

Core MPI C syntax

All routines return an int error code, not shown here, except MPI_Wtime and MPI_Wtick. For more documentation and tutorials, see: [MPI forum](#), [IBM](#), [Argonne](#), [OpenMPI](#), [NCSA](#), [LLNL](#)

Header file, startup, shutdown, timer

```
#include <mpi.h>

MPI_Init( &argc, &argv );
MPI_Comm_size( comm, &size );
MPI_Comm_rank( comm, &rank );
MPI_Finalize();
double time = MPI_Wtime();
```

Send and Receive

```
MPI_Send      ( sendbuf, sendcount, sendtype, dest, sendtag, comm );
MPI_Isend     ( sendbuf, sendcount, sendtype, dest, sendtag, comm, &request );
MPI_Sendrecv  ( sendbuf, sendcount, sendtype, dest, sendtag,
               recvbuf, recvcount, recvtype, source, recvtag, comm, &status );
MPI_Recv      ( recvbuf, recvcount, recvtype, source, recvtag, comm, &status );
MPI_Irecv     ( recvbuf, recvcount, recvtype, source, recvtag, comm, &request );
```

Non-blocking completion

```
MPI_Wait      ( &request, &status );
MPI_Test      ( &request, &flag, &status );
MPI_Waitall   ( count, requests_array, statuses_array );
MPI_Testall   ( count, requests_array, &flag, statuses_array );
MPI_Waitsome  ( count, requests_array, &outcnt, indices_array, statuses_array );
MPI_Testsome  ( count, requests_array, &outcnt, indices_array, statuses_array );
```

Collective

```
MPI_Barrier   ( comm );
MPI_Bcast     ( buf, count, datatype, root, comm );
MPI_Scatter    ( sendbuf, sendcount, sendtype,
               recvbuf, recvcount, recvtype, root, comm );
MPI_Alltoall  ( sendbuf, sendcount, sendtype,
               recvbuf, recvcount, recvtype, comm );
MPI_Reduce    ( sendbuf, recvbuf, count, datatype, op, root, comm );
MPI_Allreduce ( sendbuf, recvbuf, count, datatype, op, comm );
MPI_Reduce_scatter( sendbuf, recvbuf, recvcounts, datatype, op, comm );
```

Datatypes

```
void *buf, *sendbuf, *recvbuf;
int count, outcnt, sendcount, recvcount, recvcounts[N];
int dest, source, root, rank, size, indices_array[N];
int tag, sendtag, recvtag, flag;
MPI_Comm comm, newcomm;
MPI_Request request, requests_array[N];
MPI_Status status, statuses_array[N];
MPI_Datatype datatype, sendtype, recvtype;
MPI_Op op;
```

Constants

// Communicators	// Datatypes, selected	// Reduction operations
MPI_COMM_WORLD	MPI_CHAR	MPI_MAX
MPI_COMM_SELF	MPI_SIGNED_CHAR	MPI_MIN
	MPI_INT	MPI_SUM
// Wildcards	MPI_LONG	MPI_PROD
MPI_ANY_SOURCE	MPI_UNSIGNED_CHAR	MPI_BAND (bitwise and);
MPI_ANY_TAG	MPI_UNSIGNED	MPI_BOR (bitwise or);
	MPI_UNSIGNED_LONG	MPI_BXOR (bitwise xor);
// Options	MPI_FLOAT	MPI_LAND (logical and);
MPI_IN_PLACE	MPI_DOUBLE	MPI_LOR (logical or);
	MPI_BYTE	MPI_LXOR (logical xor);
	MPI_PACKED	MPI_MAXLOC
		MPI_MINLOC

MPI_Status

```
stat.MPI_SOURCE
stat.MPI_TAG
MPI_Get_count ( &status, datatype, &count );
```

Cartesian grid

```
MPI_Dims_create ( nnodes, ndims, dims );
MPI_Cart_create ( comm, ndims, dims, periods, reorder, comm_cart );
MPI_Cart_coords ( comm_cart, rank, maxdims, coords ); // map rank to coords
MPI_Cart_rank ( comm_cart, coords, &rank ); // map coords to rank
MPI_Cart_shift ( comm_cart, direction, disp, source, dest );
```

User-defined Types

```
MPI_Type_vector ( count, blocklength, stride, oldtype, &newtype );
MPI_Type_commit ( datatype );
MPI_Type_free ( datatype );
```

Datatypes, in addition to first page

```
int nnodes, ndims, maxdims, reorder;
int dims[ndims], periods[ndims], coords[ndims];
MPI_Communicator comm_cart;
MPI_Datatype oldtype, newtype;
```



Copyright © 2011 by Mark Gates. Updated September 7, 2011.

Available from <http://web.eecs.utk.edu/~mgates3/>

You may freely copy and modify this document under the [Creative Commons Attribution license](#). When distributing, please include the above link and copyright for this original document.