IdnList Objects

Introduction

An **IdnList** object manages a list of SERCOS Idn objects.

Methods

Create, Delete, Validate Methods

mpiIdnList**Create**	Create IdnList object
mpiIdnList**Delete**	Delete IdnList object
mpiIdnList**Validate**	Validate IdnList object

Configuration and Information Methods

mpiIdnList**Copy**	Copy the contents of idnList to *dst*.
mpiIdnList**IdnNumber**	Read and write the identification numbers of an IdnList object.

Relational Methods

List Methods- for Idn Objects

mpiIdnList**Append**	Append an Idn to an IdnList
mpiIdnList**Count**	Count the number of Idns in an IdnList
mpiIdnList**First**	Get handle to first Idn in an IdnList
mpiIdnList**Idn**	Return the position of the element on the list indicated by "index."
mpiIdnList**Index**	Get index value to an Idn in an IdnList
mpiIdnList**Insert**	Insert an Idn into an IdnList
mpiIdnList**Last**	Get handle to last Idn in an IdnList
mpiIdnList**Get**	Get array of handles to Idns in an IdnList
mpiIdnList**Set**	Create a list (IdnList) containing idnCount Idn objects
mpiIdnList**Next**	Get handle to the Idn object just after a specified Idn object in an IdnList
mpiIdnList**IdnNumberFind**	
mpiIdnList**Previous**	Get handle to the Idn object preceding a specified Idn object in an IdnList
mpiIdnList**Remove**	Remove an Idn object from an IdnList

mpiIdnListCreate

Declaration

```
const MPIIdnList mpiIdnListCreate(MPIIdn idn)
```

Required Header stdmpi.h

Description

IdnListCreate creates an IdnList object, where the ***idn*** argument specifies the initial element in the list of Idn objects. Note that ***idn*** may be MPIHandleVOID. *IdnListCreate* is the equivalent of a C++ constructor.

Return Values	
handle	to an IdnList object
MPIHandleVOID	if the object could not be created

See Also mpiIdnListDelete | mpiIdnListValidate

mpiIdnListDelete

Declaration

```
long mpiIdnListDelete(MPIIdnList idnList)
```

Required Header stdmpi.h

Description **IdnListDelete** deletes an IdnList object and invalidates its handle (***idnList***). Note that deleting an IdnList object **will also delete all of the Idn objects that it contains**. *IdnListDelete* is the equivalent of a C++ destructor.

Return Values	
MPIMessageOK	if *IdnListDelete* successfully deletes the IdnList object and invalidates its handle

See Also mpiIdnListCreate | mpiIdnListValidate

mpiIdnListValidate

Declaration `long mpiIdnListValidate(MPIIdnList idnList)`

Required Header stdmpi.h

Description **IdnListValidate** validates the IdnList object and its handle (***idnList***).

Return Values	
MPIMessageOK	if IdnList is a handle to a valid object.

See Also mpiIdnListCreate | mpiIdnListDelete

mpiIdnListCopy

Declaration

```
long mpiIdnListCopy(MPIIdnList idnList,
                    MPIIdnList dst)
```

Required Header

stdmpi.h

Description

IdnListCopy copies the contents of ***idnList*** to ***dst***.

Return Values

MPIMessageOK	if *IdnListCopy* successful

See Also

mpiIdnListIdnNumber

Declaration

```
long mpiIdnListIdnNumber(MPIIdnList    idnList,
                         MPIIdnNumber  *number)
```

Required Header

stdmpi.h

Description

IdnListIdnNumber reads the identification numbers of an IdnList object (***idnList***) and writes it to an array of MPIIdnNumbers pointed to by ***number***.

idnList	a handle to the IdnList object.
***number**	a pointer to an array of MPIIdnNumbers returned by the method.

Return Values

MPIMessageOK	if *IdnListIdnNumber* successfully gets the list of identification numbers.

See Also

mpiIdnNumberGET

mpiIdnListAppend

Declaration

```
long mpiIdnListAppend(MPIIdnList idnList,
                      MPIIdn     idn)
```

Required Header

stdmpi.h

Description

IdnListAppend appends an Idn (***idn***) to an IdnList (***idnList***).

idnList	a handle to the IdnList object.
idn	a handle to an Idn object.

Return Values

MPIMessageOK	if *IdnListAppend* successfully appends the Idn to the IdnList
MPIMessageHANDLE_INVALID	Either ***idnList*** or ***idn*** is an invalid handle.
MPIMessageNO_MEMORY	Not enough memory was available.

See Also

mpiIdnListCount

Declaration

```
long mpiIdnListCount(MPIIdnList idnList)
```

Required Header

stdmpi.h

Description

IdnListCount returns the number of elements on the list.

idnList	a handle to the IdnList object.

Return Values

count	the number of elements in idnList
(-1)	if the idnList object is invalid

See Also

mpiIdnListFirst

Declaration `MPIIdn mpiIdnListFirst(MPIIdnList idnList)`

Required Header stdmpi.h

Description **IdnListFirst** return the first element in the list. This function can be used in conjuntion with mpiIdnListIdnNext() in order to iterate through the list.

idnList	a handle to the IdnList object.

Return Values	
handle	to the first Idn object of an IdnList (***idnList***)
MPIHandleVOID	if ***idnList*** is invalid if ***idnList*** is empty
MPIMessageHANDLE_INVALID	***idnList*** is an invalid handle.

See Also mpiIdnListLast

mpiIdnListIdn

Declaration

```
MPIIdn mpiIdnListIdn(MPIIdnList idnList,
                     long       index,)
```

Required Header

stdmpi.h

Description

IdnListIdn returns the position of the element on the list indicated by "index."

idnList	a handle to the IdnList object.
index	a position in the list.

Return Values

MPIMessageARG_INVALID	if ***index*** is a negative number.
MEIListMessageELEMENT_NOT_FOUND	if ***index*** is greater than or equal to the number of elements in the list.
MPIMessageHANDLE_INVALID	if ***idnList*** is an invalid handle.

See Also

mpiIdnListIndex

Declaration

```
long mpiIdnListIndex(MPIIdnList idnList,
                     MPIIdn     idn)
```

Required Header

stdmpi.h

Description

IdnListIndex returns the position of "idn" on the list.

idnList	a handle to the IdnList object.
idn	a handle to an Idn object.

Return Values

index	of an Idn (***idn***) in an IdnList (***idnList***)
MPIHandleVOID	if ***idnList*** is invalid if ***idnList*** is empty

See Also

mpiIdnListInsert

Declaration

```
long mpiIdnListInsert(MPIIdnList idnList,
                      MPIIdn     idn,
                      MPIIdn     insert)
```

Required Header

stdmpi.h

Description

IdnListInsert inserts an Idn object (***insert***) in an IdnList (***idnList***), just after this Idn object (***idn***).

Return Values	
MPIMessageOK	if *IdnListInsert* successfully inserts the Idn into the IdnList, in the slot after the specified Idn (***idn***)

See Also

mpiIdnListLast

Declaration

```
MPIIdn mpiIdnListLast(MPIIdnList idnList)
```

Required Header

stdmpi.h

Description

The last element in the list is returned. This function can be used in conjuntion with mpiIdnListIdnPrevious() in order to iterate through the list backwards.

idnList	a handle to the IdnList object.

Return Values

handle	to the last Idn object of an IdnList (***idnList***)
MPIHandleVOID	if ***idnList*** is invalid if ***idnList*** is empty
MPIMessageHANDLE_INVALID	if ***idnList*** is an invalid handle.

See Also

mpiIdnListFirst

mpiIdnListGet

Declaration

```
long mpiIdnListGet(MPIIdnList idnList,
                   long       *idnCount,
                   MPIIdn     *idnArray)
```

Required Header

stdmpi.h

Description

IdnListGet gets an array (that contains the handles of the Idn objects held by ***idnList***), and writes that array (of ***idnCount*** handles) to the location pointed to by idnArray, and also writes the number of Idn objects (held by ***idnList***) to the location pointed to by ***idnCount***.

Return Values

MPIMessageOK	if *IdnListGet* successfully writes the array of Idn hanldes and the number of Idn objects to the ***idnArray*** and ***idnCount*** locations (respectively)

See Also

mpiIdnListSet

Declaration

```
long mpiIdnListSet(MPIIdnList idnList,
                   long       idnCount,
                   MPIIdn     *idnArray)
```

Required Header

stdmpi.h

Description

IdnListSet creates a list (***idnList***) of ***idnCount*** Idn objects, using the Idn handles specified by ***idnArray***.Any existing list is completely replaced.

The ***idnArray*** argument is the address of an array of ***idnCount*** Idn handles, or is NULL (if ***idnCount*** is equal to zero).

You can also create an IdnList incrementally (one Idn at a time), by using the mpiIdnListAppend(...) and/or mpiIdnListInsert(...) methods. To specify the initial Idn object of an IdnList, use the ***idn*** argument of mpiIdnListCreate(...). You can use any mpiIdnList method to examine and manipulate an IdnList, regardless of how the IdnList was created.

Return Values

MPIMessageOK	if *IdnListSet* successfully creates the IdnList using the handles in ***idnArray***

See Also

mpiIdnListInsert | mpiIdnListAppend | mpiIdnListCreate | mpiIdnListGet

mpiIdnListNext

Declaration

```
MPIIdn mpiIdnListNext(MPIIdnList idnList,
                      MPIIdn     idn)
```

Required Header

stdmpi.h

Description

IdnListNext returns the next element following "idn" on the list. This function can be used in conjuntion with mpiIdnListIdnFirst() in order to iterate through the list.

idnList	a handle to the IdnList object.
idn	a handle to an Idn object.

Return Values

handle	to the Idn object following another Idn object (***idn***) in an IdnList (***idnList***)
MPIHandleVOID	if ***idnList*** is invalid if ***idnList*** is the last Idn object in the IdnList (***idnList***)
MPIMessageHANDLE_INVALID	Either ***idnList*** or ***idn*** is an invalid handle.

See Also

mpiIdnListPrevious

mpiIdnListIdnNumberFind

Declaration

```
long mpiIdnListIdnNumberFind(MPIIdnList    idnList,
                             MPIIdnNumber  idnNumber
                             MPIIdn        *idn)
```

Required Header

stdmpi.h

Description

IdnListIdnNumberFind searches the ***idnList*** and writes to ***idn*** the handle to the MPIIdn object whose idn number is ***idnNumber*.**

Return Values

MPIMessageOK	if *IdnListIdnNumberFind* is successful

See Also

mpiIdnListPrevious

Declaration

```
MPIIdn mpiIdnListPrevious(MPIIdnList idnList,
                          MPIIdn     idn)
```

Required Header

stdmpi.h

Description

IdnListPrevious returns the previous element prior to "idn" on the list. This function can be used in conjuntion with mpiIdnListLast() in order to iterate through the list backwards.

idnList	a handle to the IdnList object.
idn	a handle to an Idn object.

Return Values

handle	to the Idn object preceding another Idn object (***idn***) in an IdnList (***idnList***)
MPIHandleVOID	if ***idnList*** is invalid if ***idnList*** is the first Idn object in the IdnList (***idnList***)
MPIMessageHANDLE_INVALID	Either ***idnList*** or ***idn*** is an invalid handle.

See Also

mpiIdnListLast | mpiIdnListNext

mpiIdnListRemove

Declaration

```
long mpiIdnListRemove(MPIIdnList idnList,
                      MPIIdn     idn)
```

Required Header

stdmpi.h

Description

IdnListRemove removes an Idn (***idn***) from an IdnList (***idnList***).

Return Values	
MPIMessageOK	if *IdnListRemove* successfully removes the Idn object from the IdnList

See Also