

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 explicitly freed by the program, either via calls to `MPI_COMM_DELETE_ATTR` that free one attribute instance, or by calls to `MPI_COMM_FREE` that free all attribute instances associated with the freed communicator.

This call is identical to the MPI-1 call `MPI_KEYVAL_FREE` but is needed to match the new communicator-specific creation function. The use of `MPI_KEYVAL_FREE` is deprecated.

`MPI_COMM_SET_ATTR(comm, comm_keyval, attribute_val)`

INOUT `comm` communicator from which attribute will be attached (handle)

IN `comm_keyval` key value (integer)

IN `attribute_val` attribute value

`int MPI_Comm_set_attr(MPI_Comm comm, int comm_keyval, void *attribute_val)`

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

[ticket107.]C type: <code>const int</code> (or unnamed <code>enum</code>)	C++ type: <code>const int</code> (or unnamed <code>enum</code>)
[ticket107.]Fortran type: <code>INTEGER</code>	
<code>MPI_SUCCESS</code>	<code>MPI::SUCCESS</code>
<code>MPI_ERR_BUFFER</code>	<code>MPI::ERR_BUFFER</code>
<code>MPI_ERR_COUNT</code>	<code>MPI::ERR_COUNT</code>
<code>MPI_ERR_TYPE</code>	<code>MPI::ERR_TYPE</code>
<code>MPI_ERR_TAG</code>	<code>MPI::ERR_TAG</code>
<code>MPI_ERR_COMM</code>	<code>MPI::ERR_COMM</code>
<code>MPI_ERR_RANK</code>	<code>MPI::ERR_RANK</code>
<code>MPI_ERR_REQUEST</code>	<code>MPI::ERR_REQUEST</code>
<code>MPI_ERR_ROOT</code>	<code>MPI::ERR_ROOT</code>
<code>MPI_ERR_GROUP</code>	<code>MPI::ERR_GROUP</code>
<code>MPI_ERR_OP</code>	<code>MPI::ERR_OP</code>
<code>MPI_ERR_TOPOLOGY</code>	<code>MPI::ERR_TOPOLOGY</code>
<code>MPI_ERR_DIMS</code>	<code>MPI::ERR_DIMS</code>
<code>MPI_ERR_ARG</code>	<code>MPI::ERR_ARG</code>
<code>MPI_ERR_UNKNOWN</code>	<code>MPI::ERR_UNKNOWN</code>
<code>MPI_ERR_TRUNCATE</code>	<code>MPI::ERR_TRUNCATE</code>
<code>MPI_ERR_OTHER</code>	<code>MPI::ERR_OTHER</code>
<code>MPI_ERR_INTERN</code>	<code>MPI::ERR_INTERN</code>
<code>MPI_ERR_PENDING</code>	<code>MPI::ERR_PENDING</code>

(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

[ticket107.]Buffer Address Constants

[ticket107.]C type: void * const	[ticket107.]C++ type:
[ticket107.]Fortran type: (predefined memory location)	[ticket107.] void * const
MPI_BOTTOM	MPI::BOTTOM
MPI_IN_PLACE	MPI::IN_PLACE

Assorted Constants

[ticket107.]C type: const int (or unnamed enum)	[ticket107.]C++ type:
[ticket107.]Fortran type: INTEGER	[ticket107.] 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)

[ticket107.]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)

[ticket107.]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

[ticket107.]C type: MPI_Errhandler	C++ type: MPI::Errhandler
[ticket107.]Fortran type: INTEGER	
MPI_ERRORS_ARE_FATAL	MPI::ERRORS_ARE_FATAL
MPI_ERRORS_RETURN	MPI::ERRORS_RETURN
	MPI::ERRORS_THROW_EXCEPTIONS

Maximum Sizes for Strings

[ticket107.]C type: const int (or unnamed enum)	[ticket107.]C++ type:
[ticket107.]Fortran type: INTEGER	[ticket107.] 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
[ticket107.]C type: MPI_Datatype	C++ type: MPI::Datatype	
[ticket107.]Fortran type: INTEGER		
[ticket63.] MPI_CHAR	MPI::CHAR	char
		(treated as printable [ticket18+63.]character)
MPI_SHORT	MPI::SHORT	signed short int
MPI_INT	MPI::INT	signed int ⁹
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 ²¹
MPI_DOUBLE	MPI::DOUBLE	double ²²
MPI_LONG_DOUBLE	MPI::LONG_DOUBLE	long double ²³
MPI_WCHAR	MPI::WCHAR	wchar_t ²⁴
		(defined in <code><stddef.h></code>)
		(treated as printable [ticket18+63.]character)
[ticket18.] MPI_C_BOOL	[ticket18.](use C datatype handle)	[ticket18.] _Bool
[ticket18.] MPI_INT8_T	[ticket18.](use C datatype handle)	[ticket18.] int8_t
[ticket18.] MPI_INT16_T	[ticket18.](use C datatype handle)	[ticket18.] int16_t
[ticket18.] MPI_INT32_T	[ticket18.](use C datatype handle)	[ticket18.] int32_t
[ticket18.] MPI_INT64_T	[ticket18.](use C datatype handle)	[ticket18.] int64_t
[ticket18.] MPI_UINT8_T	[ticket18.](use C datatype handle)	[ticket18.] uint8_t
[ticket18.] MPI_UINT16_T	[ticket18.](use C datatype handle)	[ticket18.] uint16_t
[ticket18.] MPI_UINT32_T	[ticket18.](use C datatype handle)	[ticket18.] uint32_t
[ticket18.] MPI_UINT64_T	[ticket18.](use C datatype handle)	[ticket18.] uint64_t
[ticket18.] MPI_AINT	[ticket18.](use C datatype handle)	[ticket18.] MPF_Aint
[ticket18.] MPI_OFFSET	[ticket18.](use C datatype handle)	[ticket18.] MPF_Offset
[ticket18.] MPI_C_COMPLEX	[ticket18.](use C datatype handle)	[ticket18.] float _Complex
[ticket18.] MPI_C_FLOAT_COMPLEX	[ticket18.](use C datatype handle)	[ticket18.] float _Complex
[ticket18.] MPI_C_DOUBLE_COMPLEX	[ticket18.](use C datatype handle)	[ticket18.] double _Complex
[ticket18.] MPI_C_LONG_DOUBLE_COMPLEX	[ticket18.](use C datatype handle)	[ticket18.] long double _Complex
MPI_BYTE	MPI::BYTE	(any C/C++ type)
MPI_PACKED	MPI::PACKED	(any C/C++ type)

⁴⁵ ticket4.

⁴⁶

⁴⁷

⁴⁸

C and C++ (no Fortran) Named Predefined Datatypes | Fortran types

MPI-2.1 Review 33.d'

MPI_Fint	MPI::Fint	INTEGER
----------	-----------	---------

MPI-2.1 Review 33.d'

]

Named Predefined Datatypes		Fortran types
[ticket107.]C type: MPI_Datatype	C++ type: MPI::Datatype	
[ticket107.]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)
[ticket18.] MPI_AINT	[ticket18.](use C datatype handle)	[ticket18.] INTEGER (KIND=MPI_ADDRESS)
[ticket18.] MPI_OFFSET	[ticket18.](use C datatype handle)	[ticket18.] INTEGER (KIND=MPI_OFFSET)
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
[ticket107.]C type: MPI_Datatype	C++ type: MPI::Datatype	
[ticket107.]Fortran type: INTEGER		
MPI_DOUBLE_COMPLEX	[ticket40.] 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
[ticket57.] MPI_INTEGER16		[ticket57.] INTEGER*16
MPI_REAL2	MPI::REAL2	REAL*2
MPI_REAL4	MPI::REAL4	REAL*4
MPI_REAL8	MPI::REAL8	REAL*8
[ticket57.] MPI_REAL16		[ticket57.] REAL*16
[ticket57.] MPI_COMPLEX4		[ticket57.] COMPLEX*4
[ticket57.] MPI_COMPLEX8		[ticket57.] COMPLEX*8
[ticket57.] MPI_COMPLEX16		[ticket57.] COMPLEX*16
[ticket57.] MPI_COMPLEX32		[ticket57.] COMPLEX*32

Datatypes for reduction functions (C and C++)

[ticket107.]C type: MPI_Datatype	C++ type: MPI::Datatype
[ticket107.]Fortran type: INTEGER	
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)

[ticket107.]C type: MPI_Datatype	C++ type: MPI::Datatype
[ticket107.]Fortran type: INTEGER	
MPI_2REAL	MPI::TWOREAL
MPI_2DOUBLE_PRECISION	MPI::TWODOUBLE_PRECISION
MPI_2INTEGER	MPI::TWOINTEGER

Special datatypes for constructing derived datatypes

[ticket107.]C type: MPI_Datatype	C++ type: MPI::Datatype
[ticket107.]Fortran type: INTEGER	
MPI_UB	MPI::UB
MPI_LB	MPI::LB

Reserved communicators

[ticket107.]C type: MPI_Comm	C++ type: MPI::Intracomm
[ticket107.]Fortran type: INTEGER	
MPI_COMM_WORLD	MPI::COMM_WORLD
MPI_COMM_SELF	MPI::COMM_SELF

Results of communicator and group comparisons

[ticket107.]C type: const int (or unnamed enum)	C++ type: const int
[ticket107.]Fortran type: INTEGER	(or unnamed enum)
MPI_IDENT	MPI::IDENT
MPI_CONGRUENT	MPI::CONGRUENT
MPI_SIMILAR	MPI::SIMILAR
MPI_UNEQUAL	MPI::UNEQUAL

Environmental inquiry keys

[ticket107.]C type: const int (or unnamed enum)	C++ type: const int
[ticket107.]Fortran type: INTEGER	(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

[ticket107.]C type: MPI_Op	C++ type: const MPI::Op
[ticket107.]Fortran type: INTEGER	
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++ name
[ticket107.]C type / Fortran type	C++ type
MPI_GROUP_NULL	MPI::GROUP_NULL
[ticket107.] MPI_Group / INTEGER	const MPI::Group
MPI_COMM_NULL	MPI::COMM_NULL
[ticket107.] MPI_Comm / INTEGER	¹⁾
MPI_DATATYPE_NULL	MPI::DATATYPE_NULL
[ticket107.] MPI_Datatype / INTEGER	const MPI::Datatype
MPI_REQUEST_NULL	MPI::REQUEST_NULL
[ticket107.] MPI_Request / INTEGER	const MPI::Request
MPI_OP_NULL	MPI::OP_NULL
[ticket107.] MPI_Op / INTEGER	const MPI::Op
MPI_ERRHANDLER_NULL	MPI::ERRHANDLER_NULL
[ticket107.] MPI_Errhandler / INTEGER	const MPI::Errhandler
MPI_FILE_NULL	MPI::FILE_NULL
[ticket107.] MPI_File / INTEGER	
MPI_INFO_NULL	MPI::INFO_NULL
[ticket107.] MPI_Info / INTEGER	[ticket107.] const MPI::Info
MPI_WIN_NULL	MPI::WIN_NULL
[ticket107.] MPI_Win / INTEGER	

¹⁾ C++ type: See Section 16.1.7 on page 492 regarding class hierarchy and the specific type of MPI::COMM_NULL

Empty group

[ticket107.]C type: MPI_Group	C++ type: const MPI::Group
[ticket107.]Fortran type: INTEGER	
MPI_GROUP_EMPTY	MPI::GROUP_EMPTY

Topologies

[ticket107.]C type: const int (or unnamed enum)	C++ type: const int
[ticket107.]Fortran type: INTEGER	(or unnamed enum)
MPI_GRAPH	MPI::GRAPH
[ticket33.]MPI_DIST_GRAPH	[ticket33.]MPI::DIST_GRAPH
MPI_CART	MPI::CART

ticket107.

Predefined functions

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

¹ See the advice to implementors on MPI_COMM_NULL_COPY_FN, ... in Section 6.7.2 on page 241

[ticket107.]**[Predefined]Deprecated predefined functions**

C/Fortran name	C++ name
[ticket107.]C type / Fortran type	C++ type
MPI_NULL_COPY_FN	MPI::NULL_COPY_FN
[ticket107.]MPI_Copy_function / COPY_FUNCTION	MPI::Copy_function
MPI_DUP_FN	MPI::DUP_FN
[ticket107.]MPI_Copy_function / COPY_FUNCTION	MPI::Copy_function
MPI_NULL_DELETE_FN	MPI::NULL_DELETE_FN
[ticket107.]MPI_Delete_function / DELETE_FUNCTION	MPI::Delete_function

Predefined Attribute Keys

[ticket107.]C type: const int (or unnamed enum)	[ticket107.]C++ type:
[ticket107.]Fortran type: INTEGER	[ticket107.] const int (or unnamed enum)
MPI_APPNUM	MPI::APPNUM
MPI_LASTUSEDPCODE	MPI::LASTUSEDPCODE
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

[ticket107.]C type: const int (or unnamed enum)	[ticket107.]C++ type:
[ticket107.]Fortran type: INTEGER	[ticket107.] 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

[ticket107.]C type: <code>const int</code> (or unnamed <code>enum</code>)	[ticket107.]C++ type:
[ticket107.]Fortran type: <code>INTEGER</code>	[ticket107.] <code>const int</code> (or unnamed <code>enum</code>)
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

[ticket107.]C type: <code>const int</code> (or unnamed <code>enum</code>)	[ticket107.]C++ type:
[ticket107.]Fortran type: <code>INTEGER</code>	[ticket107.] <code>const int</code> (or unnamed <code>enum</code>)
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

[ticket107.]C type: <code>const MPI_Offset</code> (or unnamed <code>enum</code>)	[ticket107.]C++ type:
[ticket107.]Fortran type: <code>INTEGER (KIND=MPI_OFFSET_KIND)</code>	[ticket107.] <code>const MPI::Offset</code> (or unnamed <code>enum</code>)
MPI_DISPLACEMENT_CURRENT	MPI::DISPLACEMENT_CURRENT

File Operation Constants, Part 2

[ticket107.]C type: const int (or unnamed enum)	[ticket107.]C++ type:
[ticket107.]Fortran type: INTEGER	[ticket107.] 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

[ticket107.]C type: const int (or unnamed enum)	[ticket107.]C++ type:
[ticket107.]Fortran type: INTEGER	[ticket107.] const int (or unnamed enum)
MPI_TYPECLASS_COMPLEX	MPI::TYPECLASS_COMPLEX
MPI_TYPECLASS_INTEGER	MPI::TYPECLASS_INTEGER
MPI_TYPECLASS_REAL	MPI::TYPECLASS_REAL

Handles to Assorted Structures in C and C++ (no Fortran)

MPI_File	MPI::File
MPI_Info	MPI::Info
MPI_Win	MPI::Win

Constants Specifying Empty or Ignored Input

[ticket107.]C/Fortran name	[ticket107.]C++ name
[ticket107.]C type / Fortran type	[ticket107.]C++ type
MPI_ARGVS_NULL	MPI::ARGVS_NULL
[ticket107.] char*** / 2-dim. array of CHARACTER*(*)	[ticket107.] const char ***
MPI_ARGV_NULL	MPI::ARGV_NULL
[ticket107.] char** / array of CHARACTER*(*)	[ticket107.] const char **
MPI_ERRCODES_IGNORE	Not defined for C++
[ticket107.] int* / INTEGER array	
MPI_STATUSES_IGNORE	Not defined for C++
[ticket107.] MPI_Status* / INTEGER , DIMENSION(MPI_STATUS_SIZE,*)	
MPI_STATUS_IGNORE	Not defined for C++
[ticket107.] MPI_Status* / INTEGER , DIMENSION(MPI_STATUS_SIZE)	
[ticket33.] MPI_UNWEIGHTED	[ticket33.]Not defined for C++

C Constants Specifying Ignored Input (no C++ or Fortran)[ticket107.]C type: **MPI_Fint***

MPI_F_STATUSES_IGNORE

MPI_F_STATUS_IGNORE

C and C++ preprocessor Constants and Fortran Parameters[ticket107.]C/C++ type: **const int** (or unnamed **enum**)[ticket107.]Fortran type: **INTEGER**

MPI_SUBVERSION

MPI_VERSION

A.1.2 Types

The following are defined C type definitions, included in the file `mpi.h`.

/* C opaque types */

MPI_Aint

MPI_Fint

MPI_Offset

MPI_Status

/* C handles to assorted structures */

MPI_Comm

MPI_Datatype

MPI_Errhandler

MPI_File

MPI_Group

MPI_Info

MPI_Op

MPI_Request

MPI_Win

// C++ opaque types (all within the MPI namespace)

MPI::Aint

MPI::Offset

MPI::Status

// C++ handles to assorted structures (classes,

// all within the MPI namespace)

MPI::Comm

MPI::Intracomm

MPI::Graphcomm

MPI::Distgraphcomm

MPI::Cartcomm

MPI::Intercomm

MPI::Datatype

MPI::Errhandler

MPI::Exception

ticket33.

MPI Constant and Predefined Handle Index

This index lists predefined MPI constants and handles.

MPI::[*_NULL](#), [487](#)
MPI::[_LONG_LONG](#), [489](#)
MPI::[ANY_SOURCE](#), [532](#)
MPI::[ANY_TAG](#), [532](#)
MPI::[APPNUM](#), [538](#)
MPI::[ARGV_NULL](#), [540](#)
MPI::[ARGVS_NULL](#), [540](#)
MPI::[BAND](#), [536](#)
MPI::[BOOL](#), [490](#), [492](#), [534](#)
MPI::[BOR](#), [536](#)
MPI::[BOTTOM](#), [531](#)
MPI::[BSEND_OVERHEAD](#), [532](#)
MPI::[BXOR](#), [536](#)
MPI::[BYTE](#), [489](#), [490](#), [492](#), [533](#), [534](#)
MPI::[CART](#), [537](#)
MPI::[CHAR](#), [490](#), [533](#)
MPI::[CHARACTER](#), [490](#), [534](#)
MPI::[COMBINER_CONTIGUOUS](#), [539](#)
MPI::[COMBINER_DARRAY](#), [539](#)
MPI::[COMBINER_DUP](#), [539](#)
MPI::[COMBINER_F90_COMPLEX](#), [539](#)
MPI::[COMBINER_F90_INTEGER](#), [539](#)
MPI::[COMBINER_F90_REAL](#), [539](#)
MPI::[COMBINER_HINDEXED](#), [539](#)
MPI::[COMBINER_HINDEXED_INTEGER](#), [539](#)
MPI::[COMBINER_HVECTOR](#), [539](#)
MPI::[COMBINER_HVECTOR_INTEGER](#), [539](#)
MPI::[COMBINER_INDEXED](#), [539](#)
MPI::[COMBINER_INDEXED_BLOCK](#), [539](#)
MPI::[COMBINER_NAMED](#), [539](#)
MPI::[COMBINER_RESIZED](#), [539](#)
MPI::[COMBINER_STRUCT](#), [539](#)
MPI::[COMBINER_STRUCT_INTEGER](#), [539](#)
MPI::[COMBINER_SUBARRAY](#), [539](#)
MPI::[COMBINER_VECTOR](#), [539](#)
MPI::[COMM_NULL](#), [489](#), [493](#), [536](#)
MPI::[COMM_SELF](#), [535](#)
MPI::[COMM_WORLD](#), [535](#)
MPI::[COMPLEX](#), [490](#), [492](#), [534](#)
MPI::[CONGRUENT](#), [535](#)
MPI::[DATATYPE_NULL](#), [536](#)
MPI::[DISPLACEMENT_CURRENT](#), [539](#)
MPI::[DIST_GRAPH](#), [537](#)
MPI::[DISTRIBUTE_BLOCK](#), [540](#)
MPI::[DISTRIBUTE_CYCLIC](#), [540](#)
MPI::[DISTRIBUTE_DFLT_DARG](#), [540](#)
MPI::[DISTRIBUTE_NONE](#), [540](#)
MPI::[DOUBLE](#), [490](#), [492](#), [533](#)
MPI::[DOUBLE_COMPLEX](#), [490](#), [492](#), [534](#)
MPI::[DOUBLE_INT](#), [491](#), [535](#)
MPI::[DOUBLE_PRECISION](#), [490](#), [492](#), [534](#)
MPI::[DUP_FN](#), [538](#)
MPI::[ERR_ARG](#), [530](#)
MPI::[ERR_BUFFER](#), [530](#)
MPI::[ERR_COMM](#), [530](#)
MPI::[ERR_COUNT](#), [530](#)
MPI::[ERR_DIMS](#), [530](#)
MPI::[ERR_GROUP](#), [530](#)
MPI::[ERR_IN_STATUS](#), [531](#)
MPI::[ERR_INTERN](#), [530](#)
MPI::[ERR_LASTCODE](#), [531](#)
MPI::[ERR_OP](#), [530](#)
MPI::[ERR_OTHER](#), [530](#)
MPI::[ERR_PENDING](#), [530](#)
MPI::[ERR_RANK](#), [530](#)
MPI::[ERR_REQUEST](#), [530](#)

MPI::ERR_ROOT, 530	MPI::LONG_INT, 491, 535	1
MPI::ERR_TAG, 530	MPI::LONG_LONG, 490, 533	2
MPI::ERR_TOPOLOGY, 530	MPI::LONG_LONG_INT, 533	3
MPI::ERR_TRUNCATE, 530	MPI::LOR, 536	4
MPI::ERR_TYPE, 530	MPI::LXOR, 536	5
MPI::ERR_UNKNOWN, 530	MPI::MAX, 536	6
MPI::ERRHANDLER_NULL, 536	MPI::MAX_DATAREP_STRING, 532	7
MPI::ERRORS_ARE_FATAL, 20, 532	MPI::MAX_ERROR_STRING, 532	8
MPI::ERRORS_RETURN, 20, 532	MPI::MAX_INFO_KEY, 532	9
MPI::ERRORS_THROW_EXCEPTIONS,	MPI::MAX_INFO_VAL, 532	10
20, 23, 293, 532	MPI::MAX_OBJECT_NAME, 532	11
MPI::F_COMPLEX, 490, 492, 534	MPI::MAX_PORT_NAME, 532	12
MPI::F_COMPLEX16, 491, 492	MPI::MAX_PROCESSOR_NAME, 532	13
MPI::F_COMPLEX32, 491, 492	MPI::MAXLOC, 492, 536	14
MPI::F_COMPLEX4, 491, 492	MPI::MIN, 536	15
MPI::F_COMPLEX8, 491, 492	MPI::MINLOC, 492, 536	16
MPI::F_DOUBLE_COMPLEX, 491, 492,	MPI::MODE_APPEND, 538	17
534	MPI::MODE_CREATE, 538	18
MPI::FILE_NULL, 536	MPI::MODE_DELETE_ON_CLOSE, 538	19
MPI::FLOAT, 490, 492, 533	MPI::MODE_EXCL, 538	20
MPI::FLOAT_INT, 491, 535	MPI::MODE_NOCHECK, 538	21
MPI::GRAPH, 537	MPI::MODE_NOPRECEDE, 538	22
MPI::GROUP_EMPTY, 536	MPI::MODE_NOPUT, 538	23
MPI::GROUP_NULL, 536	MPI::MODE_NOSTORE, 538	24
MPI::HOST, 535	MPI::MODE_NOSUCCEED, 538	25
MPI::IDENT, 535	MPI::MODE_RDONLY, 538	26
MPI::IN_PLACE, 531	MPI::MODE_RDWR, 538	27
MPI::INFO_NULL, 536	MPI::MODE_SEQUENTIAL, 538	28
MPI::INT, 489, 490, 533	MPI::MODE_UNIQUE_OPEN, 538	29
MPI::INTEGER, 489, 490, 534	MPI::MODE_WRONLY, 538	30
MPI::INTEGER1, 491, 492, 534	MPI::NULL_COPY_FN, 538	31
MPI::INTEGER16, 491, 492	MPI::NULL_DELETE_FN, 538	32
MPI::INTEGER2, 491, 492, 534	MPI::OP_NULL, 536	33
MPI::INTEGER4, 491, 492, 534	MPI::ORDER_C, 540	34
MPI::INTEGER8, 491, 492, 534	MPI::ORDER_FORTRAN, 540	35
MPI::IO, 535	MPI::PACKED, 489, 490, 533, 534	36
MPI::KEYVAL_INVALID, 532	MPI::PROC_NULL, 532	37
MPI::LAND, 536	MPI::PROD, 536	38
MPI::LASTUSEDPCODE, 538	MPI::REAL, 490, 492, 534	39
MPI::LB, 535	MPI::REAL16, 491, 492	40
MPI::LOCK_EXCLUSIVE, 532	MPI::REAL2, 491, 492, 534	41
MPI::LOCK_SHARED, 532	MPI::REAL4, 491, 492, 534	42
MPI::LOGICAL, 490, 492, 534	MPI::REAL8, 491, 492, 534	43
MPI::LONG, 489, 490, 533	MPI::REPLACE, 536	44
MPI::LONG_DOUBLE, 490, 492, 533	MPI::REQUEST_NULL, 536	45
MPI::LONG_DOUBLE_COMPLEX, 490,	MPI::ROOT, 532	46
492, 534	MPI::SEEK_CUR, 540	47
MPI::LONG_DOUBLE_INT, 491, 535	MPI::SEEK_END, 540	48

- 1 MPI::SEEK_SET, 540
- 2 MPI::SHORT, 489, 490, 533
- 3 MPI::SHORT_INT, 491, 535
- 4 MPI::SIGNED_CHAR, 489, 490, 533
- 5 MPI::SIMILAR, 535
- 6 MPI::SUCCESS, 530
- 7 MPI::SUM, 536
- 8 MPI::TAG_UB, 535
- 9 MPI::THREAD_FUNNELED, 539
- 10 MPI::THREAD_MULTIPLE, 539
- 11 MPI::THREAD_SERIALIZED, 539
- 12 MPI::THREAD_SINGLE, 539
- 13 MPI::TWODOUBLE_PRECISION, 491, 535
- 14 MPI::TWOINT, 491, 535
- 15 MPI::TWOINTEGER, 491, 535
- 16 MPI::TWOREAL, 491, 535
- 17 MPI::TYPECLASS_COMPLEX, 540
- 18 MPI::TYPECLASS_INTEGER, 540
- 19 MPI::TYPECLASS_REAL, 540
- 20 MPI::UB, 535
- 21 MPI::UNDEFINED, 532
- 22 MPI::UNEQUAL, 535
- 23 MPI::UNIVERSE_SIZE, 538
- 24 MPI::UNSIGNED, 489, 490, 533
- 25 MPI::UNSIGNED_CHAR, 489, 490, 533
- 26 MPI::UNSIGNED_LONG, 489, 490, 533
- 27 MPI::UNSIGNED_LONG_LONG, 489, 490,
- 28 533
- 29 MPI::UNSIGNED_SHORT, 489, 490, 533
- 30 MPI::WCHAR, 490, 533
- 31 MPI::WIN_BASE, 538
- 32 MPI::WIN_DISP_UNIT, 538
- 33 MPI::WIN_NULL, 536
- 34 MPI::WIN_SIZE, 538
- 35 MPI::WTIME_IS_GLOBAL, 535
- 36 MPI_2DOUBLE_PRECISION, 172, 173,
- 37 535
- 38 MPI_2INT, 173, 535
- 39 MPI_2INTEGER, 172, 173, 535
- 40 MPI_2REAL, 172, 173, 535
- 41 MPI_ADDRESS_KIND, 15, 16, 16, 18, 110,
- 42 500, 523, 524, 532
- 43 MPI_AINT, 31, 169, 533, 534, 610–613
- 44 MPI_ANY_SOURCE, 32, 33, 44, 54, 55,
- 45 68–70, 74, 77, 78, 259, 289, 532
- 46 MPI_ANY_TAG, 14, 32, 33, 35, 54, 55,
- 47 68–70, 74, 77–79, 532
- 48 MPI_APPNUM, 345, 346, 538
- MPI_ARGV_NULL, 15, 326, 327, 499, 540
- MPI_ARGVS_NULL, 15, 330, 499, 540
- MPI_BAND, 169, 170, 536
- MPI_BOR, 169, 170, 536
- MPI_BOTTOM, 10, 15, 16, 18, 36, 98,
- 108, 138, 270, 271, 328, 499, 502,
- 504, 507, 521, 522, 528, 531, 616
- MPI_BSEND_OVERHEAD, 50, 290, 532
- MPI_BXOR, 169, 170, 536
- MPI_BYTE, 29, 30, 37–39, 131, 170, 408,
- 447, 459, 489, 528, 533, 534, 613
- MPI_C_BOOL, 30, 170, 533, 610–613
- MPI_C_COMPLEX, 30, 533, 610–613
- MPI_C_DOUBLE_COMPLEX, 30, 170, 533,
- 610–613
- MPI_C_FLOAT_COMPLEX, 30, 170, 533,
- 610–613
- MPI_C_LONG_DOUBLE_COMPLEX, 30,
- 170, 533, 610–613
- MPI_CART, 274, 537
- MPI_CHAR, 30, 40, 90, 171, 533, 610
- MPI_CHARACTER, 29, 38–40, 171, 534
- MPI_COMBINER_CONTIGUOUS, 110,
- 114, 539
- MPI_COMBINER_DARRAY, 110, 115, 539
- MPI_COMBINER_DUP, 110, 113, 539
- MPI_COMBINER_F90_COMPLEX, 110,
- 115, 539
- MPI_COMBINER_F90_INTEGER, 110, 115,
- 539
- MPI_COMBINER_F90_REAL, 110, 115,
- 539
- MPI_COMBINER_HINDEXED, 110, 114,
- 539
- MPI_COMBINER_HINDEXED_INTEGER,
- 110, 114, 539
- MPI_COMBINER_HVECTOR, 110, 114,
- 539
- MPI_COMBINER_HVECTOR_INTEGER,
- 110, 114, 539
- MPI_COMBINER_INDEXED, 110, 114,
- 539
- MPI_COMBINER_INDEXED_BLOCK, 110,
- 114, 539
- MPI_COMBINER_NAMED, 110, 113, 539
- MPI_COMBINER_RESIZED, 110, 115, 539
- MPI_COMBINER_STRUCT, 110, 114, 539

MPI_COMBINER_STRUCT_INTEGER,	MPI_ERR_CONVERSION, 301, 454, 466	1
110, 114, 539	MPI_ERR_COUNT, 300, 530	2
MPI_COMBINER_SUBARRAY, 110, 115,	MPI_ERR_DIMS, 300, 530	3
539	MPI_ERR_DISP, 300, 380	4
MPI_COMBINER_VECTOR, 110, 114, 539	MPI_ERR_DUP_DATAREP, 301, 452, 466	5
MPI_COMM_NULL, 195, 206–210, 213,	MPI_ERR_FILE, 301, 466	6
255, 264, 266, 328, 347, 348, 536,	MPI_ERR_FILE_EXISTS, 301, 466	7
614	MPI_ERR_FILE_IN_USE, 301, 412, 466	8
MPI_COMM_PARENT, 255	MPI_ERR_GROUP, 300, 530	9
MPI_COMM_SELF, 195, 240, 255, 312,	MPI_ERR_IN_STATUS, 34, 36, 55, 63,	10
347, 409, 535, 612	65, 294, 300, 394, 424, 495, 531	11
MPI_COMM_WORLD, 14, 24, 31, 195–	MPI_ERR_INFO, 300	12
197, 203, 204, 217, 228, 229, 255,	MPI_ERR_INFO_KEY, 300, 316	13
264, 288, 289, 292, 295, 303, 310,	MPI_ERR_INFO_NOKEY, 300, 317	14
311, 313, 321, 322, 324, 325, 329–	MPI_ERR_INFO_VALUE, 300, 316	15
331, 344–347, 403, 446, 465, 516,	MPI_ERR_INTERN, 300, 530	16
528, 535, 615	MPI_ERR_IO, 301, 466	17
MPI_COMPLEX, 29, 170, 449, 508, 534	MPI_ERR_KEYVAL, 252, 300	18
MPI_COMPLEX16, 170, 534	MPI_ERR_LASTCODE, 299, 301, 303, 304,	19
MPI_COMPLEX32, 170, 534	531	20
MPI_COMPLEX4, 170, 534	MPI_ERR_LOCKTYPE, 300, 380	21
MPI_COMPLEX8, 170, 534	MPI_ERR_NAME, 300, 341	22
MPI_CONGRUENT, 204, 227, 535	MPI_ERR_NO_MEM, 291, 300	23
MPI_CONVERSION_FN_NULL, 454	MPI_ERR_NO_SPACE, 301, 466	24
MPI_DATATYPE, 20	MPI_ERR_NO_SUCH_FILE, 301, 412, 466	25
MPI_DATATYPE_NULL, 104, 536	MPI_ERR_NOT_SAME, 301, 466	26
MPI_DISPLACEMENT_CURRENT, 420,	MPI_ERR_OP, 300, 530	27
539, 616	MPI_ERR_OTHER, 299, 300, 530	28
MPI_DIST_GRAPH, 274, 537, 612	MPI_ERR_PENDING, 63, 300, 530	29
MPI_DISTRIBUTE_BLOCK, 95, 540	MPI_ERR_PORT, 300, 338	30
MPI_DISTRIBUTE_CYCLIC, 95, 540	MPI_ERR_QUOTA, 301, 466	31
MPI_DISTRIBUTE_DFLT_DARG, 95, 540	MPI_ERR_RANK, 300, 530	32
MPI_DISTRIBUTE_NONE, 95, 540	MPI_ERR_READ_ONLY, 301, 466	33
MPI_DOUBLE, 30, 169, 507, 533	MPI_ERR_REQUEST, 300, 530	34
MPI_DOUBLE_COMPLEX, 29, 170, 449,	MPI_ERR_RMA_CONFLICT, 300, 380	35
508, 534	MPI_ERR_RMA_SYNC, 300, 380	36
MPI_DOUBLE_INT, 173, 535	MPI_ERR_ROOT, 300, 530	37
MPI_DOUBLE_PRECISION, 29, 169, 508,	MPI_ERR_SERVICE, 300, 340	38
534	MPI_ERR_SIZE, 300, 380	39
MPI_DUP_FN, 538	MPI_ERR_SPAWN, 300, 327, 328	40
MPI_ERR_ACCESS, 301, 412, 466	MPI_ERR_TAG, 300, 530	41
MPI_ERR_AMODE, 301, 411, 466	MPI_ERR_TOPOLOGY, 300, 530	42
MPI_ERR_ARG, 300, 530	MPI_ERR_TRUNCATE, 300, 530	43
MPI_ERR_ASSERT, 300, 380	MPI_ERR_TYPE, 300, 530	44
MPI_ERR_BAD_FILE, 301, 466	MPI_ERR_UNKNOWN, 299, 300, 530	45
MPI_ERR_BASE, 291, 300, 380	MPI_ERR_UNSUPPORTED_DATAREP,	46
MPI_ERR_BUFFER, 300, 530	301, 466	47
MPI_ERR_COMM, 300, 530		48

MPI_ERR_UNSUPPORTED_OPERATION, 301, 466
 MPI_ERR_WIN, 300, 380
 MPI_ERRCODES_IGNORE, 15, 18, 328, 499, 502, 540
 MPI_ERRHANDLER_NULL, 298, 536
 MPI_ERROR, 34, 55, 532
 MPI_ERROR_STRING, 299
 MPI_ERRORS_ARE_FATAL, 292, 293, 304, 305, 380, 465, 532
 MPI_ERRORS_RETURN, 292, 293, 305, 465, 528, 532
 MPI_F_STATUS_IGNORE, 520, 541
 MPI_F_STATUSES_IGNORE, 520, 541
 MPI_FILE_NULL, 412, 465, 536
 MPI_FLOAT, 30, 90, 167, 169, 448, 533
 MPI_FLOAT_INT, 12, 172, 173, 535
 MPI_GRAPH, 274, 537
 MPI_GROUP_EMPTY, 194, 199, 200, 207, 208, 536
 MPI_GROUP_NULL, 194, 202, 536
 MPI_HOST, 288, 535
 MPI_IDENT, 197, 204, 535
 MPI_IN_PLACE, 15, 138, 164, 499, 502, 507, 531
 MPI_INFO_NULL, 272, 319, 327, 337, 411, 412, 421, 536
 MPI_INT, 12, 30, 82, 169, 448, 449, 507, 528, 529, 533
 MPI_INT16_T, 30, 169, 533, 610–613
 MPI_INT32_T, 30, 169, 533, 610–613
 MPI_INT64_T, 30, 169, 533, 610–613
 MPI_INT8_T, 30, 169, 533, 610–613
 MPI_INTEGER, 29, 37, 169, 507, 508, 529, 534
 MPI_INTEGER1, 29, 169, 534
 MPI_INTEGER16, 169, 534
 MPI_INTEGER2, 29, 169, 449, 534
 MPI_INTEGER4, 29, 169, 534
 MPI_INTEGER8, 169, 512, 534
 MPI_INTEGER_KIND, 15, 110, 523, 532
 MPI_IO, 288, 289, 535
 MPI_KEYVAL_INVALID, 244, 245, 532
 MPI_LAND, 169, 170, 536
 MPI_LASTUSED_CODE, 303, 538
 MPI_LB, 16, 17, 93, 96, 100–102, 105, 448, 535
 MPI_LOCK_EXCLUSIVE, 374, 532
 MPI_LOCK_SHARED, 374, 532
 MPI_LOGICAL, 29, 170, 534
 MPI_LONG, 30, 169, 533
 MPI_LONG_DOUBLE, 30, 169, 533
 MPI_LONG_DOUBLE_INT, 173, 535
 MPI_LONG_INT, 173, 535
 MPI_LONG_LONG, 30, 169, 533, 613
 MPI_LONG_LONG_INT, 30, 169, 533, 613
 MPI_LOR, 169, 170, 536
 MPI_LXOR, 169, 170, 536
 MPI_MAX, 167, 169, 170, 186, 536
 MPI_MAX_DATAREP_STRING, 15, 422, 452, 532
 MPI_MAX_ERROR_STRING, 15, 299, 304, 532
 MPI_MAX_INFO_KEY, 15, 300, 315, 317, 318, 532
 MPI_MAX_INFO_VAL, 15, 300, 315, 532
 MPI_MAX_OBJECT, 255, 614
 MPI_MAX_OBJECT_NAME, 15, 254, 255, 532, 614
 MPI_MAX_PORT_NAME, 15, 336, 532
 MPI_MAX_PROCESSOR_NAME, 15, 290, 532, 615
 MPI_MAXLOC, 169, 171, 172, 175, 536
 MPI_MIN, 169, 170, 536
 MPI_MINLOC, 169, 171, 172, 175, 536
 MPI_MODE_APPEND, 410, 411, 538
 MPI_MODE_CREATE, 410, 411, 418, 538
 MPI_MODE_DELETE_ON_CLOSE, 410–412, 538
 MPI_MODE_EXCL, 410, 411, 538
 MPI_MODE_NOCHECK, 376, 377, 538
 MPI_MODE_NOPRECEDE, 376, 377, 538
 MPI_MODE_NOPUT, 376, 377, 538
 MPI_MODE_NOSTORE, 376, 377, 538
 MPI_MODE_NOSUCCEED, 376, 377, 538
 MPI_MODE_RDONLY, 410, 411, 416, 538
 MPI_MODE_RDWR, 410, 411, 538
 MPI_MODE_SEQUENTIAL, 410, 411, 413, 414, 420, 425, 428, 438, 458, 538, 616
 MPI_MODE_UNIQUE_OPEN, 410, 411, 538
 MPI_MODE_WRONLY, 410, 411, 538
 MPI_NULL_COPY_FN, 538
 MPI_NULL_DELETE_FN, 538
 MPI_OFFSET, 31, 169, 533, 534, 610–613

MPI_OFFSET_KIND, 15 , 16 , 18 , 459 , 500 , 532	1
MPI_OP_NULL, 178 , 536	2
MPI_ORDER_C, 14 , 92 , 95 , 96 , 540	3
MPI_ORDER_FORTRAN, 14 , 92 , 95 , 540	4
MPI_PACKED, 29 , 30 , 37 , 126 , 127 , 131 , 450 , 489 , 528 , 533 , 534	5
MPI_PROC_NULL, 28 , 78 , 79 , 140 , 142 , 144 , 146 , 154 , 155 , 168 , 197 , 282 , 288 , 289 , 356 , 532 , 613 , 615 , 616	6
MPI_PROD, 169 , 170 , 536	7
MPI_REAL, 29 , 37 , 169 , 449 , 507 , 508 , 514 , 534	8
MPI_REAL16, 170 , 534	9
MPI_REAL2, 29 , 170 , 534	10
MPI_REAL4, 29 , 170 , 507 , 512 , 534	11
MPI_REAL8, 29 , 170 , 507 , 534 , 611	12
MPI_REPLACE, 363 , 536 , 612 , 616	13
MPI_REQUEST_NULL, 55–58 , 61–64 , 394 , 536	14
MPI_ROOT, 140 , 532	15
MPI_SEEK_CUR, 433 , 439 , 540	16
MPI_SEEK_END, 433 , 439 , 540	17
MPI_SEEK_SET, 433 , 434 , 439 , 540	18
MPI_SHORT, 30 , 169 , 533	19
MPI_SHORT_INT, 173 , 535	20
MPI_SIGNED_CHAR, 30 , 169 , 171 , 533 , 613	21
MPI_SIMILAR, 197 , 204 , 227 , 535	22
MPI_SOURCE, 34 , 532	23
MPI_STATUS, 21 , 35 , 36 , 55	24
MPI_STATUS_IGNORE, 10 , 15 , 18 , 35 , 36 , 393 , 424 , 499 , 502 , 507 , 520 , 528 , 540 , 608	25
MPI_STATUS_SIZE, 15 , 34 , 532	26
MPI_STATUSES_IGNORE, 14 , 15 , 36 , 393 , 394 , 499 , 502 , 520 , 540 , 608	27
MPI_SUBVERSION, 288 , 541	28
MPI_SUCCESS, 18 , 19 , 55 , 63 , 65 , 243 , 245–247 , 249 , 250 , 299 , 300 , 304 , 305 , 328 , 454 , 480 , 481 , 530	29
MPI_SUM, 169 , 170 , 528 , 536	30
MPI_TAG, 34 , 532	31
MPI_TAG_UB, 31 , 288 , 523 , 527 , 535	32
MPI_THREAD_FUNNELED, 402 , 403 , 539	33
MPI_THREAD_MULTIPLE, 403 , 405 , 539	34
MPI_THREAD_SERIALIZED, 403 , 539	35
MPI_THREAD_SINGLE, 402–404 , 539	36
MPI_TYPECLASS_COMPLEX, 513 , 540	37
MPI_TYPECLASS_INTEGER, 513 , 540	38
MPI_TYPECLASS_REAL, 513 , 540	39
MPI_UB, 12 , 16 , 17 , 93 , 97 , 100–102 , 105 , 448 , 535	40
MPI_UINT16_T, 30 , 169 , 533 , 610–613	41
MPI_UINT32_T, 30 , 169 , 533 , 610–613	42
MPI_UINT64_T, 30 , 169 , 533 , 610–613	43
MPI_UINT8_T, 30 , 169 , 533 , 610–613	44
MPI_UNDEFINED, 35 , 61 , 62 , 64 , 65 , 107 , 196 , 197 , 210 , 274 , 284 , 285 , 509 , 510 , 532 , 613	45
MPI_UNEQUAL, 197 , 204 , 227 , 535	46
MPI_UNIVERSE_SIZE, 324 , 344 , 538	47
MPI_UNSIGNED, 30 , 169 , 533	48
MPI_UNSIGNED_CHAR, 30 , 169 , 171 , 533	
MPI_UNSIGNED_LONG, 30 , 169 , 533	
MPI_UNSIGNED_LONG_LONG, 30 , 169 , 533 , 613	
MPI_UNSIGNED_SHORT, 30 , 169 , 533	
MPI_UNWEIGHTED, 15 , 269–272 , 280 , 281 , 499 , 540 , 612	
MPI_VERSION, 288 , 541	
MPI_WCHAR, 30 , 171 , 257 , 449 , 533 , 613	
MPI_WIN_BASE, 354 , 527 , 538	
MPI_WIN_DISP_UNIT, 354 , 538	
MPI_WIN_NULL, 354 , 536	
MPI_WIN_SIZE, 354 , 538	
MPI_WTIME_IS_GLOBAL, 288 , 289 , 306 , 523 , 535	

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](#), [20](#), [83](#), [83](#), [85](#), [88](#), [90](#),
[98](#), [101](#), [102](#), [111](#), [131–133](#), [352](#),
[356](#), [358](#), [362](#), [449](#), [452](#), [477–480](#),
[523](#), [523](#), [541](#)
- MPI::Cartcomm, [264](#), [486](#), [492](#), [541](#)
- MPI::Comm, [28](#), [198](#), [203–206](#), [209](#), [212](#),
[226](#), [227](#), [229](#), [242](#), [244–246](#), [486](#),
[492](#), [493](#), [541](#)
- MPI::Datatype, [20](#), [83](#), [486](#), [541](#)
- MPI::Distgraphcomm, [492](#), [541](#), [611](#)
- MPI::Errhandler, [294](#), [295–298](#), [482](#), [483](#),
[486](#), [517](#), [541](#)
- MPI::Exception, [20](#), [23](#), [486](#), [494](#), [541](#)
- MPI::File, [297](#), [298](#), [305](#), [409](#), [411](#), [413–](#)
[417](#), [419](#), [421](#), [425–439](#), [441–445](#),
[449](#), [457](#), [458](#), [486](#), [517](#), [540](#), [542](#)
- MPI::Fint, [534](#)
- MPI::Graphcomm, [266](#), [486](#), [492](#), [541](#)
- MPI::Grequest, [392](#), [392](#), [486](#), [542](#)
- MPI::Group, [196](#), [196](#), [197–202](#), [206](#), [227](#),
[354](#), [370](#), [371](#), [415](#), [486](#), [517](#), [542](#)
- MPI::Info, [290](#), [315](#), [315](#), [316–319](#), [324](#),
[327](#), [329](#), [336–341](#), [409](#), [412](#), [416](#),
[417](#), [419](#), [486](#), [517](#), [540](#), [542](#)
- MPI::Intercomm, [486](#), [492](#), [541](#)
- MPI::Intracomm, [486](#), [492](#), [541](#)
- MPI::Offset, [16](#), [16](#), [20](#), [413](#), [414](#), [419](#), [421](#),
[425–428](#), [433](#), [434](#), [438](#), [439](#), [441](#),
[442](#), [452](#), [459](#), [541](#)
- MPI::Op, [167](#), [175](#), [178](#), [179](#), [181–186](#), [362](#),
[486](#), [517](#), [542](#)
- MPI::Prequest, [73](#), [486](#), [542](#)
- MPI::Request, [52–54](#), [56](#), [56](#), [58](#), [60–65](#),
[67](#), [70](#), [72–75](#), [392](#), [395](#), [427](#), [428](#),
[431](#), [432](#), [436](#), [486](#), [517](#), [542](#)
- MPI::Status, [32](#), [34](#), [56](#), [60–65](#), [67–69](#), [71](#),
[77](#), [78](#), [106](#), [392](#), [398](#), [425–427](#),
[429–431](#), [435](#), [437](#), [438](#), [441–445](#),
[486](#), [487](#), [520](#), [541](#)
- MPI::Win, [247–249](#), [257](#), [296](#), [297](#), [304](#),
[352](#), [353](#), [354](#), [356](#), [358](#), [362](#), [369–](#)
[372](#), [374](#), [486](#), [517](#), [540](#), [542](#)
- MPI_Aint, [15](#), [15](#), [19](#), [31](#), [83](#), [83](#), [85](#), [88](#),
[90](#), [98](#), [101](#), [102](#), [111](#), [131–133](#),
[352](#), [356](#), [358](#), [362](#), [449](#), [452](#), [477–](#)
[480](#), [500](#), [523](#), [523](#), [524](#), [541](#)
- MPI_Comm, [28](#), [198](#), [203–206](#), [209](#), [212](#),
[226](#), [227](#), [229](#), [242](#), [244–246](#), [535](#),
[536](#), [541](#)
- MPI_Datatype, [83](#), [504](#), [533–536](#), [541](#)
- MPI_Errhandler, [294](#), [295–298](#), [482](#), [483](#),
[517](#), [532](#), [536](#), [541](#)
- MPI_File, [297](#), [298](#), [305](#), [409](#), [411](#), [413–](#)
[417](#), [419](#), [421](#), [425–439](#), [441–445](#),
[449](#), [457](#), [458](#), [517](#), [536](#), [540](#), [541](#)
- MPI_Fint, [517](#), [534](#), [541](#), [612](#)
- MPI_Group, [196](#), [196](#), [197–202](#), [206](#), [227](#),
[354](#), [370](#), [371](#), [415](#), [517](#), [536](#), [541](#)
- MPI_Info, [290](#), [315](#), [315](#), [316–319](#), [324](#),
[327](#), [329](#), [336–341](#), [409](#), [412](#), [416](#),
[417](#), [419](#), [517](#), [536](#), [540](#), [541](#), [615](#)
- MPI_Offset, [16](#), [16](#), [19](#), [31](#), [413](#), [414](#), [419](#),
[421](#), [425–428](#), [433](#), [434](#), [438](#), [439](#),
[441](#), [442](#), [452](#), [459](#), [459](#), [460](#), [516](#),
[541](#)
- MPI_Op, [167](#), [175](#), [178](#), [179](#), [181–186](#), [362](#),
[517](#), [536](#), [541](#)
- MPI_Request, [52–54](#), [56](#), [56](#), [58](#), [60–65](#),
[67](#), [70](#), [72–75](#), [392](#), [395](#), [427](#), [428](#),
[431](#), [432](#), [436](#), [517](#), [536](#), [541](#)

MPI_Status, [32](#), [34](#), [56](#), [60–65](#), [67–69](#), [71](#),
[77](#), [78](#), [106](#), [392](#), [398](#), [425–427](#),
[429–431](#), [435](#), [437](#), [438](#), [441–445](#),
[520](#), [540](#), [541](#)
MPI_Win, [247–249](#), [257](#), [296](#), [297](#), [304](#),
[352](#), [353](#), [354](#), [356](#), [358](#), [362](#), [369–](#)
[372](#), [374](#), [517](#), [536](#), [540](#), [541](#)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48