Double Metaphone Search

Contents of this Page

Double Metaphone Implementation Details

Search with the Lucene query syntax, using a 'sounds like' algorithm.

Algorithm:

The Double Metaphone search has the following characteristics:

- This search is case in-sensitive.
- It searches on the double metaphone property value.
- Parsing is done with the following analyzer:
  - dm_propertyValue - Uses our custom double metaphone analyzer. This has the following filters:
    - LowerCaseFilter - for setting to lowercase
    - StopFilter - to remove stop words (the, a, etc.) from the search
    - DoubleMetaphoneFilter - for testing double metaphone sounds

Example of use:

The following examples are based on the Automobiles coding scheme.

Example 1:

Search string: Automobeel

Lucene query: dm_propertyValue:ATMP

Result: 1 result

- entity code: A0001
- entity description: Automobile

Example 2:
Search string: kar truk

Lucene query: +dm_propertyValue:KR +dm_propertyValue:TRK

Result: 0 results

- entity code:
- entity description:

Associated JUnits:

Junits can be found here: https://github.com/lexevs/lexevs/blob/master/lbTest/src/test/java/org/LexGrid/LexBIG/Impl/function/query/lucene/searchAlgorithms/DoubleMetaphoneSearch.java