****

Important Notes

* 1. It’s important to note that full text index searching is useful when searching for dictionary words.
  2. This would not include codes or prefixes etc. which are already configured optimally.
  3. “Quick” searches are defined for convenience to the user and not necessarily quick as the predicates span multiple fields that can include prefix, middle, title etc. Only those fields that are “dictionary based” for example title will be configured to use full text indexing.
  4. If you know, as a user, that you want to search for example a title, description or name etc. (using index searching) or even when you know your search is a code or prefix etc. the “Extended” searches are provided for this purpose and are the preferred route for optimal searching.
  5. eQL will support two predicates CONTAINS and FREETEXT for Text Indexes.
  6. When a system is delivered if you look at the definition of most simple and extended searches you will see syntax of the form “Name CONTAINS @SearchString”.
  7. CONTAINS will strip out any wildcard reference “%” from the query unless it’s at the end of the word. This is because it’s a dictionary search on specific words not the entire field entry. This is an important distinction that is not usually understood. Consider the text “The Quick Brown Fox” using a dictionary search for Brown will find the row containing this word. If a table (not full text index) search was needed the word would have to be wrapped with wild card “%BROWN%”. The only limitation then is if the first set of characters of a word are not known for example searching for %ROWN% would find words such as BROWN and FROWN is not applicable using full text indexes. In a normal enterprise environment this is not needed and is an extremely inefficient use of the system.
  8. Note If Full Text Indexing is not enabled eQL will automatically convert the syntax to use the LIKE predicate instead resulting in a much slower search.

## Full text searching using CONTAINS

When the CONTAINS predicate is identified by the eQL compiler and text indexes are turned on, the eQL compiler will leverage these indexes. For the examples provided we’ll use the Field ‘Title’ to demonstrate the behaviour but all other text index fields will behave the same way.

* + Entering WARRINGTON OR JUNCTION will result in all rows with either WARRINGTON OR JUNCTION in the title.
  + Entering WARRINGTON AND JUNCTION (or just WARRINGTON JUNCTION) will result in all rows with WARRINGTON AND JUNCTION in the title.
  + If you need to search for adjoining words then wrap the text in quotes for example “WARRINGTON JUNCTION” will return all rows with WARRINGTON JUNCTION in the title, alternatively use a word breaker such as ‘-‘ for the title e.g. WARRINGTON-JUNCTION.
  + Wild cards are also supported only at the end of the string for example the string WARRINGTON AND JUNCT% will result in all rows with WARRINGTON AND words starting with JUNCT in the title. Leading wildcards are ignored others are converted to an asterisk.
  + WARRINGTON AND NOT JUNCTION would return all rows with WARRINGTON but not JUNCTION in the title.

## Full Text searching using FREETEXT

In some rare cases searches may be configured to use the FREETEXT predicate instead of CONTAINS. This behaves similarly to CONTAINS with the following differences:

* + Entering just WARRINGTON JUNCTION will result in all rows with WARRINGTON OR JUNCTION in the title.
  + FREETEXT is basically a fuzzy text search based on the best matching algorithm defined by the Okapi BM-25 (<https://en.wikipedia.org/wiki/Okapi_BM25>). The intext is for the FREETEXT predicate to match the meaning and not just the exact wording resulting in less precise results.
  + As an example, if three different documents contained the words, “drive”, “driven” and “drove” in the title the search using FREETEXT would return all three documents. However, CONTAINS ‘Drive’ would return just the document with Drive in the title.
  + All wild cards are ignored when using FREETEXT.

# References

* <https://en.wikipedia.org/wiki/Okapi_BM25>