exalead™

EXALEAD ONE:ENTERPRISE QUERY LANGUAGE GUIDE

AN *EXALEAD* S.A. OEM SUPPORT DOCUMENT

This document details the Exalead Query Language

Doc. No. EN.120.021.0-V4.6.0 - March 4, 2008

Copyright © 2008 by *Exalead* S.A. All rights reserved.

Legal Notice

Information in this document is subject to change without notice and does not represent a commitment on the part of *Exalead* S.A..

This document is Copyright © by *Exalead* S.A. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying or recording, for any purpose without the express written permission of *Exalead* S.A..

Exalead is a Registered Trademark of *Exalead* S.A.. ExaScript is a Trademark of *Exalead* S.A..

This document makes reference to other names and products that are Trademarks of their respective owners.

The software described in this document is Copyright © by *Exalead* S.A. and is supplied under a licence agreement, a nondisclosure agreement, or both. It is against the law to copy or transmit this software by any medium except as specifically sanctioned by these agreements.

For more information about *Exalead*:

- See www.exalead.com
- Write to us at:

  Exalead
  10 place de la Madeleine
  75008 Paris
  FRANCE
- Telephone: +33 (0)1 55 35 26 26
- Fax: +33 (0)1 55 35 26 27

For Product Support in the US and Canada only:

- Hotline: +1 (212) 786-7450
- Email: us-support-oneenterprise@exalead.com

For Product Support outside the US:

- Hotline: +33 1 55 35 26 00
- Email: support-oneenterprise@exalead.com

Exalead S.A. is incorporated with a nominal capital of 99 179.20 €.
RCS Paris B 432 887 875
APE 722C

This product has several patents pending.

Printed in France.

© 2008 BY *EXALEAD* S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

Table of Contents

Preface

Query Language

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

exalead™

PREFACE

How to use this guide

Section Overview

About this section

This section introduces the Query language for **exalead one:enterprise**. It provides an overview of the contents of this guide and conventions used in it.

What is in this section

This section covers the following.

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

Revision History

Version	Date	Author	Document revision history
1.0	June 26, 2007	A. Derbel	Creation of document.
4.6.0	March 4, 2008	A. Derbel	Modified the versioning system, document format and images. Added custom fields via field translators section.

Document status and rules for document version numbering

The version of a document is expressed in the form Vn.m

Where:

n =	The version of the **exalead one:enterprise** application.
m =	The most recent modification to the document version.

Version reference	Possible increase of the version reference
Vn.m	The reference of the first applicable version is V4.6.0
n = **4.6** **m** equal to or greater than **0**	**Vn.m** → **Vn.m+1**: Minor change to the document. **Vn.m** → **Vn+1.0**: Change in the application version.

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

Acronyms and Terminology

Acronyms

The following acronyms are used in this guide.

Term	Definition
API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
URI	Universal Resource Identifier
PAPI	Push API
IP	Internet Protocol
URL	Universal Resource Locator
XML	Extensible Markup Language

Terminology

Some important terms used in this guide are defined below.

Term	Definition
commit	Information is crawled and processed to form a partial index in memory. An index *commit* is when this partial index is committed to hard disk.
cookie	Is a file stored on a client computer that persistently identifies the user.
corpus	All the documents, data, and information contained in your information sources.
document summary	After a document's title, the search results show an automatically-generated summary of the document's content that is related to the search terms.
host	Is a server (physical machine) in the **exalead one:enterprise** platform.
idate	Is the date content was indexed.
index	Is the index used by the search engine to serve queries. At the same time it is being updated in the background from the information sources.
information sources	Information repositories such as a mail server, a database, fileshare, etc.
metadata	Is information about the document, rather than actual document content. For example, metadata can be the name of the last person to modify it, or where it was last saved.
process	Is a software module running on a host.
query	Is a user's search request sent to the search engine.
search terms	The words and syntax that users type when making their search.

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

About this Guide

Documentation set

The **exalead one:enterprise** documentation set is as follows. The technical guides are:

Document number	Title
EN.120.001	**Exalead one:enterprise** Administration
EN.120.002	**Exalead one:enterprise** Connectors
EN.120.003	**Exalead one:enterprise** Security
EN.120.004	**Exalead one:enterprise** HTML Front-end Customization
EN.120.008	**Exalead one:enterprise** Installation

The front-end user and API guides are:

Document number	Title
EN.120.007	**Exalead one:enterprise** User Guide
EN.120.018	**Exalead one:enterprise** Search Front-end API: XML v10
EN.120.020	**Exalead one:enterprise** Search Front-end API: XML v3.1
EN.120.021	**Exalead one:enterprise** Query Lanuage

The technology guides are:

Document number	Title
EN.120.010	**Exalead one:enterprise** Categorization
EN.120.011	**Exalead one:enterprise** Filtering

Document audience

This document has been written to explain how to configure **exalead one:enterprise**. It is for people who have experience with:

- Web browser point and click (hyperlink) user interfaces.
- Regular expressions.

Purpose and application scope

This document describes the Query language features in **exalead one:enterprise**.

Formatting conventions used in this guide

The following formatting conventions are used in this document.

The text attribute	Is used for
Italic	Titles used in cross-references and variable names.
Bold	Emphasis.
Lechter Condensed font	All output and for anything you would type in a console window.
COURRIER NEW FONT	Text labels from a graphic user interface.

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

QUERY LANGUAGE

A description of the Exalead Query Languange conventions and methods

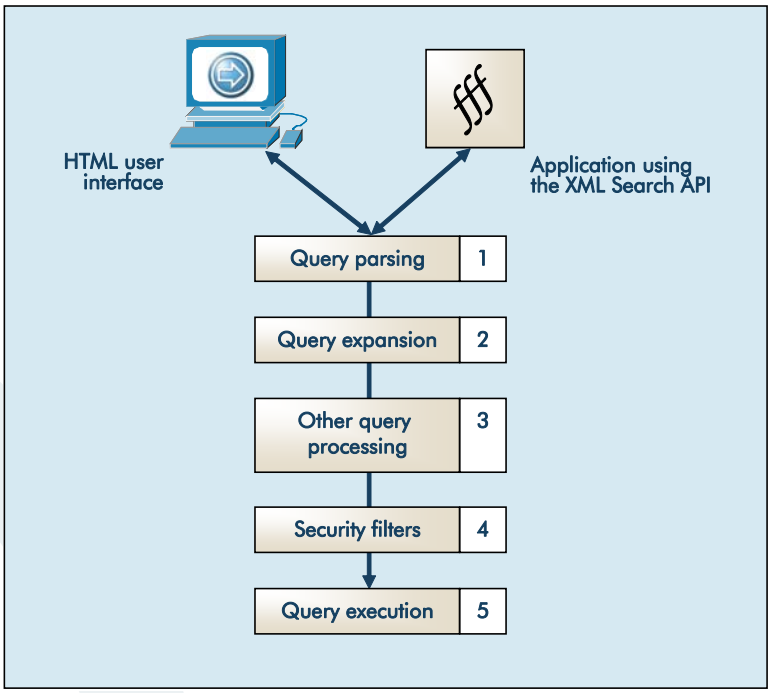
Overview

The **Exalead Query Language** provides a rich variety of operators and ranking modifiers, allowing at the same time simple "two words queries" like users type in their favorite Web search engine's search box, and elaborate search expressions such as the ones generated by semantic expansion tools.

The **Exalead Query Language** is used in all the Exalead search APIs and user interfaces, thus allowing power users to type advanced search queries in the input text box, and guaranteeing that applications using the Exalead XML Search API can take fully advantage of the same query operators and ranking parameters as the Exalead HTML front-end.

The Exalead Query Language is not only used to define the search queries sent to the Exalead search front-end, but also used internally by the Exalead query processing workflow to manipulate, enrich and optimize the search query before executing it.

Figure 1 The following diagram is a Query processing workflow



Step	Description (1 of 2)
1	The query is parsed into an XML object representing the query tree (See *Query Tree on page 13*)
2	The query tree is processed by query expansion modules, to enrich it (e.g. by using synonyms and semantics) or to interpret it (e.g. recognize city names or acronyms). The query expansion generates a new query tree. Note that the query expansion can be done by an external module, for example by calling a Web Service.
3	Other custom query processing modules can modify the query tree, thus allowing powerful customisation of the search application's behavior.

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

Step	Description (2 of 2)
4	If the application requires a document-level security, the user's access rights (user identifier and security groups) are added to the query to ensure that the results list contains only documents that the user is authorized to see.
5	The query tree is sent to the index server to be executed. The query can be executed in parallel on multiple physical servers.

Example 1 - Let's consider a typical query processing flow for a simple query: GUI design

Step	Description
1	The query is parsed, using the default 'AND' operator: `"GUI" AND "design"`
2	The query expansion module recognizes "GUI" as an acronym for "Graphical User Interface". The "GUI" leaf node of the query tree is replaced by a OR node containing the acronym expansion: `(("Graphical" NEAR "User" NEAR "Interface") OR "GUI") AND design` **NOTE- We use a NEAR operator between "graphical", "user" and "interface" because we want these two words to be close to each other in the document (otherwise the acronym expansion would be senseless)**
3	3.The query expansion modules replaces the "design" word by a list of synonyms, as this word is highly polysemic: `(("Graphical" NEAR "User" NEAR "Interface") OR "GUI") AND ("design" OR "conception")`

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

Query Tree

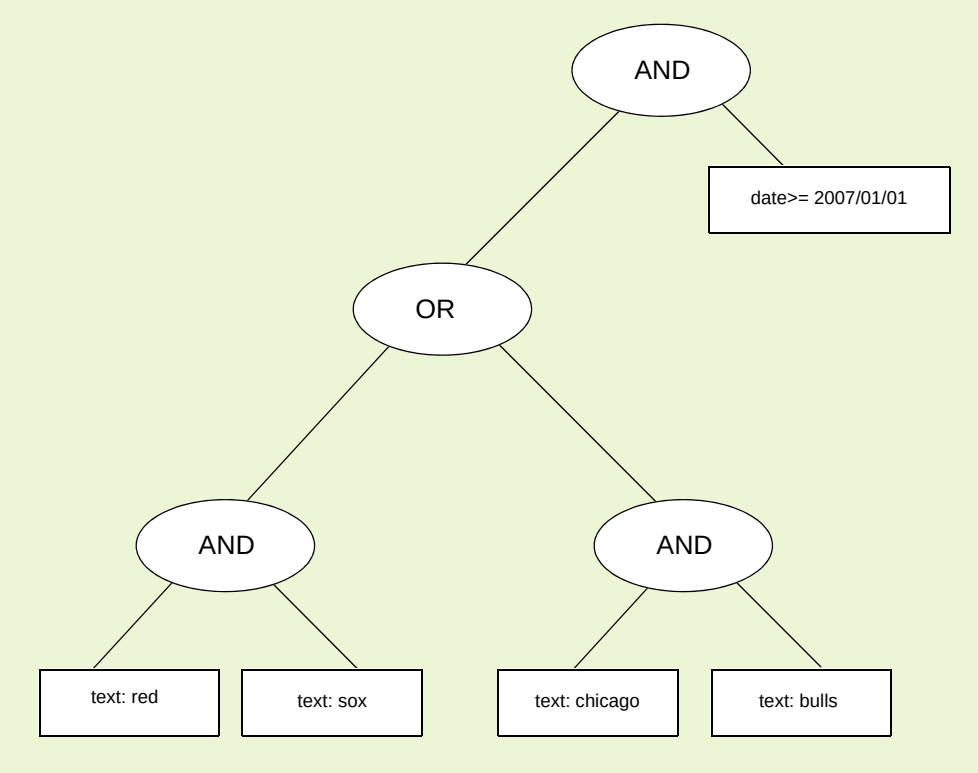
Query parsing

The internal representation of a parsed query is a tree where the inner nodes are query operators and the leaves are Boolean predicates.

The following query searches documents containing the words "Red Sox" or the words "Chicago bulls", and created or modified after the January 1st, 2007:

```
((Red Sox) OR (Chicago Bulls)) AND date>=2007/01/01
```

The query tree corresponding to this query is represented by the following diagram:



NOTE- The default operator between two text predicates (like "red" and "sox") is a boolean AND.

The Exalead query language provides various query operators (for the complete list of available operators, see *Query Operators on page 16*), such as boolean operators (AND, OR, NOT), word sequence operators (NEAR, PHRASE), score operators (MAX, MIN), etc.

Relevancy scores

Whenever a document matches a query predicate, it is given a score value, computed by the following formula:

```
Si = Z(Ri)oWi
```

where:

- Si is the score of the document for the i-th predicate
- Ri is the score class of the word's predicate, as defined in the document when it has been indexed
- Wi is the weight of the i-th predicate in the query
- Z is the mapping table between the index score classes and the query score values (the default value of Z is the identity function Z(x) = x)

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

This score value combines a ranking score that has been given to the document when it has been indexed (the Ri score class), a global query parameter set by the administrator to configure the relative weight of the index-time ranking score (the Z mapping table) and a per query term weight that is set when the query is executed (Wi).

The Wi value can be set explicitly for each query predicate by using the 'w' parameter in the score modifiers. In the following example, we set a term weight of 5000 for the word 'GUI' and of 1000 for the word 'design':

```
GUI{w=5000} AND design{w=1000}
```

When not specified, the default value of Wi is determined by the relative frequency of the query terms: the more frequent a word is, the less weighted it will be. The base weight is always 100000 for the rarest term. The default weight value of non-textual predicates (categories or numerical fields) is 0, so that without an explicit score modifier these predicates only affect the selection of matching documents, but do not have any influence on the ranking score.

For a complete list of the available score modifiers, see *Score modifiers on page 19*.

These score modifiers can be typically used to take control on the relative importance of each query term in the computation of the document relevancy. In the example given in section 1, the query expansion module would set a larger weight on the word 'GUI', which is the most meaningful in the user's query, and a light weight on 'design' and the associated synonyms, since this word is less relevant. This would the following weighted query:

```
((Graphical{w=20000} AND User{w=50000} NEAR Interface{w=50000}) OR
  GUI{w=100000})
AND
 (design{w=10000} OR conception{w=10000})
```

Query evaluation

Each tree leaf is associated with the collection of documents for which the corresponding predicate is true. These document collections are enumerated in parallel, thus producing lists of documents which are combined w.r.t. the operators found in the query tree's inner nodes. For example, a OR operator will merge the underlying lists together, whereas a AND operator will retain only the documents found in all the underlying lists.

The document lists are combined recursively, up to the root node of the tree, and the final list obtained after processing the root node is the query's results set.

At the same time the documents lists are combined, the score values computed for each predicates are also merged together to compute the overall document score. Most query operators simply add the score values of their underlying predicates, but some operators like MAX or MIN apply a different operation. Section 3.3 describes the score value combination operated by the different query operators.

The final ranking score computed after combining the score values of all the matching predicates is then added with two other score values assigned globally to the matching document:

- a 'proximity bonus' computed by analysing the relative position of the query terms in the matching document (a document where the query terms are close to each others will receive a larger bonus than a document where the query terms are far)
- a 'static score' that has been assigned to the document when it has been indexed (this score can for example be used to give a bonus value to the known 'popular' documents in the indexed corpus)

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

The relative weight of the proximity bonus and the static score in the final score are configured in the query parameters (either in the configuration of the search view, or directly in the XML request envelope associated with the query).

NOTE- The search results score computation is not affected by the query refinements (i.e. the query predicates added internally when processing a refinement on a category or keyword have a null score weight). The purpose of query refinements is to select a relevant subset of the search results and to interactively navigate through the search results, without modifying the ordering of the selected results. Categories can however be associated with a non null score when they are used in explicit query predicates with the corporate/tree field.

© 2008 BY *EXALEAD* S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

Query Operators

Predicates

ì

Kind	Syntax	Default weight value	Examples
Text	`[field:]word{options}`	100000 * (relative word frequency)	`GUI` `title:design{w=10000}`
Regular expressions	`[field:]/pattern/{options}`	100000 * (relative word frequency)	`title:/desi.*/{w=10000,#=100}`
Wildcards	`[field:]word*{options}` `[field:]*word{options}`	100000 * (relative word frequency)	`desi*` `title:*faces`
Numeric value	`field OP value{options}` where OP is one of ==, <, <=, >, >=, <>	0	`price<=10000{s=10000}`
Date/time	`field OP YYYY/MM/DD[-HH:mm:ss]{options}` where OP is one of ==, <, <=, >, >=, <>	0	`date>=2007/01/01` `time<=2007/04/12-12:00:00`
Category	`field/tree:"category path"{options}`	0	`corporate/tree:"Top/Attributes/Kind/pdf"`

NOTE- In the case when a field name is invalid, the query parser falls back to interpreting the `field:` operator as a regular word. For example, the query **xyz:foo** will be parsed as **xyz** AND **foo** if the field xyz does not exist.

Predicates matching modifiers

Operator	Description	Example
`+PREDICATE`	Prevents the predicate from being expanded (e.g. by lemmatization or approximate matching)	`+interfaces`
`matching level`	Defines the predicate's matching level. The level can be one of (i, n, *). See remarks below.	`interfaces{n}`
`g=level`	Defines a complementary matching level for a generalized" predicate. See remarks below.	`interfaces{n,g=*}`
`f=factor`	Defines the weight attenuation factor for the generalized term. The default value is 10. See remarks below.	`interfaces{n,g=*,f=10}`
`#=number`	Defines the maximum number of matching words returned by an approximate match or a regexp match	`/interfa.*/{#=100}`

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

Operator	Description	Example
l=length	Defines the maximum length for an approximate or a regexp match	/interfa.*/{#=100,l=12}
d=distance	Defines the maximum word distance for approximate matches	interfaces{#=100,d=1}
p	Allow phonetic matching	interfaces{#=100,p}

NOTE-

- The matching method between word predicates and the document words is defined by the predicate's matching level. The matching level can be one of the following:
  - **exact match**: the word predicate matches exactly the document's word. This is the default.
  - **case insensitive match**: the word predicate matches the document's word, even if the case of some letters are different (example: "the" and "The"). This level is specified by the "i" option.
  - **normalized match**: the word predicate matches the document's word, even if the case and the accents of some letters are different (example: "the" and "Thé"). This level is specified by the "n" option.
  - **lemmatized match**: the word predicate matches the document's word if both words have the same lemmatization root (example: "protector" and "protecting"). Note that the lemmatization level also ignores letter cases and accents.
- It is possible to define a second, larger matching level (called the "generalized" level, which can be used to expand the query but with a smaller ranking weight on generalized terms. The "g" option indicates the matching level used for the generalized terms, and the "f" option indicates the weight attenuation for the generalized terms. When using these options, the predicate is logically equivalent to a MAX expression combining the term and the generalized term. For example, the predicate interfaces{n,g=*,f=10} is logically equivalent to MAX(interfaces{n}, interfaces{*,w/=10})
- When a word predicate is potentially matching multiple words in the index dictionary (e.g. when using approximate or phonetic matching, or when using regular expressions), it is mandatory to specify a maximum number of matching words by using the "#" option.
- The default options are exact match, with no expansion (#=0,d=0).

Boolean operators

Operator	Predicate value	Operation on scores	Example
X1 AND X2	X1 AND X2	S1 + S2	User AND interface
X1 NEAR X2	X1 AND X2 The result is true only if the words matching X1 and X2 are close enough in the document	S1 + S2	User NEAR interface
X1 BEFORE X2	X1 AND X2 The result is true only if the words matching X1 are located before and close to those of X2	S1 + S2	User BEFORE interface
X1 AFTER X2	X1 AND X2 The result is true only if the words matching X1 are located after and close to those of X2	S1 + S2	interface AFTER user
X1 NEAR/DISTANCE X2	Same as NEAR, but with an explicit distance	S1 + S2	User NEAR/4 interface

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

Operator	Predicate value	Operation on scores	Example
X1 BEFORE/DISTANCE X2	Same as BEFORE, but with an explicit distance	S1 + S2	User BEFORE/4 interface
X1 AFTER/DISTANCE X2	Same as AFTER, but with an explicit distance	S1 + S2	interface AFTER/4 user
"X1 X2"	X1 AND X2 The result is true only if the words matching X1 and X2 are adjacent in the document	S1 + S2	"User interface"
X1 NEXT X2	Same as "X1 X2"	S1 + S2	User NEXT interface
MIN(X1, X2, …)	X1 AND X2 AND …	MIN(S1, S2, …)	MIN(User, interface)
X1 OR X2	X1 OR X2	S1 + S2 Only the matching predicates are taken into account	Design OR conception
MAX(X1, X2, …)	X1 OR X2 OR …	MAX(S1, S2, …) Only the matching predicates are taken into account	MAX(Design, conception)
NOT X1	NOT X1	S1	NOT ("design pattern")
-X1	NOT X1	S1	-design
OPT X1	Always true	S1 if X1 is true, 0 otherwise	OPT graphical
NOOPT (EXPR)	No effect on the evaluation of EXPR, but prevents EXPR from being made optional in the context of a category "widen" navigation operation	N/A	User interface, OOPT(site:microsoft.com)
X1 BUTNOT X2	X1 BUTNOT X2 The result is true only if there is at least one instance of X1 which is not an instance of X2 in the document	S1	York BUTNOT "New York"

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

NOTE-

- The "-" operator is interpreted as a `NOT` operator only if it is at the beginning of a word, after a whitespace. For example, Sarbanes -Oxley is equivalent to `Sarbanes AND NOT Oxley`, whereas `Sarbanes-Oxley` is equivalent to `Sarbanes NEXT Oxley`
- Non-whitespaces word separators (".", "&", "-", etc.) are interpreted as a NEXT operator instead of a AND, when they are used to separate words without additional whitespaces. For example ASP.NET will be parsed as ASP NEXT NET instead of ASP AND NET
- The query parser allows non-alphanumerical chars in words in a few special cases (for example "c++", "C#", "$100")
- The query operators operating on word positions (NEAR and NEXT) cannot be used to combine expressions whose "position" cannot be computed. For example, the query music NEAR (Madonna AND mp3) is illegal, because the expression Madonna AND mp3 cannot be associated with a single word position value.
- When a "widen" action on a category is processed, the query parser automatically adds a OPT operator in front of the query predicates, so that the widened query returns all the documents corresponding to the selected category, and not only the documents matching the query. However, it is possible to prevent some of the query predicates from this translation by using the NOOPT operator.

Score modifiers

Operator	Description	Example
`s=N`	Replace the predicate's score Z(Ri).Wi by an explicit value	`price<500{s=1000}`
`s+=N`	Increase the predicate's score by a given value	`GUI{s+=100000}`
`s-=N`	Decrease the predicate's score by a given value. NB: the score of a given predicate can be negative, but the final score of the document can never be lower than 0	`corporate/tree:"Top/Attributes/XXX" {s-=100000}`
`w=N`	Replace the predicate's weight Wi by an explicit value	`design{w=10000}`
`w*=N`	Multiply the predicate's weight by a given value	`design{w*=2}`
`w/=N`	Divide the predicate's weight by a given value	`design{w/=2}`

NOTE-

- The score modifiers are applied in the same order as they appear. In the case of two conflicting modifiers (e.g. {s=1000,s=2000}), the last one is applied.
- In the case when an explicit score (s=) is specified, the predicate's weight is ignored so the 'w' options have no effect.
- The explicit score should only be used for non-textual predicates (numeric values and categories) since this modifier completely ignores the ranking score class that was set when the document was indexed.

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

Standard fields

The **exalead one:enterprise** product's default configuration provides the following alphanumerical and numerical fields:

Field name	Description	Example
title	The document's title	title:(GUI design)
date	The document's last modification date/time	date>=2007/01/01
before	Equivalent to date<=DATE	before:2007/01/01
after	Equivalent to date>=DATE	after:2007/01/01
size	The document's size in bytes	size>=1048576
language	The document's language	language:fr
filetype	The document's file type	filetype:pdf
site	The document's Web site (only for HTTP connectors)	site:exalead.com

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

Custom fields

The **exalead one:enterprise** allows you to create custom operators using the following field translators:

Field name	Description	Example
`intitle:`	Allows you to search for a word or a group of words found within the title of a document. Use it, for example, to find a document where you know certain words are in the title.	`intitle:"movie star"` Searches for documents that have movie star in their title. or `intitle:insentence:(search engine)`
`inpage:`	These operators permit the user to search for words in specific sections of a document. Any section operator can be used in conjunction with any alphanumeric field search however, they cannot be nested.	`inpage:(search engine)` Searches for results that contain the words “search” and “engine” in the same page.
`insentence:`		`insentence:(search engine)` Searches for results that contain the words “search” and “engine” in the same sentence.
`inparagraph:`		`inparagraph:(search engine)` Searches for results that contain the words “search” and “engine” in the same paragraph.
`spellslike:`	This gives results that are spelled like the word you have entered.	`spellslike:exlaead`
`soundslike:`	This gives results sounding like the word you have entered.	`soundslike:exallead`

NOTE- In order for the section operators to be available, the indexing of word separators must be set in the exalead one:enterprise index configuration.

Categories

The categories attached to the indexed documents can be used to navigate throughout the search results list by using the navigation operations (refining or widening the query w.r.t the navigation topics returned in the category groups), but these categories can also be used directly as query predicates through the corporate/tree index field. The categories are organized in a virtual tree with a root node called Top, and subtrees corresponding to the category groups defined in the application. The following table describes the standard category groups defined in the **exalead one:enterprise** product.

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

Note that for convenience, some of the category groups described below are also mapped to search fields described in *Standard fields on page 20* (for example the `Top/Attributes/Kind` category group is mapped to the `filetype:` search field).

NOTE-

- In the table below, the Example column describes only the category paths. The complete query predicate is built by prepending the category field name (for example, `corporate/tree:"Top/Source/Intranet"`)
- a document can have multiple categories for a given group. For example, a document can be simultaneously in the `Top/Attribute/Authors/John Doe` and in the `Top/Attributes/Authors/Bill Smith` if the two people have contributed to the document.

Category group	Description	Example
Top/Source	The data source attached to the document	`Top/Source/Intranet`
Top/Attributes/Kind	The document's file kind	`Top/Attributes/Kind/pdf`
Top/Attributes/Language	The document's language	`Top/Attributes/Language/fr`
Top/Attributes/Authors	The document's author	`Top/Attributes/Authors/John Doe`
Top/Attributes/Recipients	The document's recipients (for e-mail documents)	`Top/Attributes/Recipients/John Doe`
Top/Attributes/Date	The document's date	`Top/Attributes/Date/2007/01/01`
Top/People	People names extracted by semantic filters	`Top/People/John Doe`
Top/Organization	Organization names extracted by semantic filters	`Top/Organization/Exalead`
Top/Location	Location names extracted by semantic filters	`Top/Location/France`

© 2008 BY EXALEAD S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

XML Queries

In addition to the "human readable" query language described in *XML Queries on page 23*, the exalead one:enterprise product also provides a XML DTD which can be used to define queries. The XML form of queries is a direct representation of the parsed query tree described in *Query Operators on page 16*: the inner nodes (operators) are XML tags containing their operands, and the tree leafs are XML tags describing the predicates.

The following tables describe the available XML tags.

Query predicates

Predicate Kind	Syntax	Examples
Text	<TextExpr kind="word" value="WORD" options="OPTIONS" />	<TextExpr kind="word" value"GUI" options="w=1000" />
Regular expression	<TextExpr kind="pattern" value="REGEXP" options="OPTIONS" />	<TextExpr kind="pattern" value"desi.*" options="w=1000" />
Numeric value	<NumExpr field="FIELD" operator="OP" value="VALUE" options="OPTIONS" /> where OP is one of EQ, NE, LE, LT, GE, GT	<NumExpr field="price" operator="LE" value="100" />
Date/time	<NumExpr field="FIELD" operator="OP" value="VALUE" options="OPTIONS" /> where OP is one of EQ, NE, LE, LT, GE, GT	<NumExpr field="date" operator="GE" value="2007/01/01" />
Category	<TextExpr field="FIELD" kind="litteral" value="PATH" options="OPTIONS" />	<TextExpr kind="litteral" field="corporate/tree" value"Top/Attributes/Kind/pdf" options="s=1000" />

Query operators

Operator	XML syntax
X1 AND X2	<AndExpr>…</AndExpr>
X1 NEAR X2	<NearExpr distance="DISTANCE" >…</NearExpr>
X1 BEFORE X2	<BeforeExpr distance="DISTANCE" >…</BeforeExpr>
X1 AFTER X2	<AfterExpr distance="DISTANCE" >…</AfterExpr>
X1 NEXT X2	<SeqExpr>…</SeqExpr>
X1 BUTNOT X2	<ButNotExpr>…</ButNotExpr>
field:(X1)	<FieldExpr field="FIELD" >…</FieldExpr>
MIN(X1, X2, …)	<MinExpr>…</MinExpr>
X1 OR X2	<OrExpr>…</OrExpr>
MAX(X1, X2, …)	<MaxExpr>…</MaxExpr>
NOT X1	<NotExpr>…</NotExpr>
OPT X1	<OptExpr>…</OptExpr>
NOOPT (EXPR)	<NoOptExpr>…</NoOptExpr>

© 2008 BY *EXALEAD* S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

© 2008 BY *EXALEAD* S.A. REPRODUCTION AND DISCLOSURE PROHIBITED.

About Exalead

Founded in 2000 by search-engine pioneers, Exalead (www.exalead.com) is a global provider of software that is designed to simplify all aspects of information search and retrieval for organizations of all sizes. Based on the first and only unified technology platform for desktop, intranet or Web search, Exalead offers easier deployment, administration and use than any other enterprise-type search software. This is true whether for one or thousands of desktops, a small business or global enterprise, and conforms to any technology environment. It also adapts to user habits for a uniquely satisfying search experience.

Exalead software is used by leading banking and financial services, media, consumer packaged goods, research, retailing sports entertainment and telecommunications companies around the world, including Air Liquide, BNP Paribas and Carlson Wagonlit. Exalead is an operating unit of Qualis, an international holding company.

Call Exalead for Your Information Access Needs

Talk to us about your information access needs. Our sales technicians will be delighted to answer any questions you may have.

Exalead - France
10 place de la Madeleine
75008 Paris, France
Tel: +33 (0)1 55 35 26 26
Fax: +33 (0)1 55 35 26 27

Exalead - USA
90 Park Avenue, Suite 1630
New York, NY 10016, USA
Tel: +1 (212) 786-7450
Fax: +1 (212) 786-7316

Exalead - Italy
Corso Giuseppe Garibaldi, 86
20121 - MILANO, Italy
Tel: +39 02 62 71 10 10
Fax: +39 02 62 71 10 11

Exalead - United Kingdom
International House
Stanley Bvd, Hamilton
Glasgow G72 0BN
Tel: +44 (0)1698 404630
Fax: +44 (0)1698 404639

Exalead - Germany
Niederlassung Deutschland
Robert-Bosch-Strasse 7
64293 Darmstadt
Tel: +49 6151 35 99 690-0
Fax: +49 6151 35 99 690-35

contact@exalead.com www.exalead.com