tInitializes and returns a new IXAttributeParser

AttributeReaders:(List *)aListSets the  IXAttributeReaders to those in aList, freeing the previous IXAttributeReaders
AttributeReaders:(List *)aListReturns in aList the IXAttributeReaders

OOL)understandsType:(const char *)aTypeReturns YES if streams or files of aType are parsed
dSourceType:(const char *)aTypeSets aType as a type that will be parsed
moveSourceType:(const char *)aTypeUnsets aType as a type that will be parsed

MinimumWeight:(unsigned int)anIntSets the minimum weight for a token to be included
signed int)minimumWeightReturns the minimum weight for a token to be included
PercentPassed:(unsigned int)anIntSets the percent of tokens dropped
signed int)percentPassedReturns the percent of tokens dropped
WeightingDomain:(IXWeightingDomain *)aDomain
Sets the weighting domain for caluculating peculiarities
IWeightingDomain *)weightingDomainReturns the weighting domain
WeightingType:(IXWeightingType)anIntSets the type of weighting to calculate
IWeightingType)weightingTypeReturns the type of weighting calculated

seFile:(const char *)filenameParses the contents of filename if possible, adding the
ofType:(const char *)aTypeinformation to the attribute-value list
seStream:(NXStream *)streamParses the contents of stream if possible, adding the
ofType:(const char *)aTypeinformation to the attribute-value list
XStream *)analyzeFile:(const char *)filenameParses the contents of filename if possible, adding the
ofType:(const char *)aTypeinformation and returning the analyzed stream
XStream *)analyzeStream:(NXStream *)streamParses the contents of stream if possible, adding the
ofType:(const char *)aTypeinformation and returning the analyzed stream
etClears all compiled information in the attribute-value list

QueryString:(const char *)aString
andAttributeParser:(IXAttributeParser *)aParser
Initializes a new IXAttributeQuery with aString as
the query expression and aParser used to parse it

IPostingList *)evaluateFor:anObjectEvaluates the query string against anObject, returning the records that ma

nsigned int)foldPlural:(char *)aStringReduces aString to its singular form
inLength:(unsigned int)aLength
nsigned int)reduceStem:(char *)aStringReduces aString to its base or root form
inLength:(unsigned int)aLength

CaseFolded:(BOOL)flagSets whether uppercase letters are changed to lowercase
OOL)isCaseFoldedReturns whether uppercase letters are changed to lowercase
PluralsFolded:(BOOL)flagSets whether plural words are reduced to singular
OOL)arePluralsFoldedReturns whether plural words are reduced to singular
StemsReduced:(BOOL)flagSets whether words are reduced to base or root forms
OOL)areStemsReducedReturns whether words are reduced to base or root forms
Punctuation:(const char *)aStringSets the set of characters used to delimit tokens to aString
ar *)punctuationReturns the set of characters used to delimit tokens
StopWords:(const char *)stopWordsSets the words that are deleted from a stream
ar *)stopWordsReturns the words that are deleted from a stream

nsigned int)countReturns the number of key-value pairs in the IXBTree
nsigned int)keyLimitReturns the maximumn allowed length for a key

ptyRemoves the contents of the IXBTree
mpactCompacts the IXBTree's  contents to consume less space

tWithBTree:(IXBTree *)aBTreeInitializes a new IXBTreeCursor to move in aBTree

IBTree *)btreeReturns the IXBTree that the cursor moves in

OOL)setKey:(void *)aKeyPositions the IXBTreeCursor at aKey if possible,
andLength:(unsigned int)aLengthusing aHint to speed search returns YES if aKey
withHint:(unsigned int)aHintis found

OOL)getKey:(void **)aKeyReturns the position of the cursor in aKey and aLength
andLength:(unsigned int *)aLengthalso returning in aHint a hint that can be used to speed
withHint:(unsigned int *)aHintlater search returns NO if the cursor is past the end of the IXBTree's key sp

OOL)writeValue:(void *)aValueWrites or inserts aValue at the cursor's position
andLength:(unsigned int)aLength

teRange:(void *)aRangeWrites aRange into the value at the cursor's position
atOffset:(unsigned int)anOffsetif the cursor isn't exactly on a key, raises an exception
forLength:(unsigned int)aLength

signed int)readValue:(void **)aValueReads the value at the cursor's position and returns its length slides forv
value

signed int)readRange:(void **)aRangeReads a portion of the value at the cursor's position and
ofLength:(unsigned int)aLengthreturns its length if the cursor isn't exactly on a key,
atOffset:(unsigned int)anOffsetraises an exception

moveValueRemoves the value at the cursor's position if the cursor isn't exactly on a key, raises an exception

tInStore:(IXStore *)aStoreInitializes a new IXFileFinder in aStore to index files
atPath:(const char *)pathin path

tFromBlock:(unsigned int)aHandleReloads an IXFileFinder from block aHandle in aStore
inStore:(IXStore *)aStore to index files in path
atPath:(const char *)path

tWithName:(const char *)aNameInitializes a new IXFileFinder named aName in filename
inFile:(const char *)filenameto index files in path
atPath:(const char *)path

tFromName:(const char *)aNameReloads an IXFileFinder stored under aName in filename,
inFile:(const char *)filenameallowing writing back to the file according to flag,
forWriting:(BOOL)flagand set to index files in path
atPath:(const char *)path

IFileFinder *)fileFinderReturns the IXFileFinder that the IXFileRecord belongs to

Filename:(const char *)aNameSets the filename that the IXFileRecord refers to to aName
nst char *)filenameReturns the filename that the IXFileRecord refers to
Filetype:(const char *)aTypeSets the recorded type for the associated file to aType
nst char *)filetypeReturns the recorded type for the associated file
Description:(const char *)aDescriptionSets the description for the associated file to aName
nst char *)descriptionReturns the description for the associated file
Filedate:(unsigned int)aDateSets the recorded creation date for the associated file to aDate
signed int)filedateReturns the recorded creation date for the associated file

nst struct stat *)statBufferReturns the cached UNIX stat buffer for the associated file, or NULL if one hasn't

XAtom)targetLanguageReturns the target language of the reader

tWithSource:(id <IXRecordReading>)aSourceInitializes a new IXPostingList to extract objects from aSource
tWithSource:(id <IXRecordReading>)aSource
andPostingsIn:(id <IXPostingExchange>)anObject
Initializes a new IXPostingList to extract objects
from aSource and initially contain the postings in anObject

<IXRecordReading>)sourceReturns the archive from which objects are extracted

dHandle:(unsigned int)aHandleAdds a new posting to the end of the list
withWeight:(unsigned int)aWeight
ertHandle:(unsigned int)aHandleInserts a new posting at index
withWeight:(unsigned int)aWeight
at:(unsigned int)index
laceHandleAt:(unsigned int)indexReplaces the posting at index with the information supplied
with:(unsigned int)aHandle
weight:(unsigned int)aWeight

dObject:anObjectAdds anObject to the end of the list with aWeight, but no
withWeight:(unsigned int)aWeightposting handle
ertObject:anObjectInserts anObject with aWeight at index, but without a
withWeight:(unsigned int)aWeightposting handle
at:(unsigned int)index
laceObjectAt:(unsigned int)indexReplaces the object at index with the object and weight
with:anObjectsupplied, but without a posting handle
weight:(unsigned int)aWeight

signed int)indexForHandle:(unsigned int)handle
Returns the index of the posting with handle handle
signed int)handleOfObjectAt:(unsigned int)index
Returns the handle for the posting at index
signed int)weightOfObjectAt:(unsigned int)index
Returns the weight for the posting at index

tByWeightAscending:(BOOL)flagSorts the postings and objects by their weight
tBySelector:(SEL)aSelectorSorts the postings and objects based on the return values
ascending:(BOOL)flagof aSelector as sent to each object

Count:(unsigned int)countSets the postings in the IXPostingSet to count postings,
andPostings:(const IXPosting *)postingscopying them if flag is YES
byCopy:(BOOL)flag

signed int)setPosition:(unsigned int)indexSets the selected posting to the one at position index

mUnionWithPostingsIn:(id <IXPostingExchange>)anObject
Performs a set union with the postings in anObject, altering the IXPostingSet's contents but not anObject's
mIntersectionWithPostingsIn:(id <IXPostingExchange>)anObject
Performs a set intersection with the postings in anObject, altering the IXPostingSet's contents but not anO
tractPostingsIn:(id <IXPostingExchange>)anObject
Performs a set subtraction with the postings in anObject, altering the IXPostingSet's contents but not anOb

dAttributeNamed:(const char *)aNameCreates an attribute named aName, based on the values
forSelector:(SEL)aSelectorreturned by objects sent aSelector
OOL)hasAttributeNamed:(const char *)aNameReturns whether an attribute named aName exists
moveAttributeNamed:(const char *)aNameRemoves the attribute named aName

ComparisonFormat:(const char *)aFormatSets the data format used for values in the attribute named
forAttributeNamed:(const char *)aNameaName
nst char *)comparisonFormatForAttributeNamed:(const char *)aName
Returns the data format used for values in the attribute named aName
Comparator:(IXComparator *)aComparatorSets the comparator function and context used for values
andContext:(const void *)aContextin the attribute named aName
forAttributeNamed:(const char *)aName
Comparator:(IXComparator **)aComparatorReturns the comparator function and context used for
andContext:(const void **)aContextvalues in the attribute named aName
forAttributeNamed:(const char *)aName

TargetClass:aClassRestricts the attribute named aName to contain references
forAttributeNamed:(const char *)aNameonly to objects of class aClass (or a subclass)
TargetName:(const char **)aNameReturns the name and class version of the target class for
andVersion:(unsigned int *)targetVersionthe attribute named aName

urns the selector used to build the attribute named aName

ar *)attributeNamesReturns the names of all attributes in the IXRecordManager

ar *)classNamesReturns the names of all classes that have instances stored in the IXRecordManager

ar *)attributeNamesForClass:aClassReturns the names of the attributes in the IXRecordManager that contain
of aClass (or a subclass)

IPostingList *)recordsForClass:aClassReturns in an IXPostingList all instances in the IXRecordManager of a
subclasses)

cardRecord:(unsigned int)aHandleDiscards the records identified by aHandle

laimRecord:(unsigned int)aHandleReclaimes the discarded record identified by aHandle

IPostingList *)discardsReturns in an IXPostingList all discarded records

anPermanently deletes all discarded records

Description:(const char *)aDescriptionSets the description for the attribute named aName to
forAttributeNamed:(const char *)aNameaDescription

Description:(char **)aDescriptionReturns the description for the attribute named aName
forAttributeNamed:(const char *)aName

Parser:(IXAttributeParser *)aParserSets the IXAttributeParser used for the attribute named
forAttributeNamed:(const char *)aNameaName

IXAttributeParser *)parserForAttributeNamed:(const char *)aName
Returns the IXAttributeParser used for the attribute named aName

tInitializes a new IXStore

pyReturns a copy of the IXStore that references the same storage

eDeallocates the IXStore, and all the storage if it's not shared

id *)openBlock:(unsigned int)aHandleOpens a portion of the block identified by aHandle for
atOffset:(unsigned int)anOffsetwriting if possible, otherwise raises an exception
forLength:(unsigned int)aLength

id *)readBlock:(unsigned int)aHandleOpens a portion of the block identified by aHandle for
atOffset:(unsigned int)anOffsetreading if possible, otherwise raises an exception
forLength:(unsigned int)aLength

seBlock:(unsigned int)aHandleCloses the open block identified by aHandle

izeBlock:(unsigned int)aHandleResizes the block identified by aHandle to aSize
toSize:(unsigned int)aSize

signed int)sizeOfBlock:(unsigned int)aHandleReturns the size of the block identified by aHandle

signed int)startTransactionStarts a new transaction and returns the nested level of transactions pending

rtTransactionAborts the current transaction

mmitTransactionCommits the changes made in the current transaction

OOL)areTransactionsEnabledReturns YES if a transaction has ever been started

signed int)nestingLevelReturns the nesting level of transactions pending

signed int)changeCountReturns the number of commitTransaction and abortTransaction messages received b

Contents:(vm_address_t)someContentsReplaces the IXStore's storage with that in someContents
andLength:(vm_size_t)aLength

Contents:(vm_address_t *)theContentsReturns in theContents and aLength the IXStore's storage
andLength:(vm_size_t *)aLengthand its length

mpactCompacts the blocks of storage to take as little space as possible delayed until all transactions are finish

id *)openAtOffset:(unsigned int)anOffsetOpens a portion of the block for writing if possible raises
forLength:(unsigned int)aLengthan exception otherwise

id *)readAtOffset:(unsigned int)anOffsetOpens a portion of the block for reading if possible raises
forLength:(unsigned int)aLengthan exception otherwise

signed int)copyAtOffset:(unsigned int)anOffset
forLength:(unsigned int)aLengthCopies a portion of the block, returning the copy's
identifier

dObjectUnarchives the object previously archived into the block
teObject:(unsigned int)anObjectArchives anObject into the block

lEntryNamed:(const char *)aNameCreates an instance of the store client class named aName,
ofClass:aNameand stores it under aName
lEntryNamed:(const char *)aNameCreates an instance of the store client class named aName,
ofClass:aNameand stores it under aName in the block of the
atBlock:(IXBlockHandle)aHandleIXStoreDirectory's IXStore identified by aHandle
lEntryNamed:(const char *)aNameStores the store client anObject under aName
forObject:anObject

eEntryNamed:(const char *)aNameRemoves from the IXStore the object stored under aName
moveName:(const char *)aNameRemoves the reference to the object stored under aName, but doesn't remove
ptyEmpties all stored objects and names
etRemoves all references to stored objects, but not the objects themselves

OOL)hasEntryNamed:(const char *)aNameReturns whether there's an object stored under aName
Block:(unsigned int *)aHandleGets the block identifier for the object stored under aName
ofEntryNamed:(const char *)aName
Class:(id *)aClassGets the class object for the object stored under aName
ofEntryNamed:(const char *)aName
enEntryNamed:(const char *)aNameActivates and returns the object stored under aName
nst char **)entriesReturns the stored names

tInitializes a new IXStoreFile in a temporary file
tWithFile:(const char *)filenameInitializes a new IXStoreFile with a file named filename
tFromFile:(const char *)filenameInitializes an IXStoreFile from existing storage in filename,
forWriting:(BOOL)flagallowing writing back to the file if flag is YES
eFrees the IXStoreFile, but doesn't affect the storage file

SizeLimit:(vm_size_t)aLimitSets the caching limit for the IXStoreFile to aLimit
m_size_t)sizeLimitReturns the caching limit for the IXStoreFile

tFromDomain:(NXStream *)streamInitializes a new IXWeightingDomain from domain format data in stream
tFromHistogram:(NXStream *)streamInitializes a new IXWeightingDomain from histogram format data in st
tFromWFTable:(NXStream *)streamInitializes a new IXWeightingDomain from NeXTSTEP 2.0 WFTable fo

teDomain:(NXStream *)streamWrite domain format data to stream
teHistogram:(NXStream *)streamWrites histogram format data to stream
teWFTable:(NXStream *)streamWrites NeXTSTEP 2.0 WFTable format data to stream

signed int)totalTokensReturns the total number of tokens in the weighting domain
signed int)uniqueTokensReturns the total number of unque tokens

signed int)countForToken:(void *)aTokenReturns the number of occurrences of aToken in the domain
ofLength:(unsigned int)aLength
signed int)rankForToken:(void *)aTokenReturns the ordinal rank of occurrences of aToken in the
ofLength:(unsigned int)aLengthdomain
oat)frequencyOfToken:(void *)aTokenReturns the frequency of aToken in the domain
ofLength:(unsigned int)aLength
oat)peculiarityOfToken:(void *)aTokenReturns the peculiarity of aToken in the domain
ofLength:(unsigned int)aLength
andFrequency:(float)aFrequency