

initQueryString:andAttributeParser:

Accessing attributes attributeNames

attributeParser

Retrieving the query expression queryString

Evaluating the query evaluateFor:

(char \*)attributeNames

Returns a newline-separated list of the attribute names found in the query string. This can be used to compare the attributes named in the query against those in an IXRecordManager, for example. The sender of this message is responsible for freeing the string returned.

initQueryString:andAttributeParser:, attributeNames (IXRecordManager)

(IXAttributeParser \*)attributeParser

Returns the IXAttributeParser used for parse() operators in the query string.

initQueryString:andAttributeParser:

(IXPostingList \*)evaluateFor:anObject

Evaluates the query string against information in anObject. anObject must be an instance of IXAttributeParser, IXFileRecord, or IXRecordManager. Returns the results of the query in an IXPostingList, or nil if there is no match. The sender of this message is responsible for freeing both the IXPostingList and the objects it contains.

If anObject is an IXAttributeParser or IXFileRecord and it matches the query, it will be the only object in the IXPostingList its handle and weight will be 0, and the IXPostingList's source will be nil. If anObject is an

language, as documented in "The Indexing Kit Query Language." aParser may be nil in this case and an IXAttributeParser will be created by the IXAttributeQuery when a parse() operator is encountered. parse() returns a well formed query, or nil if a compilation error is encountered.

aParser and its IXAttributeReaders should be configured in the same manner as the IXAttributeParser and IXAttributeReaders used to generate attribute values for the context. For example, if an IXRecordManager has a parsed attribute named Text, then any IXAttributeQuery used to query the Text attribute of that IXRecordManager should be initialized with a parsing configuration similar to the Text attribute's parsing configuration. Once the IXAttributeQuery is initialized, its parser shouldn't be sent any messages until after the IXAttributeQuery is freed or reused at that time.

queryString, attributeParser, evaluateFor:, setParser:forAttributeNamed: (IXRecordManager)

(const char \*)queryString

Returns the IXAttributeQuery's query string.

initWithQueryString:andAttributeParser: