C/FORTRAN System Calls Summary

Contents

This section summarizes the iCC system calls. It is divided into two subsections:

- NX-like whole partition routines
- MPI-like group routines

Included are the calling sequences for both the C and FORTRAN versions and short descriptions of the operations.

Notational Conventions

Bold	Identifies system call names.
Italic	Identifies system call parameters.

Whole Partition Routines

Contents

This section summarizes the whole partition routines that comprised the original, NX-like iCC library.

C System Calls Summary

Table 1: iCC C System Calls

Synopsis	Description
void **iCC_barrier()**	Loose (faster) sychronization of all nodes.
void **iCC_bcast(** char *x*[], long *xlen*, long *iroot***)**	Global broadcast operation.
void **iCC_dg***x***high(** <any type> *x*[], <any type> *y*[], long *ylens*[])	Global vector MAX operation for (d)ouble, (i)nteger and (s)ingle (float) data types with result distributed over the nodes.
void **iCC_dg***x***low(** <any type> *x*[], <any type> *y*[], long *ylens*[])	Global vector MIN operation for (d)ouble, (i)nteger and (s)ingle (float) data types with result distributed over the nodes.
void **iCC_dg***x***prod(** <any type> *x*[], <any type> *y*[], long *ylens*[])	Global vector PRODUCT operation for (d)ouble, (i)nteger and (s)ingle (float) data types with result distributed over the nodes.
void **iCC_dg***x***sum(** <any type> *x*[], <any type> *y*[], long *ylens*[])	Global vector SUM operation for (d)ouble, (i)nteger and (s)ingle (float) data types with result distributed over the nodes.
void **iCC_dgiand(** long *x*[], long *y*[], long *ylens*[])	Global vector bitwise AND operation with result distributed over the nodes.
void **iCC_dgior(** long *x*[], long *y*[], long *ylens*[])	Global vector bitwise OR operation with result distributed over the nodes.
void **iCC_dgland(** long *x*[], long *y*[], long *ylens*[])	Global vector logical AND operation with result distributed over the nodes.
void **iCC_dglor(** long *x*[], long *y*[], long *ylens*[])	Global vector logical OR operation with result distributed over the nodes.

Table 1: iCC C System Calls

Synopsis	Description
void **iCC_dgopf(** char *x*[], long *size*, char *y*[], long *ylens*[], long (**f*)())	User-defined global vector operation with result distributed over the nodes.
void **iCC_gcol(** char *x*[], long *xlen*, char *y*[], long *ylen*, long **ncnt***)**	Global collect (concatenation) operation for contributions of unknown length.
void **iCC_gcolx(** char *x*[], long *xlens*[], char *y*[])	Global collect (concatenation) operation for contributions of known length.
void **iCC_g*x*high(** <any type> *x*[], long *xlen*, <any type> *y*[])	Global vector MAX operation for (d)ouble, (i)nteger and (s)ingle (float) data types with result on all nodes.
void **iCC_g*x*highto1(** <any type> *x*[], long *xlen*, <any type> *y*[], long *iroot***)**	Global vector MAX operation for (d)ouble, (i)nteger and (s)ingle (float) data types with result on one node.
void **iCC_g*x*low(** <any type> *x*[], long *xlen*, <any type> *y*[])	Global vector MIN operation for (d)ouble, (i)nteger and (s)ingle (float) data types with result on all nodes.
void **iCC_g*x*lowto1(** <any type> *x*[], long *xlen*, <any type> *y*[], long *iroot***)**	Global vector MIN operation for (d)ouble, (i)nteger and (s)ingle (float) data types with result on one node.
void **iCC_g*x*prod(** <any type> *x*[], long *xlen*, <any type> *y*[])	Global vector PRODUCT operation for (d)ouble, (i)nteger and (s)ingle (float) data types with result on all nodes.
void **iCC_g*x*prodto1(** <any type> *x*[], long *xlen*, <any type> *y*[], long *iroot***)**	Global vector PRODUCT operation for (d)ouble, (i)nteger and (s)ingle (float) data types with result on one node.

Table 1: iCC C System Calls

Synopsis	Description
void **iCC_g*x*sum(** <any type> *x*[], long *xlen*, <any type> *y*[])	Global vector SUM operation for (d)ouble, (i)nteger and (s)ingle (float) data types with result on all nodes.
void **iCC_g*x*sumto1(** <any type> *x*[], long *xlen*, <any type> *y*[], long *iroot*)	Global vector SUM operation for (d)ouble, (i)nteger and (s)ingle (float) data types with result on one node.
void **iCC_ggath(** char *x*[], long *xlen*, char *y*[], long *ylen*, long **ncnt*, long *iroot*)	Global gather operation for contributions of unknown length.
void **iCC_ggathx(** char *x*[], long *xlens*[], char *y*[], long *iroot*)	Global gather operation for contributions of known length.
void **iCC_giand(** long *x*[], long *xlen*, long *work*[])	Global vector bitwise AND operation with result on all nodes.
void **iCC_giandto1(** long *x*[], long *xlen*, long *y*[], long *iroot*)	Global vector bitwise AND operation with result on one node.
void **iCC_gior(** long *x*[], long *xlen*, long *y*[])	Global vector bitwise OR operation with result on all nodes.
void **iCC_giorto1(** long *x*[], long *xlen*, long *y*[], long *iroot*)	Global vector bitwise OR operation with result on one node.
void **iCC_gland(** long *x*[], long *xlen*, long *y*[])	Global vector logical AND operation with result on all nodes.

Table 1: iCC C System Calls

Synopsis	Description
void **iCC_glandto1(** long *x*[], long *xlen*, long *y*[], long *iroot***)**	Global vector logical AND operation with result on one node.
void **iCC_glor(** long *x*[], long *xlen*, long *y*[]**)**	Global vector logical OR operation.
void **iCC_glorto1(** long *x*[], long *xlen*, long *y*[], long *iroot***)**	Global vector logical OR operation with result on one node.
void **iCC_gopf(** char *x*[], long *xlen*, long *size*, char *y*[], long (**f*)()**)**	User-defined global vector operation with result on all nodes.
void **iCC_gopfto1(** char *x*[], long *xlen*, long *size*, char *y*[], long *iroot*, long (**f*)()**)**	User-defined global vector operation with result on one node.
void **iCC_grid_init(** long *nrows*, long *ncols***)**	Tell iCC the dimensions of the partition (Delta and Paragon only--iPSC version is a stub).
void **iCC_gscatx(** char *x*[], char *y*[], long *ylens*[], long *iroot***)**	Global scatter operation for portions of known length.
void **iCC_gsync()**	Tight (slower) sychronization of all nodes.
void **iCC_work_init(** unsigned long *size***)**	Set the size of the internal buffer used by iCC routines.

FORTRAN System Calls Summary

Table 2: iCC FORTRAN System Calls

Synopsis	Description
SUBROUTINE **iCC_BARRIER**()	Loose (faster) synchronization of all nodes.
SUBROUTINE **iCC_BCAST**(*x, xlen, iroot*) INTEGER *x*(*) INTEGER *xlen* INTEGER *iroot*	Global broadcast operation.
SUBROUTINE **iCC_DG*x*HIGH**(*x, y, ylens*) <any type> *x*(*) <any type> *y*(*) INTEGER *ylens*(*)	Global vector MAX operation for (D)OUBLE PRECISION, (I)NTEGER and (S)INGLE PRECISION data types with result distributed over the nodes.
SUBROUTINE **iCC_DG*x*LOW**(*x, y, ylens*) <any type> *x*(*) <any type> *y*(*) INTEGER *ylens*(*)	Global vector MIN operation for (D)OUBLE PRECISION, (I)NTEGER and (S)INGLE PRECISION data types with result distributed over the nodes.
SUBROUTINE **iCC_DG*x*PROD**(*x, y, ylens*) <any type> *x*(*) <any type> *y*(*) INTEGER *ylens*(*)	Global vector PRODUCT operation for (D)OUBLE PRECISION, (I)NTEGER and (S)INGLE PRECISION data types with result distributed over the nodes.
SUBROUTINE **iCC_DG*x*SUM**(*x, y, ylens*) <any type> *x*(*) <any type> *y*(*) INTEGER *ylens*(*)	Global vector SUM operation for (D)OUBLE PRECISION, (I)NTEGER and (S)INGLE PRECISION data types with result distributed over the nodes.
SUBROUTINE **iCC_DGIAND**(*x, y, ylens*) INTEGER *x*(*) INTEGER *y*(*) INTEGER *ylens*(*)	Global vector bitwise AND operation with result distributed over the nodes.
SUBROUTINE **iCC_DGIOR**(*x, y, ylens*) INTEGER *x*(*) INTEGER *y*(*) INTEGER *ylens*(*)	Global vector bitwise OR operation with result distributed over the nodes.
SUBROUTINE **iCC_DGLAND**(*x, y, ylens*) LOGICAL *x*(*) INTEGER *y*(*) LOGICAL *ylens*(*)	Global vector logical AND operation with result distributed over the nodes.
SUBROUTINE **iCC_DGLOR**(*x, y, ylens*) LOGICAL *x*(*) INTEGER *y*(*) LOGICAL *ylens*(*)	Global vector logical OR operation with result distributed over the nodes.

Table 2: iCC FORTRAN System Calls

Synopsis	Description
SUBROUTINE **iCC_DGOPF**(*x, size, y, ylens, f*) INTEGER *x*(*) INTEGER *size* INTEGER *y*(*) INTEGER *ylens*(*) EXTERNAL *f*	User-defined global vector operation with result distributed over the nodes.
SUBROUTINE **iCC_GCOL(***x, xlen, y, ylen, ncnt***)** INTEGER *x*(*) INTEGER *xlen* INTEGER *y*(*) INTEGER *ylen* INTEGER *ncnt*	Global collect (concatenation) operation for contributions of unknown length.
SUBROUTINE **iCC_GCOLX**(*x, xlens, y*) INTEGER *x*(*) INTEGER *xlens*(*) INTEGER *y*(*)	Global collect (concatenation) operation for contributions of known length.
SUBROUTINE **iCC_G***x***HIGH**(*x, xlen, y*) <any type> *x*(*) INTEGER *xlen* <any type> *y*(*)	Global vector MAX operation for (D)OUBLE PRECISION, (I)NTEGER and (S)INGLE PRECISION data types with result on all nodes.
SUBROUTINE **iCC_G***x***HIGHTO1**(*x, xlen, y, iroot*) <any type> *x*(*) INTEGER *xlen* <any type> *y*(*) INTEGER *iroot*	Global vector MAX operation for (D)OUBLE PRECISION, (I)NTEGER and (S)INGLE PRECISION data types with result on one node.
SUBROUTINE **iCC_G***x***LOW**(*x, xlen, y*) <any type> *x*(*) INTEGER *xlen* <any type> *y*(*)	Global vector MIN operation for (D)OUBLE PRECISION, (I)NTEGER and (S)INGLE PRECISION data types with result on all nodes.
SUBROUTINE **iCC_G***x***LOWTO1**(*x, xlen, y, iroot*) <any type> *x*(*) INTEGER *xlen* <any type> *y*(*) INTEGER *iroot*	Global vector MIN operation for (D)OUBLE PRECISION, (I)NTEGER and (S)INGLE PRECISION data types with result on one node.
SUBROUTINE **iCC_G***x***PROD**(*x, xlen, y*) <any type> *x*(*) INTEGER *xlen* <any type> *y*(*)	Global vector PRODUCT operation for (D)OUBLE PRECISION, (I)NTEGER and (S)INGLE PRECISION data types with result on all nodes.
SUBROUTINE **iCC_G***x***PRODTO1**(*x, xlen, y, iroot*) <any type> *x*(*) INTEGER *xlen* <any type> *y*(*) INTEGER *iroot*	Global vector PRODUCT operation for (D)OUBLE PRECISION, (I)NTEGER and (S)INGLE PRECISION data types with result on one node.

Table 2: iCC FORTRAN System Calls

Synopsis	Description
SUBROUTINE **iCC_G*x*SUM**(*x, xlen, y*) <any type> *x*(*) INTEGER *xlen* <any type> *y*(*)	Global vector SUM operation for (D)OUBLE PRECISION, (I)NTEGER and (S)INGLE PRECISION data types with result on all nodes.
SUBROUTINE **iCC_G*x*SUMTO1**(*x, xlen, y, iroot*) <any type> *x*(*) INTEGER *xlen* <any type> *y*(*) INTEGER *iroot*	Global vector SUM operation for (D)OUBLE PRECISION, (I)NTEGER and (S)INGLE PRECISION data types with result on one node.
SUBROUTINE **iCC_GGATH**(*x, xlen, y, ylen, ncnt, iroot*) INTEGER *x*(*) INTEGER *xlen* INTEGER *y*(*) INTEGER *ylen* INTEGER *ncnt* INTEGER *iroot*	Global gather operation for contributions of unknown length.
SUBROUTINE **iCC_GGATHX**(*x, xlens, y, iroot*) INTEGER *x*(*) INTEGER *xlens*(*) INTEGER *y*(*) INTEGER *iroot*	Global gather operation for contributions of known length.
SUBROUTINE **iCC_GIAND**(*x, xlen, y*) INTEGER *x*(*) INTEGER *xlen* INTEGER *y*(*)	Global vector bitwise AND operation with result on all nodes.
SUBROUTINE **iCC_GIANDTO1**(*x, xlen, y, iroot*) INTEGER *x*(*) INTEGER *xlen* INTEGER *y*(*) INTEGER *iroot*	Global vector bitwise AND operation with result on one node.
SUBROUTINE **iCC_GIOR**(*x, xlen, y*) INTEGER *x*(*) INTEGER *xlen* INTEGER *y*(*)	Global vector bitwise OR operation with result on all nodes.
SUBROUTINE **iCC_GIORTO1**(*x, xlen, y, iroot*) INTEGER *x*(*) INTEGER *xlen* INTEGER *y*(*) INTEGER *iroot*	Global vector bitwise OR operation with result on one node.
SUBROUTINE **iCC_GLAND**(*x, xlen, y*) LOGICAL *x*(*) INTEGER *xlen* LOGICAL *y*(*)	Global vector logical AND operation with result on all nodes.

Table 2: iCC FORTRAN System Calls

Synopsis	Description
SUBROUTINE **iCC_GLANDTO1**(*x, xlen, y, iroot*) LOGICAL *x*(*) INTEGER *xlen* LOGICAL *y*(*) INTEGER *iroot*	Global vector logical AND operation with result on one node.
SUBROUTINE **iCC_GLOR**(*x, xlen, y*) LOGICAL *x*(*) INTEGER *n* LOGICAL *y*(*)	Global vector logical OR operation.
SUBROUTINE **iCC_GLORTO1**(*x, xlen, y, iroot*) LOGICAL *x*(*) INTEGER *xlen* LOGICAL *y*(*) INTEGER *iroot*	Global vector logical OR operation with result on one node.
SUBROUTINE **iCC_GOPF**(*x, xlen, size, y, f*) INTEGER *x*(*) INTEGER *xlen* INTEGER *size* INTEGER *y*(*) EXTERNAL *f*	User-defined global vector operation with result on all nodes.
SUBROUTINE **iCC_GOPFTO1**(*x, xlen, size, y, iroot, f*) INTEGER *x*(*) INTEGER *xlen* INTEGER *size* INTEGER *y*(*) INTEGER *iroot* EXTERNAL *f*	User-defined global vector operation with result on one node.
SUBROUTINE **iCC_GRID_INIT**(*nrows, ncols*) INTEGER *nrows* INTEGER *ncols*	Tell iCC the dimensions of the partition (Delta and Paragon only--iPSC version is a stub).
SUBROUTINE **iCC_GSCATX**(*x, y, ylens, iroot*) INTEGER *x*(*) INTEGER *y*(*) INTEGER *ylens*(*) INTEGER *iroot*	Global scatter operation for portions of known length.
SUBROUTINE **iCC_GSYNC**()	Tight (slower) synchronization of all nodes.
SUBROUTINE **iCC_WORK_INIT**(*size*) INTEGER *size*	Set the size of the internal buffer used by iCC routines.

Interprocessor Collective Communications (iCC) Library

A.2

Group Routines

Contents

This section summarizes the MPI-like group routines that have been added in the latest iCC library.

C System Calls Summary

Table 3: iCC C System Calls

Synopsis	Description
long **iCC_g_Allgatherv(** void **sendbuf*, long *sendcount*, iCC_g_Datatype *sendtype*, void **recvbuf*, long **recvcounts*, long **displs*, iCC_g_Datatype *recvtype*, iCC_g_Comm *comm***)**	Gathers (collects) the send buffers from all the nodes associated with the given communicator into the receive buffers on all the nodes in that group.
long **iCC_g_Allreduce(** void **sendbuf*, void **recvbuf*, long *count*, iCC_g_Datatype *datatype*, iCC_g_Op *op*, iCC_g_Comm *comm*)	Uses the specified iCC operator to reduce the send buffers of all the nodes associated with the given communicator into the receive buffer on every node in that group.
long **iCC_g_Barrier(** iCC_g_Comm *comm*)	Loose (faster) synchronization of all nodes associated with the given communicator.
long **iCC_g_Bcast(** void **buffer*, long *count*, iCC_g_Datatype *datatype*, long *root*, iCC_g_Comm *comm*)	Broadcast buffer from the specified node to all nodes associated with the given communicator.
long **iCC_g_Comm_create(** iCC_g_Comm *comm,* iCC_g_Group *group,* iCC_g_Comm **newcomm*)	Create a new communicator formed from a subgroup of the group associated with the given communicator.
long **iCC_g_Comm_dup(** iCC_g_Comm *comm,* iCC_g_Comm **newcomm*)	Creates a duplicate of the given communicator.
long **iCC_g_Comm_free(** iCC_g_Comm **comm***)**	Marks a communicator for deallocation.
long **iCC_g_Comm_grid(** iCC_g_Comm *comm,* long **nrow,* long **ncol,* long **myrow,* long **mycol***)**	Returns the logical grid and row/column for the calling node that iCC has determined for the group associated with the given communicator.

Table 3: iCC C System Calls

Synopsis	Description
long **iCC_g_Comm_group(** iCC_g_Comm *comm,* iCC_g_Group **group*)	Returns the group handle associated with the given communicator.
long **iCC_g_Comm_rank(** iCC_g_Comm *comm,* long **rank*)	Returns the rank of the calling node within the group associated with the given communicator.
long **iCC_g_Comm_size(** iCC_g_Comm *comm,* long **size*)	Returns the size of the group associated with the given communicator.
void **iCC_g_Debug(** int *level*)	Set the debug level for iCC group routines.
long **iCC_g_Gatherv(** void **sendbuf*, long *sendcount*, iCC_g_Datatype *sendtype*, void **recvbuf*, long **recvcounts*, long **displs*, iCC_g_Datatype *recvtype*, long *root*, iCC_g_Comm *comm*)	Gathers the send buffers from all the nodes assocated with the given communicator into the receive buffer on the specified node.
long **iCC_g_Group_block_incl(** iCC_g_Group *group,* long *startrow,* long *startcol,* long *nrow,* long *rowstride,* long *ncol,* long *colstride,* iCC_g_Group **newgroup*)	Creates a subgroup of nodes specified by rows and columns with the given parent group.
long **iCC_g_Group_finalize**()	Cleanup and shut down iCC group capabilities.
long **iCC_g_Group_free(** iCC_g_Group **group*)	Marks a group for deallocation.
long **iCC_g_Group_grid(** iCC_g_Group *group,* long **nrow,* long **ncol,* long **myrow,* long **mycol*)	Returns the logical grid and row/column for the calling node that iCC has determined for the given group.

Table 3: iCC C System Calls

Synopsis	Description
long **iCC_g_Group_incl(** iCC_g_Group *group,* long *n,* long **ranks,* iCC_g_Group **newgroup*)	Creates a subgroup of processes specified by ranks with the given parent group.
long **iCC_g_Group_info(** FILE **fp*, iCC_g_Group *group*)	Prints useful information about the given group.
long **iCC_g_Group_init**()	Initialize the iCC group capabilities.
long **iCC_g_Group_rank(** iCC_g_Group *group,* long **rank*)	Returns calling node's rank in the given group.
long **iCC_g_Group_size(** iCC_g_Group *group,* long **size*)	Returns the number of nodes in given group.
long **iCC_g_Op_create(** iCC_g_Uop **uop*, iCC_g_Datatype *datatype*, long *commute*, iCC_g_Op *op*)	Binds an iCC operator to the given function.
long **iCC_g_Op_free(** iCC_g_Op **op*)	Deallocates an iCC operator.
long **iCC_g_Reduce(** void **sendbuf*, void **recvbuf*, long *count*, iCC_g_Datatype *datatype*, iCC_g_Op *op*, long *root*, iCC_g_Comm *comm*)	Uses the specified iCC operator to reduce the send buffers of all the nodes assocated with the given communicator into the receive buffer on the root node of that group.
long **iCC_g_Reduce_scatter(** void **sendbuf*, void **recvbuf*, long **recvcounts*, iCC_g_Datatype *datatype*, iCC_g_Op *op*, iCC_g_Comm *comm*)	Uses the specified iCC operator to reduce the send buffers of all the nodes assocated with the given communicator, leaving the result distributed in the receive buffers of the nodes in that group.

Table 3: iCC C System Calls

Synopsis	Description
long **iCC_g_Scatterv(** void **sendbuf*, long **sendcounts*, long **displs*, iCC_g_Datatype *sendtype*, void **recvbuf*, long *recvcount*, iCC_g_Datatype *recvtype*, long *root*, iCC_g_Comm *comm***)**	Scatters portions of the send buffer on the root node into the receive buffers of all the nodes assocated with the given communicator.
long **iCC_g_Sync(** iCC_g_Comm *comm***)**	Tight (slower) synchronization of all nodes associated with the given communicator.

FORTRAN System Calls Summary

Table 4: iCC FORTRAN System Calls

Synopsis	Description
SUBROUTINE **iCC_g_ALLGATHERV(***sendbuf, sendcount, sendtype, recvbuf, recvcounts, displs, recvtype, comm, ierror***)** <any type> *sendbuf*(*) INTEGER *sendcount* INTEGER *sendtype* <any type> *recvbuf*(*) INTEGER *recvcounts* INTEGER *displs* INTEGER *recvtype* INTEGER *comm* INTEGER *ierror*	Gathers (collects) the send buffers from all the nodes associated with the given communicator into the receive buffers on all the nodes in that group.
SUBROUTINE **iCC_g_ALLREDUCE(***sendbuf, recvbuf, count, datatype, op, comm, ierror***)** <any type> *sendbuf*(*) <any type> *recvbuf*(*) INTEGER *count* INTEGER *datatype* INTEGER *op* INTEGER *comm* INTEGER *ierror*	Uses the specified iCC operator to reduce the send buffers of all the nodes associated with the given communicator into the receive buffer on every node in that group.
SUBROUTINE **iCC_g_BARRIER(***comm, ierror***)** INTEGER *comm* INTEGER *ierror*	Loose (faster) synchronization of all nodes associated with the given communicator.
SUBROUTINE **iCC_g_BCAST(***buffer, count, datatype, root, comm, ierror***)** <any type> *buffer*(*) INTEGER *count* INTEGER *datatype* INTEGER *root* INTEGER *comm* INTEGER *ierror*	Broadcast buffer from the specified node to all nodes associated with the given communicator.
SUBROUTINE **iCC_g_COMM_CREATE(***comm, group, newcomm, ierror***)** INTEGER *comm* INTEGER *group* INTEGER *newcomm* INTEGER *ierror*	Create a new communicator formed from a subgroup of the group associated with the given communicator.

Table 4: iCC FORTRAN System Calls

Synopsis	Description
SUBROUTINE **iCC_g_COMM_DUP**(*comm, newcomm, ierror*) INTEGER *comm* INTEGER *newcomm* INTEGER *ierror*	Creates a duplicate of the given communicator.
SUBROUTINE **iCC_g_COMM_FREE**(*comm, ierror*) INTEGER *comm* INTEGER *ierror*	Marks a communicator for deallocation.
SUBROUTINE **iCC_g_COMM_GRID**(*comm, nrow, ncol, myrow, mycol, ierror*) INTEGER *comm* INTEGER *nrow* INTEGER *ncol* INTEGER *myrow* INTEGER *mycol* INTEGER *ierror*	Returns the logical grid and row/column for the calling node that iCC has determined for the group associated with the given communicator.
SUBROUTINE **iCC_g_COMM_GROUP**(*comm, group, ierror*) INTEGER *comm* INTEGER *group* INTEGER *ierror*	Returns the group handle associated with the given communicator.
SUBROUTINE **iCC_g_COMM_RANK**(*comm, rank, ierror*) INTEGER *comm* INTEGER *rank* INTEGER *ierror*	Returns the rank of the calling node within the group associated with the given communicator.
SUBROUTINE **iCC_g_COMM_SIZE**(*comm, size, ierror*) INTEGER *comm* INTEGER *size* INTEGER *ierror*	Returns the size of the group associated with the given communicator.
SUBROUTINE **iCC_g_DEBUG**(*level*) INTEGER *level*	Set the debug level for iCC group routines.

Table 4: iCC FORTRAN System Calls

Synopsis	Description
SUBROUTINE **iCC_g_GATHERV(***sendbuf, sendcount, sendtype, recvbuf, recvcounts, displs, recvtype, root, comm, ierror***)** <any type> *sendbuf*(*) INTEGER *sendcount* INTEGER *sendtype* <any type> *recvbuf*(*) INTEGER *recvcounts*(*) INTEGER *displs*(*) INTEGER *recvtype* INTEGER *root* INTEGER *comm* INTEGER *ierror*	Gathers the send buffers from all the nodes assocated with the given communicator into the receive buffer on the specified node.
SUBROUTINE **iCC_g_GROUP_BLOCK_INCL(***group, startrow, startcol, nrow, rowstride, ncol, colstride, newgroup, ierror***)** INTEGER *group* INTEGER *startrow* INTEGER *startcol* INTEGER *nrow* INTEGER *rowstride* INTEGER *ncol* INTEGER *colstride* INTEGER *newgroup* INTEGER *ierror*	Creates a subgroup of nodes specified by rows and columns with the given parent group.
SUBROUTINE **iCC_g_GROUP_FINALIZE(***ierror***)** INTEGER *ierror*	Cleanup and shut down iCC group capabilities.
SUBROUTINE **iCC_g_GROUP_FREE(***group, ierror***)** INTEGER *group* INTEGER *ierror*	Marks a group for deallocation.
SUBROUTINE **iCC_g_GROUP_GRID(***group, nrow, ncol, myrow, mycol, ierror***)** INTEGER *group* INTEGER *nrow* INTEGER *ncol* INTEGER *myrow* INTEGER *mycol* INTEGER *ierror*	Returns the logical grid and row/column for the calling node that iCC has determined for the given group.

Table 4: iCC FORTRAN System Calls

Synopsis	Description
SUBROUTINE **iCC_g_GROUP_INCL**(*group, n, ranks, newgroup, ierror*) INTEGER *group* INTEGER *n* INTEGER *ranks*(*) INTEGER *newgroup* INTEGER *ierror*	Creates a subgroup of processes specified by ranks with the given parent group.
SUBROUTINE **iCC_g_GROUP_INFO**(*group*) INTEGER *group*	Prints useful information about the given group.
SUBROUTINE **iCC_g_GROUP_INIT**(*ierror*) INTEGER *ierror*	Initialize the iCC group capabilities.
SUBROUTINE **iCC_g_GROUP_RANK**(*group, rank, ierror*) INTEGER *group* INTEGER *rank* INTEGER *ierror*	Returns calling node's rank in the given group.
SUBROUTINE **iCC_g_GROUP_SIZE**(*group, size, ierror*) INTEGER *group* INTEGER *size* INTEGER *ierror*	Returns the number of nodes in given group.
SUBROUTINE **iCC_g_OP_CREATE**(*uop, datatype, commute, op, ierror*) EXTERNAL *uop* INTEGER *datatype* LOGICAL *commute* INTEGER *op* INTEGER *ierror*	Binds an iCC operator to the given function.
SUBROUTINE **iCC_g_OP_FREE**(*op, ierror*) INTEGER *op* INTEGER *ierror*	Deallocates an iCC operator.
SUBROUTINE **iCC_g_REDUCE**(*sendbuf, recvbuf, count, datatype, op, root, comm, ierror*) <any type> *sendbuf*(*) <any type> *recvbuf*(*) INTEGER *count* INTEGER *datatype* INTEGER *op* INTEGER *root* INTEGER *comm* INTEGER *ierror*	Uses the specified iCC operator to reduce the send buffers of all the nodes assocated with the given communicator into the receive buffer on the root node of that group.

Table 4: iCC FORTRAN System Calls

Synopsis	Description
SUBROUTINE **iCC_g_REDUCE_SCATTER(***sendbuf, recvbuf, recvcounts, datatype, op, comm, ierror***)** <any type> *sendbuf*(*) <any type> *recvbuf*(*) INTEGER *recvcounts*(*) INTEGER *datatype* INTEGER *op* INTEGER *comm* INTEGER *ierror*	Uses the specified iCC operator to reduce the send buffers of all the nodes assocated with the given communicator, leaving the result distributed in the receive buffers of the nodes in that group.
SUBROUTINE **iCC_g_SCATTERV(***sendbuf, sendcounts, displs, sendtype, recvbuf, recvcount, recvtype, root, comm, ierror***)** <any type> *sendbuf*(*) INTEGER *sendcounts*(*) INTEGER *displs*(*) INTEGER *sendtype* <any type> *recvbuf*(*) INTEGER *recvcount* INTEGER *recvtype* INTEGER *root* INTEGER *comm* INTEGER *ierror*	Scatters portions of the send buffer on the root node into the receive buffers of all the nodes assocated with the given communicator.
SUBROUTINE **iCC_g_SYNC(***comm, ierror***)** INTEGER *comm* INTEGER *ierror*	Tight (slower) synchronization of all nodes associated with the given communicator.