

Search Workbench Technical Document

Table of Contents

[Purpose](#)

[Dependencies](#)

[Data Model](#)

[Service Interface Design](#)

[Operators](#)

[Difference between AND/OR and NOT Operators](#)

[Cross Document Search](#)

[Query and Filter Query](#)

[Service Interface Design \(REST/SOAP\)](#)

[User Interface Design](#)

[Data Importing](#)

[Data Exporting](#)

[Workflow](#)

[System Parameters](#)

[Roles and Permissions](#)

Purpose

Search Workbench acts as a universal search engine for bibliographic data in the system. The search accepts a variety of search parameters to build the search query and allows multiple conditions to help users easily narrow down the required result. The search workbench also supports faceted search and can filter results on various criteria such as Format, Language, Genre to better narrow down the result. This document will detail aspects of the technology and the logical flow behind the Search functionality.

Dependencies

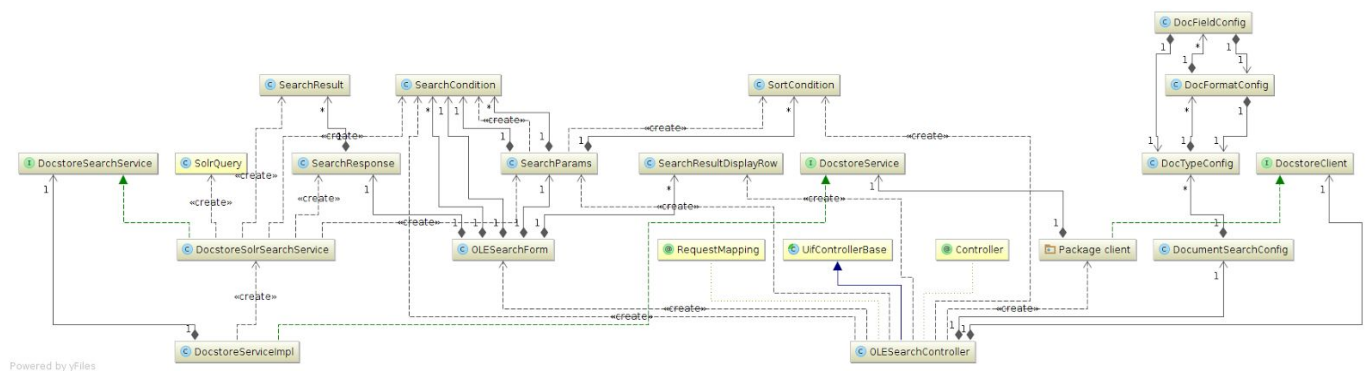
Though the bibliographic data resides in Database tables, the search workbench doesn't make use of data from the database. It neither reads nor writes data to database tables. The indexing is done by Apache Solr and the search queries are constructed as Solr queries.

Data Model

The Solr data model is unique and does not resemble the conventional RDBMS (Relational DataBase Management System) data models used throughout the rest of the system. However, they share some similarities with conventional data models.

More detailed information on Data Model is available [here](#).

Service Interface Design



The *OLESearchController* class controls the whole search functionality. The *start* method loads the initial search page with the *OLESearchForm*. The search form is loaded with default values - the document type is set as Bib, the search type is set as search, the browse field is title, operator is AND and field name is any.

The *search* method is triggered when the user enters the data in the various fields and clicks on search. The *searchDocstoreData* method is called which is where the search parameters and search conditions are processed.

After the user manipulates the search parameters and conditions as needed, the search parameters are passed to the *search* method of *DocstoreLocalClient* class from where the

Last Published October 30 2015

search method of the *DocstoreServiceImpl* class is called. From here the *search* method of the *DocstoreSolrSearchService* class is called where the search parameters are used to construct Solr queries.

In the *search* method, the *buildQueryWithSearchConditions* method is called to convert the values in the search parameters into meaningful Solr query. After the query is built it undergoes sorting as per the sort conditions in the *buildQueryWithSortConditions* method.

Operators

Search Workbench supports three operators AND, OR and NOT.

Difference between AND/OR and NOT Operators

The AND and OR operators are used to link two or more search conditions while the NOT operator is used to negate the current condition in which it resides.

For example, consider a collection of three books - Master of the game, The other side of midnight and Bloodline.

Search 1 “Master” in document type “Bibliographic” in field “Title” AND
 “Bloodline” in document type “Bibliographic” in field “Title”
Result No Search Results found

There are no records having both ‘Master’ and ‘Bloodline’ in the Title and hence the result.

Search 2 “Bloodline” in document type “Bibliographic” in field “Title” NOT
Result Master of the game, The other side of midnight

The NOT operator in a search condition negates the condition in which it is present. To explain further, look at this example:

To return results that contain ‘food’ in the Bibliographic title and filter out ‘dog food’, one would have to do a search as shown below

 “food” in document type “Bibliographic” in field “Title” AND
 “dog food” in document type “Bibliographic” in field “Title” NOT

This should be interpreted as

 “food” in document type “Bibliographic” in field “Title” AND
 NOT “dog food” in document type “Bibliographic” in field “Title”

In other words, if the conditions are *A AND B NOT* then it essentially is *A AND (NOT B)*


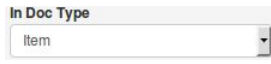
Search 3 “Dreams” in document type “Bibliographic” in field “Title” OR
 “Bloodline” in document type “Bibliographic” in field “Title”
Result Bloodline

The term 'Dreams' is not part of any of the available titles. However, since the operator is OR and since 'Bloodline' is available the result only brings up the title with 'Bloodline' in it.

Search 4 "The" in document type "Bibliographic" in field "Title" AND/OR
Result Master of the game, The other side of midnight, Bloodline
Condition The StopFilterFactory filter is used and 'The' is an entry in the *stopword.txt* file.

This results in 'The' being filtered out from the query. The empty search triggers a wildcard search bringing all the bib records. To explain further, look at this example:
If the user searches for "To be or not to be" and if 'to' and 'be' are listed under stopwords.txt and the StopFilterFactory filter is used then 'or not' would be the terms the query would use to search.



Cross Document Search

Search Workbench also allows cross document search. A cross document search is when the main document type () present at the top left of the screen, outside the search condition and one amongst the many 'in doc type' () inside the many search conditions differ.

For example, the following search is a cross document search

Search 5 Document Type: **Bibliographic**
"Bloodline" in doc type "**Bibliographic**" in field "Title" AND¹
"Book" in doc type "**Item**" in field "Item Type Name" AND²
"Available" in doc type "**Item**" in field "Item Status"

Query and Filter Query

When the main document type () and the 'in doc type' () are the same, the conditions form the **Filter Query**. The condition with the different 'in doc type' forms the **Query**.

For example, in the above search sample (**Search 5**) the main document type is **Bibliographic**. This makes the first search condition with 'in doc type' Bibliographic, the Filter Query. The second and third conditions with 'in doc type' Item becomes the Query³.

¹ AND/OR here are **not** taken into consideration since it is trying to link a Filter Query with a Query. NOT here would negate the condition and hence would be taken into consideration by the Filter Query.

² All operators here would be considered by the Query.

³ The Solr Queries are built in the *buildQueryWithSearchConditions* method of the DocstoreSolrSearchService class

The Filter Query, as the name suggests, filters data to match the search condition from the resultset of the Query. Operators between a Query and Filter Query are ignored. However, operators between Filter Queries and Queries work.

For example, in the above search sample (**Search 5**), the main query returns records with 'Item Type Name' as 'Book' AND 'Items Status' as 'Available'. The Filter Query, filters the records to check for bibliographic records having 'Bloodline' in 'Title', agnostic as to whether the operator is AND or OR between the main and filter queries.

A NOT operator however would have a different set of results. As already mentioned, a NOT operator applies to the search condition in which it is present unlike AND and OR which unites the search condition in which they are present to another. So in case of a NOT operator in the first search condition, the resultset of the query would contain items with 'Item Type Name' as 'Book' and 'Item Status' as 'Available' and the Filter Query filters the records to present those Bibliographic records without 'Bloodline' in 'Title'.

There can be only one 'in doc type' other than the 'Document Type' in a search. In other words, if the 'Document Type' is Bibliographic, then the 'in doc type' can either be Bibliographic or (Item or Holdings or Eholdings). So the pair could be one of the following - Bibliographic and Item, Bibliographic and Holdings or Bibliographic and Eholdings. It is not possible to have a search where Bibliographic, Item and Holdings/Eholdings can be specified together.

To Summarize

- Filter Query is when both the **Document Type** outside the search condition and the **in doc type** inside the condition are the same.
- Those **in doc type** differing from that of **Document Type** forms the Query.
- Query and Filter Query cannot be united with AND/OR Operators.
- Query and Filter Query can be negated with NOT Operator.
- Multiple Queries and multiple Filter Queries can be united with operators independently.
- A Filter Query doesn't alter the results from the Query. It will filter results from the resultset brought by the Query.
- If the resultset from Query is empty, the Filter Query will also return an empty resultset.
- There can be only one document type **in doc type** other than the one defined in **Document Type**.

More information on Searchable fields, Facet fields, Field definitions are found [here](#).

Service Interface Design (REST/SOAP)

OLE supports REST services for search workbench. This is documented [here](#).

User Interface Design

The search workbench uses KRAD's UIF (User Interface Framework). A very good guide on this can be found [here](#).

Data Importing

Not Applicable.

Data Exporting

The search results can be exported as XML from the Search Workbench. Multiple records can be selected using the checkbox provided and the button "Export to Doc XML" at the bottom of the search result would open a text field containing the XML.

Workflow

Not applicable.

System Parameters

NOTE: The following System Parameters are currently available only in OLE 1.6.2 and will eventually be included in OLE 2.0.

Namespace Code	Parameter Name	Description
OLE-DESC	BIB_SEARCHSCOPE_FIELD	The parameter is used to specify search scope and search field in the search workbench for the Bib Doctype.
OLE-DESC	EHOLDINGS_SEARCHSCOPE_FIELD	The parameter is used to specify search scope and search field in the search workbench for the Eholdings Doctype.
OLE-DESC	HOLDINGS_SEARCHSCOPE_FIELD	The parameter is used to specify search scope and search field in the search workbench for the Holdings Doctype.
OLE-DESC	ITEM_SEARCHSCOPE_FIELD	The parameter is used to specify search scope and search field in the search workbench for the Item Doctype.

Roles and Permissions

Not applicable.