DRE 3

revision 2

Administrator's Guide

Table of Contents

Autonomy

Autonomy employs a fundamentally different and unique combination of technologies to enable computers to form an understanding of a page of text, web pages, e-mails, voice, documents and people.

Autonomy's solution is therefore able to power any application dependent upon unstructured information within every market sector, including: e-commerce, customer relationship management, knowledge management, enterprise information portals and online publishing applications.

This is evidenced by the significant penetration of the technology in a diversity of vertical markets and has been achieved principally because every market sector needs to manage and leverage the benefits of unstructured information.

Autonomy was founded in 1996 and has offices in Boston, Chicago, Dallas, San Francisco, New York, and Washington, D.C. in the United States, as well as offices throughout EMEA, including Amsterdam, Brussels, Cambridge, Frankfurt, Milan, Paris, Oslo, and Sydney. In July 1998, the company went public on the EASDAQ exchange (EASDAQ:AUTN). Autonomy floated on The NASDAQ National Market (NASDAQ: AUTN) in May 2000, and on the London Stock Exchange (LSE: AU.) in November 2000.

To contact Autonomy, please get in touch with your nearest location listed below.

Europe and South Pacific

Autonomy Systems Ltd.
Cambridge Business Park
Cowley Road
Cambridge
CB4 0WZ

Help Desk:	+44 (0) 800 0 282 858	
Switchboard:	+44 (0) 1223 448 000	
Fax:	+44 (0) 1223 448 001	
E-mail	for information:	autonomy@autonomy.com
	for support:	uksupport@autonomy.com

The Help Desk operates from 9.30 am to 6.00 pm (GMT) Monday to Friday.

Website: www.autonomy.com

USA

Autonomy Inc.
301 Howard Street
22nd Floor
San Francisco
CA 94105

Help Desk:	+1 877 333 7744	
Switchboard:	+1 415 243 9955	
Fax:	+1 415 243 9984	
E-mail	for information:	info@us.autonomy.com
	for support:	support@us.autonomy.com

The Help Desk operates from 9.30 am to 6.00 pm (CST) Monday to Friday, toll-free.

Website: www.autonomy.com

Welcome

Thank you for choosing Autonomy and welcome to your Classification Server ™ version 2.0.0 Administrator's Guide.

Autonomy Solutions

Autonomy solutions are built in a modular architecture, based on the Autonomy Application Builder™ (API). They provide an infrastructure that is fully scalable and customizable according to customers' present and future needs. Autonomy solutions include:

- **Portal-in-a-Box™**, our comprehensive and fully-automated Information Portal for content-rich Internet and Intranet sites,
- **ActiveKnowledge™**, which conducts a real-time analysis of the ideas involved in the content of any opened application and provides real-time links to relevant internal and external information,
- **i-WAP™**, an add-on to Portal-in-a-Box™, automating the delivery and personalization of timely and relevant information to mobile users.
- **Autonomy Server™** provides a fully automated and precise means of retrieving information. It allows content to be queried in any language and any format, wherever it is stored, and presented with hyperlinks to similar information, automatically and in real-time.
- **Autonomy Update™** automatically creates a personalized report, informing individual users of developments that are relevant to their specific roles and interests. The alert capability enables users to keep track of late-breaking news, automatically and in real-time.
- **Autonomy Application Builder™** is a toolkit that enables companies and partners to customize Autonomy's products according to their individual requirements. The Application Programming Interfaces (APIs) have been modularized into functional suites to allow developers to plug in only those modules required for a given application. The APIs are distributed with the Autonomy Server™ and a set of sample code.

1. Introduction

At the heart of Autonomy's software is the Dynamic Reasoning Engine (DRE), a scalable, multithreaded process, which is based on advanced pattern-matching technology that exploits high-performance probabilistic modeling techniques.

The DRE performs the following core information operations:

Concept matching

The DRE accepts a piece of content (a sentence, paragraph or page of text, the body of an e-mail, a record containing human readable information, or the derived contextual information of an audio or speech snippet) or reference (identifier) as input and returns references to conceptually related documents ranked by relevance, or contextual distance. This is used to generate automatic hyperlinks between pieces of content.

Agent creation

The DRE provides the conceptual information that is need to create agents.
The DRE accepts a piece of content (training text, a document or a set of documents) or reference (identifier) and returns an encoded representation of the concepts, including each concept's specific underlying patterns of terms and associated probabilistic ratings.

Agent retraining

The DRE accepts an agent and a piece of content (training text, a document or a set of documents) and adapts the agent using the content.

Agent Matching

The DRE accepts an agent and returns similar agents ranked by conceptual similarity. This is used to discover users with similar interests, or find experts in a field.

Agent Alerting

The DRE accepts a piece of content (a sentence, paragraph or page of text, the body of an e-mail, a record containing human readable information, or the derived contextual information of an audio or speech snippet) and returns similar agents ranked by conceptual similarity. This is used to discover users who are interested in the content, or to find experts in a field.

Categorization

The DRE accepts a piece of content and returns categories ranked by conceptual similarity. This is used to find out which categories the content is most appropriate for, allowing subsequent tagging, routing or filing.

Summarization

The DRE accepts a piece of content and returns a summary of the information containing the most salient concepts of the content. In addition, summaries can be generated that relate to the context of the original inquiry - allowing the most applicable dynamic summary to be provided in the results of a given inquiry.

Clustering

The DRE, in conjunction with the Clusterizer module, organizes large volumes of content or large numbers of profiles into self-consistent clusters. Clustering is an automatic agglomerative technique, which partitions a corpus by grouping together information containing similar concepts.

Active matching

The DRE accepts textual information describing the current user task and returns a list of documents ordered by contextual relevance to the active task.

Retrieval

The DRE accepts a Boolean term or natural language query and returns a list of documents containing the terms ordered by contextual relevance to the query.

Query types

Once the DRE has been configured for a particular installation, it operates automatically, handling a number of different query types:

Natural Language

Natural language text is submitted as a query.

Boolean and bracketed Boolean

Standard Boolean AND/OR/NOT searches.

Fuzzy queries

If a search string is not quite accurate (for example, if it contains spelling mistakes) a fuzzy query returns results that contain words hat are similar to the entered string.

Proximity search

Words that appear close together in the search string are given higher weighting.

Soundex keyword search

If the spelling of a keyword is not quite accurate but phonetic a Soundex keyword search returns results that contain the keyword and phonetically similar keywords (using the Soundex algorithm).

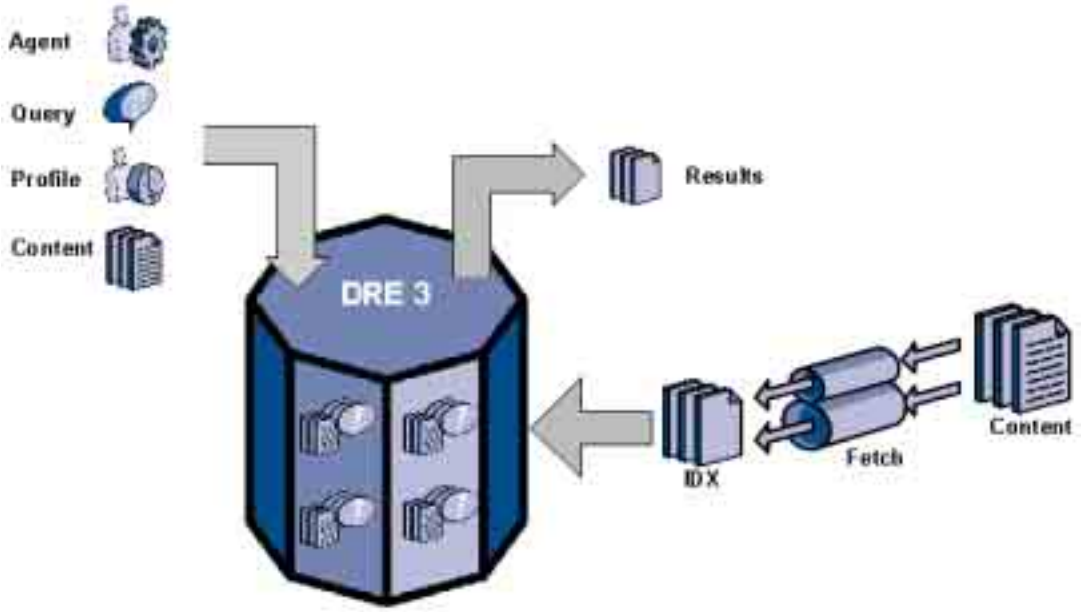
Proper names

Names are recognized and kept as a unit.

Thesaurus

If you add a Thesaurus database to your DRE, a search will not only return results that contain the words contained in the search string but also results that contain synonyms for the words in the search string.

System architecture



Documents in IDX format are indexed into the DRE (manually or using a Fetch). The DRE stores the concepts of the document, and in response to queries, agents, profiles or content returns a link to the result document as well as a percentage weighting, which indicates how relevant this result document is to the original query.

2. Installing the DRE

System requirements

Platforms supported
Microsoft Windows NT 4, 2000 and XP
Linux
Solaris
AIX 4.3
HPUX 11
Tru64

Note: DRE 3 also supports other POSIX UNIX versions on request.

Minimum server specification
Pentium CPU
256 MB RAM
Intel or Intel Compatible Processor

Note: this specification is dependent on the amount of data that is stored in the DRE. If you want to run Autonomy connectors with your DRE, you should note that due to substantially different disk usage patterns it is beneficial to run connector and DRE processes on separate drives or partitions.

3. Configuring the DRE

The settings that determine how the DRE operates are contained in the <InstallationName> configuration file, which is located in your installation directory. You can modify these settings in order to customize the DRE according to your requirements.

Entering Boolean values

For parameters that require Boolean settings the following settings are interchangeable

TRUE = true = ON = on = Y = y = 1

FALSE = false = OFF = off = N = n =0

Entering string values

If the value that you want to enter for a parameter that requires a string contains quotation marks, you must put the value into quotation marks and escape each quotation mark that the string contains by putting a slash in front of it.

For example:

FIELDSTART0="<font face=\"arial\"size=\"+1\"><b>"

Here the beginning and end of the string is indicated by quotation marks while all quotation marks that are contained in the string are escaped.

If you want to enter a comma separated list of strings for a parameter, and one of the strings contains a comma, you must indicate the start and the end of this string with quotation marks.

For example:

ParameterName=cat,dog,bird,"wing,beak",turtle

If any string within a comma separated list contains quotation marks, you must put this string into quotation marks and escaped the quotation marks in the string by putting a slash in front of them.

For example:

ParameterName="<font face=\"arial\"size=\"+1\"><b>",dog,bird,"wing,beak",turtle

Applying modifications to the DRE's operation

New configuration settings only take effect once the DRE service is stopped and restarted.
Alternatively you can send the following command to the DRE in order to reset it:

http://<host>:<IndexPort>/DRERESET>

<host> Enter the IP address (or name) of the machine on which the DRE is installed.

<IndexPort> Enter the port by which data is indexed into the DRE.

Configuration file sections

The DRE configuration file contains the following sections:

[License]

[Service]

[Server]

[Schedule]

[Database]

[Fields]

[ExternalDaily]

[License] section

This section contains the licensing details. You should not edit this section, as this could stop the UAServer working.

Holder

The name of the license holder.

Key

The license key.

[Service] section

This section determines which machines are permitted to use and control the DRE service.

ServicePort

The DRE listens for incoming service commands on this port.

ServiceStatusClients

The IP address of machines that are permitted to access information about the DRE status but are not permitted to control it.

If you want to permit a number of machines to access the DRE status, you must separate the individual addresses with a comma. For example, **196.172.86.220,196.172.87.11** (note that there is no space before or after the comma).

Alternatively, you can use a wildcard in the IP address. Enter for example **187.*.*.*** to permit any machine whose IP address begins with 187 to access the DRE status.

ServiceControlClients

The IP address of machines that are permitted to control the DRE.

If you want to permit a number of machines to control the DRE, you must separate the individual addresses with a comma. For example, **196.172.86.220,196.172.87.11** (note that there is no space before or after the comma).

Alternatively, you can use a wildcard in the IP address. Enter for example **187.*.*.*** to permit any machine whose IP address begins with 187 to control the DRE.

[Server] section

QueryClients

One or more IP addresses for machines that are permitted to query the DRE.

If you want to enter multiple addresses, you must separate the individual addresses with a comma (with no space).

For example, **196.172.86.*,196.172.87.***

IndexClients

One or more IP addresses for machines that are permitted to index data into the DRE.

If you want to enter multiple addresses, you must separate the individual addresses with a comma (with no space).

For example, **196.172.86.*,196.172.87.***

QueryPort

The port number by which queries are sent to the DRE.

IndexPort

The port number by which indexing commands are sent to the DRE.

Combine

If you have enabled **ImportBreaking** for the indexing process of documents, you can set **Combine** to the following in order to specify how sections that documents have been split up into are returned for your query:

1

The DRE returns only the most relevant section of any document that has been split up. This is the default setting.

0

The DRE returns all relevant sections of any document that has been split up (the number of sections that the DRE returns is restricted by the number of results that it is allowed to display).

Note: the **ImportBreaking** parameter is set in the [Server] section of the Content DRE's configuration file.

SuggestTerms

This specifies the default number of terms that are used when performing a **Suggest** command during querying.

For example:

SuggestTerms =40

In this example the system takes the best **40** terms from the original document and uses them as a query, which will produce further relevant documents.

IndexNumbers

Allows you to specify how numbers are indexed. You can enter one of the following:

0

Numbers or words that contains numbers (for example, **Y2K**) are not indexed.

1

All numbers are indexed.

2

Numbers that are contained in words are indexed.

ProperNames

Enter **on** to index names that consist of more than several words as one word. The name **William Shakespeare**, for example, would be indexed as one word. This is the default setting.

Enter **off** if you want all names to be indexed as individual words.

RoundDays

Enter **on** in order to round dates to the nearest day when evaluating date range queries. This is the default.

Enter **off** if you don't want to round dates.

Soundex

If you are querying for a particular term, you can set Soundex to **true** in order to return documents that contain the same term or a phonetically similar term (using the Soundex algorithm). The default setting is **false**.

Default_Xoptions

Default_Xoptions sets defaults, which are passed to every query that is made to the DRE.

For example:

Default_xoptions= sortdate+withsummary

In this example Default_Xoptions ensures that the options **sortdate** and **withsummary** are passed to every query that is made to the DRE.

You can display all available xoptions by entering the following in the address bar of your web browser:

http://<DRE_IP_Address>:<DRE_Query_Port_Number>/qmethod=H

This displays the **Autonomy DRE Server Help Pages**. Click on **Query Commands** to display the **Autonomy DRE Server Query Port Commands** page, which lists all available Default_Xoptions in its **Query Modifiers** sections.

CharConv

The language in which the DRE is running. You can enter one of the following numbers:

0	European (this is the default)	**7**	Eastern European
1	Japanese	**8**	Russian WINANSI
2	Korean	**9**	Russian KOI8
3	Thai	**10**	Hebrew
4	Simplified Chinese	**11**	Greek
5	Traditional Chinese	**12**	Swedish
6	Traditional Chinese indexed as Simplified Chinese		

Note: please contact Autonomy for additional files, if you are using a non-European language.

TermSize

Allows you to specify the maximum number of characters that any term in the DRE can be. By default this is 10 (for English and other European languages). The recommended values for the different languages are:

10	English and other European languages	**30**	German
40	Korean	**20**	Thai
30	Japanese		

StripLanguage

This optional parameter allows you to select which language to use when stripping (**running**, for example, is stripped to **run**). You can enter one of the following options:

0	English	**9**	Advanced German
1	Conversion from UK to US English	**10**	French
2	no stripping	**11**	Greek
3	German	**12**	Swedish
4	Italian	**13**	Danish
5	Russian	**14**	Portuguese
6	Advanced English	**15**	Advanced Spanish
7	Spanish	**16**	Norwegian
8	Dutch		

For example:

StripLanguage=7

In this example the DRE applies a stemming algorithm that has been designed specifically for Spanish.

Note:

You should specify the 'advanced' setting for English (any variant of English) or German. However, if **StripLanguage** was set to **0**, **1** (for English) or **3** (for German) when content was indexed into the DRE, you need to ensure that the same setting applies when queries are sent to the DRE.

For languages that have not been implemented, use **6** or **2**.

SentenceBreaking

If you are using a language in which words are not delimited by spaces, you need to specify one of the following for **SentenceBreaking**:

For Thai:	**thaibreaking.dll** or **./thaibreaking.so**
For traditional or simplified Chinese:	**chinesebreaking.dll** or **./chinesebreaking.so**
For Japanese:	**japanesebreaking.dll** or **./japanesebreaking.so**
For Korean:	**koreanbreaking.dll** or **./koreanbreaking.so**

[Schedule] section

Compact

The minimum number of deleted documents required for the Compact operation to take place. (The Compact operation compresses the space that documents take by filling up the space that has been created through the deletion of documents by filling it up with new documents, similar to the defragmentation process).

Time

The time (hh:mm) when the Compact operation is performed for the first time. For example, **23:59**.

Interval

The number of hours (hh:mm) that elapses between individual Compact operations. For example, **24:00**.

Expire

Enter **1** if you want any of your databases to expire after the specified **Interval** has elapsed.

[Databases] section

NumDBs

The total number of databases that the Content DRE contains.

n=<DatabaseName>

The ***n*** number and the name of each database. Note that the databases listed must be in consecutive order and start with **0**.

For example:

0=ExternalDaily

1=ExternalArchive

2=InternalDaily

3=Research

[Fields] section

To be able to search for documents that have been fetched and indexed into your DRE not just by content but fields (for example, title, topic or source of the document), you must ensure that the fields that fetched documents contain are defined in the DRE configuration file's [Fields] section.

Note: you must define a DREREFERENCE and DRETITLE field, otherwise the DRE cannot start.

NumFields

The total number of fields that you are defining in the DRE.

For example:

```
0=DREREFERENCE
1=INDEX10.DRETITLE
2=image
3=summary
```

In this example four fields are specified, which means that NumFields must be specified as follows:

```
NumFields=4
```

n=<FieldName>

The ***n*** number and name of each field. Note that the fields listed must be in consecutive order and start with **0**.

For example:

```
0=DREREFERENCE
1=INDEX10.DRETITLE
2=image
3=summary
4=paper
5=author
6=checksum
7=dredoctype
8=bias
```

Note: you can add the prefix **Index<WeightingPercentage>**. to a field in order to index this field and give it a higher weighting. INDEX10.DRETITLE, for example, indicates that this field should be indexed and given a 10% higher percentage weighing when a query is sent to the DRE.

[MyDatabase] sections

The [MyDatabase] sections are optional sections that allows you to specify settings that you only want to apply to the listed databases. You can specify the following settings:

ExpireTime

The number of hours after which a file in the specified database expires (that is it is removed from this database).

ExpireIntoDataBase

The database to which files are moved when they expire.

4. Indexing documents into a DRE

Before you can index files into a DRE they must be imported into IDX file format. You can import files into IDX format:

using a Connector

The Autonomy Connectors (for example, AutoIndexer, HTTPFetch, Oracle Fetch and so on) allow you to retrieve documents from different repositories and import them into IDX file format. Please refer to the appropriate Connector manual for further information on how to import documents.

using the DRE Administration dialog

If you are running on a Windows platform, you can use the DRE Administration dialog to import documents into IDX file format.

manually

You can create a text file, which contains the information that you want to index into your DRE in specific DRE fields.

Once documents have been imported into IDX file format, you can index them into a DRE:

using a Connector

The Autonomy Connectors allow you to index the IDX files that they have created into the DRE that they connect to. Please refer to the appropriate Connector manual for further information on how to index documents.

using the DRE Administration dialog

If you are running on a Windows platform, you can use the DRE Administration dialog to index documents into a DRE.

directly

You can index IDX files into a DRE using an HTTP request that you can issue from your web browser.

over a socket

You can index IDX files over a socket into a DRE. This requires a POST request method.

Indexing process: order of events

Depending on where the data that is indexed into the DRE is located, the indexing process takes place in one of the following orders.

If the DRE indexes a locally accessible file:

1. The DRE receives a filename.
2. The DRE opens the file and reads the data.
3. The indexing process takes place.

If the DRE receives data over the indexing port:

1. The DRE receives a stream of data over the port.
2. The DRE saves the data locally.
3. The DRE opens the file and reads the data.
4. The indexing process takes place.

Creating IDX files: using the DRE Administration dialog

If you are running on a Windows platform, you can use the DRE Administration dialog to import documents into IDX file format.

To create IDX files using the DRE Administration dialog:

1. In your DRE installation directory, double-click on the **dreadmin.exe** file. The **DRE Administration** dialog is displayed.
2. Display the **Import-Index** page and click on the **Import Files into IDX format** button. The **Main Import Settings** dialog is displayed.
3. In the **Import to file** field, enter the path to the IDX file that you want to create. By default this is **C:\Autonomy\<Installation_Name>\import1.idx**, but you can use the **Browse** button to navigate to another location.
4. If you want to append new data to the IDX file rather than overwrite the IDX file every time that you add new data for importing, check the **Append to output file** box.
5. Use the **Add Folders** and **Add Files** button to the **Input Files to Import** list. All documents that are contained in the list will be imported into IDX file format.

   Note: you can use the **Remove File** button to remove items from the list.
6. From the **Destination Database** drop-down list, select the DRE database into which you want to index the listed files.
7. To specify how the documents that you have listed in the **Input Files to Import** list should be imported, click on the **Set Import Parameters** button to display the **Import Module settings** dialog. Please refer to **Appendix A** for details on the **Import Module settings** dialog.
8. Close the **Import Module settings** dialog once you have specified the settings that are appropriate for the files that you want to import into IDX file format.
9. Click on the **OK** button to import the listed files into IDX file format and to close the **Main Import Settings** dialog. You can now index the IDX files into the DRE (see **Indexing IDX files into the DRE: using the DRE Administration dialog**).

Note: for further details on the DRE Admin dialog, please refer to **Appendix A**.

Creating IDX files: manually

To manually create an IDX file, you need to create a text file, which contains the information that you want to index into your DRE in DRE fields. The DRE fields give the data a format that can be indexed.

DRE fields:

#DREREFERENCE	Enter a unique reference string for the document. Usually this is a file name, URL or a unique code number.
#DRETITLE	Enter the title of the document. You can enter multiple lines.
#DRECONTENT	Enter the content of the document. You can enter multiple lines. (This parameter is optional. However, if you don't enter **#DRECONTENT**, you should specify one or more **#DREFIELD *NameN*=** as otherwise the document will not have any content).
#DREFIELD Name*n*	Specify the name of each DREFIELD that you are defining, and enter an appropriate value for it. For example, if you want to index customer details: #DREFIELD surname1="Smith" #DREFIELD forename1="Peter" #DREFIELD title1="Mr." #DREFIELD surname2="Miller" #DREFIELD forename2="Susan" #DREFIELD title2="Dr." **Note:** if your document only contains one instance of the DREFIELD that you are defining, you do not need to add a qualifier to the name of the field. For example: #DREFIELD company="Autonomy" (This parameter is optional. However, if you don't enter **#DREFIELD *NameN*=**, you should specify **#DRECONTENT** as otherwise the document will not have any content).
#DRETYPE	Enter the format type of the document (this can only be **#DRETYPE text**).
#DREDATE	Enter the creation date of the document using the format that you have specified for **DateFormat** in the DRE configuration file. By default this **yyyy/mm/dd**.

#DREDBNAME	Enter the name of the database into which you want to index the document.
#DRESTORECONTENT	Enter **y**, if you want to store the document's content in the DRE, or **n** if you don't want to store the content.
#DRESECTION *n*	If you are indexing a large document and want to split it up into smaller sections, you can give each section a DRESECTION number in order to index the defined sections as individual documents into the DRE. **Note:** If you split up a document into sections: the first section must be **#DRESECTION 0** the section numbers must be in numerical order apart from the #DRESECTION number and the #DRECONTENT each section must contain the same DRE field values. (For further information, please refer to **Sectioning a document**).
#DREENDDOC	Indicates the end of the document. You must enter this delimiter.

Note: The text file must start with **#DREREFERENCE** and end with **#DREENDDOC**.

Example text file

The following is an example of a text file that can be indexed into a DRE:

```
#DREREFERENCE 392348A0
#DREFIELD authorname1="Brown"
#DREFIELD authorname2="Edgar"
#DREFIELD title="Dr."
#DREDATE 1998/08/06
#DRETITLE
Jurassic Molecules
#DRECONTENT
Scientists announced last week the successful reproduction of a
possible precursor to all life on Earth. The molecules consist of a
part of DNA and the molecular "scissors" responsible for destroying
messenger RNA in humans.
```

```
Using a technique called test tube evolution, scientists created a
nucleic acid enzyme, the first known enzyme that uses an amino acid
to start chemical activity. Scientists hope that the creation of this
molecule will lead to the elusive precursor. The precursor, by
definition, will have to contain both the genetic code for
replication and an enzyme to trigger self replication.


#DRETYPE text
#DREDBNAME Science
#DRESTORECONTENT y
#DREENDDOC
```

Sectioning a document

If a document that you want to index contains more than 500 words, you should split it up into sections in order to make it more manageable for the DRE.

Declare a separate document for each of the sections into which you are splitting the original document, and give each section a **#DRESECTION** number. Note that if you split up a document into sections:

- the first section must be **#DRESECTION 0**
- the section numbers must be in numerical order
- you must put the content of each section into the #DRECONTENT field
- no #DRECONTENT field should contain more than 500 words.
- apart from the #DRESECTION number and the #DRECONTENT each section must contain the same DRE field values.

Example text file

The following is an example of a text file, in which a document has been split up into sections:

```
#DREREFERENCE 392348A0
#DREFIELD authorname1="Brown"
#DREFIELD authorname2="Edgar"
#DREFIELD title="Dr."
#DREDATE 1998/08/06
#DRETITLE
Jurassic Molecules
```

```
#DRESECTION 0

#DRECONTENT

Scientists announced last week the successful reproduction of a
possible precursor to all life on Earth. The molecules consist of a
part of DNA and the molecular "scissors" responsible for destroying
messenger RNA in humans.

Using a technique called test tube evolution, scientists created a
nucleic acid enzyme, the first known enzyme that uses an amino acid
to start chemical activity. Scientists hope that the creation of this
molecule will lead to the elusive precursor. The precursor, by
definition, will have to contain both the genetic code for
replication and an enzyme to trigger self replication.

At this point, no naturally occurring hybrid enzymes have been found.
Scientists speculate that such enzymes may exist in nature and most
certainly existed in Earth's early history.

#DRETYPE text

#DREDBNAME Science

#DRESTORECONTENT y

#DREENDDOC

#DREREFERENCE 392348A0

#DREFIELD authorname1="Brown"

#DREFIELD authorname2="Edgar"

#DREFIELD title="Dr."

#DREDATE 1998/08/06

#DRETITLE

Jurassic Molecules

#DRESECTION 1

#DRECONTENT

Scientists have known for some time that the key ingredients for life
are DNA, RNA, and proteins. An interesting chicken-egg dilemma has
developed: which came first, RNA, DNA, or proteins? Many believe that
a replicating RNA molecule is the likely precursor to all life on
Earth.

RNA serves as both a genetic molecule and an enzyme in the body,
which scientists believe strongly suggests the likelihood of an RNA
precursor to all life. They speculate that RNA was first, followed by
DNA, the much more stable of the two. It would serve as an efficient
storehouse for the genetic code. Proteins, better catalysts than RNA,
likely evolved later as well. At some point, the current three-based
system developed from the initial one-based system of RNA.
```

```
Scientists hope that these scissors molecules may also have practical
uses in medicine, since the molecules can efficiently shred specific
DNA. Theoretically, it may be possible to tailor such a molecule to
attack and shred harmful DNA from pathogenic organisms. These
molecules could be made to be activated only in specific
circumstances.

#DRETYPE text

#DREDBNAME Science

#DRESTORECONTENT y

#DREENDDOC
```

Indexing IDX files into the DRE: using the DRE Administration dialog

If you are running on a Windows platform, you can use the DRE Administration dialog to index documents into a DRE.

To index IDX files into the DRE using the DRE Administration dialog:

1. In your DRE installation directory, double-click on the **dreadmin.exe** file. The DRE Administration dialog is displayed.
2. Display the **Import-Index** page and click on the **Import Files into IDX format** button. The **Main Import Settings** dialog is displayed.
3. Use the **Add IDX Files** button to add the IDX files that you want to index to the list (note that if you have used the **Import Files into IDX format** button to create an IDX file, this file will already be listed).
4. If you want to change the path of an IDX file, click on the **Change IDX Files Path** button, and make the appropriate changes. (This is useful, if the DRE is not situated on the same machine as the IDX files. The location of the IDX files is not changed but the path to the IDX files is defined from the DRE host machine's point of view).
5. Enter one of the following indexing modes to determine how the DRE is going to index the IDX files:

   MATCH

   REFERENCE

   REFERENCEMATCH

   NONE
6. If you want the IDX files to be deleted once they have been indexed into the DRE, check the **Delete files after indexing** box.
7. If the IDX files that you want to index are not situated on the same machine as the DRE, check the **Use file based indexing** box.
8. Check the **Force index into database** box If you want to force the IDX files to be indexed into the database that you have selected from the drop-down list. If you don't check this box, the IDX files will be indexed into the database that was specified on the **Main Import Settings** dialog when you created the IDX file.
9. Click on the **Index Into DRE now** button to start indexing your IDX files.

Note: for further details on the DRE Admin dialog, please refer to **Appendix A**.

DREADD?: directly indexing IDX

http://<host>:<port>/DREADD?<mandatory_parameter>&<optional_parameters>

The **DREADD?** command allows you to index IDX files directly into a DRE. Note that parameters that you use with **DREADD?** override any equivalent settings that you may have specified in the DRE's configuration file.

Command parameters:

Mandatory:	**<file_name>**
	or
	<path>
Optional:	**KillDuplicates**=<kill_duplicates_option>
	Delete
	DREDBName=<database_name>

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the number of the index port by which files are indexed into the DRE.

<file_name>

The IDX file that you want to index.

<path>

The full path to the IDX file that you want to index.

<optional_parameters>

You can enter one or more of the following parameters (note that you must separate individual parameters with an ampersand):

KillDuplicates=<kill_duplicates_option>

You can enter the following kill duplicates option to determine how the DRE handles duplicate text:

NONE

The file is indexed.

REFERENCE

If data is indexed that has the same **DREREFERENCE** as a document that the DRE already contains, the DRE deletes the document that it already contains and replaces it with the new one.

REFERENCEMATCHnn

If a document is indexed that has the same **DREREFERENCE** as a document that the DRE already contains or if it is more than **nn** percent similar to the content of the document that the DRE already contains, the DRE deletes the document that it already contains and replaces it with the new one.

Note: using **ReferenceMatch*nn*** will slow down the indexing process as the DRE has to check the content of all documents that have the same DREREFERENCE as the one that you are indexing.

MATCHnn

If a document is indexed whose content more than nn percent similar to the content of a document that the DRE already contains, the DRE deletes the document that it already contains and replaces it with the new one.

Note: using **Match*nn*** will slow down the indexing process as the DRE has to check the content of all the documents it contains.

<FieldName1><FieldName2>…

Allows you to specify one or more fields that the document, which you are indexing, contains. If the DRE already contains a document that has the same fields, the DRE deletes the document that it already contains and replaces it with the new one.

Note:

- If you want to specify multiple fields, you must separate them with underscores (there must be no space before or after an underscore). You must also enter an underscore before the first field that you are specifying.
- Using the **_<FieldName1>_<FieldName2>…** parameter will slow down the indexing process as the DRE has to check the fields of all the documents it contains.

Delete

The DRE deletes the file after it has been indexed (you cannot use this parameter if you are indexing over a socket).

DREDBName=<database_name>

Allows you to specify the DRE database into which you want the document to be indexed.

DREADDDATA?: indexing data over a socket

DREADDDATA?<data>#DREENDDATA&<optional_parameters>

Note: This command requires a POST request method.

The **DREADDDATA?…#DREENDDATA** command allows you to index date over a socket into a DRE. Note that parameters that you use with **DREADDDATA?…#DREENDDATA** override any equivalent settings that you may have specified in the DRE's configuration file.

Command parameters:

Mandatory:	**<data>**
Optional:	**KillDuplicates**=<kill_duplicates_option>
	Delete
	DREDBName=<database_name>

<data>

The data that you want to index.

<optional_parameters>

You can enter one or more of the following parameters (note that you must separate individual parameters with an ampersand):

KillDuplicates=<kill_duplicates_option>

You can enter the following kill duplicates option to determine how the DRE handles duplicate text:

NONE

The file is indexed.

REFERENCE

If data is indexed that has the same **DREREFERENCE** as a document that the DRE already contains, the DRE deletes the document that it already contains and replaces it with the new one.

REFERENCEMATCHnn

If a document is indexed that has the same **DREREFERENCE** as a document that the DRE already contains or if it is more than **nn** percent similar to the content of the document that the DRE already contains, the DRE deletes the document that it already contains and replaces it with the new one.

Note: using **ReferenceMatch*nn*** will slow down the indexing process as the DRE has to check the content of all documents that have the same DREREFERENCE as the one that you are indexing.

MATCHnn

If a document is indexed whose content more than ***nn*** percent similar to the content of a document that the DRE already contains, the DRE deletes the document that it already contains and replaces it with the new one.

Note: using **Match*nn*** will slow down the indexing process as the DRE has to check the content of all the documents it contains.

<FieldName1><FieldName2>…

Allows you to specify one or more fields that the document, which you are indexing, contains. If the DRE already contains a document that has the same fields, the DRE deletes the document that it already contains and replaces it with the new one.

Note:

- If you want to specify multiple fields, you must separate them with underscores (there must be no space before or after an underscore). You must also enter an underscore before the first field that you are specifying.
- Using the **_<FieldName1>_<FieldName2>…** parameter will slow down the indexing process as the DRE has to check the fields of all the documents it contains.

Delete

The DRE deletes the data after it has been indexed (you cannot use this parameter if you are indexing over a socket).

DREDBName=<database_name>

Allows you to specify the DRE database into which you want the data to be indexed.

Checking if the indexing process was successful

You may want to check if the indexing of data into the Content DRE was successful. To do this:

1. In the **dre** directory in your installation directory, double-click on **<InstallationName>_ContentDREadmin.exe** to display the **Dynamic Reasoning Engine Administration** dialog.
2. In the **DRE Status** field on the **Administration** page, click on the **View Details** button to display the **DRE Status Details** dialog. The dialog displays the last 50 lines that have been logged in the DRE file, which allows you to trace the indexing process.

Alternatively, you can run your web browser and enter the following address:

http:// <IPAddress>:<QueryPort>/qmethod=v

<IPAddress>	Enter your computer's IP address.
<QueryPort>	Enter the number of the port that is used to send queries to the DRE.

The **qmethod=v** command displays a list of the data that has been indexed into the DRE.

4. Querying the DRE

Query and Suggest commands

You can send HTTP commands to the DRE from your web browser. The general syntax of these commands is as follows:

http://<host>:<query_port>/**qmethod=<qmethod>**&**<mandatory_parameters>**&**<optional_parameters>**

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<query_port>

Enter the query port by which commands are sent to the DRE.

<qmethod>

Enter the query method command that you want the DRE to execute (for example, **q**).

<mandatory_parameters>

Enter the parameters that the query method that you have specified requires (not all query method require parameters).

<optional_parameters>

You can enter optional parameters for the query method that you have specified (optional parameters are not available for all query methods).

Note: you must separate individual parameters with an ampersand.

qmethod=H: displaying DRE server help

http://<host>:<port>/qmethod=H

The **H** query method allows you to display the DRE server help pages. These pages provide an overview of the DRE's configuration file parameters, query commands and index commands.

Command parameters:

Mandatory:	none
Optional:	**none**

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the query port by which commands are sent to the DRE.

qmethod=q: querying a DRE

http://<host>:<port>/qmethod=q&<mandatory_parameter>&<optional_parameters>

The **q** query method allows you to submit a natural language, Boolean, bracketed Boolean or fuzzy query to a DRE (or a keyword search).

Command parameters:

Mandatory:	**querytext**=<text>
Optional:	**qnum**=<number_of_results>
	idnum=<minimum_percentage_relevance>
	high=<maximum_percentage_relevance>
	attachtoname=<database_names>
	fromdate=<from_date>
	todate=<to_date>
	xoptions=<x_option>

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the query port by which commands are sent to the DRE.

<text>

Enter your query text. You can use Boolean operators and **fname** field restrictions (please refer to **Boolean queries** and **Field specifiers** for details).

<optional_parameters>

You can enter one or more of the following parameters (note that you must separate individual parameters with an ampersand):

<number_of_results>

Enter the maximum number of results that you want the DRE to return for the query.

<minimum_percentage_relevance>

Enter the minimum percentage relevance that results must have to the specified **querytext** in order to be returned.

<maximum_percentage_relevance>

Enter the maximum percentage relevance that results can have to the specified **querytext** in order to be returned.

<database_names>

Enter one or more DRE databases in which results must be contained in order to be returned. If you want to specify multiple databases, you must separate them with plus symbols (there must be no space before or after a plus symbol).

<from_date>

Enter the earliest date that results can have to be returned.

<to_date>

Enter the latest date that results can have to be returned.

<x_option>

You can enter one or more of the available xoptions to further configure how results are returned (please refer to **Xoptions** for details). If you want to specify multiple xoptions, you must separate them with plus symbols (there must be no space before or after a plus symbol).

Example:

http://12.3.4.56:4000/qmethod=q&querytext=Listening to Mozart and other classical music is good for you

This command uses port **4000** to send the specified query to a DRE which is located on a machine with the IP address **12.3.4.56**.

In response to the command, the DRE returns documents that are conceptually similar to the query, for example:

```
numhits=6
Doc_name=http://c.moreover.com/click/here.pl?z16128731&z=28
URL_title=Bach versus Mozart in baby battle
Doc_id=84841
Doc_weight=66
links=MOZART,CLASSICAL,MUSIC
dbase=0
BLANK=
Image=moreover.com
Paper=
summary=By Iain Shedden, Music writer 06mar01 PLAY classical music to
your kids and they will grow up to be really clever. That is a theory
that has been bandied around for years, most recently when a 1997
study in the US suggested the works of Mozart were particu
DocType=
QuickSummary= By Iain Shedden, Music writer 06mar01 PLAY classical
music to your kids and they will grow up to be really clever. That is
a theory that has been bandied around for years, most recently when a
1997 study in the US suggested the works of Mozart were particularly
beneficial in stimulating young brains.
ConceptSummary=By Iain Shedden, Music writer 06mar01 PLAY classical
music to your kids and they will grow up to be really clever. Now an
American music lecturer is hoping to convince Australian parents
that, rather than Mozart, the under-4s should be spending their time
hanging out with Johann Sebastian Bach. It s all to do with
stimulating the spatial temporal reasoning areas of the brain with
complex music, Dr Shore claims.

...
```

qmethod=s: suggesting conceptually similar documents

http://<host>:<port>/qmethod=s&<mandatory_parameter>&<optional_parameter>

The **s** query method allows you to query the DRE with one or more documents. The DRE will return result documents that are conceptually similar to the documents with which it has been queried.

Command parameters:

Mandatory:	**querytext**=<docIDs_list> or **querytext**=<docrefs_list> and **xoptions**=useurl
Optional:	**qnum**=<number_of_results> **idnum**=<minimum_percentage_relevance> **high**=<maximum_percentage_relevance> **attachtoname**=<database_names> **fromdate**=<from_date> **todate**=<to_date> **xoptions**=<x_option>

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the query port by which commands are sent to the DRE.

<docid_list>

Enter the IDs of the document for which you want to return conceptually similar documents. If you want to specify multiple document references or IDs, you must separate them with plus symbols (there must be no space before or after a plus symbol).

<docref_list>

Enter the references of the document for which you want to return conceptually similar documents. If you want to specify multiple document references or IDs, you must separate them with plus symbols (there must be no space before or after a plus symbol).

xoptions=useurl

Indicates to the DRE that the querytext is a reference, not a docid.

<optional_parameters>

You can enter one or more of the following parameters (note that you must separate individual parameters with an ampersand):

<number_of_results>

Enter the maximum number of results that you want the DRE to return for the query.

<minimum_percentage_relevance>

Enter the minimum percentage relevance that results must have to the specified **querytext** in order to be returned.

<maximum_percentage_relevance>

Enter the maximum percentage relevance that results can have to the specified **querytext** in order to be returned.

<database_names>

Enter one or more DRE databases in which results must be contained in order to be returned. If you want to specify multiple databases, you must separate them with plus symbols (there must be no space before or after a plus symbol).

<from_date>

Enter the earliest date that results can have to be returned.

<to_date>

Enter the latest date that results can have to be returned.

<x_option>

You can enter one or more of the available xoptions to further configure how results are returned (please refer to **Xoptions** for details). If you want to specify multiple xoptions, you must separate them with plus symbols (there must be no space before or after a plus symbol).

Example:

http://12.3.4.56:4000/qmethod=s&querytext=84841

This command uses port **4000** to request documents that are conceptually similar to the document with the ID **84841** to a DRE which is located on a machine with the IP address **12.3.4.56**.

In response to the command, the DRE returns documents that are conceptually similar to the specified document, for example:

```
numhits=6
Doc_name=http://c.moreover.com/click/here.pl?z16109004&z=28
URL_title=Let classical music work its magic
Doc_id=79606
Doc_weight=64
links=MOZART,BACH,CLASSICAL,JOHANN,SEBASTIAN,MUSIC
dbase=0
BLANK=
Image=moreover.com
Paper=
summary=Let classical music work its magic Mike Archer Lake
commentary Posted March 5, 2001 MORE HEADLINES Johann Sebastian Bach:
Born March 21, 1685
DocType=
QuickSummary= Let classical music work its magic Mike Archer Lake
commentary Posted March 5, 2001 MORE HEADLINES Johann Sebastian Bach:
Born March 21, 1685. Died July 28, 1750. Coming to Mount Dora
Thursday, April 5, 2001.
ConceptSummary=Let classical music work its magic Mike Archer Lake
commentary Posted March 5, 2001 MORE HEADLINES Johann Sebastian Bach:
Born March 21, 1685. Bach is but one of the big names in classical
music headed to Mount Dora. Brahms will be there, along with Handel,
Mendelssohn, Mozart, Strauss, Rossini and Verdi.

...
```

qmethod=F: sending a fuzzy query to the DRE

http://<host>:<port>/qmethod=F&<mandatory_parameters>

The **F** query method allows you to send a fuzzy query to the DRE. A fuzzy query can contain words that are spelt incorrectly.

Command parameters:

Mandatory:	**querytext**=<text>
Optional:	**qnum**=<number_of_results>
	idnum=<minimum_percentage_relevance>
	high=<maximum_percentage_relevance>
	attachtoname=<database_names>
	fromdate=<from_date>
	todate=<to_date>
	xoptions=<x_option>

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the query port by which commands are sent to the DRE.

<**text>**

Enter your fuzzy query text. You can use Boolean operators and **fname** field restrictions (please refer to **Boolean queries** and **Field specifiers** for details).

<optional_parameters>

You can enter one or more of the following parameters (note that you must separate individual parameters with an ampersand):

<number_of_results>

Enter the maximum number of results that you want the DRE to return for the query.

<minimum_percentage_relevance>

Enter the minimum percentage relevance that results must have to the specified **querytext** in order to be returned.

<maximum_percentage_relevance>

Enter the maximum percentage relevance that results can have to the specified **querytext** in order to be returned.

<database_names>

Enter one or more DRE databases in which results must be contained in order to be returned. If you want to specify multiple databases, you must separate them with plus symbols (there must be no space before or after a plus symbol).

<from_date>

Enter the earliest date that results can have to be returned.

<to_date>

Enter the latest date that results can have to be returned.

<x_option>

You can enter one or more of the available xoptions to further configure how results are returned (please refer to **Xoptions** for details). If you want to specify multiple xoptions, you must separate them with plus symbols (there must be no space before or after a plus symbol).

Example:

http://12.3.4.56:4000/qmethod=F&querytext=a woman on the japanoise chrysantemum throne

This command uses port **4000** to send a request for the IDX content of the documents with the ID **119983** to a DRE which is located on a machine with the IP address **12.3.4.56**.

In response to the command, the DRE returns the document's IDX content, for example:

```
numhits=6
Doc_name=http://c.moreover.com/click/here.pl?z18824846&z=28
URL_title=Japan considers female succession to imperial throne
Doc_id=18840 Doc_weight=13
links=WOMAN,CHRYSANTHE,THRON
dbase=2
BLANK=
Image=news.ft.com
Paper=
summary=By Ken Hijino in Tokyo Published: May 9 2001 10
DocType=
QuickSummary= By Ken Hijino in Tokyo Published: May 9 2001 10:59GMT
Last Updated: May 9 2001 13:36GMT Senior Japanese politicians said on
Wednesday that the government may consider taking the highly
controversial step of revising imperial family rules to allow a woman
to ascend the Chrysanthemum Throne for the first time in over two
hundred and thirty years.
ConceptSummary=Mr Koizumi, however, also added that the succession to
the throne could develop into a large problem, and should be studied
carefully within the ruling Liberal Democratic Party. Although the
step would be in keeping with Mr Koizumi s maverick image, senior LDP
officials are keen to avoid any backlash if the move appears
unpopular or controversial. Japan s imperial family faces a
succession crisis with no male heir to the throne for the first time
in over three decades.

...
```

qmethod=z: returning a concept summary

http://<host>:<port>/qmethod=z&<mandatory_parameters>

The **z** query method allows you to return a concept summary for a document. A concept summary comprises sentences that are typical of the document's content (these sentences can be from different parts of the result document).

Command parameters:

Mandatory:	**querytext**=<docref> **idnum**=<number_of_sentences>
Optional:	none

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the query port by which commands are sent to the DRE.

<docref>

Enter the reference of the document for which you want to return a concept summary.

<number_of_sentences>

Enter the number of sentences that you want the concept summary to comprise.

Example:

http://12.3.4.56:4000/qmethod=z&querytext=84841&idnum=2

This command uses port **4000** to send a request for a **2** sentence concept summary of the document with the ID **84841**to a DRE which is located on a machine with the IP address **12.3.4.56**.

In response to the command, the DRE returns the document's IDX content, for example:

```
Now an American music lecturer is hoping to convince Australian
parents that, rather than Mozart, the under-4s should be spending
their time hanging out with Johann Sebastian Bach. It s all to do
with stimulating the spatial temporal reasoning areas of the brain
with complex music, Dr Shore claims.
```

qmethod=Z: returning a quick summary

http://<host>:<port>/qmethod=Z&<mandatory_parameters>&<optional_parameter>

The **Z** query method allows you to return a quick summary for a document. A quick summary comprises the first few words of the document.

Command parameters:

Mandatory:	**querytext**=<docref>
Optional:	**xoptions=words**=<number_of_words>

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the query port by which commands are sent to the DRE.

<docref>

Enter the reference of the document for which you want to return a quick summary.

<optional_parameter>

You can enter the following parameter:

<number_of_words>

Enter the number of words that you want the quick summary to comprise.

Example:

http://12.3.4.56:4000/qmethod=Z&querytext=84841&xoptions=words=21

This command uses port **4000** to send a request for a **21** word quick summary of the documents with the ID **84841**to a DRE, which is located on a machine with the IP address **12.3.4.56**.

In response to the command, the DRE returns the document's IDX content, for example:

```
By Iain Shedden, Music writer 06mar01 PLAY classical music to your
kids and they will grow up to be really clever.
```

qmethod=c: returning document content in text format

http://<host>:<port>/qmethod=c&<mandatory_parameters>

The **c** query method allows you to return a document's content in text format.

Command parameters:

Mandatory:	**querytext**=<docID> or **querytext**=<docref> and **xoptions**=useurl
Optional:	none

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the query port by which commands are sent to the DRE.

<**docID>**

Enter the ID of the document whose content you want to return.

<docref>

Enter the reference of the document whose content you want to return.

xoptions=useurl

Indicates to the DRE that the querytext is a reference, not a docid.

Example:

http://12.3.4.56:4000/qmethod=c&querytext=119983

This command uses port **4000** to send a request for the content of the documents with the ID **119983** to a DRE which is located on a machine with the IP address **12.3.4.56**.

In response to the command, the DRE returns the document's IDX content, for example:

```
By Bridget Gutierrez Express-News Staff Writer Holy Cross High School
doesn t have a performing arts department, let alone a theater, but
that didn t stop teacher Carmela Logan from founding the school s
Shakespeare Society. Craig Gonzales is one of fewer than a dozen Holy
Cross High School students who have become members of the West Side
school s fledgling Shakespeare Society. Photo by Gloria
Ferniz/Express-News Now in its second year, the troupe has competed
successfully in a series of recent college scholarship auditions, and
the group s founder has begun dreaming of a day when all West Side
schoolchildren are exposed to the English bard s famous works. I want
to use Shakespeare s literature to enhance and develop the language
arts, Logan said, because probably most of these children would never
have been exposed to it.
```

qmethod=C: returning the IDX content for a document

http://<host>:<port>/qmethod=C&<mandatory_parameters>

The **C** query method allows you to return a document's the IDX content.

Command parameters:

Mandatory:	**querytext**=<docID> or **querytext**=<docref> and **xoptions**=useurl
Optional:	none

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the query port by which commands are sent to the DRE.

<**docID>**

Enter the ID of the document whose content you want to return.

<docref>

Enter the reference of the document whose content you want to return.

xoptions=useurl

Indicates to the DRE that the querytext is a reference, not a docid.

Example:

http://12.3.4.56:4000/qmethod=C&querytext=119983

This command uses port **4000** to send a request for the IDX content of the documents with the ID **119983** to a DRE which is located on a machine with the IP address **12.3.4.56**.

In response to the command, the DRE returns the document's IDX content, for example:

```
#DREREFERENCE http://c.moreover.com/click/here.pl?z16417247&z=28
#DRETITLE Shakespeare group lures happy few #DREFIELD BLANK=""
#DREFIELD Image="moreover.com"
#DREFIELD Paper=""
#DREFIELD summary="By Bridget Gutierrez Express-News Staff Writer
Holy Cross High School doesn t have a performing arts department, let
alone a theater, but that didn t stop teacher Carmela Logan from
founding the school s Shakespeare Society. Craig Gonzales is one of
fewer "
#DREFIELD DocType=""
#DREWORDCOUNT 115
#DREDOCID 117547
#DRESECTION 0
#DRETYPE TEXT
#DREDATE 984644094
#DREDBNAME News
#DRESTORECONTENT y
#DRECONTENT By Bridget Gutierrez Express-News Staff Writer Holy Cross
High School doesn t have a performing arts department, let alone a
theater, but that didn t stop teacher Carmela Logan from founding the
school s Shakespeare Society. Craig Gonzales is one of fewer than a
dozen Holy Cross High School students who have become members of the
West Side school s fledgling Shakespeare Society. Photo by Gloria
Ferniz/Express-News Now in its second year, the troupe has competed
successfully in a series of recent college scholarship auditions, and
the group s founder has begun dreaming of a day when all West Side
schoolchildren are exposed to the English bard s famous works. I want
to use Shakespeare s literature to enhance and develop the language
arts, Logan said, because probably most of these children would never
have been exposed to it.
#DREENDDOC
```

qmethod=g: returning all documents from the DRE

http://<host>:<port>/qmethod=g

The **g** query method allows you to return all documents from a DRE. Note that if the DRE contains a lot of documents this command will take some time to execute.

Command parameters:

Mandatory: none

Optional: none

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the query port by which commands are sent to the DRE.

Example:

http://12.3.4.56:4000/qmethod=g

This command uses port **4000** to send a request for all documents that the DRE contains to the DRE which is located on a machine with the IP address **12.3.4.56**.

In response to the command, the DRE returns all documents, for example:

```
numhits=120821
Doc_name=http://c.moreover.com/click/here.pl?z18729464&z=28
URL_title=VSI Enterprises Featured in CEOCast.com Streaming Broadcast
Interview
Doc_id=0
Doc_weight=100
links=
dbase=2
BLANK=
Image=moreover.com
Paper=
summary=VSI Enterprises Featured in CEOCast.com Streaming Broadcast
```

```
Interview NORCROSS, Ga DocType=
QuickSummary= VSI Enterprises Featured in CEOCast.com Streaming
Broadcast Interview NORCROSS, Ga., May 7 /PRNewswire/ -- VSI
Enterprises, Inc. (OTC Bulletin Board: VSIN) is the topic of
CEOCast.com s feature story today. Business model evolution,
acceptance of the software- based Ongoer(TM) audio-visual control
system, and the impact of CycleFree(TM) programming methodology are
addressed in Editor Michael Wax s 15-minute audio interview with VSI
ConceptSummary=VSI Enterprises Featured in CEOCast.com Streaming
Broadcast Interview NORCROSS, Ga., May 7 /PRNewswire/ -- VSI
Enterprises, Inc. (OTC Bulletin Board: VSIN) is the topic of
CEOCast.com s feature story today.

...
```

qmethod=G: returning all documents from the DRE in IDX file format

http://<host>:<port>/qmethod=G

The **G** query method allows you to return all documents from a DRE in IDX file format. Note that if the DRE contains a lot of documents this command will take some time to execute.

Command parameters:

Mandatory: none

Optional: none

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the query port by which commands are sent to the DRE.

Example:

http://12.3.4.56:4000/qmethod=G

This command uses port **4000** to send a request for all documents that the DRE contains to the DRE which is located on a machine with the IP address **12.3.4.56**.

In response to the command, the DRE returns all documents in IDX format, for example:

```
numhits=120821
#DREREFERENCE http://c.moreover.com/click/here.pl?z18729464&z=28
#DRETITLE VSI Enterprises Featured in CEOCast.com Streaming Broadcast
Interview
#DREFIELD BLANK=""
#DREFIELD Image="moreover.com"
```

```
#DREFIELD Paper=""
#DREFIELD summary="VSI Enterprises Featured in CEOCast.com Streaming
Broadcast Interview NORCROSS, Ga"
#DREFIELD DocType=""
#DREWORDCOUNT 140
#DREDOCID 0
#DRESECTION 0
#DRETYPE TEXT
#DREDATE 989276683
#DREDBNAME ARCHIVE
#DRESTORECONTENT y
#DRECONTENT
VSI Enterprises Featured in CEOCast.com Streaming Broadcast Interview
NORCROSS, Ga., May 7 /PRNewswire/ -- VSI Enterprises, Inc. (OTC
Bulletin Board: VSIN) is the topic of CEOCast.com s feature story
today. Business model evolution, acceptance of the software- based
Ongoer(TM) audio-visual control system, and the impact of
CycleFree(TM) programming methodology are addressed in Editor Michael
Wax s 15-minute audio interview with VSI CEO Rick Egan. This bird s-
eye view of where VSI Enterprises and subsidiary Simtrol are headed
can be heard via the Internet at . CEO Cast, the website where Wall
Street listens, is a premier source of original and syndicated
streaming broadcast interviews of Chief Executive Officers at public
and private news-making companies. CEO Cast s programming is
distributed to millions of on-line investors at over 700 financial
web sites as well as to more than 20,000 portfolio managers, buy-side
analysts and traders at more than 3,300 North American institutions.
C

...
```

qmethod=a: returning the weight of terms

http://<host>:<port>/qmethod=a&<mandatory_parameter>&<optional_parameter>

Terms are given a weight according to their statistical importance in the DRE. They can have a weight between 0 and 255. The **a** query method allows you to return the weight of one or more specified terms.

Command parameters:

Mandatory: **querytext**=<words>

Optional: **xoptions**=strip

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the query port by which commands are sent to the DRE.

<words>

Enter the terms whose weight you want to return. If you want to specify multiple words, you must separate them with plus symbols (there must be no space before or after a plus symbol).

xoptions=useurl

Indicates to the DRE that the querytext is a reference, not a docid.

<optional_parameters>

You can enter the following parameter:

x_options=strip

You can add &**x_options=strip** to the querytext in order to stem the query terms.

Example:

http://12.3.4.56:4000/qmethod=a&querytext=genetic+engineering

This command uses port **4000** to send a request for the weight of the terms "genetic" and "engineering" to a DRE, which is located on a machine with the IP address **12.3.4.56**.

In response to the command, the DRE returns the weight of the specified terms, for example:

```
Number of hits: 2
genetic+138
engineering+110
248
genetic+engineering+
```

qmethod=e: returning the best terms in text

http://<host>:<port>/qmethod=e&<mandatory_parameters>&<optional_parameters>

The **e** query method allows you to send query text to the DRE and return the terms with the highest weight that this text contains.

Command parameters:

Mandatory:	**querytext**=<text>
	idnum=<number>
Optional:	**xoptions**=occ
	or
	xoptions=trueocc

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the query port by which commands are sent to the DRE.

<**text>**

Enter the text for which you want to return the specified number of highest weighing terms.

<number>

Specify how many terms you want to return.

<optional_parameters>

You can enter one of the following parameters:

xoptions=occ

Allows you to return the number of documents in which each <**text>** term occurs in the DRE

xoptions=trueocc

Allows you to return the total number of times each <**text>** term occurs in the DRE.

Example:

http://12.3.4.56:4000/qmethod=e&querytext=DNA was discovered as a major chemical of the nucleus at about the same time Mendel and Darwin published their work. However, during the early 1900s, proteins were considered better candidates as molecules able to transmit large amounts of hereditary information from generation to generation&idnum=10

This command uses port **4000** to send a request for the **10** best terms in the specified **querytext** to a DRE, which is located on a machine with the IP address **12.3.4.56**.

In response to the command, the DRE returns the highest weighing terms, for example:

```
Number of hits: 10
MENDEL+219
HEREDITARI+209
NUCLEU+194
1900+186
MOLECUL+175
DARWIN+169
TRANSMIT+157
PROTEIN+149
CHEMICAL+118
GENER+105
```

qmethod=v: displaying the DRE's status

http://<host>:<port>/qmethod=v

The **v** query method allows you to display the DRE status information in HTML format. The DRE status information includes the following:

- DRE version
- DRE build
- the ports the DRE uses
- the number of documents the DRE contains
- DRE database information
- Thread information
- Memory allocation
- the last 100 entries in the activity log

Command parameters:

Mandatory:	none
Optional:	none

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the query port by which commands are sent to the DRE.

Example:

http://12.3.4.56:4000/qmethod=v

This command uses port **4000** to request the status of the DRE that is located on a machine with the IP address **12.3.4.56**.

In response to the command, the DRE returns its status information, for example:

```
Autonomy DRE server version 3.1.13 Build 2.312

Query port: 8000, Index port: 8001

120821 documents
120821 document sections
120821 document slots
1535954 Terms
Process ID: 360
Directory: D:\staticdre

0 DateFix(s):

2 Database(s):
```

Name	Expiry Hours	Expiry Action	Max Hits	Date Range	Security	Documents	Document Sections
0:News	-	-	-	yes	-	43133	43133
1:Old	-	-	-	yes	-	77688	77688

```
Threads:
0 : Indexer idle
1 : Index Server Port 8001 (Waiting for connection)
2 : Merger disabled
3 : File Command Thread waiting
4 : Query Server idle.
5 : Query Server idle.
6 : Query Server idle.
7 : GET /qmethod=v HTTP/1.1

Memory Allocation:

Index Cache 0: 0Kb used. 0 Terms. 50000Kb allocated in 1 blocks
Term Cache: 50781Kb Used. 5903 Terms. 50781Kb allocated in 2 blocks
Term Cache Hit Average: 90%
Query Buffers: 14879Kb
Commit Buffer: 0Kb
Mapped Memory: 359811Kb

Reusable DOCS entries: 0

DYNTERM entries unused: 0

Last 100 entries of activity log:
16/08/2002 10:46:13 7 /qmethod=v
16/08/2002 10:46:13 7 Query connection granted to: 12.3.4.56
16/08/2002 10:43:52 6 Done
16/08/2002 10:43:52 6 /qmethod=H&querytext=query.html
16/08/2002 10:43:52 6 Query connection granted to: 12.3.4.56
```

```
16/08/2002 10:43:50 4 Done
...
16/08/2002 09:40:29 5 Query connection granted to: 12.3.4.56
16/08/2002 09:40:29 7 Done
16/08/2002 09:40:29 7 Terms.
```

qmethod=V: displaying the DRE's status and configuration file content

http://<host>:<port>/qmethod=V

The **V** query method allows you to display the DRE status information and the content of the DRE's configuration file in text format. The DRE status information includes the following:

- DRE version
- DRE build
- the ports the DRE uses
- the number of documents the DRE contains
- DRE database information
- Thread information
- Memory allocation
- the last 100 entries in the activity log

Command parameters:

Mandatory:	none
Optional:	none

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the query port by which commands are sent to the DRE.

Example:

http://12.3.4.56:4000/qmethod=V

This command uses port **4000** to request the status of the DRE that is located on a machine with the IP address **12.3.4.56**.

In response to the command, the DRE returns its status information, for example:

```
PRODUCT:Autonomy DRESERVER
VERSION:3.1.13
BUILD:2.312
LICENSED ON:Unlimited
LICENSED DAYS:Unlimited
REFHASHES:100000
REFDOCS:2
INDEXPORT:8001
QUERYPORT:8000
DIRECTORY: D:\staticdre
CHARCONV:0
STRIPLANGUAGE:0
REMOVESPACE:0
THREADS:64
MAXDBASE:1024
NUM_FIELDS:3
SIZE_OF_HEADER:10
TERMSPERDOC:40
SLOT_SIZE:10
FREE_SLOTS:4
MAX_DOC_PER_STRCTDAT:50000
ROOT_STRCTDAT:.
NEWQUERY:1
AUTOCOMPACT:0
COMPACT:1
PROPERNAMES:1
ECHO:1
LOGGING:1
URLHASH:1
FASTQUERY:1
SUGGESTTERMS:30
QUERYSUMMARY:0
DOCNUM:120821
MAXOC:146988
NUM4TERMS:1535954

NUM4TERMS_TOP:1546016
DTB_MEM_CUR:0
DTB_MEM_TOP:1
DOCUMENT_LIST_TOP:1428128

FREE_ENTRIES_TOP:500
FREE_ENTRIES_CUR:0
FREE_DOCS_TOP:1
FREE_DOCS_CUR:0
REUSED:0
DYNSYNC:0
MEMFIELDS:0
FIELDSEARCH:1
```

```
PROXIMITY:0
MAXPROX:6
MAXWORDCOUNT:3
MAXTITLEFIELD:10
TERMHASH:0
QC:0
QD:0
FIELDSORT:0
NUMSEARCH:0
READIN_SEP:
ACTIVEDBS:2
DATABASES:2[0:News,1:Old,]
DBINFO:2[0:43133(43133),1:77688(77688),]
staticdre.cfg:

[LICENSE]

HOLDER=Company
KEY=12345678901234567890123456789O


[SERVER]
QUERYPORT=8000
INDEXPORT=8001
ECHO=1

LOGGING=1
PROPERNAMES=1
COMBINE=1
HYPHENS=1
HYPHENCHARS=-
INDEXNUMBERS=2
SUGGESTTERMS=30
SUGGESTBIAS=0
DEFAULT_XOPTIONS=conceptsummary3+bias+withsummary
INDEXPRI=1
MINMATCH=70
ConceptSummaryQueryBias=5000
MAXLOGSIZE=10000000
```

```
[CUTCHART]
POINT0=0,0
POINT1=255,40
POINT2=700,30
POINT3=1500,0
POINT4=2000,0


[DATABASES]
NUMDBS=2
0=News
1=Old


[NEWS]
//EXPIRETIME=168


[FIELDS]
0=INDEX10.DRETITLE
1=300,DREREFERENCE
2=20,BLANK
3=Image=
4=64,Paper=
5=256,summary
6=16,DocType
7=
8=
9=
10=


[SCHEDULE]
COMPACT=20000
EXPIRE=1
TIME=0:0
INTERVAL=24:0


[INDEXCACHE]
MAXSIZE=50000


[TERMCACHE]
MAXSIZE=50000


DBSTAT:nTerms:-1, nTotalOcc:1, nMeanOcc:1, nWeightOcc:8
```

qmethod=p: displaying the DRE's index queue

http://<host>:<port>/qmethod=p

The **p** query method allows you to display the last 100 lines of the DRE's index queue.

Command parameters:

Mandatory: none

Optional: none

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the query port by which commands are sent to the DRE.

Example:

http://12.3.4.56:4000/qmethod=p

This command uses port **4000** to request the index queue of the DRE that is located on a machine with the IP address **12.3.4.56**.

In response to the command, the DRE returns the last 100 lines of the index queue, for example:

```
INDEX QUEUE:
END
RECENTLY PROCESSED COMMANDS (most recent first):
90031:/DRECOMPACT? - (Internally Scheduled Command ) (-1 Finished)
90030:/DREEXPIRE? - (Internally Scheduled Command ) (-1 Finished)
...
END
```

5. Restricting queries

Restricting queries to indexed fields

If you have indexed documents with index fields, you can specifically query these fields. This allows you to restrict queries to return only documents that have a specific value in the specified index field.

Examples:

qmethod=q&querytext=cat:DRETITLE

This query only returns documents that contain the term "**cat**" in the **DRETITLE** field.

qmethod=q&querytext=cat:DRETITLE+AND+dog:DRECONTENT

This query only returns documents that contain the term "cat" in the **DRETITLE** field and the term "dog" in the **DRECONTENT** field.

qmethod=q&querytext="tabby cat":DRETITLE+AND+dog:DRECONTENT

This query only returns documents that contain the term "tabby cat" in the **DRETITLE** field and the term "dog" in the **DRECONTENT** field.

Restricting queries to non-indexed fields

If you have indexed documents with fields, you can specifically query these fields. This allows you to restrict queries to return only documents that have a specific value in the specified field.

Examples:

qmethod=q&querytext=stars&fnameAUTHOR=*jones*

This query only returns documents that contain the term "stars". The documents must also contain the term "jones" in the **AUTHOR** field.

qmethod=q&querytext=*&fnameAUTHOR=*jones*&fnameKEYWORD=*galaxy*

This query only returns documents that contain the term "jones" in the **AUTHOR** field and the term "galaxy" in the **KEYWORD** field.

qmethod=q&querytext=stars&fnameAUTHOR=*jones*&fnameKEYWORD=*galaxy*&fieldbool=(fnameAUTHOR+XOR+fnameKEYWORD)

This query only returns documents that contain the term "stars". The documents must also contain the term "jones" in the **AUTHOR** field or the term "galaxy" in the KEYWORD field (documents that contain "jones" in the **AUTHOR** field and "galaxy" in the KEYWORD field are rejected).

qmethod=q&querytext=stars&fnameAUTHOR=*jones*&fnameCODE=NRANGE(1,99)&fieldbool=(fnameAUTHOR+AND+fnameCODE)

This query only returns documents that contain the term "stars". The documents must also contain the term "jones" in the **AUTHOR** field or a numeric value in the range **1** to **99** in the **CODE** field.

Field specifiers

When you query fields, you can use the following field specifiers to restrict which values the fields that you are querying can contain. This is useful, for example, to find fields that you have not indexed specifically (querying with field specifiers may slow down the querying process).

String

If you specify a string only documents are returned that contain fields whose value matches the specified string exactly.

For example:

fnameDRETITLE=newsreport

The **DRETITLE** field must contain the term "newsreport".

Wildcard

You can specify a wildcard string match.

For example:

fnameDRETITLE=*news*

The **DRETITLE** field must contain a term, which contains the string "news".

FUZZY(wrd)

Evaluates to **true** if the value that the specified field contains is similar to the specified term.

For example:

fnameDRETITLE=news

The **DRETITLE** field must contain a term, which is similar to the term "news".

ARANGE(char1-char2)

Evaluates to **true** if the field contains an alphabetical character in the range **char1** to **char2** (in ASCII order).

For example:

fnameCODE=ARANGE(a-z)

The **CODE** filed must contain a character between **a** and **z**.

NRANGE(num1,num2)

Evaluates to **true** if the field contains a numeric value in the range **num1** to **num2**.

For example:

fnameCODE=NRANGE(1,99)

The **CODE** filed must contain a number between **1** and **99**.

RANGE(date1-date2)

Evaluates to **true** if the field contains a date in the range **date1** to **date2**. The format of **date1** and **date2** is DD/MM/YYYY.

For example:

fnameDATE=RANGE(01/01/1990-01/01/2001)

The **RANGE** field must contain a date between **01/01/1990** and **01/01/2001**.

ANYC(str1+str2...)

Evaluates to **true** if any of the specified **strs** occur as a substring of the field.

For example:

fnameKEYWORDS=ANYC(go+rilla)

The **KEYWORDS** field must contain the substring "go" or "rilla".

ANYCAND(str1+str2...)

Evaluates to **true** if all of the **strs** occur as a substring of the field.

For example:

fnameKEYWORDS=ANYCAND(go+rilla)

The **KEYWORDS** field must contain the substrings "go" and "rilla".

ANY(str1+str2...)

Evaluates to **true** if any of the **strs** occur as a word in the field.

For example:

fnameKEYWORDS=ANY(gorilla+primates)

The **KEYWORDS** field must contain the word "gorilla" or "primates" or both words.

ANYAND(str1+str2...)

Returns **true** if all of the **strs** occur as a word in the field.

For example:

fnameKEYWORDS=ANYAND(gorilla+primates)

The **KEYWORDS** field must contain the words "gorilla" and "primates".

ANYF(str1+str2...)

Evaluates to **true** if any of the **strs** occur as a fuzzy substring match in the field.

For example:

fnameKEYWORDS=ANYF(gorilla+primates)

The **KEYWORDS** field must contain a substring that is similar to either "gorilla" or "primates".

ANYFF(str1+str2...)

Evaluates to **true** if any of the **strs** occur as a fuzzy word match in the field.

For example:

fnameKEYWORDS=ANYFF(gorrilla+primates)

The **KEYWORDS** field must contain a word that is similar to either "gorilla" or "primates".

OR2(string)

Evaluates to **true** if a field is a substring of string.

For example:

fnameKEYWORDS=OR2(category)

The **KEYWORDS** field must contain the substring **category**/.

OR(str1+str2...)

Evaluates to **true** if the field matches any of the strings exactly.

For example:

fnameKEYWORDS=OR(cat+cats)

The **KEYWORDS** field must contain the string "cat" or "cats".

BITANDOFF(nn,hexstr)

Evaluates to **true** if the hex value of field & hexstr is true after hexstr << nn.

For example:

fnamePRODUCT_CODE=BITANDOFF(2,0xFF)

The **PRODUCT** field must contain a hexadecimal string, which when bitshifted to the left **nn** times and bitwise ANDed with **hextstr** evaluates to **true**.

BITAND(hexstr)

Evaluates to **true** if hexadecimal value of field & hexstr is true.

For example:

fnamePRODUCT_CODE=BITAND(0xFFFF)

The **PRODUCT** field must contain a hexadecimal string, which when bitwise ANDed with **hextstr** evaluates to **true**.

BIAS(num)

Bias document score according to percentage value in field.

For example:

fnameKEYWORD=BIAS(20)

The weight of the KEYWORD field is increased by **20** percent.

GTNOW()

Evaluates to **true** if the field contains a number which is greater than the current number of seconds since 1970.

For example:

fnameTIME=GTNOW()

The **TIME** field must contain a number that is greater than the current number of seconds since 1970.

LTNOW()

Evaluates to **true** if the field contains a number which is less than the current number of seconds since 1970.

For example:

fnameTIME=LTNOW()

The **TIME** field must contain a number that is less than the current number of seconds since 1970.

Xoptions

You can add one or more of the following xoptions to your query string in order to determine how results are returned. If you want to specify multiple xoptions, you must separate them with plus symbols (there must be no space before or after a plus symbol).

absweight

Returns result scores as absolute values instead of percentages.

agent

Indicates that the DRE contains agents, and that weighting should be performed accordingly.

agentbitv

If you are querying an Agent DRE with an agent (that is the query text is an agent's training), **agentbitv** indicates to the DRE that the **agentbitv** field should be used to determine weighting.

allsections

If you are using qmethod=s, **allsections** suggests on all sections of a document, not just the first.

bias

Allows you to boost the percentage score of results for long queries. Long queries don't tend to return results with a high percentage score as it is unlikely that all the terms in the query's text are matched.

cache

Caches the results of the query. You can cache query results by default by setting the **Always** parameter in the configuration file's [QueryCache] section to **1**.

conceptsummaryN,LL

Returns a concept summary for the results that comprises **N** sentences and up to **LL** characters.

content

Returns the content of the result documents.

contentfields<field_list>

Queries only the content of specified fields. If you want to specify multiple fields, you must separate them with colons (there must be no space before or after a colon).

contextsummaryN,LL

Returns a context summary for each result document that comprises **N** sentences and up to **LL** characters..

delete

If you are using qmethod=G, **delete** allows you to delete the documents after they have been returned as results (provided the **IndexClients** parameter in the DRE configuration file's [Server] section lists your machine's IP address).

fuzzy

Indicates to the DRE that the query is a fuzzy query (you can use this if you are unsure about the query text's spelling).

inchitcount

Displays the total number of results that is available for the query at top of the result list.

nocombine

By default documents that have the same reference field are combined to ensure that only the most relevant one is returned. You can use **nocombine** to disable this.

nosort

Returns the result documents without sorting them by relevance first. This means that results are returned in order of descending document ID.

occ

If you are using qmethod=e, **occ** allows you to return the number of documents in which each term occurs in the DRE.

outfile=<file_name.txt>

Outputs results to the specified text file.

parse

Sends the query with **ProperNames** set to **1** or **2** (depending on what **ProperNames** is set to in the DRE's configuration file). If **ProperNames** is set to **0** in the DRE's configuration file, the query terms are sent with the Boolean operator WNEAR, so that if they appear close to each other in a result document, the document is given a higher weighting.

positions

Use **positions** if you are sending a fuzzy or wildcard query that uses the Boolean operator NEAR, WNEAR or DNEAR, or if you are sending a phrase query to the DRE.

queryterms

Uses the querytext when doing a conceptsummary.

reverse

Reverses the order of characters in the transliteration of oriental languages.

sortboth

Sorts results by relevance. If multiple results have the same relevance they are then sorted by date.

sortbydate

Sorts results by date.

sortdate

Sorts results by date.

startN

Returns results from the **N**th result onwards.

strip

If you are using qmethod=a, **strip** allows you to stem the query terms.

synonym

Adds the synonyms of the query's terms to the query text.

trueocc

If you are using qmethod=e, **trueocc** allows you to return the total number of times each term occurs in the DRE.

useurl

Indicates to the DRE that the querytext is a reference, not a docid.

withcount

Returns the number of indexed terms for each result document.

withdate

Returns the DREDATE field for each result document.

withsummary

Returns a quick summary for each result document.

words=N

The maximum **N** number of terms that the summary that is displayed for each result document can contain.

Boolean queries

When you are using the following qmethod command, you can specify the query text using a Boolean expression:

http:// <IPAddress>:<QueryPort>/qmethod=q&querytext=<QueryText>

You can use the following operators to manipulate a query by applying them to words, exact phrases or other Boolean expressions. Note that APCM will still be used to rank the results that match the Boolean query.

AND Binary operator. Ensures that both terms are matched in every document that is returned.

For example:

qmethod=q&querytext=cat+AND+dog

This query only returns documents that contain both “cat” and “dog”.

NOT Unary operator. Ensures that the term following NOT is excluded from any of the returned documents.

For example:

qmethod=q&querytext=cat+NOT+dog

This query only returns documents that contain "cat" but not "dog".

OR Binary operator. One or both terms must appear for the document to be returned. This is the default behavior if no explicit operator is given between two terms.

For example:

qmethod=q&querytext=cat+OR+dog

This query only returns documents that contain either “cat”, “dog” or both terms.

EOR

or

XOR

Binary operator. Logical exclusive OR. Only one of the terms is permitted to appear for the document to be returned. This is a rarely used operator.

For example:

qmethod=q&querytext=cat+XOR+dog

This query only returns documents that contain either the term “cat” or the term “doc”. Documents that contain "cat" and "dog" are not returned.

NEAR*nn* Only returns documents in which the second term is within ***nn*** words from the first term.

For example:

qmethod=q&querytext=cat+NEAR1+dog

This query only returns documents in which the term “cat” is no more than 1 words away from “dog”. This means that documents, which contain "cats and dogs" and documents that contain "dogs and cats" are returned, while documents that contain "cats do not like dogs" are not returned (as the terms are not close enough to each other).

DNEAR*nn* Directed NEAR. Only returns documents in which the second term is within ***nn*** words from the first term, in the specified order.

For example:

qmethod=q&querytext=cat+DNEAR1+dog

This query only returns documents in which the term "dog" follows the term "cat", but no more than one word away from the term "cat". This means that documents, which contain "cats and dogs" are returned, while documents that contain "dogs and cats" or "cats do not like dogs" are not returned.

WNEAR*nn* Weighted NEAR. Proximity operator that promotes relevance when term spacing is less than the specified word distance (closer together implies higher relevance).

For example:

qmethod=q&querytext=dog+WNEAR6+cat

In this query extra relevance is given to documents in which "cat" and "dog" appear within 6 words of each other in a piece of text. This weight increases as the terms get closer to each other.

BEFORE Only returns documents in which the first term precedes the second one.

For example:

qmethod=q&querytext=cat+BEFORE+dog

This query only returns documents in which the term "dog" appears later than the term "cat".

AFTER Only returns documents in which the first term appears later than the second one.

For example:

qmethod=q&querytext=cat+AFTER+dog

This query only returns documents in which the term "cat" appears later than the term "dog".

( ) Bracketed expressions. These are evaluated left to right and can be nested. They dictate the precedence and behavior of combined operator statements.

For example:

qmethod=q&querytext=(fish EOR pie) AND (chips EOR mash)

This query only returns documents that contain one of the following:

"fish" and "chips"

"fish" and "mash"

"pie" and "chips"

"pie" and "mash"

Precedence of Boolean operators

Boolean operators have the following precedence:

Highest precedence:	NOT
Lower precedence:	AND; WNEAR; NEAR
Lowest precedence:	OR; XOR; EOR

Operators that have the same level of precedence have neither left or right associativity. You can use brackets to bind terms together as appropriate.

Soundex keyword search

If you are querying for a particular term, you can use Soundex to return documents that contain the same term or a phonetically similar term (using the Soundex algorithm). You can combine a Soundex keyword search with any other query form.

For example:

qmethod=q&querytext=albert+SOUNDEX(einstine)

Exact phrases

You can search for exact phrases by putting quotation marks (" ") around a string of words. For example: “world market”

Relevance ranking

In evaluating all types of queries, the DRE employs complex algorithms based on a combination of Information Theory and Bayesian methods to weight and rank the document returns by statistical relevance. In doing so it makes use of information theoretic values calculated dynamically for all concepts on indexing, allowing relevance to be evaluated both as a percentage, and in the case of agents, such as those created by the Categorizer, as absolute values.

In practice, the relevance can be seen as a measure of the conceptual overlap between the query text and the text within a document. This can be affected in several ways; certain fields can be given extra weight by associating a weighting factor with them at indexing time. For example, extra weight can be given when query terms appear in a document's title as opposed to the body of the text.

Alternatively, relevance can be increased through use of the "WNEAR" operator; a query of "cat+WNEAR10+dog" gives extra weight to documents in which the terms "cat" and "dog" are mentioned near to one another within the text.

6. Administering the DRE

You can administer the DRE using:

- **Index port commands**

- **the DRE Administration dialog**

  If you are running on a Windows platform, you can use the DRE Administration dialog to administer the DRE. Please refer to **Appendix A** for further details on the DRE Administration dialog.

Index port commands

You can send index port commands to your DRE using your web browser or by writing your own script (note that you cannot send some commands via your web browser as they require POST request methods). Alternatively, you can use the DRE API (for example C, Java, VB, JNI and so on) in order to write code that allows you to administer your DRE.

You can use index port commands to:

- reload the DRE's configuration file
- delete documents by URL reference from the DRE
- delete individual documents and ranges of documents from the DRE
- create new DRE databases
- delete all documents in a database
- expire documents
- compact the DRE
- change field values in DRE documents
- back up the DRE
- initialize the DRE

DRERESET?: reloading the DRE's configuration file

http://<host>:<port>/DRERESET?

The **DRERESET?** command allows you reload the DRE's configuration file.

Command parameters:

Mandatory: none

Optional: none

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the number of the port that is used to index files into to the DRE.

Example:

http://12.3.4.56:4001/DRERESET?

This command uses port **4001** to reset the DRE that is located on a machine with the IP address **12.3.4.56**.

DREDELETEREF?: deleting documents by reference from the DRE

http://<host>:<port>/DREDELETEREF?<mandatory_parameter>&<optional_parameter>

The **DREDELETEREF?** command allows you delete documents from the DRE by their reference.

Command parameters:

Mandatory:	**docs**=<document_references>
Optional:	**dbname**=<database_name>

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the number of the port that is used to index files into to the DRE.

<document_references>

Enter the URL of the document that you want to delete. If you want to specify multiple references, you must separate them with plus symbols (there must be no space before or after a plus symbol).

<optional_parameters>

You can enter one of the following parameters:

dbname=<database_name>

Allows you to specify the name of the database, which contains the documents that you want to delete. If you don't specify a database and the specified document is contained in several databases, it is deleted from all of them.

Example:

http://12.3.4.56:4001/DREDELETEREF?docs=http%3A%2F%2Fnews%2Enewssite%2Ecom%2Findex%2Ehtml+http%3A%2F%2Fnews%2Enewssite%2Ecom%2Fcoverstory%2Ehtml&dbnum=3

This command uses port **4001** to delete the documents with the specified URLs from the DRE that is located on a machine with the IP address **12.3.4.56**.

DREDELETEDOC?: deleting individual documents and ranges of documents

http://<host>:<port>/DREDELETEDOC?<mandatory_parameter>

The **DREDELETEDOC?** command allows you delete individual documents and / or ranges of documents from the DRE by their DOCID.

Command parameters:

Mandatory:	**docs**=<document_references> and / or **range**=[<doc_ref>,<doc_ref>] Note that if you want to specify docs and range you must separate them with a plus symbol.
Optional:	none

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the number of the port that is used to index files into to the DRE.

<document_references>

Enter the DOCID of the document that you want to delete. If you want to specify multiple references, you must separate them with plus symbols (there must be no space before or after a plus symbol).

<doc_ref>,<doc_ref>

Enter the DOCID of the first and last document in the range of documents that you want to delete.

Example:

http://12.3.4.56:4001/DREDELETEDOC?docs=3+5+range=[7-10]

This command uses port **4001** to delete the documents with the DOCID **3**, **5**, **7**,**8**,**9** and **10** from the DRE that is located on a machine with the IP address **12.3.4.56**.

DRECREATEDBASE?: creating a new database in the DRE

http://<host>:<port>/DRECREATEDBASE?<mandatory_parameter>

The **DRECREATEDBASE?** command allows you to create a new database in the DRE.

Command parameters:

Mandatory:	**<database_name>**
Optional:	none

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the number of the port that is used to index files into to the DRE.

<database_name>

Enter the name of the database that you want o create in the DRE.

Example:

http://12.3.4.56:4001/DRECREATEDBASE?Archive

This command uses port **4001** to create a new **Archive** database in the DRE that is located on a machine with the IP address **12.3.4.56**.

DREDELDBASE?: deleting all documents in a database

http://<host>:<port>/DREDELDBASE?<mandatory_parameter>

The **DREDELDBASE?** command allows you to delete all documents in a database.

Command parameters:

Mandatory:	**dbname**=<database_name>
Optional:	none

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the number of the port that is used to index files into to the DRE.

<database_name>

Enter the name of the database from which you want to delete all documents.

Example:

http://12.3.4.56:4001/DREDELDBASE?Archive

This command uses port **4001** to delete all **Archive** database documents from the DRE that is located on a machine with the IP address **12.3.4.56**.

DREEXPIRE: expiring documents

http://<host>:<port>/DREEXPIRE

The **DREEXPIRE** command forces the Expiry operation, which checks all databases in the DRE for documents that have been stored longer than the specified expire time of the respective database. Any documents that have expired are either moved or deleted (depending on the DRE's configuration file settings).

Command parameters:

Mandatory: none

Optional: none

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the number of the port that is used to index files into to the DRE.

Example:

http://12.3.4.56:4001/DREEXPIRE

This command uses port **4001** to expire the documents from the DRE that is located on a machine with the IP address **12.3.4.56**.

DRECOMPACT: compacting the DRE

http://<host>:<port>/DRECOMPACT

The **DRECOMPACT** command forces the Compact operation, which compresses the space that documents take by filling up the space that has been created through the deletion of documents by filling it up with new documents (similar to the defragmentation process).

Command parameters:

Mandatory: none

Optional: none

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the number of the port that is used to index files into to the DRE.

Example:

http://12.3.4.56:4001/DRECOMPACT

This command uses port **4001** to compact the DRE that is located on a machine with the IP address **12.3.4.56**.

DREREPLACE: changing field values in DRE documents

DREREPLACE?<data>#DREENDDATA

Note: This command requires a POST request method

The **DREREPLACE … #DREENDDATA** command allows you to change field values in DRE documents.

Command parameters:

Mandatory:	<data>
Optional:	none

<data>

The fields that you want to replace in the DRE. You need to specify each field as follows:

#DREDOCID ***n*** or #DREDOCREF ***n***

#DREFIELDNAME ***x***

#DREFIELDNAME ***y***

n

The DocID or reference (URL) of the document that contains the field, which you want to replace.

x

The name of the field whose value you want to change.

y

The value that you want field x to change to.

For example:

```
#DREDOCID 1
#DREFIELDNAME Price
#DREFIELDVALUE 10
#DREDOCREF http://www.autonomy.com/autonomy/dynamic/autopage442.shtml
#DREFIELDNAME Country
#DREFIELDVALUE UK
#DREENDDATA
```

In this example, the value of the **Price** field in the document with the DocID **1** is changed to **10**. The value of the **Country** field in the document with the reference **http://www.autonomy.com/autonomy/dynamic/autopage442.shtml** is changed to **UK**.

DREBACKUP?: backing up the DRE

http://<host>:<port>/DREBACKUP?<mandatory_parameter>

The **DREBACKUP?** command copies all the DRE's *.DB files to an new location. This is useful to create a safe copy of the DRE. You can then use the DREINITIAL? Command to reset the DRE to the state it was in when it was backed up.

Command parameters:

Mandatory:	<path>
Optional:	none

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the number of the port that is used to index files into to the DRE.

<path>

Enter the path to the location where you want to create the DRE backup.

Example:

http://12.3.4.56:4001/DREBACKUP?E:\DynamicReasoningEngine

This command uses port **4001** to create a backup of the DRE that is located on a machine with the IP address **12.3.4.56** on **E:\DynamicReasoningEngine**.

DREINITIAL?: initializing the DRE

http://<host>:<port>/DREINITIAL?<optional_parameter>

The **DREINITIAL?** command allows you to reset the DRE to its original state or to a particular backup state.

Command parameters:

Mandatory: none

Optional: <path>

<host>

Enter the IP address (or name) of the machine on which the DRE is installed.

<port>

Enter the number of the port that is used to index files into to the DRE.

<optional_parameters>

You can enter one of the following parameters:

<path>

Allows you to specify the path to the DRE backup that you want to reset the DRE to.

Example:

http://12.3.4.56:4001/DREINITIAL?E:\DynamicReasoningEngine

This command uses port **4001** to reset the DRE that is located on a machine with the IP address **12.3.4.56** to the state it was in when it was backed up on **E:\DynamicReasoningEngine**.

Appendix A: the DRE Administration dialog

The GUI Admin is a user-friendly way of administering a DRE. It allows you to alter which DRE you are looking at, index documents into the DRE, remove documents from the DRE, query the Databases in the DRE, add new databases and delete documents from databases.

Starting up the GUI Admin

To start the GUI Admin, you can execute the program through the start menu or double click on the <InstallationName>_ContentDREadmin.exe file located in the **dre** subdirectory. Once you have started up the GUI Admin, it will automatically connect to the DRE with which it was installed and the main window will be displayed.

Using the GUI Admin

GUI ITEM	DESCRIPTION
DRE Settings	Displays the DRE Details.
DRE Status	Displays the statistics of the DRE including version and contents of the DRE.
Database List	Displays: • the Databases contained in the DRE • number of hours that documents will remain in the database after indexing • name of the database that expired documents will be moved into
"Change DRE Settings" Button	Sets which DRE you are administering
"View Details" Button	Allows you to view the last few commands sent to the DRE, both on the Query Port and Index Port.
"View DRE.CFG" Button (Under "View Details" Button	Allows you to view the DRE.CFG file of the DRE.
"Create New Database" Button	Allows you to create a new database in the DRE.
"Delete documents from Database" Button	Empties a selected Database of all its contents.

If you double-click on the databases you can see the statistics for each of the databases:

You cannot remove databases using the GUI Admin. To do this you must manually remove the database from the DRE configuration file. Also, it is not possible to set databases to expire from here. You will need to edit the DRE configuration file.

Changing DRE settings

If you want to change the DRE settings click on the “Change DRE Settings” button and edit the settings as necessary. From here you can also specify the name of the service corresponding with the DRE executable and start and stop the service using the equivalent buttons.

To change these settings the DRE must be local. The changes can be made if an NT Server is used and the full path is given to the DRE configuration file as an argument.

Importing files into an .IDX file

It is only possible to index .IDX files into the DRE. If you want to index any other type of file then you have to import it first. If you are importing files other than in HTML or PDF format you must have the Omnislave .DLL files together with the Omnislave.exe executable file installed in your working directory. For .PDF files you must have the pdfslave.exe executable file installed in your working directory. HTML and Text files require no extra files for importing.

To import files click on the “Import” button in the main GUI Admin window. Add the files to the “Input files to Import” list by clicking on the “Add” button. Alternatively you can drag and drop files onto the “Input Files to Import” list. Select the database in which the files are to be indexed into and once all of the criteria have been entered, click on the “Next” button.

GUI ITEM	DESCRIPTION
Input Files to Import	Lists the files to import into an IDX file
“Add Files” Button	Use to add files to the list.
“Add Folders” Button	Use to add directories to the list. Tick on the “Recurse Folder” box if you want to recurse subdirectories.
“Remove File” Button	Use to remove files/directories from the list.
Output Import File	.IDX filename into which list of files will be imported.
DRE Database	The database into which the IDX file will be indexed in the DRE.
“Set Import Parameters” Button	Launches the Import wizard enabling you to specify various criteria according to which the documents should be imported into the idx format.

Using the Import Wizard

The Import Wizard is used to specify the idx format corresponding to each document. It consists of multiple tabs used for different settings which control how the file will be generated during importing. These sets of options are:

Input

On the left of the input page, you can select the extensions of files to be treated as text and on the right you can select the extensions of files to be treated as HTML.

Content

The content can be stored in three different ways. All of these are possible at once. These are:

- the plain text content
- a summary
- intelligent title and summary

The number of sentences for a summary can be specified.

This property sheet also allows you to specify the criteria that the document must meet for it to be imported. You can specify the minimum and maximum size of the document to be imported in both words and bytes.

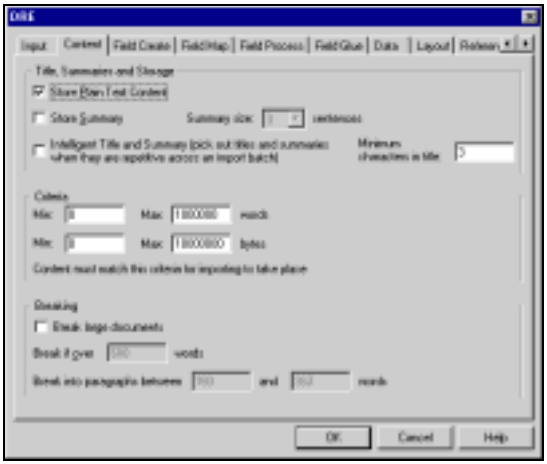
If you would like large documents to be broken up into a number of paragraphs, then you must check the breaking checkbox on this page. You can then specify the size that a document must be for it to be broken up and the size of the paragraphs produced.

Field Create	Specifies the fields that are to be stored in the DRE. These are both fixed fields and dynamic fields. For further details, please see the various field sections below.
Field Map	Specifies the mapping of fields within the DRE. See below
Field Process	This enables processes to be specified for a field set up in the DRE with parameters attached to that process. For example extract part of the field starting at a certain character.
Field Glue	This specifies fields that can be glued together or a field that can have a string glued to it.
Data	Specifies certain file data (date and length) that are to be extracted from the documents and stored in fields in the DRE. The details of the way in which data is to be extracted is specified here. These details include where to extract data from, where to extract it to and in what format. These details are explained further in the data section below.
Layout	Specifies what content in the documents is to be stored. You must define the beginning and end of documents. This page includes HTML start and end definitions and text start and end definitions. The HTML start and end definitions are used before the content has been converted into plain text and the text start and end definitions are used after wards. You must also specify the position of page breaks on this sheet.
Reference	Specifies reference replacements for documents imported. This includes the part of the path to replace and what to replace it with. It is also on this property sheet that you must specify how to truncate a path.
Rendering	Specifies how to treat the temporary HTML documents generated when processing omnislave/pdf imported documents.
Parsing	Specifies data that should be stripped from the imported document.
Delimited	Specifies how to deal with delimited files. You must specify the files to be treated as delimited by listing the extensions of those files. You must also specify document start and end delimiters and field start and end delimiters.
XML	Specifies how to deal with XML files. You must list the extensions of files to be treated as XML in the box provided.
Custom Slaves	Specifies extra custom made slaves that are needed to import certain file types. You must give your newly created slave a name and enter the file type that it is to import. The type will then be associated with that slave. You can have more than one associated file type for each slave but you must make a separate entry for each file type i.e. you will have to enter your new slave more than once too.
Advanced	Specifies miscellaneous details that can be applied in the importing process. These details are explained in the Advanced section below.

Input

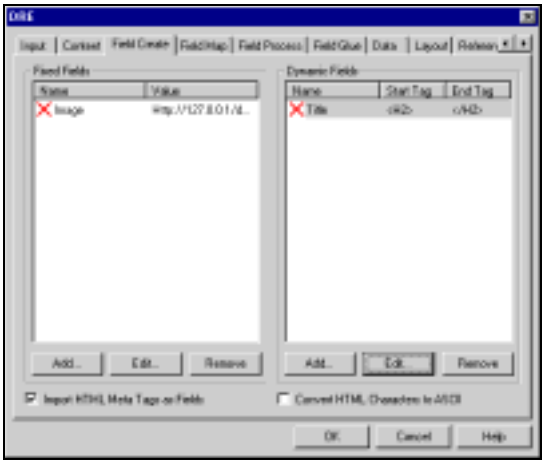
Parameter	Description
File types to treat as text	File types that will be imported as though they are text files. You can add and remove extensions from the list using the corresponding buttons.
File types to treat as HTML	File types that will be imported as though they are HTML files. You can ad and remove extensions from the list using the corresponding buttons.
Default Extension	Specifies the extension by which all unrecognized file types are to be masked as.
Directories	Specifies whether or not to import documents that appear in a subdirectory of the working directory.

Content



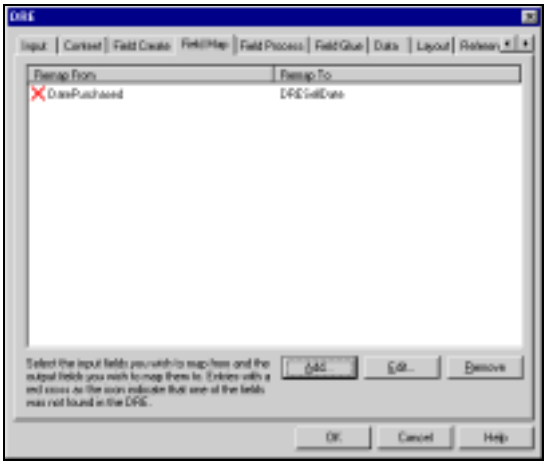
Parameter	Description
Store Plain Text Content	Specifies whether or not the text content of the documents are to be imported
Store Summary	Specifies whether or not the first n lines of the documents are to be imported as a summary.
Intelligent Title and Summary	Specifies whether or not a unique intelligent title and summary are to be generated for the documents that are imported.
Minimum Characters in Title	This specifies the minimum amount of characters that can be used as a title. If a title of less than 3 characters is found the title will be taken from the content of the document.
Criteria – Min/Max Words	Indicates that documents with less than Min words and more than Max words are not to be imported.
Criteria – Min/Max Bytes	Indicates that documents with less than Min Bytes and more than Max Bytes are not to be imported.
Breaking – Break large documents	Specifies whether or not documents over a certain size should be broken up into smaller sections for easier indexing.
Breaking – Break if over	If documents are over this number of words, then the document is broken up into smaller sections for easier indexing.
Breaking – Break into paragraphs between	When documents are broken up into smaller sections, the sections have at least n paragraphs and at most m paragraphs.

Field Create



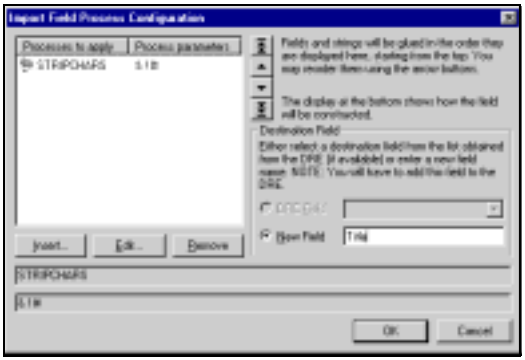
Parameter	Description
Fixed Fields – Name	Name of the field in the DRE that the value in the document is to be stored in.
Fixed Fields – Value	Value that is to be stored in the DRE field. This value will be the same for all documents imported in this session.
Dynamic Fields – Name	Name of the field in the DRE that the value in the document is to be stored in.
Dynamic Fields – Start Tag	Set of characters that are to be used to denote the start of the value that is to be stored in the DRE field. This value can potentially be different for each document imported in this session.
Dynamic Fields – End Tag	Set of characters that are to be used to denote the end of the value that is to be stored in the DRE field. This value can potentially be different for each document imported in this session.
Import HTML Meta Tags as Fields	Specifies whether or not the values of HTML Meta Tags are to be stored in DRE fields of the equivalent name. These fields must be specified in the DRE.CFG before the documents are indexed.
Convert HTML Characters to ASCII	Specifies whether or not to convert HTML characters to ASCII. For example, & would be converted to &.

Field Map



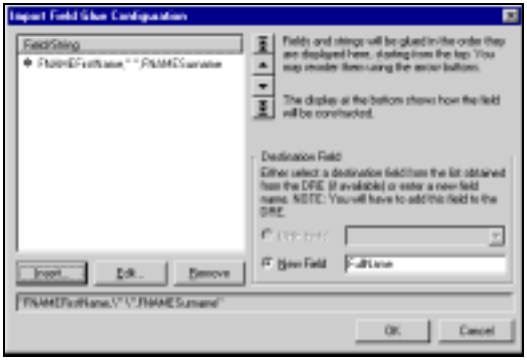
Parameter	Description
Field Remapping – Remap From	Specifies the name of a field whose value is to be taken and inserted as a value for an alternative field.
Field Remapping – Remap To	Specifies the name of a field whose value is to be the same as a value in an alternative field.

Field Process



Parameter	Description
Process to Apply	This specifies the name of the operation to apply to the specified field See import parameters for list of available entries here.
Process Parameters	This specifies the parameters to go with the operation/process being applied to the specified field See import parameters for list of available entries here.
New Field	This is the Field to apply the operations to.

Field Glue

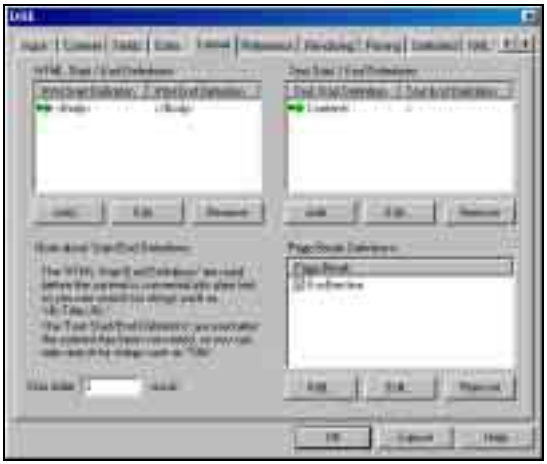


Parameter	Description
Destination Field (New Field)	This is the destination field to glue the source CSVs into. See the Import Parameters for more details.
Source CSVs	This allows fields and or strings to be glued together. Specify FNAME<FieldName> and / or a string. See Import Parameters for more details.

Data

Parameter	Description
Date – Extract From	Indicates from where in the File data a date is to be extracted. This can be either: **Now** Current date **Accessed** Date it was last accessed **Created** Date it was created **Modified** Date it was last modified **Field** Date taken from a specific file field **Content** First date found in the content of the document **Filename** First date found in the file name
Date – Format	Specifies the formats in which the required date can be found.
Date – Extract from field	Specifies the name of a date field in the DRE whose value is to be extracted.
Date – Extract to field	Specifies the name of the DRE field where the date extracted is to be stored.
Date – Extract to format	Specifies the format that the extracted date will be stored as in the DRE. If you are extracting to the field DREDATE, then the format must be yyyy/mm/dd or number of seconds since 1970 or NOW
Length – Extract from	Specifies from where the length of the document is to be extracted. Currently this can only be “File”.
Length – Extract to field	Specifies the name of the field in the DRE where the length of the document is to be stored.

Layout

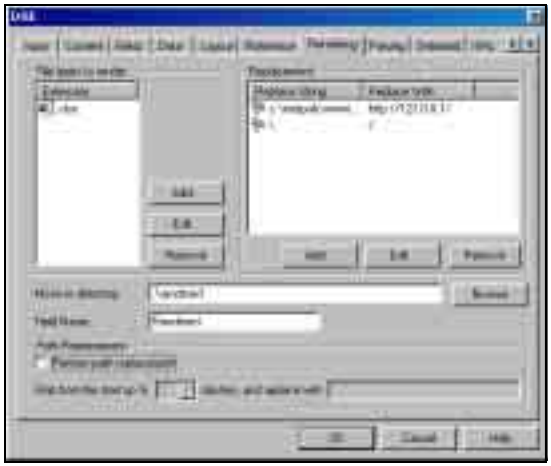


Parameter	Description
HTML Start/End Definitions – HTML Start Definition	A string of characters that mark the start of the content to be imported in HTML documents.
HTML Start/End Definitions – HTML End Definition	A string of characters that mark the end of the content to be imported in HTML documents.
Text Start/End Definitions – Text Start Definition	A string of characters that mark the start of the content to be imported in text documents.
Text Start/End Definitions – Text End Definition	A string of characters that mark the end of the content to be imported in text documents.
Page Break Definitions	Specifies a string that is used to mark a document break. This is used for splitting up documents into segments with each segment being contained in one idx format. When the idx files have been indexed into the DRE, the individual segments are joined back together to produce the original document.
Skip initial ***n*** words	Specifies that the first n words in a document are to be skipped when importing the document content.

Reference

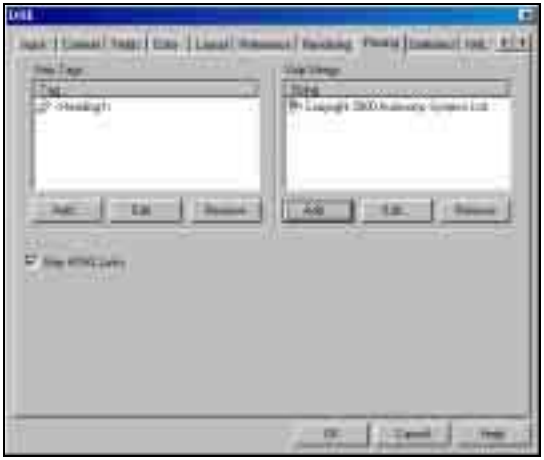
Parameter	Description
Replace up to – Perform Path Replacement	Specifies whether or not the string up to a certain \ in the file reference is to be replaced.
Replace up to – Strip from the start up to…	Specifies the nth slash where the preceding string is replaced with the string specified.
Replace up to - …and replace with	Specifies a string that is to replace the string preceding the nth slash in a document reference.
Truncation – Perform path truncation	Specifies whether or not to truncate the file reference for imported documents.
Truncation – truncate from copy	Specifies that the reference must be truncated after the nth occurrence of the string specified.
Truncation - …of…	Specifies the string that is used to truncate the reference after the nth occurrence.
Replacement CSVs – Replace string	Specifies the string that is to be replaced.
Replacement CSVs – Replace with	Specifies the string that is to replace the original reference of the file.

Rendering



Parameter	Description
File types to render – Extension	Specifies the extension of files on which to perform HTML rendering
Replacement – Replace string	Specifies the string that is to be replaced. Used to set the right reference for the Field Name field.
Replacement – Replace with	Specifies the string that is to replace the original reference of the ImportRenderedHtmlFieldName field. Used to set the right reference for the Field Name field.
Move to directory	For any non-HTML file that gets imported, the HTML file that comes out of omnislave/PDF gets moved into the specified directory.
Field Name	For any non HTML file that gets imported, the HTML file that comes out of omnislave/PDF gets moved into the this DRE field.
Path Replacement – Strip from the start up to …	Specifies a string that is to be replaced up to a certain '\'. Used to set the right reference for the Field Name field.
Path Replacement – …replace with…	Specifies the replacement string of the string immediately preceding the nth '\'.

Parsing



Parameter	Description
Strip Tags	Strips the content between begin and end tags. E.g. ImportStripTagCSVs=<B>, will strip all content between <B> and <\B>.
Strip Strings	Comma separated list of strings that, when matched in a document, are to be stripped from the idx format.
Strip HTML Links	Specifies whether or not to remove all content that is a hypertext link.

Delimited

Parameter	Description
File types to treat as delimited	Specifies the extension of the files to treat as delimited files.
Document Start Delimiter	In the case of importing a delimited file, e.g. a .CSV file, this specifies the beginning of a file.
Document End Delimiter	In the case of importing a delimited file, e.g. a .CSV file, this specifies the end of a file.
Field Start/End Delimiters – Field Name	Specifies the name of the field where data from a delimited file is stored.
Field Start/End Delimiters – Start Delimiter	Specifies the start string of data to be inserted into the idx field denoted by Field Name.
Field Start/End Delimiters – End Delimiter	Specifies the end string of data to be inserted into the idx field denoted by Field Name.
Field Start/End Delimiters – Skip Character	Specifies the number of characters to skip from the beginning of a document when producing the idx file.

XML

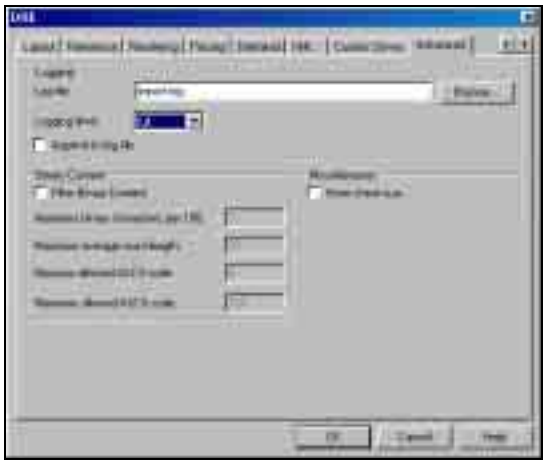
Parameter	Description
File types to treat as XML	A list of all XML file extensions.
XML Parameters – Field Name	Will specify the field in the DRE where the content taken from the XML File is stored.
XML Parameters – Tags	Searches for the appropriate XML tag(s). This can be a hierarchy level of search tags. For example, ImportSMLSearchTags0=body,head,title will take the content between the tags: <body></body>, <head></head> and <title></title>.
XML Parameters – Strip Tags	If set to yes, will strip the strings between the specified XML tags upon import of the file.

Parameter	Description
XML Parameters – Entry Only	Only imports the content between the searched tags at that level. For example if an XML file contained the following strings: <TAG1> <TAG3> Some Other Tag3 entry </TAG3> <TAG2> Some data <TAG3> The entry for Tag3 </TAG3> </TAG2> </TAG1> and ImportXMLSearchCSVTags0=TAG1,TAG2 ImportXMLEntryOnly0=TRUE, then the content imported = “Some data” If ImportXMLEntryOnly0=FALSE, then the content imported = “Some data The entry for Tag3”

Custom Slaves

Parameter	Description
Slave Exe Name	Name of the custom slave executable.
Extension	Extension of files that are to be processed by the custom slave executable.

Advanced



Parameter	Description
Log File	Name to be given to the import log file
Logging Level	This is the type of logging that is to be performed. This can be either: • Warnings • Errors • Full
Append to log	Specifies whether or not import logging for this session should be appended to an existing log file.
Binary Content – Filter Binary Content	Specifies whether or not binary content is to be processed.
Binary Content – Maximum binary characters per 100	Maximum number of binary characters tolerated per 100 characters. Anything binary exceeding this is to be ignored.
Binary Content – Maximum Average Word length	Binary words with more than this number of characters will not be imported.

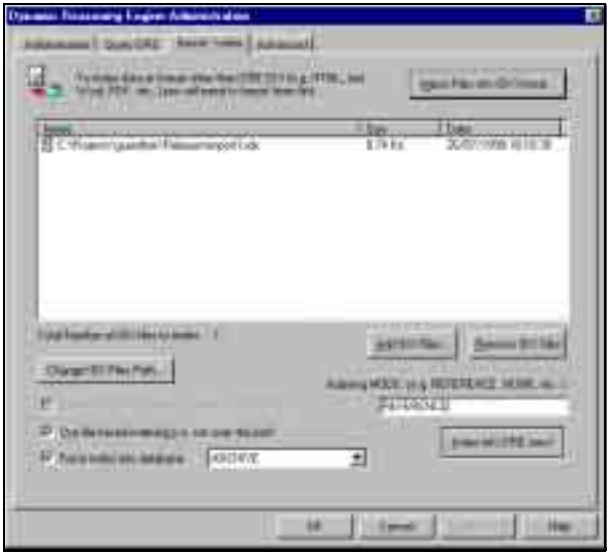
Parameter	Description
Binary Content – Minimum allowed ASCII code	The minimum ASCII value allowed in characters in the binary content. Anything lower is treated as binary and removed.
Binary Content – Maximum allowed ASCII code	The maximum ASCII value allowed in characters in the binary content. Anything higher is treated as binary and removed.
Store Checksum	When checked, a value is added to the Checksum field in the [Field] section in the DRE.CFG. This field value is used to determine whether or not to show a document result in the front-end. This field is typically used with Autonomy Update.

Once you have added all your import parameters you can then click on the “OK” button, which will take you back to the Main Import Property Sheet of the DRE GUI Admin. From here you can select the “OK” button and the idx file will be created. This file will be displayed in the idx list of the Import – Index Property Sheet of the DRE GUI Admin. Once you have added all idx files that you need to index into your DRE, click the “Index into DRE Now” button and the process will be executed with the listed idx files.

Indexing IDX Files Into The DRE

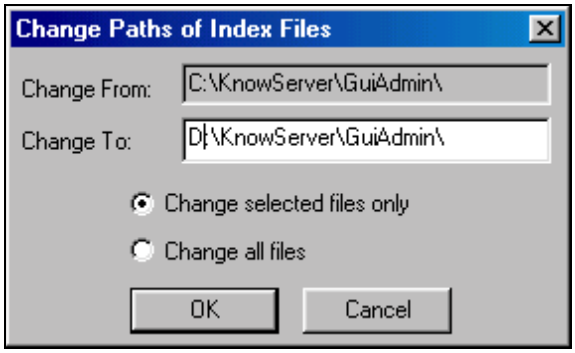
When indexing .IDX files into the DRE, you can only force the .IDX file to be indexed into the specific database when the “Use file based indexing” checkbox has been marked. If you do not force indexing into a specific database, the file will be indexed into the database specified when the IDX file was created. If your specified IDX file is not already included in the “Files To Index” list, add the IDX files to the list by clicking on the “Add” button and browsing for the required file. Alternatively you can drag and drop the .idx files onto the list. At this point if you wish to view and/or edit a particular .IDX file, double clicking on the listed file path will launch MS Write and display the file.

If you are indexing a large file over the socket, the GUI Admin will ask you if you are sure that this is what you want to do.



Parameter	Description
Delete Files after indexing	This can only be executed if indexing over the port. It will automatically delete all idx files after indexing has taken place.
Use file based indexing	This can only be used if the DRE GUI Admin and the indexing DRE resides on the same machine. The idx file is not sent over the port.
Force index into database	This can only be executed if using file based indexing. The documents to be indexed in the DRE are forced into the specified database as opposed to the database specified in the idx file.
Indexing MODE	Specifies how documents are to be replaced in the DRE.

Select the extra features that you wish to be performed during/after the indexing process by marking the appropriate check boxes. If you are doing a file based index, you will need to change the Index File Path so that the path is with respect to the DRE into which it is being indexed. To do this, click on the “Change Index Files Path.” button and alter the path appropriately.

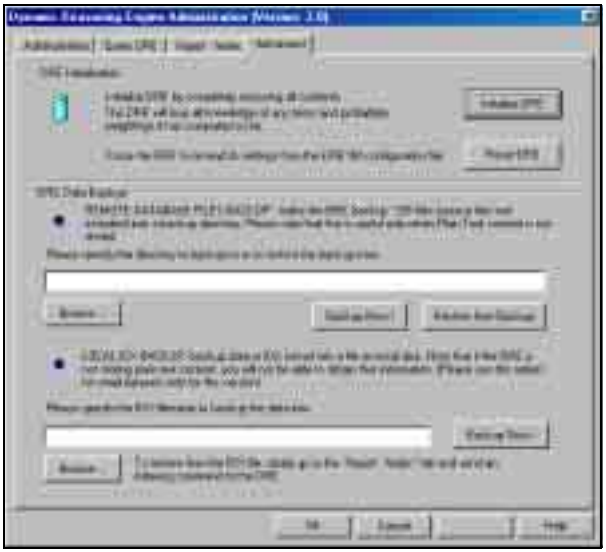


If you select “Change All Files” then the paths of all files listed will be changed otherwise just the selected path will be altered.

Once you have added all the IDX files that you wish to index and have given the required criteria, click on the “Index to DRE Now!” button. The file will then be added to the DRE.

Advanced Settings

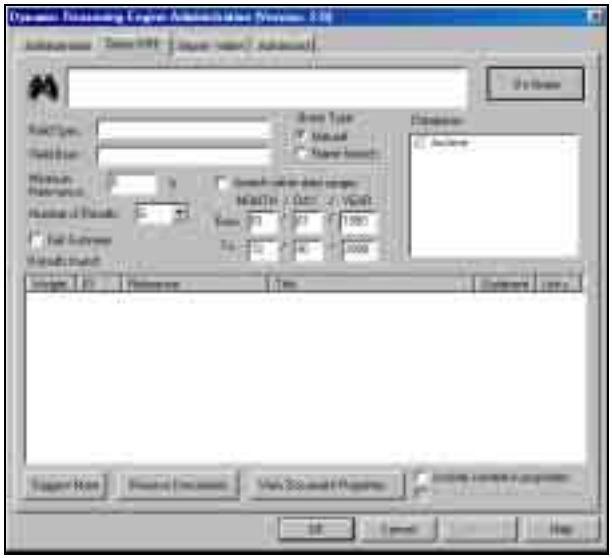
This screen contains advanced settings for administering the DRE, and backing up the data within the DRE.



GUI ITEM	DESCRIPTION
DRE Initialization – Initialize DRE	Initializes the DRE, which deletes all documents as well as all terms and weights in the Engine.
DRE Initialization – Reset DRE	Forces the DRE to re-read its settings from the DRE.CFG file.
DRE Data Backup	• Remote database backup: sends a command to the DRE for it to save all its database files into a directory (remember the path must be with respect to the DRE). You can restore from that database as well. • Local IDX backup: it sends a command to the DRE so that it saves all its data in IDX format into the file specified. To restore from this IDX file, simply index it from the Index Tab.

Querying The DRE Through The GUI Admin

To query the DRE, click on the “Query” tab.

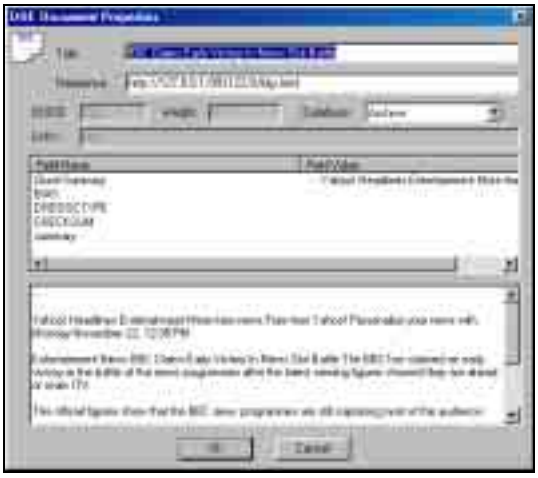


GUI ITEM	DESCRIPTION
Field Name	This specifies field restriction for the query. E.G. fnameTITLE=ANY(money)
Field Bool	This specifies how the fields are restricted with respect to each other. E.G. fnameTITLE+AND+fnamePRICE
Query Type	Natural means a natural language search. Proper Name search is where Proper Names, such as John Smith get treated as one compound word.
“Get Summary” check box	This will obtain the first few lines as a summary field which will be shown in the Document Properties window
Minimum Relevance	Obtains results that have at least this relevance value
Number of Results	Number of results to return

GUI ITEM	DESCRIPTION
Search within Date Ranges	Restrict results to a data range
Databases List	List of available databases. When doing a query you can select only the ones you want to query for. If none are selected, all of them will be queried.
“Suggest More” Button	Suggest documents related to the selected result(s).
“Remove Documents” Button	Remove selected document(s) from result list.
“View Document Properties” Button	View the properties of select document. Properties include, field values, date, doc id, relevance, etc..
“Include Content In Properties” tick box	Includes the plain text content in the Document Properties window.
“Re-index changes to Properties”	When changes are made in the Document Properties, it will re-index the changes into the DRE. NOTE: This can only be done when you are storing plain text content in the DRE.

Enter the appropriate criteria for your query and click on the “Do Query” button. Your results will then be displayed in the Results table as shown below.

From here it is possible to view the content of each result returned. If the reference starts with http:// then the GUI Admin will launch your default browser when you double click on the result. Otherwise, simply select the required document, make sure that the “Include content in properties” check box is checked and click on the “View Document Properties” button:



If you have marked the “Re-index changes to properties” checkbox in the Query DRE Window, you will be able to edit the contents of the document and this will then be saved. As well as editing results it is also possible to remove results from the DRE. Simply select the appropriate document and click on the “Remove from DRE” button.

Also, from the Query DRE Window, it is possible to suggest further documents based on a selected result. To do this select the appropriate result in the Results box and click on the “Suggest More” button. These results will then be displayed in the Results box and again it is possible to perform the above stated functions on these results.

DRE Admin Naming Conventions

In order to be able to administer the DRE using the DRE Admin you must ensure that you the DRE Admin is named after the DRE executable and the DRE's configuration file. Note that you must postfix this name with "**Admin**" for the DRE Admin executable.

For example:

CompanyDRE.exe

CompanyDRE.cfg

CompanyDREAdmin.exe

In this example, the DreAdmin executable, DRE executable and DRE configuration file all share the name "Company", which ensures that they are able to communicate with each other.

Appendix B: Languages and the DRE

By default DREs are configured to process English data. If you want a DRE to run another language, you need to:

- enter appropriate values for the **language settings** that are contained in the DRE configuration file.
- place **sentence breaking files** into the directory that contains the DRE configuration file (if you want to use a language in which words are not delimited by spaces, for example Japanese).

Language settings

CharConv

The language in which the DRE is running. You can enter one of the following numbers:

0	European (this is the default)	**7**	Eastern European
1	Japanese	**8**	Russian WINANSI
2	Korean	**9**	Russian KOI8
3	Thai	**10**	Hebrew
4	Simplified Chinese	**11**	Greek
5	Traditional Chinese	**12**	Swedish
6	Traditional Chinese indexed as Simplified Chinese		

Note: please contact Autonomy for additional files, if you are using a non-European language.

TermSize

Allows you to specify the maximum number of characters that any term in the DRE can be. By default this is 10 (for English and other European languages). The recommended values for the different languages are:

10	English and other European languages	**30**	German
40	Korean	**20**	Thai
30	Japanese		

StripLanguage

This optional parameter allows you to select which language to use when stripping (**running**, for example, is stripped to **run**). You can enter one of the following options:

0	English	**9**	Advanced German
1	Conversion from UK to US English	**10**	French
2	no stripping	**11**	Greek
3	German	**12**	Swedish
4	Italian	**13**	Danish
5	Russian	**14**	Portuguese
6	Advanced English	**15**	Advanced Spanish
7	Spanish	**16**	Norwegian
8	Dutch		

For example:

StripLanguage=7

In this example the DRE applies a stemming algorithm that has been designed specifically for Spanish.

Note:

You should specify the 'advanced' setting for English (any variant of English) or German. However, if **StripLanguage** was set to **0**, **1** (for English) or **3** (for German) when content was indexed into the DRE, you need to ensure that the same setting applies when queries are sent to the DRE.

For languages that have not been implemented, use **6** or **2**.

SentenceBreaking

If you are using a language in which words are not delimited by spaces, you need to specify one of the following for **SentenceBreaking**:

For Thai:	**thaibreaking.dll** or **./thaibreaking.so**
For traditional or simplified Chinese:	**chinesebreaking.dll** or **./chinesebreaking.so**
For Japanese:	**japanesebreaking.dll** or **./japanesebreaking.so**
For Korean:	**koreanbreaking.dll** or **./koreanbreaking.so**

Sentence breaking files

If you want to use languages in which words are not delimited by spaces (Thai, Japanese, Chinese and Korean), you need to place external sentence breaking files into the directory that contains the Content DRE. The following table lists the files that the individual languages require and the language settings that you need to enter in the Content DRE's configuration file.

Thai

Required files:	**NT**	**UNIX**
	thaibreaking.dll	thaibreaking.so
	thaiconvlist.txt	thaiconvlist.txt
	thaidict.txt	thaidict.txt

Required DRE configuration settings:

CharConv=3

TermSize=20

StripLanguage=2

SentenceBreaking=thaibreaking.dll

Traditional Chinese

Required files:	**NT**	**UNIX**
	chinesebreaking.dll	chinesebreaking.so
	big5togb.txt	big5togb.txt
	wordlist.txt	wordlist.txt
	chineseconvlist.txt	chineseconvlist.txt

Required DRE configuration settings:

CharConv=5

TermSize=40

StripLanguage=2

SentenceBreaking=chinesebreaking.dll

Simplified Chinese

Required files:	NT	UNIX
	chinesebreaking.dll	chinesebreaking.so
	big5togb.txt	big5togb.txt
	wordlist.txt	wordlist.txt
	chineseconvlist.txt	chineseconvlist.txt

Required DRE configuration settings:

CharConv=4

TermSize=40

StripLanguage=2

SentenceBreaking=chinesebreaking.dll

Japanese

Required files:	NT	UNIX
	japanesebreaking.dll	japanesebreaking.so
	\dic\jtag.attr	/dic/system/jtag.attr
	dic\JTAG.hash	/dic/system/jtag.hash
	dic\jtag.id	/dic/system/jtag.id
	\dic\jtag.mrph	/dic/system/jtag.mrph
	dic\JTAG.offset	/dic/system/jtag.offset
	\dic\jtag.table	/dic/system/jtag.table
	jtag.dll	/dic/system/jtag.trie
	jtag.ini	jtag.ini
	jtag_at.dll	libcodeconv.so
	japaneseconvlist.txt	libjtag_at.so
	dic\JTAG.trie	libjtag.so
		japaneseconvlist.txt

Required DRE configuration settings

CharConv=1

TermSize=30

StripLanguage=2

SentenceBreaking=japanesebreaking.dll

Korean

Required files:	**NT**	**UNIX**
	koreanbreaking.dll	koreanbreaking.so
	koreanconvlist.txt	koreanconvlist.txt
	Koma.dll (NT only)	main.dat
	HanTag.dll (NT only)	prob.dat
	main.dat	main.fst
	prob.dat	prob.fst
	main.fst	pos.nam
	prob.fst	tag.nam
	pos.nam	tagout.nam
	tag.nam	connection.txt
	tagout.nam	stopposnam.txt
	connection.txt	tagname.txt
	stopposnam.txt	
	tagname.txt	

Note: by default all words are stemmed to verbs. If you want to stem to nouns rather than verbs, you need to set the following sentence breaking settings in the **stopposnam.txt** file to **0** (by default they are set to **1**).

Required DRE configuration settings

CharConv=2

TermSize=40

StripLanguage=2

SentenceBreaking=koreanbreaking.dll

Appendix C: Error messages

Error message	Cause	Suggested action
ERROR: Not enough memory to create phrase cache of nn bytes	The DRE failed to allocate memory to the Index Cache.	Re-boot your computer and restart the DRE.
Could not stat “filename”	The DRE failed to retrieve information about this file.	Make sure that this file exists.
Queryh caught SIGINT signal – exiting	Ctrl-C has been pressed.	
Queryh caught SIGQUIT signal - exiting.	Ctrl-\ has been pressed.	
Queryh caught SIGBUS signal - exiting.	Access to an invalid address is attempted.	Restart the DRE.
Queryh caught SIGSEGV signal - exiting.	An invalid access has been attempted to valid memory segment.	Restart the DRE.
Insufficient memory to allocate string	The DRE failed to allocate memory for a default query that has been set in the configuration file.	Re-boot your computer and restart the DRE.
Couldn't allocate memory for query thread	The DRE failed to allocate memory for this query.	Re-boot your computer and restart the DRE.
out of mem : nn	The DRE failed to load terms in the memory.	Re-boot your computer and restart the DRE.
Can't allocate memory for Commit Buffer.	The addition and deletion of nodes are tracked by means of a buffer. The DRE failed to allocate memory for this buffer.	Re-boot your computer and restart the DRE.
Unable To Create File Mapping for “filename”, size=nn, errornum = nn, then the following message, Fatal Memory Map error (CreateFileMapping)	Memory mapping failed.	Re-boot your computer and restart the DRE.
File open failure for memory mapping for file	Memory mapping failed.	Re-boot your computer and restart the DRE.

Error message	Cause	Suggested action
Error while creating “file name”	Memory mapping failed.	Re-boot your computer and restart the DRE.
Failed to create document list	Memory mapping failed.	Re-boot your computer and restart the DRE.
Failed to initialize memory. Exiting	Memory mapping failed.	Re-boot your computer and restart the DRE.
Out of Memory in index terms membufferAlloc	The DRE failed to allocate memory for individual terms during indexing.	Re-boot your computer and restart the DRE.
Out of memory in refindexDeleteDocs nn	The DRE failed to allocate the memory that is needed to perform the compact operation.	Re-boot your computer and restart the DRE.
Could not stat “file_name”. Error:”ERROR_CODE”	The file's statistics are not available.	Make sure that the file exists.
Cannot open file “filename”	The file is missing.	Make sure that the file exists and is not a read-only file.
Unable to realloc tempdoclist nn	DRE uses a temporary buffer to keep track of terms, weight etc. If the DRE fails to allocate memory for this buffer the following message is produced.	Re-boot your computer and restart the DRE.
ERORR: could not load library	The security library file is missing.	Check that the security library file exists.
Warning: Failed to find function \"securityCreate\" in library “lib_name”	Problem with the security file.	Check that the library file is not corrupted or missing.
Error: Unable to create instruction thread	Thread creation problem.	Re-boot your computer and restart the DRE.
Could not Open Q.PID to output process id: “ERROR_CODE”	The Q.PID file is missing or read-only.	Make sure that the file is not write-protected or missing.

Glossary

APCM (Adaptive Probabilistic Concept Modelling)

Terms are given a weight according to their statistical importance in the DRE. Terms can have a weight between 0 and 255.

Concept summary

A brief summary of each result document that is returned for a query. The concept summary displays a few sentences that are typical of the result's content (these sentences can be from different parts of the result document).

Connector

A Connector is an Autonomy fetching solution (for example HTTPFetch, Oracle Fetch, AutoIndexer and so on) that allows you to retrieve information from any type of local or remote repository (for example, a database or a web site). It imports the fetched documents into IDX or XML file format and indexes them into a DRE from where you can retrieve them (for example by sending queries to the DRE).

Context summary

Returns a conceptual summary of the result document that is biased by the terms in the query. A context summary comprises sentences that are particularly relevant to the terms in the query (these sentences can be from different parts of the result document).

Database

An Autonomy database is a DRE data pool that stores indexed information. The administrator can set up one or more databases, and specifies how data is fed to the databases.

DRE (Dynamic Reasoning Engine)

The Dynamic Reasoning Engine is a scalable, multithreaded process which is based on advanced pattern-matching technology that exploits high-performance probabilistic modeling techniques. The DRE contains databases into which you can index information using a Connector, the DRE Administration tool or an indexing command. You can then access this information through a front end or by sending query commands to the DRE.

IDX

Apart from XML files only files that are in IDX format can be indexed into a DRE. You can use a Connector to import files into this format or manually create IDX files (please refer to the chapter **Importing and indexing data into the DRE**).

Link term

Link terms (also referred to as "Links") are terms in query text that are also contained in the result documents that the DRE returns for this query.

Query

You can submit a natural language query to the DRE which analyzes the concept of the query and returns documents that are conceptually similar to the query. You can also submit Boolean, bracketed Boolean and keyword searches to the DRE.

Quick summary

A brief summary of each result document that is returned for a query. The quick summary displays the first few sentences of the result document.

Index

E

H

I

K

L

N

P

Q

R

S

T

W