MPI_Sendrecv

Sends and receives a message

Synopsis

```
#include "mpi.h"
int MPI_Sendrecv( sendbuf, sendcount, sendtype, dest, sendtag,
                  recvbuf, recvcount, recvtype, source, recvtag,
                  comm, status )
void         *sendbuf;
int           sendcount;
MPI_Datatype  sendtype;
int           dest, sendtag;
void         *recvbuf;
int           recvcount;
MPI_Datatype  recvtype;
int           source, recvtag;
MPI_Comm      comm;
MPI_Status   *status;
```

Input Parameters

sendbuf - initial address of send buffer (choice)

sendcount - number of elements in send buffer (integer)

sendtype - type of elements in send buffer (handle)

dest - rank of destination (integer)

sendtag - send tag (integer)

recvcount - number of elements in receive buffer (integer)

recvtype - type of elements in receive buffer (handle)

source - rank of source (integer)

recvtag - receive tag (integer)

comm - communicator (handle)

Output Parameters

recvbuf - initial address of receive buffer (choice)

status - status object (Status). This refers to the receive operation.

Notes for Fortran

All MPI routines in Fortran (except for `MPI_WTIME` and `MPI_WTICK`) have an additional argument `ierr` at the end of the argument list. `ierr` is an integer and has the same meaning as the return value of the routine in C. In Fortran, MPI routines are subroutines, and are invoked with the `call` statement.

All MPI objects (e.g., `MPI_Datatype`, `MPI_Comm`) are of type `INTEGER` in Fortran.

Errors

All MPI routines (except `MPI_Wtime` and `MPI_Wtick`) return an error value; C routines as the value of the function and Fortran routines in the last argument. Before the value is returned, the current MPI error handler is called. By default, this error handler aborts the MPI job. The error handler may be changed with `MPI_Errhandler_set`; the predefined error handler `MPI_ERRORS_RETURN` may be used to cause error values to be returned. Note that MPI does *not* guarentee that an MPI program can continue past an error.

MPI_SUCCESS - No error; MPI routine completed successfully.

MPI_ERR_COMM - Invalid communicator. A common error is to use a null communicator in a call (not even allowed in `MPI_Comm_rank`).

MPI_ERR_COUNT - Invalid count argument. Count arguments must be non-negative; a count of zero is often valid.

MPI_ERR_TYPE - Invalid datatype argument. May be an uncommitted

MPI_Datatype (see `MPI_Type_commit`).

MPI_ERR_TAG - Invalid tag argument. Tags must be non-negative; tags in a receive (`MPI_Recv`, `MPI_Irecv`, `MPI_Sendrecv`, etc.) may also be `MPI_ANY_TAG`. The largest tag value is available through the the attribute `MPI_TAG_UB`.

MPI_ERR_RANK - Invalid source or destination rank. Ranks must be between zero and the size of the communicator minus one; ranks in a receive (`MPI_Recv`, `MPI_Irecv`, `MPI_Sendrecv`, etc.) may also be `MPI_ANY_SOURCE`.

Location:sendrecv.c