IN) :: op
TYPE(MPI_Win), INTENT(IN) :: win
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Rget_accumulate(origin_addr, origin_count, origin_datatype,
        result_addr, result_count, result_datatype, target_rank,
        target_disp, target_count, target_datatype, op, win, request,
        ierror)
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: origin_addr
TYPE(*), DIMENSION(..), ASYNCHRONOUS :: result_addr
INTEGER, INTENT(IN) :: origin_count, result_count, target_rank,
target_count
TYPE(MPI_Datatype), INTENT(IN) :: origin_datatype, target_datatype,
result_datatype
INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: target_disp
TYPE(MPI_Op), INTENT(IN) :: op

```

```

1      TYPE(MPI_Win), INTENT(IN) :: win
2      TYPE(MPI_Request), INTENT(OUT) :: request
3      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
4
5      MPI_Rget(origin_addr, origin_count, origin_datatype, target_rank,
6              target_disp, target_count, target_datatype, win, request,
7              ierror)
8      TYPE(*), DIMENSION(..), ASYNCHRONOUS :: origin_addr
9      INTEGER, INTENT(IN) :: origin_count, target_rank, target_count
10     TYPE(MPI_Datatype), INTENT(IN) :: origin_datatype, target_datatype
11     INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: target_disp
12     TYPE(MPI_Win), INTENT(IN) :: win
13     TYPE(MPI_Request), INTENT(OUT) :: request
14     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
15
16     MPI_Rput(origin_addr, origin_count, origin_datatype, target_rank,
17             target_disp, target_count, target_datatype, win, request,
18             ierror)
19     TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: origin_addr
20     INTEGER, INTENT(IN) :: origin_count, target_rank, target_count
21     TYPE(MPI_Datatype), INTENT(IN) :: origin_datatype, target_datatype
22     INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: target_disp
23     TYPE(MPI_Win), INTENT(IN) :: win
24     TYPE(MPI_Request), INTENT(OUT) :: request
25     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
26
27     MPI_Win_allocate_shared(size, disp_unit, info, comm, baseptr, win, ierror)
28     USE, INTRINSIC :: ISO_C_BINDING, ONLY : C_PTR
29     INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: size
30     INTEGER, INTENT(IN) :: disp_unit
31     TYPE(MPI_Info), INTENT(IN) :: info
32     TYPE(MPI_Comm), INTENT(IN) :: comm
33     TYPE(C_PTR), INTENT(OUT) :: baseptr
34     TYPE(MPI_Win), INTENT(OUT) :: win
35     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
36
37     MPI_Win_allocate(size, disp_unit, info, comm, baseptr, win, ierror)
38     USE, INTRINSIC :: ISO_C_BINDING, ONLY : C_PTR
39     INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: size
40     INTEGER, INTENT(IN) :: disp_unit
41     TYPE(MPI_Info), INTENT(IN) :: info
42     TYPE(MPI_Comm), INTENT(IN) :: comm
43     TYPE(C_PTR), INTENT(OUT) :: baseptr
44     TYPE(MPI_Win), INTENT(OUT) :: win
45     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
46
47     MPI_Win_attach(win, base, size, ierror)
48     TYPE(MPI_Win), INTENT(IN) :: win
49     TYPE(*), DIMENSION(..), ASYNCHRONOUS :: base
50     INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: size

```

```

    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
1
MPI_Win_complete(win, ierror)
2
    TYPE(MPI_Win), INTENT(IN) :: win
3
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
4
5
MPI_Win_create(base, size, disp_unit, info, comm, win, ierror)
6
    TYPE(*), DIMENSION(..), ASYNCHRONOUS :: base
7
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: size
8
    INTEGER, INTENT(IN) :: disp_unit
9
    TYPE(MPI_Info), INTENT(IN) :: info
10
    TYPE(MPI_Comm), INTENT(IN) :: comm
11
    TYPE(MPI_Win), INTENT(OUT) :: win
12
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
13
14
MPI_Win_create_dynamic(info, comm, win, ierror)
15
    TYPE(MPI_Info), INTENT(IN) :: info
16
    TYPE(MPI_Comm), INTENT(IN) :: comm
17
    TYPE(MPI_Win), INTENT(OUT) :: win
18
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
19
20
MPI_Win_detach(win, base, ierror)
21
    TYPE(MPI_Win), INTENT(IN) :: win
22
    TYPE(*), DIMENSION(..), ASYNCHRONOUS :: base
23
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
24
25
MPI_Win_fence(assert, win, ierror)
26
    INTEGER, INTENT(IN) :: assert
27
    TYPE(MPI_Win), INTENT(IN) :: win
28
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
29
30
MPI_Win_flush_all(win, ierror)
31
    TYPE(MPI_Win), INTENT(IN) :: win
32
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
33
34
MPI_Win_flush_local_all(win, ierror)
35
    TYPE(MPI_Win), INTENT(IN) :: win
36
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
37
38
MPI_Win_flush_local(rank, win, ierror)
39
    INTEGER, INTENT(IN) :: rank
40
    TYPE(MPI_Win), INTENT(IN) :: win
41
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
42
43
MPI_Win_flush(rank, win, ierror)
44
    INTEGER, INTENT(IN) :: rank
45
    TYPE(MPI_Win), INTENT(IN) :: win
46
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
47
48
MPI_Win_free(win, ierror)
49
    TYPE(MPI_Win), INTENT(INOUT) :: win
50
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror
51

```

```

1  MPI_Win_get_group(win, group, ierror)
2      TYPE(MPI_Win), INTENT(IN) :: win
3      TYPE(MPI_Group), INTENT(OUT) :: group
4      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
5
6  MPI_Win_get_info(win, info_used, ierror)
7      TYPE(MPI_Win), INTENT(IN) :: win
8      TYPE(MPI_Info), INTENT(OUT) :: info_used
9      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
10
11 MPI_Win_lock_all(assert, win, ierror)
12     INTEGER, INTENT(IN) :: assert
13     TYPE(MPI_Win), INTENT(IN) :: win
14     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
15
16 MPI_Win_lock(lock_type, rank, assert, win, ierror)
17     INTEGER, INTENT(IN) :: lock_type, rank, assert
18     TYPE(MPI_Win), INTENT(IN) :: win
19     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
20
21 MPI_Win_post(group, assert, win, ierror)
22     TYPE(MPI_Group), INTENT(IN) :: group
23     INTEGER, INTENT(IN) :: assert
24     TYPE(MPI_Win), INTENT(IN) :: win
25     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
26
27 MPI_Win_set_info(win, info, ierror)
28     TYPE(MPI_Win), INTENT(IN) :: win
29     TYPE(MPI_Info), INTENT(IN) :: info
30     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
31
32 MPI_Win_shared_query(win, rank, size, disp_unit, baseptr, ierror)
33     USE, INTRINSIC :: ISO_C_BINDING, ONLY : C_PTR
34     TYPE(MPI_Win), INTENT(IN) :: win
35     INTEGER, INTENT(IN) :: rank
36     INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) :: size
37     INTEGER, INTENT(OUT) :: disp_unit
38     TYPE(C_PTR), INTENT(OUT) :: baseptr
39     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
40
41 MPI_Win_start(group, assert, win, ierror)
42     TYPE(MPI_Group), INTENT(IN) :: group
43     INTEGER, INTENT(IN) :: assert
44     TYPE(MPI_Win), INTENT(IN) :: win
45     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
46
47 MPI_Win_sync(win, ierror)
48     TYPE(MPI_Win), INTENT(IN) :: win
49     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
50
51 MPI_Win_test(win, flag, ierror)
52     TYPE(MPI_Win), INTENT(IN) :: win

```

```

    LOGICAL, INTENT(OUT) :: flag
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Win_unlock_all(win, ierror)
    TYPE(MPI_Win), INTENT(IN) :: win
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Win_unlock(rank, win, ierror)
    INTEGER, INTENT(IN) :: rank
    TYPE(MPI_Win), INTENT(IN) :: win
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Win_wait(win, ierror)
    TYPE(MPI_Win), INTENT(IN) :: win
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

A.3.10 External Interfaces Fortran 2008 Bindings

MPI_Grequest_complete(request, ierror)
    TYPE(MPI_Request), INTENT(IN) :: request
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Grequest_start(query_fn, free_fn, cancel_fn, extra_state, request,
    ierror)
    PROCEDURE(MPI_Grequest_query_function) :: query_fn
    PROCEDURE(MPI_Grequest_free_function) :: free_fn
    PROCEDURE(MPI_Grequest_cancel_function) :: cancel_fn
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: extra_state
    TYPE(MPI_Request), INTENT(OUT) :: request
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Init_thread(required, provided, ierror)
    INTEGER, INTENT(IN) :: required
    INTEGER, INTENT(OUT) :: provided
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Is_thread_main(flag, ierror)
    LOGICAL, INTENT(OUT) :: flag
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Query_thread(provided, ierror)
    INTEGER, INTENT(OUT) :: provided
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Status_set_cancelled(status, flag, ierror)
    TYPE(MPI_Status), INTENT(INOUT) :: status
    LOGICAL, INTENT(OUT) :: flag
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Status_set_elements(status, datatype, count, ierror)
    TYPE(MPI_Status), INTENT(INOUT) :: status
    TYPE(MPI_Datatype), INTENT(IN) :: datatype

```



```

1      INTEGER, INTENT(IN) :: count
2      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
3
4      MPI_Status_set_elements_x(status, datatype, count, ierror)
5      TYPE(MPI_Status), INTENT(INOUT) :: status
6      TYPE(MPI_Datatype), INTENT(IN) :: datatype
7      INTEGER(KIND = MPI_COUNT_KIND), INTENT(IN) :: count
8      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
9

```

### A.3.11 I/O Fortran 2008 Bindings

```

10
11
12      MPI_CONVERSION_FN_NULL(userbuf, datatype, count, filebuf, position,
13          extra_state, ierror)
14      USE, INTRINSIC :: ISO_C_BINDING, ONLY : C_PTR
15      TYPE(C_PTR), VALUE :: userbuf, filebuf
16      TYPE(MPI_Datatype) :: datatype
17      INTEGER :: count, ierror
18      INTEGER(KIND=MPI_OFFSET_KIND) :: position
19      INTEGER(KIND=MPI_ADDRESS_KIND) :: extra_state
20
21      MPI_File_close(fh, ierror)
22      TYPE(MPI_File), INTENT(INOUT) :: fh
23      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
24
25      MPI_File_delete(filename, info, ierror)
26      CHARACTER(LEN=*), INTENT(IN) :: filename
27      TYPE(MPI_Info), INTENT(IN) :: info
28      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
29
30      MPI_File_get_amode(fh, amode, ierror)
31      TYPE(MPI_File), INTENT(IN) :: fh
32      INTEGER, INTENT(OUT) :: amode
33      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
34
35      MPI_File_get_atomicity(fh, flag, ierror)
36      TYPE(MPI_File), INTENT(IN) :: fh
37      LOGICAL, INTENT(OUT) :: flag
38      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
39
40      MPI_File_get_byte_offset(fh, offset, disp, ierror)
41      TYPE(MPI_File), INTENT(IN) :: fh
42      INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: offset
43      INTEGER(KIND=MPI_OFFSET_KIND), INTENT(OUT) :: disp
44      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
45
46      MPI_File_get_group(fh, group, ierror)
47      TYPE(MPI_File), INTENT(IN) :: fh
48      TYPE(MPI_Group), INTENT(OUT) :: group
49      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
50
51      MPI_File_get_info(fh, info_used, ierror)
52

```

```

TYPE(MPI_File), INTENT(IN) :: fh
TYPE(MPI_Info), INTENT(OUT) :: info_used
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_get_position(fh, offset, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
INTEGER(KIND=MPI_OFFSET_KIND), INTENT(OUT) :: offset
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_get_position_shared(fh, offset, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
INTEGER(KIND=MPI_OFFSET_KIND), INTENT(OUT) :: offset
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_get_size(fh, size, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
INTEGER(KIND=MPI_OFFSET_KIND), INTENT(OUT) :: size
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_get_type_extent(fh, datatype, extent, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
TYPE(MPI_Datatype), INTENT(IN) :: datatype
INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) :: extent
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_get_view(fh, disp, etype, filetype, datarep, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
INTEGER(KIND=MPI_OFFSET_KIND), INTENT(OUT) :: disp
TYPE(MPI_Datatype), INTENT(OUT) :: etype, filetype
CHARACTER(LEN=*), INTENT(OUT) :: datarep
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_iread_all(fh, buf, count, datatype, request, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buf
INTEGER, INTENT(IN) :: count
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_iread_at_all(fh, offset, buf, count, datatype, request, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: offset
TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buf
INTEGER, INTENT(IN) :: count
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_iread_at(fh, offset, buf, count, datatype, request, ierror)
TYPE(MPI_File), INTENT(IN) :: fh

```

```

1      INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: offset
2      TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buf
3      INTEGER, INTENT(IN) :: count
4      TYPE(MPI_Datatype), INTENT(IN) :: datatype
5      TYPE(MPI_Request), INTENT(OUT) :: request
6      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
7
8      MPI_File_iread(fh, buf, count, datatype, request, ierror)
9      TYPE(MPI_File), INTENT(IN) :: fh
10     TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buf
11     INTEGER, INTENT(IN) :: count
12     TYPE(MPI_Datatype), INTENT(IN) :: datatype
13     TYPE(MPI_Request), INTENT(OUT) :: request
14     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
15
16     MPI_File_iread_shared(fh, buf, count, datatype, request, ierror)
17     TYPE(MPI_File), INTENT(IN) :: fh
18     TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buf
19     INTEGER, INTENT(IN) :: count
20     TYPE(MPI_Datatype), INTENT(IN) :: datatype
21     TYPE(MPI_Request), INTENT(OUT) :: request
22     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
23
24     MPI_File_iwrite_all(fh, buf, count, datatype, request, ierror)
25     TYPE(MPI_File), INTENT(IN) :: fh
26     TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
27     INTEGER, INTENT(IN) :: count
28     TYPE(MPI_Datatype), INTENT(IN) :: datatype
29     TYPE(MPI_Request), INTENT(OUT) :: request
30     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
31
32     MPI_File_iwrite_at_all(fh, offset, buf, count, datatype, request, ierror)
33     TYPE(MPI_File), INTENT(IN) :: fh
34     INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: offset
35     TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
36     INTEGER, INTENT(IN) :: count
37     TYPE(MPI_Datatype), INTENT(IN) :: datatype
38     TYPE(MPI_Request), INTENT(OUT) :: request
39     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
40
41     MPI_File_iwrite_at(fh, offset, buf, count, datatype, request, ierror)
42     TYPE(MPI_File), INTENT(IN) :: fh
43     INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: offset
44     TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
45     INTEGER, INTENT(IN) :: count
46     TYPE(MPI_Datatype), INTENT(IN) :: datatype
47     TYPE(MPI_Request), INTENT(OUT) :: request
48     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
49
50     MPI_File_iwrite(fh, buf, count, datatype, request, ierror)

```

```

TYPE(MPI_File), INTENT(IN) :: fh
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
INTEGER, INTENT(IN) :: count
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_iwrite_shared(fh, buf, count, datatype, request, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
INTEGER, INTENT(IN) :: count
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Request), INTENT(OUT) :: request
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_open(comm, filename, amode, info, fh, ierror)
TYPE(MPI_Comm), INTENT(IN) :: comm
CHARACTER(LEN=*), INTENT(IN) :: filename
INTEGER, INTENT(IN) :: amode
TYPE(MPI_Info), INTENT(IN) :: info
TYPE(MPI_File), INTENT(OUT) :: fh
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_preallocate(fh, size, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: size
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_read_all_begin(fh, buf, count, datatype, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buf
INTEGER, INTENT(IN) :: count
TYPE(MPI_Datatype), INTENT(IN) :: datatype
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_read_all_end(fh, buf, status, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buf
TYPE(MPI_Status) :: status
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_read_all(fh, buf, count, datatype, status, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
TYPE(*), DIMENSION(..) :: buf
INTEGER, INTENT(IN) :: count
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Status) :: status
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_read_at_all_begin(fh, offset, buf, count, datatype, ierror)
TYPE(MPI_File), INTENT(IN) :: fh

```

```

1      INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: offset
2      TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buf
3      INTEGER, INTENT(IN) :: count
4      TYPE(MPI_Datatype), INTENT(IN) :: datatype
5      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
6
7  MPI_File_read_at_all_end(fh, buf, status, ierror)
8      TYPE(MPI_File), INTENT(IN) :: fh
9      TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buf
10     TYPE(MPI_Status) :: status
11     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
12
13 MPI_File_read_at_all(fh, offset, buf, count, datatype, status, ierror)
14     TYPE(MPI_File), INTENT(IN) :: fh
15     INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: offset
16     TYPE(*), DIMENSION(..) :: buf
17     INTEGER, INTENT(IN) :: count
18     TYPE(MPI_Datatype), INTENT(IN) :: datatype
19     TYPE(MPI_Status) :: status
20     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
21
22 MPI_File_read_at(fh, offset, buf, count, datatype, status, ierror)
23     TYPE(MPI_File), INTENT(IN) :: fh
24     INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: offset
25     TYPE(*), DIMENSION(..) :: buf
26     INTEGER, INTENT(IN) :: count
27     TYPE(MPI_Datatype), INTENT(IN) :: datatype
28     TYPE(MPI_Status) :: status
29     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
30
31 MPI_File_read(fh, buf, count, datatype, status, ierror)
32     TYPE(MPI_File), INTENT(IN) :: fh
33     TYPE(*), DIMENSION(..) :: buf
34     INTEGER, INTENT(IN) :: count
35     TYPE(MPI_Datatype), INTENT(IN) :: datatype
36     TYPE(MPI_Status) :: status
37     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
38
39 MPI_File_read_ordered_begin(fh, buf, count, datatype, ierror)
40     TYPE(MPI_File), INTENT(IN) :: fh
41     TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buf
42     INTEGER, INTENT(IN) :: count
43     TYPE(MPI_Datatype), INTENT(IN) :: datatype
44     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
45
46 MPI_File_read_ordered_end(fh, buf, status, ierror)
47     TYPE(MPI_File), INTENT(IN) :: fh
48     TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buf
49     TYPE(MPI_Status) :: status
50     INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

MPI_File_read_ordered(fh, buf, count, datatype, status, ierror)      1
    TYPE(MPI_File), INTENT(IN) :: fh                                2
    TYPE(*), DIMENSION(..) :: buf                                  3
    INTEGER, INTENT(IN) :: count                                    4
    TYPE(MPI_Datatype), INTENT(IN) :: datatype                      5
    TYPE(MPI_Status) :: status                                       6
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                         7
                                                                    8
MPI_File_read_shared(fh, buf, count, datatype, status, ierror)      9
    TYPE(MPI_File), INTENT(IN) :: fh                                10
    TYPE(*), DIMENSION(..) :: buf                                  11
    INTEGER, INTENT(IN) :: count                                    12
    TYPE(MPI_Datatype), INTENT(IN) :: datatype                      13
    TYPE(MPI_Status) :: status                                       14
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                         15
                                                                    16
MPI_File_seek(fh, offset, whence, ierror)                            17
    TYPE(MPI_File), INTENT(IN) :: fh                                18
    INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: offset            19
    INTEGER, INTENT(IN) :: whence                                   20
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                         21
                                                                    22
MPI_File_seek_shared(fh, offset, whence, ierror)                    23
    TYPE(MPI_File), INTENT(IN) :: fh                                24
    INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: offset            25
    INTEGER, INTENT(IN) :: whence                                   26
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                         27
                                                                    28
MPI_File_set_atomicity(fh, flag, ierror)                             29
    TYPE(MPI_File), INTENT(IN) :: fh                                30
    LOGICAL, INTENT(IN) :: flag                                     31
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                         32
                                                                    33
MPI_File_set_info(fh, info, ierror)                                  34
    TYPE(MPI_File), INTENT(IN) :: fh                                35
    TYPE(MPI_Info), INTENT(IN) :: info                              36
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                         37
                                                                    38
MPI_File_set_size(fh, size, ierror)                                   39
    TYPE(MPI_File), INTENT(IN) :: fh                                40
    INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: size              41
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror                         42
                                                                    43
MPI_File_set_view(fh, disp, etype, filetype, datarep, info, ierror) 44
    TYPE(MPI_File), INTENT(IN) :: fh                                45
    INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: disp              46
    TYPE(MPI_Datatype), INTENT(IN) :: etype, filetype              47
    CHARACTER(LEN=*), INTENT(IN) :: datarep                        48
    TYPE(MPI_Info), INTENT(IN) :: info                              49
    INTEGER, OPTIONAL, INTENT(OUT) :: ierror

```

```

1  MPI_File_sync(fh, ierror)
2      TYPE(MPI_File), INTENT(IN) :: fh
3      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
4
5  MPI_File_write_all_begin(fh, buf, count, datatype, ierror)
6      TYPE(MPI_File), INTENT(IN) :: fh
7      TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
8      INTEGER, INTENT(IN) :: count
9      TYPE(MPI_Datatype), INTENT(IN) :: datatype
10     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
11
12 MPI_File_write_all_end(fh, buf, status, ierror)
13     TYPE(MPI_File), INTENT(IN) :: fh
14     TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
15     TYPE(MPI_Status) :: status
16     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
17
18 MPI_File_write_all(fh, buf, count, datatype, status, ierror)
19     TYPE(MPI_File), INTENT(IN) :: fh
20     TYPE(*), DIMENSION(..), INTENT(IN) :: buf
21     INTEGER, INTENT(IN) :: count
22     TYPE(MPI_Datatype), INTENT(IN) :: datatype
23     TYPE(MPI_Status) :: status
24     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
25
26 MPI_File_write_at_all_begin(fh, offset, buf, count, datatype, ierror)
27     TYPE(MPI_File), INTENT(IN) :: fh
28     INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: offset
29     TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
30     INTEGER, INTENT(IN) :: count
31     TYPE(MPI_Datatype), INTENT(IN) :: datatype
32     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
33
34 MPI_File_write_at_all_end(fh, buf, status, ierror)
35     TYPE(MPI_File), INTENT(IN) :: fh
36     TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
37     TYPE(MPI_Status) :: status
38     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
39
40 MPI_File_write_at_all(fh, offset, buf, count, datatype, status, ierror)
41     TYPE(MPI_File), INTENT(IN) :: fh
42     INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: offset
43     TYPE(*), DIMENSION(..), INTENT(IN) :: buf
44     INTEGER, INTENT(IN) :: count
45     TYPE(MPI_Datatype), INTENT(IN) :: datatype
46     TYPE(MPI_Status) :: status
47     INTEGER, OPTIONAL, INTENT(OUT) :: ierror
48
49 MPI_File_write_at(fh, offset, buf, count, datatype, status, ierror)
50     TYPE(MPI_File), INTENT(IN) :: fh
51     INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) :: offset

```

```

TYPE(*), DIMENSION(..), INTENT(IN) :: buf
INTEGER, INTENT(IN) :: count
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Status) :: status
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_write(fh, buf, count, datatype, status, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
TYPE(*), DIMENSION(..), INTENT(IN) :: buf
INTEGER, INTENT(IN) :: count
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Status) :: status
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_write_ordered_begin(fh, buf, count, datatype, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
INTEGER, INTENT(IN) :: count
TYPE(MPI_Datatype), INTENT(IN) :: datatype
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_write_ordered_end(fh, buf, status, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS :: buf
TYPE(MPI_Status) :: status
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_write_ordered(fh, buf, count, datatype, status, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
TYPE(*), DIMENSION(..), INTENT(IN) :: buf
INTEGER, INTENT(IN) :: count
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Status) :: status
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_File_write_shared(fh, buf, count, datatype, status, ierror)
TYPE(MPI_File), INTENT(IN) :: fh
TYPE(*), DIMENSION(..), INTENT(IN) :: buf
INTEGER, INTENT(IN) :: count
TYPE(MPI_Datatype), INTENT(IN) :: datatype
TYPE(MPI_Status) :: status
INTEGER, OPTIONAL, INTENT(OUT) :: ierror

MPI_Register_datarep(datarep, read_conversion_fn, write_conversion_fn,
dtype_file_extent_fn, extra_state, ierror)
CHARACTER(LEN=*), INTENT(IN) :: datarep
PROCEDURE(MPI_Datarep_conversion_function) :: read_conversion_fn
PROCEDURE(MPI_Datarep_conversion_function) :: write_conversion_fn
PROCEDURE(MPI_Datarep_extent_function) :: dtype_file_extent_fn
INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) :: extra_state

```



```

1      INTEGER, OPTIONAL, INTENT(OUT) :: ierror
2
3

```

#### 4 A.3.12 Language Bindings Fortran 2008 Bindings

```

5      MPI_F_sync_reg(buf)
6          TYPE(*), DIMENSION(..), ASYNCHRONOUS :: buf
7
8      MPI_Sizeof(x, size, ierror)
9          TYPE(*), DIMENSION(..) :: x
10         INTEGER, INTENT(OUT) :: size
11         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
12
13     MPI_Status_f082f(f08_status, f_status, ierror)
14         TYPE(MPI_Status), INTENT(IN) :: f08_status
15         INTEGER, INTENT(OUT) :: f_status(MPI_STATUS_SIZE)
16         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
17
18     MPI_Status_f2f08(f_status, f08_status, ierror)
19         INTEGER, INTENT(IN) :: f_status(MPI_STATUS_SIZE)
20         TYPE(MPI_Status), INTENT(OUT) :: f08_status
21         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
22
23     MPI_Type_create_f90_complex(p, r, newtype, ierror)
24         INTEGER, INTENT(IN) :: p, r
25         TYPE(MPI_Datatype), INTENT(OUT) :: newtype
26         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
27
28     MPI_Type_create_f90_integer(r, newtype, ierror)
29         INTEGER, INTENT(IN) :: r
30         TYPE(MPI_Datatype), INTENT(OUT) :: newtype
31         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
32
33     MPI_Type_create_f90_real(p, r, newtype, ierror)
34         INTEGER, INTENT(IN) :: p, r
35         TYPE(MPI_Datatype), INTENT(OUT) :: newtype
36         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
37
38     MPI_Type_match_size(typeclass, size, datatype, ierror)
39         INTEGER, INTENT(IN) :: typeclass, size
40         TYPE(MPI_Datatype), INTENT(OUT) :: datatype
41         INTEGER, OPTIONAL, INTENT(OUT) :: ierror
42
43

```

#### 41 A.3.13 Tools / Profiling Interface Fortran 2008 Bindings

```

42     MPI_Pcontrol(level)
43         INTEGER, INTENT(IN) :: level
44
45
46
47
48

```