# TIBCO EBX® Information Search Add-on Documentation

*Version 2.4.22*
*October 2022*

TIBCO®

## *Important Information*

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

ANY SOFTWARE ITEM IDENTIFIED AS THIRD PARTY LIBRARY IS AVAILABLE UNDER SEPARATE SOFTWARE LICENSE TERMS AND IS NOT PART OF A TIBCO PRODUCT. AS SUCH, THESE SOFTWARE ITEMS ARE NOT COVERED BY THE TERMS OF YOUR AGREEMENT WITH TIBCO, INCLUDING ANY TERMS CONCERNING SUPPORT, MAINTENANCE, WARRANTIES, AND INDEMNITIES. DOWNLOAD AND USE OF THESE ITEMS IS SOLELY AT YOUR OWN DISCRETION AND SUBJECT TO THE LICENSE TERMS APPLICABLE TO THEM. BY PROCEEDING TO DOWNLOAD, INSTALL OR USE ANY OF THESE ITEMS, YOU ACKNOWLEDGE THE FOREGOING DISTINCTIONS BETWEEN THESE ITEMS AND TIBCO PRODUCTS.



TIBCO and TIBCO EBX are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. Please see the readme.txt file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (*https://www.tibco.com/patents*) for details.

# Table of contents

## User Guide

# Developer Guide

# Reference Guide

# Release Notes

# User Guide

CHAPTER 1

# About the add-on

The TIBCO EBX® Information Search Add-on allows you to find and retrieve data contained in your MDM repository. You can configure search queries to execute on one field in a single table, or multiple fields spanning multiple tables. The global search feature enables search scope to encompass multiple dataspaces and datasets.

Search functionality resembles that of a web search engine and enables the add-on to efficiently process a large volume of information. Search queries use fuzzy execution, which allows information retrieval even if the search term does not exactly match the data being searched. The values returned depend on how you configure the add-on. For example, the search term John can return results such as Jone, Johnson or Cohn. You can fine tune the search to avoid unwanted results. Additionally, you can choose a search policy based on phonetics, or distance to best meet requirements for the type of data being queried.

An administrator must configure the add-on before running any queries. In fact, the options to perform a global search and search via the **Actions** menu will not display until a configuration exists. See Configuration overview and examples [p 12] to get started with basic configuration options.

**See also**

*The Configuration domain [p 62]*

*Configuration overview and examples [p 12]*

*Running searches [p 45]*

# Configuring

CHAPTER 2

# Configuration overview and examples

This chapter contains the following topics:



## 2.1 Configuration overview

Administrators can configure the add-on to search data local to where the search was executed, or to search data spanning multiple datasets and dataspaces. The following two configuration examples are provided to get you started:

- Creating a basic configuration [p 13]
- Configuring global search [p 15]

As described below, basic configuration entails:

- setting search criteria for, and registering a data model. This includes setting options such as search sensitivity and permissions. The criteria you specify applies to any table included in this data model that users can search.
- determining which tables users can search. Because of the model-driven configuration, you only need to setup a table one time. Users can run a search on the table regardless of its dataset location.
- indicating the fields that the add-on searches when users run queries. This includes specifying the field, and search algorithm applied to queries on this field. Each algorithm processes data differently. The ability to apply algorithms to individual fields, based on that field's data type, helps ensure that searches return the most relevant results.

In addition to the previously mentioned options, you can also:

- Create synonyms used to return results when a query does not find an exact match. [p 23]
- Expose add-on functionality as a webservice. [p 41]
- Configure your own search algorithm. [p 17]
- Set values to be excluded from a search. [p 29]

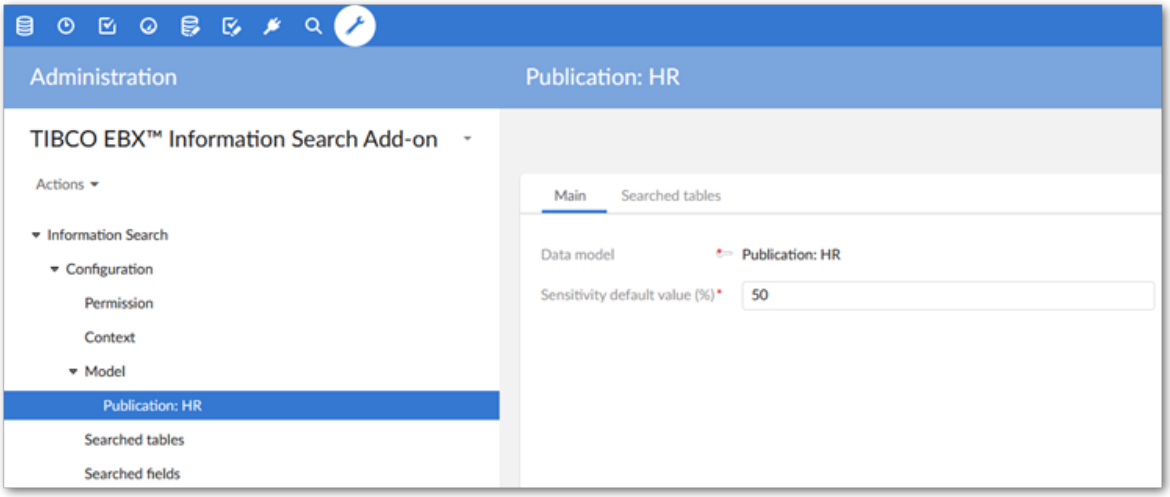
- Set up search history. [p 31]

## 2.2 Creating a basic configuration

The sample presented in this section demonstrates a configuration that enables search on a single field in a table. Use the same steps to enable search for multiple tables and fields. If you have questions regarding an option not discussed in the instructions, click the property's tooltip to obtain more information.

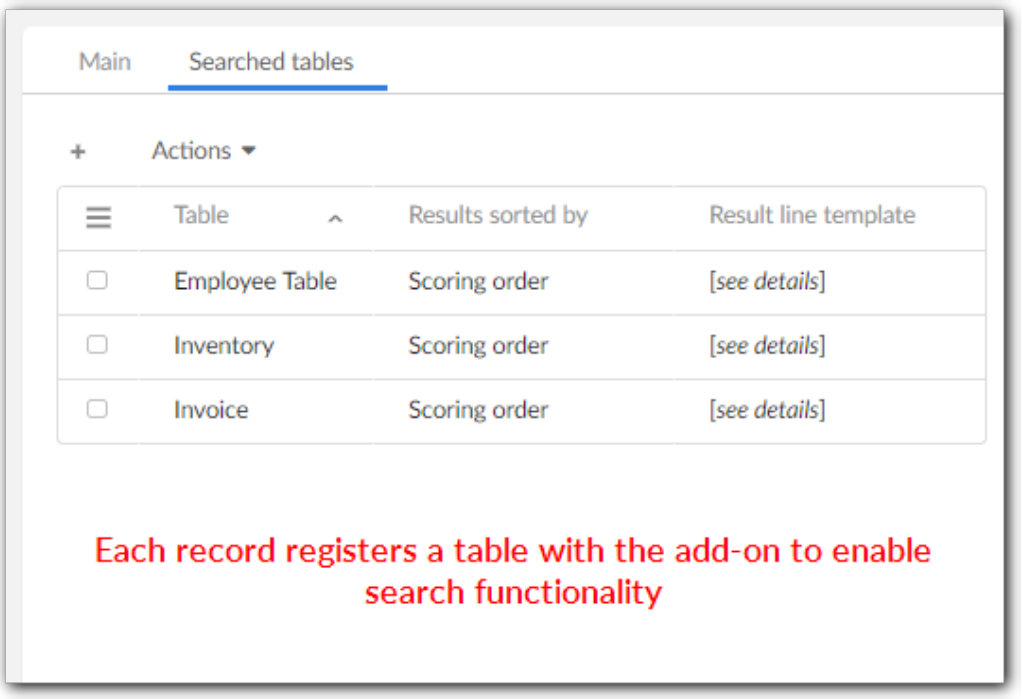
To enable EBX® Information Search Add-on functionality:

1. Navigate to *Administration* > *Data quality & analytics* > *TIBCO EBX® Information Search Add-on* > *Information Search* > *Configuration* > *Model*. Create a new record.
2. Select the data model you want to register. At this stage you can also specify the default search sensitivity that the add-on applies to all tables included in this data model. This value plays an important role in returning the most relevant values. When set too high, potentially correct results may be overlooked. On the other hand, a search may return irrelevant results when the sensitivity is too low. As a recommended practice you can test the setting to ensure the add-on returns the best results (Search sensitivity [p 15]).




3. Save, but do not close your configuration.

4. Specify the table on which you want to enable search by selecting the **Searched tables** tab and creating a new record.




5. After adding a table, save, but do not close the record.
6. Select the **Searched fields** tab and create a new record to indicate the field you want the add-on to search. Note that there are some special considerations to take into account when configuring search on a foreign key field. See Behavior of foreign key labels [p 16] for more information.

> **Attention**
> When configured for a foreign key field that is a numeric data type, no results are returned if a keyword includes characters such as decimal points and commas.

   You can specify the following criteria for searched fields:

   - Chose a primary, and optionally, a secondary search algorithm. Depending on the field's data type, you can choose from pre-existing phonetic, or distance based algorithms. Alternatively, you can create your own algorithm. See Working with algorithms [p 17] for more information.
   - Use a Java class to filter the search. If you have created any filtered values in the add-on's configuration, use the default class (`com.orchestranetworks.addon.tese.SearchValueFilter`) to filter out values. Alternatively, you can create and specify your own custom class to filter values. See Excluding values from searches [p 29] for more information.
   - Allow the add-on to return synonyms when an exact match is not found. You can limit available synonyms to a single group, or enable multiple groups. See Expanding searches with synonyms [p 23] for more information.

7. After saving and closing the field record, you can continue to configure additional fields on this table. When finished, save and close out of the configuration.

Users that can access datasets and tables configured to use the add-on can run the search service from either **Actions** menu. At any time you can use the **Model**, **Searched tables**, **Searched fields**, and other configuration groups to add, or alter configuration information.

**See also**

*Excluding values from search results* [p 29]

*Working with algorithms* [p 17]

*Expanding search with synonyms* [p 23]

### *Search sensitivity*

Generally, when you lower the sensitivity, searches return more results. For example, when searching for "Alice" with a setting of 70, a search may return "Alice". After lowering the sensitivity to 60, the search would still include the previous result, but may also include "Aliz" and "Alric".

## 2.3 Configuring global search

When enabling global search, you associate a user profile with one, or more *contexts* that point to tables. When a user profile performs a global search, it will include all tables from the linked contexts. To include a table in a context, register the table's containing model and the table with the add-on, and configure search options for the desired fields. These steps are demonstrated in the Creating a basic configuration [p 13] heading above. The example below uses previously configured tables.

To enable global search:

1. Navigate to *Administration* > *Data quality & analytics* > *TIBCO EBX® Information Search Add-on* > *Information Search* > *Configuration*.
2. Open the **Permission** table and create a new record.
3. Select the desired profile from the drop-down list, then add a context. If no context exists:
   - Open the **Context** drop-down and select **+ Create**, or create a new record in the **Context** table.
   - Supply a code and optionally a custom title.
   - Use the **Dataspace**, **Dataset**, and **Tables** fields to include tables in this context. Remember, you can only add tables that have been previously configured in the add-on.
4. After adding all desired contexts, save and close. The global search icon will now be available for the user, or users included in the role, registered in the **Permission** table.

**Attention**

To use the global search feature for any dataspace, it must remain enabled in the **Reference** dataspace permissions.

# 2.4 **Behavior of foreign key labels**

You may want to take the following information into account when enabling search on foreign key (FK) fields:

- If a search field is a FK field *and* has a programmatic label definition, the add-on considers it a normal search field.
- If a search field is a FK field, that field's (the source field) search query is cloned to all fields in the target table that participate in providing the record's label. Then, the target table's related records are processed as follows:
  - The related record's score in the target table is the average score from all fields, there is no weight here.
  - If the source field is multi-valued, then there are multiple related records on target table. All records are processed and the search returns the highest score.
- The score of a FK field also follows the synthesis label option.

CHAPTER 3

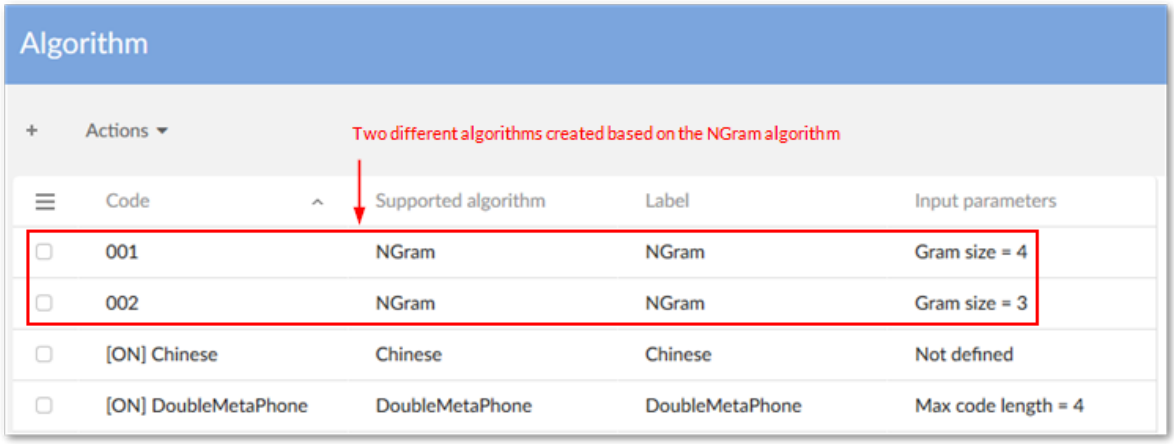
# Working with algorithms

This chapter contains the following topics:



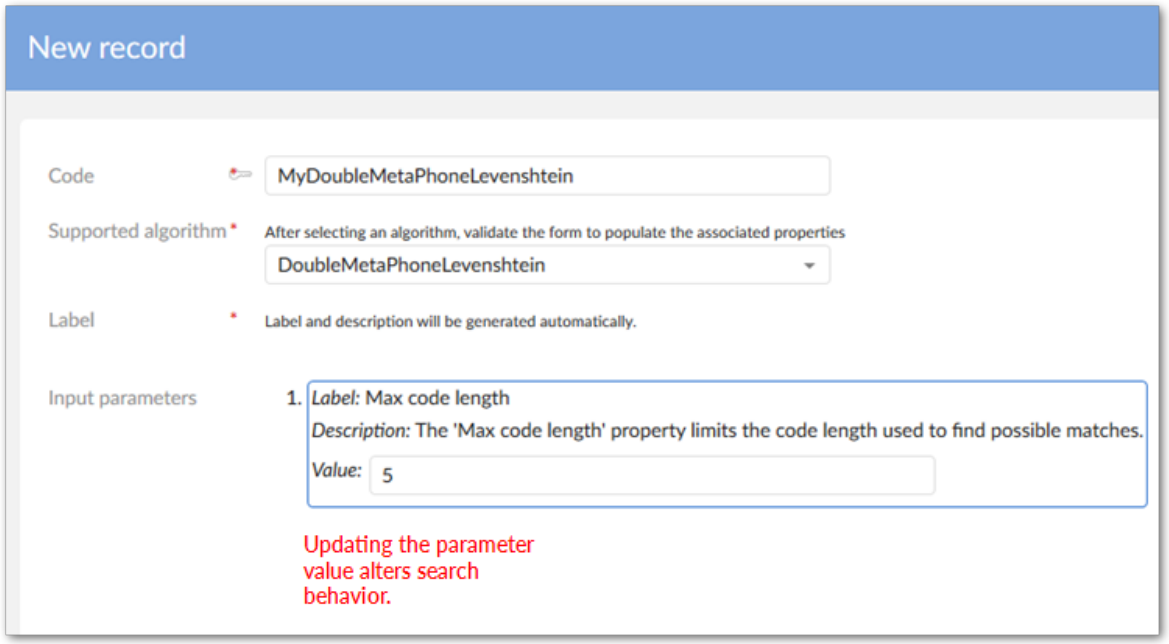
## 3.1 Customizing search algorithms

The add-on comes bundled with a set of ready-to-use algorithms. You can create a new algorithm based on one of these default algorithms and edit its parameters to customize behavior. As shown below, you could use the **N-Gram** algorithm to create two algorithm definitions with different values for the input parameters.




When you create a new algorithm, fill in the **Code** field, choose a **Supported algorithm**, and click the **Save** button. The label and description automatically update, and a list of default parameters displays in the **Input parameters** section. The parameters that display depend on the selected algorithm. After you finish customizing parameter values for the new algorithm, click **Save and close**. Your newly created algorithm displays in the **Algorithm** table and is ready to use.

The following image shows a newly created algorithm configuration based on the DoubleMetaphoneLevenstein pre-defined algorithm with its **Max code length** value updated to 5.



The table below lists all algorithms from which you can customize parameters in order to create new algorithms.

| Algorithms | Default parameters | Parameter configuration |
|---|---|---|
| **DoubleMetaphone** | Max code length = 4 | This phonetic algorithm works best on short strings, such as proper names. It is especially adept at returning words or names whose actual pronunciation may be different than the search text entered.<br>The **Max code length** property limits the code length used to find possible matches. When you enter a search string, the algorithm encodes it as a key and returns words with matching keys. You should set this property to a value that reflects the length of text being searched. For example:<br>If you specify a value of 4, the algorithm encodes the three words "cricket", criket" and cricketgame" as "KRKT". The algorithm considers the three words a match. If you changed the value to 8, "cricket" and "criket" are still encoded as "KRKT". However, it encodes "cricketgame" as "KRKTKM". In this case, "cricketgame" no longer matches.<br>Note that this algorithm cannot be used to search numeric, date/time, or special character formats. Also, due to the way the algorithm processes phonetic structures, a search for "www" returns no result. |
| **DoubleMetaphone Levenshtein** | Max code length = 4 | Being a phonetic algorithm, Double Metaphone may fail to match misspelled words when the misspelling substantially alters the phonetic structure of a word.<br>The Double Metaphone Levenshtein algorithm can compute distance between two long strings, but at the cost to compute it, which is roughly proportional to the product of the two string lengths.<br>So, a combination of these algorithms reduces their limitations. Levenshtein may find similarity between encoded strings, and the length of encoded strings is limited by Double Metaphone. |
| **NGram** | Item size (n) = 2 | This algorithm partitions search criteria into subsets of a specified length called NGrams. You set this length using the **Gram size** property. For example, if you set this property to a value of 3, the algorithm splits the word PHASED into the following N-Grams: PHA, HAS, ASE and SED. PHASED is then added to the lists of words containing those N-Grams.<br>Keep in mind that if you set the size too small, the algorithm may not capture important differences and return too many terms. If the size is too large, the opposite is true and may result in few returned results. Therefore, when used for names, a value of 3 or 4 is recommended. For phone numbers, a value of 7. |
| **JaroWinkler** | threshold = 0.7 (a condition to add **Winkler** distance or not. Value is from 0 to 1) | This algorithm works best on short strings, such as proper names. It tallies the number of characters in common and places a higher emphasis on differences at the start of the string. Therefore, the lower you set the **Threshold** parameter, the more impact differences at the beginning of strings have. **Threshold** parameter values should be from 0.0 to 1.0. |
| **FuzzyFullText** | Similarity = 0.7<br>Prefix length = 0 | This algorithm works best for general strings like those contained in descriptions. This algorithm finds a similar, or fuzzy, match of the keyword text entered. |

| Algorithms | Default parameters | Parameter configuration |
|---|---|---|
| | | The **Similarity** parameter determines how similar results have to be before they are returned. The higher you set the value, the fewer results and vice versa.<br>The **Prefix length** parameter specifies that a number of characters from the beginning of the keyword must exactly match data being searched in order to return a result. For example, if you set the value to 2 and use the keyword "Automotive", the algorithm only considers words that begin with "au" as potential matches. |
| **Chinese** | | This algorithm performs search on Chinese text. |
| **FuzzyJapanese** | Similarity = 0.7<br>Prefix length = 0 | This algorithm performs a search on Japanese text and finds a similar, or "fuzzy" match. This algorithm allows you to use the following character types or any combination thereof: Kanji, Katakana and Hiragana.<br>The **Similarity** parameter defines a value between 0 and 1, which is used to set the required similarity between the query terms and the matching terms. The similarity level is calculated based on the Levenshtein algorithm. For example: For a similarity of 0.5, a term of the same length as the query term is considered similar to the query term if the edit distance between both terms is less than length(term)*0.5. |
| **Korean** | | This algorithm performs search on Korean text. |
| **FuzzyKorean** | Similarity = 0.7<br>Prefix length = 0 | This algorithm performs a search on Korean text and finds a similar, or "fuzzy" match.<br>The **Similarity** parameter defines a value between 0 and 1, which is used to set the required similarity between the query terms and the matching terms. The similarity level is calculated based on the Levenshtein algorithm. For example: For a similarity of 0.5, a term of the same length as the query term is considered similar to the query term if the edit distance between both terms is less than length(term)*0.5. |
| **SearchDate** | Threshold = 5 | This algorithm allows you to search on fields with date or, date-time data types. In order for a date to match, it must be in the range specified by the search input plus/minus the number of days specified in the **Threshold** parameter. The closer the search input is to the data being searched, the higher the score.<br>When you increase the **Threshold** parameter's value, the score decreases: Score = 100-(distance*100/threshold) |
| **SearchNumber** | Threshold = 5 | This algorithm allows you to search on fields with a numeric data type. In order for a number to match, it must be in the range specified by the search input plus/minus the value set in the **Threshold** parameter. The closer the search input is to the numbers being searched, the higher the score.<br>In order for a number to match, it must be in the range specified by the search input plus or minus the value set in the **Threshold** parameter. If the **Threshold** value increases, the score decreases. Score = 100-(distance*100/threshold). |

## 3.2 Algorithms for number and date/time data types

You can search in fields whose data types are integer, decimal, date, or date time. To search fields with date data types, you must configure the desired fields to use the **Search Date** algorithm. To search fields with integer, or decimal data types, you must use the **Search Number** algorithm.

After you have configured fields to use the **Search Date** algorithm, to search for Date, input the Date string in 'DD/MM/YYYY' format for French and 'MM/DD/YYYY' format for other languages.

## 3.3 Japanese search algorithms

The Japanese and Fuzzy Japanese search algorithms allow you to retrieve results written in the Japanese language by using one, or a combination of three Japanese alphabet charts: Kanji, Hiragana, and Katakana. The Fuzzy Japanese algorithm works best for general strings like those contained in descriptions by finding a similar, or fuzzy, match of the keyword entered.

To apply the Japanese search algorithm, you navigate to the **Searched fields** table, located under *Administration > Data quality & analytics > TIBCO EBX® Information Search Add-on > Information Search > Configuration*, then specify the Japanese search option on the **Primary search algorithm** field.

Once you have completed the configuration, you can execute a search as usual.

CHAPTER 4

# Expanding search with synonyms

This chapter contains the following topics:

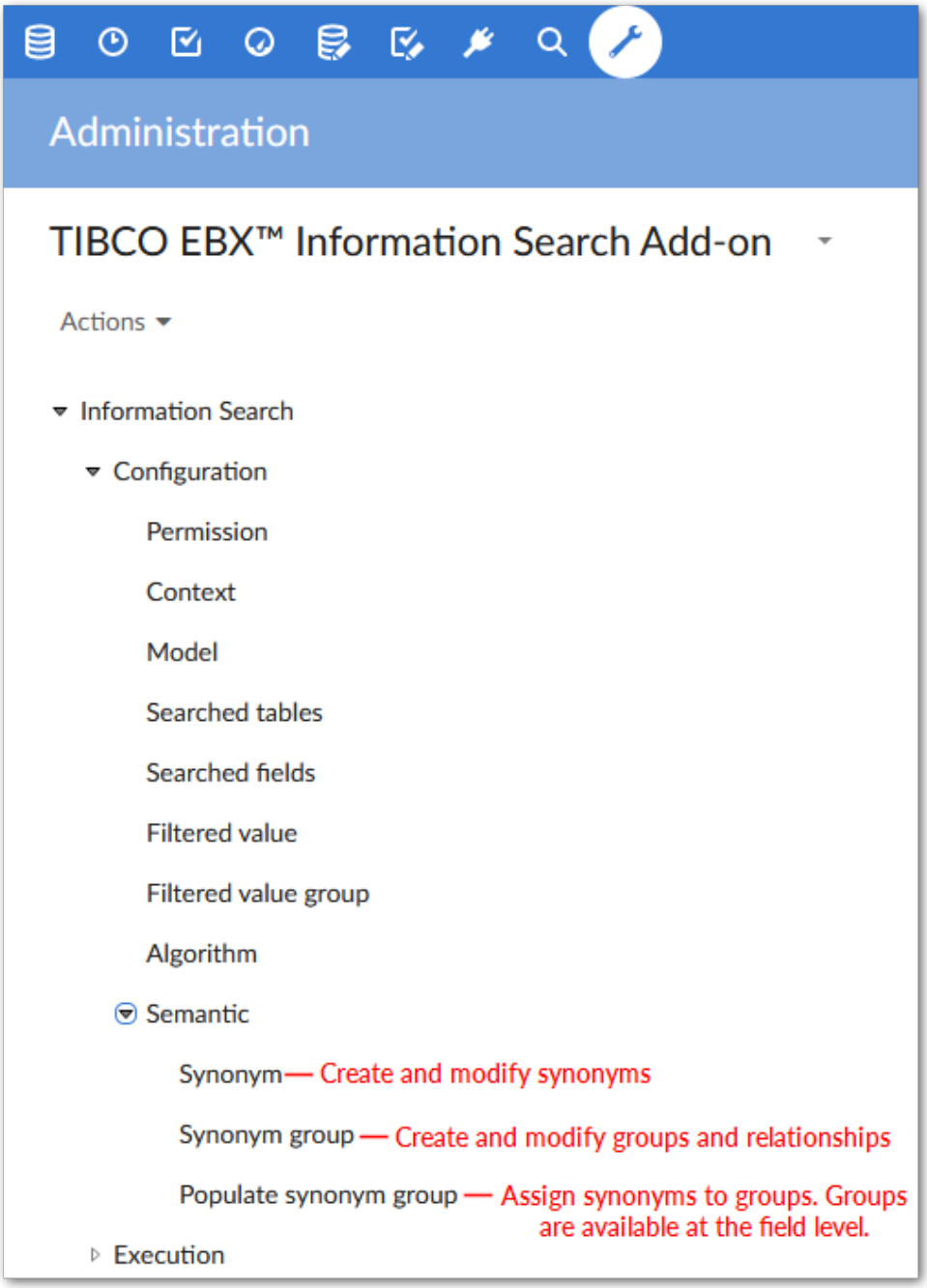


## 4.1 Overview

You can create and modify groups of synonyms to increase search flexibility. If the add-on does not find a match on the initial search value, it turns to the values contained in the synonym group to find a possible match. Of course, depending on business requirements, you need not limit this functionality to literal synonyms. For example, you could group common abbreviations.

# 4.2 **Managing synonyms**

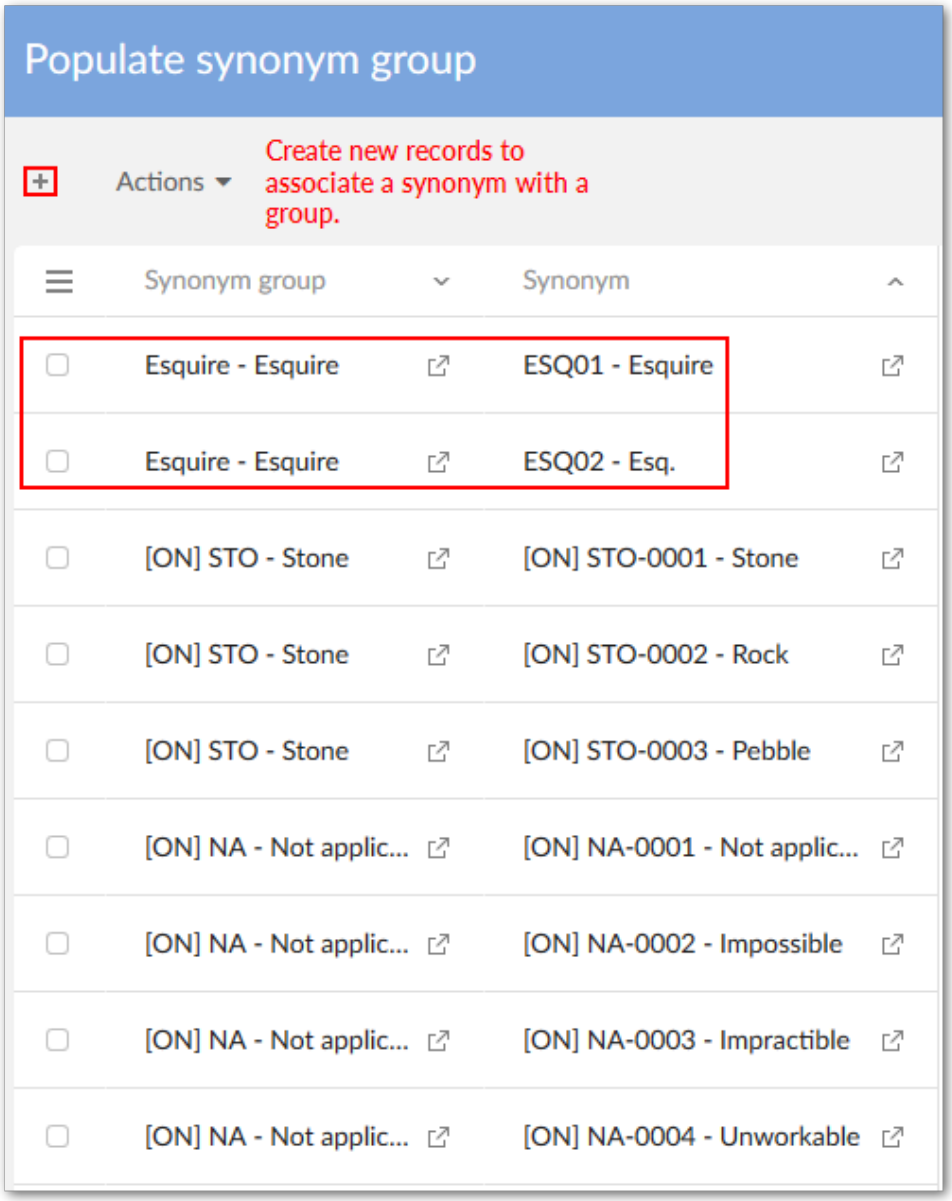
The following image highlights the semantic group where you can work with synonyms.



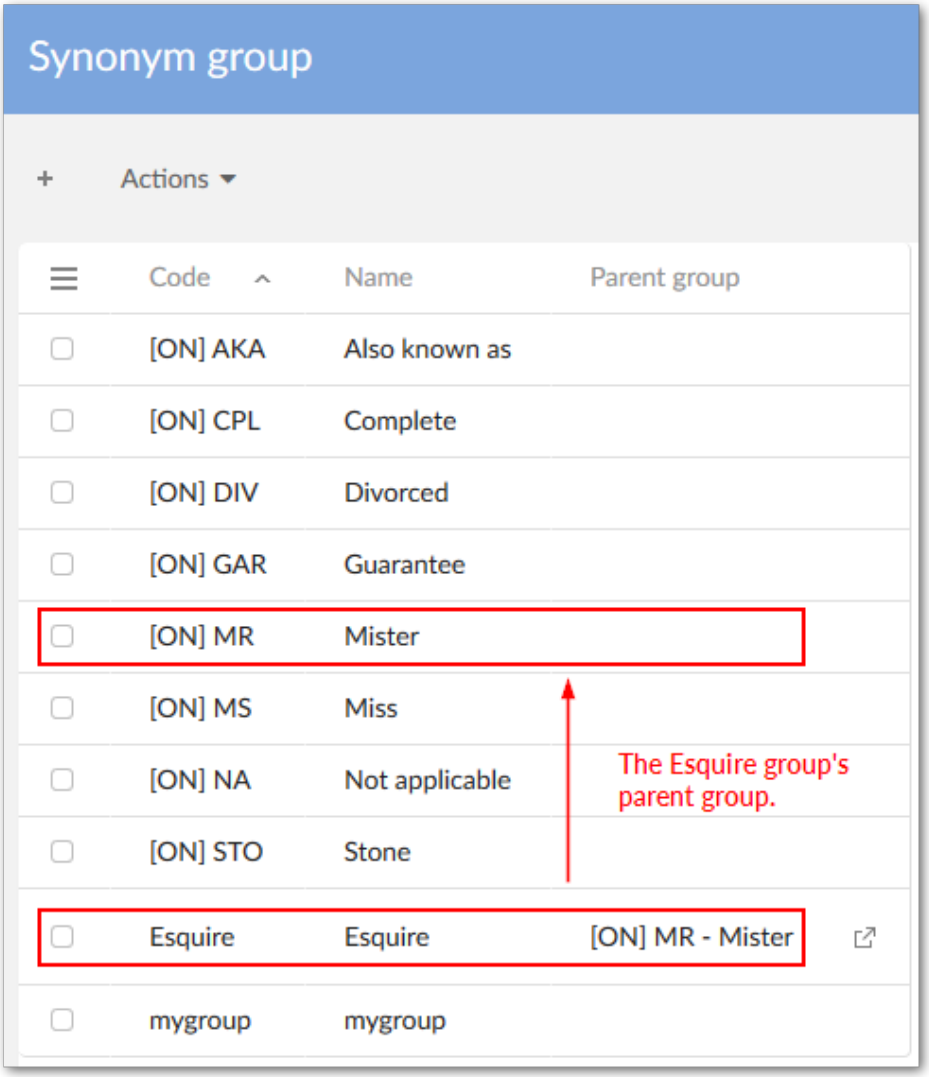

To create synonyms and add them to groups:

1. Navigate to *Administration* > *Data quality & analytics* > *TIBCO EBX® Information Search Add-on* > *Information Search* > *Configuration* > *Semantic*.
2. Create synonyms in the **Synonym** table.
3. Create a **Synonym group** to organize your synonyms.

4. Use the **Populate synonym group** table to assign synonyms to groups by adding each desire synonym.



5. Optionally, create a relationship between synonym groups. When enabling search, you can specify that the add-on check for matches in related groups. As shown below, use the **Synonym group** table to create the relationships.




6. Be sure to save after completing your changes.

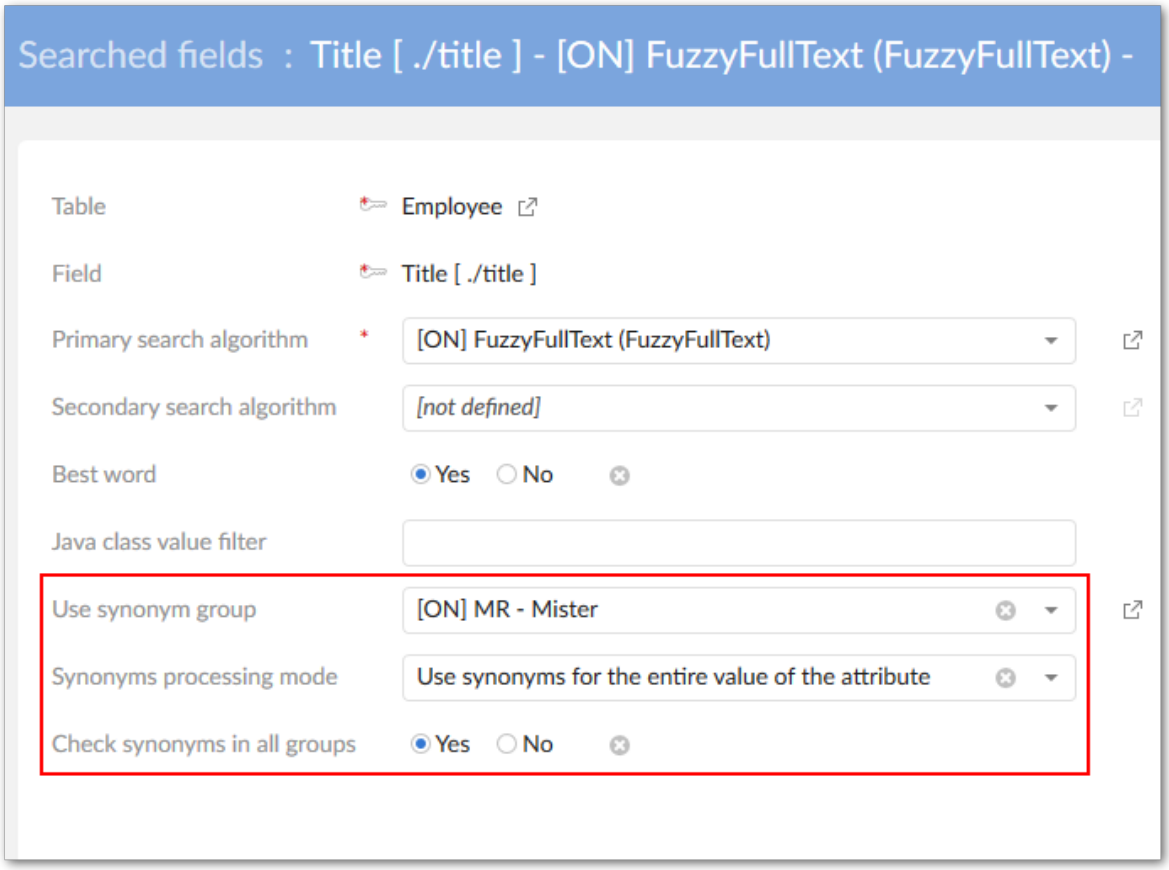
# 4.3 Enabling search with synonyms

You can enable synonym search from a **Searched field** configuration as follows:

1. In the **Searched fields** table, select a synonym group from the drop-down menu of the **Use synonym group** property.
   - Select a synonym group from the **Use synonym group** drop-down menu.
   - Use the **Synonyms processing mode** to tell the add-on how to evaluate search terms against configured synonyms. The available options are:
     - **Use synonyms for the entire value of the attribute:** This option compares synonyms with the entire word or phrase being searched. Take fore example the following group of synonyms: "Mr", "Mister", and "Sir". Taken as a whole, the search phrase "Mr. Bob

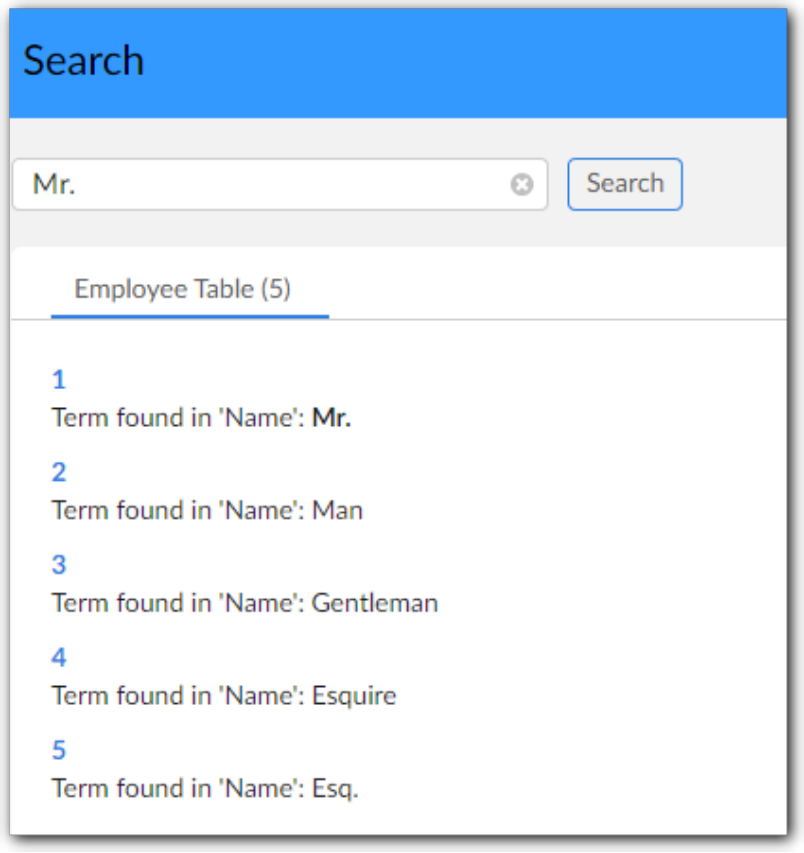
Smith" may not match close enough to trigger results. However, if the search term was "Sir", it could return the record for "Sir Bob Smith".

- **Use synonyms for every word of the attribute:** This option compares synonyms with each part of a phrase being searched. Using the above example, the search phrase "Mr. Bob Smith" could return "Sir Bob Smith" as a result.




2. Activate the **Check synonyms in all groups** property (Optional). When enabled, the add-on looks for matching synonyms in related groups.

In the configuration example shown below, we created an Esquire group and associated it with the add-on's default [ON] MR group. When searching for Mr., the returned results include the synonyms defined in each group.



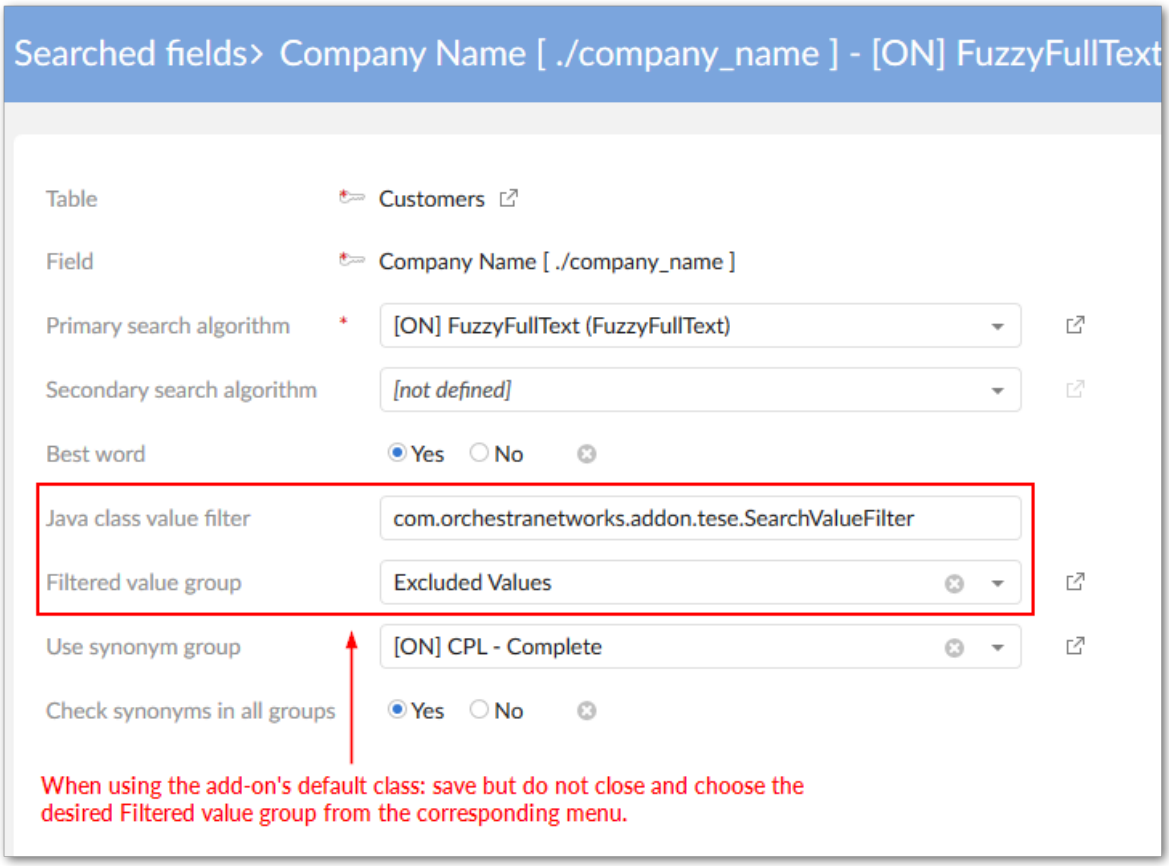
CHAPTER 5

# Excluding values from search results

The add-on allows you to omit values from search results on a field-by-field basis. Indicate the values you want to omit by:

1. Creating a list of the values to omit in one of the following ways:
   - Navigate to *Administration > Data quality & analytics > TIBCO EBX® Information Search Add-on > Information Search > Configuration* and use the **Filtered value** and **Filtered value group** tables to create values and group them together. Use a field's configuration settings to point to this group. When a user runs a search that includes this field, none of the results specified in the group display.
   - Create your own Java class that implements the `SearchTextNormalize` interface and lists the values you want to filter.
2. Open a field's configuration settings and in the **Value filter by Java class** enter:
   - `com.orchestranetworks.addon.tese.SearchValueFilter` (if you used the add-on to create values and groups)
   - The name of your custom class. When using this option, you can save and close out of the configuration.
3. Save, but do not close the configuration to display the **Filtered value group** drop-down menu.

4. From the **Filtered value group** menu, select the group of values you want to exclude from this field's search results.




5. Save and close out of the configuration.

CHAPTER 6

# Setting user preferences

The add-on allows you to set user preferences that determine how much search history the add-on records and whether search history automatically displays. Navigate to *Administration* > *Data quality & analytics* > *TIBCO EBX® Information Search Add-on* > *Information Search* > *Execution* and use the options in the **User preference** table to configure preferences.

CHAPTER 7

# Customizing how results display

This chapter contains the following topics:



## 7.1 Overview

The add-on allows administrators to customize how search results display. You apply customization options to tables configured to use the add-on. When a user performs a global or dataset search, they will see the customized results in the **All** tab and when viewing individual table results.

Each search result can display up to five lines of information. You can determine the:

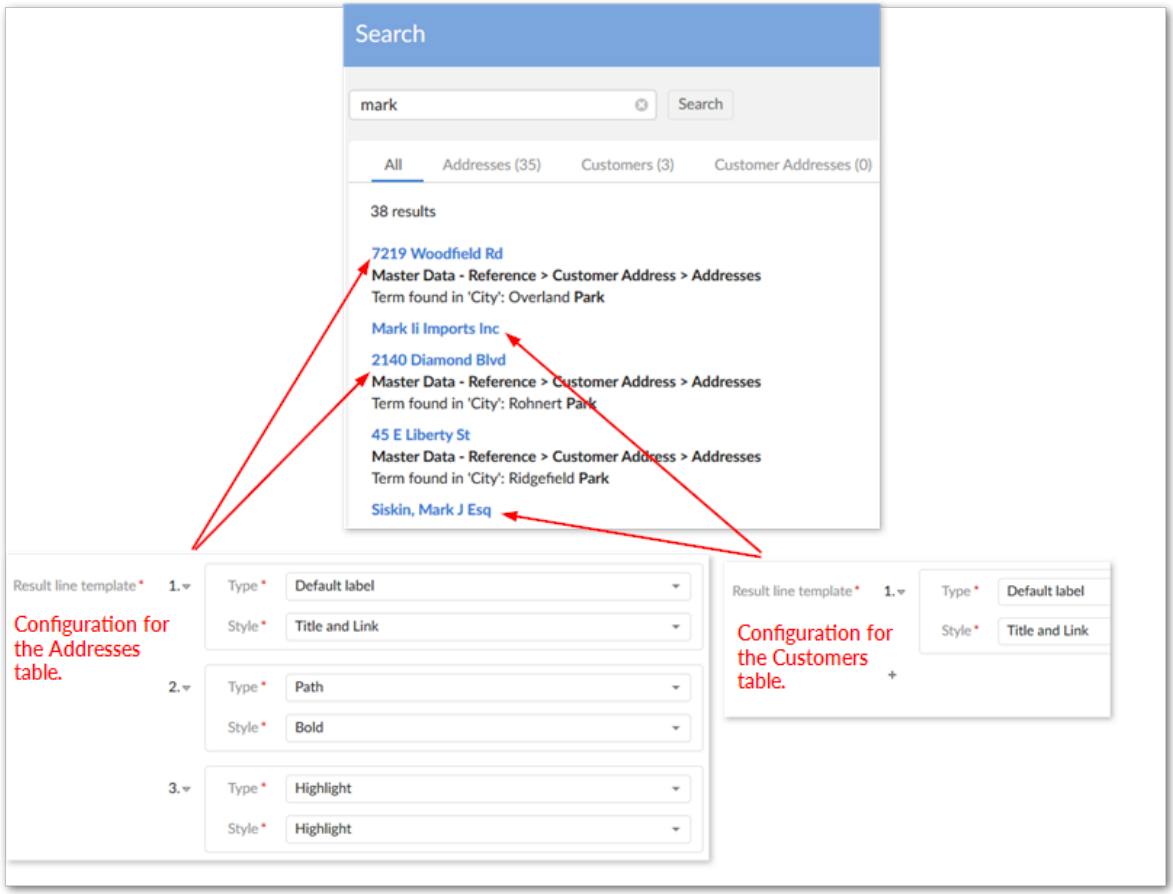
- order in which each result line displays.
- content of each result line, such as table path, record link, or location where the keyword was found.
- style applied to each result line.

## 7.2 Where and how to configure

Result configuration settings are located under *Administration > Data quality & analytics > TIBCO EBX® Information Search Add-on > Information Search > Configuration >Searched tables*. Use the options in the **Result line template** group to set result display. Each entry in this group corresponds to a search result line. When adding or editing an entry, you can configure the following:

- Type: The type of information displayed on the line.
- Label: This option is only available when you specify **Custom** in the **Type** field. You can select a combination of data model elements that make up the label.
- Style: The style you want to apply to this line.

The following example shows how configuration options affect the search results:



# 7.3 **Style quick-reference**

The following table describes the outcome of combining search result line styles and types:

| Style/Type | Result |
|---|---|
| **Title and Link**<br>**Default label** | Displays the default label in bold using the main color configured for EBX®. |
| **Title and Link**<br>**Custom label** | Displays the custom label defined in the **Label** field in bold using the main color configured for EBX®. A link to the record is also embedded. |
| **Title and Link**<br>**Path** | Displays the path in bold using the main color configured for EBX®. A link to the record is also embedded. |
| **Title and Link**<br>**Highlight** | Displays the search result location in bold using the main color configured for EBX®. A link to the record is also embedded. |
| **Title**<br>**Default label** | Displays the default label in bold using the main color configured for EBX®. |
| **Title**<br>**Custom label** | Displays the custom label defined in the **Label** field in bold using the main color configured for EBX®. |
| **Title**<br>**Path** | Displays the path in bold using the main color configured for EBX®. |
| **Title**<br>**Highlight** | Displays the search result location in bold using the main color configured for EBX®. |
| **Bold**<br>**Default label** | Displays the default label using normal text color in bold. |
| **Bold**<br>**Custom label** | Displays the custom label defined in the **Label** field using normal text color and in bold. |
| **Bold**<br>**Path** | Displays the path using regular text color and in bold. |
| **Bold**<br>**Highlight** | Displays the search result location using regular text color and in bold. |
| **Regular**<br>**Default label, Custom label, Path and Highlight** | Displays each line type using normal text color and no bold. |
| **Highlight** | Displays the label using normal text color, no bold, and the term found in bold. |

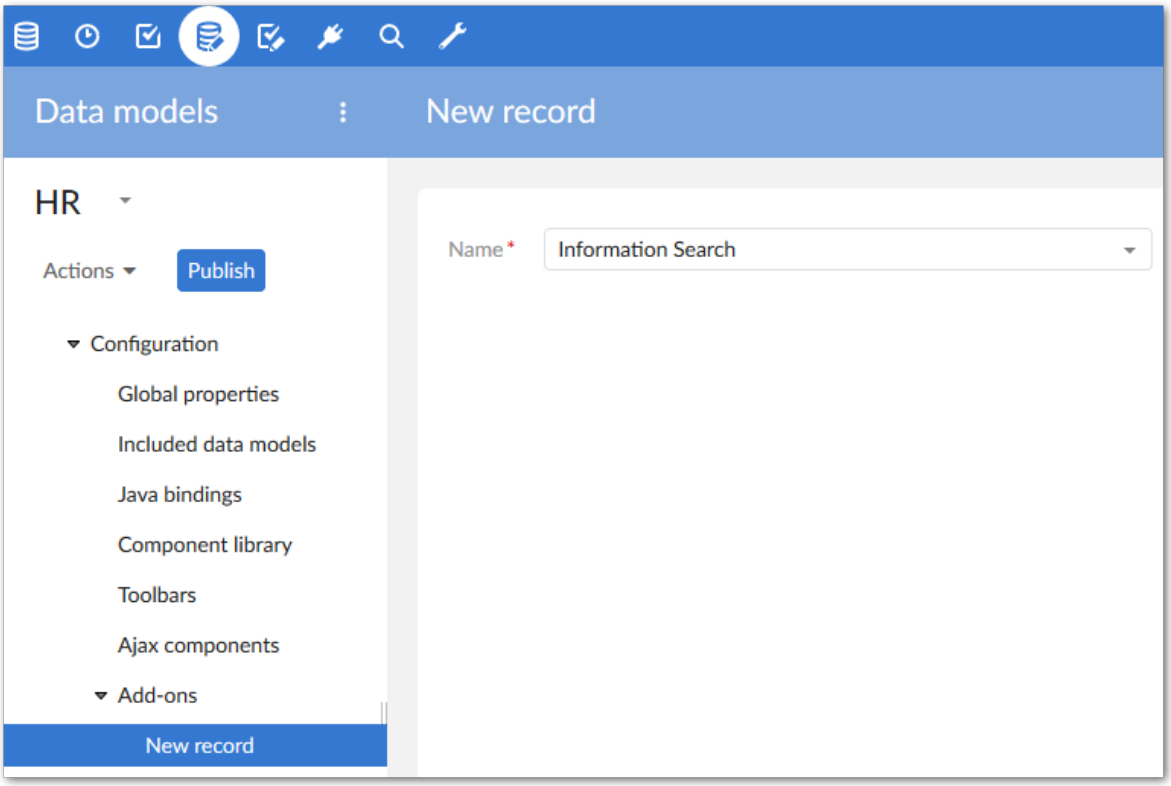
| Style/Type | Result |
| --- | --- |
| **Highlight** | |

CHAPTER 8

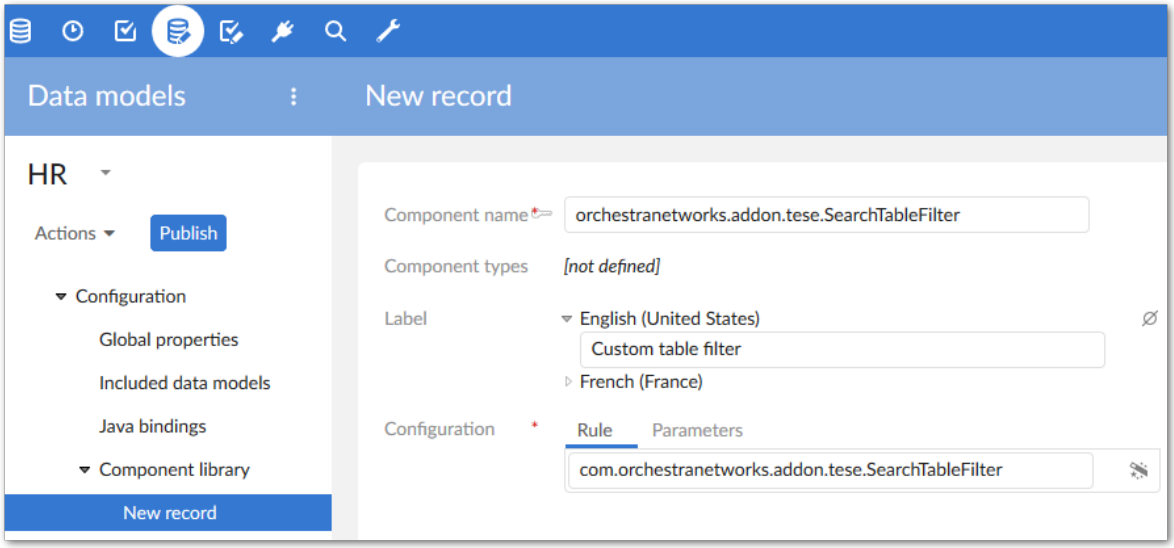
# Extending EBX® table filters

You can extend the standard EBX® table filter with EBX® Information Search Add-on capabilities. The panel content is based on the add-on's configuration. For the sake of simplicity, you can configure to automatically apply table filter when searching at table level. To activate that, under the desired data model, you must:

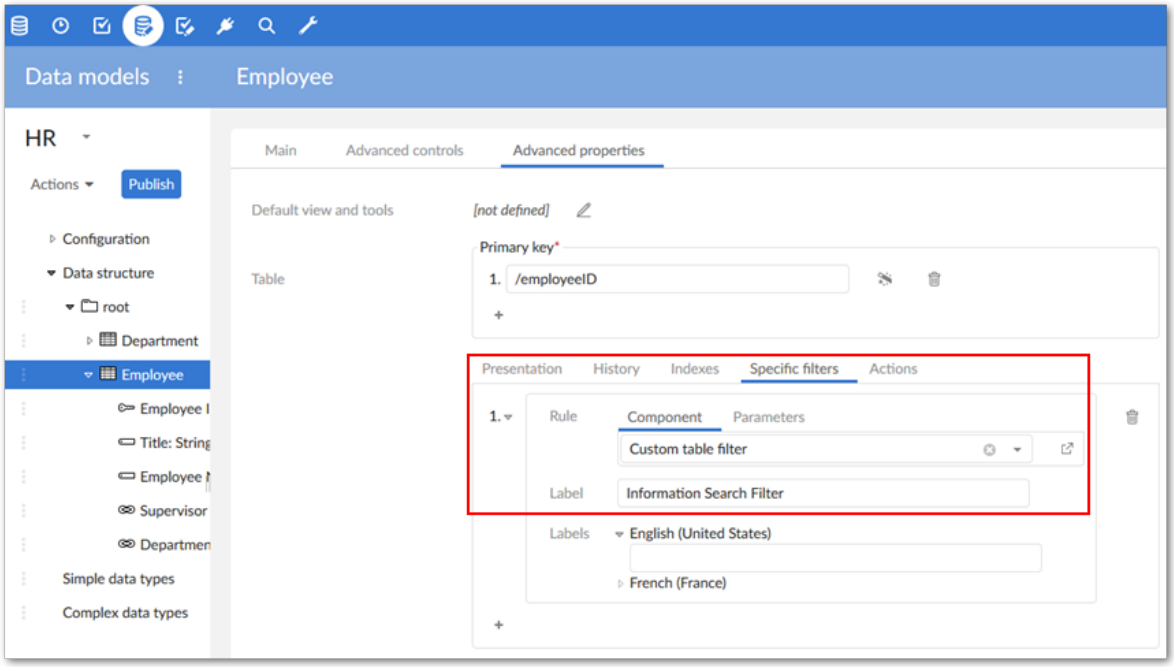
- Go to **Add-ons** table.
- On the **Name** field, select **Information Search**.
- Click **Save** to finish.



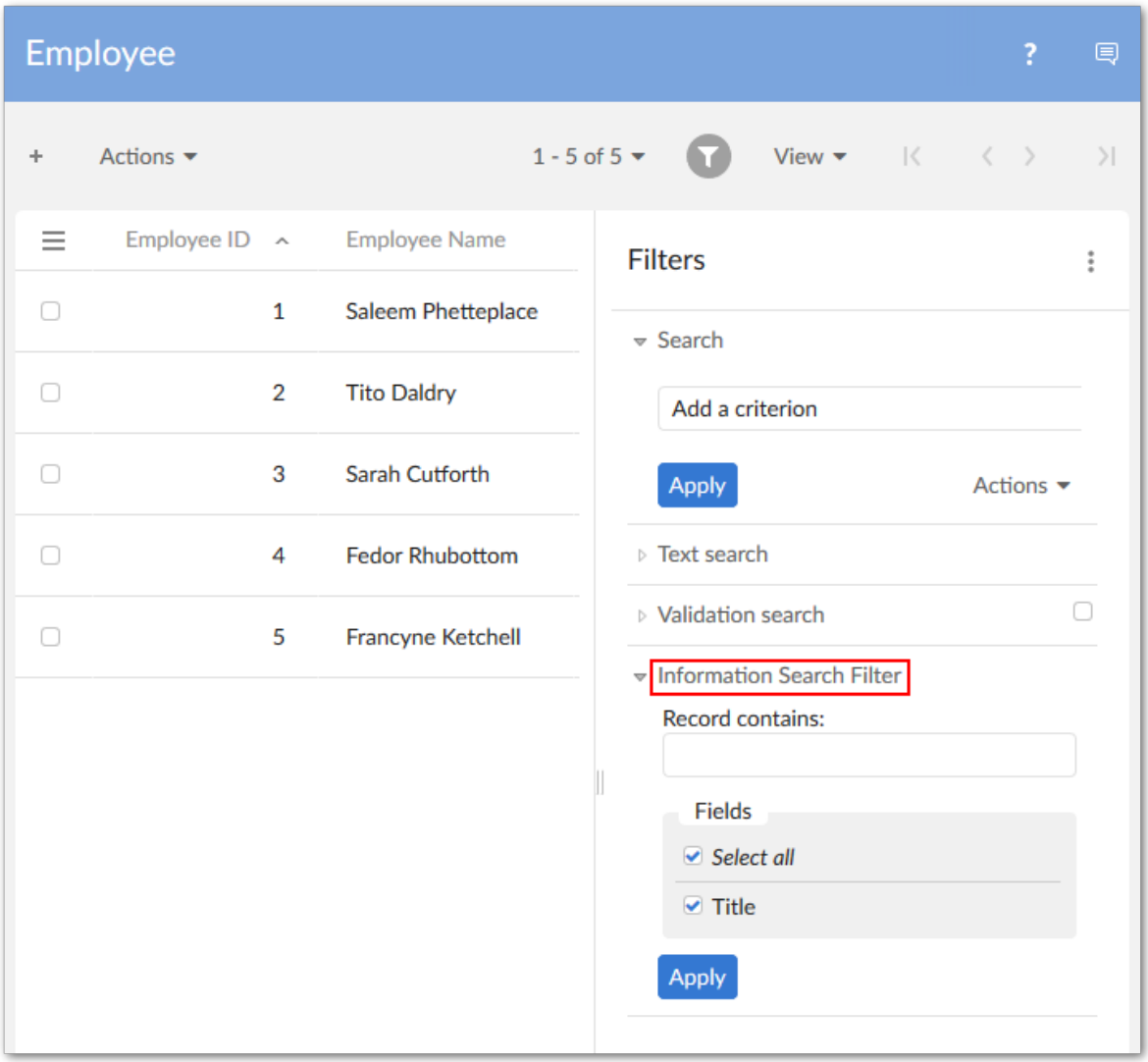
Alternatively, the custom filter can be declared into the data model by adding a the `com.orchestranetworks.addon.tese.SearchTableFilter` component library as described below:




You can create a label that identifies the EBX® Information Search Add-on option in the EBX® search panel. The following image shows a configuration that creates a **Filter by fields** label on the **Employee** table:



At execution time, the UI filter applied to the table is enriched by the add-on functionality and executes the configured primary and secondary search algorithms as illustrated below:




> **Note**
>
> If desired, you can also combine native filters such as validation or text search with add-on capabilities. Using the example above, you could set the **Validation search** option to show warnings along with searched text in the **Title** field.

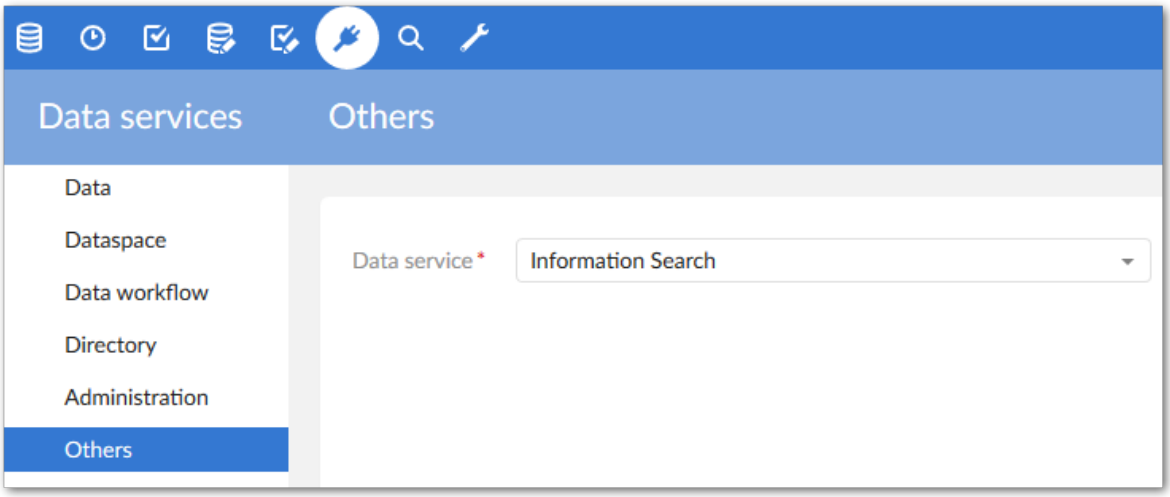
CHAPTER 9

# Using a web service

A web service can expose EBX® Information Search Add-on functionality using standard Simple Object Access Protocol (SOAP) request and response. If you input information to search into a SOAP request then you can receive the result in a SOAP response.

You can expose the fuzzy search feature via a web service. The following shows how to perform a search using a web service based on the add-on configuration:

- Go to **Others** service in the **Data services** tab choose **Information Search** and click **Next**:




- Select **Download WSDL**.

- The following is an example of the SOAP request for the downloaded WSDL:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sec="http:
  <soapenv:Header>
     <sec:Security>
        <UsernameToken>
           <Username>admin</Username>
           <Password>admin</Password>
        </UsernameToken>
     </sec:Security>
  </soapenv:Header>
  <soapenv:Body>
     <urn:searchData>
        <branch>Reference</branch>
        <instance>Tese</instance>
        <tablePath>/root/Employee</tablePath>
        <locale>en_US</locale>
        <keyword>Abraham</keyword>
        <sensitivity>50</sensitivity>
        <!--Optional:-->
        <persistedResults>false</persistedResults>
     </urn:searchData>
  </soapenv:Body>
</soapenv:Envelope>
```

You can use the **persistedResults** property to enable or disable search result storage by specifying a value of **`true`** or **`false`**, respectively.

- The following is an example of the SOAP response that is returned by executing the above SOAP request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSch
   <soapenv:Body>
      <ns1:searchData_Response xmlns:ns1="urn:ebx-addon-tese:tese_search">
         <results>
            <result>
               <predicate>./Employee_id=4</predicate>
               <highest_score>100.00</highest_score>
               <details>
                  <detail>
                     <field>/Name</field>
                     <score>100.00</score>
                     <text>Abraham</text>
                  </detail>
               </details>
            </result>
            <result>
               <predicate>./Employee_id=6</predicate>
               <highest_score>99.90</highest_score>
               <details>
                  <detail>
                     <field>/Name</field>
                     <score>99.90</score>
                     <text>abraham</text>
                  </detail>
               </details>
            </result>
            <result>
               <predicate>./Employee_id=5</predicate>
               <highest_score>99.90</highest_score>
               <details>
                  <detail>
                     <field>/Name</field>
                     <score>99.90</score>
                     <text>Abrahaam</text>
                  </detail>
               </details>
            </result>
         </results>
      </ns1:searchData_Response>
   </soapenv:Body>
</soapenv:Envelope>
```

CHAPTER 10

# Running searches

This chapter contains the following topics:

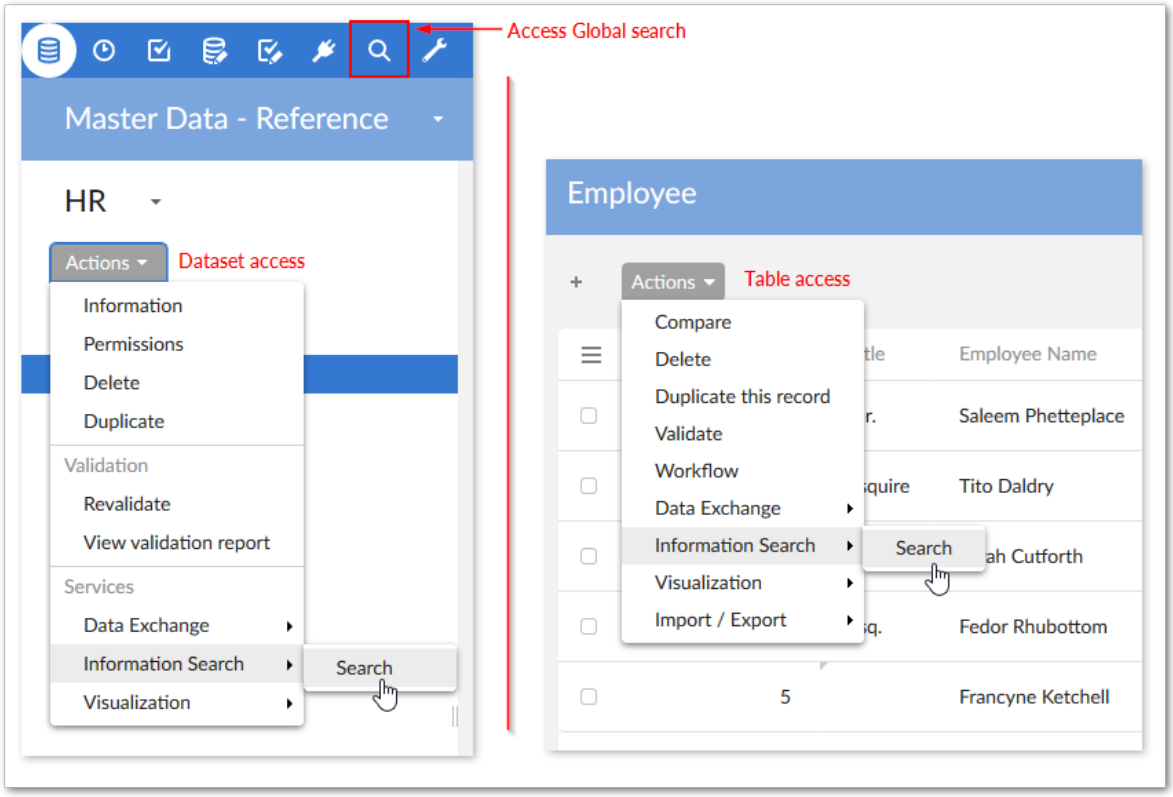


## 10.1 Executing a search

To run a new search:

1. As shown in the image below, you can access the search feature from a dataset or table **Actions** menu. When you perform a search at the dataset level, the search spans all of the dataset's tables configured to use the add-on. Whereas, running a search from a table limits the search scope to the fields in the table. If the global search feature is enabled for your profile, use the **Search** icon

in the main toolbar to access it. A global search can include tables from multiple dataspace and dataset locations.

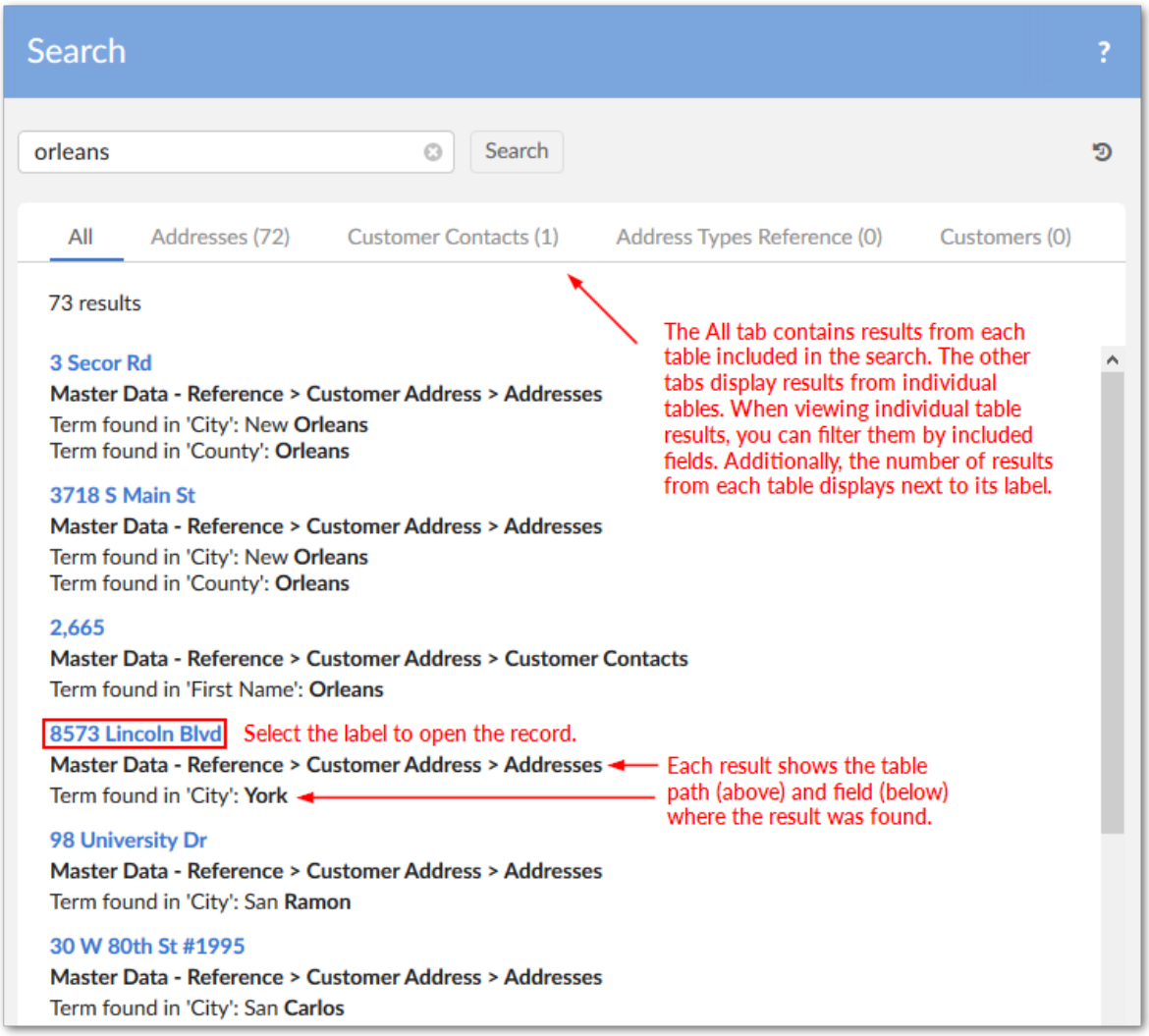



2. Input your search term in the text box and click **Search**, or Enter on your keyboard. If an administrator has enabled the **Best word** option, you can use double-quotes around multiple words to force a search on the entire phrase.

> **Note**
>
> If you are searching for a date, apply the format 'DD/MM/YYYY' for French and 'MM/DD/YYYY' for other languages. Additionally, under some conditions, when you use a number as a keyword and include characters such as decimal points and commas no results are returned.

3. Browse the search results. The following image shows the result of running a search at the dataset level. The fist tab displays a collection of results from all the tables. Each subsequent tab label displays the number results found in the corresponding table. The search result screen lists tabs in

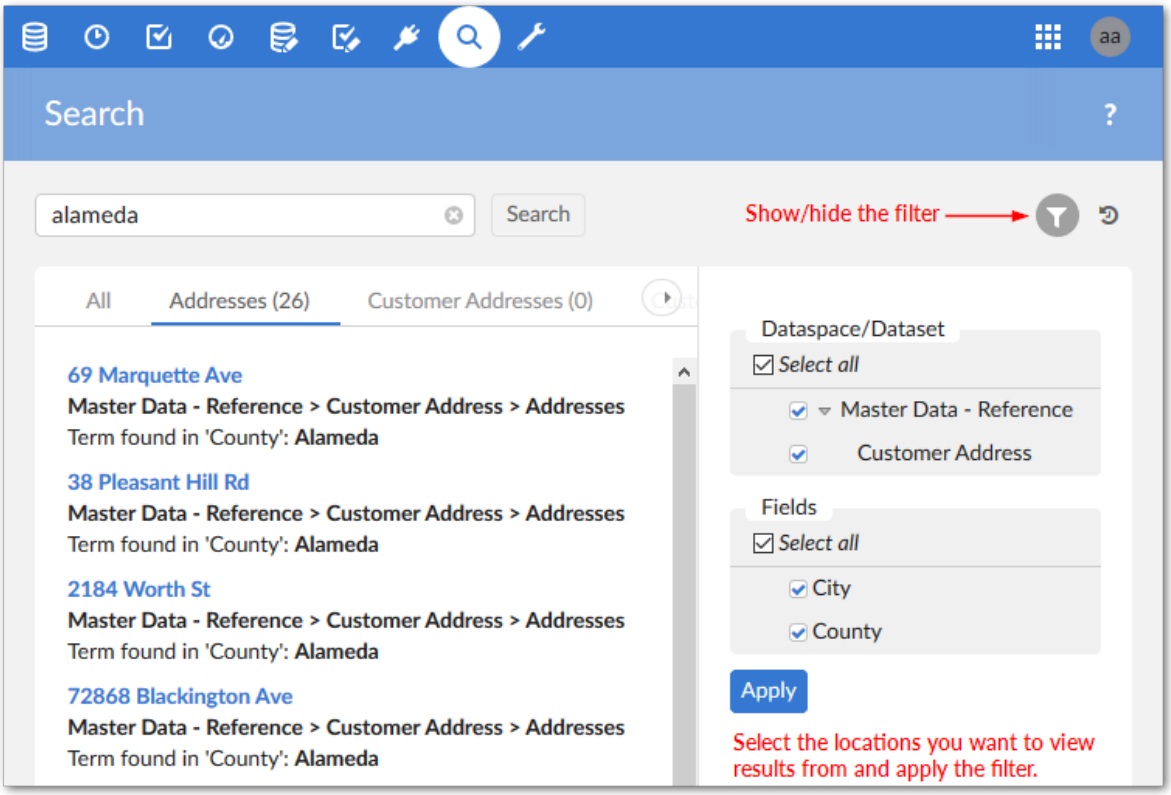
descending order based on the number of results found in the table. Select a tab to view individual table results.




> **Note**
>
> If table data is updated after performing a search, you need to re-execute the search to synchronize the results.

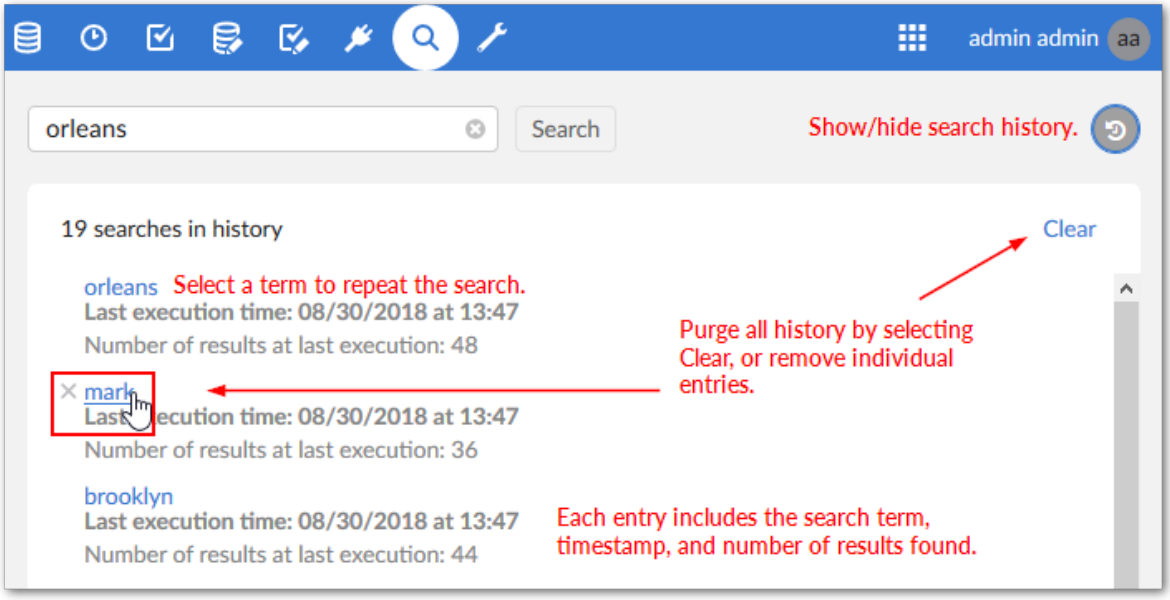
## *Filtering search results*

When viewing table tabs, you can use the filter to show results from desired fields. If you run a global search, the filter also allows you to filter results from dataspaces and datasets. The following image highlights the filter functionality after running a global search:



### *Displaying search history*

You can show your search history by clicking **History** in the top-right of the **Search** window. Administrators can configure options that determine how much history is kept before overwriting and can purge all history. As a user, you can clear all of your history, or just individual entries in the history.




# 10.2 Search using a REST request

This section describes how you can perform a search using REST. Rather than using complex mechanisms to create connections between machines, REST uses simple HTTP to make calls between them. Within the scope of the add-on, REST supports reading data. By submitting a REST from the address bar of your web browser, you can quickly execute a search. This type of search works for standard search configuration using configured tables and fields, and when the global search option is enabled.

Use the following syntax to search using a REST request:

```
http://<host>:<port>/ebx-addon-tese/rest?login=<user name>&password=<password>&dataspace=<dataspace name>&dataset=<dataset name>&tablePath=<table path>&keyword=<search keyword>&sensitivity=<search sensitivity setting>&searchdate=<search date info>
```

Use the following syntax to perform a search when the global search option is enabled:

`http://<host>:<port>/ebx-addon-tese/rest?login=<user name>&password=<password>&keyword=<search keyword>&sensitivity=<search sensitivity setting>`

| Input | Description |
|---|---|
| Host | The name of your host. |
| Port | The port number. |
| User | Your user name. |
| Pass | Your password. |
| DataspaceName | The name of the dataspace you want to search. |
| DatasetName | The name of the dataset you want to search. |
| TablePath | The path to the table you want to search. It is possible to search in multiple tables at a time by employing the syntax:*tablePath=...&tablePath=...* |
| Keyword | The keyword you want to search. It is possible to search for multiple keywords by using the syntax: keyword=...&keyword=... Value of keyword parameters must be converted to the application/x-www-form-urlencoded MIME format. For more information about HTML form encoding, consult the HTML specification (https://www.w3.org/TR/html4/) |
| Sensitivity | Your desired sensitivity level. |
| Search date | If **Yes**: The search date is enabled.<br>If **No**: The search date remains disable.<br>Default value: **No**.<br>Keyword parameter must set with date format as '**YYYY-MM-DD'.** |

Table 1: Elements of REST search syntax

The result of a REST search will be displayed as in the example below.

CHAPTER 11

# How indexing is handled

The first time a user performs a search, the add-on indexes configured tables. The add-on updates the index each time a record is inserted, updated, or deleted. The following describes additional details on how the add-on handles indexing:

- If the process runs out of memory, the JVM can remove indexed data. The next search performed triggers the indexing process.
- When a user updates a record, indexed data is updated if a field configured for search is modified.
- If searched fields contain a foreign key field and this field doesn't have a programmatic label rule, a cache will be built for the target table—the table the FK points to. The following also applies:
  - If a record in the target table updates a field that participates in defining the record's label, the cache will be updated.
  - If a record in target table is deleted, the cache will be updated.
  - If the table being searched and the target table are in the same dataspace, the update process will be performed real-time. If those tables are in different dataspaces, the update process will be performed at the next search.

CHAPTER 12

# API

The EBX® Information Search Add-on API allows you to perform a programmatic search. Please refer to the Java doc available online.

# Developer Guide

CHAPTER 13

# Implementing a custom search algorithm

This chapter contains the following topics:



## 13.1 Overview

You can use the add-on's API to implement a custom search algorithm in Java. Administrators can use the algorithm when configuring the add-on. The following bullets highlight the requirements to implement a custom algorithm:

- Write the class that includes the logic for the search algorithm.
- Create a definition class.
- Register the definition class in your custom module.

> **Attention**
> To use the sample code in the following sections, you must create and deploy your own custom module. For instructions, see the TIBCO EBX® developer documentation.

## 13.2 Example algorithm class

The following sample class defines a custom search algorithm. This class must extend the `SearchDistance` abstract class. The `getDistance()` method is where you define how the add-on calculates the similarity between a search keyword and the data being searched.

```
public class MyContainsAlgorithm extends SearchDistance
{

 public MyContainsAlgorithm(SchemaNode schemaNode, Locale locale)

 {
  super(schemaNode, locale);
 }
```

```
 @Override
 public float getDistance(String keyword, String text)
 {
  if (text.contains(keyword))
  {
   return 100f;
  }
  return 0;
 }

 @Override
 public boolean isNumericSupported()
 {
  return false;
 }

 @Override
 public boolean isFullTableScanRequired()
 {
  return true;
 }

 @Override
 public SearchComparator cloneMatching(SchemaNode newNode)
 {
  return new MyContainsAlgorithm(newNode, this.getLocale());
 }

}
```

# 13.3 **Sample definition class**

A definition class provides any labels, descriptions or parameters that will display in the UI during the configuration process. Additionally, this class instantiates the algorithm class from the previous section.

```
public class MyContainsAlgorithmDefinition implements AlgorithmDefinition
{

 @Override
 public UserMessage getLabel()
 {
  return UserMessage.createInfo("My Custom Algorithm");
 }

 @Override
 public UserMessage getDescription()
 {
  return UserMessage.createInfo("This is a custom algorith!");
 }

 @Override
 public List<ParameterDefinition> getParameters()
 {
  return null;
 }

 @Override
 public SearchComparator getAlgorithm(AlgorithmDefinitionContext adContext)
 {
  if (adContext == null)
  {
   return new MyContainsAlgorithm(null, Locale.US);
  }
  return new MyContainsAlgorithm(adContext.getField(), adContext.getLocale());
 }

 @Override
 public Class<SearchAlgorithmConfiguration> getSearchConfigurationClass()
 {
  return SearchAlgorithmConfiguration.class;
 }

 @Override
 public UserMessage checkParameterValueRange(List<InputParameter> parameters, Locale locale)
 {
  return null;
 }
```

```
}
```

## 13.4 **Register the definition class**

To register you definition class, add the following to the `handleRepositoryStartup()` method in your module's registration servlet.

```
private void registerTeseAlgorithms()
 {
  AlgorithmCatalog.add(new MyContainsAlgorithmDefinition());
 }
```

# Reference Guide

CHAPTER 14

# The Configuration domain

This chapter contains the following topics:



## 14.1 Overview

The **Configuration** domain allows you to:

- Create search configurations by pointing the add-on to a location to search.
- Specify values you want to exclude from search results.
- Create and modify algorithms.
- Create synonyms that the add-on can return when no results are found for a particular search term.

To give users access to global search and control which tables they can search, admins create entries in the **Permission** and **Context** tables. Each record in the **Context** table defines one or more tables as a search context. Only tables previously configured in the add-on can be included in a search context. Entries in the **Permission** table associate user profiles with a context. Global search will only be active for a user if they, or a role containing their profile, is associated with at least one context.

### *The Permission table*

The **Permission** table associates user profiles with search contexts to enable global search for those profiles. The **Context** table lets you control the scope of tables that are searchable by a particular profile.

| Property | Definition |
|---|---|
| Profile | The profile, or role that will be allowed to access global search functionality. Set the search scope for this profile using the **Context** field. |
| Context | The context defines which table(s) will be included when the chosen profile performs a global search. |

## The Context table

This table allows you to define the scope for a global search. Each context can be linked to a user profile, thereby enabling the global search feature for that profile on the tables defined in the context. Use the **Permission** table to link a profile with one or more contexts.

| Property | Definition |
|---|---|
| Code | A unique identifier for the context. No whitespaces are allowed. |
| Custom title | Use this field to customize the context's title that will be displayed in the global search screen. If no custom title is defined, or several contexts are defined for a single profile, the title defaults to **Search**. |
| Dataspace | Dataspace where the search will be executed. |
| Dataset | Dataset where the search will be executed. |
| Tables | The list of tables included in this context that will be searched during a global search operation. Note that only tables that have been previously configured in the add-on will be available for selection. |

## The Model table

The **Model** table allows you to apply search criteria to a data model. These settings affect search behavior for all tables included in the model. As a result, the table's dataset location does not matter. As long as you have registered a table and configured one or more of its fields, users can execute search. See the **Searched tables** and **Searched fields** tables for more information.

The table below lists all properties and how those properties are used in the **Model** configuration screen.

| Property | Definition |
|---|---|
| Data model | The data model on which the options on this page apply. These settings apply to all tables based on this data model for which you enable search. |
| Sensitivity default value (%) | Fuzzy search gives a percentage level of the similarity between information and the search criterion. For example, the criterion Jone is considered 73% similar to John. The level from which the information will be considered a relevant search result must be configured carefully. If the sensitivity level is set too high, then some results could be missed. Conversely, if this level is too low then unwanted information could be retrieved.<br>This means that the sensitivity value must be tested to check that the results are relevant. You may need to adjust the default sensitivity level-on a case by case basis-depending on the nature of the information on which the search is applied (string, digit, short text, long description, languages, etc.) and the algorithms that are used for the fuzzy search (phonetics, distance).<br>By default the value of the sensitivity is set to 70%. |

## The Searched tables table

You can use the **Searched tables** table to register tables with the add-on. Keep in mind that all properties related to the sensitivity level of the search and the number of possible results are configured

at the data model and field levels. The table configuration should only be used to select which tables inside the data model benefit from the add-on's functionality.

| Property | Definition |
|---|---|
| Data model | A reference to the data model configured in the **Model** table. |
| Table | A table in the data model. Once a table is configured it benefits from add-on functionality no matter its location in the datasets. |
| Results sorted by | Defines how the results are displayed:<br>• Scoring order: The result are displayed in order of descending score.<br>• Field order: The results are displayed in order of field configuration.<br>• Default value: Scoring order |
| Result line template | Defines how each result will be displayed. Up to five lines can be configured. At least one result line template must be configured. |

## *The Searched fields table*

In the **Searched fields** table you specify on which table fields the add-on executes. You can configure how the search executes by:

- Specifying **Primary search algorithm** or a **Secondary search algorithm**.
- Enabling the **Best word** option.

- Using a Java class to filter out specific values before the search executes.

| Property | Definition |
|---|---|
| Table | A table configured to use add-on functionality. |
| Field | A field in the configured table on which the add-on executes. |
| Primary search algorithm | The primary algorithm used to execute a fuzzy search. Different algorithms are available to either enforce a phonetics search or a distance search. See the following recommendations:<br>• For name and first name: "Jarowinkler".<br>• For products titles, films names, etc.: "FuzzyFulText" with suitable configuration for the **Similarity** and **Prefix length** properties to get the best expected results.<br>• For postal code or similar conventions: "Levenshtein".<br>• For number: Search Number<br>• For date: Search Date |
| Secondary search algorithm | This algorithm executes after the primary algorithm and you can use it to manage records considered a false negative result by the primary algorithm. This works to ensure that no possible record has been missed. |
| Best word | A **Best word** search scores results based solely on how close a word matches the search criteria. It assigns each word a score and returns the word with the best score. This type of search works well when searching fields such as descriptions that contain many words. You can use this option with distance and phonetic algorithms. This option does not work for Search Number, Search Date algorithms.<br>For example, using the search criteria "lice" on a table's Name field could return a result of "Alice". The basic fuzzy search gives this field a high score because of the similarity between the search criteria and the information contained in the field. If the same search criteria is applied instead to search a descriptive field containing the text "A cartoon character named Alice", the search score would greatly decrease. This is because the percentage of similarity-or score-is computed using all of the words in the text.<br>If set to **Yes**: The search bases a word's score solely on matching against the criteria. Users can also enclose multiple search terms in double-quotes to force a search on the entire phrase.<br>If set to **No** or **Undefined**: The search bases a word's score on the percentage of similarity between the criteria and the data being searched.<br>Default value: No |
| Java class value filter | This property accepts a Java class to define values you want to exclude from a search operation. You can use the add-on's predefined class (`com.orchestranetworks.addon.tese.SearchValueFilter`), or create your own custom class. The predefined class gets values to filter from the **Filtered value group** table.<br>If you decide to create a custom class, include in the class the values to filter and be sure the class implements the SearchTextNormalize interface. The value filter depends on the record's fields. |
| Filtered value group | This property allows you to choose a group that contains values to exclude from the search. If you haven't created a group yet, use the **Filtered value** and **Filtered value group** to do so.<br>Note that this property only displays when you use the predefined `com.orchestranetworks.addon.tese.SearchValueFilter` class in the **Java class value filter** property. |
| Synonyms processing mode | Defines the processing mode for synonyms, you can select from the following options:<br>• **Use synonyms for the entire value of the attribute**: Apply synonyms on the entire value of the attribute.<br>• **Use synonyms for every words of the attribute**: Apply synonyms on every words of the attribute. |

| Property | Definition |
|---|---|
| Check synonyms in all groups | If **Yes**: In addition to the configured group, all the groups with a parent relationship are used to look for the synonyms.<br>If **No** or **Undefined**: Records in different child groups cannot be matched with each other.<br>Default value: **No** |

## The Filtered value table

In the **Filtered value** table you input a value to exclude from a search and assign the value to a group.

| Property | Definition |
|---|---|
| Value | A value to ignore when performing a search. |
| Group | The **Filtered value group** this value is assigned to. |

## The Filtered value group table

The **Filtered value group** table allows you to define groups that contain values to exclude from a search. After creating a group, you can use the **Filtered value** table to assign values to the group. For more information, see the **Filtered value group** table.

| Property | Definition |
|---|---|
| Group | Name of the **Filtered value group**. |

## The Algorithm table

Algorithm table provides a list of ready-to-use algorithms with [ON] prefix for searching. It is also possible for user to create new algorithms with different parameter values using the supported algorithms.

| Property | Definition |
|---|---|
| Code | A unique name without white spaces. |
| Label | Any naming convention without white spaces can be used to define this property. |
| Supported algorithm | The chosen algorithm from which you can customize its input parameters to create a new algorithm used to search. |
| Input parameters | A list of customized parameters. The available parameters depend on the corresponding **Supported algorithm**. |

## The Synonym table

This table allows you to create and store synonyms. After creating synonyms you can create groups, and add synonyms to groups using the **Synonym group** and **Populate synonym group** tables, respectively.

| Properties | Definition |
|---|---|
| Code | Any naming convention except the prefix [ON] that is reserved for the synonyms provided by the add-on can be used. |
| Name | The name of the synonym. |

## The Synonym group table

You can use this table to create a group of synonyms. To add synonyms to groups, use the **Populate synonym group** table. Additionally, you can create a parent or child relationship between groups. On the **Searched fields** configuration, if the **Check synonyms in all groups** is set to **Yes**, all child synonym groups are used to look for a synonym. If set to **No**: only the synonyms in the same group can be used to search.

| Properties | Definition |
|---|---|
| Code | Any naming convention except the prefix [ON] that is reserved for the synonyms provided by the add-on can be used. |
| Name | The name of the synonym group. The **Synonym by group** data hierarchy view on the **Synonym** table is used to attach the synonyms to a group. |
| Parent group | Groups of synonyms can be arranged through a relationship hierarchy. When you use a synonym group to search a field, you can specify that the add-on looks for synonyms through these relationships. |

## The Populate synonym group table

You can use this table to assign synonyms to desired groups.

| Properties | Definition |
|---|---|
| Synonym group | A record of the Synonym group table. |
| Synonym | A record of the Synonym table. |

CHAPTER 15

# The Execution domain

This chapter contains the following topics:



## 15.1 Overview

The tables in the execution group allow you to set user preferences, view, and purge the search history.

### *The User preference table*

The user preference table allows you to apply preference settings to individual users. If a preference is not defined for a user, the default preference settings will be applied.

| Property | Definition |
|---|---|
| User | Determines the user these preferences apply to. |
| Max No. of saved searches | Maximum number of history items that can be stored for a user profile per data model, table or global search. For example: if a value of 20 is defined, the user can have maximum of 20 records for each data model (search at dataset level), 20 records for each table (search at the table level) and 20 records with empty data model and table (search at global level). Default value: 20 |
| Display history by default | Determines whether history displays by default on the search home page. |
| Maximum No. of results | This property determines the maximum number of records found in the search result. Only the highest ranking results are displayed. If left undefined, the number of results is unbounded. |

## *The History table*

The **History** table displays a recorded search history for all users. To purge the history, run the **Purge history** service from the table's **Actions** menu.

| Property | Definition |
|---|---|
| Timestamp | Date and time of the search was executed. |
| Keyword | The keyword used to search. All keywords are converted to lower case in the **History** table. |
| No. of results | The number of results that have been found. |
| User | User in charge of the search execution. |
| Data model | The data model on which the search is applied. This field is specified when searching at the dataset level. If executed at the table level or global level, this field is empty. |
| Table | The table on which the search is applied. This field is specified when searching at the table level. If executed at the dataset level or global level, this field is empty. |

# Release Notes

CHAPTER 16

# Version 2.4.22

**Released: October 2022**

This chapter contains the following topics:



## 16.1 New features

This release contains no new features.

## 16.2 Changes in Functionality

This release contains no changes in functionality.

## 16.3 Changes to third-party libraries

This release contains the following changes to third-party libraries:

- The Apache Taglibs library was removed.
- The FasterXML jackson-databind library was updated to version 2.14.0-rc2.

## 16.4 Closed issues

This release contains no closed issues.

## 16.5 Known issues

This release contains the following known issues:

- Search is executed on fields of string, text, date and number types only.
- Search with computed values or inherited fields might lead to performance issues.
- The use of keywords that contain only special characters is not supported.

CHAPTER 17

# All release notes

This chapter contains the following topics:

27. Release Note 2.2.3
28. Release Note 2.2.2
29. Release Note 2.2.1
30. Release Note 2.2.0
31. Release Note 2.1.3
32. Release Note 2.1.2
33. Release Note 2.1.1
34. Release Note 2.1.0
35. Release Note 2.0.5
36. Release Note 2.0.4
37. Release Note 2.0.3
38. Release Note 2.0.2
39. Release Note 2.0.1
40. Release Note 2.0.0
41. Release Note 1.7.5 fix 001
42. Release Note 1.7.5
43. Release Note 1.7.4
44. Release Note 1.7.3
45. Release Note 1.7.2
46. Release Note 1.7.1
47. Release Note 1.7.0
48. Release Note 1.6.5
49. Release Note 1.6.4
50. Release Note 1.6.3
51. Release Note 1.6.2
52. Release Note 1.6.1
53. Release Note 1.6.0
54. Release Note 1.5.4
55. Release Note 1.5.3 fix 001
56. Release Note 1.5.3
57. Release Note 1.5.2
58. Release Note 1.5.1
59. Release Note 1.5.0
60. Release Note 1.4.3
61. Release Note 1.4.2
62. Release Note 1.4.1

# 17.1 Version 2.4.22

**Released: October 2022**

## *New features*

This release contains no new features.

## *Changes in Functionality*

This release contains no changes in functionality.

## *Changes to third-party libraries*

This release contains the following changes to third-party libraries:

- The Apache Taglibs library was removed.
- The FasterXML jackson-databind library was updated to version 2.14.0-rc2.

## *Closed issues*

This release contains no closed issues.

## *Known issues*

This release contains the following known issues:

- Search is executed on fields of string, text, date and number types only.
- Search with computed values or inherited fields might lead to performance issues.
- The use of keywords that contain only special characters is not supported.

# 17.2 Version 2.4.21

**Released: August 2022**

## *New features*

This release contains no new features.

## *Changes in Functionality*

This release contains no changes in functionality.

## *Changes to third-party libraries*

This release contains no changes to third-party libraries:

## *Closed issues*

**[TESE-1352]** A `NullPointerException` occurs when executing a matching operation.

## *Known issues*

This release contains the following known issues:

- Search is executed on fields of string, text, date and number types only.
- Search with computed values or inherited fields might lead to performance issues.
- The use of keywords that contain only special characters is not supported.

# 17.3 Version 2.4.20

**Released: May 2022**

## *New features*

This release contains no new features.

## *Changes in Functionality*

This release contains no changes in functionality.

## *Changes to third-party libraries*

This release contains the following changes to third-party libraries:

- The Spring framework was updated to version 5.2.22.

## *Closed issues*

This release contains no closed issues.

## *Known issues*

This release contains the following known issues:

- Search is executed on fields of string, text, date and number types only.
- Search with computed values or inherited fields might lead to performance issues.
- The use of keywords that contain only special characters is not supported.

# 17.4 Version 2.4.19

**Released: April 2022**

### *New features*

This release contains no new features.

### *Changes in Functionality*

This release contains no changes in functionality.

### *Changes to third-party libraries*

This release contains the following changes to third-party libraries:

- The Spring framework was updated to version 5.2.20.
- The FasterXML/jackson-databind library was updated to version 2.13.2.1.

### *Closed issues*

This release contains no closed issues.

### *Known issues*

This release contains the following known issues:

- Search is executed on fields of string, text, date and number types only.
- Search with computed values or inherited fields might lead to performance issues.
- The use of keywords that contain only special characters is not supported.

# 17.5 Version 2.4.18

**Released: March 2022**

### *New features*

This release includes support for new features in the TIBCO EBX® Match and Merge Add-on (version 2.5.16).

### *Changes in Functionality*

This release contains no changes in functionality.

### *Changes to third-party libraries*

This release contains the following changes to third-party libraries:

- The Spring framework was updated to version 5.2.19.

### *Closed issues*

This release contains no closed issues.

### *Known issues*

This release contains the following known issues:

- Search is executed on fields of string, text, date and number types only.
- Search with computed values or inherited fields might lead to performance issues.
- The use of keywords that contain only special characters is not supported.

## 17.6 **Version 2.4.17**

**Released: January 2022**

### *New features*

This release contains no new features.

### *Changes in Functionality*

Sections are now hidden in the search UI when they do not contain results.

### *Changes to third-party libraries*

This release contains no changes to third-party libraries.

### *Closed issues*

**[TESE-1310]** A `ClassCastException` occurs when updating a record in a reference table.

### *Known issues*

This release contains the following known issues:

- Search is executed on fields of string, text, date and number types only.
- Search with computed values or inherited fields might lead to performance issues.
- The use of keywords that contain only special characters is not supported.

## 17.7 **Version 2.4.16**

**Released: December 2021**

### *New features*

This release contains no new features.

### *Changes in Functionality*

This release contains no functionality changes.

### *Changes to third-party libraries*

The Spring Framework was updated to version 5.2.15.

### *Closed issues*

This release contains no closed issues.

### *Known issues*

This release contains the following known issues:

- Search is executed on fields of string, text, date and number types only.
- Search with computed values or inherited fields might lead to performance issues.
- The use of keywords that contain only special characters is not supported.

## 17.8 **Version 2.4.15**

**Released: October 2021**

### *Bug fixes*

This release contains the following bug fixes:

- **[TESE-1290]** A NullPointerException occurs when starting the repository.

### *Changes in Functionality*

This release contains no functionality changes.

### *Changes to third-party libraries*

This release contains no changes to third-party libraries.

### *Closed issues*

This release contains no closed issues.

### *Known issues*

This release contains the following known issues related to matching functionality:

- Search is executed on fields of string, text, date and number types only.
- Search with computed values or inherited fields might lead to performance issues.
- The use of keywords that contain only special characters is not supported.

## 17.9 **Version 2.4.14**

**Released: August 2021**

### *New features*

This release includes support for new features and bug fixes in the TIBCO EBX® Match and Merge Add-on (version 2.5.12).

### *Changes in Functionality*

This release contains no functionality changes.

### *Changes to third-party libraries*

This release contains no changes to third-party libraries.

### *Closed issues*

This release contains no closed issues.

## 17.10 **Version 2.4.13**

**Released: June 2021**

### *Library updates*

Spring Data was removed from `ui-framework-dependencies.jar`.

### *Bug fixes*

This release includes support for bug fixes in the TIBCO EBX® Match and Merge Add-on (version 2.5.11).

## 17.11 **Version 2.4.12**

**Released: May 2021**

### *Optimizations*

There will be no change in the searching cache when you modify a non-search field.

### *Bug fixes*

This release includes support for bug fixes in the TIBCO EBX® Match and Merge Add-on (version 2.5.10).

## 17.12 **Version 2.4.11**

**Released: March 2021**

### *Updates*

An update was applied to correct an issue with the Apache Standard Taglibs library.

# 17.13 Version 2.4.10

**Released: February 2021**

### *Bug fixes*

**[TESE-1243]** The add-on executes indexing to cache data continuously when executing a **Search** operation.

# 17.14 Version 2.4.9

**Released: January 2021**

### *Product updates*

The following libraries were updated in this release:

- Apache Standard Taglibs library to version 1.2.3.
- Spring framework library to version 5.2.9.
- Jackson Databind library to version 2.11.2.

### *Bug fixes*

This release includes support for bug fixes in the TIBCO EBX® Match and Cleanse Add-on (version 2.5.7).

# 17.15 Release Note 2.4.8

**Release Date: October 20, 2020**

### *Bug fixes*

This release includes support for bug fixes in the TIBCO EBX® Match and Cleanse Add-on (version 2.5.6).

# 17.16 Release Note 2.4.7

**Release Date: September 18, 2020**

### *New features and optimizations*

This release contains the following new features and optimizations:

- An empty line separator has been added between each result in the **Search** screen.
- The add-on has been updated to support the OpenJDK8 libraries.

### *Bug fixes*

This release contains the following bug fixes:

- **[TESE-1145]** A blank page is returned when executing a **Global search** and the user has insufficient permissions.

- **[TESE-1148]** An incorrect highlight is returned when searching a compound keyword.
- **[TESE-1150]** A `NumberFormatException` occurs when searching a multi-value foreign key field under multi-value group.
- **[TESE-1151]** An incorrect result returned when searching a compound keyword in the multi-value string field and the **Best word** option is activated.

# 17.17 Release Note 2.4.6

**Release Date: June 23, 2020**

## *New features*

This release includes support for new features in the TIBCO EBX® Match and Cleanse Add-on (version 2.5.4).

## *Bug fixes*

This release contains the following bug fixes:

- **[TESE-1144]** An incorrect result is returned when searching on a foreign key with a numeric keyword.
- **[TESE-1146]** An incorrect result is returned when executing the **Global search** operation when the **Specific policy** property is configured.

# 17.18 Release Note 2.4.5

**Release Date: April 20, 2020**

## *New features*

This release contains the following new features:

- The add-on can now display custom table labels on search results. You can use this feature to clarify search result display. For example, a global search might include results from tables with the same name but that come from different data models. In this case, a custom label could help differentiate between tables by naming each table's containing model. Administrators set the custom label using a configured table's **Label** option.
- It is now possible to search a label of a String enumeration field.

## *Bug fixes*

This release contains the following bug fixes:

- **[TESE-1119]** An incorrect result is returned when searching on a raw-value of a foreign key.
- **[TESE-1121]** An incorrect result is returned when searching on an enumeration field of the Integer, Date and HTML data types.
- **[TESE-1124]** An incorrect result is returned when searching on a dataset through REST.
- **[TESE-1135]** An incorrect result is returned when searching on an external foreign key and the **Best word** property is activated.

# 17.19 Release Note 2.4.4

**Release Date: January 15, 2020**

## *Product updates*

The TIBCO EBX® Information Search Add-on no longer ships with certain third-party Limited General Public License (LGPL) libraries. You must re-package the add-on before you can successfully deploy it. The add-on bundle includes a script to automate this process. Please see the *TIBCO EBX™ Add-ons Versioning and Packaging Guide* for detailed instructions. You can find this guide in the PDF documents included with your add-on bundle, or on *https://docs.tibco.com* under the page dedicated to the TIBCO EBX™ 4.3.2 Add-ons Bundle.

## *Bug fixes*

This release contains the following bug fixes:

- **[TESE-1027]** An incorrect result is returned when executing a **Search** operation in a dataspace after merging a child dataspace.

# 17.20 Release Note 2.4.3

**Release Date: December 10, 2019**

## *Bug fixes*

This release includes support for bug fixes in the TIBCO EBX® Match and Cleanse Add-on (version 2.5.1).

# 17.21 Release Note 2.4.2

**Release Date: November 8, 2019**

## *New features*

The **Contains** search algorithm is now available for use. This algorithm finds data containing the keyword. It is not case-sensitive.

## *Bug fixes*

This release contains the following bug fixes:

- **[TESE-998]** An incorrect result is returned when searching on a foreign key with a tabular filter.
- **[TESE-1001]** An incorrect result is returned when searching on a parent dataspace after merging a child dataspace.

# 17.22 Release Note 2.4.1

**Release Date: June 20, 2019**

### *Updates*

This release includes support for new features in the TIBCO EBX® Match and Cleanse Add-on (version 2.4.0).

### *Bug fixes*

**[TESE-977]** A concurrent issue occurs when executing multiple search threads on a table at the same time.

**[TESE-990]** A NullPointerException occurs when executing a programmatic search.

## 17.23 Release Note 2.4.0

**Release Date: March 25, 2019**

### *New features and updates*

The new features and updates for this release are described in the following sections:


- General updates [p 85]
- UI updates and enhancements [p 85]


#### General updates

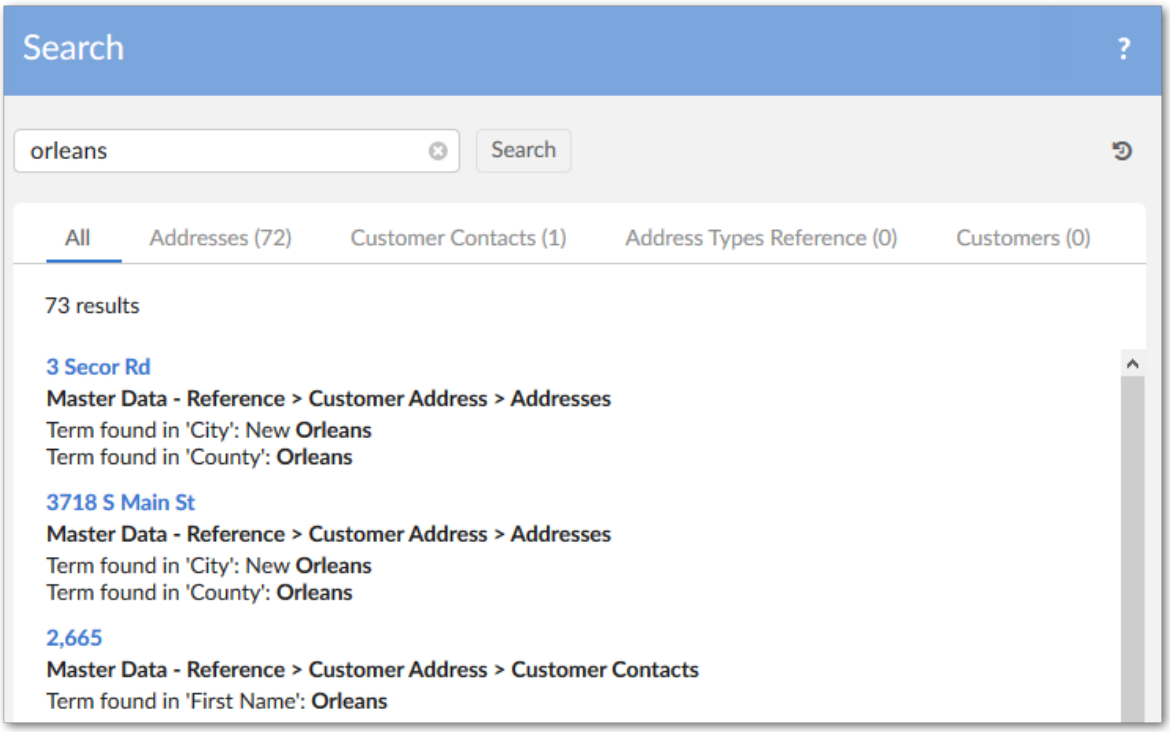
This release contains the following new features and updates:

- Add-on configuration settings have been moved under *Administration > Data quality & analytics > TIBCO EBX™ Information Search Add-on*.
- When using the **Search table filter**, the configured primary and secondary search algorithms are now applied.
- One of the following options can be selected when configuring the add-on to use synonyms during search:
  - **Use synonyms for the entire value of the attribute:** This option compares synonyms with the entire word or phrase being searched. Take fore example the following group of synonyms: "Mr", "Mister", and "Sir". Taken as a whole, the search phrase "Mr. Bob Smith" may not match close enough to trigger results. However, if the entire search term was "Sir", it could return the record for "Sir Bob Smith".
  - **Use synonyms for every words of the attribute:** This option compares synonyms with each part of a phrase being searched. Using the above example, the search phrase "Mr. Bob Smith" could return "Sir Bob Smith" as a result.
- The **Best word** option is now disabled when the keyword is surrounded by double quotes.
- This release contains support for new features in the TIBCO EBX™ GO for iOS and TIBCO EBX™ GO for Android mobile applications.

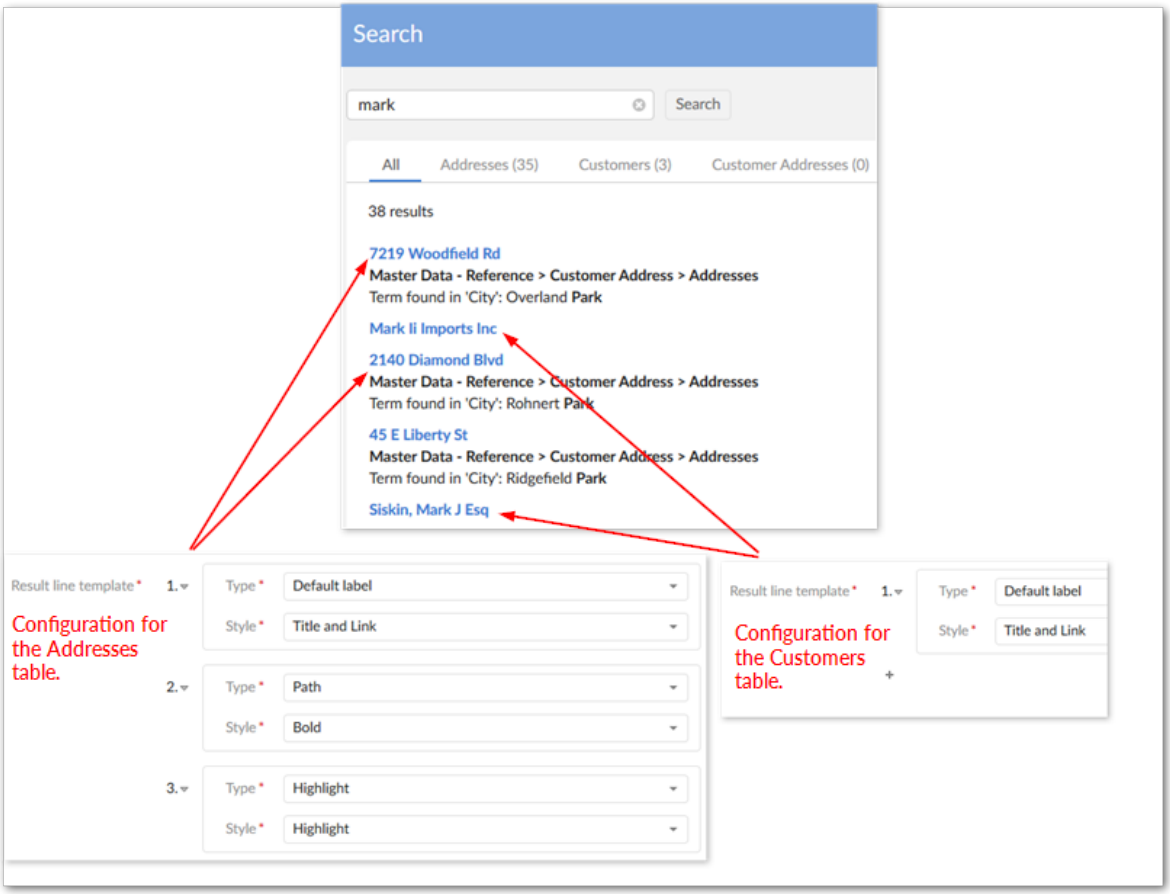
#### UI updates and enhancements

This release includes the following UI updates and enhancements:

- Search result tabs now display, and are ordered according to the number of results:



- Search result display can now be configured by administrators. Each line of a search result can be customized according to order of display, content, and formatting. The image below shows an example of different configurations for two tables:




### *Bug fixes*

This release contains the following bug fixes:

- **[TESE-885]** The results display incorrectly when executing **Global search** and the **Results sorted by** property is set to **Field order**.
- **[TESE-887]** An error message is displayed when users configure the parent group for a synonym group.

## 17.24 **Release Note 2.3.1**

**Release Date: December 13, 2018**

### *Featured updates*

This release contains the following updates and enhancements:

- Memory usage has been enhanced when loading the cache.

# 17.25 Release Note 2.3.0

**Release Date: October 26, 2018**

### *Featured updates*

This release contains the following updates and enhancements:

- The EBX® Information Search Add-on has undergone significant updates to ensure compatibility with the EBX® 5.9.0 GA release.
- To use the **Global Search** service within a perspective, you must now specify a value for the **Dataspace** parameter during configuration.

# 17.26 Release Note 2.2.4

**Release Date: October 12, 2018**

### *Bug fixes*

- **[36046]** A 404 error is returned when opening the **Search view** on a WebLogic server.

# 17.27 Release Note 2.2.3

**Release Date: October 5, 2018**

### *Bug fixes*

- **[35714]** The search result is incorrectly highlighted when using the **Best word** option, and there are at least two words that match the keyword.

# 17.28 Release Note 2.2.2

**Release Date: September 13, 2018**

### *Bug fixes*

- **[35409]** A Null pointer exception occurs when updating a **Filter by** field that was empty.

# 17.29 Release Note 2.2.1

**Release Date: July 31, 2018**

### *New features*

You can now search on a string field containing only numeric characters.

### *Bug fixes*

- **[34369]** A ClassCastException is thrown when searching with two algorithms and the **Primary search algorithm** returns no results.

# 17.30 Release Note 2.2.0

**Release Date: July 17, 2018**

### *New features*

It is now possible to define the display order for search results.

### *Bug fixes*

- **[32526]** A blank page is displayed when searching with the **Best word** option.
- **[33877]** A ClassCastException error occurs when searching with the **Levenshtein** algorithm and the **Best word** option activated.

# 17.31 Release Note 2.1.3

**Release Date: June 22, 2018**

### *New features*

This release includes support for new features in the TIBCO EBX™ GO Add-on.

### *Bug fixes*

- **[32779]** An incorrect result is returned when applying a search filter that has no columns selected.

# 17.32 Release Note 2.1.2

**Release Date: April 20, 2018**

### *New features*

This release includes support for new features in the Match and Cleanse Add-on (version 2.1.0).

# 17.33 Release Note 2.1.1

**Release Date: March 28, 2018**

### *Bug fixes*

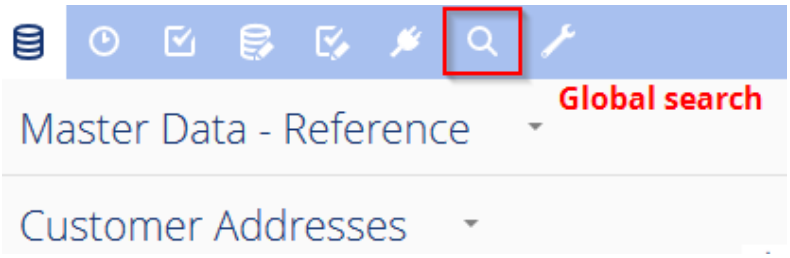
- **[31696]** The cache is not updated when updating data on foreign key tables.

# 17.34 Release Note 2.1.0

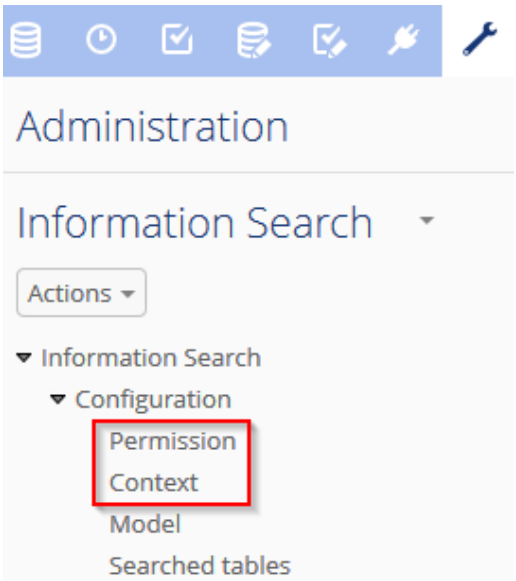
**Release Date: March 16, 2018**

## New features

This release includes the new global search feature. With this feature, searches can span multiple dataspaces and datasets. Global search is accessed from the main toolbar, which lifts the requirement that users navigate to the dataset or table they want to search.




When configuring global search, administrators create contexts that point to tables, and link the contexts with profiles. Note that in order to include a table in a context, it must already be configured in the add-on for localized dataset and table searches. The image below highlights the location of new tables used to configure global search.




The global search feature can be used when searching via a REST request. Additionally, global search is available as a perspective action and workflow task. For more information on using and configuring global search, see:

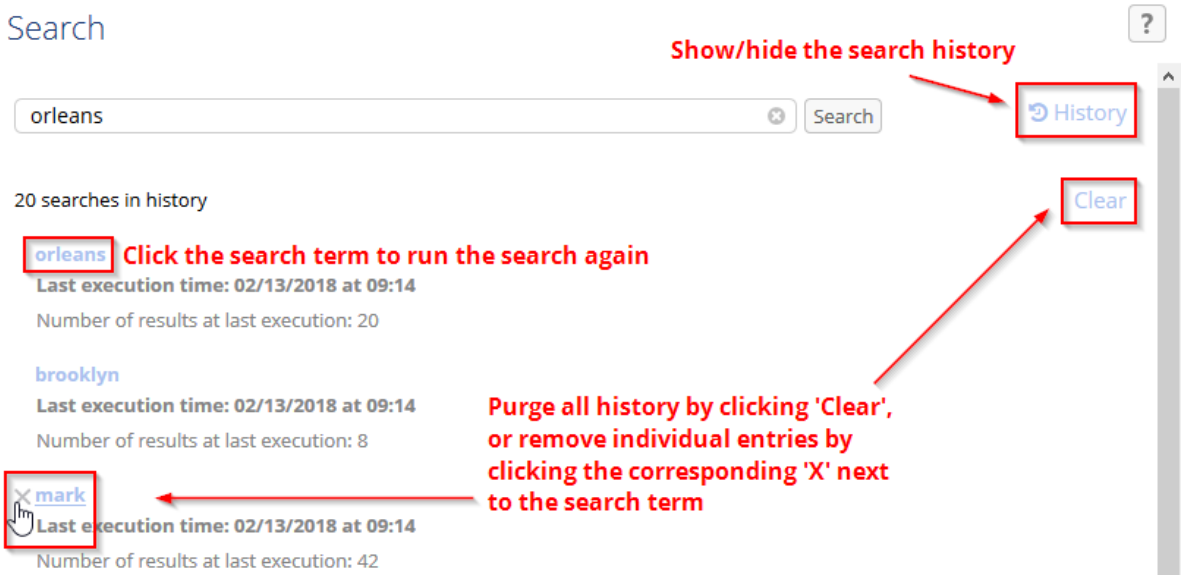
- Running searches [p 45]
- Configuration overview and examples [p 12]
- The Configuration domain [p 62]

## Updates and improvements

This release contains the following updates and improvements:

- The **Maximum No. of results** field has been moved to the **User preference** table. All previous configurations for this field have been reset to the default value of unbounded.

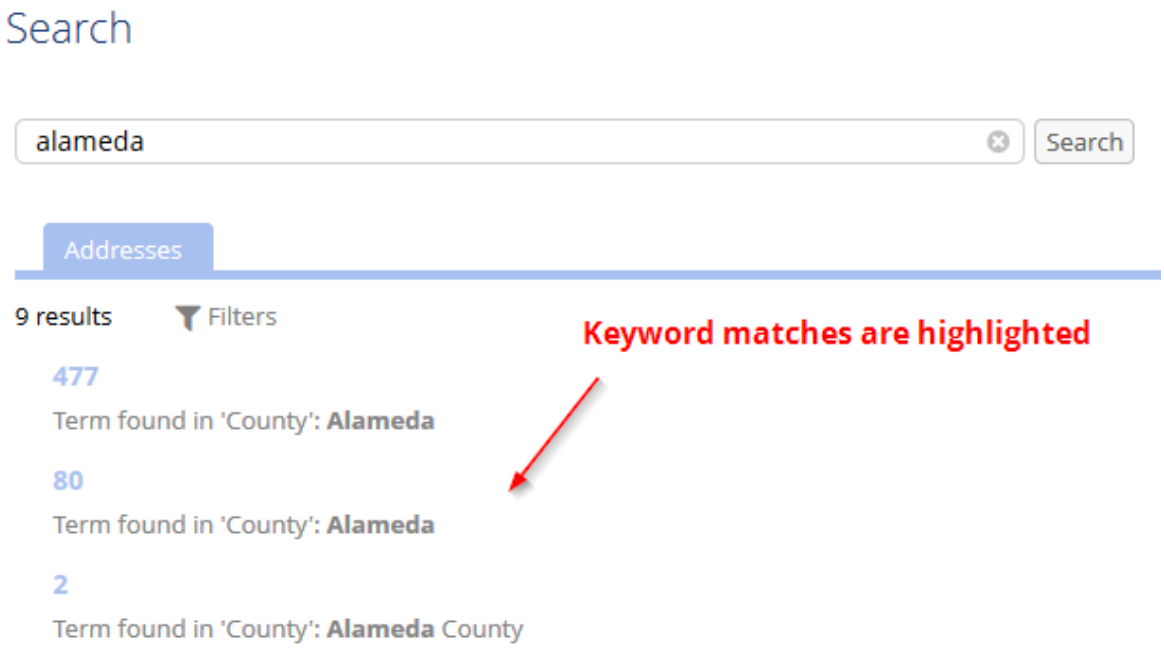
- You can now clear your own history. As shown below, individual entries can be removed, or you can completely reset your history:




- The **Check permission** option has been removed from the **Model** table, and permissions are now checked by default. This option is still included in the search APIs and you can enable/disable permission check via APIs.
- The table search filter appearance has been updated.
- New APIs are available to search at the repository level and within a user's history.
- In the SOAP and REST search services, the **sensibility** parameter has been updated to **sensitivity**. Both parameters are still supported for backwards compatibility.
- The field path is now included in the search result returned by REST service.


```
rest_search_results:
  0:
    keyword:                "New Orleans"
    result_all_tables:
      0:
        table:              "Addresses"
        results:
          0:
            predicate:      "165"
            highest_score:  100
            details:
              0:
                field:      "City"
                fieldPath:  "/city"
                score:      100
                text:       "New Orleans"
              1:            {…}
```

- The tokens in search results that match with the keyword are now highlighted.




- Search is now included as perspective actions and workflow tasks.

### *Bug fixes*

- **[30399]** Searching on a foreign key field in a multi-valued group returns no results after changing the EBX® Information Search Add-on configuration.
- **[30400]** An error message is displayed when searching with the Date data type using REST service.
- **[30531]** The label and description of records in the **Algorithm** table cannot be updated.

## 17.35 **Release Note 2.0.5**

**Release Date: February 28, 2018**

### *New features*

- A new method that allows you to scan all of a table's records when using a custom search algorithm has been added to SearchComparator in the API.

### *Bug fixes*

- **[31003]** You cannot use a search keyword that contains numeric values when using the custom distance algorithm.

## 17.36 **Release Note 2.0.4**

**Release Date: January 31, 2018**

### *New features*

This release includes support for new features and bug fixes in Match and Cleanse Add-on 1.13.0.

### *Bug fixes*

- **[30440]** A search on a dataset cannot be completed when searching on two tables with foreign keys referring to each other.

## 17.37 **Release Note 2.0.3**

**Release Date: December 22, 2017**

### *New features*

This release contains the following new features:

- Support of case sensitive search has been added for all Match and Cleanse Add-on algorithms.

## 17.38 **Release Note 2.0.2**

**Release Date: December 15, 2017**

### *New features*

This release contains the following new features:

- Support for global search in the TIBCO EBX™ GO Add-on.
- Memory usage when loading the cache has been optimized.

## 17.39 **Release Note 2.0.1**

**Release Date: November 10, 2017**

### *Bug fixes*

- **[29359]** Bugs fixed in Match and Cleanse Add-on 1.12.0.

## 17.40 **Release Note 2.0.0**

**Release Date: October 16, 2017**

### *New features*

- **[22743]** It is now possible to modify pre-built algorithm parameters.
- **[22768]** You can now perform a search at table level.
- **[26295]** Search has been optimized to remove irrelevant results.
- **[26551]** There is a new search UI and search result format.
- **[26555]** You can now store/display search history.
- **[26557]** You now have the ability to filter search results.

- **[26559]** A new service allows you to purge the search history.
- **[26640]** The **Maximum No. of results** property in the **Model** table has been updated to apply at the record level.
- **[26663]** Some tables and fields that are no longer used were removed from search configuration.
- **[26849]** The add-on no longer supports relational models.
- **[26857]** Search performance has been optimized by using a cache mechanism.
- **[27705]** Searching with special characters is not supported.
- **[27954]** A new API has been added to preload the cache.

### *Bug fixes*

- **[26468]** Searching in a list returns an incorrect score.
- **[27561]** An association field can be registered with EBX® Information Search Add-on.
- **[27956]** A blank page is displayed when searching with the EBX® table filter and **Check permissions** deactivated.
- **[28520]**An error occurs when performing a search using REST service on a multi-valued field that is a number data type.

## 17.41 **Release Note 1.7.5 fix 001**

**Release Date: August 2, 2017**

### *New features*

- **[24249]** When searching via the API, you can get the schema node that leads to the record being returned in the result.

## 17.42 **Release Note 1.7.5**

**Release Date: April 18, 2017**

### *Optimization*

- **[25012]** Search performance on surrogate fields was improved.

### *Bug fixes*

- **[25008]** The tab in the search UI is not selectable if a table label contains special characters.

## 17.43 **Release Note 1.7.4**

**Release Date: March 20, 2017**

### *Bug fixes*

- **[24855]** The Preview button does not work when including EBX® Information Search Add-on in a specific portal.

# 17.44 Release Note 1.7.3

**Release Date: January 23, 2017**

### *New features*

- **[23695]** A new surrogate field matching was added.

### *Bug fixes*

- **[23885]** Soap WS fails when search results do not contain additional scores.

# 17.45 Release Note 1.7.2

**Release Date: December 16, 2016**

### *Optimization*

- **[23528]** Search performance for the predicate filter has been improved.

### *Bug fixes*

- **[23252]** Null value options are not taken into account during relational matching.

# 17.46 Release Note 1.7.1

**Release Date: November 18, 2016**

### *New features*

- **[23005]** Crosswalk search results are updated when searching with multiple target tables.

### *Optimization*

- **[23061]** Data indexing has been optimized for lower memory usage.

# 17.47 Release Note 1.7.0

**Release Date: October 12, 2016**

### *New features*

- **[14900]** An option is available to search in fields via the API.
- **[17904]** All the search algorithm parameters now have clear labels and descriptions.
- **[18048]** It is possible to search using REST.
- **[20111]** Japanese search algorithms are now available.
- **[20747]** Synonym management functionality was added to enhance search flexibility.
- **[21541]** The table filter is automatically applied when searching at table level.
- **[22216]** Search now can be initiated outside a Procedure.

### *Optimization*

- **[21995]** Search performance on multiple fields was improved.
- **[22107]** Indexing data was optimized to improve search performance.

### *Bug fixes*

- **[21690]** A JavaScript error is displayed when choosing 'Metadata' option in the Search box.
- **[21695]** Incorrect behavior in the 'Keyword' table when searching with keyword is long text and contain spaces.

## 17.48 **Release Note 1.6.5**

**Release Date: September 9, 2016**

### *New features*

- **[21415]** Null values are now taken into account when executing matching with any algorithms.
- **[21416]** It is possible to match on multiple fields in crosswalk.

## 17.49 **Release Note 1.6.4**

**Release Date: August 4, 2016**

### *Bug fixes*

- **[21363]** Searching a foreign key field in an external dataset using a combination of programmatic label and Levenshtein, JaroWinkler, or NGram algorithm returns an incorrect result.

## 17.50 **Release Note 1.6.3**

**Release Date: July 8, 2016**

### *Bug fixes*

- **[21045]** Incorrect search result when searching on inherited fields or computed value fields.

## 17.51 **Release Note 1.6.2**

**Release Date: June 10, 2016**

### *Bug fixes*

- **[20678]** When performing a metadata search, the web view was unable to display long table labels.
- **[20679]** The search UI did not display the horizontal scroll-bar for the table and field lists when their labels were excessively long.

# 17.52 Release Note 1.6.1

**Release Date: May 19, 2016**

### *Optimization*

Improve performance by enabling permission checking in the search configuration.

# 17.53 Release Note 1.6.0

**Release Date: April 13, 2016**

### *New features*

- Instead of using a dataspace and dataset to point to a data model (that you want to enable EBX® Information Search Add-on on) you only need to specify the data model.
- It is now possible to make changes to algorithm parameters.
- Searches on fields that are integer, decimal, date or date time data types are allowed.
- A primary and a secondary algorithm can be used in a search.
- The 'Best word' option is integrated in distance searches such as: Levenshtein and Jarowinkler.

### *New updates*

- The default value for the 'Search criteria on data' table's 'Check permission' property is now 'Yes'.

### *Bug fixes*

- **[14201]** The HTML values are saved in the 'Result' table located under 'EBX® Information Search Add-on configuration'.
- **[19714]** The field's label is not displayed in the result when the foreign key field is one of the fields being searched.
- **[19787]** A waiting prompt occurs when searching on the hidden dataset.
- **[19810]** The table label does not change in the search panel when changing the label's table in Information Governance Add-on.
- **[19852]** Search cannot be performed on an enumeration field.
- **[19864]** Searching on an enumeration field defined by the 'Label' field value does not work correctly.
- **[19996]** There is a performance problem when searching with distance algorithms.

# 17.54 Release Note 1.5.4

**Release Date: February 26, 2016**

### *New features*

Access rules can now be taken into consideration when searching.

### *Bug fixes*

- **[19428]** An exception is raised when more than one search executes simultaneously on multiple fields.
- **[19429]** A JavaScript error occurs when searching in the Internet Explorer 8 browser.

## 17.55 **Release Note 1.5.3 fix 001**

**Release Date: February 4, 2016**

### *Bug fixes*

- **[19191] Records with scores lower than the min score threshold are returned.**
- **[19293]** The redundant '&' character is displayed in the generated URL when searching in the Web view.

## 17.56 **Release Note 1.5.3**

**Release Date: January 18, 2016**

### *Bug fixes*

- **[18486]** A runtime exception occurs when running the table filter on EBX®'s search panel.
- **[18061]** Invalid format warnings for WSDL file.
- **[19018]** An incorrect score is returned when searching on more than one field and when a field has a score lower than the minScore threshold.

## 17.57 **Release Note 1.5.2**

**Release Date: January 7, 2016**

### *Bug fixes*

- **[18757]** Incorrect score is returned when the search keyword contains special characters.

## 17.58 **Release Note 1.5.1**

**Release Date: November 3, 2015**

### *New features*

In case you need to ignore some characters, words in the information used to search, you can now configure filters as direct values or through a custom Java class.

### *New API*

It is now possible to set an XPath filter in the SearchContext API in order to reduce the scope of the search.

# 17.59 Release Note 1.5.0

**Release Date: October 9, 2015**

### *New features*

- When searching via API or Web Service you can specify whether to store results.
- You can execute a search on a multi-value field inside a terminal group.
- You can execute a search on a multi-value field inside a multi-value group.

### *Bug fixes*

- **[17565]** A search returns an empty result when performed on a foreign key field using a programmatic label.

### *Optimization*

Searching process has been optimized.

# 17.60 Release Note 1.4.3

**Release Date: September 3, 2015**

### *Bug fixes*

- **[17503]** Searching simple fields under a terminal group may return an incorrect score.
- **[17504]** Searching simple fields under a multi-value group returns an incorrect score.

### *Limitations*

- Searching based on a multi-value field inside a terminal group is not supported.
- Searching based on a multi-value field inside a multi-value group is not supported.

# 17.61 Release Note 1.4.2

**Release Date: July 31, 2015**

### *New features*

Accept filter with predicate when searching in group.

### *New API*

It is now possible to retrieve the search configuration associated to a dataset.

# 17.62 Release Note 1.4.1

**Release Date: June 29, 2015**

### *New features*

You can now define a custom distance algorithm on any simple type.

## 17.63 **Release Note 1.4.0**

**Release Date: June 10, 2015**

### *New features*

- EBX® Information Search Add-on can now be used to extend the EBX® table filter. It allows user to apply the fuzzy search from the table search UI component.
- A new web service allows user to expose searching.

## 17.64 **Release Note 1.3.2**

**Release Date: May 6, 2015**

### *Bug fixes*

- **[15943]** In case of many similarities between more records than the number of records searched for (for instance the cluster size in Matching), results can be inconsistent. It is especially true in funneling mode.

## 17.65 **Release Note 1.3.1**

**Release Date: April 15, 2015**

### *API*

- Fix the API in order to be used within a trigger or a procedure.

## 17.66 **Release Note 1.3.0**

**Release Date: January 26, 2015**

### *New features*

- When deleting a configuration in the 'Search criteria on data', the system now applies a delete 'on cascade' of all the related records in the 'Table' and 'Field' tables. This change simplifies the administration process of removing a search configuration.
- The 'Similarity' and 'Prefix length' properties are now configured only when the 'FuzzyFullText' algorithm is selected.
- The search result UI is improved and allows you to preview records.

## 17.67 **Release Note 1.2.5**

**Release Date: January 22, 2015**

### *New features*

- Add the possibility to define filters to normalize values when searching.

## 17.68 **Release Note 1.2.4**

**Release Date: January 19, 2015**

### *Bug fixes*

- **[14213]** The option best word is not taken into account.

## 17.69 **Release Note 1.2.3**

**Release Date: November 10, 2014**

### *New features*

- Add Chinese search algorithm.

## 17.70 **Release Note 1.2.2**

**Release Date: October 10, 2014**

### *Bug fixes*

- **[12833]** The configuration of the search is not unique for a same data model.
- **[12568]** The creation of new record in Result table of TIBCO EBX™ Information Search Add-on dataspace is enabled.

### *Optimization*

Update the display of fields in configuration.

## 17.71 **Release Note 1.2.1**

**Release Date: September 12, 2014**

### *Bug fixes*

- **[12526]** Exception when searching with null foreign key using 'Exact algorithm'.

## 17.72 **Release Note 1.2.0**

**Release Date: August 29, 2014**

### *New features*

It is now possible to create custom distance algorithms. See **Java API** `SearchDistance`[API] for more information. It can be used only in Match and Cleanse Add-on.

### *Optimization*

The performances have been improved.

## 17.73 **Release Note 1.1.1**

**Release Date: June 19, 2014**

### *Optimization*

Some drastic improvement of the performances have been done.

## 17.74 **Release Note 1.1.0**

**Release Date: June 3, 2014**

### *New features*

It is now possible to search a 'best word' among a long text such as a description. With this option, the score of the search is based on the sole matching against the search criteria. It is no longer the average score based on the number of words in the targeted text.

## 17.75 **Release Note 1.0.1 fix 001**

**Release Date: April 7, 2014**

### *Bug fixes*

Search panel background color does not follow the 'Colors and themes' from the general preferences.

## 17.76 **Release Note 1.0.0**

**Release Date: Dec 19, 2013**

EBX® Information Search Add-on allows retrieving data anywhere in the repository. The search is multi-table-oriented. This means that the result of a search query is a collection of data sourced from one or multiple tables. The ability to search over a set of tables allows querying a large scope of information, similar to a web search engine.

The search query is based on a fuzzy execution retrieves information even if the search criteria do not match the data exactly. For example, the search criterion 'John' will return information such as 'Jone', 'Johnson', 'Cohn' depending on the configuration of the search engine. This configuration allows for tuning the search to avoid unwanted results and for deciding which fuzzy search policies to apply (phonetics, distance).

EBX® Information Search Add-on allows for retrieving information not only on data value of the repository but also on meta-data, namely the data models and workflow descriptions.

### *Search scope*

All tables in a dataset.

### *Search algorithms*

Phonetics and distance.

### *UI results*

Generic tabular view and web view.