

Melingo Concept Search for Elasticsearch

- Technical Manual -

Version 7.0

July 2019

Melingo Concept Search for Apache Elasticsearch - Technical Manual

Copyright © 2019

Version 7.0, July 2019

All rights reserved. No part of this manual may be reproduced or copied in any form by any means - graphic, electronic or mechanical, including photocopying, typing, or information retrieval systems - without written permission of Melingo.

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

Melingo

Natural Language Technology

16 Tozeret Haaretz Street,

Tel-Aviv 67891

Israel

Tel: (972) 6070-403

Fax: (312) 6070-401

E-mail: techsupp@melingo.com

Website: <http://www.melingo.com>

Table of Contents

CHAPTER 1. Introduction	5
1.1. Welcome	5
1.2. Morphological Search	5
1.3. Additional Help	5
CHAPTER 2. Installation	6
2.1. Prerequisites & Technical Information	6
2.2. Installing Melingo Concept Search for Elasticsearch	6
2.3. Licensing	8
CHAPTER 3. Testing and Troubleshoots	9
3.1. Testing the integration	9
3.2. Definitions	9
APPENDIX A. Glossary	Error! Bookmark not defined.

1.1. Welcome

Thank you for choosing the Melingo Concept Search for Elasticsearch™.

Once installed, users have access to the powerful capabilities of Melingo 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 Melingo Concept Search with the Elasticsearch.

1.2. Morphological Search

Morphological search is basically, 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 Melingo Concept Search User Manual for a specific language.

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

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

1.3. Additional Help

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

2.1. Prerequisites & Technical Information

In order to use Melingo Concept Search for Elasticsearch, the following prerequisites are required:

- Oracle's (Sun) JDK version 1.8 or higher.
- An installed version of Elasticsearch.

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

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

2.2. Installing Melingo Concept Search for Elasticsearch

- To install the Concept Search for the Elasticsearch software package

Windows:

1. Download the appropriate installation .exe files.
2. Install the base Concept Search for Elasticsearch package by running the program:
`concept_search_for_elasticsearch.<version>.exe`
During the installation, click **Next** and accept all defaults.
Provide a valid license key (see [section 2.3](#) below for additional details).
3. Make sure the following environment variable has defined:
MEL_LUCENE_SEARCH_HOME needs to point to the folder where the package has installed.
4. Copy the configuration file **InstallMelingoPlugin.bat** from the folder **plugin** in Melingo Elasticsearch directory into the Elasticsearch **bin** folder, and run the batch file in order to install Melingo plug-in into your local Elasticsearch.
5. For testing the integration, go to [section 3.1](#)
6. For troubleshoot, go to [section 3.2](#)

Linux:

1. Download the appropriate installation file.
2. Extract the base Concept Search for Lucene Java package from the `concept_search_elasticsearch_base_<platform>.<version>.tar.gz` file to the same folder that was used for the base package (`$MEL_LUCENE_SEARCH_HOME`)
3. Define the following environment variables (can be copy from the file `exports.sh` in plugin folder):
 - **MEL_LUCENE_SEARCH_HOME** needs to point to the folder where the package has extracted to (`export MEL_LUCENE_SEARCH_HOME=...`)
 - **LD_LIBRARY_PATH** needs to point the **lib64** folder inside the `$MEL_LUCENE_SEARCH_HOME` folder.
 - When using elasticsearch as a **service** please update the environment variables by:
Edit `/usr/lib/systemd/system/elasticsearch.service` and add 2 lines under `[service]`
`Environment=MEL_LUCENE_SEARCH_HOME=/usr/melingo/melingo_es`
`Environment=LD_LIBRARY_PATH=/usr/melingo/melingo_es/lib64`

*** For Redhat & Centos ***

Edit `/etc/init.d/elasticsearch` and add 2 lines

```
export MEL_LUCENE_SEARCH_HOME="/usr/melingo/melingo_es"
export LD_LIBRARY_PATH="/usr/melingo/melingo_es/lib64"
```

4. Add the following line inside grant section in file `java.policy`:
`permission java.lang.RuntimePermission "loadLibrary.*";`
The `java.policy` file is in Java directory
 - Java 8: `/usr/java/.../lib/security`
 - Java 11: `/usr/java/.../conf/security`
5. Provide a valid license key (see [section 2.3](#) below for additional details).
6. Copy from the Melingo Elasticsearch directory the configuration file: **installmelingoplugin.sh**, located in the **plugin** folder into the Elasticsearch **bin** folder. Run the batch file as root user (su) in order to install Melingo plugin into your local Elasticsearch.
7. For testing the integration go to [section 3.1](#)

2.3. Licensing

A valid license is necessary in order to use the Melingo Concept Search system. The licensing steps are performed by manually invoking the licensing command-line utility, for the applicable product after it has been installed. The license utility for each language package can be found inside the respective folder under the installation directory, in a subfolder called *license*. For example, the license utility for the Hebrew package can be found at *Hebrew/license* and is 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 Melingo machine code for the computer on which the software is being installed.
2. To obtain a license, please copy the machine code and e-mail it to your Melingo contact or to the e-mail support line: *techsupp@melingo.com*. You will be 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."*

3.1. Testing the integration

A: Sanity test:

1. Restart Elasticsearch server.
2. Go to Melingo's example folder.
3. Run `analyze_hebrew.bat/analyze_hebrew.sh` to check if plugin successfully installed.

B: Checking indexing and search with Melingo Analyzer plug-in

1. Run `Mappings.bat/mappings.sh` in order to setting and mapping new index
2. Run `Indexing.bat/indexing.sh` to index the example files.
3. Run `Searching.bat/searching.sh` to search words in files.

C: Checking indexing and search with Melingo Tokenizer plug-in

1. Run `Tok_Mappings.bat/tok_mappings.sh` in order to setting and mapping new index
2. Run `Indexing.bat/indexing.sh` to index the example files.
3. Run `Searching.bat/searching.sh` to search words in files.

3.2. Definitions

Definition for the Melingo **Analyzers** in Elasticsearch:

- Hebrew Search Analyzer: `"melingo_hebrew_query"`
- Hebrew Indexing Analyzer: `"melingo_hebrew_index"`
- Arabic Search Analyzer: `"melingo_arabic_query"`
- Arabic Indexing Analyzer: `"melingo_arabic_index"`

Definition for the Melingo **Tokenizers** in Elasticsearch:

- Hebrew Search Tokenizer: `"melingo_hebrew_tokenizer_query"`
- Hebrew Indexing Tokenizer: `"melingo_hebrew_tokenizer_index"`
- Arabic Search Tokenizer: `"melingo_arabic_tokenizer_query"`
- Arabic Indexing Tokenizer: `"melingo_arabic_tokenizer_index"`

The following terms, may be encountered 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. <i>dll</i> file), as a measure against the situation of " <i>dll</i> 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/Wordbreaker	A tool that breaks text up into tokens.