A.3 Fortran 2008 Bindings with the mpi_f08 Module

#388

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

BIND(C) is removed from all mpi_f08 interfaces

```
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
```

```
    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
```

```
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Mprobe(source, tag, comm, message, status, ierror)
    INTEGER, INTENT(IN) ::  source, tag
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    TYPE(MPI_Message), INTENT(OUT) ::  message
    TYPE(MPI_Status) ::  status
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

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

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

MPI_Recv_init(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_Request_free(request, ierror)
    TYPE(MPI_Request), INTENT(INOUT) ::  request
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Request_get_status(request, flag, status, ierror)
    TYPE(MPI_Request), INTENT(IN) ::  request
    LOGICAL, INTENT(OUT) ::  flag
    TYPE(MPI_Status) ::  status
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Rsend(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_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
```

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

MPI_Ssend_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_Startall(count, array_of_requests, ierror)
    INTEGER, INTENT(IN) ::  count
    TYPE(MPI_Request), INTENT(INOUT) ::  array_of_requests(count)
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Start(request, ierror)
    TYPE(MPI_Request), INTENT(INOUT) ::  request
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

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

MPI_Test_cancelled(status, flag, ierror)
    TYPE(MPI_Status), INTENT(IN) ::  status
    LOGICAL, INTENT(OUT) ::  flag
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Test(request, flag, status, ierror)
    TYPE(MPI_Request), INTENT(INOUT) ::  request
    LOGICAL, INTENT(OUT) ::  flag
    TYPE(MPI_Status) ::  status
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Testsome(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

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)
```

```
    TYPE(MPI_Status), INTENT(IN) ::  status
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER(KIND = MPI_COUNT_KIND), INTENT(OUT) ::  count
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Pack_external(datarep, inbuf, incount, datatype, outbuf, outsize,
             position, ierror)
    CHARACTER(LEN=*), INTENT(IN) ::  datarep
    TYPE(*), DIMENSION(..), INTENT(IN) ::  inbuf
    TYPE(*), DIMENSION(..)  ::  outbuf
    INTEGER, INTENT(IN) ::  incount
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) ::  outsize
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(INOUT) ::  position
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Pack_external_size(datarep, incount, datatype, size, ierror)
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER, INTENT(IN) ::  incount
    CHARACTER(LEN=*), INTENT(IN) ::  datarep
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) ::  size
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Pack(inbuf, incount, datatype, outbuf, outsize, position, comm, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN) ::  inbuf
    TYPE(*), DIMENSION(..)  ::  outbuf
    INTEGER, INTENT(IN) ::  incount, outsize
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER, INTENT(INOUT) ::  position
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Pack_size(incount, datatype, comm, size, ierror)
    INTEGER, INTENT(IN) ::  incount
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(OUT) ::  size
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_commit(datatype, ierror)
    TYPE(MPI_Datatype), INTENT(INOUT) ::  datatype
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_contiguous(count, oldtype, newtype, ierror)
    INTEGER, INTENT(IN) ::  count
    TYPE(MPI_Datatype), INTENT(IN) ::  oldtype
    TYPE(MPI_Datatype), INTENT(OUT) ::  newtype
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_create_darray(size, rank, ndims, array_of_gsizes,
             array_of_distribs, array_of_dargs, array_of_psizes, order,
```

```
            oldtype, newtype, ierror)
    INTEGER, INTENT(IN) ::  size, rank, ndims, array_of_gsizes(ndims),
    array_of_distribs(ndims), array_of_dargs(ndims),
    array_of_psizes(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)
```

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

MPI_Type_create_subarray(ndims, array_of_sizes, array_of_subsizes,
             array_of_starts, order, oldtype, newtype, ierror)
    INTEGER, INTENT(IN) ::  ndims, array_of_sizes(ndims),
    array_of_subsizes(ndims), array_of_starts(ndims), order
    TYPE(MPI_Datatype), INTENT(IN) ::  oldtype
    TYPE(MPI_Datatype), INTENT(OUT) ::  newtype
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_dup(oldtype, newtype, ierror)
    TYPE(MPI_Datatype), INTENT(IN) ::  oldtype
    TYPE(MPI_Datatype), INTENT(OUT) ::  newtype
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_free(datatype, ierror)
    TYPE(MPI_Datatype), INTENT(INOUT) ::  datatype
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_get_contents(datatype, max_integers, max_addresses, max_datatypes,
             array_of_integers, array_of_addresses, array_of_datatypes,
             ierror)
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER, INTENT(IN) ::  max_integers, max_addresses, max_datatypes
    INTEGER, INTENT(OUT) ::  array_of_integers(max_integers)
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) ::
    array_of_addresses(max_addresses)
    TYPE(MPI_Datatype), INTENT(OUT) ::  array_of_datatypes(max_datatypes)
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_get_envelope(datatype, num_integers, num_addresses, num_datatypes,
             combiner, ierror)
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER, INTENT(OUT) ::  num_integers, num_addresses, num_datatypes,
    combiner
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_get_extent(datatype, lb, extent, ierror)
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) ::  lb, extent
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_get_extent_x(datatype, lb, extent, ierror)
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER(KIND = MPI_COUNT_KIND), INTENT(OUT) ::  lb, extent
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror
```

```
MPI_Type_get_true_extent(datatype, true_lb, true_extent, ierror)
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) ::  true_lb, true_extent
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_get_true_extent_x(datatype, true_lb, true_extent, ierror)
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER(KIND = MPI_COUNT_KIND), INTENT(OUT) ::  true_lb, true_extent
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

MPI_Type_size(datatype, size, ierror)
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER, INTENT(OUT) ::  size
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_size_x(datatype, size, ierror)
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER(KIND=MPI_COUNT_KIND), INTENT(OUT) ::  size
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

MPI_Unpack_external(datarep, inbuf, insize, position, outbuf, outcount,
            datatype, ierror)
    CHARACTER(LEN=*), INTENT(IN) ::  datarep
    TYPE(*), DIMENSION(..), INTENT(IN) ::  inbuf
    TYPE(*), DIMENSION(..)  ::  outbuf
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) ::  insize
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(INOUT) ::  position
    INTEGER, INTENT(IN) ::  outcount
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Unpack(inbuf, insize, position, outbuf, outcount, datatype, comm,
            ierror)
    TYPE(*), DIMENSION(..), INTENT(IN) ::  inbuf
    TYPE(*), DIMENSION(..)  ::  outbuf
    INTEGER, INTENT(IN) ::  insize, outcount
    INTEGER, INTENT(INOUT) ::  position
```

```
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror
```

A.3.3 Collective Communication Fortran 2008 Bindings

```
MPI_Allgather(sendbuf, sendcount, sendtype, recvbuf, recvcount, recvtype,
             comm, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN) ::  sendbuf
    TYPE(*), DIMENSION(..)  ::  recvbuf
    INTEGER, INTENT(IN) ::  sendcount, recvcount
    TYPE(MPI_Datatype), INTENT(IN) ::  sendtype, recvtype
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

MPI_Allreduce(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_Alltoall(sendbuf, sendcount, sendtype, recvbuf, recvcount, recvtype,
             comm, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN) ::  sendbuf
    TYPE(*), DIMENSION(..)  ::  recvbuf
    INTEGER, INTENT(IN) ::  sendcount, recvcount
    TYPE(MPI_Datatype), INTENT(IN) ::  sendtype, recvtype
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Alltoallv(sendbuf, sendcounts, sdispls, sendtype, recvbuf, recvcounts,
             rdispls, recvtype, comm, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN) ::  sendbuf
    TYPE(*), DIMENSION(..)  ::  recvbuf
    INTEGER, INTENT(IN) ::  sendcounts(*), sdispls(*), recvcounts(*),
    rdispls(*)
    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, recvcounts,
            rdispls, recvtypes, comm, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN) ::  sendbuf
    TYPE(*), DIMENSION(..)  ::  recvbuf
    INTEGER, INTENT(IN) ::  sendcounts(*), sdispls(*), recvcounts(*),
    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, recvcount, recvtype,
            root, comm, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN) ::  sendbuf
    TYPE(*), DIMENSION(..)  ::  recvbuf
    INTEGER, INTENT(IN) ::  sendcount, recvcount, 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, recvcounts, displs,
            recvtype, root, comm, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN) ::  sendbuf
    TYPE(*), DIMENSION(..)  ::  recvbuf
    INTEGER, INTENT(IN) ::  sendcount, recvcounts(*), displs(*), root
    TYPE(MPI_Datatype), INTENT(IN) ::  sendtype, recvtype
    TYPE(MPI_Comm), INTENT(IN) ::  comm
```

```
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Iallgather(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_Iallgatherv(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_Iallreduce(sendbuf, recvbuf, count, datatype, op, comm, request,
              ierror)
    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS ::  sendbuf
    TYPE(*), DIMENSION(..), ASYNCHRONOUS ::  recvbuf
    INTEGER, INTENT(IN) ::  count
    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_Ialltoall(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_Ialltoallv(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_Ialltoallw(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(*), sdispls(*),
    recvcounts(*), 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_Ibarrier(comm, request, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    TYPE(MPI_Request), INTENT(OUT) ::  request
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Ibcast(buffer, count, datatype, root, comm, request, ierror)
    TYPE(*), DIMENSION(..), ASYNCHRONOUS ::  buffer
    INTEGER, INTENT(IN) ::  count, root
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    TYPE(MPI_Request), INTENT(OUT) ::  request
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Iexscan(sendbuf, recvbuf, count, datatype, op, comm, request, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS ::  sendbuf
    TYPE(*), DIMENSION(..), ASYNCHRONOUS ::  recvbuf
    INTEGER, INTENT(IN) ::  count
    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_Igather(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_Igatherv(sendbuf, sendcount, sendtype, recvbuf, recvcounts, displs,
             recvtype, root, comm, request, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS ::  sendbuf
    TYPE(*), DIMENSION(..), ASYNCHRONOUS ::  recvbuf
    INTEGER, INTENT(IN) ::  sendcount, root
    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_Ireduce_scatter_block(sendbuf, recvbuf, recvcount, datatype, op, comm,
             request, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS ::  sendbuf
    TYPE(*), DIMENSION(..), ASYNCHRONOUS ::  recvbuf
    INTEGER, INTENT(IN) ::  recvcount
    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_Ireduce_scatter(sendbuf, recvbuf, recvcounts, datatype, op, comm,
             request, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS ::  sendbuf
    TYPE(*), DIMENSION(..), ASYNCHRONOUS ::  recvbuf
    INTEGER, INTENT(IN), ASYNCHRONOUS ::  recvcounts(*)
    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_Ireduce(sendbuf, recvbuf, count, datatype, op, root, comm, request,
             ierror)
    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS ::  sendbuf
    TYPE(*), DIMENSION(..), ASYNCHRONOUS ::  recvbuf
    INTEGER, INTENT(IN) ::  count, root
    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_Iscan(sendbuf, recvbuf, count, datatype, op, comm, request, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS ::  sendbuf
    TYPE(*), DIMENSION(..), ASYNCHRONOUS ::  recvbuf
    INTEGER, INTENT(IN) ::  count
    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_Iscatter(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_Iscatterv(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)
```

```
    TYPE(*), DIMENSION(..), INTENT(IN) ::  sendbuf
    TYPE(*), DIMENSION(..)  ::  recvbuf
    INTEGER, INTENT(IN) ::  recvcount
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    TYPE(MPI_Op), INTENT(IN) ::  op
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Reduce_scatter(sendbuf, recvbuf, recvcounts, datatype, op, comm,
            ierror)
    TYPE(*), DIMENSION(..), INTENT(IN) ::  sendbuf
    TYPE(*), DIMENSION(..)  ::  recvbuf
    INTEGER, INTENT(IN) ::  recvcounts(*)
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    TYPE(MPI_Op), INTENT(IN) ::  op
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Reduce(sendbuf, recvbuf, count, datatype, op, root, comm, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN) ::  sendbuf
    TYPE(*), DIMENSION(..)  ::  recvbuf
    INTEGER, INTENT(IN) ::  count, root
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    TYPE(MPI_Op), INTENT(IN) ::  op
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

MPI_Scatterv(sendbuf, sendcounts, displs, sendtype, recvbuf, recvcount,
            recvtype, root, comm, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN) ::  sendbuf
    TYPE(*), DIMENSION(..)  ::  recvbuf
    INTEGER, INTENT(IN) ::  sendcounts(*), displs(*), recvcount, 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
```

```
MPI_Comm_dup_with_info(comm, info, newcomm, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    TYPE(MPI_Info), INTENT(IN) ::  info
    TYPE(MPI_Comm), INTENT(OUT) ::  newcomm
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

MPI_Comm_free_keyval(comm_keyval, ierror)
    INTEGER, INTENT(INOUT) ::  comm_keyval
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Comm_get_attr(comm, comm_keyval, attribute_val, flag, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(IN) ::  comm_keyval
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) ::  attribute_val
    LOGICAL, INTENT(OUT) ::  flag
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Comm_get_info(comm, info_used, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    TYPE(MPI_Info), INTENT(OUT) ::  info_used
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Comm_get_name(comm, comm_name, resultlen, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    CHARACTER(LEN=MPI_MAX_OBJECT_NAME), INTENT(OUT) ::  comm_name
    INTEGER, INTENT(OUT) ::  resultlen
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

MPI_Comm_idup(comm, newcomm, request, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    TYPE(MPI_Comm), INTENT(OUT), ASYNCHRONOUS ::  newcomm
    TYPE(MPI_Request), INTENT(OUT) ::  request
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_COMM_NULL_COPY_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

```

```
MPI_COMM_NULL_DELETE_FN(comm, comm_keyval, attribute_val, extra_state,
            ierror)
    TYPE(MPI_Comm) ::  comm
    INTEGER ::  comm_keyval
    INTEGER(KIND=MPI_ADDRESS_KIND) ::  attribute_val, extra_state
    INTEGER ::  ierror

MPI_Comm_rank(comm, rank, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(OUT) ::  rank
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

MPI_Comm_remote_size(comm, size, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(OUT) ::  size
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Comm_set_attr(comm, comm_keyval, attribute_val, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(IN) ::  comm_keyval
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) ::  attribute_val
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Comm_set_info(comm, info, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    TYPE(MPI_Info), INTENT(IN) ::  info
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Comm_set_name(comm, comm_name, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    CHARACTER(LEN=*), INTENT(IN) ::  comm_name
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Comm_size(comm, size, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(OUT) ::  size
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Comm_split(comm, color, key, newcomm, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(IN) ::  color, key
    TYPE(MPI_Comm), INTENT(OUT) ::  newcomm
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Comm_split_type(comm, split_type, key, info, newcomm, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(IN) ::  split_type, key
```

```
    TYPE(MPI_Info), INTENT(IN) ::  info
    TYPE(MPI_Comm), INTENT(OUT) ::  newcomm
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Comm_test_inter(comm, flag, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    LOGICAL, INTENT(OUT) ::  flag
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Group_compare(group1, group2, result, ierror)
    TYPE(MPI_Group), INTENT(IN) ::  group1, group2
    INTEGER, INTENT(OUT) ::  result
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Group_difference(group1, group2, newgroup, ierror)
    TYPE(MPI_Group), INTENT(IN) ::  group1, group2
    TYPE(MPI_Group), INTENT(OUT) ::  newgroup
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Group_excl(group, n, ranks, newgroup, ierror)
    TYPE(MPI_Group), INTENT(IN) ::  group
    INTEGER, INTENT(IN) ::  n, ranks(n)
    TYPE(MPI_Group), INTENT(OUT) ::  newgroup
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Group_free(group, ierror)
    TYPE(MPI_Group), INTENT(INOUT) ::  group
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Group_incl(group, n, ranks, newgroup, ierror)
    TYPE(MPI_Group), INTENT(IN) ::  group
    INTEGER, INTENT(IN) ::  n, ranks(n)
    TYPE(MPI_Group), INTENT(OUT) ::  newgroup
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Group_intersection(group1, group2, newgroup, ierror)
    TYPE(MPI_Group), INTENT(IN) ::  group1, group2
    TYPE(MPI_Group), INTENT(OUT) ::  newgroup
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Group_range_excl(group, n, ranges, newgroup, ierror)
    TYPE(MPI_Group), INTENT(IN) ::  group
    INTEGER, INTENT(IN) ::  n, ranges(3,n)
    TYPE(MPI_Group), INTENT(OUT) ::  newgroup
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Group_range_incl(group, n, ranges, newgroup, ierror)
    TYPE(MPI_Group), INTENT(IN) ::  group
    INTEGER, INTENT(IN) ::  n, ranges(3,n)
    TYPE(MPI_Group), INTENT(OUT) ::  newgroup
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror
```

```
MPI_Group_rank(group, rank, ierror)
    TYPE(MPI_Group), INTENT(IN) ::  group
    INTEGER, INTENT(OUT) ::  rank
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Group_size(group, size, ierror)
    TYPE(MPI_Group), INTENT(IN) ::  group
    INTEGER, INTENT(OUT) ::  size
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Group_translate_ranks(group1, n, ranks1, group2, ranks2, ierror)
    TYPE(MPI_Group), INTENT(IN) ::  group1, group2
    INTEGER, INTENT(IN) ::  n, ranks1(n)
    INTEGER, INTENT(OUT) ::  ranks2(n)
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Group_union(group1, group2, newgroup, ierror)
    TYPE(MPI_Group), INTENT(IN) ::  group1, group2
    TYPE(MPI_Group), INTENT(OUT) ::  newgroup
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Intercomm_create(local_comm, local_leader, peer_comm, remote_leader,
            tag, newintercomm, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  local_comm, peer_comm
    INTEGER, INTENT(IN) ::  local_leader, remote_leader, tag
    TYPE(MPI_Comm), INTENT(OUT) ::  newintercomm
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Intercomm_merge(intercomm, high, newintracomm, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  intercomm
    LOGICAL, INTENT(IN) ::  high
    TYPE(MPI_Comm), INTENT(OUT) ::  newintracomm
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_create_keyval(type_copy_attr_fn, type_delete_attr_fn, type_keyval,
            extra_state, ierror)
    PROCEDURE(MPI_Type_copy_attr_function) ::  type_copy_attr_fn
    PROCEDURE(MPI_Type_delete_attr_function) ::  type_delete_attr_fn
    INTEGER, INTENT(OUT) ::  type_keyval
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) ::  extra_state
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_delete_attr(datatype, type_keyval, ierror)
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER, INTENT(IN) ::  type_keyval
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_TYPE_DUP_FN(oldtype, type_keyval, extra_state, attribute_val_in,
            attribute_val_out, flag, ierror)
    TYPE(MPI_Datatype) ::  oldtype
    INTEGER ::  type_keyval
```

```
    INTEGER(KIND=MPI_ADDRESS_KIND) ::  extra_state, attribute_val_in
    INTEGER(KIND=MPI_ADDRESS_KIND) ::  attribute_val_out
    LOGICAL ::  flag
    INTEGER ::  ierror

MPI_Type_free_keyval(type_keyval, ierror)
    INTEGER, INTENT(INOUT) ::  type_keyval
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_get_attr(datatype, type_keyval, attribute_val, flag, ierror)
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER, INTENT(IN) ::  type_keyval
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) ::  attribute_val
    LOGICAL, INTENT(OUT) ::  flag
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_get_name(datatype, type_name, resultlen, ierror)
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    CHARACTER(LEN=MPI_MAX_OBJECT_NAME), INTENT(OUT) ::  type_name
    INTEGER, INTENT(OUT) ::  resultlen
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_TYPE_NULL_COPY_FN(oldtype, type_keyval, extra_state, attribute_val_in,
             attribute_val_out, flag, ierror)
    TYPE(MPI_Datatype) ::  oldtype
    INTEGER ::  type_keyval
    INTEGER(KIND=MPI_ADDRESS_KIND) ::  extra_state, attribute_val_in
    INTEGER(KIND=MPI_ADDRESS_KIND) ::  attribute_val_out
    LOGICAL ::  flag
    INTEGER ::  ierror

MPI_TYPE_NULL_DELETE_FN(datatype, type_keyval, attribute_val, extra_state,
             ierror)
    TYPE(MPI_Datatype) ::  datatype
    INTEGER ::  type_keyval
    INTEGER(KIND=MPI_ADDRESS_KIND) ::  attribute_val, extra_state
    INTEGER, INTENT(OUT) ::  ierror

MPI_Type_set_attr(datatype, type_keyval, attribute_val, ierror)
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER, INTENT(IN) ::  type_keyval
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) ::  attribute_val
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_set_name(datatype, type_name, ierror)
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    CHARACTER(LEN=*), INTENT(IN) ::  type_name
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Win_create_keyval(win_copy_attr_fn, win_delete_attr_fn, win_keyval,
             extra_state, ierror)

```

```
    PROCEDURE(MPI_Win_copy_attr_function) ::  win_copy_attr_fn
    PROCEDURE(MPI_Win_delete_attr_function) ::  win_delete_attr_fn
    INTEGER, INTENT(OUT) ::  win_keyval
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) ::  extra_state
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

MPI_WIN_DUP_FN(oldwin, win_keyval, extra_state, attribute_val_in,
            attribute_val_out, flag, ierror)
    TYPE(MPI_Win) ::  oldwin
    INTEGER ::  win_keyval
    INTEGER(KIND=MPI_ADDRESS_KIND) ::  extra_state, attribute_val_in
    INTEGER(KIND=MPI_ADDRESS_KIND) ::  attribute_val_out
    LOGICAL ::  flag
    INTEGER ::  ierror

MPI_Win_free_keyval(win_keyval, ierror)
    INTEGER, INTENT(INOUT) ::  win_keyval
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Win_get_attr(win, win_keyval, attribute_val, flag, ierror)
    TYPE(MPI_Win), INTENT(IN) ::  win
    INTEGER, INTENT(IN) ::  win_keyval
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) ::  attribute_val
    LOGICAL, INTENT(OUT) ::  flag
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Win_get_name(win, win_name, resultlen, ierror)
    TYPE(MPI_Win), INTENT(IN) ::  win
    CHARACTER(LEN=MPI_MAX_OBJECT_NAME), INTENT(OUT) ::  win_name
    INTEGER, INTENT(OUT) ::  resultlen
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_WIN_NULL_COPY_FN(oldwin, win_keyval, extra_state, attribute_val_in,
            attribute_val_out, flag, ierror)
    TYPE(MPI_Win) ::  oldwin
    INTEGER ::  win_keyval
    INTEGER(KIND=MPI_ADDRESS_KIND) ::  extra_state, attribute_val_in
    INTEGER(KIND=MPI_ADDRESS_KIND) ::  attribute_val_out
    LOGICAL ::  flag
    INTEGER ::  ierror

MPI_WIN_NULL_DELETE_FN(win, win_keyval, attribute_val, extra_state, ierror)
    TYPE(MPI_Win) ::  win
    INTEGER ::  win_keyval
    INTEGER(KIND=MPI_ADDRESS_KIND) ::  attribute_val, extra_state
    INTEGER ::  ierror
```

```
MPI_Win_set_attr(win, win_keyval, attribute_val, ierror)
    TYPE(MPI_Win), INTENT(IN) ::  win
    INTEGER, INTENT(IN) ::  win_keyval
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) ::  attribute_val
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Win_set_name(win, win_name, ierror)
    TYPE(MPI_Win), INTENT(IN) ::  win
    CHARACTER(LEN=*), INTENT(IN) ::  win_name
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror
```

A.3.5 Process Topologies Fortran 2008 Bindings

```
MPI_Cart_coords(comm, rank, maxdims, coords, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(IN) ::  rank, maxdims
    INTEGER, INTENT(OUT) ::  coords(maxdims)
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Cart_create(comm_old, ndims, dims, periods, reorder, comm_cart, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm_old
    INTEGER, INTENT(IN) ::  ndims, dims(ndims)
    LOGICAL, INTENT(IN) ::  periods(ndims), reorder
    TYPE(MPI_Comm), INTENT(OUT) ::  comm_cart
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Cartdim_get(comm, ndims, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(OUT) ::  ndims
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Cart_get(comm, maxdims, dims, periods, coords, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(IN) ::  maxdims
    INTEGER, INTENT(OUT) ::  dims(maxdims), coords(maxdims)
    LOGICAL, INTENT(OUT) ::  periods(maxdims)
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Cart_map(comm, ndims, dims, periods, newrank, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(IN) ::  ndims, dims(ndims)
    LOGICAL, INTENT(IN) ::  periods(ndims)
    INTEGER, INTENT(OUT) ::  newrank
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Cart_rank(comm, coords, rank, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(IN) ::  coords(*)
    INTEGER, INTENT(OUT) ::  rank
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror
```

```
MPI_Cart_shift(comm, direction, disp, rank_source, rank_dest, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(IN) ::  direction, disp
    INTEGER, INTENT(OUT) ::  rank_source, rank_dest
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Cart_sub(comm, remain_dims, newcomm, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    LOGICAL, INTENT(IN) ::  remain_dims(*)
    TYPE(MPI_Comm), INTENT(OUT) ::  newcomm
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Dims_create(nnodes, ndims, dims, ierror)
    INTEGER, INTENT(IN) ::  nnodes, ndims
    INTEGER, INTENT(INOUT) ::  dims(ndims)
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Dist_graph_create_adjacent(comm_old, indegree, sources, sourceweights,
             outdegree, destinations, destweights, info, reorder,
             comm_dist_graph, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm_old
    INTEGER, INTENT(IN) ::  indegree, sources(indegree), outdegree,
    destinations(outdegree)
    INTEGER, INTENT(IN) ::  sourceweights(*), destweights(*)
    TYPE(MPI_Info), INTENT(IN) ::  info
    LOGICAL, INTENT(IN) ::  reorder
    TYPE(MPI_Comm), INTENT(OUT) ::  comm_dist_graph
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Dist_graph_create(comm_old, n, sources, degrees, destinations, weights,
             info, reorder, comm_dist_graph, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm_old
    INTEGER, INTENT(IN) ::  n, sources(n), degrees(n), destinations(*)
    INTEGER, INTENT(IN) ::  weights(*)
    TYPE(MPI_Info), INTENT(IN) ::  info
    LOGICAL, INTENT(IN) ::  reorder
    TYPE(MPI_Comm), INTENT(OUT) ::  comm_dist_graph
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Dist_graph_neighbors(comm, maxindegree, sources, sourceweights,
             maxoutdegree, destinations, destweights, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(IN) ::  maxindegree, maxoutdegree
    INTEGER, INTENT(OUT) ::  sources(maxindegree),
    destinations(maxoutdegree)
    INTEGER ::  sourceweights(*), destweights(*)
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Dist_graph_neighbors_count(comm, indegree, outdegree, weighted, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
```

```
    INTEGER, INTENT(OUT) ::  indegree, outdegree
    LOGICAL, INTENT(OUT) ::  weighted
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Graph_create(comm_old, nnodes, index, edges, reorder, comm_graph,
             ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm_old
    INTEGER, INTENT(IN) ::  nnodes, index(nnodes), edges(*)
    LOGICAL, INTENT(IN) ::  reorder
    TYPE(MPI_Comm), INTENT(OUT) ::  comm_graph
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Graphdims_get(comm, nnodes, nedges, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(OUT) ::  nnodes, nedges
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Graph_get(comm, maxindex, maxedges, index, edges, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(IN) ::  maxindex, maxedges
    INTEGER, INTENT(OUT) ::  index(maxindex), edges(maxedges)
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Graph_map(comm, nnodes, index, edges, newrank, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(IN) ::  nnodes, index(nnodes), edges(*)
    INTEGER, INTENT(OUT) ::  newrank
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Graph_neighbors(comm, rank, maxneighbors, neighbors, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(IN) ::  rank, maxneighbors
    INTEGER, INTENT(OUT) ::  neighbors(maxneighbors)
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Graph_neighbors_count(comm, rank, nneighbors, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(IN) ::  rank
    INTEGER, INTENT(OUT) ::  nneighbors
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Ineighbor_allgather(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_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
```

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

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

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

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

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

MPI_Topo_test(comm, status, ierror)
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    INTEGER, INTENT(OUT) ::  status
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror
```

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)
```

```
    TYPE(MPI_Errhandler), INTENT(INOUT) ::  errhandler
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Error_class(errorcode, errorclass, ierror)
    INTEGER, INTENT(IN) ::  errorcode
    INTEGER, INTENT(OUT) ::  errorclass
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Error_string(errorcode, string, resultlen, ierror)
    INTEGER, INTENT(IN) ::  errorcode
    CHARACTER(LEN=MPI_MAX_ERROR_STRING), INTENT(OUT) ::  string
    INTEGER, INTENT(OUT) ::  resultlen
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_File_call_errhandler(fh, errorcode, ierror)
    TYPE(MPI_File), INTENT(IN) ::  fh
    INTEGER, INTENT(IN) ::  errorcode
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_File_create_errhandler(file_errhandler_fn, errhandler, ierror)
    PROCEDURE(MPI_File_errhandler_function) ::  file_errhandler_fn
    TYPE(MPI_Errhandler), INTENT(OUT) ::  errhandler
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_File_get_errhandler(file, errhandler, ierror)
    TYPE(MPI_File), INTENT(IN) ::  file
    TYPE(MPI_Errhandler), INTENT(OUT) ::  errhandler
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_File_set_errhandler(file, errhandler, ierror)
    TYPE(MPI_File), INTENT(IN) ::  file
    TYPE(MPI_Errhandler), INTENT(IN) ::  errhandler
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

MPI_Finalize(ierror)
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Free_mem(base, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS ::  base
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Get_library_version(version, resultlen, ierror)
    CHARACTER(LEN=MPI_MAX_LIBRARY_VERSION_STRING), INTENT(OUT) ::  version
    INTEGER, INTENT(OUT) ::  resultlen
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Get_processor_name(name, resultlen, ierror)
    CHARACTER(LEN=MPI_MAX_PROCESSOR_NAME), INTENT(OUT) ::  name
    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
```

```
MPI_Info_get(info, key, valuelen, value, flag, ierror)
    TYPE(MPI_Info), INTENT(IN) ::  info
    CHARACTER(LEN=*), INTENT(IN) ::  key
    INTEGER, INTENT(IN) ::  valuelen
    CHARACTER(LEN=valuelen), INTENT(OUT) ::  value
    LOGICAL, INTENT(OUT) ::  flag
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Info_get_nkeys(info, nkeys, ierror)
    TYPE(MPI_Info), INTENT(IN) ::  info
    INTEGER, INTENT(OUT) ::  nkeys
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

MPI_Info_get_valuelen(info, key, valuelen, flag, ierror)
    TYPE(MPI_Info), INTENT(IN) ::  info
    CHARACTER(LEN=*), INTENT(IN) ::  key
    INTEGER, INTENT(OUT) ::  valuelen
    LOGICAL, INTENT(OUT) ::  flag
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

A.3.8 Process Creation and Management Fortran 2008 Bindings

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

MPI_Comm_accept(port_name, info, root, comm, newcomm, ierror)
    CHARACTER(LEN=*), INTENT(IN) ::  port_name
    TYPE(MPI_Info), INTENT(IN) ::  info
    INTEGER, INTENT(IN) ::  root
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    TYPE(MPI_Comm), INTENT(OUT) ::  newcomm
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Comm_connect(port_name, info, root, comm, newcomm, ierror)
    CHARACTER(LEN=*), INTENT(IN) ::  port_name
    TYPE(MPI_Info), INTENT(IN) ::  info
    INTEGER, INTENT(IN) ::  root
    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
```

```
    CHARACTER(LEN=*), INTENT(IN) ::  service_name, port_name
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

A.3.9 One-Sided Communications Fortran 2008 Bindings

```
MPI_Accumulate(origin_addr, origin_count, origin_datatype, target_rank,
            target_disp, target_count, target_datatype, op, 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_Op), INTENT(IN) ::  op
    TYPE(MPI_Win), INTENT(IN) ::  win
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Compare_and_swap(origin_addr, compare_addr, result_addr, datatype,
            target_rank, target_disp, win, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS ::  origin_addr
    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS ::  compare_addr
    TYPE(*), DIMENSION(..), ASYNCHRONOUS ::  result_addr
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER, INTENT(IN) ::  target_rank
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) ::  target_disp
    TYPE(MPI_Win), INTENT(IN) ::  win
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Fetch_and_op(origin_addr, result_addr, datatype, target_rank,
            target_disp, op, win, ierror)
    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS ::  origin_addr
    TYPE(*), DIMENSION(..), ASYNCHRONOUS ::  result_addr
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER, INTENT(IN) ::  target_rank
    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_accumulate(origin_addr, origin_count, origin_datatype, result_addr,
            result_count, result_datatype, target_rank, target_disp,
            target_count, target_datatype, op, win, 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
    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
```

```
    TYPE(MPI_Win), INTENT(IN) ::  win
    TYPE(MPI_Request), INTENT(OUT) ::  request
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Rget(origin_addr, origin_count, origin_datatype, target_rank,
             target_disp, target_count, target_datatype, win, request,
             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
    TYPE(MPI_Request), INTENT(OUT) ::  request
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Rput(origin_addr, origin_count, origin_datatype, target_rank,
             target_disp, target_count, target_datatype, 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_Win), INTENT(IN) ::  win
    TYPE(MPI_Request), INTENT(OUT) ::  request
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Win_allocate_shared(size, disp_unit, info, comm, baseptr, win, ierror)
    USE, INTRINSIC ::  ISO_C_BINDING, ONLY : C_PTR
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) ::  size
    INTEGER, INTENT(IN) ::  disp_unit
    TYPE(MPI_Info), INTENT(IN) ::  info
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    TYPE(C_PTR), INTENT(OUT) ::  baseptr
    TYPE(MPI_Win), INTENT(OUT) ::  win
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Win_allocate(size, disp_unit, info, comm, baseptr, win, ierror)
    USE, INTRINSIC ::  ISO_C_BINDING, ONLY : C_PTR
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) ::  size
    INTEGER, INTENT(IN) ::  disp_unit
    TYPE(MPI_Info), INTENT(IN) ::  info
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    TYPE(C_PTR), INTENT(OUT) ::  baseptr
    TYPE(MPI_Win), INTENT(OUT) ::  win
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Win_attach(win, base, size, ierror)
    TYPE(MPI_Win), INTENT(IN) ::  win
    TYPE(*), DIMENSION(..), ASYNCHRONOUS ::  base
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) ::  size
```

```
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

MPI_Win_create(base, size, disp_unit, info, comm, win, ierror)
    TYPE(*), DIMENSION(..), ASYNCHRONOUS ::  base
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(IN) ::  size
    INTEGER, INTENT(IN) ::  disp_unit
    TYPE(MPI_Info), INTENT(IN) ::  info
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    TYPE(MPI_Win), INTENT(OUT) ::  win
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Win_create_dynamic(info, comm, win, ierror)
    TYPE(MPI_Info), INTENT(IN) ::  info
    TYPE(MPI_Comm), INTENT(IN) ::  comm
    TYPE(MPI_Win), INTENT(OUT) ::  win
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Win_detach(win, base, ierror)
    TYPE(MPI_Win), INTENT(IN) ::  win
    TYPE(*), DIMENSION(..), ASYNCHRONOUS ::  base
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

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

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

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

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

MPI_Win_free(win, ierror)
    TYPE(MPI_Win), INTENT(INOUT) ::  win
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror
```

```
MPI_Win_get_group(win, group, ierror)
    TYPE(MPI_Win), INTENT(IN) ::  win
    TYPE(MPI_Group), INTENT(OUT) ::  group
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Win_get_info(win, info_used, ierror)
    TYPE(MPI_Win), INTENT(IN) ::  win
    TYPE(MPI_Info), INTENT(OUT) ::  info_used
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

MPI_Win_lock(lock_type, rank, assert, win, ierror)
    INTEGER, INTENT(IN) ::  lock_type, rank, assert
    TYPE(MPI_Win), INTENT(IN) ::  win
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Win_post(group, assert, win, ierror)
    TYPE(MPI_Group), INTENT(IN) ::  group
    INTEGER, INTENT(IN) ::  assert
    TYPE(MPI_Win), INTENT(IN) ::  win
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Win_set_info(win, info, ierror)
    TYPE(MPI_Win), INTENT(IN) ::  win
    TYPE(MPI_Info), INTENT(IN) ::  info
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Win_shared_query(win, rank, size, disp_unit, baseptr, ierror)
    USE, INTRINSIC ::  ISO_C_BINDING, ONLY : C_PTR
    TYPE(MPI_Win), INTENT(IN) ::  win
    INTEGER, INTENT(IN) ::  rank
    INTEGER(KIND=MPI_ADDRESS_KIND), INTENT(OUT) ::  size
    INTEGER, INTENT(OUT) ::  disp_unit
    TYPE(C_PTR), INTENT(OUT) ::  baseptr
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Win_start(group, assert, win, ierror)
    TYPE(MPI_Group), INTENT(IN) ::  group
    INTEGER, INTENT(IN) ::  assert
    TYPE(MPI_Win), INTENT(IN) ::  win
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

MPI_Win_test(win, flag, ierror)
    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
```

```
    INTEGER, INTENT(IN) ::  count
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Status_set_elements_x(status, datatype, count, ierror)
    TYPE(MPI_Status), INTENT(INOUT) ::  status
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER(KIND = MPI_COUNT_KIND), INTENT(IN) ::  count
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror
```

A.3.11 I/O Fortran 2008 Bindings

```
MPI_CONVERSION_FN_NULL(userbuf, datatype, count, filebuf, position,
             extra_state, ierror)
    USE, INTRINSIC ::  ISO_C_BINDING, ONLY : C_PTR
    TYPE(C_PTR), VALUE ::  userbuf, filebuf
    TYPE(MPI_Datatype) ::  datatype
    INTEGER ::  count, ierror
    INTEGER(KIND=MPI_OFFSET_KIND) ::  position
    INTEGER(KIND=MPI_ADDRESS_KIND) ::  extra_state

MPI_File_close(fh, ierror)
    TYPE(MPI_File), INTENT(INOUT) ::  fh
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_File_delete(filename, info, ierror)
    CHARACTER(LEN=*), INTENT(IN) ::  filename
    TYPE(MPI_Info), INTENT(IN) ::  info
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_File_get_amode(fh, amode, ierror)
    TYPE(MPI_File), INTENT(IN) ::  fh
    INTEGER, INTENT(OUT) ::  amode
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_File_get_atomicity(fh, flag, ierror)
    TYPE(MPI_File), INTENT(IN) ::  fh
    LOGICAL, INTENT(OUT) ::  flag
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

MPI_File_get_group(fh, group, ierror)
    TYPE(MPI_File), INTENT(IN) ::  fh
    TYPE(MPI_Group), INTENT(OUT) ::  group
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_File_get_info(fh, info_used, ierror)
```

```
    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
```

```
    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(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_shared(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_iwrite_all(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_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(..), 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_at(fh, offset, buf, count, datatype, request, ierror)
    TYPE(MPI_File), INTENT(IN) ::  fh
    INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) ::  offset
    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(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
```

```
    INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) ::  offset
    TYPE(*), DIMENSION(..), ASYNCHRONOUS ::  buf
    INTEGER, INTENT(IN) ::  count
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_File_read_at_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_at_all(fh, offset, buf, count, datatype, status, ierror)
    TYPE(MPI_File), INTENT(IN) ::  fh
    INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) ::  offset
    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(fh, offset, buf, count, datatype, status, ierror)
    TYPE(MPI_File), INTENT(IN) ::  fh
    INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) ::  offset
    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(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_ordered_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_ordered_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_ordered(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_shared(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_seek(fh, offset, whence, ierror)
    TYPE(MPI_File), INTENT(IN) ::  fh
    INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) ::  offset
    INTEGER, INTENT(IN) ::  whence
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

MPI_File_set_atomicity(fh, flag, ierror)
    TYPE(MPI_File), INTENT(IN) ::  fh
    LOGICAL, INTENT(IN) ::  flag
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

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

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

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

```
MPI_File_sync(fh, ierror)
    TYPE(MPI_File), INTENT(IN) ::  fh
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_File_write_all_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_all_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_all(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_at_all_begin(fh, offset, buf, count, datatype, ierror)
    TYPE(MPI_File), INTENT(IN) ::  fh
    INTEGER(KIND=MPI_OFFSET_KIND), INTENT(IN) ::  offset
    TYPE(*), DIMENSION(..), INTENT(IN), ASYNCHRONOUS ::  buf
    INTEGER, INTENT(IN) ::  count
    TYPE(MPI_Datatype), INTENT(IN) ::  datatype
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_File_write_at_all_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_at_all(fh, offset, buf, count, datatype, status, ierror)
    TYPE(MPI_File), INTENT(IN) ::  fh
    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_at(fh, offset, buf, count, datatype, status, ierror)
    TYPE(MPI_File), INTENT(IN) ::  fh
    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
```

```
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror
```

A.3.12 Language Bindings Fortran 2008 Bindings

```
MPI_F_sync_reg(buf)
    TYPE(*), DIMENSION(..), ASYNCHRONOUS ::  buf

MPI_Sizeof(x, size, ierror)
    TYPE(*), DIMENSION(..)  ::  x
    INTEGER, INTENT(OUT) ::  size
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Status_f082f(f08_status, f_status, ierror)
    TYPE(MPI_Status), INTENT(IN) ::  f08_status
    INTEGER, INTENT(OUT) ::  f_status(MPI_STATUS_SIZE)
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Status_f2f08(f_status, f08_status, ierror)
    INTEGER, INTENT(IN) ::  f_status(MPI_STATUS_SIZE)
    TYPE(MPI_Status), INTENT(OUT) ::  f08_status
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_create_f90_complex(p, r, newtype, ierror)
    INTEGER, INTENT(IN) ::  p, r
    TYPE(MPI_Datatype), INTENT(OUT) ::  newtype
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_create_f90_integer(r, newtype, ierror)
    INTEGER, INTENT(IN) ::  r
    TYPE(MPI_Datatype), INTENT(OUT) ::  newtype
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_create_f90_real(p, r, newtype, ierror)
    INTEGER, INTENT(IN) ::  p, r
    TYPE(MPI_Datatype), INTENT(OUT) ::  newtype
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror

MPI_Type_match_size(typeclass, size, datatype, ierror)
    INTEGER, INTENT(IN) ::  typeclass, size
    TYPE(MPI_Datatype), INTENT(OUT) ::  datatype
    INTEGER, OPTIONAL, INTENT(OUT) ::  ierror
```

A.3.13 Tools / Profiling Interface Fortran 2008 Bindings

```
MPI_Pcontrol(level)
    INTEGER, INTENT(IN) ::  level
```