Annex A.2 Summary of the Semantics of all Communicating MPI Routines

Stages: i=initialization, s=starting, c=completion, f=freeing
Blk: b=blocking, nb=nonblocking. Note that from a user's view point, this column is only a hint. Relevant is, whether a routine is local or not and which resources are blocked until when. See next and last column.
Loc: l=local, nl=non-local
Bold: exceptions, e.g., b+l and nb+nl
Ixx: Using names with "I", Ix2 = "I" means immediate and incomplete, Im = "I" means only immediate
Op: part of operation type: b-op = blocking operation, nb-op = nonblocking operation, p-op = persistent operation
Collective procedures:
- C = all processes of the group must call the procedure
- sq = in the same sequence
- S1 = blocking synchronization, S2 = start-complete-synchronization

Blocked resources: They are blocked after the call until the end of subsequent stage where this resource is not further mentioned.

Remarks:
1) Must not return before the corresponding MPI_receive operation is started.
2) In a correct MPI program, a call to MPI_(I)RSEND requires that the receiver has already started the corresponding receive. Under this asumption, the call is local.
3) Usually, MPI_Wait is non-local, but in this case it is local.
4) In case of a MPI_(I)BARRIER, the S1/S2 synchronization is required (instead of "may or may not").
5) In this case, MPI_REQUEST_FREE is nonlocal, see the Advice to implementors in Section 6.4.3
6) It also may not return until MPI_INIT has been called in the children.
7) Cached on the request handle.
8) One of the rare cases that a nonblocking call is non-local.
9) One shall not free or deallocate the buffer before the operation is freed, that is MPI_REQUEST_FREE returned.
10) For MPI_WAIT and MPI_TEST, see corresponding lines for a) MPI_BSEND, or b) MPI_IBCAST.

Procedure	Stages	Blk	Loc	Ixx	Op	Collective C sq S1/2	Blocked resources and remarks
MPI_SEND	i-s-c-f	b	nl		b-op	-	
MPI_SSEND	i-s-c-f	b	nl		b-op	-	1)
MPI_RSEND	i-s-c-f	**b**	**l**		b-op	-	2)
MPI_BSEND	i-s-c-f	**b**	**l**		b-op	-	
MPI_RECV	i-s-c-f	b	nl		b-op	-	
MPI_ISEND, MPI_ISSEND	i-s----	nb	l	Ix2	nb-op	-	buffer
MPI_IRECV	i-s----	nb	l	Ix2	nb-op	-	buffer
corresponding MPI_Wait	----c-f		nl		nb-op	-	
corr. MPI_TEST returning flag=TRUE	----c-f		l		nb-op	-	
corr. MPI_TEST returning flag=FALSE	-------		l		nb-op	-	buffer cached on req
MPI_IBSEND	i-s----	nb	l	Ix2	nb-op	-	buffer
MPI_IRSEND	i-s----	nb	l	Ix2	nb-op	-	buffer 2)
corresponding MPI_Wait	----c-f		**l**		nb-op	-	3)
corr. MPI_TEST returning flag=TRUE	----c-f		l		nb-op	-	
corr. MPI_TEST returning flag=FALSE	-------		l		nb-op	-	buffer 7)
MPI_PROBE	i-s-c-f	b	nl		b-op	-	
MPI_IPROBE	i-s-c-f	**b**	**l**	**Im**	b-op	-	
MPI_RECV of a probed message	i-s-c-f	**b**	**l**		b-op	-	
MPI_IRECV of a probed message	i-s----	nb	l	Ix2	nb-op	-	buffer
corresponding MPI_Wait	----c-f		l		nb-op	-	3)
corr. MPI_TEST returning flag=TRUE	----c-f		l		nb-op	-	
corr. MPI_TEST returning flag=FALSE	-------		l		nb-op	-	buffer 7)
MPI_MPROBE	i-s-c-f	**nb**	**nl**		b-op	-	the message itself **8)**
MPI_IMPROBE	i-s-c-f	nb	l	Ix2	b-op	-	the message itself
MPI_MRECV of a probed message	i-s-c-f	**b**	**l**		b-op	-	
MPI_IMRECV of a probed message	i-s----	nb	l	Ix2	nb-op	-	buffer
corresponding MPI_Wait	----c-f		l		nb-op	-	3)
corr. MPI_TEST returning flag=TRUE	----c-f		l		nb-op	-	
corr. MPI_TEST returning flag=FALSE	-------		l		nb-op	-	buffer 7)
MPI_(-|S|R)SEND_INIT, MPI_RECV_INIT	i------	nb	l		p-op	-	buffer address
corresponding MPI_START, MPI_STARTALL	--s----	nb	l		p-op	-	buffer address+content 7)
corresponding MPI_Wait	----c--		nl		p-op	-	buffer address 7)
corr. MPI_TEST returning flag=TRUE	----c--		l		p-op	-	buffer address 7)
corr. MPI_TEST returning flag=FALSE	-------		l		p-op	-	buffer content+address 7)
corr. MPI_REQUEST_FREE (for inactive req-handle)	------f		l		p-op	-	

Procedure	Stages	Blk	Loc	Ixx	Op	Collective C sq	Collective S1/2	Blocked resources and remarks
MPI_BSEND_INIT	i------	nb	l		p-op	-		buffer address 9)
corresponding MPI_START, MPI_STARTALL	--s----	nb	l		p-op	-		buffer address+content 7)
corresponding MPI_Wait	----c--		l		p-op	-		buffer address 7,9)
corr. MPI_TEST returning flag=TRUE	----c--		l		p-op	-		buffer address 7,9)
corr. MPI_TEST returning flag=FALSE	-------		l		p-op	-		buffer address+content 7)
corr. MPI_REQUEST_FREE (for inactive req-handle)	------f		l		p-op	-		
MPI_CANCEL of nonblock./persistent pt-to-pt			l		p-op	-		
MPI_SENDRECV(_REPLACE)	i-s-c-f	b	nl		b-op	-		
MPI_BCAST and others	i-s-c-f	b	nl		b-op	C sq	S1	4)
MPI_IBCAST and others	i-s----	nb	l	Ix2	nb-op	C sq		buffer 4)
MPI_IGATHER**V** and other **…V / …W**	i-s----	nb	l	Ix2	nb-op	C sq		buffer, array arguments
corresponding MPI_Wait	----c--		nl		nb-op	C	S2	4)
corr. MPI_TEST returning flag=TRUE	----c--		l		nb-op	C	S2	4)
corr. MPI_TEST returning flag=FALSE	-------		l		nb-op			buffer, array arguments 7)
MPI_BCAST_INIT and others	i------	**nb**	**nl**		p-op	C sq	S1	buffer address **8)** 9)
MPI_GATHER**V**_INIT and other **…V/…W**_INIT	i-s----	**nb**	**nl**		p-op	C sq	S1	buffer address, array arguments **8)** 9)
corresponding MPI_START, MPI_STARTALL	--s----	nb	l		p-op	C		buffer addr.+content, 4,7)
corresponding MPI_Wait	----c--		nl		p-op	C	S2	buffer address and array arguments cached on the reqest handle, 4,7,9)
corr. MPI_TEST returning flag=TRUE	----c--		l		p-op	C	S2	buf-addr&arr-args 4,7,9)
corr. MPI_TEST returning flag=FALSE	-------		l		p-op			buf addr+content&arr-args 7)
corr. MPI_REQUEST_FREE	------f		**nl**		p-op	C sq	S1	5)
MPI_COMM_CREATE	i-s-c--	b	nl		b-op	C sq	S1	coll. over comm arg.
MPI_COMM_CREATE_GROUP	i-s-c--	b	nl		b-op	C sq	S1	coll. over group arg.
MPI_COMM_DUP, MPI_COMM_DUP_WITH_INFO, MPI_COMM_SPLIT, MPI_COMM_SPLIT_TYPE, MPI_CART_CREATE, MPI_GRAPH_CREATE, MPI_DIST_GRAPH_CREATE_ADJACENT, MPI_DIST_GRAPH_CREATE, MPI_CART_SUB: see MPI_COMM_CREATE								
MPI_INTERCOMM_CREATE, MPI_INTERCOMM_MERGE	i-s-c--	b	nl		b-op	C sq	S1	coll. over union of local & remote group
MPI_COMM_IDUP	i-s----	nb	l	Ix2	nb-op	C sq		communicator handle
corresponding MPI_Wait	----c--		nl		nb-op	C	S2	
corr. MPI_TEST returning flag=TRUE	----c--		l		nb-op	C	S2	
corr. MPI_TEST returning flag=FALSE	-------		l		nb-op			
MPI_COMM_FREE	------f	b	nl		b-op	C sq	S1	see, 6.4.3, Adv. to impl.
MPI_INIT, MPI_INIT_THREAD	i-s-c-f	b	nl		b-op	C sq	S1	collective over MPI_COMM_WORLD
MPI_FINALIZE	i-s-c-f	b	nl		b-op	C sq	S1	collective over all connected processes
MPI_COMM_SPAWN, …._MULTIPLE	i-s-c-f	b	nl		b-op	C sq	S1	collective over comm, 6)
MPI_COMM_ACCEPT, MPI_COMM_CONNECT	i-s-c-f	b	nl		b-op	C sq	S1	collective over comm
One-sided procedures								See corresponding chapter
MPI_FILE_READ/WRITE[_AT|SHARED], MPI_FILE_DELETE/SEEK/GET_VIEW	i-s-c-f	**b**	**l**		b-op	-		
MPI_FILE_READ/WRITE[_AT]**_[ALL|ORDERED]**, MPI_FILE_OPEN/CLOSE/SEEK_SHARED, MPI_FILE_PREALLOCATE/SYNC, MPI_FILE_SET_VIEW/SIZE/INFO/ATOMICITY	i-s-c-f	b	nl		b-op	C sq	S1	
MPI_FILE_**I**READ/**I**WRITE[_AT|SHARED]	i-s----	nb	l	Ix2	nb-op	-		buffer 10a)
MPI_FILE_**I**READ/**I**WRITE[_AT]**_ALL**	i-s----	nb	l	Ix2	nb-op	C sq		buffer 10b)
MPI_FILE_READ/WRITE[_AT]_ALL|ORDERED**_BEGIN**	i-s----	**nb**	**nl**		b-op	C sq	S1	buffer **8)**
MPI_FILE_READ/WRITE[_AT]_ALL|ORDERED**_END**	----c-f	b	nl		b-op	C sq	S1	