MPI_COMM_FREE or when a call is made explicitly to MPI_COMM_DELETE_ATTR. comm_delete_attr_fn should be of type MPI_Comm_delete_attr_function.

This function is called by MPI_COMM_FREE, MPI_COMM_DELETE_ATTR, and MPI_COMM_SET_ATTR to do whatever is needed to remove an attribute. The function returns MPI_SUCCESS on success and an error code on failure (in which case MPI_COMM_FREE will fail).

The argument comm_delete_attr_fn may be specified as MPI_COMM_NULL_DELETE_FN from either C, C++, or Fortran. MPI_COMM_NULL_DELETE_FN is a function that does nothing, other than returning MPI_SUCCESS. MPI_COMM_NULL_DELETE_FN replaces MPI_NULL_DELETE_FN, whose use is deprecated.

If an attribute copy function or attribute delete function returns other than MPI_SUCCESS, then the call that caused it to be invoked (for example, MPI_COMM_FREE), is erroneous.

The special key value MPI_KEYVAL_INVALID is never returned by MPI_KEYVAL_CREATE. Therefore, it can be used for static initialization of key values.

Advice to implementors. To be able to use the predefined C functions MPI_COMM_NULL_COPY_FN or MPI_COMM_DUP_FN as comm_copy_attr_fn argument and/or MPI_COMM_NULL_DELETE_FN as the comm_delete_attr_fn argument in a call to the C++ routine MPI::Comm::Create_keyval, this routine may be overloaded with 3 additional routines that accept the C functions as the first, the second, or both input arguments (instead of an argument that matches the C++ prototype). (*End of advice to implementors.*)

Advice to users. If a user wants to write a "wrapper" routine that internally calls MPI::Comm::Create_keyval and comm_copy_attr_fn and/or comm_delete_attr_fn are arguments of this wrapper routine, and if this wrapper routine should be callable with both user-defined C++ copy and delete functions and with the predefined C functions, then the same overloading as described above in the advice to implementors may be necessary. (*End of advice to users.*)

MPI_COMM_FREE_KEYVAL(comm_keyval)

INOUT	comm_keyval	key value (integer)

```
int MPI_Comm_free_keyval(int *comm_keyval)

MPI_COMM_FREE_KEYVAL(COMM_KEYVAL, IERROR)
    INTEGER COMM_KEYVAL, IERROR
```

{`static void MPI::Comm::Free_keyval(int& comm_keyval)` *(binding deprecated, see Section 15.2)* }

Frees an extant attribute key. This function sets the value of keyval to MPI_KEYVAL_INVALID. Note that it is not erroneous to free an attribute key that is in use, because the actual free does not transpire until after all references (in other communicators on the process) to the key have been freed. These references need to be explictly freed by the program, either via calls to MPI_COMM_DELETE_ATTR that free one attribute instance,

Annex A

Language Bindings Summary

In this section we summarize the specific bindings for C, Fortran, and C++. First we present the constants, type definitions, info values and keys. Then we present the routine prototypes separately for each binding. Listings are alphabetical within chapter.

A.1 Defined Values and Handles

A.1.1 Defined Constants

The C and Fortran name is listed in the left column and the C++ name is listed in the middle or right column. Constants with the type `const int` may also be implemented as literal integer constants substituted by the preprocessor.

Return Codes

C type: `const int` (or unnamed `enum`) Fortran type: `INTEGER`	C++ type: `const int` (or unnamed `enum`)
MPI_SUCCESS	MPI::SUCCESS
MPI_ERR_BUFFER	MPI::ERR_BUFFER
MPI_ERR_COUNT	MPI::ERR_COUNT
MPI_ERR_TYPE	MPI::ERR_TYPE
MPI_ERR_TAG	MPI::ERR_TAG
MPI_ERR_COMM	MPI::ERR_COMM
MPI_ERR_RANK	MPI::ERR_RANK
MPI_ERR_REQUEST	MPI::ERR_REQUEST
MPI_ERR_ROOT	MPI::ERR_ROOT
MPI_ERR_GROUP	MPI::ERR_GROUP
MPI_ERR_OP	MPI::ERR_OP
MPI_ERR_TOPOLOGY	MPI::ERR_TOPOLOGY
MPI_ERR_DIMS	MPI::ERR_DIMS
MPI_ERR_ARG	MPI::ERR_ARG
MPI_ERR_UNKNOWN	MPI::ERR_UNKNOWN
MPI_ERR_TRUNCATE	MPI::ERR_TRUNCATE
MPI_ERR_OTHER	MPI::ERR_OTHER
MPI_ERR_INTERN	MPI::ERR_INTERN
MPI_ERR_PENDING	MPI::ERR_PENDING

(Continued on next page)

Return Codes (continued)

MPI_ERR_IN_STATUS	MPI::ERR_IN_STATUS
MPI_ERR_ACCESS	MPI::ERR_ACCESS
MPI_ERR_AMODE	MPI::ERR_AMODE
MPI_ERR_ASSERT	MPI::ERR_ASSERT
MPI_ERR_BAD_FILE	MPI::ERR_BAD_FILE
MPI_ERR_BASE	MPI::ERR_BASE
MPI_ERR_CONVERSION	MPI::ERR_CONVERSION
MPI_ERR_DISP	MPI::ERR_DISP
MPI_ERR_DUP_DATAREP	MPI::ERR_DUP_DATAREP
MPI_ERR_FILE_EXISTS	MPI::ERR_FILE_EXISTS
MPI_ERR_FILE_IN_USE	MPI::ERR_FILE_IN_USE
MPI_ERR_FILE	MPI::ERR_FILE
MPI_ERR_INFO_KEY	MPI::ERR_INFO_VALUE
MPI_ERR_INFO_NOKEY	MPI::ERR_INFO_NOKEY
MPI_ERR_INFO_VALUE	MPI::ERR_INFO_KEY
MPI_ERR_INFO	MPI::ERR_INFO
MPI_ERR_IO	MPI::ERR_IO
MPI_ERR_KEYVAL	MPI::ERR_KEYVAL
MPI_ERR_LOCKTYPE	MPI::ERR_LOCKTYPE
MPI_ERR_NAME	MPI::ERR_NAME
MPI_ERR_NO_MEM	MPI::ERR_NO_MEM
MPI_ERR_NOT_SAME	MPI::ERR_NOT_SAME
MPI_ERR_NO_SPACE	MPI::ERR_NO_SPACE
MPI_ERR_NO_SUCH_FILE	MPI::ERR_NO_SUCH_FILE
MPI_ERR_PORT	MPI::ERR_PORT
MPI_ERR_QUOTA	MPI::ERR_QUOTA
MPI_ERR_READ_ONLY	MPI::ERR_READ_ONLY
MPI_ERR_RMA_CONFLICT	MPI::ERR_RMA_CONFLICT
MPI_ERR_RMA_SYNC	MPI::ERR_RMA_SYNC
MPI_ERR_SERVICE	MPI::ERR_SERVICE
MPI_ERR_SIZE	MPI::ERR_SIZE
MPI_ERR_SPAWN	MPI::ERR_SPAWN
MPI_ERR_UNSUPPORTED_DATAREP	MPI::ERR_UNSUPPORTED_DATAREP
MPI_ERR_UNSUPPORTED_OPERATION	MPI::ERR_UNSUPPORTED_OPERATION
MPI_ERR_WIN	MPI::ERR_WIN
MPI_ERR_LASTCODE	MPI::ERR_LASTCODE

Buffer Address Constants

C type: `void * const` Fortran type: (predefined memory location)	C++ type: `void * const`
MPI_BOTTOM	MPI::BOTTOM
MPI_IN_PLACE	MPI::IN_PLACE

Assorted Constants

C type: `const int` (or unnamed `enum`) Fortran type: `INTEGER`	C++ type: `const int` (or unnamed `enum`)
MPI_PROC_NULL	MPI::PROC_NULL
MPI_ANY_SOURCE	MPI::ANY_SOURCE
MPI_ANY_TAG	MPI::ANY_TAG
MPI_UNDEFINED	MPI::UNDEFINED
MPI_BSEND_OVERHEAD	MPI::BSEND_OVERHEAD
MPI_KEYVAL_INVALID	MPI::KEYVAL_INVALID
MPI_LOCK_EXCLUSIVE	MPI::LOCK_EXCLUSIVE
MPI_LOCK_SHARED	MPI::LOCK_SHARED
MPI_ROOT	MPI::ROOT

Status size and reserved index values (Fortran only)

Fortran type: `INTEGER`	
MPI_STATUS_SIZE	Not defined for C++
MPI_SOURCE	Not defined for C++
MPI_TAG	Not defined for C++
MPI_ERROR	Not defined for C++

Variable Address Size (Fortran only)

Fortran type: `INTEGER`	
MPI_ADDRESS_KIND	Not defined for C++
MPI_INTEGER_KIND	Not defined for C++
MPI_OFFSET_KIND	Not defined for C++

Error-handling specifiers

C type: `MPI_Errhandler` Fortran type: `INTEGER`	C++ type: `MPI::Errhandler`
MPI_ERRORS_ARE_FATAL	MPI::ERRORS_ARE_FATAL
MPI_ERRORS_RETURN	MPI::ERRORS_RETURN
	MPI::ERRORS_THROW_EXCEPTIONS

Maximum Sizes for Strings

C type: `const int` (or unnamed `enum`) Fortran type: `INTEGER`	C++ type: `const int` (or unnamed `enum`)
MPI_MAX_PROCESSOR_NAME	MPI::MAX_PROCESSOR_NAME
MPI_MAX_ERROR_STRING	MPI::MAX_ERROR_STRING
MPI_MAX_DATAREP_STRING	MPI::MAX_DATAREP_STRING
MPI_MAX_INFO_KEY	MPI::MAX_INFO_KEY
MPI_MAX_INFO_VAL	MPI::MAX_INFO_VAL
MPI_MAX_OBJECT_NAME	MPI::MAX_OBJECT_NAME
MPI_MAX_PORT_NAME	MPI::MAX_PORT_NAME

Named Predefined Datatypes		C/C++ types
C type: `MPI_Datatype`	C++ type: `MPI::Datatype`	
Fortran type: `INTEGER`		
MPI_CHAR	MPI::CHAR	`char` (treated as printable character)
MPI_SHORT	MPI::SHORT	`signed short int`
MPI_INT	MPI::INT	`signed int`
MPI_LONG	MPI::LONG	`signed long`
MPI_LONG_LONG_INT	MPI::LONG_LONG_INT	`signed long long`
MPI_LONG_LONG	MPI::LONG_LONG	`long long` (synonym)
MPI_SIGNED_CHAR	MPI::SIGNED_CHAR	`signed char` (treated as integral value)
MPI_UNSIGNED_CHAR	MPI::UNSIGNED_CHAR	`unsigned char` (treated as integral value)
MPI_UNSIGNED_SHORT	MPI::UNSIGNED_SHORT	`unsigned short`
MPI_UNSIGNED	MPI::UNSIGNED	`unsigned int`
MPI_UNSIGNED_LONG	MPI::UNSIGNED_LONG	`unsigned long`
MPI_UNSIGNED_LONG_LONG	MPI::UNSIGNED_LONG_LONG	`unsigned long long`
MPI_FLOAT	MPI::FLOAT	`float`
MPI_DOUBLE	MPI::DOUBLE	`double`
MPI_LONG_DOUBLE	MPI::LONG_DOUBLE	`long double`
MPI_WCHAR	MPI::WCHAR	`wchar_t` (defined in `<stddef.h>`) (treated as printable character)
MPI_C_BOOL	(use C datatype handle)	`_Bool`
MPI_INT8_T	(use C datatype handle)	`int8_t`
MPI_INT16_T	(use C datatype handle)	`int16_t`
MPI_INT32_T	(use C datatype handle)	`int32_t`
MPI_INT64_T	(use C datatype handle)	`int64_t`
MPI_UINT8_T	(use C datatype handle)	`uint8_t`
MPI_UINT16_T	(use C datatype handle)	`uint16_t`
MPI_UINT32_T	(use C datatype handle)	`uint32_t`
MPI_UINT64_T	(use C datatype handle)	`uint64_t`
MPI_AINT	(use C datatype handle)	`MPI_Aint`
MPI_OFFSET	(use C datatype handle)	`MPI_Offset`
MPI_C_COMPLEX	(use C datatype handle)	`float _Complex`
MPI_C_FLOAT_COMPLEX	(use C datatype handle)	`float _Complex`
MPI_C_DOUBLE_COMPLEX	(use C datatype handle)	`double _Complex`
MPI_C_LONG_DOUBLE_COMPLEX	(use C datatype handle)	`long double _Complex`
MPI_BYTE	MPI::BYTE	(any C/C++ type)
MPI_PACKED	MPI::PACKED	(any C/C++ type)

Named Predefined Datatypes		Fortran types
C type: MPI_Datatype	C++ type: MPI::Datatype	
Fortran type: INTEGER		
MPI_INTEGER	MPI::INTEGER	INTEGER
MPI_REAL	MPI::REAL	REAL
MPI_DOUBLE_PRECISION	MPI::DOUBLE_PRECISION	DOUBLE PRECISION
MPI_COMPLEX	MPI::F_COMPLEX	COMPLEX
MPI_LOGICAL	MPI::LOGICAL	LOGICAL
MPI_CHARACTER	MPI::CHARACTER	CHARACTER(1)
MPI_AINT	(use C datatype handle)	INTEGER (KIND=MPI_ADDRESS_KIND)
MPI_OFFSET	(use C datatype handle)	INTEGER (KIND=MPI_OFFSET_KIND)
MPI_BYTE	MPI::BYTE	(any Fortran type)
MPI_PACKED	MPI::PACKED	(any Fortran type)

C++-Only Named Predefined Datatypes	C++ types
C++ type: MPI::Datatype	
MPI::BOOL	bool
MPI::COMPLEX	Complex<float>
MPI::DOUBLE_COMPLEX	Complex<double>
MPI::LONG_DOUBLE_COMPLEX	Complex<long double>

Optional datatypes (Fortran)		Fortran types
C type: MPI_Datatype	C++ type: MPI::Datatype	
Fortran type: INTEGER		
MPI_DOUBLE_COMPLEX	MPI::F_DOUBLE_COMPLEX	DOUBLE COMPLEX
MPI_INTEGER1	MPI::INTEGER1	INTEGER*1
MPI_INTEGER2	MPI::INTEGER2	INTEGER*8
MPI_INTEGER4	MPI::INTEGER4	INTEGER*4
MPI_INTEGER8	MPI::INTEGER8	INTEGER*8
MPI_INTEGER16		INTEGER*16
MPI_REAL2	MPI::REAL2	REAL*2
MPI_REAL4	MPI::REAL4	REAL*4
MPI_REAL8	MPI::REAL8	REAL*8
MPI_REAL16		REAL*16
MPI_COMPLEX4		COMPLEX*4
MPI_COMPLEX8		COMPLEX*8
MPI_COMPLEX16		COMPLEX*16
MPI_COMPLEX32		COMPLEX*32

Datatypes for reduction functions (C and C++)

C type: MPI_Datatype Fortran type: INTEGER	C++ type: MPI::Datatype
MPI_FLOAT_INT	MPI::FLOAT_INT
MPI_DOUBLE_INT	MPI::DOUBLE_INT
MPI_LONG_INT	MPI::LONG_INT
MPI_2INT	MPI::TWOINT
MPI_SHORT_INT	MPI::SHORT_INT
MPI_LONG_DOUBLE_INT	MPI::LONG_DOUBLE_INT

Datatypes for reduction functions (Fortran)

C type: MPI_Datatype Fortran type: INTEGER	C++ type: MPI::Datatype
MPI_2REAL	MPI::TWOREAL
MPI_2DOUBLE_PRECISION	MPI::TWODOUBLE_PRECISION
MPI_2INTEGER	MPI::TWOINTEGER

Special datatypes for constructing derived datatypes

C type: MPI_Datatype Fortran type: INTEGER	C++ type: MPI::Datatype
MPI_UB	MPI::UB
MPI_LB	MPI::LB

Reserved communicators

C type: MPI_Comm Fortran type: INTEGER	C++ type: MPI::Intracomm
MPI_COMM_WORLD	MPI::COMM_WORLD
MPI_COMM_SELF	MPI::COMM_SELF

Results of communicator and group comparisons

C type: const int (or unnamed enum) Fortran type: INTEGER	C++ type: const int (or unnamed enum)
MPI_IDENT	MPI::IDENT
MPI_CONGRUENT	MPI::CONGRUENT
MPI_SIMILAR	MPI::SIMILAR
MPI_UNEQUAL	MPI::UNEQUAL

Environmental inquiry keys

C type: const int (or unnamed enum) Fortran type: INTEGER	C++ type: const int (or unnamed enum)
MPI_TAG_UB	MPI::TAG_UB
MPI_IO	MPI::IO
MPI_HOST	MPI::HOST
MPI_WTIME_IS_GLOBAL	MPI::WTIME_IS_GLOBAL

Collective Operations

C type: MPI_Op Fortran type: INTEGER	C++ type: const MPI::Op
MPI_MAX	MPI::MAX
MPI_MIN	MPI::MIN
MPI_SUM	MPI::SUM
MPI_PROD	MPI::PROD
MPI_MAXLOC	MPI::MAXLOC
MPI_MINLOC	MPI::MINLOC
MPI_BAND	MPI::BAND
MPI_BOR	MPI::BOR
MPI_BXOR	MPI::BXOR
MPI_LAND	MPI::LAND
MPI_LOR	MPI::LOR
MPI_LXOR	MPI::LXOR
MPI_REPLACE	MPI::REPLACE

Null Handles

C/Fortran name C type / Fortran type	C++ name C++ type
MPI_GROUP_NULL MPI_Group / INTEGER	MPI::GROUP_NULL const MPI::Group
MPI_COMM_NULL MPI_Comm / INTEGER	MPI::COMM_NULL [1])
MPI_DATATYPE_NULL MPI_Datatype / INTEGER	MPI::DATATYPE_NULL const MPI::Datatype
MPI_REQUEST_NULL MPI_Request / INTEGER	MPI::REQUEST_NULL const MPI::Request
MPI_OP_NULL MPI_Op / INTEGER	MPI::OP_NULL const MPI::Op
MPI_ERRHANDLER_NULL MPI_Errhandler / INTEGER	MPI::ERRHANDLER_NULL const MPI::Errhandler
MPI_FILE_NULL MPI_File / INTEGER	MPI::FILE_NULL
MPI_INFO_NULL MPI_Info / INTEGER	MPI::INFO_NULL const MPI::Info
MPI_WIN_NULL MPI_Win / INTEGER	MPI::WIN_NULL

[1]) C++ type: See Section 16.1.7 on page 474 regarding class hierarchy and the specific type of MPI::COMM_NULL

Empty group

C type: MPI_Group Fortran type: INTEGER	C++ type: const MPI::Group
MPI_GROUP_EMPTY	MPI::GROUP_EMPTY

Topologies

C type: `const int` (or unnamed `enum`) Fortran type: `INTEGER`	C++ type: `const int` (or unnamed `enum`)
MPI_GRAPH	MPI::GRAPH
MPI_DIST_GRAPH	MPI::DIST_GRAPH
MPI_CART	MPI::CART

Predefined functions

C/Fortran name C type / Fortran type	C++ name C++ type
MPI_COMM_NULL_COPY_FN `MPI_Comm_copy_attr_function` / `COMM_COPY_ATTR_FN`	MPI_COMM_NULL_COPY_FN same as in C [1])
MPI_COMM_DUP_FN `MPI_Comm_copy_attr_function` / `COMM_COPY_ATTR_FN`	MPI_COMM_DUP_FN same as in C [1])
MPI_COMM_NULL_DELETE_FN `MPI_Comm_delete_attr_function` / `COMM_DELETE_ATTR_FN`	MPI_COMM_NULL_DELETE_FN same as in C [1])
MPI_WIN_NULL_COPY_FN `MPI_Win_copy_attr_function` / `WIN_COPY_ATTR_FN`	MPI_WIN_NULL_COPY_FN same as in C [1])
MPI_WIN_DUP_FN `MPI_Win_copy_attr_function` / `WIN_COPY_ATTR_FN`	MPI_WIN_DUP_FN same as in C [1])
MPI_WIN_NULL_DELETE_FN `MPI_Win_delete_attr_function` / `WIN_DELETE_ATTR_FN`	MPI_WIN_NULL_DELETE_FN same as in C [1])
MPI_TYPE_NULL_COPY_FN `MPI_Type_copy_attr_function` / `TYPE_COPY_ATTR_FN`	MPI_TYPE_NULL_COPY_FN same as in C [1])
MPI_TYPE_DUP_FN `MPI_Type_copy_attr_function` / `TYPE_COPY_ATTR_FN`	MPI_TYPE_DUP_FN same as in C [1])
MPI_TYPE_NULL_DELETE_FN `MPI_Type_delete_attr_function` / `TYPE_DELETE_ATTR_FN`	MPI_TYPE_NULL_DELETE_FN same as in C [1])

[1] See the advice to implementors on MPI_COMM_NULL_COPY_FN, ... in Section 6.7.2 on page 226

Deprecated predefined functions

C/Fortran name C type / Fortran type	C++ name C++ type
MPI_NULL_COPY_FN MPI_Copy_function / COPY_FUNCTION	MPI::NULL_COPY_FN MPI::Copy_function
MPI_DUP_FN MPI_Copy_function / COPY_FUNCTION	MPI::DUP_FN MPI::Copy_function
MPI_NULL_DELETE_FN MPI_Delete_function / DELETE_FUNCTION	MPI::NULL_DELETE_FN MPI::Delete_function

Predefined Attribute Keys

C type: const int (or unnamed enum) Fortran type: INTEGER	C++ type: const int (or unnamed enum)
MPI_APPNUM	MPI::APPNUM
MPI_LASTUSEDCODE	MPI::LASTUSEDCODE
MPI_UNIVERSE_SIZE	MPI::UNIVERSE_SIZE
MPI_WIN_BASE	MPI::WIN_BASE
MPI_WIN_DISP_UNIT	MPI::WIN_DISP_UNIT
MPI_WIN_SIZE	MPI::WIN_SIZE

Mode Constants

C type: const int (or unnamed enum) Fortran type: INTEGER	C++ type: const int (or unnamed enum)
MPI_MODE_APPEND	MPI::MODE_APPEND
MPI_MODE_CREATE	MPI::MODE_CREATE
MPI_MODE_DELETE_ON_CLOSE	MPI::MODE_DELETE_ON_CLOSE
MPI_MODE_EXCL	MPI::MODE_EXCL
MPI_MODE_NOCHECK	MPI::MODE_NOCHECK
MPI_MODE_NOPRECEDE	MPI::MODE_NOPRECEDE
MPI_MODE_NOPUT	MPI::MODE_NOPUT
MPI_MODE_NOSTORE	MPI::MODE_NOSTORE
MPI_MODE_NOSUCCEED	MPI::MODE_NOSUCCEED
MPI_MODE_RDONLY	MPI::MODE_RDONLY
MPI_MODE_RDWR	MPI::MODE_RDWR
MPI_MODE_SEQUENTIAL	MPI::MODE_SEQUENTIAL
MPI_MODE_UNIQUE_OPEN	MPI::MODE_UNIQUE_OPEN
MPI_MODE_WRONLY	MPI::MODE_WRONLY

Datatype Decoding Constants

C type: `const int` (or unnamed `enum`) Fortran type: `INTEGER`	C++ type: `const int` (or unnamed `enum`)
MPI_COMBINER_CONTIGUOUS	MPI::COMBINER_CONTIGUOUS
MPI_COMBINER_DARRAY	MPI::COMBINER_DARRAY
MPI_COMBINER_DUP	MPI::COMBINER_DUP
MPI_COMBINER_F90_COMPLEX	MPI::COMBINER_F90_COMPLEX
MPI_COMBINER_F90_INTEGER	MPI::COMBINER_F90_INTEGER
MPI_COMBINER_F90_REAL	MPI::COMBINER_F90_REAL
MPI_COMBINER_HINDEXED_INTEGER	MPI::COMBINER_HINDEXED_INTEGER
MPI_COMBINER_HINDEXED	MPI::COMBINER_HINDEXED
MPI_COMBINER_HVECTOR_INTEGER	MPI::COMBINER_HVECTOR_INTEGER
MPI_COMBINER_HVECTOR	MPI::COMBINER_HVECTOR
MPI_COMBINER_INDEXED_BLOCK	MPI::COMBINER_INDEXED_BLOCK
MPI_COMBINER_INDEXED	MPI::COMBINER_INDEXED
MPI_COMBINER_NAMED	MPI::COMBINER_NAMED
MPI_COMBINER_RESIZED	MPI::COMBINER_RESIZED
MPI_COMBINER_STRUCT_INTEGER	MPI::COMBINER_STRUCT_INTEGER
MPI_COMBINER_STRUCT	MPI::COMBINER_STRUCT
MPI_COMBINER_SUBARRAY	MPI::COMBINER_SUBARRAY
MPI_COMBINER_VECTOR	MPI::COMBINER_VECTOR

Threads Constants

C type: `const int` (or unnamed `enum`) Fortran type: `INTEGER`	C++ type: `const int` (or unnamed `enum`)
MPI_THREAD_FUNNELED	MPI::THREAD_FUNNELED
MPI_THREAD_MULTIPLE	MPI::THREAD_MULTIPLE
MPI_THREAD_SERIALIZED	MPI::THREAD_SERIALIZED
MPI_THREAD_SINGLE	MPI::THREAD_SINGLE

File Operation Constants, Part 1

C type: `const MPI_Offset` (or unnamed `enum`) Fortran type: `INTEGER (KIND=MPI_OFFSET_KIND)`	C++ type: `const MPI::Offset` (or unnamed `enum`)
MPI_DISPLACEMENT_CURRENT	MPI::DISPLACEMENT_CURRENT

File Operation Constants, Part 2

C type: `const int` (or unnamed `enum`) Fortran type: `INTEGER`	C++ type: `const int` (or unnamed `enum`)
MPI_DISTRIBUTE_BLOCK	MPI::DISTRIBUTE_BLOCK
MPI_DISTRIBUTE_CYCLIC	MPI::DISTRIBUTE_CYCLIC
MPI_DISTRIBUTE_DFLT_DARG	MPI::DISTRIBUTE_DFLT_DARG
MPI_DISTRIBUTE_NONE	MPI::DISTRIBUTE_NONE
MPI_ORDER_C	MPI::ORDER_C
MPI_ORDER_FORTRAN	MPI::ORDER_FORTRAN
MPI_SEEK_CUR	MPI::SEEK_CUR
MPI_SEEK_END	MPI::SEEK_END
MPI_SEEK_SET	MPI::SEEK_SET

F90 Datatype Matching Constants

C type: `const int` (or unnamed `enum`) Fortran type: `INTEGER`	C++ type: `const int` (or unnamed `enum`)
MPI_TYPECLASS_COMPLEX	MPI::TYPECLASS_COMPLEX
MPI_TYPECLASS_INTEGER	MPI::TYPECLASS_INTEGER
MPI_TYPECLASS_REAL	MPI::TYPECLASS_REAL

Constants Specifying Empty or Ignored Input

C/Fortran name C type / Fortran type	C++ name C++ type
MPI_ARGVS_NULL `char***` / 2-dim. array of `CHARACTER*(*)`	MPI::ARGVS_NULL `const char ***`
MPI_ARGV_NULL `char**` / array of `CHARACTER*(*)`	MPI::ARGV_NULL `const char **`
MPI_ERRCODES_IGNORE `int*` / `INTEGER` array	Not defined for C++
MPI_STATUSES_IGNORE `MPI_Status*` / `INTEGER, DIMENSION(MPI_STATUS_SIZE,*)`	Not defined for C++
MPI_STATUS_IGNORE `MPI_Status*` / `INTEGER, DIMENSION(MPI_STATUS_SIZE)`	Not defined for C++
MPI_UNWEIGHTED	Not defined for C++

C Constants Specifying Ignored Input (no C++ or Fortran)

C type: `MPI_Fint*`
MPI_F_STATUSES_IGNORE
MPI_F_STATUS_IGNORE

C and C++ preprocessor Constants and Fortran Parameters

C/C++ type: `const int` (or unnamed `enum`) Fortran type: `INTEGER`
MPI_SUBVERSION
MPI_VERSION

MPI Constant and Predefined Handle Index

This index lists predefined MPI constants and handles.

MPI::*_NULL, 469
MPI::_LONG_LONG, 471
MPI::ANY_SOURCE, 514
MPI::ANY_TAG, 514
MPI::APPNUM, 520
MPI::ARGV_NULL, 522
MPI::ARGVS_NULL, 522
MPI::BAND, 518
MPI::BOOL, 472, 474, 516
MPI::BOR, 518
MPI::BOTTOM, 513
MPI::BSEND_OVERHEAD, 514
MPI::BXOR, 518
MPI::BYTE, 471, 472, 474, 515, 516
MPI::CART, 519
MPI::CHAR, 472, 515
MPI::CHARACTER, 472, 516
MPI::COMBINER_CONTIGUOUS, 521
MPI::COMBINER_DARRAY, 521
MPI::COMBINER_DUP, 521
MPI::COMBINER_F90_COMPLEX, 521
MPI::COMBINER_F90_INTEGER, 521
MPI::COMBINER_F90_REAL, 521
MPI::COMBINER_HINDEXED, 521
MPI::COMBINER_HINDEXED_INTEGER, 521
MPI::COMBINER_HVECTOR, 521
MPI::COMBINER_HVECTOR_INTEGER, 521
MPI::COMBINER_INDEXED, 521
MPI::COMBINER_INDEXED_BLOCK, 521
MPI::COMBINER_NAMED, 521
MPI::COMBINER_RESIZED, 521
MPI::COMBINER_STRUCT, 521
MPI::COMBINER_STRUCT_INTEGER, 521
MPI::COMBINER_SUBARRAY, 521
MPI::COMBINER_VECTOR, 521
MPI::COMM_NULL, 471, 475, 518
MPI::COMM_SELF, 517
MPI::COMM_WORLD, 517
MPI::COMPLEX, 472, 474, 516
MPI::CONGRUENT, 517
MPI::DATATYPE_NULL, 518
MPI::DISPLACEMENT_CURRENT, 521
MPI::DIST_GRAPH, 519
MPI::DISTRIBUTE_BLOCK, 522
MPI::DISTRIBUTE_CYCLIC, 522
MPI::DISTRIBUTE_DFLT_DARG, 522
MPI::DISTRIBUTE_NONE, 522
MPI::DOUBLE, 472, 474, 515
MPI::DOUBLE_COMPLEX, 472, 474, 516
MPI::DOUBLE_INT, 473, 517
MPI::DOUBLE_PRECISION, 472, 474, 516
MPI::DUP_FN, 520
MPI::ERR_ARG, 512
MPI::ERR_BUFFER, 512
MPI::ERR_COMM, 512
MPI::ERR_COUNT, 512
MPI::ERR_DIMS, 512
MPI::ERR_GROUP, 512
MPI::ERR_IN_STATUS, 513
MPI::ERR_INTERN, 512
MPI::ERR_LASTCODE, 513
MPI::ERR_OP, 512
MPI::ERR_OTHER, 512
MPI::ERR_PENDING, 512
MPI::ERR_RANK, 512
MPI::ERR_REQUEST, 512

MPI::ERR_ROOT, 512
MPI::ERR_TAG, 512
MPI::ERR_TOPOLOGY, 512
MPI::ERR_TRUNCATE, 512
MPI::ERR_TYPE, 512
MPI::ERR_UNKNOWN, 512
MPI::ERRHANDLER_NULL, 518
MPI::ERRORS_ARE_FATAL, 19, 514
MPI::ERRORS_RETURN, 19, 514
MPI::ERRORS_THROW_EXCEPTIONS, 19, 23, 277, 514
MPI::F_COMPLEX, 472, 474, 516
MPI::F_COMPLEX16, 473, 474
MPI::F_COMPLEX32, 473, 474
MPI::F_COMPLEX4, 473, 474
MPI::F_COMPLEX8, 473, 474
MPI::F_DOUBLE_COMPLEX, 473, 474, 516
MPI::FILE_NULL, 518
MPI::FLOAT, 472, 474, 515
MPI::FLOAT_INT, 473, 517
MPI::GRAPH, 519
MPI::GROUP_EMPTY, 518
MPI::GROUP_NULL, 518
MPI::HOST, 517
MPI::IDENT, 517
MPI::IN_PLACE, 513
MPI::INFO_NULL, 518
MPI::INT, 471, 472, 515
MPI::INTEGER, 471, 472, 516
MPI::INTEGER1, 473, 474, 516
MPI::INTEGER16, 473, 474
MPI::INTEGER2, 473, 474, 516
MPI::INTEGER4, 473, 474, 516
MPI::INTEGER8, 473, 474, 516
MPI::IO, 517
MPI::KEYVAL_INVALID, 514
MPI::LAND, 518
MPI::LASTUSEDCODE, 520
MPI::LB, 517
MPI::LOCK_EXCLUSIVE, 514
MPI::LOCK_SHARED, 514
MPI::LOGICAL, 472, 474, 516
MPI::LONG, 471, 472, 515
MPI::LONG_DOUBLE, 472, 474, 515
MPI::LONG_DOUBLE_COMPLEX, 472, 474, 516
MPI::LONG_DOUBLE_INT, 473, 517
MPI::LONG_INT, 473, 517
MPI::LONG_LONG, 472, 515
MPI::LONG_LONG_INT, 515
MPI::LOR, 518
MPI::LXOR, 518
MPI::MAX, 518
MPI::MAX_DATAREP_STRING, 514
MPI::MAX_ERROR_STRING, 514
MPI::MAX_INFO_KEY, 514
MPI::MAX_INFO_VAL, 514
MPI::MAX_OBJECT_NAME, 514
MPI::MAX_PORT_NAME, 514
MPI::MAX_PROCESSOR_NAME, 514
MPI::MAXLOC, 474, 518
MPI::MIN, 518
MPI::MINLOC, 474, 518
MPI::MODE_APPEND, 520
MPI::MODE_CREATE, 520
MPI::MODE_DELETE_ON_CLOSE, 520
MPI::MODE_EXCL, 520
MPI::MODE_NOCHECK, 520
MPI::MODE_NOPRECEDE, 520
MPI::MODE_NOPUT, 520
MPI::MODE_NOSTORE, 520
MPI::MODE_NOSUCCEED, 520
MPI::MODE_RDONLY, 520
MPI::MODE_RDWR, 520
MPI::MODE_SEQUENTIAL, 520
MPI::MODE_UNIQUE_OPEN, 520
MPI::MODE_WRONLY, 520
MPI::NULL_COPY_FN, 520
MPI::NULL_DELETE_FN, 520
MPI::OP_NULL, 518
MPI::ORDER_C, 522
MPI::ORDER_FORTRAN, 522
MPI::PACKED, 471, 472, 515, 516
MPI::PROC_NULL, 514
MPI::PROD, 518
MPI::REAL, 472, 474, 516
MPI::REAL16, 473, 474
MPI::REAL2, 473, 474, 516
MPI::REAL4, 473, 474, 516
MPI::REAL8, 473, 474, 516
MPI::REPLACE, 518
MPI::REQUEST_NULL, 518
MPI::ROOT, 514
MPI::SEEK_CUR, 522
MPI::SEEK_END, 522

MPI::SEEK_SET, 522
MPI::SHORT, 471, 472, 515
MPI::SHORT_INT, 473, 517
MPI::SIGNED_CHAR, 471, 472, 515
MPI::SIMILAR, 517
MPI::SUCCESS, 512
MPI::SUM, 518
MPI::TAG_UB, 517
MPI::THREAD_FUNNELED, 521
MPI::THREAD_MULTIPLE, 521
MPI::THREAD_SERIALIZED, 521
MPI::THREAD_SINGLE, 521
MPI::TWODOUBLE_PRECISION, 473, 517
MPI::TWOINT, 473, 517
MPI::TWOINTEGER, 473, 517
MPI::TWOREAL, 473, 517
MPI::TYPECLASS_COMPLEX, 522
MPI::TYPECLASS_INTEGER, 522
MPI::TYPECLASS_REAL, 522
MPI::UB, 517
MPI::UNDEFINED, 514
MPI::UNEQUAL, 517
MPI::UNIVERSE_SIZE, 520
MPI::UNSIGNED, 471, 472, 515
MPI::UNSIGNED_CHAR, 471, 472, 515
MPI::UNSIGNED_LONG, 471, 472, 515
MPI::UNSIGNED_LONG_LONG, 471, 472, 515
MPI::UNSIGNED_SHORT, 471, 472, 515
MPI::WCHAR, 472, 515
MPI::WIN_BASE, 520
MPI::WIN_DISP_UNIT, 520
MPI::WIN_NULL, 518
MPI::WIN_SIZE, 520
MPI::WTIME_IS_GLOBAL, 517
MPI_2DOUBLE_PRECISION, 168, 169, 517
MPI_2INT, 168, 169, 517
MPI_2INTEGER, 168, 169, 517
MPI_2REAL, 168, 169, 517
MPI_ADDRESS_KIND, 15, 15, 106, 481, 505, 506, 514
MPI_AINT, 29, 165, 515, 516, 592–595
MPI_ANY_SOURCE, 30, 31, 42, 52, 53, 65–67, 71, 74, 75, 243, 273, 514
MPI_ANY_TAG, 14, 30, 31, 33, 52, 53, 65–67, 71, 74–76, 514
MPI_APPNUM, 329, 330, 520
MPI_ARGV_NULL, 15, 310, 311, 481, 522
MPI_ARGVS_NULL, 15, 314, 481, 522
MPI_BAND, 165, 166, 518
MPI_BOR, 165, 166, 518
MPI_BOTTOM, 10, 15, 16, 34, 94, 104, 134, 254, 255, 312, 481, 484, 486, 489, 503, 504, 510, 513, 598
MPI_BSEND_OVERHEAD, 48, 274, 514
MPI_BXOR, 165, 166, 518
MPI_BYTE, 27, 28, 35–37, 127, 166, 390, 429, 430, 441, 471, 510, 515, 516, 595
MPI_C_BOOL, 28, 165, 515, 592–595
MPI_C_COMPLEX, 28, 515, 592–595
MPI_C_DOUBLE_COMPLEX, 28, 166, 515, 592–595
MPI_C_FLOAT_COMPLEX, 28, 165, 515, 592–595
MPI_C_LONG_DOUBLE_COMPLEX, 28, 166, 515, 592–595
MPI_CART, 258, 519
MPI_CHAR, 28, 38, 86, 167, 515, 592
MPI_CHARACTER, 27, 36–38, 167, 516
MPI_COMBINER_CONTIGUOUS, 106, 110, 521
MPI_COMBINER_DARRAY, 106, 111, 521
MPI_COMBINER_DUP, 106, 109, 521
MPI_COMBINER_F90_COMPLEX, 106, 111, 521
MPI_COMBINER_F90_INTEGER, 106, 111, 521
MPI_COMBINER_F90_REAL, 106, 111, 521
MPI_COMBINER_HINDEXED, 106, 110, 521
MPI_COMBINER_HINDEXED_INTEGER, 106, 110, 521
MPI_COMBINER_HVECTOR, 106, 110, 521
MPI_COMBINER_HVECTOR_INTEGER, 106, 110, 521
MPI_COMBINER_INDEXED, 106, 110, 521
MPI_COMBINER_INDEXED_BLOCK, 106, 110, 521
MPI_COMBINER_NAMED, 106, 109, 521
MPI_COMBINER_RESIZED, 106, 111, 521
MPI_COMBINER_STRUCT, 106, 110, 521

MPI_COMBINER_STRUCT_INTEGER, 106, 110, 521
MPI_COMBINER_SUBARRAY, 106, 111, 521
MPI_COMBINER_VECTOR, 106, 110, 521
MPI_COMM_NULL, 191, 203–206, 208, 240, 248, 250, 312, 331, 332, 518, 596
MPI_COMM_PARENT, 240
MPI_COMM_SELF, 191, 224, 240, 295, 296, 331, 391, 517, 594
MPI_COMM_WORLD, 14, 24, 29, 191–193, 199, 200, 211, 219, 220, 240, 248, 272, 273, 276, 279, 287, 294, 295, 297, 305, 306, 308, 309, 313–315, 328–331, 385, 428, 447, 498, 509, 517, 597
MPI_COMPLEX, 27, 165, 432, 490, 516
MPI_COMPLEX16, 166, 516
MPI_COMPLEX32, 166, 516
MPI_COMPLEX4, 166, 516
MPI_COMPLEX8, 166, 516
MPI_CONGRUENT, 200, 218, 517
MPI_CONVERSION_FN_NULL, 436
MPI_DATATYPE, 19
MPI_DATATYPE_NULL, 100, 518
MPI_DISPLACEMENT_CURRENT, 402, 521, 598
MPI_DIST_GRAPH, 258, 519, 594
MPI_DISTRIBUTE_BLOCK, 91, 522
MPI_DISTRIBUTE_CYCLIC, 91, 522
MPI_DISTRIBUTE_DFLT_DARG, 91, 522
MPI_DISTRIBUTE_NONE, 91, 522
MPI_DOUBLE, 28, 165, 489, 515
MPI_DOUBLE_COMPLEX, 27, 166, 432, 490, 516
MPI_DOUBLE_INT, 168, 169, 517
MPI_DOUBLE_PRECISION, 27, 165, 490, 516
MPI_DUP_FN, 520
MPI_ERR_ACCESS, 285, 394, 448
MPI_ERR_AMODE, 285, 393, 448
MPI_ERR_ARG, 284, 512
MPI_ERR_ASSERT, 284, 363
MPI_ERR_BAD_FILE, 285, 448
MPI_ERR_BASE, 275, 284, 363
MPI_ERR_BUFFER, 284, 512
MPI_ERR_COMM, 284, 512
MPI_ERR_CONVERSION, 285, 437, 448
MPI_ERR_COUNT, 284, 512
MPI_ERR_DIMS, 284, 512
MPI_ERR_DISP, 284, 363
MPI_ERR_DUP_DATAREP, 285, 434, 448
MPI_ERR_FILE, 285, 448
MPI_ERR_FILE_EXISTS, 285, 448
MPI_ERR_FILE_IN_USE, 285, 394, 448
MPI_ERR_GROUP, 284, 512
MPI_ERR_IN_STATUS, 32, 34, 53, 60, 62, 278, 284, 376, 406, 477, 513
MPI_ERR_INFO, 284
MPI_ERR_INFO_KEY, 284, 300
MPI_ERR_INFO_NOKEY, 284, 301
MPI_ERR_INFO_VALUE, 284, 300
MPI_ERR_INTERN, 284, 512
MPI_ERR_IO, 285, 448
MPI_ERR_KEYVAL, 236, 284
MPI_ERR_LASTCODE, 283, 285, 287, 288, 513
MPI_ERR_LOCKTYPE, 284, 363
MPI_ERR_NAME, 284, 325
MPI_ERR_NO_MEM, 275, 284
MPI_ERR_NO_SPACE, 285, 448
MPI_ERR_NO_SUCH_FILE, 285, 394, 448
MPI_ERR_NOT_SAME, 285, 448
MPI_ERR_OP, 284, 512
MPI_ERR_OTHER, 283, 284, 512
MPI_ERR_PENDING, 60, 284, 512
MPI_ERR_PORT, 284, 322
MPI_ERR_QUOTA, 285, 448
MPI_ERR_RANK, 284, 512
MPI_ERR_READ_ONLY, 285, 448
MPI_ERR_REQUEST, 284, 512
MPI_ERR_RMA_CONFLICT, 284, 363
MPI_ERR_RMA_SYNC, 284, 363
MPI_ERR_ROOT, 284, 512
MPI_ERR_SERVICE, 284, 324
MPI_ERR_SIZE, 284, 363
MPI_ERR_SPAWN, 284, 311, 312
MPI_ERR_TAG, 284, 512
MPI_ERR_TOPOLOGY, 284, 512
MPI_ERR_TRUNCATE, 284, 512
MPI_ERR_TYPE, 284, 512
MPI_ERR_UNKNOWN, 283, 284, 512
MPI_ERR_UNSUPPORTED_DATAREP, 285, 448

MPI_ERR_UNSUPPORTED_OPERATION, 285, 448
MPI_ERR_WIN, 284, 363
MPI_ERRCODES_IGNORE, 15, 312, 481, 522
MPI_ERRHANDLER_NULL, 282, 518
MPI_ERROR, 32, 53, 514
MPI_ERROR_STRING, 283
MPI_ERRORS_ARE_FATAL, 276, 277, 288, 289, 363, 447, 514
MPI_ERRORS_RETURN, 276, 277, 289, 447, 509, 514
MPI_F_STATUS_IGNORE, 502, 522
MPI_F_STATUSES_IGNORE, 502, 522
MPI_FILE_NULL, 394, 447, 518
MPI_FLOAT, 28, 86, 163, 165, 431, 515
MPI_FLOAT_INT, 12, 168, 169, 517
MPI_GRAPH, 258, 519
MPI_GROUP_EMPTY, 190, 195, 196, 202–204, 518
MPI_GROUP_NULL, 190, 198, 518
MPI_HOST, 272, 517
MPI_IDENT, 193, 200, 517
MPI_IN_PLACE, 15, 134, 160, 481, 489, 513
MPI_INFO_NULL, 256, 303, 311, 321, 393, 394, 403, 518
MPI_INT, 12, 28, 78, 165, 431, 432, 489, 509, 511, 515
MPI_INT16_T, 28, 165, 515, 592–595
MPI_INT32_T, 28, 165, 515, 592–595
MPI_INT64_T, 28, 165, 515, 592–595
MPI_INT8_T, 28, 165, 515, 592–595
MPI_INTEGER, 27, 35, 165, 489, 490, 511, 516
MPI_INTEGER1, 27, 165, 516
MPI_INTEGER16, 165, 516
MPI_INTEGER2, 27, 165, 431, 516
MPI_INTEGER4, 27, 165, 516
MPI_INTEGER8, 165, 494, 516
MPI_INTEGER_KIND, 15, 106, 505, 514
MPI_IO, 272, 273, 517
MPI_KEYVAL_INVALID, 228–230, 514
MPI_LAND, 165, 166, 518
MPI_LASTUSEDCODE, 287, 520
MPI_LB, 16, 17, 89, 92, 96–98, 101, 430, 517
MPI_LOCK_EXCLUSIVE, 356, 514
MPI_LOCK_SHARED, 356, 514
MPI_LOGICAL, 27, 165, 516
MPI_LONG, 28, 165, 515
MPI_LONG_DOUBLE, 28, 165, 515
MPI_LONG_DOUBLE_INT, 168, 517
MPI_LONG_INT, 168, 169, 517
MPI_LONG_LONG, 28, 165, 515, 595
MPI_LONG_LONG_INT, 28, 165, 515, 595
MPI_LOR, 165, 166, 518
MPI_LXOR, 165, 166, 518
MPI_MAX, 163, 165, 166, 181, 518
MPI_MAX_DATAREP_STRING, 15, 404, 434, 514
MPI_MAX_ERROR_STRING, 15, 283, 288, 514
MPI_MAX_INFO_KEY, 15, 284, 299, 301, 302, 514
MPI_MAX_INFO_VAL, 15, 284, 299, 514
MPI_MAX_OBJECT_NAME, 15, 239, 240, 514, 596
MPI_MAX_PORT_NAME, 15, 320, 514
MPI_MAX_PROCESSOR_NAME, 15, 274, 514, 597
MPI_MAXLOC, 165, 167, 168, 171, 518
MPI_MIN, 165, 166, 518
MPI_MINLOC, 165, 167, 168, 171, 518
MPI_MODE_APPEND, 392, 393, 520
MPI_MODE_CREATE, 392, 393, 400, 520
MPI_MODE_DELETE_ON_CLOSE, 392–394, 520
MPI_MODE_EXCL, 392, 393, 520
MPI_MODE_NOCHECK, 359, 360, 520
MPI_MODE_NOPRECEDE, 359, 520
MPI_MODE_NOPUT, 359, 520
MPI_MODE_NOSTORE, 359, 520
MPI_MODE_NOSUCCEED, 359, 520
MPI_MODE_RDONLY, 392, 393, 398, 520
MPI_MODE_RDWR, 392, 393, 520
MPI_MODE_SEQUENTIAL, 392, 393, 395, 396, 402, 407, 410, 420, 440, 520, 598
MPI_MODE_UNIQUE_OPEN, 392, 393, 520
MPI_MODE_WRONLY, 392, 393, 520
MPI_NULL_COPY_FN, 520
MPI_NULL_DELETE_FN, 520
MPI_OFFSET, 29, 165, 515, 516, 592–595

MPI_OFFSET_KIND, 15, 16, 442, 481, 514
MPI_OP_NULL, 174, 518
MPI_ORDER_C, 14, 88, 91, 92, 522
MPI_ORDER_FORTRAN, 14, 88, 91, 522
MPI_PACKED, 27, 28, 35, 122, 123, 127, 432, 471, 510, 515, 516
MPI_PROC_NULL, 26, 75, 76, 137, 138, 140, 142, 150, 151, 164, 193, 265, 272, 273, 339, 514, 595, 597, 598
MPI_PROD, 165, 166, 518
MPI_REAL, 27, 35, 165, 432, 489, 490, 496, 516
MPI_REAL16, 165, 516
MPI_REAL2, 27, 165, 516
MPI_REAL4, 27, 165, 489, 494, 516
MPI_REAL8, 27, 165, 489, 516, 593
MPI_REPLACE, 346, 518, 594, 598
MPI_REQUEST_NULL, 53–55, 58–61, 376, 518
MPI_ROOT, 137, 514
MPI_SEEK_CUR, 415, 421, 522
MPI_SEEK_END, 415, 421, 522
MPI_SEEK_SET, 415, 416, 421, 522
MPI_SHORT, 28, 165, 515
MPI_SHORT_INT, 168, 517
MPI_SIGNED_CHAR, 28, 165, 167, 515, 595
MPI_SIMILAR, 193, 200, 218, 517
MPI_SOURCE, 32, 514
MPI_STATUS, 21, 33, 34, 53
MPI_STATUS_IGNORE, 10, 15, 33, 34, 375, 406, 481, 489, 502, 510, 522, 590
MPI_STATUS_SIZE, 15, 32, 514
MPI_STATUSES_IGNORE, 14, 15, 34, 375, 376, 481, 502, 522, 590
MPI_SUBVERSION, 272, 522
MPI_SUCCESS, 17, 18, 53, 60, 62, 227–232, 234, 235, 283, 284, 288, 289, 312, 437, 462, 463, 512
MPI_SUM, 165, 166, 509, 518
MPI_TAG, 32, 514
MPI_TAG_UB, 29, 272, 505, 509, 517
MPI_THREAD_FUNNELED, 384, 385, 521
MPI_THREAD_MULTIPLE, 385, 387, 521
MPI_THREAD_SERIALIZED, 385, 521
MPI_THREAD_SINGLE, 384–386, 521
MPI_TYPECLASS_COMPLEX, 495, 522
MPI_TYPECLASS_INTEGER, 495, 522
MPI_TYPECLASS_REAL, 495, 522
MPI_UB, 12, 16, 17, 89, 93, 96–98, 101, 430, 517
MPI_UINT16_T, 28, 165, 515, 592–595
MPI_UINT32_T, 28, 165, 515, 592–595
MPI_UINT64_T, 28, 165, 515, 592–595
MPI_UINT8_T, 28, 165, 515, 592–595
MPI_UNDEFINED, 33, 58, 59, 61, 62, 103, 192, 193, 206, 258, 267, 268, 491, 514, 595
MPI_UNEQUAL, 193, 200, 218, 517
MPI_UNIVERSE_SIZE, 308, 328, 520
MPI_UNSIGNED, 28, 165, 515
MPI_UNSIGNED_CHAR, 28, 165, 167, 515
MPI_UNSIGNED_LONG, 28, 165, 515
MPI_UNSIGNED_LONG_LONG, 28, 165, 515, 595
MPI_UNSIGNED_SHORT, 28, 165, 515
MPI_UNWEIGHTED, 15, 253–256, 263, 264, 481, 522, 593
MPI_VERSION, 272, 522
MPI_WCHAR, 28, 167, 241, 432, 515, 595
MPI_WIN_BASE, 338, 509, 520
MPI_WIN_DISP_UNIT, 338, 520
MPI_WIN_NULL, 338, 518
MPI_WIN_SIZE, 338, 520
MPI_WTIME_IS_GLOBAL, 272, 273, 290, 505, 517

MPI Declarations Index

This index refers to declarations needed in C/C++, such as address kind integers, handles, etc. The underlined page numbers is the "main" reference (sometimes there are more than one when key concepts are discussed in multiple areas).

MPI::Aint, 15, 15, 19, 79, 79, 81, 84, 86, 94, 97, 98, 107, 127–129, 336, 340, 342, 345, 431, 434, 459–462, 504, 504, 505, 523
MPI::Cartcomm, 248, 468, 474, 523
MPI::Comm, 26, 194, 199–202, 205, 208, 217, 218, 220, 226, 229, 230, 468, 474, 475, 523
MPI::Datatype, 19, 79, 468, 523
MPI::Distgraphcomm, 474, 523, 593
MPI::Errhandler, 278, 279–282, 464, 465, 468, 499, 523
MPI::Exception, 19, 23, 468, 476, 523
MPI::File, 281, 282, 289, 391, 393, 395–399, 401, 403, 407–421, 423–427, 431, 439, 440, 468, 499, 523
MPI::Graphcomm, 250, 468, 474, 523
MPI::Grequest, 374, 374, 468, 523
MPI::Group, 192, 192, 193–198, 202, 218, 338, 353, 354, 397, 468, 499, 523
MPI::Info, 274, 299, 299, 300–303, 308, 311, 313, 320–325, 391, 394, 398, 399, 401, 468, 499, 523
MPI::Intercomm, 468, 474, 523
MPI::Intracomm, 468, 474, 523
MPI::Offset, 16, 16, 19, 395, 396, 401, 403, 407–410, 415, 416, 420, 421, 423, 424, 435, 442, 523
MPI::Op, 163, 171, 174, 175, 177–181, 345, 468, 499, 523
MPI::Prequest, 70, 468, 523
MPI::Request, 50–52, 53, 54, 55, 57–62, 64, 67, 69–72, 374, 377, 409, 410, 413, 414, 418, 468, 499, 523
MPI::Status, 30, 32, 53, 54, 57–62, 64–66, 68, 74, 75, 102, 374, 380, 407–409, 411–413, 417, 419, 420, 423–427, 468, 469, 502, 523
MPI::Win, 231–233, 241, 242, 280, 281, 288, 336, 337, 338, 340, 342, 345, 352–357, 468, 499, 523
MPI_Aint, 15, 15, 18, 29, 79, 79, 81, 84, 86, 94, 97, 98, 107, 127–129, 336, 340, 342, 345, 431, 434, 459–462, 482, 504, 504, 505, 506, 523
MPI_Comm, 26, 194, 199–202, 205, 208, 217, 218, 220, 226, 229, 230, 517, 518, 523
MPI_Datatype, 79, 486, 515–518, 523
MPI_Errhandler, 278, 279–282, 464, 465, 499, 514, 518, 523
MPI_File, 281, 282, 289, 391, 393, 395–399, 401, 403, 407–421, 423–427, 431, 439, 440, 499, 518, 523
MPI_Fint, 499, 522, 523, 594
MPI_Group, 192, 192, 193–198, 202, 218, 338, 353, 354, 397, 499, 518, 523
MPI_Info, 274, 299, 299, 300–303, 308, 311, 313, 320–325, 391, 394, 398, 399, 401, 499, 518, 523, 597
MPI_Offset, 16, 16, 18, 29, 395, 396, 401, 403, 407–410, 415, 416, 420, 421, 423, 424, 435, 442, 442, 498, 523
MPI_Op, 163, 171, 174, 175, 177–181, 345, 499, 518, 523
MPI_Request, 50–52, 53, 54, 55, 57–62, 64, 67, 69–72, 374, 377, 409, 410, 413, 414, 418, 499, 518, 523

MPI_Status, 30, 32, 53, 54, 57–62, 64–66, 68, 74, 75, 102, 374, 380, 407–409, 411–413, 417, 419, 420, 423–427, 502, 522, 523
MPI_Win, 231–233, 241, 242, 280, 281, 288, 336, 337, 338, 340, 342, 345, 352–357, 499, 518, 523