

Macintosh<sup>®</sup> <sup>®</sup>**Apple<sup>®</sup> Event Suite Proposal  
for Text Retrieval**

**Public Draft 1.0.6** 7/6/92

Jeremy Bornstein  
ferret designer  
Information Access Research / Information Technologies  
Advanced Technology Group  
© Apple Computer, Inc. 1991-1992

# The Text Retrieval Suite (Proposal)

The Text Retrieval suite contains definitions of Apple event constructs that are used in indexing and retrieval front and back ends. These applications should support most of the constructs defined in the Core suite.

The current proposal concentrates on the back-end functionality, assuming that the main purpose is to provide indexing services to other applications. Using this suite, client applications may use engines to index and search their data, and retrieve the text or locations of relevant documents.

There are many types of search engines, as you probably already know if you are reading this. Two of the (currently) most common types are inverted index engines and signatures engines, but more will probably appear in the future. The capabilities of each type of engine are different, and different engines sometimes require slightly different protocols of interaction. This suite makes it possible for applications to take advantage of searching engines without being bound to one particular engine for the life of the application and without having to know all of the intimate details of the engine being used. However, if applications still want to take advantage of the unique features of a given search engine, they should be able to do so with this interface.

Explanation of the various algorithms referred to here is somewhat beyond the scope of this document, but again, if you're reading it in the first place, you probably understand.

Please do not hesitate to contact the author with comments:

AppleLink: JEREMY  
AOL: Mmytacist  
Internet: jeremy@apple.com  
| Snail-mail: 20525 Mariani Ave. MS 76-4A  
Cupertino, CA 95014

(Note: this document looks best if you have the "Caramond" font installed.)

---

## Overview of the Text Retrieval Suite

Tables 1, 2, and 3 list, respectively, the Apple events, Apple event object classes, and descriptor types defined in (and unique to) the Text Retrieval suite. The Text Retrieval suite makes use of some events, classes, and types which are defined in the Core suite; these definitions are not duplicated here.

**Table 1** Apple events defined in (and unique to) the Text Retrieval suite

Name	Requested action
Session Start	Start a new session with the search engine
Session End	End a session with the search engine
Process Transaction	Prepare to incorporate changes into the index
Evaluate Query	Submit a query to be executed by a search engine

**Table 2** Apple event object classes defined in the Text Retrieval suite

Object class ID	Description
cTRApplication	<p>Applications with text indexing and searching capabilities</p> <p><i>Properties:</i> pIsLocal, pWillIndexRemotely, pIndexType, pCanRelevant, pCanKeepDicts, pCanKeepTermFreqs, pCanKeepDocs, pCanEncrypt, pHasProvisionalHits, pCanAbort, pOwnerKey, pUserKey, pStorageTotal, pStorageFree, pStopwordDoc, pSynonymDoc, pQuerySyntax, pStemmingAlgorithm, pEncryptionAlgorithm, pEngineDesc, pBusyPercentage</p> <p>&lt; also includes properties of the superclass &gt;</p> <p><i>Element Classes:</i> cTextIndex, cTextQuery</p>
cTextIndex	<p>An index of a group of source texts</p> <p><i>Properties</i> pIndexParams, pCanRelevant, pKeepsDocs, pIsEncrypted, pOwnerKey, pUserKey, pMaxNumDocuments, pIndexSize, pMaxIndexSize, pDocsSize, pMaxDocsSize, pTotalSize, pMaxTotalSize, pScriptTag, pStemmingAlgorithm, pEncryptionAlgorithm, pQuerySyntax, pExplanationText, pOccurringWordsDict, pIndexRec, pIndexAux</p>

	<i>Element Classes:</i>	cDocumentWithText
cTextQuery	A request for information about which documents in an index (or set of indices) contain certain terms <i>Properties:</i>	pRefNum,pQueryText,pQuerySyntax, pMaxHitsDesired,pMinRankDesired, pSearchComplete,pRestrictionQuery,pProgressText pPercentDone,pTextQueryRec
	<i>Element Classes:</i>	cHitList
cHitList	A list of documents which satisfy a query, in relevance order if applicable <i>Properties:</i>	pQueryRefNum,pIndex,pVerified,pRelevant, pExecutionDateTime,pTotalHits,pSearchComplete, pProgressBuffer
	<i>Element Classes:</i>	cHit
cHit	An individual occurrence of a term (or set of terms) in a document <i>Properties:</i>	pDocID,pFieldID,pOffset,pExtent,pRelevance
	<i>Element Classes:</i>	< none >
cIndexType	Signatures for the algorithms used by the application to produce indices <i>Properties:</i>	pMajorType,pMinorType
	<i>Element Classes:</i>	< none >
cOccurringWordsDict	A list of words and perhaps occurrence counts <i>Properties:</i>	pCountTypes,pWordCountsGuaranteed
	<i>Element Classes:</i>	cOccurringWord
cOccurringWord	A single word that occurs in an index <i>Properties:</i>	pOccurrenceCounts
	<i>Element Classes:</i>	cWord
cDocumentWithText	A document that has some text in it. (The document does not have to be made up entirely of text.) <i>Properties:</i>	cDocument
	<i>Element Classes:</i>	< undefined >

n **Table 3** Descriptor types defined in the Text Retrieval suite

---

Descriptor type	Description
typeHitList	A record containing all of the properties in a cHitList
typeRelevantHit	A record containing all of the information in a cHit
typeHit	A record containing all of the information in a cHit except for the pRelevance

---

## Apple events defined in the Text Retrieval suite

The Apple events defined in the Text Retrieval suite are described in the following sections.

Note that all events used by the Text Retrieval suite, except for the "Start Session" and "End Session" events, should contain the key "keyAESessionID". This is so that the engine can separate clients who just happen to be living in the same application program. Although this document does not contain documentation for the events from other suites which are used by the Text Retrieval suite, when using these events, a "keyAESessionID" must be attached with the proper value for the session.

---

## Evaluate Query—find documents which satisfy a given query

**Requested Action** Examine the index or indices to determine which documents are relevant to the given query.

**Event Class** kAETextRetrievalSuite

**Event ID** kAETextQuery

### Parameters

keyAEDirectObject

Description: The index or indices to query  
Descriptor Type: typeObjectSpecifier  
Required or Optional? Required

keyAETextQuery

Description: The query to use  
Descriptor Type: typeObjectSpecifier (or typeTextQuery)  
Required or Optional? Required

keyAESessionID

Description: The associated session ID for this event.  
Descriptor Type: typeSessionID  
Required or Optional? Required

keyAESendHitList

Description: If true, the hit list will be sent in the reply  
Descriptor Type: typeBoolean  
Required or Optional? Optional

keyAEMaxHitsToSend

Description: The maximum number of hits to be sent in the reply  
Descriptor Type: typeLongInteger  
Required or Optional? Optional

keyAETrickleHits

Description: If true, hit progress events will be sent as described in the section, "About Progressive Disclosure."  
Descriptor Type: typeBoolean  
Required or Optional? Optional

keyAETrickleProgress

Description: If true, general progress events will be sent as described in the section, "About Progressive Disclosure."  
Descriptor Type: typeBoolean  
Required or Optional? Optional

keyAEJustSendDone	Description:	If true, specifies that a general progress event (as described in the section, "About Progressive Disclosure.") will be sent when the query has completed.
	Descriptor Type:	typeBoolean
	Required or Optional?	Optional
keyAEUpdateProgressBuffer	Description:	If true, the engine will keep the hit progress buffer of the hit list up to date, as described in the section, "About Progressive Disclosure."
	Descriptor Type:	typeBoolean
	Required or Optional?	Optional
keyAEVerifyHits	Description:	If true, the hits will be verified before the event completes.
	Descriptor Type:	typeBoolean
	Required or Optional?	Optional
keyAERelevanceHits	Description:	If true, the hits will be returned with relevance information, sorted by relevance scores.
	Descriptor Type:	typeBoolean
	Required or Optional?	Optional
<b>Reply Parameters</b>		
keyDirectObject	Description:	The refNum assigned to the query, which may be used in retrieving the hit list or progress information.
	Descriptor Type:	typeLongInteger
	Required or Optional?	Optional
keyAEHitList	Description:	The hit list generated by the query. This property is present in the reply only if keyAESendHitList was true in the original event. This is not recommended; see Notes.
	Descriptor Type:	typeHitList
	Required or Optional?	Optional

## keyAEHowBusy

Description:	A number which tells the client how busy the engine is, after beginning this query.
Descriptor Type:	typeShortInteger
Required or Optional?	Optional

## Result Codes

errAEUnknownSession	-6660	The client has not registered itself with the engine.
errAEFail	-1732	Failed to handle this Apple event

## Notes

Anywhere an index object specifier is requested, an index refNum or indexDesc may be passed.

Anywhere a query object specifier is requested, a query refNum or textQueryDesc may be passed. Note that if the query passed to this event already contains a pRefNum, then that value will be ignored and will NOT be used by the search engine. The search engine is in charge of assigning all refNums.

For "one shot" queries, which open the indicated indices, perform the query, and close the indices, unopened indices may be passed as the direct parameter. This is not recommended and may be removed from the suite.

If keyAERelevanceHits is TRUE, hits must be returned in relevance order.

See "About Progressive Disclosure" for explanations of the keyAETrickleHits and keyAETrickleProgress parameters.

Clients should never use kAEWaitReply with this event, since the reply may originate from an address different from the original server.

It is recommended that clients do not specify keyAESendHitList, since then they will not be able to correlate progress information about the query with the actual query. Progress information uses the query's refNum, which is returned in the reply. If they don't get the reply until the query is complete, then they won't have the refNum when the progress information is received.



---

## Process Transaction—prepare to commit a transaction

**Requested Action** The engine should perform all the operations necessary to incorporate the new material into its index without actually changing the index. (This is the first phase of a two-phase commit.) This event should be sent before End Transaction when an index has been changed.

**Event Class** kAETextRetrievalSuite

**Event ID** kAEProcessTransaction

### Parameters

keyDirectObject

Description:	The transaction ID to process.
Descriptor Type:	typeLongInteger
Required or Optional?	Required

keyAESessionID

Description:	The associated session ID for this event.
Descriptor Type:	typeSessionID
Required or Optional?	Required

### Reply Parameters

keyDirectObject

Description:	Transaction status record
Descriptor Type:	typeTransactionStatus
Required or Optional?	Optional

### Result Codes

errAEUnknownSession	-6660	The client has not registered itself with the engine.
---------------------	-------	---

---

## Start Session—start a session with the search engine

**Requested Action** The engine should perform all the operations necessary to begin a new session. Clients may have multiple sessions with the same search engine.

**Event Class** kAETextRetrievalSuite

**Event ID** kAESTartSession

**Parameters** **None**

### Reply Parameters

keyDirectObject

Description:	Session ID
Descriptor Type:	typeSessionID
Required or Optional?	Required

### Result Codes

errAETooManySessions	-6662	The engine cannot accomodate another session.
----------------------	-------	---

---

## End Session—terminate a session with the search engine

**Requested Action** The engine should perform all the operations necessary to end a currently open session with a client. This probably includes such things as closing all indices open in the session, erasing all queries, &c.

**Event Class** kAETextRetrievalSuite

**Event ID** kAEEndSession

### Parameters

keyDirectObject

Description:	The session ID to terminate.
Descriptor Type:	typeSessionID
Required or Optional?	Required

**Reply Parameters** None

### Result Codes

errAESMGeneralError	-6660	Unspecified error.
---------------------	-------	--------------------

---

## Object classes defined in the Text Retrieval suite

The Apple event object classes defined in the Core suite are described in the following sections, currently listed in approximate descending order of conceptual complexity.

---

## cTRApplication— an application with text indexing and retrieval functionality

**Description** The cTRApplication class is the class of applications which support the model of text indexing and retrieval defined in this document.

**Superclass** cApplication (Core suite)

**Default Desc Type** typeTextEngineRec

### Property

pIsLocal

Description: If this property is TRUE, the engine resides on a machine local to the client.

Object Class ID: cBoolean

Inherited? No

Read-Only or Read/Write? Read-only

pBusyPercentage

Description: A number from 0 to 200 indicating the load of the engine. A value of 100 indicates the maximum normal load; values over 100 indicate that efficiency is suboptimal and response may be slow.

Object Class ID: cBoolean

Inherited? No

Read-Only or Read/Write? Read-only

pWillIndexRemotely

Description: If this property is TRUE, the engine will index documents not accessible to its own file system. (That is, it is possible to send documents to the application over the wire.)

Object Class ID: cBoolean

Inherited? No

Read-Only or Read/Write? Read-only

pIndexType	Description:	This property indicates what kind of index the engine produces. Major and minor types are included. Valid major types at this point include: kSignaturesIndex, kInvertedIndex, and kObscureTypeOfIndex.
	Object Class ID:	cIndexType
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pCanRelevant	Description:	If this property is TRUE, the engine is capable of providing hit lists sorted by relevance to a query.
	Object Class ID:	cBoolean
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pCanKeepDicts	Description:	If this property is TRUE, the engine is capable of keeping dictionaries of terms which occur in a given index.
	Object Class ID:	cBoolean
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pDictInfoTypes	Description:	This property contains a list of the types of information which the dictionary will maintain about each word. Each type is indicated by a cEnumeration.
	Object Class ID:	cAEList
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pCanKeepDocs	Description:	If this property is TRUE, the engine is capable of storing the text of documents which it has indexed.
	Object Class ID:	cBoolean
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pCanEncrypt	Description:	If this property is TRUE, the engine is capable of encrypting materials which it stores (indices and documents).
	Object Class ID:	cBoolean
	Inherited?	No
	Read-Only or Read/Write?	Read-only

pHasProvisionalHits	<p>Description: If this property is TRUE, the engine needs a verification pass for hits to be guaranteed.</p> <p>Object Class ID: cBoolean</p> <p>Inherited? No</p> <p>Read-Only or Read/Write? Read-only</p>
pCanAbort	<p>Description: If this property is TRUE, the engine is capable of aborting transactions in progress.</p> <p>Object Class ID: cBoolean</p> <p>Inherited? No</p> <p>Read-Only or Read/Write? Read-only</p>
pOwnerKey	<p>Description: This property is a signature which identifies the owner of the engine. If the engine is remote, clients must <i>write</i> to this property to establish that they have the authorization to create and/or modify indices. Remote clients will always read this property as empty.</p> <p>Object Class ID: cText</p> <p>Inherited? No</p> <p>Read-Only or Read/Write? Read/Write or <i>Write-only</i></p>
pUserKey	<p>Description: This property is a signature which identifies the legitimate users of the engine. If the engine is remote, clients must <i>write</i> to this property to establish that they have the authorization to use indices. Remote clients will always read this property as empty.</p> <p>Object Class ID: cText</p> <p>Inherited? No</p> <p>Read-Only or Read/Write? Read/Write or <i>Write-only</i></p>
pStorageTotal	<p>Description: Total of the storage available to the engine, both occupied and free, for the storage of indices and associated data. This does not include the engine's working reserve, if any.</p> <p>Object Class ID: cLongInteger</p> <p>Inherited? No</p> <p>Read-Only or Read/Write? Read-only</p>
pStorageFree	<p>Description: Total of the free storage available to the engine, for the storage of indices and associated data. This does not</p>

	Object Class ID:	include the engine's working reserve, if any. cLongInteger
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pStopwordDoc	Description:	This property indicates a document which is a return-delimited list of stopwords to be used as the default stopword list by the engine. <sup>1</sup>
	Object Class ID:	cFile (or cDocument?)
	Inherited?	No
	Read-Only or Read/Write?	Read-only or Read-write (depending on access rights of user)
pSynonymDoc <sup>2</sup>	Description:	This property indicates a document which is a return-delimited list of synonyms to be used as the default synonym list by the engine.
	Object Class ID:	cFile (or cDocument?)
	Inherited?	No
	Read-Only or Read/Write?	Read-only or Read-write (depending on access rights of user)
pQuerySyntax	Description:	This property indicates which query syntaxes are supported by the engine.
	Object Class ID:	cQuerySyntaxList (really a cAEList that is a list of cEnumerateds)
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pStemmingAlgorithm	Description:	This property indicates which stemming algorithms are supported by the engine.
	Object Class ID:	cStemmingAlgorithmList (really a cAEList that a list of cEnumerateds)
	Inherited?	No
	Read-Only or Read/Write?	Read-only

---

<sup>1</sup>The first line of the document is a configuration line which indicates the format of the file. ( the word "stemmed," then the stopword list is pre-stemmed.)

<sup>2</sup>This may be expanded into a full-blown thesaurus mechanism.



## pEncryptionAlgorithm

Description:	This property indicates which encryption algorithms are supported by the engine.
Object Class ID:	cEncryptionAlgorithmList (a list of cEnumerateds)
Inherited?	No
Read-Only or Read/Write?	Read-only

(All inherited fields have definitions unchanged from the superclass.)

## Element Classes

### cTextIndex

Description:	Every currently open index is represented by an object of class cTextIndex.
Inherited?	No
Read-Only or Read/Write?	Read-only or Read/Write
Key Forms:	formAbsolutePosition,formRange,formTest

### cTextQuery

Description:	Every query sent to the engine is represented by an object of class cTextQuery.
Inherited?	No
Read-Only or Read/Write?	Read/Write
Key Forms:	formAbsolutePosition,formRange,formTest

## Apple Events

### *Events defined in the Core suite:*

Clone	Inherited from cAbstractObject
Close	Inherited from cOpenableThing
Copy	Inherited from cAbstractObject
Cut	Inherited from cAbstractObject
Delete Element	Inherited from cAbstractObject
Does Object Exist	Inherited from cAbstractObject
Get Data	Inherited from cAbstractObject
Get Structure	Inherited from cAbstractObject
Move	Inherited from cAbstractObject
New Element	Inherited from cAbstractObject
Number of Elements	Inherited from cAbstractObject
Open	Inherited from cOpenableThing
Paste	Inherited from cAbstractObject
Set Data	Inherited from cAbstractObject

---

## cTextIndex— an index of documents containing text

<b>Description</b>	The cTextIndex class is the class which represents indices created and managed by a cTRApplication.	
	Note: Although all of the parameters of a cTextIndex are Read/Write, changing some of them after the index has been created may require a complete rebuild of the index!	
<b>Superclass</b>	cDocument (Core suite)	
<b>Default Desc Type</b>	typeIntlText	
<b>Property</b>		
pIndexParams	Description:	Contains the parameters unique to an engine type.
	Object Class ID:	cIndexParams
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pCanRelevant	Description:	If this property is TRUE, the index can return hit lists with relevance information.
	Object Class ID:	cBoolean
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pKeepsDocs	Description:	If this property is TRUE, the index keeps copies of the documents it indexes.
	Object Class ID:	cBoolean
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pIsEncrypted	Description:	If this property is TRUE, the index is encrypted.
	Object Class ID:	cBoolean
	Inherited?	No
	Read-Only or Read/Write?	Read/Write

pOwnerKey	Description:	This property is a signature which identifies the owner of the index. If the index is on a remote machine, clients must <i>write</i> to this property to establish that they have the authorization to modify this index. Remote clients will always read this property as empty.
	Object Class ID:	cText
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pUserKey	Description:	This property is a signature which identifies the authorized users of the index. If the index is on a remote machine, clients must <i>write</i> to this property to establish that they have the authorization to use this index. Remote clients will always read this property as empty.
	Object Class ID:	cText
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pMaxNumDocuments	Description:	The maximum number of documents which the index is allowed to have indexed.
	Object Class ID:	cLongInteger
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pIndexSize	Description:	The size (in bytes) of the actual index, not counting any stored documents.
	Object Class ID:	cLongInteger
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pMaxIndexSize	Description:	The maximum size (in bytes) that the index is allowed to occupy, not counting any stored documents.
	Object Class ID:	cLongInteger
	Inherited?	No
	Read-Only or Read/Write?	Read/Write

pDocsSize	Description:	The size (in bytes) of the documents stored with the index.
	Object Class ID:	cLongInteger
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pMaxDocsSize	Description:	The maximum size (in bytes) that the documents stored with the index are allowed to occupy.
	Object Class ID:	cLongInteger
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pTotalSize	Description:	The size (in bytes) of all of the storage associated with the index.
	Object Class ID:	cLongInteger
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pMaxTotalSize	Description:	The maximum size (in bytes) that everything associated with the index is allowed to occupy.
	Object Class ID:	cLongInteger
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pScriptTag	Description:	The script system and language of the documents in the index. (Note that most indices work best if the language and script system are consistent throughout the index.)
	Object Class ID:	cIntIWritingCode
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pStemmingAlgorithm	Description:	The signature of the stemming algorithm used by this index.
	Object Class ID:	cEnumeration
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pEncryptionAlgorithm	Description:	The signature of the encryption algorithm used by this

	Object Class ID:	index.
	Inherited?	cEnumeration
	Read-Only or Read/Write?	No
pQuerySyntax		Read/Write
	Description:	The signature of the query syntax used by this index.
	Object Class ID:	cEnumeration
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pExplanationText		
	Description:	A user-level explanation of what's in the index. Returned as the default descriptor?
	Object Class ID:	cText
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pOccurringWordsDict		
	Description:	Every word which appears in any of the documents in the index has an entry in the occurring words dictionary.
	Object Class ID:	cOccurringWordsDict
	Inherited?	No
	Read-Only or Read/Write?	Read-only

(All inherited fields have definitions unchanged from the superclass.)

## Element Classes

cDocumentWithText		
	Description:	A reference used to obtain the text of documents. (May actually contain the document's text; see below.)
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
	Key Forms:	formAbsolutePosition,formRange,formTest

## Apple Events

*Events defined in the Text Retrieval suite:*

Execute Query                      Not inherited

*Events defined in the Core suite:*

Clone                                Inherited from cAbstractObject

Close                                 Inherited from cOpenableThing

Copy                                 Inherited from cAbstractObject

Cut                                  Inherited from cAbstractObject

Delete Element                    Inherited from cAbstractObject

Does Object Exist                 Inherited from cAbstractObject

Get Data                             Inherited from cAbstractObject

Get Structure                      Inherited from cAbstractObject

Move                                 Inherited from cAbstractObject

New Element                        Inherited from cAbstractObject

Number of Elements                Inherited from cAbstractObject

Open                                 Inherited from cOpenableThing

Paste                                Inherited from cAbstractObject

Set Data                             Inherited from cAbstractObject

---

## cTextQuery— a request for documents with certain contents

<b>Description</b>	The cTextQuery class is the class which represents queries to be sent to objects of type cTextIndex.	
<b>Superclass</b>	cAbstractObject (Core suite)	
<b>Default Desc Type</b>	typeIntIText	
<b>Property</b>		
pRefNum	Description:	Sequence number (unique to a session) used to refer to queries. This element must be used when referring to queries already executed by the engine.
	Object Class ID:	cShortInteger
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pQueryText	Description:	The text of the query, i.e. the terms to search for. This is represented in the query language specified by the pQuerySyntax property.
	Object Class ID:	cText
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pQuerySyntax	Description:	This property indicates the signature of the query syntax used for the pQueryText property..
	Object Class ID:	cEnumeration
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pMaxHitsDesired	Description:	This property indicates the maximum number of hits which the query should return.
	Object Class ID:	cLongInteger
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pMinRelevanceDesired	Description:	This property indicates the minimum relevance score of the hits to be returned by the query.

---

	Object Class ID:	cLongInteger
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pSearchComplete	Description:	This property will be TRUE only if the engine has completed the search over all indices.
	Object Class ID:	cBoolean
	Inherited?	No
	Read-Only or Read/Write?	Read-Only
pRestrictionQuery	Description:	If this property is non-nil, it is another cTextQuery, to be used with the pRestrictionRelation property to limit or extend the hit list generated by the query.
	Object Class ID:	cTextQuery
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pRestrictionRelation	Description:	Specifies a logical operation for the combination of the results of the query with the results from the pRestrictionQuery.
	Object Class ID:	cEnumeration (valid values: kNoRestriction, kAnd, kOr, kXor, kAAndNotB, kBAndNotA. "A" is the query and "B" is the restriction query.)
	Inherited?	No
	Read-Only or Read/Write?	Read/Write
pProgressText	Description:	If pTrickleProgress is TRUE, this property contains a text string indicating which stage of processing is currently occurring in the engine.
	Object Class ID:	cText
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pPercentDone	Description:	If pTrickleProgress is TRUE, this property contains a number from 0 to 100 indicating the engine's guess at how much of the processing is completed.
	Object Class ID:	cText
	Inherited?	No
	Read-Only or Read/Write?	Read-only



## Element Classes

cHitList

Description:	Every searched index generates a separate cHitList containing the relevant documents from that index.
Inherited?	No
Read-Only or Read/Write?	Read-Only
Key Forms:	formAbsolutePosition,formRange,formTest

## Apple Events

*Events defined in the Core suite:*

Clone	Inherited from cAbstractObject
Close	Inherited from cOpenableThing
Copy	Inherited from cAbstractObject
Cut	Inherited from cAbstractObject
Delete Element	Inherited from cAbstractObject
Does Object Exist	Inherited from cAbstractObject
Get Data	Inherited from cAbstractObject
Get Structure	Inherited from cAbstractObject
Move	Inherited from cAbstractObject
New Element	Inherited from cAbstractObject
Number of Elements	Inherited from cAbstractObject
Open	Inherited from cOpenableThing
Paste	Inherited from cAbstractObject
Set Data	Inherited from cAbstractObject

---

## cHitList— a list of documents (the results of a cTextQuery)

<b>Description</b>	The cHitList class is the class which represents the results of queries.	
	Superclass	cAEList (Core suite)
<b>Default Desc Type</b>	typeIntIText	
<b>Property</b>		
pQueryRefNum	Description:	Sequence number (unique to a session) of the query which generated this hit list.
	Object Class ID:	cShortInteger
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pIndex	Description:	Reference to the index which generated the hits.
	Object Class ID:	cObjectSpecifier or cLongInteger (the index's refNum)
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pVerified	Description:	This property is TRUE if the hits have been verified.
	Object Class ID:	cBoolean
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pRelevant	Description:	This property is TRUE if the hits have relevance information.
	Object Class ID:	cBoolean
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pExecutionDateTime	Description:	The date that the hits were generated.
	Object Class ID:	cDate
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pTotalHits	Description:	The number of hits in the list.
	Object Class ID:	cLongInteger
	Inherited?	No

pSearchComplete	Read-Only or Read/Write?	Read-only
	Description:	This property is TRUE if the hit list is complete and will not be modified further by the engine.
	Object Class ID:	cLongInteger
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pProgressBuffer	Description:	If the pTrickleHits property of the query is true, this buffer contains hits waiting to be retrieved by the client. The buffer is erased each time it is read. See "About Progressive Disclosure," below, for details.
	Object Class ID:	cAEList
	Inherited?	No
	Read-Only or Read/Write?	Read-only

## Element Classes

cHit	Description:	An actual hit.
	Inherited?	No
	Read-Only or Read/Write?	Read-only
	Key Forms:	formAbsolutePosition,formRange,formTest
	NOTE:	Usually, clients will request a range of hits: "hits 1 to 100 of the hitlist of query 4."

## Apple Events

*Events defined in the Core suite:*

Copy	Inherited from cAbstractObject
Does Object Exist	Inherited from cAbstractObject
Get Data	Inherited from cAbstractObject
Get Structure	Inherited from cAbstractObject
Number of Elements	Inherited from cAbstractObject

---

## cHit— an individual hit

**Description** The cHit class is the class which represents individual hits.

**Superclass** cAbstractObject

**Default Desc Type** typeIntlText

### Property

pDocID

Description:	The document ID of the hit.
Object Class ID:	cLongInteger
Inherited?	No
Read-Only or Read/Write?	Read-only

pFieldID

Description:	The field ID of the hit; documents without fields are all field ID 0.
Object Class ID:	cLongInteger
Inherited?	No
Read-Only or Read/Write?	Read-only

pOffset:

Description:	The offset of the hit from the start of the field. Should be 0 if the engine does not support hit offsets.
Object Class ID:	cLongInteger
Inherited?	No
Read-Only or Read/Write?	Read-only

pExtent:

Description:	The extent of the hit from the start of the offset. Should be 0 if the engine does not support hit extents.
Object Class ID:	cLongInteger
Inherited?	No
Read-Only or Read/Write?	Read-only

pRelevance:

Description:	The engine's estimate of the relevance of this hit to the query. 0xFFFFFFFF is the highest possible value. Should be 0 if the engine does not support relevance.
Object Class ID:	cLongInteger
Inherited?	No
Read-Only or Read/Write?	Read-only

**Element Classes** None

## Apple Events

*Events defined in the Core suite:*

Copy	Inherited from cAbstractObject
Does Object Exist	Inherited from cAbstractObject
Get Data	Inherited from cAbstractObject
Get Structure	Inherited from cAbstractObject

---

## cIndexType— the algorithms used by an engine to make indices

<b>Description</b>	The cIndexType class is the class which represents the kind of index constructed by a given engine.	
<b>Superclass</b>	cAbstractObject	
<b>Default Desc Type</b>	typeIntlText	
<b>Properties</b>		
pMajorType	Description:	This is the major type of the index, which indicates what general class of algorithm is used by the engine: i.e. signatures, inverted, &c. Currently defined: kSignaturesIndex, kInvertedIndex, and kObscureTypeOfIndex.
	Object Class ID:	cEnumeration
	Inherited?	No
	Read-Only or Read/Write?	Read-only
pMinorType	Description:	This is the minor type of the index, indicating which variation of the major type is used by the engine. For example, two valid types for a signatures engine are kSuperimposedCoding and kWordSignatures.
	Object Class ID:	cEnumeration
	Inherited?	No
	Read-Only or Read/Write?	Read-only
<b>Element Classes</b>	None	
<b>Apple Events</b>	<i>Events defined in the Core suite:</i>	
	Copy	Inherited from cAbstractObject
	Does Object Exist	Inherited from cAbstractObject
	Get Data	Inherited from cAbstractObject
	Get Structure	Inherited from cAbstractObject
	Number of Elements	Inherited from cAbstractObject
	Open	Inherited from cOpenableThing

---

## cOccurringWordsDict— a dictionary of words in a corpus

**Description** The cOccurringWordsDict class is the class which represents the collection of words which appear in a corpus.

**Superclass** cAEList

**Default Desc Type** typeIntlText

### Properties

pCountTypes

Description: A list of the types of information which the dictionary keeps about each word.  
Object Class ID: cAEList (of cEnumerateds. valid values include kGlobalFrequency and kDocumentFrequency)  
Inherited? No  
Read-Only or Read/Write? Read-only

pWordCountsGuaranteed

Description: This property is TRUE if the word counts are guaranteed to be completely accurate. (This may not be true for some types of index.)  
Object Class ID: cLongInteger  
Inherited? No  
Read-Only or Read/Write? Read-only

### Element Classes

cOccurringWord

Description: A word which appears in the corpus.  
Inherited? No  
Read-Only or Read/Write? Read-only  
Key Forms: formAbsolutePosition,formRange,formTest

### Apple Events

*Events defined in the Core suite:*

Close Inherited from cOpenableThing  
Copy Inherited from cAbstractObject  
Does Object Exist Inherited from cAbstractObject  
Get Data Inherited from cAbstractObject  
Get Structure Inherited from cAbstractObject  
Number of Elements Inherited from cAbstractObject  
Open Inherited from cOpenableThing

---

## cOccurringWord— a word in a corpus

**Description** The cOccurringWord class is the class which represents an individual word which appears in a corpus.

**Superclass** cAbstractObject

**Default Desc Type** typeIntlText

### Properties

pText

Description:	The text of the word.
Object Class ID:	cText
Inherited?	No
Read-Only or Read/Write?	Read-only

pOccurrenceCounts

Description:	The list of occurrence counts for the word. Counts appear in the same order as the dictionary's pCountTypes property.
Object Class ID:	cAEList (of cLongIntegers)
Inherited?	No
Read-Only or Read/Write?	Read-only

**Element Classes** None

### Apple Events

*Events defined in the Core suite:*

Copy	Inherited from cAbstractObject
Does Object Exist	Inherited from cAbstractObject
Get Data	Inherited from cAbstractObject
Get Structure	Inherited from cAbstractObject
Number of Elements	Inherited from cAbstractObject



---

## cDocumentWithText— any document which contains text

**Description** The cDocumentWithText class is the class which references documents containing text or in some cases actually contains the text of these documents.

**Superclass** cAbstractObject

**Default Desc Type** typeIntIText

### Properties

pDocumentType

Description: The type of information in this object: if kAliasType, the pAlias property should be used. If anything else, is the file type of the document whose text is stored in the pText property. If straight ASCII, this must be 'TEXT' (kASCIIType).

Object Class ID: cEnumeration

Inherited? No

Read-Only or Read/Write? Read/Write

pAlias

Description: An alias to the document, which may be passed as an FSSpec if the engine is local. If passed as an alias, the engine should be able to find the document when a user moves it.

Object Class ID: cAlias

Inherited? No

Read-Only or Read/Write? Read/Write

pText

Description: The text of the document, in raw ASCII.

Object Class ID: cText

Inherited? No

Read-Only or Read/Write? Read/Write

**Element Classes** None

## Apple Events

### *Events defined in the Core suite:*

Clone	Inherited from cAbstractObject
Close	Inherited from cOpenableThing
Copy	Inherited from cAbstractObject
Cut	Inherited from cAbstractObject
Delete Element	Inherited from cAbstractObject
Does Object Exist	Inherited from cAbstractObject
Get Data	Inherited from cAbstractObject
Get Structure	Inherited from cAbstractObject
Move	Inherited from cAbstractObject
New Element	Inherited from cAbstractObject
Number of Elements	Inherited from cAbstractObject
Open	Inherited from cOpenableThing
Paste	Inherited from cAbstractObject
Set Data	Inherited from cAbstractObject

---

## cIndexParams— parameters for adjusting algorithms

**Description** The cIndexParams class is the class which represents the parameters an engine is to use for its basic algorithms. The constant which names properties specific to a minor algorithm type should be identical to the algorithm type. Other constants and accompanying record types should be established as they are needed.

**Superclass** cAbstractObject

**Default Desc Type** typeIntlText

### Properties

pSuperimposedCodingParams

Description:	A record specific to superimposed coding algorithms, containing the two fields "maxBitsToSetPerBlock" and "blockSizeInBytes."
Object Class ID:	cData
Inherited?	No
Read-Only or Read/Write?	Read/Write

pImpenetrableIndexParams

Description:	A record specific to the engine in use.
Object Class ID:	cData
Inherited?	No
Read-Only or Read/Write?	Read/Write?

*Other properties may be added as needed!*

**Element Classes** None

### Apple Events

*Events defined in the Core suite:*

Copy	Inherited from cAbstractObject
Cut	Inherited from cAbstractObject
Does Object Exist	Inherited from cAbstractObject
Get Data	Inherited from cAbstractObject
Get Structure	Inherited from cAbstractObject
Number of Elements	Inherited from cAbstractObject
Paste	Inherited from cAbstractObject
Set Data	Inherited from cAbstractObject

---

## Primitive object classes defined in the Text Retrieval suite

Table 3-4 lists the primitive Apple event object classes (classes with no properties and only one element) defined in the Text Retrieval suite.

**Table 3-4** Primitive object classes defined in the Core suite

Object class ID	Descriptor type of element	Description
cTransactionStatus	typeTransactionStatus	Information about transaction in progress

---

## Descriptor types defined in the Text Retrieval suite

The descriptor types defined in the Text Retrieval suite are described in the following section.

---

### typeHitList—everything in a cHitList

<b>Description</b>	A compact, unified representation of a cHitList.
<b>Data Size</b>	sizeof(typeHitList)
typeTextQuery	A record containing all of the properties in a cTextQuery

---

### typeRelevantHit—everything in a cHit

<b>Description</b>	A record which contains all the information needed to represent a relevance-ranked hit.
<b>Data Size</b>	sizeof(typeRelevantHit)

---

## **typeHit—everything in a cHit except the relevance score**

**Description**            A record which contains all the information in a non-relevance-ranked hit.

**Data Size**             sizeof(typeHit)

---

## **typeTransactionStatus—status of transaction in progress**

**Description**            Information about a transaction in progress. (We're not sure what's in here yet.)

**Data Size**             sizeof(typeTransactionStatus)

---

## **typeSessionID—identifier for a session**

**Description**            An identifier to associate events with a session.

**Data Size**             sizeof(typeLongInteger)

---

## About Progressive Disclosure

Some clients will want to keep track of the hits generated by queries in real-time, some clients will want to know how far the engine has progressed with queries, and some clients don't care about queries until they have completed. The current proposal can accommodate all of these styles of interaction, including a combination of the first two.

There are many things that can happen to a hit after it has first been detected: some engines may not have a final judgement of what is a hit or not until the query is virtually complete. This makes it slightly complicated for the client to keep the user informed about the current status of the query.

The progressive disclosure mechanism allows clients to keep track of an engine's progress on a query. There are two ways to use this mechanism, one for cases in which the engine is local to the client, and one where it is not local. The distinguishing factor is whether or not the engine can send events to the client. If the engine can, then it sends events representing the various things that can happen to a hit. If it can't, then the client can read properties of the query and of the engine in order to determine the same information.

### Progressive Disclosure Events

Progress information is available on two levels: information about the hit list in progress, and information about progress in general.

#### General Progress Information Event

This event provides two kinds of information: a text string which is the engine's statement of what it's doing for you in general ("Processing Query," "Sorting terms," &c) and a number which indicates its estimate of how much of the work it has already completed. Note that this event is received, not sent by the client and that the client should send no reply for this event.

Event: Progress Report

**Requested Action:** Record the engine's current progress, and perhaps notify the user.

**Event Class** kAETextRetrievalSuite

**Event ID** kAEPDProgressReport

Parameters

keyAECurrentStatus

Description:	Status string
Descriptor Type:	typeIntlText
Required or Optional	optional

keyAEPercntDone

Description:	Estimated percentage done
Descriptor Type:	typeShortInteger
Required or Optional	optional

## Hit Information Progress Events

There are three events which are sent to inform the client about progress with regard to hits: New Hit, Obsolete Hit, and Changed Hit. These are described briefly here. Note that these are events which the **client** is expected to respond to. The only programs which send these events are search engines. Note also that no replies should be sent for these events.

These events all use an identical data structure, the `typeHitInfo`, which looks like this:

```
struct HitInfo {
    long   hitNumber;
    long   docID;
    long   relevance;
};
```

Event: New Hit

Requested Action: Add a new hit to the list of hits in progress.

Event Class kAETextRetrievalSuite

Event ID kAEPDNewHit

Parameters:

keyAEDirectObject

Description:	New hit information
Descriptor Type:	typeHitInfo
Required or Optional	required

Event: Obsolete Hit

Requested Action: Delete a previously-existing hit from the list of hits in progress.

Event Class kAETextRetrievalSuite

Event ID kAEPDObsoleteHit

Parameters:

keyAEDirectObject

Description:	Obsolete hit information
Descriptor Type:	typeHitInfo
Required or Optional	required

Event: Changed Hit

Requested Action: Change information about a previously-existing hit.

Event Class kAETextRetrievalSuite

Event ID kAEPDChangeHit

Parameters:

keyAEDirectObject

Description:	Changed hit information
Descriptor Type:	typeHitInfo
Required or Optional	required

Note that for the Changed Hit event, the client is required to match the hitInfo to a hit which was previously sent. The docID is guaranteed to be invariant, but the hitNumber or relevance scores may change, perhaps at the same time. Good techniques for matching a new to an old hitInfo will not be discussed here.

## Progressive Disclosure Buffers

In an ideal world the engine would always be able to send events to the client with information about current status. However, if the engine resides on a different machine from the client, the user of the client machine would need to have made the engine an authorized user and everything would get more complicated.

To simplify this problem, there is a mechanism which clients can use to get current information from the engine: each query has properties called pProgressText and pPercentDone, and each query has a property called pProgressBuffer. When the client reads the pProgressText or the pPercentDone property, the engine should return information appropriate to the current state. When the client reads the pProgressBuffer property, the engine should return an AEList of typeHitEventInfo elements which represent what's happened with the hit list since that last time the buffer was read. Note that the pProgressBufferActive property of the query must be TRUE to tell the engine to maintain the information which will allow it to properly fill the buffer.

The typeHitEventInfo mentioned above includes the typeHitInfo described in the "Progressive Disclosure Events" section:

```
struct HitEventInfo {
    Byte    eventType;
    HitInfo hit;
};
```

The possible values for eventType are: kNewHit, kObsoleteHit, and kChangedHit. See the "Progressive Disclosure Events" section for information on how to interpret the data.



---

## Usage Scenarios

It's hard to get a good idea of how useful a suite will be just from the description of events and classes. Included below are examples of two usage scenarios which will probably be fairly common.

### Creating a New Index and Adding Documents

Following are descriptions of events sent from client to server for a situation in which the client wishes to create a new index and add documents to it.

#### Begin Transaction

**Action:** Registers the client with the engine.  
**Parameters:** < none >  
**Reply parameters:** keyAETheTransID: the "client ID" which the engine assigns to the client for that session. [Right now, this ID is only used in the End Transaction event when the client is finished with the engine.]

#### New Element

**Action:** Creates a new empty index.  
**Parameters:** keyAEObjectClass: (cTextIndex)  
keyAEData: typeTextIndex describing the index, which may be null if the client wishes to set the parameters using Set Data events.  
**Reply parameters:** keyDirectObject: Object specifier for the index

#### Open

**Action:** Opens the just-created index.  
**Parameters:** keyDirectObject: Object specifier (or typeTextIndex with enough information to uniquely specify the index).  
**Reply parameters:** keyIndexRefNum: RefNum for the index during this session. (Do we really need this?)

#### Begin Transaction

**Action:** Starts a new transaction which will allow us to change the index.  
**Parameters:** keyDirectObject: Object specifier for the index  
**Reply parameters:** keyAETheTransID: Transaction ID to use.  
keyTransactionStatus: < currently undefined >

#### New Element

**Action:** Adds a document to the transaction.  
**Parameters:** keyAETheTransID: Transaction ID to use.  
[keyAEInsertHere: Object specifier for the index] (We don't really need this-- we know the transaction id.)  
keyAEObjectClass: (cDocumentWithText)  
keyAEData: typeDocumentWithText describing the document (could be a vRefNum/fileID or actual text)  
**Reply parameters:** keyTransactionStatus: < currently undefined >

**NOTE:** The "New Element" event is sent once for every document or set of documents. To represent a set of documents, pass cAEList as the object class and pass a list of cDocumentWithTexts as the data.

#### Process Transaction

**Action:** Tells the engine that there is no more material to be added to the transaction, and that it should prepare to incorporate the transaction's material into the index.  
**Parameters:** keyAETheTransID: Transaction ID to use.  
**Reply parameters:** keyTransactionStatus: < currently undefined >

#### End Transaction

**Action:** Tells the engine what to do with the material in the transaction: either to incorporate it into the index, or to discard it.  
**Parameters:** keyAETheTransID: Transaction ID to use.  
keyAEAbortTrans: if present, the transaction is to be aborted. Otherwise, the transaction is to be committed.  
**Reply parameters:** keyTransactionStatus: < currently undefined >

#### End Transaction

**Action:** Tells the engine that we're done using it for the moment.  
**Parameters:** keyAETheTransID: the "client ID" obtained from the very first "Begin Transaction" event  
**Reply parameters:** < none >

### Opening an Existing Index and Searching

Following are descriptions of events sent from client to server for a situation in which the client wishes to open an existing index and perform queries on it.

#### Begin Transaction

**Action:** Registers the client with the engine.  
**Parameters:** < none >  
**Reply parameters:** keyAETheTransID: the "client ID" which the engine assigns to the client for that session. [Right now, this ID is only used in the End Transaction event when the client is finished with the engine.]

#### Open

**Action:** Opens the index.  
**Parameters:** keyDirectObject: Object specifier (or typeTextIndex with enough information to uniquely specify the index).  
**Reply parameters:** keyIndexRefNum: RefNum for the index during this session.  
(Do we really need this?)

#### New Element

**Action:** Creates a new query.  
**Parameters:** keyAEObjectClass: (cTextQuery)  
keyAEData: typeTextQuery describing the query  
**Reply parameters:** keyDirectObject: Object specifier for the query

## Evaluate Query

<b>Action:</b>	Tells the query to begin executing itself on a certain index or set of indices.	
<b>Parameters:</b>	keyDirectObject:	Object specifier for the query
	keyAESearchWhere:	Object specifier for the index (or list of specifiers for multiple indices)
	keyAESendHitList:	false (don't send the hit list)
	keyAESendDone:	true (send a reply when completed)
	keyAETrickleHits:	false (don't send hits one by one)
<b>Reply parameters:</b>	keyAETrickleProgress:	false (don't send progress events)
	keyDirectObject:	Object specifier for the index

(Client does other things until the reply comes in...)

## Get Data

<b>Action:</b>	Tells the engine to send the hit list generated by the query.	
<b>Parameters:</b>	keyDirectObject:	Object specifier for the query or the hit list or a range of hits from the hit list.
	keyRequestedType:	typeHitList
<b>Reply parameters:</b>	keyDirectObject:	The hit list.

## End Transaction

<b>Action:</b>	Tells the engine that we're done using it for the moment.	
<b>Parameters:</b>	keyAETheTransID:	the "client ID" obtained from the very first "Begin Transaction" event
<b>Reply parameters:</b>	< none >	

---

## Key forms defined in the Text Retrieval suite

Table 5 lists the key forms defined in the Text Retrieval suite.

**Table 5** Key forms defined in the Text Retrieval suite

Key form constant	Description
formAbsolutePosition	Specifies the position of an element in relation to the beginning or end of its container (for example, "word 5 of . . ."), or specifies one or more elements with a constant defined in the Apple Event Object Support such as kAEFirst (for example, "the <i>first</i> word in paragraph 12 . . .") or kAEAll (for example, " <i>all</i> the words in paragraph 12 . . .")
formName	Specifies an element by its name (for example, "the document <i>whose name is 'MyDoc'</i> ")
formPropertyID	Specifies a property of an object by its four-character property ID (for example, " <i>the font</i> of word 1")
formRange	Specifies a list of elements between two other elements (for example, "the words <i>between 'Wild' and 'Zanzibar', inclusive</i> ")
formRelativePosition	Specifies an element immediately before or after a container (for example, " <i>the next</i> word <i>after</i> the words whose style is bold")
formTest	Specifies one or more elements that pass a test; values of one or more properties or elements can be tested (for example, "the first paragraph <i>that is centered and that begins with the word 'Wild'</i> ")

For complete information about these key forms, see the *Apple Event Object Support Library Developer Note*.

---

## Comparison operators defined in the Text Retrieval suite

Table 6 lists the comparison operators defined in the Text Retrieval suite.

n **Table 6** Comparison operators defined in the Core suite

Comparison operator constant	Operator	Meaning
kAEEquals	' = '	The value of the first operand is equal to the value of the second operand.
kAEGreaterThan	' > '	The value of the first operand is greater than the value of the second operand.
kAEGreaterThanEquals	' > = '	The value of the first operand is greater than or equal to the value of the second operand.
kAELessThan	' < '	The value of the first operand is less than the value of the second operand.
kAELessThanEquals	' < = '	The value of the first operand is less than or equal to the value of the second operand.

## Constants defined in the Text Retrieval suite

Table 7 lists the constants defined in the Text Retrieval suite.

n **Table 7** Constants defined in the Text Retrieval suite

Constant	Value
cTRApplication	'cbmb'
cTextIndex	'cHip'
cTextQuery	'cltm'
cHitList	'cwmb'
cHit	'cblp'
cIndexType	'crac'
cOccurringWordsDict	'cluc'
cOccurringWord	'cgil'
cDocumentWithText	'cmtv'
typeHitList	'twmb'
typeRelevantHit	'rblp'
typeHit	'tblp'
errAESMGeneralError	-6660 /* preliminary! */
errAEUnknownSession	-6661 /* preliminary! */

errAETooManySessions	-6662	/* preliminary! */
kAEEvaluateQuery	'vist'	
kAEProcessTransaction	'ptrn'	
kAEStartSession	'sssn'	
kAEEndSession	'essn'	
kAEPDProgressReport	'prgr'	
kAEPDNewHit	'pnht'	
kAEPDObsoleteHit	'splc'	
kAEPDChangeHit	'ospc'	
keyAEHitList	'smic'	
keyAEHowBusy	'lgcb'	
keyAEMaxHitsToSend	'bobb'	
keyAERelevanceHits	'orga'	
keyAESendHitList	'anmj'	
keyAETextQuery	'jjbn'	
keyAETrickleHits	'brni'	
keyAETrickleProgress	'frgy'	
keyAEUpdateProgressBuffer	'chly'	
keyAEVerifyHits	'snpy'	
keyAECurrentStatus	'bntl'	
keyAEPercentDone	'aotb'	
pAlias	'ethe'	
pBusyPercentage	'mdup'	
pCanAbort	'ojer'	
pCanKeepDicts	'hisf'	
pCanKeepDocs	'llpr'	
pCanEncrypt	'anin'	
pCanRelevant	'ifyo'	
pCountTypes	'eare'	
pDictInfoTypes	'enow'	
pDocID	'dian'	
pDocsSize	'thos'	
pDocumentType	'odin'	
pEncryptionAlgorithm	'hiso'	

pExecutionDateTime	'rest'
pExplanationText	'ment'
pExtent	'avor'
pFieldID	'itet'
pHasProvisionalHits	'obab'
pKeepsDocs	'uran'
pImpenetrableIndexParams	'ufig'
pIndex	'icht'
pIndexParams	'lybe'
pIndexSize	'itek'
pIndexType	'napp'
pIsEncrypted	'inwh'
pIsLocal	'tfor'
pMajorType	'utth'
pMaxDocsSize	'ofyo'
pMaxHitsDesired	'youo'
pMaxIndexSize	'eryp'
pMaxNumDocuments	'urse'
pMaxTotalSize	'nkan'
pMinorType	'feat'
pMinRelevanceDesired	'hify'
pOccurrenceCounts	'indo'
pOccurringWordsDict	'spen'
pOffset	'uret'
pOwnerKey	'doin'
pPercentDone	'timE'
pProgressBuffer	'taKe'
pProgressText	'utto'
pQueryRefNum	'enwr'
pQuerySyntax	'lbev'
pQueryText	'lffo'
pRefNum	'emyo'
pRelevant	'tHat'
pRelevance	'osoa'
pRestrictionQuery	'firs'
pRestrictionRelation	'tall'
pScriptTag	'itwi'
pSearchComplete	'rwha'
pStemmingAlgorithm	'leli'
pStopwordDoc	'lish'

pStorageFree	'uwil'
pStorageTotal	'stab'
pSuperimposedCodingParams	'ouar'
pSynonymDoc	'ttod'
pText	'ndhe'
pTotalSize	'gand'
pTotalHits	'inge'
pUserKey	'tyou'
pVerified	'roud'
pWillIndexRemotely	'ndyo'
pWordCountsGuaranteed	'lunc'