{void Op::Init(User_function* function, bool commute) *(binding deprecated,*
        see Section 15.2) }

{bool Op::Is_commutative() const *(binding deprecated, see Section 15.2)* }

{void Comm::Reduce(const void* sendbuf, void* recvbuf, int count,
        const Datatype& datatype, const Op& op, int root) const = 0
        (binding deprecated, see Section 15.2) }

{void Op::Reduce_local(const void* inbuf, void* inoutbuf, int count,
        const Datatype& datatype) const *(binding deprecated, see*
        Section 15.2) }

{void Comm::Reduce_scatter(const void* sendbuf, void* recvbuf,
        int recvcounts[], const Datatype& datatype, const Op& op)
        const = 0 *(binding deprecated, see Section 15.2)* }

{void Comm::Reduce_scatter_block(const void* sendbuf, void* recvbuf,
        int recvcount, const Datatype& datatype, const Op& op)
        const = 0 *(binding deprecated, see Section 15.2)* }

{void Intracomm::Scan(const void* sendbuf, void* recvbuf, int count,
        const Datatype& datatype, const Op& op) const *(binding*
        deprecated, see Section 15.2) }

{void Comm::Scatter(const void* sendbuf, int sendcount, const
        Datatype& sendtype, void* recvbuf, int recvcount,
        const Datatype& recvtype, int root) const = 0 *(binding*
        deprecated, see Section 15.2) }

{void Comm::Scatterv(const void* sendbuf, const int sendcounts[],
        const int displs[], const Datatype& sendtype, void* recvbuf,
        int recvcount, const Datatype& recvtype, int root) const = 0
        (binding deprecated, see Section 15.2) }

};

A.4.4 Groups, Contexts, Communicators, and Caching C++ Bindings

namespace MPI {

{Comm& Comm::Clone() const = 0 *(binding deprecated, see Section 15.2)* }

{Cartcomm& Cartcomm::Clone() const *(binding deprecated, see Section 15.2)* }

{Distgraphcomm& Distgraphcomm::Clone() const *(binding deprecated, see*
        Section 15.2) }

{Graphcomm& Graphcomm::Clone() const *(binding deprecated, see Section 15.2)* }

{Intercomm& Intercomm::Clone() const *(binding deprecated, see Section 15.2)* }

{Intracomm& Intracomm::Clone() const *(binding deprecated, see Section 15.2)* }

{static int Comm::Compare(const Comm& comm1, const Comm& comm2) *(binding*

[1] ticket150.
[2] ticket150.
[3]
[4] ticket150.
[5] ticket150.
[6] ticket150.
[7] ticket150.
[8]
[9] ticket150.
[10] ticket150.
[11]
[12] ticket150.
[13]
[14] ticket150.
[15]
[16] ticket150.
[17]
[18] ticket150.
[19] ticket150.
[20] ticket150.
[21]
[22]
[23] ticket150.
[24]
[25] ticket150.
[26]
[27] ticket150.
[28]
[29]
[30] ticket150.
[31]
[32]
[33]
[34]
[35]
[36]
[37]
[38] ticket150.
[39] ticket150.
[40] ticket150.
[41] ticket150.
[42] ticket150.
ticket150.
[43]
[44] ticket150.
ticket150.
[45]
[46] ticket150.
[47] ticket150.
[48] ticket150.
ticket150.
ticket150.

1   *deprecated, see Section 15.2)* }

2
ticket150. 3   {static int Group::Compare(const Group& group1, const Group& group2)
ticket150. 4   *(binding deprecated, see Section 15.2)* }

ticket150. 5   {Intracomm Intracomm::Create(const Group& group) const *(binding deprecated,*
ticket150. 6   *see Section 15.2)* }

7
ticket150. 8   {Intercomm Intercomm::Create(const Group& group) const *(binding deprecated,*
ticket150. 9   *see Section 15.2)* }

ticket150. 10   {Intercomm Intracomm::Create_intercomm(int local_leader, const
ticket150. 11   Comm& peer_comm, int remote_leader, int tag) const *(binding*
      12   *deprecated, see Section 15.2)* }

13
ticket150. 14   {static int Comm::Create_keyval(Comm::Copy_attr_function*
      15   comm_copy_attr_fn,
      16   Comm::Delete_attr_function* comm_delete_attr_fn,
ticket150. 17   void* extra_state) *(binding deprecated, see Section 15.2)* }

ticket150. 18   {static int Datatype::Create_keyval(Datatype::Copy_attr_function*
      19   type_copy_attr_fn, Datatype::Delete_attr_function*
ticket150. 20   type_delete_attr_fn, void* extra_state) *(binding deprecated, see*
      21   *Section 15.2)* }

22
ticket150. 23   {static int Win::Create_keyval(Win::Copy_attr_function* win_copy_attr_fn,
      24   Win::Delete_attr_function* win_delete_attr_fn,
ticket150. 25   void* extra_state) *(binding deprecated, see Section 15.2)* }

ticket150. 26   {void Comm::Delete_attr(int comm_keyval) *(binding deprecated, see Section 15.2)* }
ticket150. 27
ticket150. 28   {void Datatype::Delete_attr(int type_keyval) *(binding deprecated, see*
ticket150. 29   *Section 15.2)* }

30
ticket150. 31   {void Win::Delete_attr(int win_keyval) *(binding deprecated, see Section 15.2)* }
ticket150.
ticket150. 32   {static Group Group::Difference(const Group& group1, const Group& group2)
ticket150. 33   *(binding deprecated, see Section 15.2)* }

34
ticket150. 35   {Cartcomm Cartcomm::Dup() const *(binding deprecated, see Section 15.2)* }
ticket150.
ticket150. 36   {Distgraphcomm Distgraphcomm::Dup() const *(binding deprecated, see Section 15.2)*
ticket150. 37   }

38
ticket150. 39   {Graphcomm Graphcomm::Dup() const *(binding deprecated, see Section 15.2)* }
ticket150.
ticket150. 40   {Intercomm Intercomm::Dup() const *(binding deprecated, see Section 15.2)* }
ticket150. 41
ticket150. 42   {Intracomm Intracomm::Dup() const *(binding deprecated, see Section 15.2)* }
ticket150. 43   {Group Group::Excl(int n, const int ranks[]) const *(binding deprecated, see*
ticket150. 44   *Section 15.2)* }
ticket150. 45
ticket150. 46   {void Comm::Free() *(binding deprecated, see Section 15.2)* }

ticket150. 47   {void Group::Free() *(binding deprecated, see Section 15.2)* }
ticket150. 48
ticket150.