

setCount:andPostings:

getCount:andPostings:

NXTransport encodeRemotelyFor:freeAfterEncoding:isBycopy:

Manipulating objects by handle addHandle:withWeight:
 insertHandle:withWeight:at:
 replaceHandleAt:with:weight:

Manipulating objects by id addObject:withWeight:
 insertObject:withWeight:at:
 replaceObjectAt:with:weight:

Manipulating objects by index indexForHandle:
 handleOfObjectAt:
 weightOfObjectAt:

Sorting the contents sortByWeightAscending:
 sortBySelector:ascending:

addHandle:(unsigned int)aHandle withWeight:(unsigned int)aWeight

Inserts aHandle with aWeight at the end of the IXPostingList. The object identified in the IXPostingList by aHandle can be retrieved by id with objectAt:. Returns self.

insertHandle:at:withWeight:., addObject:withWeight:., insertObject:withWeight:at:., handleOfObjectAt:., weightOfObjectAt:.

addObject:anObject withWeight:(unsigned int)aWeight

Inserts anObject with aWeight at the end of the IXPostingList, and returns self. anObject is added with no handle addHandle:withWeight: should be used instead of this method whenever possible, to ensure a handle for every record.

insertHandle:at:withWeight:., handleOfObjectAt:., weightOfObjectAt:.

getCount:(unsigned int *)count andPostings:(IXPosting **)thePostings

Returns by reference the number of postings, and a copy of the postings sorted by handle. The sender is responsible for freeing the postings when they are no longer needed. Returns self.

Since objects can be added to an IXPostingList by id instead of by handle, or inserted in any order, the order of postings may not conform to the requirements imposed by the IXPostingExchange protocol (that the postings contain no null handles). In a future release, IXPostingList may sort its postings by handle when returning the postings with this method.

setCount:andPostings:

(unsigned int)handleOfObjectAt:(unsigned int)index

Returns the handle of the posting at index if there is a posting there and it has a valid handle. If index is out of range, returns 0.

in the IXPostingList.

handleOfObjectAt:, indexOf: (List)

initWithSource:(id <IXRecordReading>)aSource

Initializes the receiver, a newly allocated IXPostingList, with aSource providing record activation. object that conforms to the IXRecordReading protocol, for example, an IXRecordManager. The IXPostingList contains no postings. Returns self.

initWithSource:andPostingsIn:, source

initWithSource:(id <IXRecordReading>)aSource
andPostingsIn:(id <IXPostingExchange>)anObject

Initializes the receiver, a newly allocated IXPostingList, with aSource providing record activation, providing an initial set of postings (this will usually be an IXPostingCursor or IXPostingSet). anObject provides the same source as the IXPostingList of this message. This is the designated initializer for the IXPostingList. Returns self.

initWithSource:, source, setCount:andPostings:, IXRecordReading protocol

insertHandle:(unsigned int)aHandle
withWeight:(unsigned int)aWeight
at:(unsigned int)index

Inserts aHandle with aWeight at position index in the IXPostingList, moving existing postings down if necessary. If index is equal to the number of postings in the IXPostingList, aHandle is added at the end. If index is less than 0, the method fails, and this method returns nil, if index is greater than the number of postings in the list or if aHandle is not a valid handle for every record.

If the insertion is successful, returns self if not, returns nil.

insertObject:withWeight:at:, addHandle:withWeight:, addObject:withWeight:, handleOfObjectAt:

insertObject:anObject
withWeight:(unsigned int)aWeight
at:(unsigned int)index

Inserts anObject with aWeight at position index in the IXPostingList, moving existing objects down if necessary. If index is equal to the number of postings in the IXPostingList, anObject is added at the end. The method returns nil, if index is greater than the number of postings in the IXPostingList. anObject must have a valid handle insertHandle:withWeight:at: should be used instead of this method whenever possible. Returns self if successful, nil otherwise.

If the insertion is successful, returns self if not, returns nil.

insertObject:withWeight:at:, addHandle:withWeight:, addObject:withWeight:, handleOfObjectAt:

```
replaceObjectAt:(unsigned int)index
with:anObject
weight:(unsigned int)aWeight
```

Replaces the object and its posting at index with anObject and a posting with a handle of 0 and weight of 1. If the replacement fails, and this method returns nil, if index is greater than or equal to the number of postings in the IXPostingList, or if anObject is nil. anObject is inserted with no handle your code should use replacePostingWithObject:weight: whenever possible, in order to store a valid handle for every posting.

If the replacement is successful, returns self if not, returns nil.

replaceHandleAt:with:weight:

sortBySelector:(SEL)aSelector ascending:(BOOL)flag

Sorts the contents of the IXPostingList by constructing a key from the value each record returns with it. If flag is YES, the sort is ascending (ABCD...), if flag is NO, the sort is descending (ZXYW...).

The sort ordering used is determined by the return type of `aSelector`. The `IXPostingList` determines the Indexing Kit comparator functions to use, and applies the appropriate function to the result of each `aSelector`. However, unlike the keys of an `IXBTree`, the data being compared doesn't have to be inline (serializable). The return of `aSelector` can be a pointer type, and the `IXPostingList` will construct a proper key for it. See the protocol specification for more information on legal comparison values.

sortByWeightAscending:, IXCompareBytes() (C Functions)

sortByWeightAscending:(BOOL)flag

Sorts the contents of the IXPostingList based on the weight of each record. If flag is YES, the sort is from low weight to high, if flag is NO, the sort is from high weight to low.

sortBySelector:ascending:

(id <IXRecordReading>)source

Returns the object which provides storage for the records referenced by the IXPostingList.

initWithSource:, initWithSource:andPostings:, IXRecordReading protocol

```
(unsigned int)weightOfObjectAt:(unsigned int)index
```

Returns the weight of the posting at index, or 0 if index is greater than or equal to the number of postings in the IXPostingList.

handleOfObjectAt:, addHandle:withWeight: