

MPI: A Message-Passing Interface Standard

Version 3.1

Message Passing Interface Forum

September 21, 2012

Annex A.3.4 :
Callbacks without
- BIND(C)
- any INTENT(...)

#345

```

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

```

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

```

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

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

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

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

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

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

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

```

```

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

```

```

MPI_COMM_NULL_DELETE_FN(comm, comm_keyval, attribute_val, extra_state,
    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

```

```

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

```

```

MPI_Group_rank(group, rank, ierror)
    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

```

```

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

```

```

PROCEDURE(MPI_Win_copy_attr_function) :: win_copy_attr_fn
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

```