Britannica Concept Search for Apache Solr Technical Manual

Compatible with Apache Solr
Version 5.x and up

Version 5.4.0
October 2018

Britannica Concept Search for Apache Solr - Technical Manual

System specifications and the information contained in this document are subject to change without notice.

Encyclopaedia Britannica

Natural Language Technology Division

331 N.LaSalle Street,

Chicago, Illinois

USA 60654

Tel: (312) 205-6440

Fax: (312) 294-2162

E-mail: nltsupport@eb.com

Website: http://corporate.britannica.com/nlt/

Table of Contents

Chapter 1

Introduction

1.1. Welcome

Thank you for choosing the Britannica Concept Search for Apache Solr™.

Once installed, users have access to the powerful capabilities of Britannica Concept Search to perform thorough, yet efficient morphological searches (see the next two sections for additional details).

This document describes the technical steps for installing and integrating Britannica Concept Search with the Apache Solr.

1.2. Morphological Search

Morphological Search means that the analysis of a text, whether on a query or upon indexing, is done based on all inflected forms of the stems in the text. For elaborated information on morphological search as well as on the concept of stem, see the Britannica Concept Search User Manual for a specific language.

Britannica Concept Search is a robust solution provided by Encyclopaedia Britannica's Natural Language Technologies (NLT) Division for analyzing languages characterized by a complex and rich morphology such as Hebrew, Arabic and Persian.

Britannica Concept Search for Apache Solr enables Apache Solr based search solutions to support morphological searches (both inflectional and derivational), synonym search, and cross language search.

1.3. Notes on Britannica Concept Search Editions

Installation is the same for all three editions of Britannica Concept Search. The following is a summary of the features available in each edition of the Britannica Concept Search software:

Concept Search				
Standard Edition	░	▒	▓	Morphological Search & Context Analysis Exact Match Search
Enterprise Edition		▒	▓	Expanded Morphological Search Synonym Search English Search Cross Language Search Proper Name Transcription Search
Ultimate Edition			▓	Entity Search Faceted Navigation

1.4. Additional Help

For further explanations and clarifications, please contact us by e-mail at nltsupport@eb.com.

Installation

2.1. Prerequisites & Technical Information

In order to use Britannica Concept Search for Apache Solr, the following prerequisites are required:

- Oracle's (Sun) Java JRE version 1.8 or higher.
- An installed version of Apache Solr 5 or higher.

A maximal memory footprint of 100 MB in RAM needed by Britannica Concept Search. The total memory requirement should include this memory size in addition to Apache Solr's own requirements. The memory footprint not affected by loading Britannica Concept Search by multiple search threads.

- Disk space of 200MB for the complete installation of Britannica Concept Search.

2.2. Installing Britannica Concept Search for Apache Solr

▸ To install the Concept Search for the Apache Solr base software package

Windows:

1. Download the appropriate installation exe files.
2. Install the base Concept Search for Apache Solr package by running the program (use "run as administrator"): `concept_search_solr_base_win.<version>.exe`

   During the installation, click **<Next>** and accept all defaults.

   Provide a valid license key (see section 3 below for additional details).
3. Define the following environment variables (if needed):

   MEL_LUCENE_SEARCH_HOME to point to the folder where the package installed.
4. Update Solr's configuration files, located in the 'conf' folder in the relevant Solr collection\core folder, (ex. C:\solr-<version>\server\solr\collection1\conf) to use Melingo's Analyzer see the example files under %MEL_LUCENE_SEARCH_HOME%\example
5. Copy entire Melingo's **jar** directory to $SOLR_DIR\server\solr-webapp\webapp\WEB-INF\lib

Linux:

1. Download the appropriate installation file.
2. Extract the base Concept Search for Lucene Java package from the **`concept_search_solr_base_<platform>.<version>.tar.gz`** file to the same folder that was used for the base package (`$MEL_LUCENE_SEARCH_HOME`)
3. Define the following environment variables (see file export.sh in Melingo example folder):
   - **MEL_LUCENE_SEARCH_HOME** to point to the folder where the package extracted to (`export MEL_LUCENE_SEARCH_HOME=...`)
   - **LD_LIBRARY_PATH** to point the lib64 folder inside the `$MEL_LUCENE_SEARCH_HOME` folder.
4. Provide a valid license key (see on page 9 for additional details).
5. Copy entire Melingo's **jar** directory to $SOLR_DIR\server\solr-webapp\webapp\WEB-INF\lib

Update files "**manage-schema**" in directory:

...\solr-<version>\server\solr\configsets\data_driven_schema_configs\conf

When a new core/collocation created, the config files copied from this directory.

schema.xml / manage-schema:

The following section assumes that you have installed Solr, and explains how to incorporate the plugin into its example server.

Add a field type definition for each of the languages you wish to support, for instance the Hebrew analyzer can be associated with the **text_melingo** field type in the "schema.xml"/"manage-schema" as follows:

Define Analyze with auto language detection:

```
<fieldType name="text_melingo" class="solr.TextField">
<analyzer type="index" class="com.melingo.search.lucene.MorphologicalIndexAnalyzer" />
<analyzer type="query" class="com.melingo.search.lucene.MorphologicalQueryAnalyzer" />
</fieldType>
```

If you wish to use tokenizer, you can define it as follows:

```
<fieldType name="text_melingo" class="solr.TextField">
 <analyzer type="index" >
   <tokenizer class="com.melingo.search.lucene.MorphologicalIndexingTokenizer"/>
 </analyzer>
 <analyzer type="query">
   <tokenizer class="com.melingo.search.lucene.MorphologicalQueryTokenizer"/>
 </analyzer>
</fieldType>
```

Define one of the fields in your xml file to be of **text_melingo** type as in the content field below:

```
<field name="content" type="text_melingo" indexed="true" stored="true" multiValued="true"/>
```

Note that in order to use the plug properly, writing permissions on the installation directory should be gave to the user running the engine.

Here are the definitions for using Hebrew/Arabic/Farsi languages directly:

For Hebrew Analyzer:

```
<fieldType name="melingo_he" class="solr.TextField">
<analyzer type="index" class="com.melingo.search.lucene.MorphologicalIndexAnalyzerHe" />
<analyzer type="query" class="com.melingo.search.lucene.MorphologicalQueryAnalyzerHe" />
</fieldType>
```

For Hebrew Tokenizer:

```
<fieldType name="Melingo_he" class="solr.TextField">
 <analyzer type="index" >
   <tokenizer class="com.melingo.search.lucene.MorphologicalIndexingTokenizerHe"/>
 </analyzer>
 <analyzer type="query">
   <tokenizer class="com.melingo.search.lucene.MorphologicalQueryTokenizerHe"/>
 </analyzer>
</fieldType>
```

For Arabic Analyzer:

```
<fieldType name="melingo_ar" class="solr.TextField">
<analyzer type="index" class="com.melingo.search.lucene.MorphologicalIndexAnalyzerAr" />
<analyzer type="query" class="com.melingo.search.lucene.MorphologicalQueryAnalyzerAr" />
</fieldType>
```

For Arabic Tokenizer:

```
<fieldType name="Melingo_ar" class="solr.TextField">
 <analyzer type="index" >
   <tokenizer class="com.melingo.search.lucene.MorphologicalIndexingTokenizerAr"/>
 </analyzer>
 <analyzer type="query">
   <tokenizer class="com.melingo.search.lucene.MorphologicalQueryTokenizerAr"/>
 </analyzer>
</fieldType>
```

For Farsi Analyzer:

```
<fieldType name="melingo_fa" class="solr.TextField">
<analyzer type="index" class="com.melingo.search.lucene.MorphologicalIndexAnalyzerFa" />
<analyzer type="query" class="com.melingo.search.lucene.MorphologicalQueryAnalyzerFa" />
</fieldType>
```

For Farsi Tokenizer:

```
<fieldType name="Melingo_fa" class="solr.TextField">
 <analyzer type="index" >
   <tokenizer class="com.melingo.search.lucene.MorphologicalIndexingTokenizerFa"/>
 </analyzer>
 <analyzer type="query">
   <tokenizer class="com.melingo.search.lucene.MorphologicalQueryTokenizerFa"/>
 </analyzer>
</fieldType>
```

2.3. Licensing

A valid license is necessary in order to use the Britannica Concept Search system. The licensing steps performed by manually invoking the licensing command-line utility for the applicable product after it installed. The license utility for each language package can found inside the respective folder under the installation directory, in a subfolder called *license*. For example, the license utility for the Hebrew package can found at under */Hebrew/license* and called *HeCSKey.exe*.

In order to license a product:

1. Run the appropriate licensing utility from the command line or from the terminal (in Windows and Linux accordingly). It will output a unique Britannica machine code for the computer on which the software installed.
2. To obtain a license, please copy the machine code and e-mail it to your Encyclopaedia Britannica contact or to the e-mail support line: *nltsupport@eb.com*. You will provided with the appropriate license key.
3. When you receive the license key in a return e-mail from technical support, run the licensing utility again, this time passing it the license key as a parameter. You will receive the message *"License has been successfully installed."*

Chapter 3

Integration

3.1. Testing Concept Search on an Apache Solr Project

1. Start apache Solr server.

2. Index some documents using the schema that modified during the installation of the plug.

3. Navigate to http://localhost:8983/solr/ and select the appropriate core

4. Press "Query" and fill '**q**' field with a query text and the '**df**' field with the string "content".

5. Press "Execute Query" and make sure those morphological results shown in the results JSon output.

Note:

In case of getting exception: "access denied java.lang.RuntimePermission loadLibrary…"

- Edit file "java.policy" which located under Java directory in lib/security folder.

  Add the following line inside grant section:

```
permission java.lang.RuntimePermission "loadLibrary.*";
```

Appendix A
Glossary

The following terms may be encounter in your use of this product:

Term	Definition
Corpora	A set of texts that comprise a database on which a morphological search engine indexes and performs queries. Short for text corpus.
JRE	Stands for Java Runtime Environment, an implementation of the Java Virtual Machine, which executes Java programs.
Morphology	The structure and form of words in language, including inflection, derivation, and the formation of compounds.
Strong Key	See Strong Name.
Strong Name	A mechanism used in the Microsoft Java framework to uniquely identify a component (e.g. *dll* file), as a measure against the situation of "*dll* hell", in which the existence of more than one component with the same naming but with different versions leads to many conflicts.
Thread	A way for a program to split itself into two or more simultaneously running tasks. Short for thread of execution.
Tokenizer	A tool that breaks text up into tokens.